Causally Structured Differential Network Modeling for Single-Cell Perturbation Prediction

Jiayi Dong[1,2] **Xinyue Gu**[1,2] **Fei Wang**[1,2,3*]
[1]College of Computer Science and Artificial Intelligence, Fudan University
[2]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
[3]Shanghai Academy of AI for Science
Shanghai, China
wangfei@fudan.edu.cn

Abstract

Predicting cellular responses to perturbations can support biological research and therapeutic discovery. Existing models predict perturbation-induced changes in gene expression, but often leave changes in the underlying regulatory dependencies implicit. To address this gap, we propose scPertCRL, a causally structured generative framework for single-cell perturbation prediction. scPertCRL models perturbation effects through condition-specific structural modulations and exogenous shifts around a shared latent structural backbone. Biologically informed perturbation embeddings condition low-rank structural updates, while differential-expression supervision guides the joint representation of structural changes and exogenous shifts. Experiments on genetic and pharmacological perturbation benchmarks show improvements in predictive performance and generalization to unseen conditions, while providing model-based insights into perturbation-associated structural changes.

1 Introduction

Cellular perturbation assays have become central to functional genomics and systems biology, enabling systematic interrogation of gene regulatory systems through genetic or pharmacological interventions [1, 2]. Recent advances in high-throughput single-cell perturbation sequencing now allow cellular responses to be profiled at unprecedented scale and resolution [3]. These advances have motivated growing interest in computational models that predict cellular responses to unseen perturbations, a key task for understanding regulatory mechanisms and accelerating downstream applications such as drug discovery [4].

Despite recent progress, perturbation prediction remains challenging because interventions can alter regulatory dependencies among genes, while existing prediction models often leave these condition-specific structural changes implicit [5]. Existing approaches include embedding-fusion frameworks [6], knowledge-guided methods [7], and foundation-model-based strategies [8]. While effective empirically, these methods typically treat perturbations as modifiers of latent representations, leaving the underlying mechanism-level variation implicit, which may limit robustness under complex or unseen conditions.

From a modeling perspective, perturbations can be naturally viewed as changes to the data-generating process. This motivates the use of structured models that explicitly parameterize how dependencies among variables vary across conditions. Prior work in causal representation learning introduces structural assumptions into latent-variable models to improve robustness and interpretability [9–12].

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, approaches that assume a fixed dependency structure across conditions may be restrictive when perturbations alter these dependencies, while estimating a separate graph for each condition does not explicitly exploit shared structure across conditions.

From a biological standpoint, genetic or pharmacological interventions often act on specific targets and pathways, with downstream effects propagating through cellular regulatory systems [13, 14]. This motivates a mechanism-centered modeling assumption in which perturbation conditions share a background dependency structure while allowing structured, condition-specific deviations. Such a decomposition supports information sharing across conditions and provides a basis for compact parameterization without requiring a separate global model for each perturbation.

Building on this perspective, we propose scPertCRL, a causally structured generative framework for single-cell perturbation prediction. Our approach uses a Structural Causal Model (SCM)-inspired parameterization as an inductive bias to model a shared backbone together with perturbation-specific structural changes and exogenous shifts. Rather than aiming to recover the true underlying causal graph, the model leverages structured parameterization to capture mechanism-level variation in a compact and interpretable way. By combining variational inference with low-rank structural updates conditioned on biologically informed perturbation embeddings, scPertCRL learns low-complexity representations of perturbation effects that support out-of-distribution generalization. Experiments on genetic and pharmacological perturbation datasets demonstrate that scPertCRL improves predictive accuracy and generalization to unseen perturbations, while also providing model-based insights into perturbation-associated structural changes. Our contributions are summarized as follows:

- We formulate single-cell perturbation prediction by modeling condition-specific structural changes and exogenous shifts relative to a shared latent backbone, explicitly representing perturbation-associated variation in the generative process.
- We develop scPertCRL, a causally structured generative framework integrating variational inference with low-rank structural updates conditioned on biologically informed perturbation embeddings. Differential-expression supervision guides the joint representation of structural changes and exogenous shifts.
- We evaluate scPertCRL on genetic and pharmacological perturbation benchmarks, demonstrating improved prediction and generalization to unseen perturbations and cellular contexts. Complementary analyses examine the learned structural changes through expression correlations and biological references.

2 Related Work

2.1 Single-Cell Perturbation Prediction

Single-cell perturbation prediction aims to predict post-perturbation expression profiles from control-cell data and perturbation information. Existing methods can be examined along two dimensions: how they represent perturbations and how they incorporate these representations into response prediction.

Perturbation Representation. A prominent line of work incorporates external biological knowledge into perturbation representations. For instance, GEARS [7] and GraphVCI [15] employ graph neural networks over protein-protein interaction or co-expression networks to propagate perturbation signals. GRAPE [16] enriches gene-level perturbation representations using pretrained biological sequence and text models [17, 18]. PRESAGE [19] integrates pathway information, textual descriptions, and transcriptional data to learn perturbation representations for prediction under unseen perturbations.

Conditioning Mechanisms. Existing models differ in how they incorporate perturbation information into response prediction. Latent-variable approaches use conditional decoding or latent-space operations: scVI [20] incorporates observed covariates through a conditional decoder, scGen [21] transfers perturbation effects through latent-space vector arithmetic, and CPA [6] combines basal cell states with additive perturbation and covariate embeddings. Transformer-based approaches employ different conditioning strategies: scGPT [8] incorporates gene-level perturbation indicators into its input embeddings, STATE [22] combines perturbation information with control-cell representations to predict responses across cell sets, and STACK [23] conditions cell-state generation on reference-cell

prompts. These approaches generally do not explicitly parameterize condition-specific changes in the underlying dependency structure.

In contrast, scPertCRL goes beyond conditioning cell representations by using perturbation embeddings to explicitly parameterize structural changes and exogenous shifts relative to a shared latent backbone.

2.2 Causal Modeling for Perturbation Analysis

Biological perturbations can alter the data-generating process, motivating structured models of dependencies among cellular variables. Some approaches introduce graph-based or structural constraints to reconstruct gene regulatory networks [24]. A related question is how perturbations alter the underlying mechanisms and resulting cellular responses. DiscrepancyVAE [10] models interventions on latent causal mechanisms and propagates their effects through a shared latent structure. SENA-Discrepancy-VAE [25] extends this framework with a pathway-informed encoder, linking latent representations to biological processes. PDAE [26] represents perturbations as additive mean shifts in latent space and studies when these shifts support extrapolation to unseen combinations. From a differential-network perspective, CDN [27] uses differences between inferred observational and interventional structures, together with statistical features, to identify intervention targets.

More generally, causal representation learning investigates how interventions and changes across environments can help identify latent causal representations [28]. Liu et al. [29, 11] establish identifiability results by exploiting changes in latent causal influences across environments, extending this perspective from linear-Gaussian to polynomial causal models. GIM [12] connects perturbation features to distributions over interventions in a jointly learned causal model, enabling prediction for unseen perturbations. These studies motivate explicitly representing mechanism variation when modeling perturbation responses. Within this line of work, scPertCRL explicitly separates perturbation-induced changes in latent dependencies from exogenous shifts, using a shared structural backbone as the reference for response prediction.

3 Method

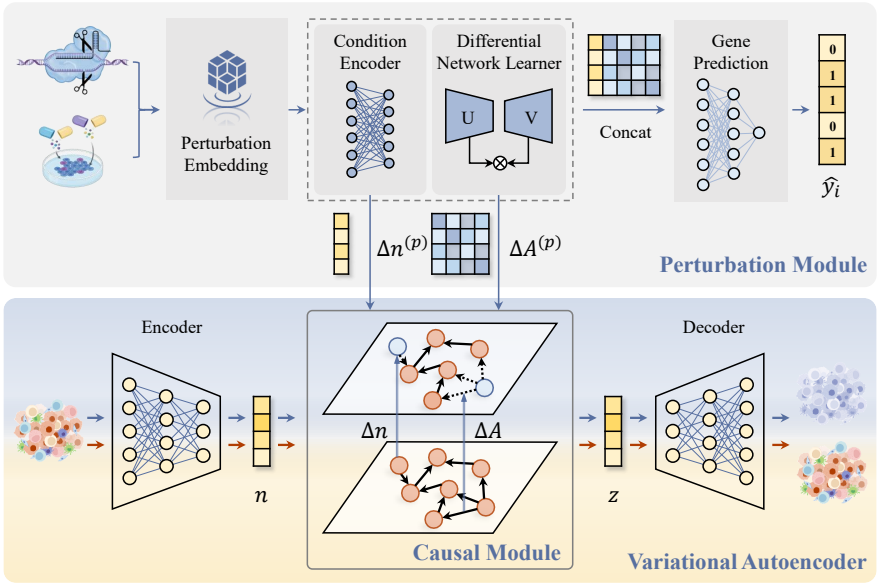


Figure 1: Overview of scPertCRL. The Causal Module models a shared regulatory backbone together with perturbation-specific mechanism changes. The VAE Module maps gene expression to latent exogenous variables and reconstructs cell states. The Perturbation Module captures differential information that supports prediction and interpretation of condition-specific cellular responses.

We develop a mechanism-centered generative framework for single-cell perturbation prediction. We model perturbation responses through condition-specific changes in latent dependencies and exogenous variation.

Let $\mathbf{x}^{(p)} \in \mathbb{R}^m$ denote the gene expression profile of a single cell under perturbation condition p, where $p = \emptyset$ denotes the control condition. Our goal is to map the control distribution $\mathbb{P}(\mathbf{x}^{(\emptyset)})$ and a target perturbation p to the cell-state distribution $\mathbb{P}(\mathbf{x}^{(p)})$.

scPertCRL combines varying Structural Causal Models (SCMs) with variational inference (Figure 1). The causal module models both stable and perturbation-specific mechanisms, the VAE module infers latent exogenous variables and reconstructs gene expression, and the perturbation module provides embeddings that condition structural changes and exogenous shifts.

3.1 Causally Structured Latent Model

We introduce latent regulatory variables $\mathbf{z}$ to represent gene-level regulatory states underlying observed expression $\mathbf{x}$. Under the control condition, we parameterize these latent variables using a generative process inspired by SCMs:

$$z_i = f_i\big(\mathbf{z}_{\mathrm{pa}(i)}, n_i\big), \quad i = 1, \ldots, d, \tag{1}$$

where $\mathrm{pa}(i)$ denotes the index set of parent variables and $\mathbf{n}$ represents independent exogenous variation. For tractability, we use a linear-Gaussian parameterization within the latent space:

$$z_i = \sum_{j \neq i} A_{ij} z_j + n_i, \quad n_i \sim \mathcal{N}(0, \sigma_i^2). \tag{2}$$

Here, $\mathbf{A}$ defines a shared structural backbone representing model-dependent baseline dependencies within the selected gene panel.

Under perturbation condition p, interventions may change both structural coefficients and exogenous noise. The latent variables are then generated by a varying SCM:

$$z_i^{(p)} = \sum_{j \neq i} A_{ij}^{(p)} z_j^{(p)} + n_i^{(p)}. \tag{3}$$

We represent perturbation effects through condition-specific structural and exogenous changes:

$$\mathbf{A}^{(p)} = \mathbf{A} + \Delta\mathbf{A}^{(p)}, \quad \mathbf{n}^{(p)} = \mathbf{n} + \Delta\mathbf{n}^{(p)}. \tag{4}$$

Here, $\Delta\mathbf{A}^{(p)}$ and $\Delta\mathbf{n}^{(p)}$ represent perturbation-specific structural changes and exogenous shifts, respectively. This decomposition is motivated by the biological intuition that perturbations affect specific targets and pathways against a largely shared regulatory background.

3.2 Variational Inference

To infer the latent variables introduced above, we use a VAE framework. The latent variables $\mathbf{z}$ are obtained from the exogenous variables $\mathbf{n}$ through the structural propagation defined by the model. We therefore perform variational inference over $\mathbf{n}$, whose components have an independent prior. For control samples, we optimize the Evidence Lower Bound (ELBO):

$$\mathcal{L}_{\mathrm{ELBO}}^{(\emptyset)} = \mathbb{E}_{q_\phi(\mathbf{n}|\mathbf{x}^{(\emptyset)})}\big[\log p_\theta(\mathbf{x}^{(\emptyset)} \mid \mathbf{z})\big] - D_{\mathrm{KL}}\big(q_\phi(\mathbf{n} \mid \mathbf{x}^{(\emptyset)}) \,\|\, p(\mathbf{n})\big), \tag{5}$$

where $\mathbf{z}$ is obtained from $\mathbf{n}$ through the shared structural model. Under perturbation p, the exogenous noise is modeled as the sum of an intrinsic component $\mathbf{n}$ and a perturbation-conditioned shift $\Delta\mathbf{n}^{(p)}$. We use standard normal priors for both the intrinsic component and the perturbation-induced shift. The variational distribution of the shift is conditioned on the perturbation embedding. We use a factorized variational approximation:

$$q_\phi(\mathbf{n}, \Delta\mathbf{n}^{(p)} \mid \mathbf{x}^{(\emptyset)}, p) = q_\phi(\mathbf{n} \mid \mathbf{x}^{(\emptyset)})\, q_\phi(\Delta\mathbf{n}^{(p)} \mid p). \tag{6}$$

Here, $q_\phi(\mathbf{n} \mid \mathbf{x}^{(\emptyset)})$ is parameterized by the cell encoder to capture cell-specific intrinsic variation, while $q_\phi(\Delta\mathbf{n}^{(p)} \mid p)$ is parameterized by a perturbation-conditioned network to capture condition-dependent shifts. Cells under the same perturbation condition share the parameters of the shift

distribution. Conditioning the shift distribution on the perturbation provides a shared description of perturbation effects across unpaired cells. Cell-specific responses can still arise from differences in intrinsic states and sampled shifts, propagated through the structural model and nonlinear decoder.

If paired observations were available, an idealized conditional ELBO could be written as

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}}^{(p)} =& \mathbb{E}_{q_\phi(\mathbf{n}|\mathbf{x}^{(\emptyset)})q_\phi(\Delta\mathbf{n}^{(p)}|p)}\left[\log p_\theta\left(\mathbf{x}^{(p)}|\mathbf{z}^{(p)}\right)\right] \\ &- D_{\mathrm{KL}}\left(q_\phi(\mathbf{n}|\mathbf{x}^{(\emptyset)})\|p(\mathbf{n})\right) - D_{\mathrm{KL}}\left(q_\phi(\Delta\mathbf{n}^{(p)}|p)\|p(\Delta\mathbf{n}^{(p)})\right). \end{aligned} \tag{7}$$

However, single-cell perturbation measurements are typically unpaired: a control cell does not have a directly observed counterpart after perturbation. The paired expression is therefore an explanatory idealization, not the optimized objective. In the implementation, its pointwise reconstruction term is replaced by a set-level distributional surrogate comparing the generated perturbed cell set $\hat{\mathbf{X}}^{(p)}$ with the observed perturbed cell set $\mathbf{X}^{(p)}$:

$$\mathbb{E}_{q_\phi(\mathbf{n}|\mathbf{x}^{(\emptyset)})q_\phi(\Delta\mathbf{n}^{(p)}|p)}\left[\log p_\theta\left(\mathbf{x}^{(p)}|\mathbf{z}^{(p)}\right)\right] \rightsquigarrow -D\left(\hat{\mathbf{X}}^{(p)}, \mathbf{X}^{(p)}\right). \tag{8}$$

3.3 Mechanism Modeling via Differential Structure

Structure learning has two parts: estimating the baseline adjacency from control cells and parameterizing condition-specific deviations from perturbation embeddings.

Base Structure Learning. We begin by estimating the shared adjacency matrix $\mathbf{A}$ using control data through the continuous optimization framework [30, 31]. Acyclicity is characterized by the following condition:

$$h(\mathbf{A}) = \mathrm{tr}\big(e^{\mathbf{A}\circ\mathbf{A}}\big) - d = 0. \tag{9}$$

Following prior work [30, 31], we encourage acyclicity in the shared backbone through a penalty on $h(\mathbf{A})$. The resulting backbone captures baseline dependencies in control cells and serves as the reference structure for subsequent perturbation-specific updates.

Differential Network Learning. The Differential Network Learner takes a perturbation embedding $e^{(p)}$ as input and maps it to a structural offset. Gene perturbations are represented using Qwen embeddings of gene descriptions [32], while drug perturbations use ChemBERTa embeddings of SMILES strings [33]; details are provided in Appendix D.5. Rather than learning an independent dense adjacency matrix for each perturbation, scPertCRL uses two perturbation-conditioned neural maps:

$$\Delta\mathbf{A}^{(p)} = \mathbf{U}(e^{(p)})\mathbf{V}(e^{(p)}). \tag{10}$$

Here, $\mathbf{U}(\cdot)$ and $\mathbf{V}(\cdot)$ map $e^{(p)}$ to $\mathbf{U}(e^{(p)}) \in \mathbb{R}^{d\times r}$ and $\mathbf{V}(e^{(p)}) \in \mathbb{R}^{r\times d}$, where $r \ll d$, and their product gives $\Delta\mathbf{A}^{(p)}$. The low-rank factorization provides an efficient parameterization: it reduces perturbation-specific structural parameters, shares latent factors across correlated mechanism changes, and supports embedding-conditioned generalization. Computational complexity is discussed in Appendix C.

3.4 Differentially Expressed Gene Prediction

In addition to predicting global expression profiles, scPertCRL predicts differentially expressed genes (DEGs) from structural changes and exogenous shifts. For each gene i, we construct a mechanism-shift feature vector $\mathbf{g}_i^{(p)}$ by concatenating its incoming structural changes and exogenous shift:

$$\mathbf{g}_i^{(p)} = \left[\Delta\mathbf{A}_{i,\cdot}^{(p)},\ \Delta n_i^{(p)}\right], \quad \hat{y}_i = \sigma\Big(f(\mathbf{g}_i^{(p)})\Big), \tag{11}$$

where $\Delta\mathbf{A}_{i,\cdot}^{(p)}$ captures changes in the modeled dependencies affecting gene i, and $\Delta n_i^{(p)}$ represents its perturbation-conditioned exogenous shift. For classification, we use the posterior mean of $\Delta n_i^{(p)}$. Together, these features describe perturbation-specific changes in the dependency structure and exogenous inputs associated with gene i. We use a weight-shared feed-forward classifier f to map

each feature vector to the probability that the gene is differentially expressed. Sharing the classifier across genes imposes the same decision rule for all genes and helps reduce gene-specific overfitting.

Following STAMP [34], we supervise this gene classifier using DEG labels derived from the training data via a class-weighted binary cross-entropy loss:

$$\mathcal{L}_{\text{diff}} = \sum_i \text{BCE}(y_i, \hat{y}_i). \tag{12}$$

During training, we add noise to perturbation embeddings to encourage smooth condition-dependent mappings and reduce overfitting. DEG supervision constrains the structural-change and exogenous-shift features through the shared classifier, rather than acting directly on expression outputs.

3.5 Optimization and Training Strategy

The overall objective combines generative modeling, structural regularization, and differential-expression supervision. The full fine-tuning objective is

$$\mathcal{L} = -\lambda_{\text{ELBO}}\mathcal{L}_{\text{ELBO}} + \lambda_{\text{str}}\mathcal{L}_{\text{str}} + \lambda_{\text{diff}}\mathcal{L}_{\text{diff}}. \tag{13}$$

Each loss term serves a distinct role. $\mathcal{L}_{\text{ELBO}}$ fits control cells with the standard ELBO objective and matches perturbed cell sets through distribution alignment in the unpaired branch. $\mathcal{L}_{\text{str}}$ encourages acyclicity in $\mathbf{A}^{(p)}$ and applies L_1 and squared L_2 penalties to $\Delta\mathbf{A}^{(p)}$, promoting sparse structural changes and controlling their magnitude. $\mathcal{L}_{\text{diff}}$ provides gene-level supervision that aligns the learned mechanism-shift features with observed DEG labels.

Training proceeds in two stages (Appendix A) to separate shared baselines from condition-specific effects. In the first stage, control cells are used to learn the encoder, decoder, and shared structural backbone $\mathbf{A}$. In the second stage, perturbation data are introduced to learn $\Delta\mathbf{A}^{(p)}$, $\Delta\mathbf{n}^{(p)}$, and the gene predictor under the full objective. The shared structural parameter $\mathbf{A}$ is frozen, while the encoder, decoder, and perturbation-specific modules remain trainable. This design helps reduce conflation between baseline dependencies and perturbation-specific changes while fitting the observed perturbation distributions.

3.6 Identifiability and Limitations

The parameterization of scPertCRL is motivated by identifiability results for varying causal models [11]. Appendix B relates our design to these theoretical conditions. In practical single-cell settings, the available intervention diversity and the learned encoder-decoder mappings are not guaranteed to satisfy the assumptions required for strict identifiability. The linear-Gaussian latent SCM also serves as a tractable approximation to latent structural variation, without assuming that biological regulation is globally linear. We therefore use the causal equations as a structural inductive bias for perturbation prediction and interpret the learned differential matrices as model-dependent summaries of perturbation-associated mechanism changes within the selected gene panel, rather than recovered ground-truth causal graphs.

4 Experiments

We evaluate scPertCRL in terms of predictive accuracy, generalization, and the interpretive value of its learned structural representations. Specifically, we ask: (i) does it improve predictive accuracy, (ii) does it generalize to unseen perturbations and cellular contexts, and (iii) do its learned features support DEG prediction and provide interpretable summaries of perturbation-associated structural changes?

4.1 Settings

We evaluate scPertCRL on five single-cell perturbation benchmarks covering both genetic and chemical interventions: RPE1, K562 [3], Norman [35], Sci_Plex_Combo [2], and PBMC [36]. We evaluate generalization to held-out single-gene perturbations, held-out perturbation combinations, and held-out cell types, with detailed data splits provided in Appendix D.

Table 1: Performance comparison across single-gene perturbation datasets. Best results are shown in bold and second-best results are underlined.

Method	RPE1			K562		
	PDS ↑	DEG Overlap ↑	Pearson Delta ↑	PDS ↑	DEG Overlap ↑	Pearson Delta ↑
Control	0.5041 ± 0.0004	0.0000 ± 0.0000	0.0000 ± 0.0000	0.5076 ± 0.0010	0.0000 ± 0.0000	0.0000 ± 0.0000
Pert	0.5031 ± 0.0000	**0.4116 ± 0.0162**	0.6418 ± 0.0224	0.5049 ± 0.0000	0.2213 ± 0.0084	0.3840 ± 0.0148
CPA	0.5039 ± 0.0022	0.2446 ± 0.0092	0.4656 ± 0.0116	0.5193 ± 0.0059	0.1927 ± 0.0065	0.3301 ± 0.0078
DiscrepancyVAE	0.5763 ± 0.0152	0.3676 ± 0.0250	0.4516 ± 0.1217	0.5900 ± 0.0211	0.2031 ± 0.0112	0.1758 ± 0.0615
GEARS	0.5431 ± 0.0270	0.3224 ± 0.0118	0.5439 ± 0.1105	0.5879 ± 0.0202	0.1897 ± 0.0099	0.3327 ± 0.0895
scGPT	0.5725 ± 0.0107	0.4072 ± 0.0124	0.6860 ± 0.0134	0.5870 ± 0.0099	0.2436 ± 0.0181	0.4888 ± 0.0090
scVI	0.5284 ± 0.0116	0.2780 ± 0.0166	0.6030 ± 0.0281	0.5538 ± 0.0180	0.2021 ± 0.0077	0.4154 ± 0.0176
STATE	0.5483 ± 0.0167	0.3018 ± 0.0283	0.6398 ± 0.0364	0.6028 ± 0.0054	0.1455 ± 0.0116	0.2352 ± 0.0138
scPertCRL	**0.6162 ± 0.0158**	0.4062 ± 0.0147	**0.7038 ± 0.0147**	**0.6634 ± 0.0068**	**0.2803 ± 0.0066**	**0.5407 ± 0.0207**

Table 2: Performance comparison on combinatorial perturbation prediction tasks.

Method	Norman		
	PDS ↑	DEG Overlap ↑	Pearson Delta ↑
Control	0.5059 ± 0.0005	0.0033 ± 0.0003	0.0000 ± 0.0000
Pert	0.5045 ± 0.0002	0.0428 ± 0.0019	0.4621 ± 0.0321
CPA	0.5743 ± 0.0224	0.0728 ± 0.0043	0.3846 ± 0.0231
DiscrepancyVAE	0.6284 ± 0.0614	0.0654 ± 0.0082	0.3946 ± 0.0639
GEARS	0.6983 ± 0.0321	0.1146 ± 0.0046	0.3808 ± 0.0655
scGPT	0.6359 ± 0.0273	0.0586 ± 0.0152	0.5304 ± 0.0105
scVI	0.6837 ± 0.0154	0.0499 ± 0.0063	0.4579 ± 0.0293
STATE	0.6454 ± 0.0431	0.2913 ± 0.0204	0.3003 ± 0.0481
scPertCRL	**0.6996 ± 0.0391**	**0.3661 ± 0.0271**	**0.5782 ± 0.0308**

Method	Sci_Plex_Combo		
	PDS ↑	DEG Overlap ↑	Pearson Delta ↑
PRNet	0.7885 ± 0.1305	0.6661 ± 0.0560	0.5492 ± 0.1443
Control	0.6200 ± 0.0112	0.0001 ± 0.0001	0.0000 ± 0.0000
CPA	0.6645 ± 0.0599	0.6038 ± 0.0790	0.5577 ± 0.0503
scVI	0.7555 ± 0.0829	0.6350 ± 0.0723	0.6042 ± 0.0823
STATE	0.8170 ± 0.1382	**0.7177 ± 0.0690**	0.7800 ± 0.1282
scPertCRL	**0.8694 ± 0.0462**	0.6976 ± 0.0495	**0.7861 ± 0.1766**

Following [22], we use complementary metrics capturing different aspects of prediction quality. **Pearson Delta** measures correlation between predicted and observed average perturbation effects across genes; it does not assess absolute effect magnitudes. **Perturbation Discrimination Score (PDS)** evaluates retrieval of the correct condition among candidate responses. **DEG Overlap** measures the overlap between DEG sets identified from generated and observed expression profiles. **Average Precision (AP)** evaluates how well the gene classifier ranks DEGs using its predicted scores.

We compare against diverse baselines, including statistical methods (Control, Pert), VAE-based models (scVI, CPA), causal representation learning approaches (DiscrepancyVAE), graph-based models (GEARS), and transformer-based methods (scGPT, STATE). For drug perturbations, we additionally include PRNet [37]. For DEG prediction, we compare with STAMP [34] and its variant STAMP*. Additional details are provided in Appendix D.

4.2 Predictive Performance on Unseen Perturbations

We first evaluate generalization to unseen perturbations, examining whether scPertCRL predicts cellular responses to conditions not observed during training.

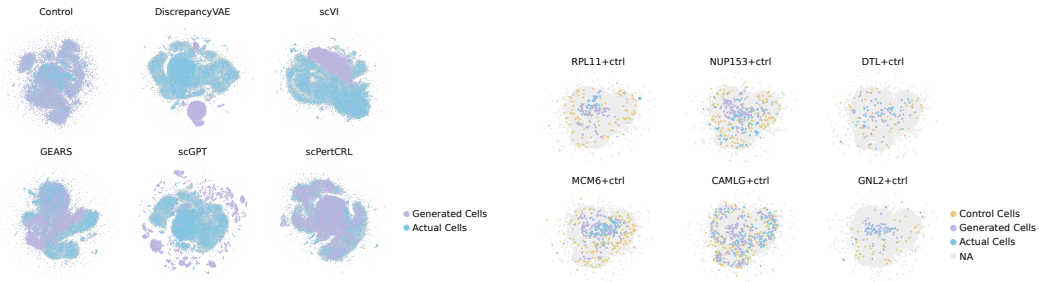
Single-gene Perturbations. Table 1 reports results on RPE1 and K562. On RPE1, scPertCRL achieves the highest mean Pearson Delta and PDS, while maintaining DEG Overlap close to the best-performing baselines. This indicates improved agreement with the pattern of average perturbation effects and better condition discrimination, alongside competitive DEG Overlap. On K562, scPertCRL achieves the highest mean scores across all three metrics. Together, these results demonstrate strong performance across the evaluated aspects of single-gene perturbation prediction, including average response patterns, condition discrimination, and DEG recovery.

Combinatorial Perturbations. We next evaluate generalization on Norman and Sci_Plex_Combo (Table 2). Norman uses a more complex setting with both held-out single-gene perturbations and gene combinations, while Sci_Plex_Combo evaluates held-out drug combinations. The combination tests assess prediction of joint responses that may differ from the sum of individual perturbation effects.

On Norman, scPertCRL achieves the highest mean Pearson Delta and DEG Overlap, with a PDS comparable to that of GEARS. These results indicate better agreement with average perturbation effects and DEG labels while maintaining comparable condition discrimination.

To examine prediction without external biological embeddings, we additionally evaluate held-out double-gene combinations whose constituent single-gene perturbations are observed during training. In this setting, scPertCRL with one-hot perturbation encoding achieves higher mean scores across all three metrics than its Qwen-based counterpart. These results show that gene-identity encoding alone can support prediction of unseen combinations of observed single-gene perturbations (Appendix E.1).

On Sci_Plex_Combo, scPertCRL achieves the highest mean PDS and Pearson Delta, while maintaining competitive DEG Overlap. Its stronger condition discrimination, together with accurate prediction of average perturbation-effect patterns, supports its ability to predict held-out drug combinations.



(a) UMAP comparison across methods.

(b) UMAP visualization of different conditions.

Figure 2: UMAP visualization of control, actual perturbed, and generated perturbed cells in the RPE1 dataset.

Qualitative Analysis. Figure 2 summarizes the qualitative comparison on RPE1. Figure 2a compares generated and observed perturbed cells across methods, while Figure 2b presents selected condition-level examples. In these examples, scPertCRL generates cell populations that overlap with the observed perturbed cells in the UMAP visualizations. These comparisons provide a qualitative view of prediction agreement across methods and complement the quantitative metrics. Additional examples are provided in Appendix F.

4.3 Cross-Cell-Type Generalization on PBMC

Cross-cell-type prediction evaluates generalization to cellular contexts absent from training. We use leave-one-cell-type-out evaluation on PBMC, training on the remaining cell types and predicting responses in the held-out type under the same perturbations. Results in Table 3 show that scPertCRL achieves the highest mean Pearson Delta across all six held-out cell types, with particularly large gains on CD4+ and CD8+ T cells, B cells, and NK cells.

On CD4+ and CD8+ T cells, scPertCRL achieves Pearson Delta above 0.5, compared with values below 0.2 for all evaluated baselines. On B cells and NK cells, it achieves correlations of 0.3282 and 0.3687, respectively, while baseline scores remain near zero or negative. These results indicate improved agreement with observed average perturbation-effect patterns in held-out cell types and support the ability of scPertCRL to transfer response predictions across the evaluated cellular contexts.

Table 3: Pearson Delta on the PBMC dataset under leave-one-cell-type-out evaluation. Each column reports performance on one held-out cell type.

Method	T cells CD8+	B cells	Myeloid cells	T regulatory cells	NK cells	T cells CD4+
PRNet	0.1416 ± 0.0074	-0.0572 ± 0.0109	0.1590 ± 0.0093	$\underline{0.3148 \pm 0.0093}$	-0.0677 ± 0.0054	0.0446 ± 0.0209
CPA	0.1464 ± 0.0068	-0.0659 ± 0.0019	0.0924 ± 0.0052	0.1515 ± 0.0051	-0.0264 ± 0.0047	0.0358 ± 0.0030
Pert	0.0463 ± 0.0689	$\underline{0.0000 \pm 0.0274}$	0.0020 ± 0.0216	0.0310 ± 0.0602	$\underline{0.0058 \pm 0.0554}$	0.0309 ± 0.0863
scVI	0.1591 ± 0.0135	-0.0520 ± 0.0042	0.1243 ± 0.0209	0.2357 ± 0.0075	-0.0186 ± 0.0136	0.0748 ± 0.0104
STATE	$\underline{0.1820 \pm 0.0144}$	-0.0665 ± 0.0045	$\underline{0.1727 \pm 0.0047}$	0.2858 ± 0.0682	-0.0038 ± 0.0288	$\underline{0.1671 \pm 0.0157}$
scPertCRL	$\mathbf{0.5011 \pm 0.0091}$	$\mathbf{0.3282 \pm 0.0312}$	$\mathbf{0.2203 \pm 0.1017}$	$\mathbf{0.4795 \pm 0.0266}$	$\mathbf{0.3687 \pm 0.0581}$	$\mathbf{0.5470 \pm 0.0210}$

4.4 Differential Information Analysis

Table 4: DEG identification performance measured by AP.

Method	RPE1	K562	Norman
ESM2 + STAMP	0.4040 ± 0.0188	0.2294 ± 0.0101	0.5058 ± 0.0121
GENEPT + STAMP	0.4009 ± 0.0270	0.2366 ± 0.0102	0.5086 ± 0.0129
Qwen + STAMP	0.3998 ± 0.0240	0.2424 ± 0.0104	0.5116 ± 0.0174
ESM2 + STAMP*	0.3715 ± 0.0187	0.2099 ± 0.0111	0.4451 ± 0.0173
GENEPT + STAMP*	0.3699 ± 0.0190	0.2105 ± 0.0161	0.4327 ± 0.0165
Qwen + STAMP*	0.3279 ± 0.0732	0.2046 ± 0.0104	0.4465 ± 0.0085
ESM2 + scPertCRL	0.4225 ± 0.0231	0.2576 ± 0.0344	0.5233 ± 0.0097
GENEPT + scPertCRL	$\underline{0.4662 \pm 0.0224}$	$\underline{0.2974 \pm 0.0122}$	$\mathbf{0.5368 \pm 0.0064}$
Qwen + scPertCRL	$\mathbf{0.4684 \pm 0.0224}$	$\mathbf{0.3171 \pm 0.0233}$	$\underline{0.5318 \pm 0.0145}$

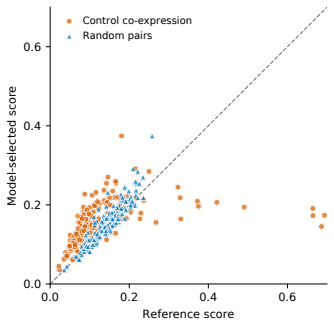


Figure 3: Differential-correlation comparison on RPE1.

DEG Identification. Beyond predicting global expression profiles, we evaluate whether the learned structural-change and exogenous-shift features support DEG prediction. We compare scPertCRL with STAMP and its variant on RPE1, K562, and Norman.

When both methods use the same perturbation embeddings, scPertCRL achieves higher mean AP than STAMP [34] on all three datasets (Table 4). On K562, the Qwen-based variant improves AP from 0.2424 to 0.3171, a relative increase of 30.8%. Improvements also extend to Norman, whose evaluation includes combinatorial perturbations, supporting the use of the learned features for DEG prediction across the evaluated settings.

Embedding choice affects AP, but scPertCRL outperforms the corresponding STAMP variants with ESM2, GENEPT, and Qwen embeddings. This consistency indicates that the observed gains are not explained by embedding choice alone.

Differential Mechanism Analysis. We examine whether learned structural changes correspond to observed correlation changes. For each held-out condition p, we select the top-100 gene pairs ranked by $|\Delta \mathbf{A}^{(p)}|$ and compare their mean absolute correlation change with that of control co-expression and randomly selected pairs.

In Figure 3, points above the diagonal indicate higher model-selected scores. Across five RPE1 runs with 163 held-out conditions each, model-selected pairs exceed the co-expression and random references in 80.9% and 56.8% of conditions on average, respectively. These results suggest an association with observed correlation changes, with a modest advantage over random selection (Appendix E.2).

Complementary analyses examine the biological context of the learned structures. Pathway-level comparisons with reference relationships derived from STRING connectivity and GO *regulates* relations suggest associations between the shared backbone $\mathbf{A}$ and biological reference relationships (Appendix E.3). An exploratory pathway analysis further examines whether perturbation-specific structural changes are associated with pathways containing the perturbed gene (Appendix E.4). Together with differential correlation, these analyses provide complementary views of baseline dependencies and perturbation-associated changes, without establishing individual causal edges.

4.5 Ablation Study

We conduct ablation studies to assess the contributions of key components on the RPE1 dataset, with results reported in Table 5.

Removing structural modulation (w/o Diff Net) or exogenous shifts (w/o Diff Dist) lowers mean PDS and Pearson Delta, while DEG Overlap changes little. Removing structural modulation produces a larger drop in PDS, from 0.6162 to 0.5694, than removing exogenous shifts, which reduces PDS to 0.6042. Removing both components (w/o Diff) further lowers PDS to 0.5076, close to the control baseline, although Pearson Delta remains relatively high at 0.6801. This contrast shows that

agreement with average perturbation-effect patterns can coexist with limited condition discrimination, highlighting the complementary roles of the evaluation metrics.

To complement the retrained ablations, we evaluate three inference branches of the same trained RPE1 models: the shared mechanism alone, the addition of exogenous shifts, and the subsequent addition of structural modulation. Across five runs, both additions improve the prediction summaries under this fixed ordering (Appendix E.5). This analysis examines the incremental effects of the two components within trained models.

Removing DEG supervision (w/o Target) lowers mean scores across all three metrics, suggesting that supervision of the structural-change and exogenous-shift features also benefits expression prediction. Removing perturbation-embedding noise (w/o Noise) likewise lowers the mean scores, consistent with its intended regularization role.

Together, these results support the use of structural modulation and exogenous shifts for perturbation prediction on RPE1, with additional benefits from DEG supervision and embedding-noise regularization.

Table 5: Ablation study on different components of scPertCRL on the RPE1 dataset.

Method	PDS ↑	DEG Overlap ↑	Pearson Delta ↑
scPertCRL	**0.6162 ± 0.0158**	0.4062 ± 0.0147	**0.7038 ± 0.0147**
w/o Diff	0.5076 ± 0.0026	0.3773 ± 0.0106	0.6801 ± 0.0136
w/o Diff Dist	0.6042 ± 0.0136	**0.4070 ± 0.0203**	0.6921 ± 0.0259
w/o Diff Net	0.5694 ± 0.0152	0.4033 ± 0.0169	0.6971 ± 0.0199
w/o Noise	0.5909 ± 0.0137	0.3854 ± 0.0116	0.6855 ± 0.0210
w/o Target	0.6137 ± 0.0139	0.4019 ± 0.0080	0.6916 ± 0.0187

5 Conclusion

We propose scPertCRL, a causally structured generative framework that models perturbation responses through structural changes and exogenous shifts relative to a shared latent backbone. Perturbation-conditioned low-rank updates capture variation in modeled dependencies while retaining shared baseline structure across conditions. Combined with differential-expression supervision, this design connects cellular response prediction with gene-level analysis of perturbation-associated changes. Experiments on genetic and pharmacological perturbation benchmarks support the predictive value of the framework, while complementary structural analyses provide biological context for its learned representations. scPertCRL thus offers a unified approach to predicting cellular responses and examining the latent dependency changes associated with perturbations.

Acknowledgments and Disclosure of Funding

This work received no third-party funding or support. The authors declare no competing interests.

References

[1] 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.

[2] 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.

[3] 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.

[4] Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024.

[5] Chris J Frangieh, Johannes C Melms, Pratiksha I Thakore, Kathryn R Geiger-Schuller, Patricia Ho, Adrienne M Luoma, Brian Cleary, Livnat Jerby-Arnon, Shruti Malu, Michael S Cuoco, et al. Multimodal pooled perturb-cite-seq screens in patient models define mechanisms of cancer immune evasion. *Nature genetics*, 53(3):332–341, 2021.

[6] Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular systems biology*, 19(6):e11517, 2023.

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

[8] 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.

[9] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9593–9602, 2021.

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

[11] Yuhang Liu, Zhen Zhang, Dong Gong, Mingming Gong, Biwei Huang, Anton van den Hengel, Kun Zhang, and Javen Qinfeng Shi. Identifiable latent polynomial causal models through the lens of change. *arXiv preprint arXiv:2310.15580*, 2023.

[12] Nora Schneider, Lars Lorch, Niki Kilbertus, Bernhard Schölkopf, and Andreas Krause. Generative intervention models for causal perturbation modeling. *arXiv preprint arXiv:2411.14003*, 2024.

[13] Ronan Perry, Julius Von Kügelgen, and Bernhard Schölkopf. Causal discovery in heterogeneous environments under the sparse mechanism shift hypothesis. *Advances in Neural Information Processing Systems*, 35:10904–10917, 2022.

[14] Trey Ideker and Nevan J Krogan. Differential network biology. *Molecular systems biology*, 8(1):565, 2012.

[15] Yulun Wu, Robert A Barton, Zichen Wang, Vassilis N Ioannidis, Carlo De Donno, Layne C Price, Luis F Voloch, and George Karypis. Predicting cellular responses with variational causal inference and refined relational information. *arXiv preprint arXiv:2210.00116*, 2022.

[16] Changxi Chi, Jun Xia, Jingbo Zhou, Jiabei Cheng, Chang Yu, and Stan Z Li. Grape: Heterogeneous graph representation learning for genetic perturbation with coding and non-coding biotype. *arXiv preprint arXiv:2505.03853*, 2025.

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

[18] Yiqun Chen and James Zou. Genept: a simple but effective foundation model for genes and cells built from chatgpt. *bioRxiv*, 2024. preprint.

[19] Russell Littman, Jacob Levine, Sepideh Maleki, Yongju Lee, Vladimir Ermakov, Lin Qiu, Alexander Wu, Kexin Huang, Romain Lopez, Gabriele Scalia, et al. Gene-embedding-based prediction and functional evaluation of perturbation expression responses with presage. *bioRxiv*, 2025. preprint.

[20] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

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

[22] Abhinav K Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, et al. Predicting cellular responses to perturbation across diverse contexts with state. *BioRxiv*, 2025. preprint.

[23] Mingze Dong, Abhinav Adduri, Dhruv Gautam, Christopher Carpenter, Rohan Shah, Chiara Ricci-Tam, Yuval Kluger, Dave P Burke, and Yusuf H Roohani. Stack: In-context learning of single-cell biology. *bioRxiv*, 2026.

[24] Hantao Shu, Jingtian Zhou, Qiuyu Lian, Han Li, Dan Zhao, Jianyang Zeng, and Jianzhu Ma. Modeling gene regulatory networks using neural network architectures. *Nature Computational Science*, 1(7):491–501, 2021.

[25] Jesus de la Fuente Cedeño, Robert Lehmann, Carlos Ruiz-Arenas, Jan Voges, Irene Marín-Goñi, Xabier Martinez de Morentin, David Gomez-Cabrero, Idoia Ochoa, Jesper Tegnér, Vincenzo Lagani, et al. Interpretable causal representation learning for biological data in the pathway space. In *The Thirteenth International Conference on Learning Representations*, 2024.

[26] Julius von Kügelgen, Jakob Ketterer, Michael Vollenweider, Michael Scholkemper, Xinwei Shen, Nicolai Meinshausen, and Jonas Peters. Extrapolation guarantees for perturbation modeling under the additive latent shift assumption, 2026. Version 3.

[27] Menghua Wu, Umesh Padia, Sean H Murphy, Regina Barzilay, and Tommi Jaakkola. Identifying biological perturbation targets through causal differential networks. *arXiv preprint arXiv:2410.03380*, 2024.

[28] Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *International conference on machine learning*, pages 372–407. PMLR, 2023.

[29] Yuhang Liu, Zhen Zhang, Dong Gong, Mingming Gong, Biwei Huang, Anton van den Hengel, Kun Zhang, and Javen Qinfeng Shi. Identifying weight-variant latent causal models. *arXiv preprint arXiv:2208.14153*, 2022.

[30] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

[31] Zizhen Deng, Hu Tian, Xiaolong Zheng, and Daniel Dajun Zeng. Deep causal learning: representation, discovery and inference. *ACM Computing Surveys*, 58(2):1–36, 2025.

[32] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

[33] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[34] Yicheng Gao, Zhiting Wei, Kejing Dong, Ke Chen, Jingya Yang, Guohui Chuai, and Qi Liu. Toward subtask-decomposition-based learning and benchmarking for predicting genetic perturbation outcomes and beyond. *Nature Computational Science*, 4(10):773–785, 2024.

[35] 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.

[36] Kaggle Competitions. Open problems – single-cell perturbations. https://www.kaggle.com/competitions/open-problems-single-cell-perturbations/data, 2023. Kaggle competition dataset. Predict how small molecules change gene expression in different cell types. Accessed: 2026-05-06.

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

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

A Training Process

We provide details about our training procedure in Algorithm 1. Here, c denotes the cell type used to group cells for training and evaluation. For each condition p, we observe a population of n cells, forming an expression matrix $\mathbf{X}^{(p)} \in \mathbb{R}^{n \times m}$. Single-cell measurements are inherently noisy and sparse. We organize cells into condition-specific sets, each representing an empirical distribution of expression profiles, for distribution-level training on unpaired perturbation data. This formulation allows us to focus on distribution-level perturbation effects, which is well suited to unpaired perturbation settings.

Algorithm 1 Forward and Training Procedure of SCPERTCRL

$(\mathbf{X}^{(\emptyset)}, \mathbf{X}^{(p)}, p) \sim \mathcal{D}$

— Stage 1: Baseline Backbone Learning —

for each batch of B cells $\mathbf{X}^{(\emptyset)} \sim \mathcal{D}$ **do**

$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) \leftarrow \mathrm{Enc}(\mathbf{X}^{(\emptyset)})$

$\mathbf{N} \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_1, \quad \boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\mathbf{Z} \leftarrow \mathbf{N}(\mathbf{I} - \mathbf{A}^\top)^{-1}$

$\hat{\mathbf{X}}^{(\emptyset)} \leftarrow \mathrm{Dec}(\mathbf{Z})$

$\mathcal{L}_{\text{stage1}} \leftarrow -\lambda_{\text{ELBO}} \mathcal{L}_{\text{ELBO}}^{(\emptyset)} + \lambda_{\text{dag}} \mathcal{L}_{\text{dag}}$

Update $\Theta \leftarrow \Theta - \eta \nabla_\Theta \mathcal{L}_{\text{stage1}}$

end for

— Stage 2: Perturbation-Specific Mechanism Learning —

Freeze the shared backbone $\mathbf{A}$

for each batch of B cells $(\mathbf{X}^{(\emptyset)}, \mathbf{X}^{(p)}, p) \sim \mathcal{D}$ **do**

$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) \leftarrow \mathrm{Enc}(\mathbf{X}^{(\emptyset)})$

$\mathbf{N} \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_2, \quad \boldsymbol{\epsilon}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\mathbf{Z} \leftarrow \mathbf{N}(\mathbf{I} - \mathbf{A}^\top)^{-1}$

$\hat{\mathbf{X}}^{(\emptyset)} \leftarrow \mathrm{Dec}(\mathbf{Z})$ ▷ Control state

$\mathbf{e}^{(p)} \leftarrow \mathrm{Embed}(p)$

$\tilde{\mathbf{e}}^{(p)} \leftarrow \mathbf{e}^{(p)} + \boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I})$

$\Delta \mathbf{A}^{(p)} \leftarrow \mathbf{U}(\tilde{\mathbf{e}}^{(p)}) \mathbf{V}(\tilde{\mathbf{e}}^{(p)})$

$(\boldsymbol{\mu}_\Delta^{(p)}, \log \boldsymbol{\sigma}_\Delta^{2(p)}) \leftarrow f_n(\tilde{\mathbf{e}}^{(p)})$

$\Delta \mathbf{n}_b^{(p)} \leftarrow \boldsymbol{\mu}_\Delta^{(p)} + \boldsymbol{\sigma}_\Delta^{(p)} \odot \boldsymbol{\epsilon}_{3,b}, \quad \boldsymbol{\epsilon}_{3,b} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad b = 1, \ldots, B$

$\mathbf{A}^{(p)} \leftarrow \mathbf{A} + \Delta \mathbf{A}^{(p)}$

$\Delta \mathbf{N}^{(p)} \leftarrow [\Delta \mathbf{n}_1^{(p)}, \ldots, \Delta \mathbf{n}_B^{(p)}]^\top, \quad \mathbf{N}^{(p)} \leftarrow \mathbf{N} + \Delta \mathbf{N}^{(p)}$

$\mathbf{Z}^{(p)} \leftarrow \mathbf{N}^{(p)} (\mathbf{I} - (\mathbf{A}^{(p)})^\top)^{-1}$

$\hat{\mathbf{X}}^{(p)} \leftarrow \mathrm{Dec}(\mathbf{Z}^{(p)})$ ▷ Perturbation state

$\mathbf{g}_i^{(p)} \leftarrow \left[\Delta \mathbf{A}_{i,\cdot}^{(p)},\ \Delta \mathbf{n}_i^{(p)}\right]$

$\hat{y}_i \leftarrow \sigma\Big(f(\mathbf{g}_i^{(p)})\Big)$ ▷ DEG prediction

$\mathcal{L}_{\text{total}} \leftarrow -\lambda_{\text{ELBO}} \mathcal{L}_{\text{ELBO}}^{(\emptyset)} - \lambda_{\text{ELBO}} \mathcal{L}_{\text{ELBO}}^{(p)} + \lambda_{\text{str}} \mathcal{L}_{\text{str}} + \lambda_{\text{diff}} \mathcal{L}_{\text{diff}}$

Update $\Theta \leftarrow \Theta - \eta \nabla_\Theta \mathcal{L}_{\text{total}}$

end for

B Identifiability Analysis

In this section, we map scPertCRL to the general identifiability theory for varying causal models. The goal is to clarify the theoretical motivation for our parameterization, rather than to claim that the practical implementation fully satisfies all assumptions required for exact causal recovery in single-cell data.

B.1 Definition and Mapping

We align our model with the causal generative process described in [29, 11]. The general formulation parameterizes the structural equation model (SEM) as follows:

$$\begin{aligned} n_i &\sim \mathcal{N}\big(\beta_{i,1}(p),\, \beta_{i,2}(p)\big), \\ z_i &:= \lambda_i(p)^T \mathbf{z} + n_i, \\ \mathbf{x} &:= \mathbf{f}(\mathbf{z},\, \varepsilon), \end{aligned}$$

where p denotes the auxiliary domain variable (perturbation condition), $\mathbf{z}$ denotes the latent causal variables, and parameters $(\boldsymbol{\lambda}, \boldsymbol{\beta})$ are functions of p.

In scPertCRL, we define the adjacency matrix as $\mathbf{A}^{(p)} = \mathbf{A} + \Delta\mathbf{A}^{(p)}$. Consequently, the coefficient function $\boldsymbol{\lambda}(p)$ in the general theory is instantiated as the sum of a baseline mechanism and a perturbation-specific delta:

$$\lambda_{i,j}(p) \triangleq A_{ij} + \Delta A_{ij}^{(p)}.$$

Similarly, the decomposition $\mathbf{n}^{(p)} = \mathbf{n} + \Delta\mathbf{n}^{(p)}$ provides a conceptual connection to condition-dependent exogenous distributions in the cited theory. In an idealized setting with independent Gaussian components, their means and variances add, yielding the following parameterization:

$$\beta_{i,1}(p) \triangleq \bar{\mu}_i^{(0)} + \bar{\mu}_i^{\Delta}(p), \quad \beta_{i,2}(p) \triangleq \bar{\sigma}_i^{2(0)} + \bar{\sigma}_i^{2\Delta}(p).$$

Here, the barred quantities denote population-level generative means and variances of the intrinsic and perturbation-induced components in this idealized setting, rather than outputs of the variational networks or the fixed standard normal priors used for KL regularization.

B.2 Identifiability of Latent Variables

We summarize the assumptions underlying the cited identifiability results that are most relevant to our model design:

1. **Decoder Assumptions:** The cited results impose regularity and invertibility conditions on the latent-to-observed mapping, together with assumptions on observation noise. These conditions concern the idealized generative model and are not enforced by the decoder architecture of scPertCRL.
2. **Noise Diversity:** The cited identifiability results require sufficiently rich variation in the parameters of the exogenous distributions across conditions. This requirement concerns distributional parameters rather than individual samples of $\Delta\mathbf{n}^{(p)}$. Our perturbation-conditioned shift distribution allows such variation to be represented, but does not by itself establish that the diversity conditions of the theory are satisfied.
3. **Structural Diversity:** The cited theory also requires sufficiently many perturbation conditions whose structural-parameter changes are linearly independent. In our parameterization, this corresponds to diversity in the perturbation-induced adjacency changes $\Delta\mathbf{A}^{(p)}$ across conditions.
4. **Smoothness with respect to perturbation embeddings:** Since perturbation labels are discrete, the Taylor-series condition from the cited theory does not apply directly to p. We interpret this condition with respect to the continuous perturbation embedding $e^{(p)}$, from which $\lambda_{i,j}(p)$ is generated through the neural maps defining $\Delta\mathbf{A}^{(p)}$. This is a modeling approximation and does not by itself guarantee global analyticity.

These conditions motivate our perturbation-conditioned parameterization of structural coefficients and exogenous shifts, as variation across conditions can help distinguish latent representations. We use these results as theoretical motivation, while the learned structures are interpreted under the practical limitations described above.

B.3 Identifiability of Causal Structures

In an idealized setting where the latent variables are recovered under the cited assumptions, structural identification can be studied in latent space using intervention-induced variation in the conditional mechanisms. The remaining ambiguity matters: component-wise nonzero scaling and permutation preserve the presence or absence of dependencies up to relabeling, whereas a general mixing of latent coordinates need not. Thus, when the applicable theory restricts the latent ambiguity appropriately and its structural-identification conditions hold, latent recovery can support identification of the dependency structure.

This connection motivates modeling a shared backbone $\mathbf{A}$ together with perturbation-specific changes $\Delta\mathbf{A}^{(p)}$: the parameterization explicitly represents both baseline dependencies and the mechanism variation that is informative in the theoretical setting. As noted above, our empirical setting does not establish the sufficient conditions, so the learned matrices remain model-based summaries of effective dependencies rather than recovered causal edges.

B.4 Other Assumptions

The theoretical framework motivates the following modeling assumptions for single-cell data:

Assumption 1 (Shared Causal Backbone). We assume that control and perturbed conditions share a largely conserved causal backbone. The model permits condition-specific structural changes through $\Delta\mathbf{A}^{(p)}$ without enforcing identical support across conditions. This aligns with the biological intuition that many regulatory relationships are conserved while intervention-specific rewiring can occur.

Assumption 2 (Low-Complexity Variation). We assume that perturbation-induced mechanism changes admit a low-complexity representation. Specifically, we use a low-rank factorization for $\Delta\mathbf{A}^{(p)}$ and encourage sparse structural changes through L_1 regularization. Sparsity reflects that perturbations typically affect a limited subset of pathways, while the low-rank constraint captures the correlation among mechanism changes driven by shared latent intervention factors.

C Computational Complexity

Computational Complexity Analysis. Structural propagation can become computationally expensive as the latent dimension grows. Let C denote the number of condition-level cell sets in a batch, S the number of cells per set, and m the latent dimension, so that the batch contains CS cells. We use two complementary strategies to reduce redundant structural computation and the cost of propagation.

First, cells within the same condition-level set share a coefficient matrix. The model therefore computes the structural coefficients once per set and reuses them across its S cells, while allowing up to C different matrices in a batch. To illustrate the benefit of this sharing, a direct-inversion implementation would reduce the structural-solve cost from $\mathcal{O}(CSm^3)$ for separate per-cell inversions to $\mathcal{O}(Cm^3 + CSm^2)$ for one inversion per set followed by propagation to its cells.

Second, our implementation uses a truncated Neumann expansion to avoid explicit inversion. With nonlinear step maps disabled, it sums the initial exogenous input and the outputs of K matrix propagation steps per cell. We use at most three propagation steps ($K \leq 3$) in our experiments. This yields a dense propagation cost of $\mathcal{O}(KCSm^2)$. The finite expansion approximates the inverse-based structural mapping; convergence to that mapping requires additional conditions and sufficiently many terms.

During training, the acyclicity penalty is evaluated on each condition-specific coefficient matrix. A matrix exponential of an $m \times m$ matrix costs $\mathcal{O}(m^3)$, giving $\mathcal{O}(Cm^3)$ for C condition-level

sets per batch. The combined cost of structural propagation and acyclicity evaluation is therefore $\mathcal{O}(Cm^3 + KCSm^2)$. This accounting excludes coefficient generation, encoders, and decoders. The cubic term arises from the acyclicity penalty; latent propagation uses the finite expansion described above.

Empirical Efficiency Analysis. We evaluate the training efficiency of condition-specific cell sets by fixing the total number of cells processed per forward pass at 400 and varying the number of cells per set.

Table 6: Comparison of model computational efficiency under different cell set partitioning strategies.

Set Size	Batch Size	Stage 1 Time per Epoch (s)	Stage 2 Time per Epoch (s)	Total Training Time (mins)
1	400	Out of Memory (OOM)		
10	40	14.47	83.94	~ 78
20	20	12.04	45.73	~ 45
50	8	10.15	24.75	~ 26

As shown in Table 6, the one-cell-per-set setting exceeds the available memory on an NVIDIA RTX 3090 with 24 GB VRAM. Increasing the set size from 10 to 50 reduces the number of sets per batch from 40 to 8. Stage 1 epoch time decreases from 14.47 to 10.15 seconds, Stage 2 epoch time decreases from 83.94 to 24.75 seconds, and total training time decreases from approximately 78 to 26 minutes. These measurements demonstrate improved training efficiency with larger condition-specific cell sets in the evaluated configuration.

Training Efficiency Across Gene Set Sizes. Table 7 compares total training time and model size for different numbers of input genes. As the gene set size increases from 1,554 to 2,429 and 5,000, total training time rises from approximately 18 to 25 and 59 minutes, respectively, while parameter counts increase from 81 to 129 and 277 million. These results illustrate the computational cost of increasing the number of modeled genes, with training completed in approximately one hour for the 5,000-gene setting. Total training time combines both training stages; the measurements reflect the stopping criteria used in each setting rather than a fixed-epoch comparison.

Table 7: Total training time and parameter count across different gene set sizes on RPE1. Training times are rounded to the nearest minute; parameter counts in millions are truncated to integers.

Input genes	Parameters (M)	Total Training Time (mins)
1,554	81	~ 18
2,429	129	~ 25
5,000	277	~ 59

D Experimental Details

D.1 Data Preprocessing

We use normalized and log-transformed expression matrices. For gene knockout/knockdown datasets, these transformations are applied during preprocessing before the processed AnnData matrices are loaded for training. Chemical perturbation data likewise undergo normalization and $\log(1 + x)$ transformation before model input. Where an HVG-filtered artifact is not already available, we select the top 2,000 genes by normalized dispersion using the `cell_ranger` criterion. For gene knockout/knockdown datasets, we also retain perturbation target genes present in the matrix and remove duplicate gene names. Thus, 2,000 is an HVG quota rather than necessarily the final input dimension; Table 8 reports the actual dimensions. Cells are grouped by perturbation condition into cell sets, without assuming one-to-one pairing.

D.2 Baseline Methods

Control. This identity baseline predicts the post-perturbation profile as the control state, i.e., $\hat{\mathbf{x}}^{(p)} = \mathbf{x}^{(\emptyset)}$.

Table 8: Summary of gene and drug perturbation datasets used in this study.

Dataset	# Cells	# Genes	# Perturbations	Perturbation Type	Notes
RPE1	76,231	2,429	651	Gene	Single-gene perturbations
K562	68,716	2,302	413	Gene	Single-gene perturbations
Norman	89,357	2,069	276	Gene	148 single-gene + 128 combinatorial perturbations
Sci_Plex_Combo	63,378	2,000	31	Drug	6 single-drug + 25 combinatorial perturbations
PBMC	240,084	2,000	602	Drug	6 cell types, single perturbations

Pert. This baseline predicts perturbed expression by adding a training-derived average perturbation offset to each control input cell. For strictly unseen perturbation splits, the offset cannot be estimated from the held-out test perturbation itself. We therefore compute it only from perturbation conditions observed in the training set.

Let $\mathcal{P}_{\text{train}}$ denote the set of non-control perturbation conditions available during training. We first compute the training control mean and the mean profile for each training perturbation $q \in \mathcal{P}_{\text{train}}$:

$$\boldsymbol{\mu}_{\text{ctrl}} = \frac{1}{|C|} \sum_{i \in C} \mathbf{x}^{(i)}, \quad \boldsymbol{\mu}_q = \frac{1}{|P_q|} \sum_{i \in P_q} \mathbf{x}^{(i)}.$$

Here, C denotes the set of control cells used for training and P_q denotes the training cells under perturbation q. The baseline offset is the average perturbation effect over training perturbations:

$$\bar{\boldsymbol{\delta}}_{\text{train}} = \frac{1}{|\mathcal{P}_{\text{train}}|} \sum_{q \in \mathcal{P}_{\text{train}}} \left(\boldsymbol{\mu}_q - \boldsymbol{\mu}_{\text{ctrl}} \right).$$

For a test control cell and target perturbation label p, the prediction is

$$\hat{\mathbf{x}}^{(p)} = \mathbf{x}^{(\emptyset)} + \bar{\boldsymbol{\delta}}_{\text{train}}.$$

Thus, this baseline is not perturbation-specific for held-out test conditions and does not use expression profiles from held-out perturbation conditions.

For the double-gene perturbation experiment, we distinguish three variants, all using offsets estimated exclusively from training data relative to $\boldsymbol{\mu}_{\text{ctrl}}$. **Pert (Shared Offset)** is the baseline defined above and uses the same average offset for every target combination. **Pert (Single-gene Additive)** sums the offsets from the two constituent single-gene training conditions and adds the sum to each input control cell. **Pert (Gene-conditioned Additive)** estimates an offset for each constituent gene by pooling all training conditions containing that gene, including single- and multi-gene conditions, with cell-count weighting. It sums the two gene-conditioned offsets and adds the sum to the training control mean. Thus, the first two variants preserve input control-cell variability, whereas the third predicts a common mean profile.

CPA. CPA uses an autoencoder with learnable perturbation, covariate, and dosage embeddings. Combinatorial treatments are represented by adding perturbation embeddings, enabling zero-shot inference for unseen combinations. We use the official implementation and adapt it to our normalization and splitting protocols [6].

scVI. scVI is a VAE-based generative model that maps gene expression profiles to a probabilistic latent representation and decodes distributional parameters. We concatenate the same perturbation representations used by scPertCRL with the hidden features and use the official implementation with our data splits [20].

DiscrepancyVAE. DiscrepancyVAE is a causal representation learning baseline that models perturbations as additive shifts in latent space and includes a counterfactual loss. We extend its existing one-hot condition encoding to accommodate the perturbation conditions in our benchmarks and use the same data and condition splits [10].

GEARS. GEARS is a GNN-based baseline that uses gene-gene interactions and perturbation relationships for response prediction, with an additive formulation for combinatorial perturbations. We use the official implementation and default training scheme, adapted to our preprocessing and splits [7].

scGPT. scGPT represents cells as sequences of gene tokens that combine expression values and gene identity embeddings. We initialize the model with pretrained scGPT weights and fine-tune the model using gene-level perturbation indicators under our preprocessing and data splits [8].

STATE. STATE is a Transformer-based model for virtual cell tasks that accounts for cellular heterogeneity within and across experiments. We focus on its transition module and train it from scratch using the official codebase under the same evaluation protocol [22].

CellFlow. CellFlow uses conditional flow matching to learn transformations from control to perturbed cell distributions [38]. We include it as a baseline for predicting responses to held-out double-gene perturbations.

PRNet. PRNet predicts responses to novel chemical perturbations by combining chemical structure information with control transcriptional profiles. We follow the official codebase, retain its SMILES-based chemical embedding module, and use our benchmark data splits [37].

STAMP. STAMP decomposes perturbation prediction into DEG identification, direction prediction, and magnitude estimation. We compare against its DEG-identification subtask and follow the official codebase [34]. STAMP* is an extension of STAMP, which primarily adopts a gene-wise shared network.

D.3 Metrics

Perturbation Discrimination Score (PDS). PDS formulates evaluation as a rank-based retrieval task: the predicted perturbation-induced change for a target perturbation should be closer to the ground-truth change of the same perturbation than to changes from other conditions. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ denote the set of N perturbation conditions. For each perturbation $k \in \mathcal{P}$, let $\hat{\mathbf{X}}^{(k)}$ and $\mathbf{X}^{(k)}$ represent the predicted and ground-truth log-normalized expression matrices. First, we compute pseudobulk profiles, where n_k and n_k^{true} denote the numbers of predicted and observed cells, respectively, for condition k:

$$\hat{\mathbf{y}}^{(k)} = \frac{1}{n_k} \sum_{i=1}^{n_k} \hat{\mathbf{X}}_i^{(k)}, \quad \mathbf{y}^{(k)} = \frac{1}{n_k^{\text{true}}} \sum_{j=1}^{n_k^{\text{true}}} \mathbf{X}_j^{(k)}.$$

Second, we compute perturbation deltas relative to the corresponding control pseudobulk profiles:

$$\hat{\boldsymbol{\delta}}^{(k)} = \hat{\mathbf{y}}^{(k)} - \hat{\mathbf{y}}^{(\emptyset)}, \quad \boldsymbol{\delta}^{(k)} = \mathbf{y}^{(k)} - \mathbf{y}^{(\emptyset)}.$$

The CellEval L_1 discrimination score compares the *absolute* pseudobulk effects. For query p, let $G(p) = \{g : \text{gene name}_g \neq p\}$; this is the exact-string mask used by the evaluator. It removes a target gene only when the full condition label equals that gene name. The distance to real candidate t is

$$d_{p,t} = \sum_{g \in G(p)} \left| \left| \hat{\delta}_g^{(p)} \right| - \left| \delta_g^{(t)} \right| \right|.$$

For example, the RPE1 test labels have the form `GENE+ctrl`, whereas the gene names are `GENE`; thus $G(p)$ contains all genes for those rows despite the evaluator's target-exclusion option being enabled. Candidate labels are the common non-control conditions present in both real and predicted evaluation files.

Candidate ground-truth perturbations are ordered by increasing distance. Let r_p be the one-based position of the matching condition $t = p$ in the evaluator's `argsort` output:

$$r_p = \text{rank}\big(d_{p,p} \text{ in } \{d_{p,t}\}_{t=1}^{N}\big).$$

The normalized discrimination score is

$$\text{PDS}_p = 1 - \frac{r_p - 1}{N}.$$

A score of 1 means that the matching condition ranks first; the last position scores $1/N$. The implementation does not average tied ranks, so tied distances inherit the ordering returned by `argsort`. The reported aggregate is the mean across test queries. This definition follows the CellEval procedure used for the reported benchmark results.

DEG Overlap. DEG Overlap evaluates how well generated data preserve perturbation-associated differential-expression signals relative to real data.

For each perturbation condition k, we identify differentially expressed genes by comparing perturbed cells against the control population with a Wilcoxon rank-sum test. This yields $\mathcal{G}_{k,\text{true}}$ from real data and $\mathcal{G}_{k,\text{pred}}$ from model-generated data.

Let $n_{k,\text{true}} = |\mathcal{G}_{k,\text{true}}|$. We select the top $n_{k,\text{true}}$ predicted DE genes ranked by absolute log-fold change (LFC):

$$\tilde{\mathcal{G}}_{k,\text{pred}} = \text{Top}_{n_{k,\text{true}}} \left(\mathcal{G}_{k,\text{pred}} \text{ sorted by } |\text{LFC}|\right).$$

The DEG Overlap score is

$$\text{DEGOverlap}_k = \frac{|\tilde{\mathcal{G}}_{k,\text{pred}} \cap \mathcal{G}_{k,\text{true}}|}{n_{k,\text{true}}}.$$

The CellEval implementation uses adjusted $p < 0.05$ for both real and predicted DE lists. If either list is absent for a condition, its overlap is set to zero; when no significant genes occur in either dataset, all condition scores are set to zero. If fewer than $n_{k,\text{true}}$ predicted genes pass the filter, all available predicted genes are used while the denominator remains $n_{k,\text{true}}$. The final DEG Overlap is reported as the mean across perturbation conditions. A higher value indicates greater overlap between predicted and real DE gene sets.

Average Precision (AP). For each perturbation condition, we average the gene classifier scores across cells and compute AP across genes using the DEG labels provided by preprocessing. Condition-level AP scores are summarized by their arithmetic mean within each run. We report the mean and standard deviation across runs.

Pearson Delta. Pearson Delta measures the correlation between predicted and true average expression changes across genes relative to the control condition. It is invariant to positive rescaling of either effect vector and therefore does not assess absolute effect magnitudes.

For each perturbation p, we first aggregate single-cell predictions and ground-truth measurements by averaging gene expression across all cells subjected to p. Let $\hat{\boldsymbol{\mu}}_p$ and $\boldsymbol{\mu}_p$ denote the predicted and true mean expression vectors under perturbation p, respectively, and let $\hat{\boldsymbol{\mu}}_{\text{ctrl}}$ and $\boldsymbol{\mu}_{\text{ctrl}}$ denote their corresponding control means. The evaluation uses the same control cells for the predicted and observed datasets, so the two control means coincide.

We then define the perturbation effect vectors as gene-wise expression shifts relative to the control:

$$\Delta\hat{\mathbf{x}}_p = \hat{\boldsymbol{\mu}}_p - \hat{\boldsymbol{\mu}}_{\text{ctrl}}, \quad \Delta\mathbf{x}_p = \boldsymbol{\mu}_p - \boldsymbol{\mu}_{\text{ctrl}}.$$

The Pearson Delta score for perturbation p is computed as the Pearson correlation coefficient between the predicted and true perturbation effect vectors across genes:

$$\text{Pearson Delta}_p = \rho\big(\Delta\hat{\mathbf{x}}_p,\ \Delta\mathbf{x}_p\big).$$

Finally, the overall Pearson Delta metric is obtained by averaging perturbation-wise scores over the test set. A higher value indicates better agreement in the direction and relative magnitude of average gene-level perturbation effects. For the Control baseline, Pearson Delta is undefined and reported as 0 by convention.

D.4 Implementation Details

All computational experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU. The input to scPertCRL is log-normalized gene expression, and the decoder likelihood is modeled as a Gaussian distribution $p_\theta(\mathbf{x} \mid \mathbf{z})$.

We set the latent dimension equal to the observed gene dimension ($d = m$) and use gene-specific decoders to facilitate gene-level interpretation of the latent representation. The encoder is parameterized as an MLP with hidden dimensions $m \to 256 \to 256 \to d$. To map latent states back to observed expression, we employ d independent gene-specific decoders, each implemented as a $1 \to 16 \to 1$ MLP. The training objective uses $\lambda_{\text{ELBO}} = 1$, $\lambda_{\text{str}} = 0.001$, and $\lambda_{\text{diff}} = 0.2$. Following prior work [30, 31], the first-stage acyclicity penalty combines a linear term and a quadratic term in

$h(\mathbf{A})$. Within the perturbed-branch reconstruction surrogate $\mathcal{L}_{\text{recon}}^{(p)}$, we use a Sinkhorn Wasserstein weight $\alpha_{\text{W}} = 0.001$ and an energy-distance weight $\alpha_{\text{E}} = 0.1$. For the perturbation-specific structural update, we use a low-rank factorization $\Delta\mathbf{A}^{(p)} = \mathbf{U}(e^{(p)})\mathbf{V}(e^{(p)})$ with rank $r = 48$.

The set-level distributional surrogate D is implemented using the `SamplesLoss` interface of GeomLoss (`https://www.kernel-operations.io/geomloss/`). It combines the Sinkhorn loss (`loss="sinkhorn"`, `p=2`, `blur=0.05`) and the energy loss (`loss="energy"`, `blur=0.05`), weighted by α_{W} and α_{E}, respectively.

For DEG supervision, we compare perturbed and control cells using Scanpy's `t-test_overestim_var` and use adjusted $p \leq 0.01$ to mark positive genes. Following STAMP [34], training uses these labels with class-weighted BCE, while DEG Overlap evaluation uses the Wilcoxon-based procedure described above. Here, BCE in $\mathcal{L}_{\text{diff}}$ denotes the class-weighted loss.

We train with AdamW using an initial learning rate of 1×10^{-3}, a cell set size of 50, and a batch size of 8. The first stage runs for 20 epochs, and the second stage runs for up to 50 epochs with an early-stopping patience of 10. These hyperparameters are kept fixed across datasets for scPertCRL; official baseline implementations use their default training schemes when available, adapted only to the shared preprocessing and splits described above.

Training Convergence. In two RPE1 runs with available validation-loss histories, the summed validation terms decrease overall from the first to the final epoch in both training stages, although the trajectories are not monotonic. The final epoch does not necessarily attain the lowest validation loss. Since the prediction objective is introduced in the second stage, loss magnitudes are not directly comparable across stages. These observations illustrate the optimization behavior of these runs rather than establish convergence across all experiments.

The code for scPertCRL is available at `https://github.com/fdu-wangfeilab/scPertCRL`. Dataset access follows the public sources cited in the Experiments section, and preprocessing follows Appendix D.

D.5 Embedding Strategy

We use publicly available pretrained models to obtain frozen perturbation embeddings. Gene perturbations are represented by precomputed 2,560-dimensional Qwen [32] embeddings of gene descriptions obtained from NCBI. Drug perturbations are encoded from chemical SMILES strings using the frozen `DeepChem/ChemBERTa-77M-MTR` model [33]. We extract the first-token representation from its final hidden layer to obtain a 384-dimensional vector for each drug. For combinatorial perturbations, we sum the constituent embeddings element-wise before projecting them into a task-specific latent space. The gene branch uses a trainable two-layer nonlinear adapter, while the drug branch uses a trainable linear projection.

Of note, when a baseline supports external perturbation embeddings, we provide the same semantic perturbation features used by scPertCRL. For methods whose official implementation relies on specific perturbation representations or biological graphs, we retain their original design and adapt only the data split and preprocessing. This avoids weakening baselines by forcing incompatible input representations.

D.6 Evaluation Splits and Repeated Runs

The main scPertCRL runs use the condition-level partitions summarized in Tables 9 and 10. The tables report the median number of distinct training, validation, and test conditions over seeds 0–4; these are condition counts, not cell counts. The RPE1, K562, and Norman training counts include the control condition. For RPE1 and K562, all test conditions are single-gene perturbations whose target genes occur in neither the final training nor validation conditions. Control cells define the control distribution.

As a concrete characterization of test-set breadth, the RPE1 seed-0 split contains 163 unseen perturbation genes, all excluded from both training and validation. Of these, 152 have Gene Ontology Biological Process annotations and span 15 GO-based functional modules, including RNA metabolism, cell-cycle regulation, chromosome organization, nuclear transport, and vesicle trafficking; the re-

maining 11 lack the annotations required for this clustering analysis. The functional clustering was performed after the split was constructed and did not determine membership in the training, validation, or test sets. Thus, this held-out set was not deliberately restricted to a single pathway or a narrowly selected class of perturbations, although functional breadth alone does not establish that all modules are equally difficult to predict.

Norman test conditions include held-out single-gene perturbations and gene combinations. The combo_seen0, combo_seen1, and combo_seen2 labels count how many constituent genes belong to the initial candidate training-gene set, before the subsequent validation split; unseen_single denotes a held-out single-gene perturbation. Consequently, these labels need not equal the number of components actually present in the final training conditions. Table 11 gives the test subgroup sizes for each seed. Sci_Plex_Combo uses held-out two-drug combinations: the exact combinations are absent from training, while the visibility of each constituent drug in the final training set varies by seed.

For PBMC, each seed has six leave-one-cell-type-out folds. Perturbation conditions from the target cell type are excluded from training and validation, while conditions from the other five types are split between them. Control cells from the target type are available as prediction inputs and are not counted among the partitioned perturbation conditions in Table 10. Table 3 reports the Pearson Delta for each held-out cell type. Unless otherwise stated, quantitative results are reported as mean $\pm$ standard deviation over seeds 0–4, reflecting variation from split construction when applicable, model initialization, and optimization.

Table 9: Condition counts across five random seeds.

Dataset	Train	Validation	Test
RPE1	439	49	163
K562	279	31	103
Norman	139	28	111
Sci_Plex_Combo	23	4	4

Table 10: Median PBMC condition counts across five seeds by held-out cell type.

Held-out cell type	Train	Validation	Test
B cells	526	56	14
Myeloid cells	527	56	13
NK cells	408	44	144
T cells CD4+	408	44	144
T cells CD8+	413	44	139
T regulatory cells	410	44	142

E Additional Results

E.1 Detailed Performance on the Norman Dataset

We group Norman test conditions according to the membership of their target genes in the initial candidate training-gene set $\mathcal{G}_{\text{train}}^{\text{cand}}$, defined before the validation split. For a held-out two-gene combination (g_1, g_2), combo_seen0 means neither gene belongs to this set, combo_seen1 means exactly one belongs to it, and combo_seen2 means both belong to it. For example, if $g_1 \in \mathcal{G}_{\text{train}}^{\text{cand}}$ and $g_2 \notin \mathcal{G}_{\text{train}}^{\text{cand}}$, their held-out combination is classified as combo_seen1. The suffix counts constituent genes, not previously observed combinations; the exact test combination is held out. The unseen_single subgroup contains held-out single-gene perturbations. These labels describe the initial split construction and do not necessarily count genes represented in the final training conditions after validation is withheld. Table 11 reports subgroup sizes.

As shown in Table 12, combo_seen1 and combo_seen2 have higher Pearson Delta than combo_seen0 and unseen_single. This pattern is consistent with a benefit from greater exposure to perturbation components, subject to the distinction between candidate-set membership and final training visibility described above.

Table 11: Statistics of seen and unseen perturbation combinations in the Norman dataset.

seed	combo_seen0	combo_seen1	combo_seen2	unseen_single
0	10	48	18	41
1	9	43	19	36
2	12	52	16	37
3	4	51	19	35
4	4	52	18	37

Table 12: Performance comparison across different test subgroups.

Test subgroup	PDS ↑	DEG Overlap ↑	Pearson Delta ↑
combo_seen0	0.6783 ± 0.0767	0.3557 ± 0.0411	0.5827 ± 0.1416
combo_seen1	0.7242 ± 0.0814	0.4048 ± 0.0273	0.6178 ± 0.0368
combo_seen2	0.8242 ± 0.0495	0.4243 ± 0.0184	0.7263 ± 0.0631
unseen_single	0.5954 ± 0.0304	0.2691 ± 0.0499	0.4309 ± 0.0426

Prediction of Double-Gene Perturbation Effects. We evaluate whether scPertCRL can predict the combined effects of two gene perturbations when their joint condition has not been observed during training. For each of five seeds, the Norman test set contains 37 held-out double-gene combinations whose constituent single-gene perturbations are available in training. We compare scPertCRL using one-hot or Qwen embeddings with three Pert baselines and CellFlow. The Pert variants test whether a shared response or an additive combination of training-derived gene effects can explain the observed double-perturbation responses; their definitions are provided in Appendix D.2. All methods use the same test conditions, target cells, and 2,069-gene evaluation space within each seed. Scores are averaged equally across test conditions, and Table 13 reports the mean $\pm$ standard deviation across seeds. This experiment focuses on unseen combinations of observed single-gene perturbations and is evaluated separately from the broader combinatorial benchmark in Table 2.

Table 13: Performance comparison on held-out double-gene perturbations in Norman.

Method	PDS ↑	DEG Overlap ↑	Pearson Delta ↑
scPertCRL (One-hot)	0.9480 ± 0.0333	$\mathbf{0.4370 \pm 0.0180}$	$\mathbf{0.8922 \pm 0.0057}$
scPertCRL (Qwen)	0.8057 ± 0.0326	0.4311 ± 0.0160	0.8518 ± 0.0217
Pert (Shared Offset)	0.5175 ± 0.0018	0.0477 ± 0.0029	0.4743 ± 0.0196
Pert (Single-gene Additive)	0.9724 ± 0.0129	0.2052 ± 0.0174	0.8888 ± 0.0096
Pert (Gene-conditioned Additive)	$\mathbf{0.9812 \pm 0.0088}$	0.1856 ± 0.0129	0.8888 ± 0.0112
CellFlow	0.5236 ± 0.0064	0.0399 ± 0.0047	0.5141 ± 0.0586

Notes. Definitions of the three Pert variants are provided in Appendix D.2.

As shown in Table 13, scPertCRL (One-hot) achieves the highest mean DEG Overlap and Pearson Delta. The additive Pert baselines attain similar mean Pearson Delta and higher mean PDS, but substantially lower DEG Overlap. Both scPertCRL variants also outperform Pert (Shared Offset) and CellFlow on all three reported metrics. These results show that additive baselines capture much of the average double-perturbation response, while scPertCRL better preserves the differential-expression signals in this evaluation.

For this comparison, CellFlow is trained for 50,000 updates, with scaling and 100-component PCA fitted exclusively on the training cells.

E.2 Differential Correlation Analysis of Structural Changes

We assess whether large structural changes predicted by the model correspond to large changes in gene-pair correlations in the observed expression data. For each perturbation p, we merge the two directions of each gene pair using the larger absolute value in $\Delta\mathbf{A}^{(p)}$. Excluding self-pairs, we select the top 100 distinct unordered gene pairs. We then evaluate these selected pairs using the change in Pearson correlation between the observed perturbed and control cell populations:

$$\Delta\mathrm{Corr}_{i,j}^{(p)} = \left|\mathrm{Corr}_{\mathbf{X}^{(p)}}(i,j) - \mathrm{Corr}_{\mathbf{X}^{(\emptyset)}}(i,j)\right|.$$

For each selected group, we average this score over its 100 gene pairs. Thus, model-derived structural changes determine which pairs are selected, while observed expression data determine how much their correlations change. The evaluation concerns changes in gene-pair correlations rather than changes in individual gene expression levels.

We compare the model with two reference selection strategies using the same candidate gene pairs. The control co-expression reference selects 100 pairs with the highest absolute Pearson correlation estimated from one half of the control cells. The random reference uniformly samples 100 pairs without replacement; its evaluation score is averaged over 1,000 repetitions per condition. All selected pairs are evaluated using the same observed perturbed cells and evaluation control cells. The latter are sampled from the remaining half of the control cells, matching the number of perturbed cells for each condition. Correlations are computed from the log-transformed expression matrix used by the model, without additional preprocessing. Perturbed-cell expression is used only for evaluation, not for gene-pair selection.

For each perturbation condition, we compare the average absolute correlation change of the model-selected pairs with that of each reference. A condition is counted as a win only when the model score is strictly higher; ties are not counted as wins. We calculate the percentage of winning conditions separately for each run and split, then average these percentages across five runs, as reported in Table 14.

Table 14: Percentage of RPE1 conditions with higher differential correlation for model-selected gene pairs than reference pairs.

Split	**vs. Control Co-expression**	**vs. Random Pairs**
Training	81.5%	59.2%
Held-out Test	80.9%	56.8%

Notes. Entries are means of five seed-level percentages. Each seed includes 438 training and 163 held-out test conditions. A higher percentage indicates that model-selected pairs more frequently exceed the reference score; it does not measure edge-recovery accuracy.

On held-out conditions, model-selected pairs exceed the control co-expression reference in 80.9% of cases and the random reference in 56.8%. These results suggest an association between the learned differential structure and observed correlation changes, with a more limited advantage over random selection. The percentages summarize how often the model exceeds each reference, without quantifying the size or statistical significance of the differences.

E.3 Comparison of the Shared Structure with Biological References

We examine whether the shared backbone $\mathbf{A}$ contains information associated with known biological relationships. We compare five RPE1 models for each of two input gene sets: 2,000 highly variable genes (HVGs) with retained perturbation targets (HVG + targets), and the same selection augmented with transcription factors (HVG + targets + TF). The resulting input dimensions are 2,429 and 2,699 genes, respectively.

We select 200 GO Biological Process pathways with 10–200 annotated genes and at least 10 modeled genes, prioritizing specific terms and limiting pairwise gene-set Jaccard similarity to 0.5. STRING-positive pairs require at least three STRING-connected genes per pathway and three cross-pathway connections exceeding twice the background expectation. GO-positive pairs are defined by mapping *regulates* relations through the ontology hierarchy and merging directions and regulatory signs.

We aggregate the learned coefficients between pathway member genes using the root mean square and merge the two directions by taking the larger score. We evaluate how well the learned scores rank reference-positive pairs above pairs not annotated in the reference. Table 15 reports two ranking metrics as mean $\pm$ standard deviation across five seeds.

AUPRr measures precision–recall performance relative to the prevalence of known reference relations. A value above 1 indicates performance above the random-ranking baseline. AUROC measures how well the scores distinguish known reference relations from unannotated pairs, with 0.5 representing random ranking. Higher values indicate better performance for both metrics.

Table 15: Pathway-level comparison of the learned shared structure with STRING-derived and GO *regulates* reference relations on RPE1. Entries are mean ± standard deviation across five seeds.

Input gene set	Reference	AUPRr ↑	AUROC ↑
HVG + targets	STRING	4.399 ± 0.190	0.694 ± 0.006
HVG + targets	GO regulates	1.816 ± 0.042	0.573 ± 0.005
HVG + targets + TF	STRING	2.442 ± 0.030	0.608 ± 0.002
HVG + targets + TF	GO regulates	1.823 ± 0.014	0.570 ± 0.002

For STRING-derived pathway relations, the mean AUPRr is 4.399 with the HVG + targets gene set and 2.442 with HVG + targets + TF, with corresponding AUROCs of 0.694 and 0.608. For GO pathway relations, AUPRr is approximately 1.82 and AUROC approximately 0.57 for both gene sets. Thus, the scores rank reference relations above the random baselines on average, with modest discrimination as measured by AUROC. The STRING scores are lower for HVG + targets + TF, whereas the GO scores change little. Both input gene sets are evaluated using the same pathways and pathway member genes, with models trained separately for each input gene set. Overall, the comparisons provide exploratory evidence of association with biological reference relations, rather than validation of individual regulatory edges.

E.4 Exploratory Pathway Alignment

We examine whether pathways containing a perturbed gene exhibit relatively strong outgoing structural changes. For each perturbation target, we perform GO enrichment on the 200 genes with the largest absolute outgoing coefficient changes. We select 100 pathways by their frequency among the top 15 enrichment terms per perturbation, limiting pairwise gene-set Jaccard similarity to 0.5. Pathway-to-pathway strengths are computed using weighted RMS, downweighting genes shared across pathways. Each pathway's outgoing score is its mean strength toward other eligible pathways. We adjust these scores by subtracting the perturbation-wise and pathway-wise means and adding back the overall mean.

Among 163 held-out perturbations in one RPE1 run, 124 have target genes belonging to at least one selected pathway. For each eligible perturbation, we average the adjusted scores over pathways containing its target gene, then average equally across perturbations to obtain the test statistic. A perturbation-label permutation test with 10,000 permutations suggests an association between perturbation identity and outgoing changes in its target gene's pathways ($p = 0.0070$). This result remains exploratory because pathways were selected from the same model outputs, and selection was not repeated during permutation.

E.5 Predictive Effects of Exogenous Shifts and Structural Modulation

We examine how exogenous shifts and structural modulation affect prediction by comparing three inference branches of the trained RPE1 models. Starting from the encoded control component $\mathbf{n}$, we use the shared mechanism alone, add the perturbation-specific exogenous shift, and then additionally enable structural modulation:

$$\begin{aligned}
\mathbf{z}_{\text{base}} &= \mathcal{P}(\mathbf{A}, \mathbf{n}), \\
\mathbf{z}_{\text{shift}} &= \mathcal{P}(\mathbf{A}, \mathbf{n} + \Delta\mathbf{n}^{(p)}), \\
\mathbf{z}_{\text{full}} &= \mathcal{P}(\mathbf{A} + \Delta\mathbf{A}^{(p)}, \mathbf{n} + \Delta\mathbf{n}^{(p)}),
\end{aligned}$$

where $\mathcal{P}$ denotes structural propagation. All branches reuse the trained parameters without retraining. Table 16 reports mean ± standard deviation across runs after averaging over held-out perturbation conditions within each run.

Adding the exogenous shift reduces MSE and increases Pearson Delta, while enabling structural modulation further improves both metrics. These results suggest that both components provide useful predictive information in the trained models. The improvements are conditional on the order of addition and should be interpreted as an inference-time comparison rather than independent component contributions.

Table 16: Prediction performance of three inference branches on RPE1.

Branch	MSE ↓	Pearson Delta ↑
Shared mechanism	0.0255 ± 0.0029	0.4102 ± 0.1088
+ Exogenous shift	0.0162 ± 0.0028	0.6659 ± 0.0380
+ Structural modulation	0.0113 ± 0.0008	0.7417 ± 0.0172

Notes. Components are added cumulatively. Since RPE1 contains a single cell type, expression profiles are averaged within each perturbation condition p. MSE is computed between predicted and observed group-mean expression profiles. Pearson Delta is computed across genes within each condition.

E.6 Sensitivity to Gene Set Size

We evaluate sensitivity to the number of input genes using gene sets of 1,554, 2,429, and 5,000 genes on RPE1, with the same 163 held-out perturbation conditions. Table 17 reports mean $\pm$ standard deviation across runs for PDS, DEG Overlap, and Pearson Delta.

Table 17: Performance comparison across different gene set sizes on RPE1.

Input genes	PDS ↑	DEG Overlap ↑	Pearson Delta ↑
1,554	0.626 ± 0.013	0.467 ± 0.020	0.717 ± 0.016
2,429	0.615 ± 0.027	0.413 ± 0.020	0.717 ± 0.029
5,000	0.611 ± 0.011	0.351 ± 0.018	0.693 ± 0.033

The 1,554-gene setting achieves the highest mean PDS and DEG Overlap, while the 1,554- and 2,429-gene settings have the same mean Pearson Delta at the reported precision (0.717). Increasing the gene set to 5,000 reduces mean PDS to 0.611, DEG Overlap to 0.351, and Pearson Delta to 0.693. Thus, increasing the number of modeled genes does not improve the reported prediction scores in this experiment. Since the genes used for training and evaluation both change across settings, these trends do not isolate input dimensionality on a fixed prediction task or establish a universally optimal gene set size.

E.7 Sensitivity to the Low-Rank Dimension

We examine how the rank r of the perturbation-specific structural change affects predictive performance and model size. We compare $r \in \{16, 32, 48, 64\}$ on RPE1, holding the remaining settings and held-out perturbation conditions fixed. Table 18 reports mean $\pm$ standard deviation across runs and the corresponding parameter counts.

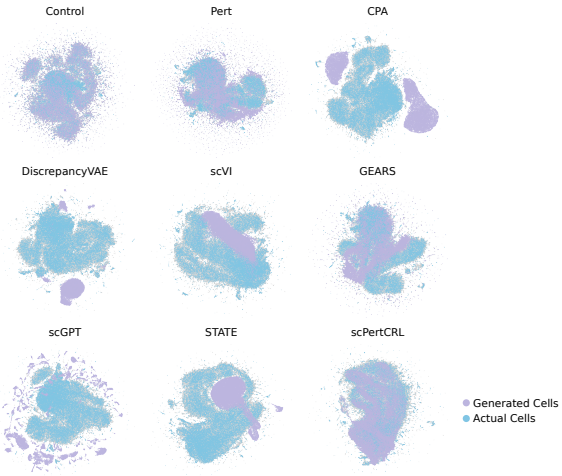
Table 18: Performance comparison across different low-rank dimensions on RPE1.

Rank (r)	PDS ↑	DEG Overlap ↑	Pearson Delta ↑	Parameters (M)
16	0.619 ± 0.016	0.414 ± 0.020	0.716 ± 0.035	49.58
32	0.620 ± 0.017	0.412 ± 0.019	0.713 ± 0.020	89.68
48	0.615 ± 0.027	0.413 ± 0.020	0.717 ± 0.029	129.78
64	0.618 ± 0.015	0.408 ± 0.013	0.707 ± 0.027	169.89

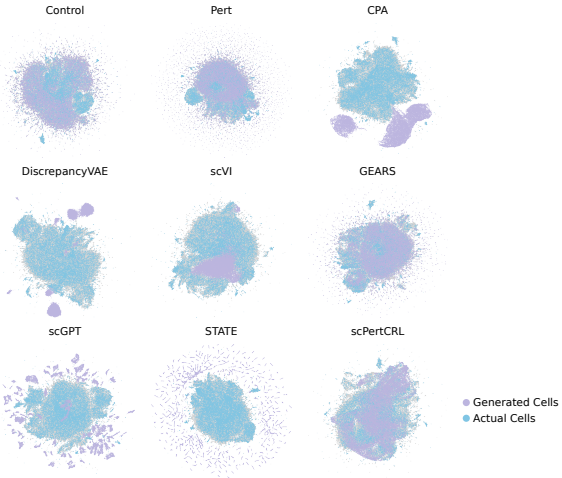
Ranks from 16 to 48 yield similar mean prediction scores. Rank 48 achieves the highest mean Pearson Delta (0.717), with PDS and DEG Overlap comparable to those obtained at smaller ranks. Increasing the rank to 64 raises the parameter count from 129.78M to 169.89M, while mean Pearson Delta decreases to 0.707 and DEG Overlap to 0.408, with only a small increase in PDS. We retain $r = 48$ as the default setting based on its competitive performance across metrics and the limited benefit of additional capacity. Smaller ranks also achieve comparable scores with fewer parameters.

F Qualitative Comparison on Gene Perturbation Datasets

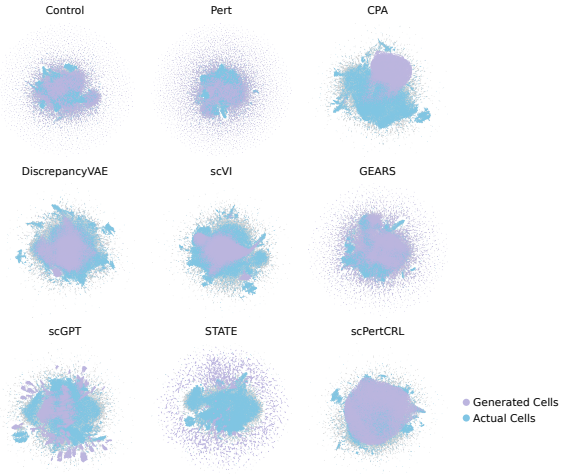
Figure 4 compares UMAP visualizations of generated and observed perturbed cells on RPE1, K562, and Norman.



(a) RPE1



(b) K562



(c) Norman

Figure 4: UMAP visualizations comparing control, actual perturbed, and generated perturbed cells across datasets.

For each shown perturbation, we first identify significant DEGs by comparing actual perturbed and control cells, then select the 20 DEGs with the largest absolute mean expression changes. Figure 5 compares the expression changes over control for actual perturbed cells and generated cells. Overall, our method closely follows the expression patterns of these genes in most cases. For some genes, the predicted mean changes deviate from the observed mean responses, indicating that the model does not reproduce all gene-level effect magnitudes equally well.

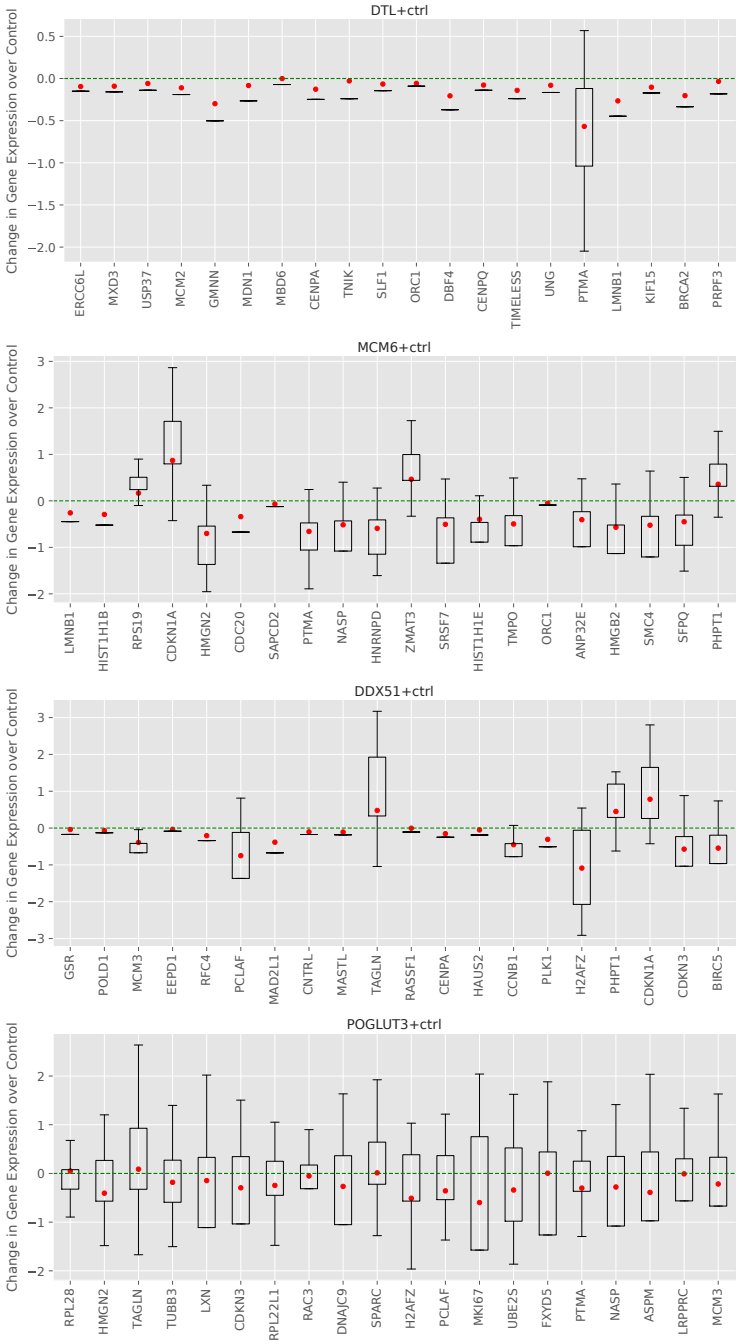


Figure 5: Top-20 differentially expressed genes for selected RPE1 perturbations. Boxplots show expression changes over control in actual perturbed cells, and red points show the corresponding mean changes predicted by generated cells.

G Broader Impact

scPertCRL may support biological discovery by generating computational hypotheses about cellular perturbation responses and helping prioritize conditions for experimental evaluation. However, its predictive performance may degrade in unseen contexts or underrepresented populations. To prevent potential misuse, such as over-interpreting inferred mechanisms, scPertCRL must not be used as standalone evidence for clinical or safety-critical decisions. We mitigate these risks by explicitly stating our causal limitations and emphasizing that all computational insights require independent experimental validation.

H Large Language Model Usage

Pretrained Qwen features are used as frozen gene-perturbation embeddings, as described in Appendix D.5; the pretrained model is not fine-tuned as part of scPertCRL training. Separately, LLMs assisted with language clarity, literature search queries, and understanding technical concepts. The authors are responsible for verifying generated suggestions and for the scientific claims, analyses, and final manuscript.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction present scPertCRL as a predictive framework with structured, model-based interpretation of perturbation effects. The Identifiability and Limitations subsection and the additional analyses distinguish these interpretations from recovery of biological causal graphs.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The Identifiability and Limitations subsection and Appendix B discuss the assumptions and limits of latent causal interpretation. Appendix C and the sensitivity analyses describe computational costs and the scope of the empirical findings.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A].

Justification: We do not claim a new identifiability theorem or provide a new theoretical guarantee. Appendix B explains the connection to cited results and distinguishes their assumptions from properties established for the trained model.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: We provide the scPertCRL code at `https://github.com/fdu-wangfeilab/scPertCRL`. Algorithm 1 specifies the training procedure, and Appendix D documents the data sources, preprocessing, condition splits, architecture, loss coefficients, training settings, and evaluation procedures needed to reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The code for scPertCRL is publicly available at `https://github.com/fdu-wangfeilab/scPertCRL.git`. The paper cites the public data sources, and Appendix D describes data preprocessing, model implementation, training settings, and evaluation procedures.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Appendix D specifies the datasets, training and test splits, evaluation metrics, model architecture, loss coefficients, AdamW optimizer, learning rate, batch and cell-set sizes, and two-stage training schedule with early stopping. The scPertCRL hyperparameters are kept fixed across datasets, and baseline training settings are described alongside the corresponding implementations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes].

Justification: The main benchmark results are reported as mean ± standard deviation across five runs (seeds 0–4), with split variation where applicable, initialization, and optimization described in Appendix D. Supplemental analyses state their aggregation or permutation procedures; these standard deviations are not confidence intervals or claims of statistically significant differences between methods.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix D states that computational experiments use a single NVIDIA GeForce RTX 3090 GPU. Appendix C specifies its 24 GB memory capacity and reports representative per-epoch times for both training stages and total training times across cell-set and gene-set sizes, providing hardware and runtime information for the evaluated configurations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work develops a computational prediction method using previously released perturbation datasets, without new participant recruitment or biological interventions. Potential misuse and the need for independent validation are discussed in Appendix G.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Appendix G discusses potential scientific benefits, risks of over-interpreting predictions, reduced reliability in unseen contexts, and the need for independent validation before clinical or safety-critical use.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The contribution is a computational perturbation-response predictor evaluated on existing datasets. The paper does not introduce a scraped dataset or release a new general-purpose pretrained language model; limitations on interpreting its biological predictions are discussed in Appendix G.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We credit the original creators of the datasets, models, and baseline methods through citations and use these assets in accordance with their applicable licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The scPertCRL repository is publicly linked, and Appendix D describes the model and experimental procedures.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The study uses existing perturbation datasets and includes no crowdsourcing, participant recruitment, or new experiments involving direct interaction with human participants.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The work is a secondary computational analysis of previously released datasets and involves no new participant recruitment or intervention. It does not collect new identifiable participant data.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes].

    Justification: Appendix D.5 describes the use of frozen Qwen-derived features as gene-perturbation embeddings in the method. Appendix H separately discloses LLM assistance with manuscript preparation and technical understanding; no claim is made that LLMs are absent from the methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.