URFPert: Unrolled Regulatory Flow Network for Out-of-Distribution Single-Cell Perturbation Prediction

Xiaoqi Sheng[1,†], **Jiawen Liu**[2,†], **Yutong Li**[1,†], **Sankar Mondal**[1], **Jiaming Liang**[2], **Tinghe Zhang**[1,*] and **Hongmin Cai**[1,2,*]

[1]School of Future Technology, South China University of Technology, Guangzhou, China
[2]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
zhangth@scut.edu.cn, hmcai@scut.edu.cn

Abstract

Single-cell perturbation screens enable systematic discovery of gene regulatory mechanisms, yet the exponential expansion of perturbation space makes comprehensive experimentation impractical. Although *in silico* predictors have been increasingly proposed to address this challenge, most existing methods either assume static regulatory priors or learn unconstrained data-driven representations. As a result, they often suffer from poor adaptation to state-dependent shifts and weak out-of-distribution generalization. In response, we propose URFPert, an unrolled regulatory flow network tailored for single-cell perturbation prediction. Specifically, URFPert is composed of three fundamental modules: (i) Regulatory Structure Unrolling module, which employs an iterative unrolling optimization algorithm to infer state-dependent gene regulatory networks (GRNs); (ii) Heterogeneous Graph Flow Network, which incorporates the inferred GRNs into a conditional velocity field to yield state-consistent and perturbation-specific trajectories; (iii) Discriminative Optimization Strategy, which combines Mean Squared Error (MSE) with Maximal Coding Rate Reduction (MCR^2) to promote discriminative perturbation responses and mitigate training confounders. Comprehensive evaluations on five single-cell perturbation datasets demonstrate that URFPert outperforms state-of-the-art methods in unseen perturbation regimes, providing a powerful tool for interpreting regulatory mechanisms. The project code is public via https://github.com/Jiniretttt/URFPert.

1 Introduction

Single-cell perturbation sequencing provides a powerful tool to characterize how genetic perturbations reshape cellular transcriptional states at scale. The accurate interpretation of how perturbation responses influence diverse gene functions and regulatory mechanisms is essential for enhancing our comprehension of fundamental biology [Dixit *et al.*, 2016; Bunne *et al.*, 2023]. Nevertheless, given the exponential growth of potential genetic combinations, it is impractical to systematically screen all possible combinations through experimental methods [Lotfollahi *et al.*, 2023]. As a result, developing *in silico* models for the accurate prediction of cellular perturbation effects has become necessary.

A variety of approaches have been developed for modeling single-cell perturbation. Early statistical frameworks such as MIMOSCA [Dixit *et al.*, 2016] and hypothesis-testing pipelines such as SCEPTRE [Barry *et al.*, 2021] quantify perturbation effects from scRNA-seq readouts. More recently, learning-based methods have been introduced to improve experimental annotations and trajectory-level reasoning, including HiDDeN [Goeva *et al.*, 2024] for perturbation label refinement and CellOT [Bunne *et al.*, 2023] for inferring perturbation-driven state transitions from unpaired single-cell measurements. However, these methods are developed and evaluated in in-distribution settings and do not explicitly aim for generalization to unseen perturbations.

Practical deployment increasingly relies on out-of-distribution (OOD) generalization. This is crucial because only a small fraction of the combinatorial perturbation space has been experimentally measured. *In silico* predictors must extend beyond in-distribution settings to address unseen perturbations and their combinations. Recent foundation models such as scGPT [Cui *et al.*, 2024] learn general-purpose gene and cell-state representations from millions of cells, providing strong encoders for perturbation response prediction. However, they often fail to generalize to unseen perturbations, as their static representations assume invariant regulatory dependencies across cell states. To address OOD generalization, GEARS [Roohani *et al.*, 2024] incorporates a prior regulatory graph to guide perturbation response prediction. However, such priors are typically static and may not capture state-dependent regulatory remodeling under perturbation, making performance sensitive to graph coverage and quality. Scouter [Zhu and Li, 2025] instead leverages LLM-derived gene embeddings to handle unseen genes without relying on complete graph coverage. Nevertheless, both approaches only weakly adapt responses across cellular contexts, which can limit stability under distribution shifts.

To address the challenges of modeling cellular perturbation responses, we propose URFPert, a framework that simultaneously captures regulatory structures and cellular dynamics.

*Corresponding author.

By treating the transformation from controlled to perturbed states as a continuous-time conditional flow, URFPert learns a state-dependent, heterogeneous gene regulatory network (GRN) along the cellular trajectory. Specifically, we model perturbation responses as flows driven by ordinary differential equations (ODE), with the vector field of perturbation response learned via conditional flow matching. A crucial innovation of URFPert is the dynamic inference of a state-dependent GRN during ODE integration, rather than relying on a static regulatory network. This is achieved through the Regulatory Structure Unrolling (RSU) module, which infers GRN through state-aware attention and sparsity-constrained unrolled proximal optimization. Building on the inferred GRN, URFPert utilizes the Heterogeneous Graph Flow Network (HGFN), which integrates continuous condition vectors with perturbations conditions to generate a time-consistent velocity field adaptive to evolving regulatory topologies.

Finally, we employ a Discriminative Optimization Strategy (DOS) to train URFPert, coupling velocity supervision (MSE) with Maximal Coding Rate Reduction (MCR^2) regularization. This strategy ensures the learning of perturbation-specific trajectories with high transcriptomic fidelity. We conduct experiments under explicit OOD settings with unseen single and paired genetic perturbations. Across multiple benchmarks, URFPert consistently outperforms state-of-the-art (SOTA) models and provides interpretable regulatory relationships that align with known biology. Our main contributions are summarized as follows:

- We propose URFPert, a framework that couples structure and dynamics via continuous-time conditional flow to model perturbation responses. The model is guided by evolving heterogeneous regulatory networks, enabling adaptive learning of cellular state transitions.
- We introduce RSU, a heterogeneous graph structure learning module based on deep unrolled proximal optimization, which infers state-dependent and interpretable GRNs from biological priors with an explicit sparsity–fit trade-off.
- An effective state-aware module, termed HGFN, is designed to condition the neural velocity field on inferred gene regulatory networks and perturbation conditions, yielding time-consistent and perturbation-specific dynamics for out-of-distribution prediction.
- Comprehensive benchmark evaluations show that the proposed model consistently surpasses SOTA baselines under OOD settings, encompassing unseen single and combinatorial perturbations.

2 Preliminaries

This section introduces the essential preliminaries for this study. A summary of the notation used in this paper is provided in Appendix A.

2.1 Problem Definition

Let $\mathcal{D} = \{(\mathcal{X}_0, \mathcal{X}_1, \mathcal{P})\}$ denote a single-cell perturbation dataset, where $\mathcal{X}_0 = \{\mathbf{x}_0^i \in \mathbb{R}^g\}_{i=1}^{N_0} \sim \pi_0$ represents the population of unperturbed control cells, and $\mathcal{P} = \{p_1, p_2, \ldots, p_Q\}$ is the set of Q distinct perturbation conditions. The perturbed population $\mathcal{X}_1 = \{\mathbf{x}_1^j\}_{j=1}^{N_1} \sim \pi_1(\cdot|p_j)$ represents cells under perturbation condition $p_j \in \mathcal{P}$. Here, g denotes the number of genes in $\mathcal{D}$. For single-cell perturbation prediction tasks, the core objective is to learn a parameterized predictor f_θ conditioned on p, which can reliably map unperturbed cell profiles $\mathcal{X}_0$ to corresponding perturbed profiles $\mathcal{X}_1$ across diverse cellular contexts.

To guide the model towards learning biologically consistent perturbation responses, we inject GRNs as prior knowledge. Formally, URFPert models the GRNs as a *directed heterogeneous graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{R}, \phi, \psi)$ with an adjacency matrix $\boldsymbol{A}$, in which $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and directed edges, respectively. Each node $n \in \mathcal{V}$ is assigned a node type via $\phi : \mathcal{V} \to \mathcal{T}$, where the node-type set $\mathcal{T} = \{\text{TF, R, G}\}$ corresponds to Transcription Factors (TF), Regulon (R), and Genes (G) [Van de Sande *et al.*, 2020]. Therein, each edge $E \in \mathcal{E}$ is assigned an edge type via $\psi : \mathcal{E} \to \mathcal{R}$, where the relation set $\mathcal{R}$ is defined as: (i) **Activates (TF-R)**: a TF activates a regulon (regulatory program); (ii) **Regulates (TF-G)**: candidate TF-gene interaction inferred by co-expression; (iii) **Includes (R-G)**: gene membership within a specific regulon.

Given the definitions above, the adjacency matrix $\boldsymbol{A}$ can be decomposed into a set of relation-specific sub-adjacency matrices. Specifically, each sub-adjacency matrix $\boldsymbol{A}^r$ corresponds to a unique interaction type r in the relation set $\mathcal{R}$, leading to the formalization: $\boldsymbol{A} = \{\boldsymbol{A}^r \mid r \in \mathcal{R}\}$.

2.2 Perturbation Dynamics

A Conditional Flow Matching (CFM) method [Esser *et al.*, 2024] is used to parameterize the perturbation-dependent transition function f_θ. Specifically, the f_θ is defined implicitly as the solution of an ODE driven by a neural velocity field v_θ. This field models the continuous time dynamics transforming an unperturbed cell distribution π_0 into a perturbed distribution π_1 under a given perturbation condition p.

Given paired observations of an unperturbed cell $\mathbf{x}_0^i \sim \pi_0$ and its corresponding perturbed state $\mathbf{x}_1^i \sim \pi_1(\cdot \mid p_i)$, a continuous trajectory can be defined as $\{\mathbf{x}_i^t\}_{t\in[0,1]}$ governed by the following neural ODE:

$$d\mathbf{x}_i^t = v_\theta\big(\mathbf{x}_t^i, t, p_i, \boldsymbol{A}_t\big)\, dt, \quad t \in [0,1], \tag{1}$$

where $\boldsymbol{A}_t$ is the cellular regulatory graph at time t and v_θ denotes a learnable velocity field conditioned on p_i and $\boldsymbol{A}_t$.

Training Objective. Following [Liu *et al.*, 2022], a simple straight line path is constructed using the interpolation operation between the source and target samples:

$$\mathbf{x}_t^i = (1-t)\,\mathbf{x}_0^i + t\,\mathbf{x}_1^i, \quad t \sim \mathcal{U}[0,1]. \tag{2}$$

Differentiating Eq. (2) with respect to time yields a constant conditional velocity field:

$$v\left(\mathbf{x}_t^i, t, p_i, \boldsymbol{A}_t\right) = \frac{d\mathbf{x}_t^i}{dt} = \mathbf{x}_1^i - \mathbf{x}_0^i, \tag{3}$$

which serves as the regression target for the perturbation velocity field v_θ. The v_θ can be optimized by minimizing the conditional flow matching objective:

$$\mathcal{L}(\theta) = \mathbb{E}\|v_\theta\big(\mathbf{x}_t^i,\, t,\, p_i, \boldsymbol{A}_t\big) - v\big(\mathbf{x}_t^i,\, t,\, p_i, \boldsymbol{A}_t\big)\,\|_2^2. \tag{4}$$

Inference. At the inference phase, the perturbed cell state can be obtained by integrating the learned velocity vector from $t = 0$ to $t = 1$:

$$\hat{\mathbf{x}}_1^i = \mathbf{x}_0^i + \int_0^1 v_\theta\left(\mathbf{x}_t^0, t, p_i, \boldsymbol{A}_t\right)\, dt. \tag{5}$$

The resulting ODE in Eq. (5) is solved numerically using the Runge–Kutta method [Chen *et al.*, 2018] or Euler method.

3 Methods

3.1 Overview

Fig. 1 illustrates the overall framework of the proposed URFPert. The training phase of URFPert (Fig. 1a) consists of three key components: the Regulatory Structure Unrolling (RSU) module, the Heterogeneous Graph Flow Network (HGFN), and the Discriminative Optimization Strategy (DOS). The RSU infers state-dependent GRNs by unrolling a proximal optimization procedure. It allows the regulatory topology to adapt dynamically to the current perturbation cell state $\mathbf{x}_t$. Conditioned on the inferred GRNs, the HGFN models perturbation-driven dynamics by parameterizing the state-transition velocity field. Finally, the DOS jointly minimizes MSE and MCR^2 to preserve cross-perturbation separability and prevent representation collapse. During the Inference Phase (Fig. 1b), URFPert predicts the perturbed transcriptional state x_1 by numerically integrating the learned velocity field from $t = 0$ to $t = 1$.

3.2 Regulatory Structure Unrolling

The RSU module encodes the regulatory structure from a prior biological-relation adjacency matrix $\boldsymbol{A}$ generated by pySCENIC [Van de Sande *et al.*, 2020], together with the semantic context of gene interactions from the cell population $\mathcal{X}_\mathrm{t} = \{\mathbf{x}_t^i\}_{i=1}^{N_t}$ at time t. Through unrolled optimization for heterogeneous graph learning, the RSU iteratively updates relation-specific adjacencies to recover a state-conditioned regulatory topology.

Specifically, the initial cell representation $H_t \in \mathbb{R}^{g\times h}$ is built by integrating gene identity with scalar expression values. For gene identity, the RSU utilizes gene embeddings $e_{id} \in \mathbb{R}^{g\times d}$ from scGPT as a frozen lookup table and projects them to the hidden dimension via a learnable matrix $W_{id} \in \mathbb{R}^{d\times h}$: $h_{id} = e_{id}W_{id} \in \mathbb{R}^{g\times h}$. Then, the representation H_i is formed by addition:

$$H_t = \mathbf{x}_t^i W_s + h_{id} \in \mathbb{R}^{g\times h}, \tag{6}$$

where $W_s \in \mathbb{R}^{g\times h}$ is a learnable weight. For regulation relation $r \in \mathcal{R}$, the relation-specific node representation is calculated as $H^r = H_t^r W^r \in \mathbb{R}^{g\times h}$, where $W^r \in \mathbb{R}^{h\times h}$ is a learnable weight to capture interaction patterns specific to relation r. Specifically, H_t^r denotes the node representations specific to relation graph $\boldsymbol{A}^r$. Using these relation-aware embeddings, the data-driven gene association matrix $C^r \in \mathbb{R}^{g\times g}$ is constructed using additive attention [Busbridge *et al.*, 2019]. In detail, for any gene pair (m, n), the attention score quantifying their association strength under relation r is defined as follows:

$$\boldsymbol{C}_{m,n}^r = \tanh\left(\text{LeakyReLU}\left(H_m^r a_m^r + H_n^r a_n^r\right)\right), \tag{7}$$

where H_m^r and H_n^r are the relation-aware features of the source and target genes, respectively. The weights $a_m^r, a_n^r \in \mathbb{R}^{h\times 1}$ are learned separately for each biological relation $r \in \mathcal{R}$. Here, $\tanh(\cdot)$ and $\text{LeakyReLU}(\cdot)$ are tanh and LeakyReLU functions, respectively. Following Graph Structure Learning [Saboksayr and Mateos, 2021], the optimal relation-specific GRN adjacency is obtained by solving:

$$\arg\min_{A^r} \langle \boldsymbol{A}^r, -\boldsymbol{C}^r\rangle + \frac{\beta_r}{2}\|\boldsymbol{A}^r\|_F^2 + \lambda_r\|\boldsymbol{A}^r\|_1, \tag{8}$$

where the inner product $\langle \boldsymbol{A}^r, -\boldsymbol{C}^r\rangle$ maximizes the structural alignment between the learned adjacency matrix $\boldsymbol{A}^r$ and the inferred structure $\boldsymbol{C}^r$. The Frobenius penalty $\frac{\beta_r}{2}\|\boldsymbol{A}^r\|_F^2$ provides smoothness that suppresses noise and stabilizes estimation, while the L_1 penalty $\|\boldsymbol{A}^r\|_1$ promotes sparsity by pruning weak or indirect interactions. The RSU uses the proximal gradient descent method to solve Eq. (8), iteratively refining $\boldsymbol{A}^r$. At iteration k, the $\boldsymbol{A}_k^r$ is updated by a gradient step on the smooth component followed by a proximal mapping $\text{prox}_{\Gamma q}(\cdot)$ for the non-smooth L_1 term:

$$\boldsymbol{A}_{k+1}^r = \text{prox}_{\Gamma q}(\boldsymbol{A}_k^r - \eta_r \nabla f(\boldsymbol{A}_k^r)), \tag{9}$$

where $f(\boldsymbol{A}^r) = \langle \boldsymbol{A}^r, -\boldsymbol{C}^r\rangle + \frac{\beta_r}{2}\|\boldsymbol{A}^r\|_F^2$ is the differentiable component and $q(\boldsymbol{A}^r) = \lambda_r\|\boldsymbol{A}^r\|_1$ is the non-differentiable term. Let η_r be the step size constant, $\Gamma = \eta_r\lambda_r$ be the threshold and $Y = \boldsymbol{A}_k^r - \eta_r\nabla f(\boldsymbol{A}_k^r)$, then the $\text{prox}_{\eta_r q}(Y)$ can be defined as:

$$\text{prox}_{\eta_r q(\cdot)}(Y) = \arg\min_X \frac{1}{2}\|Y - X\|_F^2 + \eta_r\lambda_r\|X\|_1. \tag{10}$$

The optimization problem of Eq. (10) admits a closed-form solution known as the soft-thresholding operator [Beck and Teboulle, 2009], denoted as $\Psi_\Gamma(\cdot)$:

$$\text{prox}_{\eta_r q(\cdot)}(Y) = \Psi_\Gamma(Y) = (|Y| - \Gamma)_+ \,\text{sign}(Y), \tag{11}$$

where $(\cdot)_+ = \max(0, \cdot)$ denotes the positive-part operator and $\text{sign}(\cdot)$ is the element-wise sign function. The $\Psi_\Gamma(\cdot)$ shrinks the input magnitude by the threshold Γ while preserving its sign, thereby strictly enforcing the sparsity of the learned graph structure. Thus, the Eq. (9) can be rewritten as:

$$\begin{aligned}\boldsymbol{A}_{k+1}^r &= \text{prox}_{\Gamma q}(\boldsymbol{A}_k^r - \eta_r(-\boldsymbol{C}^r + \beta_r A_r^k)) \\ &= \Psi_\Gamma((1 - \eta_r\beta_r)\boldsymbol{A}_r^k + \eta_r\boldsymbol{C}^r),\end{aligned} \tag{12}$$

where the initial value is given as $\boldsymbol{A}_0 = \boldsymbol{A}$. Following deep unrolling technology [Monga *et al.*, 2021], the RSU transforms the iterative updating rule derived in Eq. (12) into a cascaded neural network architecture. Formally, the RSU maps each iteration step k to a corresponding network layer. Rather than using fixed parameters, the RSU treats the step size η_r, smoothness coefficient β_r, and sparsity threshold λ_r as learnable parameters. These parameters are automatically adapted from training data in an end-to-end manner. The operation of the k-th layer in RSU can be formally defined as:

$$\text{RSU-Layer}_k^r : \begin{cases}\boldsymbol{G}_k^r = \beta_k^r\boldsymbol{A}_k^r - \boldsymbol{C}^r \\ \widetilde{\boldsymbol{A}}_k^r = \boldsymbol{A}_k^r - \eta_k^r\boldsymbol{G}_k^r \\ \boldsymbol{A}_{k+1}^r = \Psi_{\Gamma_k^r}\left(\widetilde{\boldsymbol{A}}_k^r\right),\end{cases} \tag{13}$$

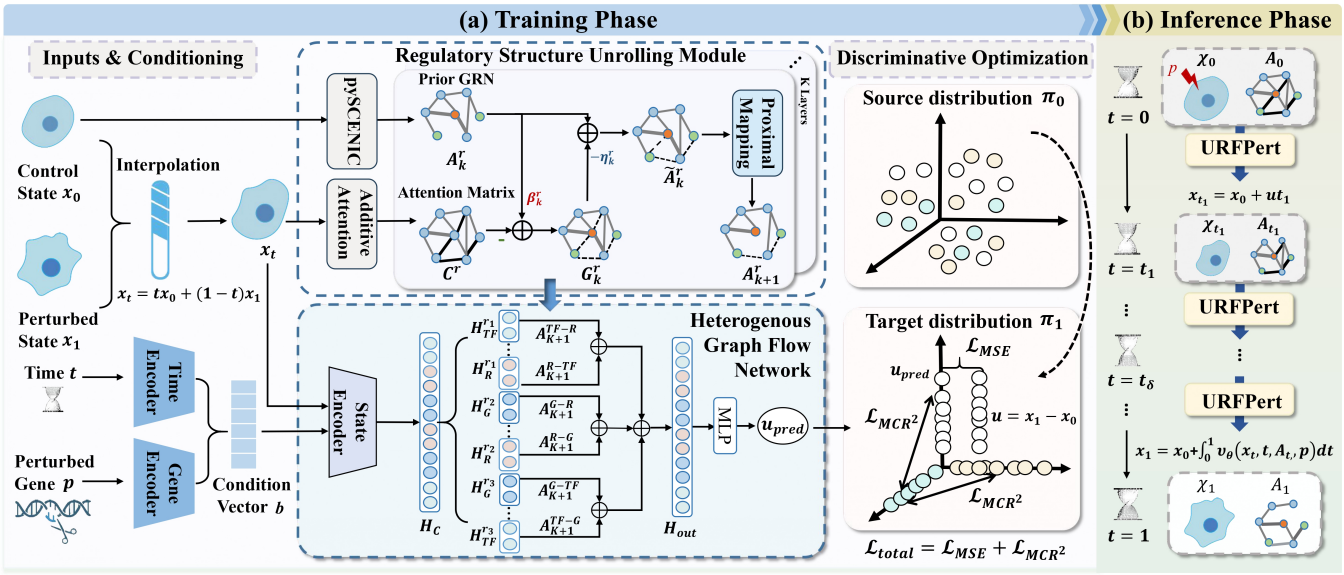

Figure 1: Overview structure of URFPert. (a) The Training Phase of URFPert comprises four components: (i) Inputs and Conditioning, (ii) a Regulatory Structure Unrolling Module, (iii) a Heterogeneous Graph Flow Network,(iv) a Discriminative Optimization Process. (b) The Inference Phase of URFPert.

where $\Gamma_k^r = \eta_k^r \lambda_k^r$ is the learnable threshold parameter for the k-th layer. The state-dependent heterogeneous GRN at time t is defined as the set $\boldsymbol{A}_t = \{\boldsymbol{A}_K^r\}_{r\in\mathcal{R}}$. It is finalized by stacking K deep unrolling layers for each specific relation type r. The $\boldsymbol{A}_t$ allows the model to capture the dynamic reconfiguration of regulatory topologies as the cellular state evolves.

3.3 Heterogeneous Graph Flow Network

After getting the current GRN $\boldsymbol{A}_t$, an HGFN is proposed to implement the neural velocity field v_θ. The velocity field v_θ maps the tuple $(H_t, t, p, \boldsymbol{A}_t)$ to the instantaneous velocity vector u_t, parameterizing the direction and magnitude of the transcriptional transition. The prediction of the velocity vector u_t is conditioned on the current cell state representation H_t, the continuous time variable t, perturbation condition p_i, and the inferred GRN structure $\boldsymbol{A}_t$ derived from RSU:

$$u_t = v_\theta(H_t, t, p, \boldsymbol{A}_t). \tag{14}$$

Specifically, the HGFN integrates H_t, t, p_i, and $\boldsymbol{A}_t$ to update the subsequent latent representation. The HGFN employs Gaussian Fourier Projection [Tancik *et al.*, 2020] to encode the $t \in [0, 1]$ into a latent space, thereby enabling the explicit modeling of continuous temporal dynamics:

$$\boldsymbol{e}_t = \left[\sin(2\pi\hat{W}t), \cos(2\pi\hat{W}t)\right], \tag{15}$$

where $\hat{W} \in \mathbb{R}^{g\times h}$ is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, with the scaling factor σ empirically fixed at 30.0. Subsequently, the perturbation state p_i is encoded via a Gene Encoder $\mathcal{F}_1$: $e_{pert} = \mathcal{F}_1(p) \in \mathbb{R}^{g\times h}$. Therein, the $\mathcal{F}_1$ is implemented by a Multi-Layer Perceptron (MLP), utilizing scGPT embeddings (Appendix I). Next, the HGFN concatenates e_{time} and e_{pert} to generate the unified conditional embedding: $\boldsymbol{b} = [\boldsymbol{e}_{time}||\boldsymbol{e}_{pert}] \in \mathbb{R}^{g\times 2h}$, which provides a global context to ensure the learned dynamics are both time-consistent and perturbation-specific.

Finally, the HGFN utilizes a state encoder SE to process cellular contexts and gene expression to get an initial context-aware representation H_c:

$$H_c = SE(H_i \| \boldsymbol{b}) \in \mathbb{R}^{g\times h}, \tag{16}$$

The HGFN captures regulatory influence by aggregating H_c with the state-dependent GRNs derived from the RSU. Since the adjacency matrices $\boldsymbol{A}^r$ already quantify regulatory intensities, a parameter-efficient direct summation is adopted:

$$H_{out} = \sum_{r\in\mathcal{R}} \boldsymbol{A}^r H_c^r. \tag{17}$$

Finally, the H_{out} is decoded into the global velocity vector u_{pred} via a learnable linear projection τ:

$$u_{pred} = \tau(H_{out}) \in \mathbb{R}^{g\times 1}, \tag{18}$$

where $u_{pred} \in \mathbb{R}^{g\times 1}$ denotes the estimated velocity vector for all g genes, and $\tau : \mathbb{R}^{g\times h} \to \mathbb{R}^{g\times 1}$ maps the latent features back to the original expression space.

3.4 Discriminative Optimization Strategy

Following the velocity estimation u_{pred} via HGFN, we introduce DOS to supervise the learning of v_θ via a dual-objective formulation. The DOS jointly enforces trajectory fidelity and enhances discriminability in the latent space, thereby improving sensitivity to subtle perturbation-specific gene signatures while maintaining accurate state-transition dynamics. Specifically, the DOS first minimizes the MSE between the estimated vector u_{pred} and the ground-truth vector u:

$$\mathcal{L}_{MSE} = \frac{1}{g}\sum_{i=1}^{g}(u_{pred}^i - u^i)^2, \tag{19}$$

where $u = \mathbf{x}_1^i - \mathbf{x}_0^i$. Then, the DOS incorporates MCR^2 [Yu *et al.*, 2020] to enhance the discriminability and structural organization of perturbation-specific trajectories. To avoid interfering with ODE dynamics, the MCR^2 is computed on the final cell representation H_{out} instead of the velocity field u_{pred}. The fundamental premise of MCR^2 is that high-dimensional gene expression profiles lie on the mixture of lower-dimensional representation $\boldsymbol{H}_{out}$. For a prescribed distortion tolerance ϵ, the coding cost of $\boldsymbol{Z} = [H_{out}^1, \cdots, H_{out}^{N_1}]$ is bounded by the log-determinant of the covariance matrix:

$$M(\boldsymbol{Z}, \epsilon) = \frac{1}{2} \log\det(\boldsymbol{I} + \frac{d_z}{N_1 \epsilon^2} \boldsymbol{Z}\boldsymbol{Z}^\top), \tag{20}$$

where $\boldsymbol{I}$ is the identity matrix, $d_z = g \times h$ denotes the dimension of each entry of $\boldsymbol{Z}$ and ϵ is the tolerated reconstruction error (default set to 0.01). The coding rate reduction ΔM quantifies the difference between the global coding rate $M(\boldsymbol{Z}, \epsilon)$ (Expansion) and the weighted sum of individual cluster coding rates (Compression). The ΔM is maximized by the DOS to learn discriminative features for perturbation trajectories. Let $\Pi = \{\Pi_j \in \mathbb{R}^{N_1 \times N_1}\}_{j=1}^{Q}$ be a set of diagonal matrices whose diagonal entries encode the membership of the N_1 samples in the Q perturbation types. The regularization objective is formulated as:

$$\mathcal{L}_{MCR^2} = -\Delta M(\boldsymbol{Z}, \Pi, \epsilon) = M^p(\boldsymbol{Z}, \epsilon|\Pi) - M(\boldsymbol{Z}, \epsilon), \tag{21}$$

where the compression term M^P is the sum of coding rates for each perturbation trajectory j:

$$M^P(\boldsymbol{Z}, \epsilon|\Pi) = \sum_{j=1}^{Q} \frac{\mathrm{tr}(\Pi_j)}{2N_1} \log\det(\boldsymbol{I} + \frac{d_z}{\mathrm{tr}(\Pi_j)\epsilon^2} \boldsymbol{Z}\Pi_j\boldsymbol{Z}^\top). \tag{22}$$

To jointly optimize for both trajectory alignment and latent discriminability, the final training objective of the velocity field v_θ modeled by URFPert is defined as:

$$\arg\min_{v_\theta} \mathcal{L}_{total} = \arg\min_{v_\theta} (L_{MSE} + \mathcal{L}_{MCR^2}). \tag{23}$$

Once trained, the predicted perturbation state $\hat{x}_i^1$ is obtained by numerically integrating the v_θ:

$$\hat{\mathbf{x}}_i^1 = \mathbf{x}_i^0 + \int_0^1 v_\theta\left(\mathbf{x}_t^0, t, p_i, \boldsymbol{A}_t\right) dt. \tag{24}$$

4 Experiments

In this work, we evaluated the predictive performance of URFPert by benchmarking it against five SOTA computational frameworks across five diverse genetic perturbation datasets, including three unseen single perturbation (Papalexi [Papalexi *et al.*, 2021], Adamson and TianAct [Tian *et al.*, 2021]) and two unseen double perturbation (Wessels [Wessels *et al.*, 2023] and Replogle [Replogle *et al.*, 2022]). We report performance using five complementary metrics at both the top-100 differentially expressed genes (DEG) level and the all-gene level. Dataset details are provided in Appendix D.

4.1 Experimental Setup

Implementation Details. URFPert was implemented with PyTorch 2.9.0 and trained on four NVIDIA RTX PRO 6000 Blackwell Workstation Edition GPUs with CUDA 13.0 support. It was trained using the AdamW [Loshchilov and Hutter, 2017] optimizer to minimize the loss $\mathcal{L}_{total}$, with a learning rate of $1e{-}4$, a decay factor of 0.1, over 600 training epochs, and a batch size of 256. Comprehensive model settings and details are respectively elaborated in Appendix C and Appendix I.

Comparative Settings. In order to provide a comprehensive evaluation of URFPert, we benchmarked URFPert against five SOTA baselines spanning a broad range of modeling architectures: Linear Model [Ahlmann-Eltze *et al.*, 2025], Gears, scGPT, GenePert [Chen and Zou, 2025], and scouter.

Data Splitting and Metrics. We adopted a specific partitioning strategy based on perturbation type. For single perturbation datasets, perturbation conditions were randomly partitioned into training/validation/test sets with a $70\%/10\%/20\%$ ratio.

For the combined perturbations dataset, we followed the setting of [Wei *et al.*, 2025]: all single perturbations and 50% of randomly sampled dual-gene combinations were used for training, while the remaining 50% of unseen combinations were evenly split between the validation and test sets.

The systematic benchmark protocols [Wei *et al.*, 2025] evaluated perturbation methods using five metrics: MSE, Pearson Correlation Coefficient delta (ΔPCC), Energy Distance (E-Dist), Wasserstein Distance (W-Dist), and common differentially expressed genes (C-DEGs) over both the top 100 DEGs and the full gene set. At the all gene level, W-Dist becomes computationally prohibitive, and C-DEG scores tend to saturate, reducing their discriminative power [Wei *et al.*, 2025]. Therefore, for all-gene evaluation, we omit W-Dist and replace C-DEGs with the Differential Expression Score (DES), as adopted in [Roohani *et al.*, 2025]. Detailed mathematical definitions and calculation procedures for all evaluation metrics are provided in the Appendix K. To ensure fair comparison and mitigate split-induced bias, we evaluated all methods over ten independent trials with different random seeds for dataset splitting, while keeping the splitting ratio fixed. In each trial, each model was optimized on the training set, and the validation set was utilized for hyperparameter tuning. The trained model was subsequently applied to the test dataset to evaluate the performance.

4.2 Overall Performance

The results of all metrics across the top 100 DEGs level and all genes level are reported in Table 1. Overall, our method achieves the best overall performance across five perturbation benchmarks at both levels.

Performance on Single-Gene Perturbations. URFPert consistently outperforms SOTA baselines across both evaluation levels, demonstrating its capability in modeling perturbation-induced transcriptional responses. On Papalexi, URFPert achieves the highest C-DEGs score (11.08) comparing over the best competing method (0.88), while most baselines remain near zero. On Adamson, URFPert obtains

Dataset	Method	Top 100 DEGs Level					All Gene Level			
		C-DEGs↑	ΔPCC↑	E-Dist↓	W-Dist↓	MSE↓	DES↑	ΔPCC↑	E-Dist↓	MSE↓
Papalexi	Gears	0.000 ± 0.000	0.266 ± 0.139	0.229 ± 0.126	50.505 ± 6.464	0.019 ± 0.012	0.020 ± 0.018	0.306 ± 0.107	0.532 ± 0.226	0.003 ± 0.002
	scGPT	0.000 ± 0.000	0.030 ± 0.093	3.051 ± 0.370	41.650 ± 7.421	0.028 ± 0.016	0.012 ± 0.011	0.215 ± 0.072	10.127 ± 0.436	0.005 ± 0.002
	GenePert	0.400 ± 0.316	-0.014 ± 0.049	0.214 ± 0.079	57.493 ± 5.519	0.019 ± 0.008	0.003 ± 0.006	0.177 ± 0.059	0.325 ± 0.100	0.002 ± 0.001
	scouter	0.050 ± 0.112	0.265 ± 0.219	8.968 ± 0.531	37.594 ± 5.262	0.016 ± 0.012	0.059 ± 0.035	0.421 ± 0.250	2.458 ± 0.288	0.003 ± 0.002
	linearModel	0.880 ± 0.415	-0.107 ± 0.184	0.335 ± 0.153	55.900 ± 6.275	0.028 ± 0.013	0.027 ± 0.024	0.187 ± 0.106	0.610 ± 0.248	0.003 ± 0.002
	OurModel	11.080 ± 6.975	0.978 ± 0.015	0.078 ± 0.044	14.053 ± 5.223	0.004 ± 0.003	0.112 ± 0.063	0.990 ± 0.005	0.231 ± 0.131	0.002 ± 0.001
Adamson	Gears	0.000 ± 0.000	0.208 ± 0.124	0.451 ± 0.098	37.681 ± 0.156	0.030 ± 0.008	0.007 ± 0.005	0.306 ± 0.102	0.491 ± 0.072	0.002 ± 0.004
	scGPT	0.000 ± 0.000	0.155 ± 0.098	2.916 ± 0.187	27.680 ± 1.533	0.030 ± 0.013	0.006 ± 0.005	0.412 ± 0.061	10.596 ± 0.495	0.002 ± 0.001
	GenePert	2.163 ± 0.480	0.260 ± 0.065	0.513 ± 0.160	41.061 ± 1.906	0.036 ± 0.012	0.039 ± 0.012	0.412 ± 0.051	0.510 ± 0.106	0.002 ± 0.001
	scouter	0.036 ± 0.050	0.141 ± 0.112	2.451 ± 0.258	29.471 ± 2.318	0.047 ± 0.018	0.026 ± 0.012	0.340 ± 0.083	8.298 ± 0.195	0.003 ± 0.001
	linearModel	2.186 ± 0.283	0.182 ± 0.061	0.571 ± 0.126	41.793 ± 1.290	0.041 ± 0.010	0.000 ± 0.000	0.384 ± 0.056	0.523 ± 0.098	0.003 ± 0.001
	OurModel	2.469 ± 0.282	0.986 ± 0.002	0.093 ± 0.008	10.206 ± 0.762	0.004 ± 0.000	0.106 ± 0.064	0.995 ± 0.001	0.163 ± 0.033	0.001 ± 0.000
TianAct.	Gears	0.053 ± 0.053	0.102 ± 0.043	0.297 ± 0.089	43.767 ± 1.332	0.016 ± 0.007	0.000 ± 0.000	0.310 ± 0.037	0.735 ± 0.230	0.004 ± 0.001
	scGPT	0.011 ± 0.024	0.109 ± 0.023	2.766 ± 0.091	33.518 ± 2.178	0.020 ± 0.007	0.000 ± 0.000	0.328 ± 0.031	8.869 ± 0.107	0.002 ± 0.001
	GenePert	0.000 ± 0.000	0.083 ± 0.068	0.267 ± 0.079	46.068 ± 1.258	0.015 ± 0.007	0.000 ± 0.000	0.372 ± 0.037	0.429 ± 0.064	0.001 ± 0.000
	scouter	0.200 ± 0.136	0.136 ± 0.026	2.859 ± 0.069	31.968 ± 0.706	0.014 ± 0.004	0.000 ± 0.000	0.474 ± 0.036	9.338 ± 0.164	0.001 ± 0.000
	linearModel	0.000 ± 0.000	0.093 ± 0.029	0.268 ± 0.069	45.174 ± 1.442	0.016 ± 0.006	0.041 ± 0.015	0.404 ± 0.023	0.429 ± 0.060	0.001 ± 0.000
	OurModel	6.811 ± 4.440	0.946 ± 0.025	0.191 ± 0.089	11.878 ± 2.052	0.009 ± 0.005	0.043 ± 0.025	0.994 ± 0.001	0.240 ± 0.072	0.002 ± 0.001
Wessels	Gears	0.000 ± 0.000	0.672 ± 0.163	0.611 ± 0.194	49.403 ± 1.623	0.039 ± 0.016	0.001 ± 0.000	0.706 ± 0.098	1.065 ± 0.219	0.004 ± 0.001
	scGPT	0.000 ± 0.000	0.420 ± 0.135	6.205 ± 0.257	110.257 ± 1.050	0.104 ± 0.020	0.000 ± 0.000	0.645 ± 0.046	27.675 ± 0.245	0.013 ± 0.002
	GenePert	0.000 ± 0.000	0.736 ± 0.019	0.795 ± 0.121	54.560 ± 1.830	0.056 ± 0.010	0.000 ± 0.000	0.722 ± 0.026	0.927 ± 0.073	0.004 ± 0.000
	scouter	0.181 ± 0.215	0.425 ± 0.082	4.198 ± 0.372	44.270 ± 2.954	0.113 ± 0.026	0.001 ± 0.000	0.522 ± 0.043	9.784 ± 0.411	0.006 ± 0.001
	linearModel	0.000 ± 0.000	0.364 ± 0.098	4.881 ± 0.364	45.970 ± 2.923	0.119 ± 0.026	0.074 ± 0.012	0.398 ± 0.081	13.128 ± 0.252	0.006 ± 0.001
	OurModel	2.393 ± 0.821	0.758 ± 0.032	0.734 ± 0.112	22.719 ± 0.745	0.039 ± 0.008	0.110 ± 0.093	0.975 ± 0.001	1.186 ± 0.118	0.006 ± 0.001
Replogle	Gears	0.000 ± 0.000	0.500 ± 0.141	0.254 ± 0.048	37.195 ± 1.321	0.014 ± 0.004	0.000 ± 0.000	0.436 ± 0.063	0.471 ± 0.027	0.002 ± 0.000
	scGPT	0.000 ± 0.000	0.252 ± 0.036	2.433 ± 0.147	28.692 ± 0.803	0.022 ± 0.005	0.000 ± 0.000	0.300 ± 0.052	7.390 ± 0.274	0.002 ± 0.000
	GenePert	0.000 ± 0.000	0.400 ± 0.055	0.349 ± 0.024	166.142 ± 5.512	0.039 ± 0.004	0.000 ± 0.000	0.404 ± 0.025	0.692 ± 0.063	0.006 ± 0.000
	scouter	0.314 ± 0.110	0.287 ± 0.067	4.671 ± 0.151	113.392 ± 1.145	0.055 ± 0.010	0.000 ± 0.000	0.326 ± 0.050	17.513 ± 1.151	0.008 ± 0.002
	linearModel	0.000 ± 0.000	0.294 ± 0.054	3.180 ± 0.099	28.434 ± 1.176	0.019 ± 0.006	0.025 ± 0.007	0.294 ± 0.054	10.626 ± 0.154	0.001 ± 0.000
	OurModel	13.956 ± 5.879	0.967 ± 0.020	0.134 ± 0.051	11.176 ± 0.847	0.005 ± 0.002	0.140 ± 0.036	0.995 ± 0.001	0.180 ± 0.057	0.001 ± 0.000

Table 1: Comprehensive evaluation results on five genetic perturbation datasets (Mean ± Standard deviation). ↑ (↓) indicates that higher (lower) values are better. The best results are highlighted in red and the runner-up in blue. When mean values are tied, the entry with the smaller standard deviation is ranked higher.

ΔPCC of 0.986 for top 100 DEGs level, markedly surpassing Gears (0.260), highlighting the advantage of RSU module in dynamically inferring state-dependent gene interactions rather than relying on fixed priors. Notably, URFPert consistently ranks first in E-Dist across all single-perturbation datasets. It indicates URFPert effectively models the stochastic nature of gene expression perturbations beyond simple point-estimate accuracy.

Performance on Combined Perturbations. For the evaluations on combined perturbation (Wessels and Replogle), URFPert outperforms linear, static-prior, and embedding-centric baselines. Although the linear model captures global trends, it misses non-additive gene interactions, yielding near-zero C-DEGs scores. Unlike discrete perturbation-to-expression approaches (scouter, GenePert), URFPert models a continuous conditional velocity field, dynamically guided by unrolled, state-dependent regulatory remodeling. Enhanced by MCR^2 regularization in the Discriminative Optimization module, URFPert maintains exceptional feature fidelity, achieving a W-Dist of 11.176 and a DES of 0.140 on the Replogle dataset. Overall, these results demonstrate that coupling dynamic graph structure with continuous flow trajectories is essential for accurately modeling perturbation responses in combinatorial spaces.

4.3 Ablation Study

We conducted ablation studies to evaluate the specific contributions of the RSU, HGFN, and DOS within URFPert. Fig. 2 compares evaluation metrics across the Papalexi and Replogle datasets. For cross-metric comparison, we report normalized scores (higher is better). Full evaluation metrics and calculation details are provided in Appendix E and L.

Overall, the complete URFPert framework consistently achieves the best performance on both datasets at both evaluation levels. Excluding RSU (w/o RSU) results in significant performance reductions in C-DEGs and W-Dist scores, particularly on the Replogle dataset. This underscores the critical role of dynamic structural modeling in capturing perturbation-specific signals. Excluding HGFN (w/o HGFN) mainly degrades the performance of E-Dist at the all gene level, indicating that the absence of heterogeneous information weakens the model's ability to integrate patterns specific to diverse regulatory relations. Furthermore, we validated the necessity of the DOS by removing the MCR^2 regularization: $\mathcal{L}_{total} = \mathcal{L}_{MSE}$ (w/o MCR^2). Note that $\mathcal{L}_{MSE}$ was retained in this variant to ensure basic model convergence. The results (Fig. 2) show that removing MCR^2 degrades performance across all metrics, particularly regarding distributional fidelity (W-Dist, E-Dist). UMAP visualizations (Fig. 3) further confirm that MCR^2 effectively prevents feature collapse, producing well-separated, perturbation-specific clusters with significantly higher ARI and NMI scores. These findings underscore the necessity of MCR^2 for learning discriminative and high-fidelity latent representations.

4.4 Interpretable Analysis

We assessed the inferred regulatory structures by measuring the overlap between top-$\mathbb{K}$ predicted edges and the GO-

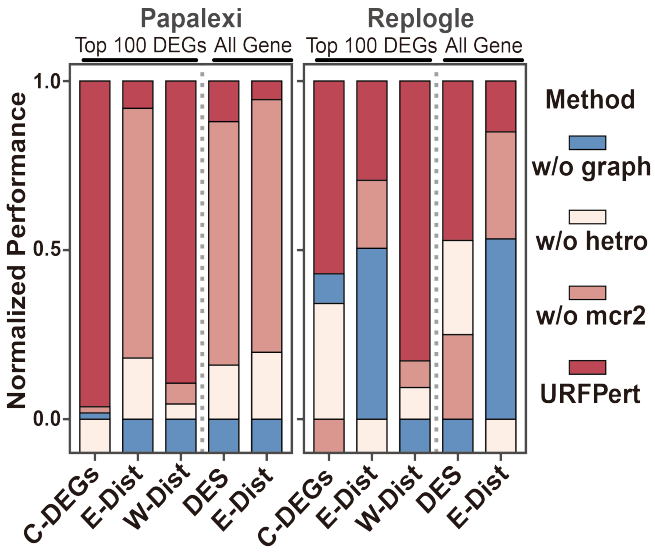

Figure 2: The results of ablation studies. Each metric is normalized, with higher bar segments indicating superior performance relative to other variants.

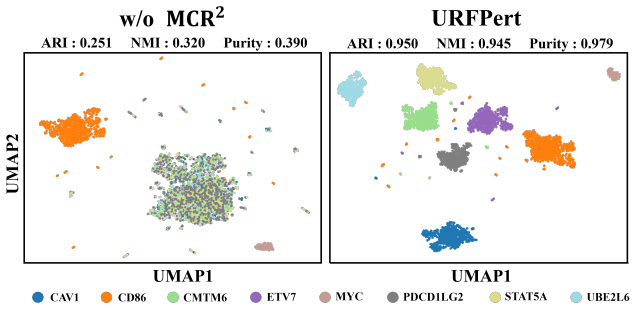

Figure 3: The UMAP comparison of cell representation H_{out} in two settings: w/o MCR^2 and URFPert.

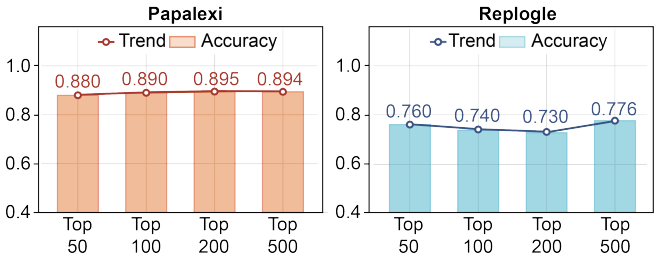

Figure 4: The accuracy assessment of gene regulatory network inference.

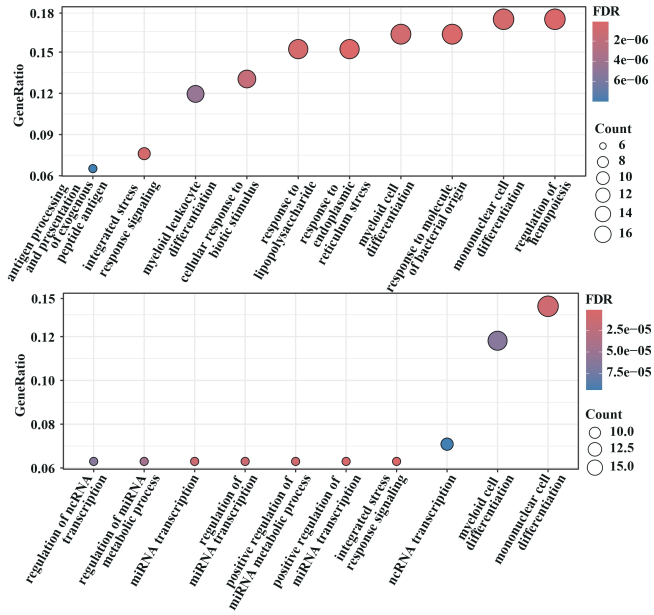

Figure 5: GO-BP enrichment analysis of top perturbation-responsive programs. The dot plots illustrate significant biological agreement between known priors and inferred GRN from MYC knockout (top) and combined ETAA1+FDPS perturbation (bottom).

derived [Consortium, 2021] co-functional network (Fig. 4). URFPert achieves high edge support across both benchmarks (up to 89.5% on Papalexi and 77.6% on Replogle), providing a principled basis for the following enrichment analyses (Appendix N). The evaluation process and metric definition are detailed in Appendix M. For the MYC single perturbation ($\mathbb{K}$ = 100), the inferred GRN (93 genes) aligns strongly with prior knowledge, with 89 of the top 100 edges supported by GO annotations (Appendix F). The network topology reflects the loss of MYC-driven proliferation and the derepression of innate immunity [Macek *et al.*, 2018], characterized by central STAT1/IRF hubs and downstream antigen-presentation (e.g., HLA-DRA) and UPR (e.g., DDIT3) modules. This structural shift is consistent with the enrichment of antigen processing and stress response pathways observed in the functional analysis (Fig. 5) [Zou *et al.*, 2018].

Similarly, the inferred GRN from ETAA1+FDPS double perturbation (142 genes) demonstrates high biological alignment when $\mathbb{K}$=100 (75/100 edges supported; Appendix F). The topology captures a convergence of replication stress (via ETAA1 loss) and metabolic stress (via FDPS-mediated mevalonate blockade) [Dhar *et al.*, 2013]. This is evidenced by high-centrality hubs for immediate early genes (JUN, EGR1) and UPR regulators (ATF4, DDIT3), which align with GO enrichment for the integrated stress response and DNA damage-associated ncRNA regulation pathways [Pakos-Zebrucka *et al.*, 2016; Wan *et al.*, 2011].

5 Conclusion

In this study, we present URFPert, an unrolled regulatory flow network designed to predict single-cell perturbation responses in OOD settings. By integrating dynamic regulatory structure learning with continuous-time cellular trajectories, URFPert effectively captures state-dependent regulatory remodeling during perturbation responses. Extensive benchmarks show URFPert excels in distributional fidelity and combinatorial modeling, offering a powerful and interpretable tool to decipher the molecular mechanisms behind complex cellular transitions.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China (2024YFF1206600); in part by Guangdong S&T Programme (2025B0101130001); in part by the National Natural Science Foundation of China (62325204, 62502161, 62502163); in part by the China Postdoctoral Science Foundation (2025M782912).

Contribution Statement

Xiaoqi Sheng: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing-Original Draft. Jiawen Liu: Conceptualization, Methodology, Software, Validation, Formal analysis, Data Curation, Writing-Original Draft. Yutong Li: Methodology, Software, Validation, Investigation, Formal analysis, Writing-Original Draft. Sankar Mondal: Validation, Formal analysis, Investigation, Writing-Review & Editing. Jiaming Liang: Software, Data Curation, Validation, Writing-Review & Editing, Visualization. Tinghe Zhang: Conceptualization, Methodology, Supervision, Formal analysis, Writing-Review & Editing. Hongmin Cai: Conceptualization, Supervision, Project administration, Funding acquisition, Writing-Review & Editing. Xiaoqi Sheng, Jiawen Liu, and Yutong Li contributed equally to this work and share first authorship. Tinghe Zhang (zhangth@scut.edu.cn) and Hongmin Cai (hmcai@scut.edu.cn) are the corresponding authors.

References

[Ahlmann-Eltze *et al.*, 2025] Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22(8):1657–1661, 2025.

[Barry *et al.*, 2021] Timothy Barry, Xuran Wang, John A Morris, Kathryn Roeder, and Eugene Katsevich. Sceptre improves calibration and sensitivity in single-cell crispr screen analysis. *Genome biology*, 22(1):344, 2021.

[Beck and Teboulle, 2009] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

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

[Busbridge *et al.*, 2019] Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. Relational graph attention networks. *arXiv preprint arXiv:1904.05811*, 2019.

[Chen and Zou, 2025] Yiqun Chen and James Zou. Simple and effective embedding model for single-cell biology built from chatgpt. *Nature biomedical engineering*, 9(4):483–493, 2025.

[Chen *et al.*, 2018] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[Consortium, 2021] The Gene Ontology Consortium. The gene ontology resource: enriching a gold mine. *Nucleic Acids Research*, 49(D1):D325–D334, 01 2021.

[Cui *et al.*, 2024] 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.

[Dhar *et al.*, 2013] Manoj K Dhar, Archana Koul, and Sanjana Kaul. Farnesyl pyrophosphate synthase: a key enzyme in isoprenoid biosynthetic pathway and potential molecular target for drug development. *New biotechnology*, 30(2):114–123, 2013.

[Dixit *et al.*, 2016] 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.

[Esser *et al.*, 2024] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[Goeva *et al.*, 2024] Aleksandrina Goeva, Michael-John Dolan, Judy Luu, Eric Garcia, Rebecca Boiarsky, Rajat M Gupta, and Evan Macosko. Hidden: a machine learning method for detection of disease-relevant populations in case-control single-cell transcriptomics data. *Nature Communications*, 15(1):9468, 2024.

[Liu *et al.*, 2022] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[Lotfollahi *et al.*, 2023] 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.

[Macek *et al.*, 2018] Pavel Macek, Matthew J Cliff, Kevin J Embrey, Geoffrey A Holdgate, J Willem M Nissink, Stanislava Panova, Jonathan P Waltho, and Rick A Davies. Myc phosphorylation in its basic helix–loop–helix region destabilizes transient α-helical structures, disrupting max and dna binding. *Journal of Biological Chemistry*, 293(24):9301–9310, 2018.

[Monga *et al.*, 2021] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

[Pakos-Zebrucka *et al.*, 2016] Karolina Pakos-Zebrucka, Izabela Koryga, Katarzyna Mnich, Mila Ljujic, Afshin Samali, and Adrienne M Gorman. The integrated stress response. *EMBO reports*, 17(10):1374–1395, 2016.

[Papalexi *et al.*, 2021] Efthymia Papalexi, Eleni P Mimitou, Andrew W Butler, Samantha Foster, Bernadette Bracken, William M Mauck III, Hans-Hermann Wessels, Yuhan Hao, Bertrand Z Yeung, Peter Smibert, et al. Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens. *Nature genetics*, 53(3):322–331, 2021.

[Replogle *et al.*, 2022] 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.

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

[Roohani *et al.*, 2025] Yusuf H Roohani, Tony J Hua, Po-Yuan Tung, Lexi R Bounds, Feiqiao B Yu, Alexander Dobin, Noam Teyssier, Abhinav Adduri, Alden Woodrow, Brian S Plosky, et al. Virtual cell challenge: Toward a turing test for the virtual cell. *Cell*, 188(13):3370–3374, 2025.

[Saboksayr and Mateos, 2021] Seyed Saman Saboksayr and Gonzalo Mateos. Accelerated graph learning from smooth signals. *IEEE Signal Processing Letters*, 28:2192–2196, 2021.

[Tancik *et al.*, 2020] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

[Tian *et al.*, 2021] Ruilin Tian, Anthony Abarientos, Jason Hong, Sayed Hadi Hashemi, Rui Yan, Nina Dräger, Kun Leng, Mike A Nalls, Andrew B Singleton, Ke Xu, et al. Genome-wide crispri/a screens in human neurons link lysosomal failure to ferroptosis. *Nature neuroscience*, 24(7):1020–1034, 2021.

[Van de Sande *et al.*, 2020] Bram Van de Sande, Christopher Flerin, Kristofer Davie, Maxime De Waegeneer, Gert Hulselmans, Sara Aibar, Ruth Seurinck, Wouter Saelens, Robrecht Cannoodt, Quentin Rouchon, et al. A scalable scenic workflow for single-cell gene regulatory network analysis. *Nature protocols*, 15(7):2247–2276, 2020.

[Wan *et al.*, 2011] Guohui Wan, Rohit Mathur, Xiaoxiao Hu, Xinna Zhang, and Xiongbin Lu. mirna response to dna damage. *Trends in biochemical sciences*, 36(9):478–484, 2011.

[Wei *et al.*, 2025] Zhiting Wei, Yiheng Wang, Yicheng Gao, Shuguang Wang, Ping Li, Duanmiao Si, Yuli Gao, Siqi Wu, Danlu Li, Kejing Dong, et al. Benchmarking algorithms for generalizable single-cell perturbation response prediction. *Nature Methods*, pages 1–14, 2025.

[Wessels *et al.*, 2023] Hans-Hermann Wessels, Alejandro Méndez-Mancilla, Yuhan Hao, Efthymia Papalexi, William M Mauck III, Lu Lu, John A Morris, Eleni P Mimitou, Peter Smibert, Neville E Sanjana, et al. Efficient combinatorial targeting of rna transcripts in single cells with cas13 rna perturb-seq. *Nature methods*, 20(1):86–94, 2023.

[Yu *et al.*, 2020] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in neural information processing systems*, 33:9422–9434, 2020.

[Zhu and Li, 2025] Ouyang Zhu and Jun Li. Scouter predicts transcriptional responses to genetic perturbations with large language model embeddings. *Nature Computational Science*, pages 1–8, 2025.

[Zou *et al.*, 2018] Jiahuan Zou, Mengwei Zhuang, Xiaopeng Yu, Na Li, Rudi Mao, Zhida Wang, Jianing Wang, Xiaoyan Wang, Huaiyu Zhou, Lining Zhang, et al. Myc inhibition increases pd-l1 expression induced by ifn-γ in hepatocellular carcinoma cells. *Molecular immunology*, 101:203–209, 2018.