INTERVENTIONAL WORLD MODELS WITH CONDITIONAL GENE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting cellular responses to unseen intervention combinations requires modeling how interventions transform cellular states and interact. We introduce the Interventional Graph World Model (IGWM), which represents interventions as reusable state transition operators rather than endpoint labels. IGWM learns a state and intervention conditioned graph editor that selects local gene interactions from a candidate graph. Message passing on the resulting sparse graph defines a conditional velocity field for response prediction. Flow matching enables learning from unpaired endpoint single cell data without longitudinal trajectories. This design supports direct prediction and sequential application of interventions. Across two datasets, IGWM-sparse improves all reported mean metrics over prior baselines for unseen combinations on Norman19 and held-out cells under observed interventions on VCC-H1. IGWM provides a structured framework for modeling context dependent intervention dynamics.

1 INTRODUCTION

Predicting cellular responses can reduce experimental search in functional genomics and therapeutic discovery. However, intervention combinations grow rapidly and can produce nonadditive effects through shared pathways. Recent benchmarks also show that apparent gains may reflect shared treatment effects rather than perturbation specific biology (Ahlmann-Eltze et al., 2025; Viñas Torné et al., 2026; Wei et al., 2026). The challenge is to learn how interventions transform cellular state and interact.

A world model learns action conditioned state transitions and supports rollout (Ha & Schmidhuber, 2018; Hafner et al., 2025; Bruce et al., 2024). We adopt this view for cells, learning an intervention operator $T_{\boldsymbol{a}}$ that maps a cellular state $\boldsymbol{x}$ to its response under action $\boldsymbol{a}$. This enables direct combination prediction and comparison with sequential application, making composition a property of learned dynamics. Figure 1 illustrates this motivation: state and intervention dependent local interactions provide a transition mechanism for predicting unseen combinations and reusing learned transitions for further interventions.

Latent perturbation models leave gene interactions implicit (Lotfollahi et al., 2023; Piran et al., 2024), while graph methods commonly use fixed networks (Roohani et al., 2024). Task decomposition, language model embeddings, and pretrained cell representations further improve prediction (Gao et al., 2024; Zhu & Li, 2026; Zeng et al., 2025). Our focus is instead the active interactions governing each state and intervention conditioned transition.

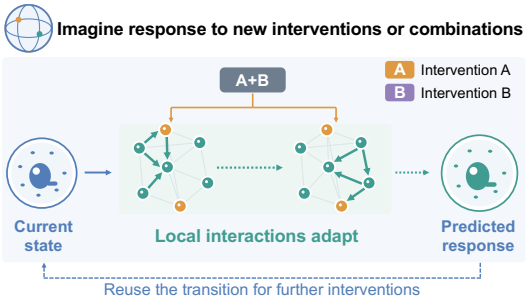

Figure 1: A cellular world model predicts unseen intervention combinations by learning state and intervention dependent local dynamics.

We model these interactions through an *effective mechanism graph*, without claiming recovery of the causal network. Two hypotheses guide its design: *mechanistic locality*, which limits active interactions, and *intervention dependence*, which makes them depend on state and

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

action. The resulting sparse graph $\mathcal{G}_{\boldsymbol{x},\boldsymbol{a}} \subseteq \mathcal{G}_0$ is selected from a candidate graph $\mathcal{G}_0$ constructed without test responses and forms part of the transition law.

We introduce the Interventional Graph World Model (IGWM). A state and intervention conditioned editor selects top-K candidate edges with straight-through gradients. Graph messages parameterize a vector field whose integration defines $T_{\boldsymbol{a}}$. Conditional flow matching learns the field from unpaired control and treated populations (Lipman et al., 2023), enabling direct prediction and sequential application.

We evaluate unseen double perturbations on Norman19 with component singles observed in training (Norman et al., 2019), and held-out cells under observed interventions on VCC-H1. Against the strongest baseline for each metric, IGWM-sparse reduces Norman19 MSE by 8.2% and improves DEG F1 by 27.1% and accuracy by 3.3%. It also improves all five mean metrics over CPA and GEARS on VCC-H1, a within-condition task rather than unseen intervention generalization.

Our contributions are threefold:

- We formulate combinatorial perturbation prediction as structured transition learning, where interventions act as composable operators and effective gene interactions belong to the dynamics.
- We introduce IGWM, which learns a conditional sparse mechanism graph and a flow matching transition operator from endpoint single cell measurements.
- We compare IGWM with prior baselines on unseen combinations and within-condition prediction across two datasets, and examine conditional weighting and sparse selection through ablations.

2 Related Work

2.1 Single Cell Perturbation Prediction

CPA and biolord separate basal state from perturbation factors, while chemCPA and PRnet use chemical information (Lotfollahi et al., 2023; Piran et al., 2024; Hetzel et al., 2022; Qi et al., 2024). PerturbNet uses structured intervention descriptions, STAMP separates response gene, direction, and magnitude prediction, and Scouter uses language model embeddings (Yu et al., 2025; Gao et al., 2024; Zhu & Li, 2026). CellOT learns unpaired population transport and GEARS propagates through a gene knowledge graph (Bunne et al., 2023; Roohani et al., 2024). Benchmarks emphasize sensitivity to splits, baselines, and metrics (Ahlmann-Eltze et al., 2025; Viñas Torné et al., 2026; Wei et al., 2026). IGWM makes the active interaction graph part of the transition, focusing on combinations of observed single interventions.

2.2 Biological Structure and Context Dependent Graphs

Geneformer, scGPT, scFoundation, GeneCompass, CellFM, and RegFormer learn gene or cell representations from large transcriptomic collections (Theodoris et al., 2023; Cui et al., 2024; Hao et al., 2024; Yang et al., 2024; Zeng et al., 2025; Hu et al., 2026). SATURN, SCimilarity, and UCE support transfer across datasets or species (Rosen et al., 2024; Heimberg et al., 2025; Rosen et al., 2026). GRouNdGAN simulates regulatory interventions, while SCARlink and CREME study regulatory relationships (Zinati et al., 2024; Mitra et al., 2024; Toneyan & Koo, 2024). IGWM instead separates the control-derived candidate graph from a sparse effective graph selected for each state and intervention.

2.3 World Models and Continuous Cellular Dynamics

World models learn reusable action conditioned transitions (Ha & Schmidhuber, 2018; Hafner et al., 2025; Bruce et al., 2024), while structured world models and graph simulators use entities and local relations for dynamics (Kipf et al., 2020; Pfaff et al., 2021). MIOFlow models cellular dynamics with transport geometry, CellOT learns population transport, and flow matching trains vector fields without simulating trajectories (Huguet et al., 2022; Bunne et al., 2023; Lipman et al., 2023). IGWM combines an intervention conditioned field with an editable gene graph, yielding inspectable transitions that can be applied sequentially.

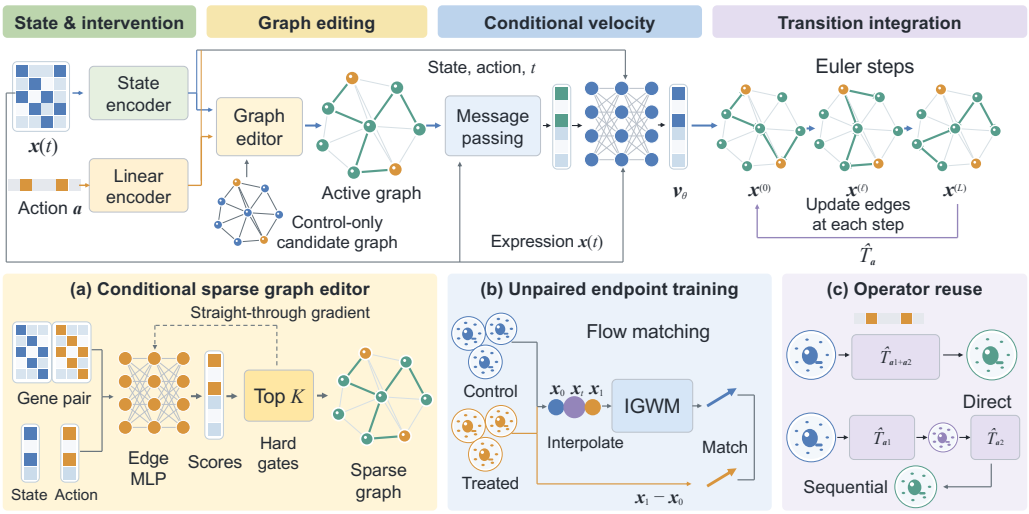

Figure 2: Overview of IGWM. State and intervention features guide sparse graph selection from a control-derived candidate graph. Message passing parameterizes a conditional velocity field, integrated with updated graphs to predict cellular responses. (a) The graph editor selects top-K edges with straight-through gradients. (b) Flow matching learns from independently sampled control and treated endpoints. (c) Transition operators support direct combination prediction and sequential application, without assuming equal outcomes.

3 Method

3.1 Problem Formulation

Let G denote the number of genes retained after preprocessing. A cellular state is a normalized expression vector $\boldsymbol{x} \in \mathbb{R}^G$, and an intervention is a multi hot vector $\boldsymbol{a} \in \{0,1\}^G$. The entry $a_i = 1$ means that gene i is targeted, while $a_i = 0$ means that it is not targeted. A combined intervention is represented by more than one nonzero entry. We observe a control population sampled from a distribution $p_0(\boldsymbol{x})$ and, for every training intervention $\boldsymbol{a}$, a treated population sampled from $p_{\boldsymbol{a}}(\boldsymbol{x})$. The measurements are unpaired because sequencing destroys each cell, so no treated cell has an observed control ancestor.

Our goal is to learn a family of transition operators $T_{\boldsymbol{a}} : \mathbb{R}^G \to \mathbb{R}^G$. Given a starting state $\boldsymbol{x}_0 \sim p_0$, the output $T_{\boldsymbol{a}}(\boldsymbol{x}_0)$ should follow the treated distribution $p_{\boldsymbol{a}}$. The same operator should remain meaningful when evaluated on a new starting state or combined with another learned operator. The main benchmark tests compositional generalization by withholding selected double gene interventions while retaining their component single gene interventions during training.

We model each operator through a continuous state path $\boldsymbol{x}(t)$ indexed by interpolation time $t \in [0,1]$. The path starts at $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and follows a conditional vector field $\boldsymbol{v}_\theta$, where θ denotes all learned model parameters. The transition is defined by

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{v}_\theta\big(\boldsymbol{x}(t), t, \boldsymbol{a}, \mathcal{G}_{\boldsymbol{x}(t),\boldsymbol{a}}\big), \qquad T_{\boldsymbol{a}}(\boldsymbol{x}_0) = \boldsymbol{x}_0 + \int_0^1 \boldsymbol{v}_\theta\big(\boldsymbol{x}(t), t, \boldsymbol{a}, \mathcal{G}_{\boldsymbol{x}(t),\boldsymbol{a}}\big)\, dt. \tag{1}$$

Here, $\mathcal{G}_{\boldsymbol{x}(t),\boldsymbol{a}}$ is an effective gene graph selected for the current state $\boldsymbol{x}(t)$ and intervention $\boldsymbol{a}$. This graph is part of the transition law rather than a fixed representation computed before the transition.

Fig. 2 provides an overview of IGWM, including conditional graph editing, flow matching, and intervention composition. At each integration step, the model selects active gene interactions based on the current state and intervention, then uses these interactions to predict the velocity of state change.

3.2 CANDIDATE GENE GRAPH

We construct a directed candidate graph $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E}_0)$ from control cells only, with one node per gene. For each source gene, we retain edges to its $r = 12$ most positively correlated available neighbors. No treated samples enter this construction. The graph encodes possible interactions, not an identified causal network. Under mechanistic locality and intervention dependence, the editor selects a small subset that changes with the current state and action. Appendix A.1 gives the construction details.

3.3 STATE AND INTERVENTION CONDITIONED GRAPH EDITING

Let H be the hidden dimension. A state encoder maps $\boldsymbol{x}(t)$ to $\boldsymbol{h}_t = f_x(\boldsymbol{x}(t)) \in \mathbb{R}^H$. A bias-free linear action encoder gives $\boldsymbol{u} = \boldsymbol{W}_a \boldsymbol{a} \in \mathbb{R}^H$, where $\boldsymbol{W}_a \in \mathbb{R}^{H \times G}$ is learned. Thus, a combined action sums the contributions of its target genes rather than using a separate combination identifier.

Each gene i has a learned embedding $\boldsymbol{q}_i \in \mathbb{R}^H$. We summarize the encoded state with $c_t = H^{-1} \sum_{d=1}^{H} h_{t,d}$, where $h_{t,d}$ is dimension d of $\boldsymbol{h}_t$. Let $\mathbf{1}_H \in \mathbb{R}^H$ denote a vector of ones. For every candidate edge $e = (i, j) \in \mathcal{E}_0$, an edge network s_ϕ with parameters ϕ computes

$$\ell_e = s_\phi\big(\boldsymbol{q}_i \parallel \boldsymbol{q}_j \parallel (\boldsymbol{u} + c_t \mathbf{1}_H)\big), \qquad p_e = \mathrm{sigmoid}(\ell_e), \tag{2}$$

Here, $\parallel$ denotes concatenation, ℓ_e is a scalar edge logit, and $p_e \in (0, 1)$ is its soft gate. The scores depend on both state and intervention. Network details are in Appendix A.2.

Let $E = |\mathcal{E}_0|$ be the number of candidate edges and let $K < E$ be the sparse edge budget. To enforce locality, we retain the K highest scoring edges for each state and action. The score margin determines when this selected graph is robust to a change in state.

Proposition 1 (Local stability of graph editing)**.** *Fix an action $\boldsymbol{a}$ and a state $\boldsymbol{x}$. Suppose the gap between the Kth and $(K+1)$th largest edge scores at $\boldsymbol{x}$ is $\gamma > 0$, and each score is L_s-Lipschitz near $\boldsymbol{x}$. The selected edge set is unchanged at any state $\boldsymbol{x}'$ in that neighborhood with $\|\boldsymbol{x}' - \boldsymbol{x}\|_2 < \gamma/(2L_s)$, with no distance restriction within the neighborhood when $L_s = 0$.*

The proof is in Appendix B.1. In sparse mode, $\widehat{g}_e$ equals one for the selected edges and zero otherwise. To train this hard selection, we use the gate

$$g_e = \widehat{g}_e + p_e - \mathrm{stopgrad}(p_e), \tag{3}$$

where $\mathrm{stopgrad}$ returns its argument during the forward pass and blocks its gradient during backpropagation. Eq. (3) uses exactly K active edges in the forward pass while passing gradients through the soft scores. We compare sparse mode with static mode, which assigns $g_e = 1$ to every candidate edge, and dense mode, which uses $g_e = p_e$ without top K selection. These modes test conditional weighting and sparse selection.

3.4 GRAPH CONDITIONED VECTOR FIELD

The graph transmits the current expression value of each source gene to its destination. Consider a minibatch of B samples and let $g_{ij}^{(b)}$ be the gate for edge (i, j) in sample $b \in \{1, \ldots, B\}$. We define the batch mean gate as $\overline{g}_{ij} = B^{-1} \sum_{b=1}^{B} g_{ij}^{(b)}$ and the normalized incoming mass of gene j as $d_j = \max\big(1, \sum_{i:(i,j)\in\mathcal{E}_0} \overline{g}_{ij}\big)$. The propagated feature for gene j in sample b is

$$m_{t,j}^{(b)} = \frac{1}{d_j} \sum_{i:(i,j)\in\mathcal{E}_0} g_{ij}^{(b)} x_i^{(b)}(t), \tag{4}$$

where $x_i^{(b)}(t)$ is the current expression of source gene i. The lower bound in d_j prevents amplification when the incoming gate mass is small.

The velocity network receives the current state, propagated graph features, encoded state, encoded action, and interpolation time. Its output is

$$\boldsymbol{v}_\theta(\boldsymbol{x}(t), t, \boldsymbol{a}, \mathcal{G}_{\boldsymbol{x}(t),\boldsymbol{a}}) = F_\theta\big(\boldsymbol{x}(t) \parallel \boldsymbol{m}_t \parallel \boldsymbol{h}_t \parallel \boldsymbol{u} \parallel t\big) \in \mathbb{R}^G, \tag{5}$$

Here, F_θ predicts one velocity per gene and $\boldsymbol{m}_t = (m_{t,1}, \ldots, m_{t,G})$ is the propagated expression vector. The graph is updated along the integration path. Write $\boldsymbol{V}_{\boldsymbol{a}}(t, \boldsymbol{x}) = \boldsymbol{v}_\theta(\boldsymbol{x}, t, \boldsymbol{a}, \mathcal{G}_{\boldsymbol{x},\boldsymbol{a}})$ for the complete single-state field, including graph editing.

Proposition 2 (Local stability of the transition)**.** *Fix an action $\boldsymbol{a}$. Suppose the exact trajectories from $\boldsymbol{x}$ and $\boldsymbol{y}$ stay in a region away from TopK switches where $\boldsymbol{V}_{\boldsymbol{a}}(t, \cdot)$ is L_x-Lipschitz. Then $\|T_{\boldsymbol{a}}(\boldsymbol{x}) - T_{\boldsymbol{a}}(\boldsymbol{y})\|_2 \le e^{L_x}\|\boldsymbol{x} - \boldsymbol{y}\|_2$.*

Appendix B.2 gives the proof. The claim is local because a change in selected edges may break the Lipschitz condition.

3.5 Learning from Unpaired Endpoint Populations

We use conditional flow matching (Lipman et al., 2023), independently sampling a control cell $\boldsymbol{x}_0 \sim p_0$ and a treated cell $\boldsymbol{x}_1 \sim p_{\boldsymbol{a}}$. For uniform interpolation time $t \sim \mathcal{U}(0, 1)$ and standard Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_G)$, where $\boldsymbol{I}_G$ is the G-dimensional identity matrix, we form

$$\boldsymbol{x}_t = (1 - t)\boldsymbol{x}_0 + t\boldsymbol{x}_1 + \sigma\boldsymbol{\epsilon}, \tag{6}$$

where σ is the noise scale. The target velocity is $\boldsymbol{x}_1 - \boldsymbol{x}_0$, so training requires endpoint populations but no observed cell pairs or trajectories. The following result identifies the population target.

Proposition 3 (Population flow target)**.** *Assume $\mathbb{E}\|\boldsymbol{x}_1 - \boldsymbol{x}_0\|_2^2 < \infty$. Over all square-integrable measurable per-sample fields, the minimizer of the unregularized population squared velocity loss is $\boldsymbol{v}^*(\boldsymbol{x}, t, \boldsymbol{a}) = \mathbb{E}[\boldsymbol{x}_1 - \boldsymbol{x}_0 \mid \boldsymbol{x}_t = \boldsymbol{x}, t, \boldsymbol{a}]$ almost everywhere.*

Appendix B.3 gives the proof. This conditional expectation motivates the flow matching term in the objective below.

Let $\mathbb{E}$ be expectation over the sampled intervention, cells, time, and noise. The training objective is

$$\mathcal{L}_{\mathrm{IGWM}}(\theta) = \mathbb{E}\left[\frac{1}{G}\left\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t, \boldsymbol{a}, \mathcal{G}_{\boldsymbol{x}_t,\boldsymbol{a}}) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\right\|_2^2\right] + \lambda\mathbb{E}\left[\frac{1}{E}\sum_{e\in\mathcal{E}_0} g_e\right], \tag{7}$$

Here, $\|\cdot\|_2$ is the Euclidean norm and λ controls gate regularization. With exactly K active edges, this term has a constant forward value, but its straight-through gradient acts on the soft gates. The population result does not guarantee convergence of finite-network training. Optimization settings are in Appendix A.4.

3.6 Transition Inference and Intervention Composition

At inference, we approximate Eq. (1) with L explicit Euler steps. Let $\boldsymbol{x}^{(0)} = \boldsymbol{x}_0$ and let $\Delta t = 1/L$ be the step size. For step $\ell \in \{0, \ldots, L-1\}$, we update

$$\boldsymbol{x}^{(\ell+1)} = \boldsymbol{x}^{(\ell)} + \Delta t\, \boldsymbol{v}_\theta\big(\boldsymbol{x}^{(\ell)}, \ell\Delta t, \boldsymbol{a}, \mathcal{G}_{\boldsymbol{x}^{(\ell)},\boldsymbol{a}}\big), \qquad \widehat{T}_{\boldsymbol{a}}(\boldsymbol{x}_0) = \boldsymbol{x}^{(L)}, \tag{8}$$

where ℓ is the integration step index and $\widehat{T}_{\boldsymbol{a}}$ is the numerical approximation to $T_{\boldsymbol{a}}$. Euler integration makes the transition computable with a controlled discretization error under local regularity. Under a Lipschitz field and a bounded velocity derivative along the trajectory, Euler endpoint error is $O(L^{-1})$ (Proposition 4, Appendix B.4). This bounds numerical error for the learned field, not error relative to biological dynamics. We use $L = 16$ steps and start from the shared control mean for each test condition.

For two actions $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, sequential prediction is $\widehat{T}_{\boldsymbol{a}_2}(\widehat{T}_{\boldsymbol{a}_1}(\boldsymbol{x}_0))$, while direct prediction uses $\widehat{T}_{\boldsymbol{a}_1+\boldsymbol{a}_2}(\boldsymbol{x}_0)$. If the individual approximation errors are bounded by $\varepsilon_1, \varepsilon_2$ and the second reference transition has Lipschitz constant C_2, sequential error is bounded by $\varepsilon_2 + C_2\varepsilon_1$ (Proposition 5). Thus, rollout may amplify error and need not equal a simultaneous intervention. Appendix B.5 gives the formal assumptions and proof. Main results use direct prediction trained only with Eq. (7). Sequential predictions are evaluated separately without sequential supervision.

4 EXPERIMENTS

We evaluate IGWM on two tasks: prediction of unseen intervention combinations on Norman19 and held-out cells under observed interventions on VCC-H1. We compare against CPA and GEARS and examine conditional edge weighting and sparse selection through graph variants. Further analyses on Norman19 assess response strength, data efficiency, graph sparsity, sequential prediction, and intervention-conditioned graph editing.

4.1 EXPERIMENTAL SETUP

Datasets and tasks. We use Norman19 (Norman et al., 2019) and VCC-H1 (Roohani et al., 2025). Norman19 tests 25 unseen double perturbations whose component singles appear in training, with other eligible conditions available for fitting. Qualitative response and gene-level analyses use controls and single perturbations only for training. VCC-H1 tests held-out cells under 110 single-gene interventions observed in training, not unseen targets or combinations. Within each dataset, all methods share expression, gene order, controls, and the fixed split. Appendix A.3 details preprocessing and split construction.

Baselines. We use public CPA (Lotfollahi et al., 2023) and GEARS (Roohani et al., 2024) implementations. Three IGWM variants share architecture and training budgets: static uses unit edge weights, dense learns conditional weights, and sparse retains the top K edges. They test weighting and sparsity, not graph propagation against a graph-free model.

Metrics. MSE measures full-vector expression error. DEG F1 compares predicted and observed top-20 gene sets ranked by absolute change from control. Accuracy measures sign agreement on the observed top-20 genes. Pearson and Delta Pearson measure correlation of expression and expression changes, respectively. Lower MSE and higher other metrics are better. We average over test conditions and report means and sample standard deviations across five runs.

Implementation details. We use five training seeds, a fixed data split, and at most 100 epochs for all methods. The full model selects $K = 512$ edges and uses 16 Euler steps. Architecture, optimization, and baseline settings are in Appendix A.4.

4.2 PREDICTION OF UNSEEN INTERVENTION COMBINATIONS

Table 1 shows that IGWM-sparse obtains the best mean among the displayed methods on all five metrics on Norman19. Relative to CPA, which is the strongest baseline on MSE and DEG F1, IGWM-sparse reduces MSE by 8.2 percent and improves DEG F1 by 0.114. Relative to GEARS, which is the strongest baseline on direction accuracy, it improves direction accuracy by 0.0316. It also raises Delta Pearson from the best baseline value of 0.7509 to 0.8107. These results indicate that the gain is not limited to average expression error. IGWM more accurately identifies the dominant response genes and their directions. The graph variants test conditional weighting and sparse selection. On Norman19, dense weighting raises DEG F1 from 0.3676 to 0.4824 relative to IGWM-static. Sparse selection further lowers MSE by 36.6 percent relative to IGWM-dense and raises DEG F1 by 0.0516 and direction accuracy by 0.0040. These comparisons

Table 1: Results on two datasets with distinct prediction tasks: Norman19 evaluates 25 unseen double perturbations whose singles appear in training, and VCC-H1 evaluates held-out cells under 110 training interventions. Values are means and sample standard deviations over five random seeds (×100); bold denotes the best mean per dataset.

Method	MSE ↓	DEG F1 ↑	Accuracy ↑	Pearson ↑	Delta Pearson ↑
Norman19: unseen intervention combinations					
CPA	$0.79_{\pm 0.05}$	$42.00_{\pm 1.68}$	$91.20_{\pm 1.64}$	$99.18_{\pm 0.05}$	$75.09_{\pm 0.71}$
GEARS	$0.88_{\pm 0.03}$	$39.16_{\pm 0.93}$	$94.76_{\pm 0.79}$	$99.17_{\pm 0.03}$	$74.68_{\pm 0.47}$
IGWM-static	$2.83_{\pm 0.47}$	$36.76_{\pm 2.19}$	$95.20_{\pm 0.99}$	$97.73_{\pm 0.43}$	$76.22_{\pm 2.35}$
IGWM-dense	$1.15_{\pm 0.41}$	$48.24_{\pm 6.04}$	$97.52_{\pm 0.97}$	$98.97_{\pm 0.33}$	$78.75_{\pm 4.54}$
IGWM-sparse	$\mathbf{0.73}_{\pm 0.11}$	$\mathbf{53.40}_{\pm 2.25}$	$\mathbf{97.92}_{\pm 0.96}$	$\mathbf{99.29}_{\pm 0.09}$	$\mathbf{81.07}_{\pm 3.57}$
VCC-H1: held-out cells under observed interventions					
CPA	$0.91_{\pm 0.00}$	$9.24_{\pm 0.51}$	$67.74_{\pm 0.38}$	$96.41_{\pm 0.00}$	$16.29_{\pm 0.04}$
GEARS	$0.84_{\pm 0.01}$	$14.71_{\pm 1.36}$	$69.58_{\pm 1.23}$	$96.67_{\pm 0.05}$	$17.60_{\pm 1.60}$
IGWM-static	$0.88_{\pm 0.06}$	$11.63_{\pm 2.45}$	$70.53_{\pm 3.64}$	$96.52_{\pm 0.23}$	$18.67_{\pm 4.25}$
IGWM-dense	$0.78_{\pm 0.11}$	$18.91_{\pm 5.85}$	$\mathbf{74.99}_{\pm 2.91}$	$96.91_{\pm 0.43}$	$24.63_{\pm 5.44}$
IGWM-sparse	$\mathbf{0.74}_{\pm 0.06}$	$\mathbf{27.20}_{\pm 2.53}$	$74.94_{\pm 1.25}$	$\mathbf{97.06}_{\pm 0.22}$	$\mathbf{27.33}_{\pm 2.09}$

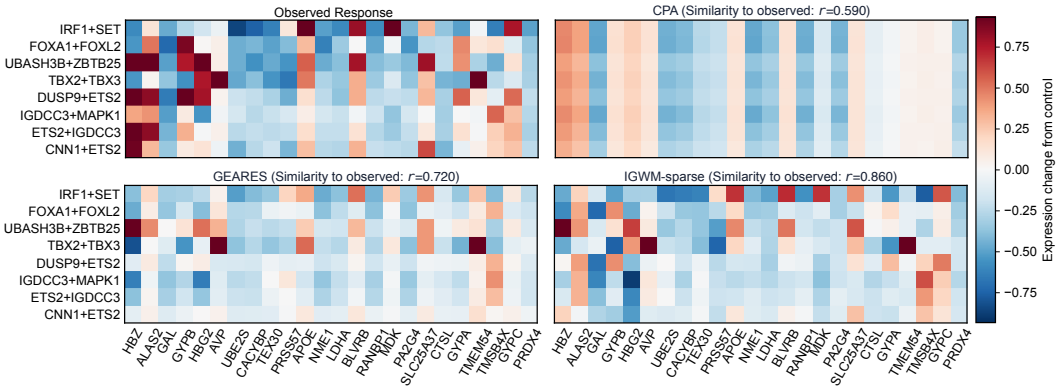

Figure 3: Observed and predicted expression changes for eight held-out combinations from the strict composition benchmark. Columns show genes with the largest observed responses. Similarities are Pearson correlations over displayed entries. IGWM outperforms GEARS on these examples across all four metrics. Table 1 reports the complete comparison.

support conditional weighting and sparse selection within the graph-based architecture, rather than establishing an advantage over graph-free models.

Fig. 3 provides a closer view of representative combinations in the stricter single-only training setting. IGWM reaches a displayed-matrix correlation of 0.860, compared with 0.720 for GEARS and 0.590 for CPA. It reproduces several strong positive and negative expression blocks that are weakened or missed by the baselines. Since these eight conditions were selected using condition-level performance, this figure serves as a qualitative case analysis rather than an unbiased estimate of average performance.

4.3 Prediction under Observed Interventions

Table 1 reports held-out cell prediction under 110 VCC-H1 interventions observed in training. IGWM-sparse improves all five mean metrics over CPA and GEARS. Relative to GEARS, it reduces MSE by 11.1%, improves DEG F1 by 84.9%, and raises direction accuracy by 5.35 percentage points. Sparse selection also improves MSE, DEG F1, Pearson, and Delta Pearson over dense weighting, although dense weighting has slightly higher direction accuracy. These results support within-condition prediction on a second dataset, not generalization to unseen interventions. The comparison is descriptive and does not establish statistical significance.

4.4 Performance across Response Strengths

We examine how prediction quality varies with response strength. Genes are ranked by their mean absolute observed change from control across the strict composition test conditions and divided into five equal groups. We report gene-level MAE across conditions. Figure 4a shows that error increases with response strength for all methods. IGWM outperforms GEARS from Q1 to Q4 and is comparable to CPA from Q1 to Q3. It achieves the lowest Q4 MAE at 0.0789, compared with 0.0805 for CPA and 0.0831 for GEARS. In Q5, GEARS performs best with 0.1259, while CPA and IGWM obtain 0.1374 and 0.1378. Thus, IGWM performs well on weak and moderate responses, while strong expression changes remain challenging.

4.5 Data Efficiency

To measure data efficiency, we train IGWM-sparse with 25, 50, 75, and 100 percent of the available training observations while keeping the validation and test conditions fixed. Subsampling is performed within training conditions so that the experiment changes observation coverage without redefining the prediction task. Every fraction is evaluated with the same five seeds used in the main experiment. Fig. 4b shows steady improvement as more observations become available. Increasing the training fraction from 25 to 75 percent reduces MSE from 0.00979 to 0.00741, a relative reduction of 24.3 percent, while DEG F1 rises from 0.4880 to 0.5548 and direction accuracy rises from 0.9572 to 0.9760. Moving from 75 to 100 percent yields only a further 1.6 percent MSE reduction.

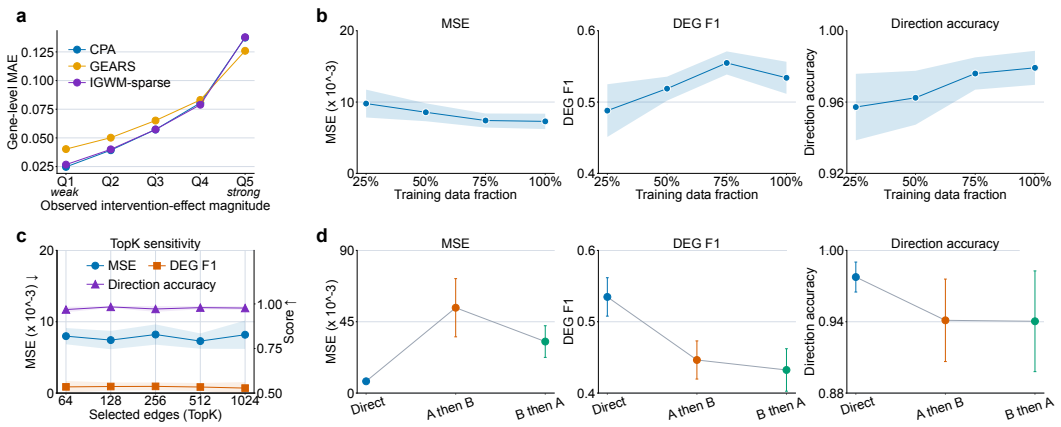

Figure 4: Robustness and compositional behavior of IGWM. (a) Gene-level prediction error across intervention response strengths. (b) Performance with different fractions of training data. (c) Sensitivity to the number of selected graph edges. (d) Direct and sequential intervention prediction, including consistency across intervention orders. Lines and points show means, while shaded regions and error bars show sample standard deviations over five random runs.

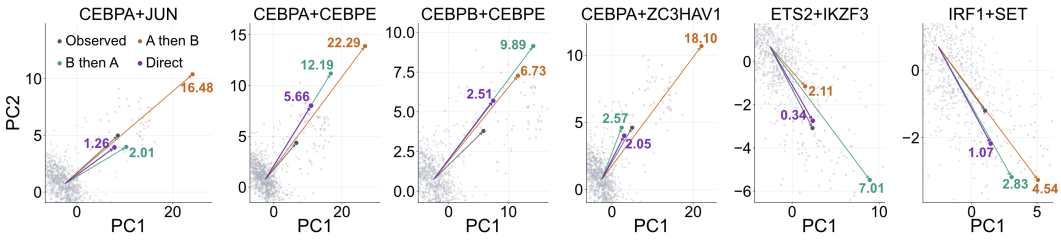

Figure 5: State transitions for six strong-response combinations in the first two principal components. Gray arrows end at observed responses. Colored arrows show direct prediction and the two sequential orders. Endpoint labels report Euclidean distance to the observed endpoint, where lower values are better. Each panel uses its own axis range.

DEG F1 peaks at 75 percent, while direction accuracy and MSE improve slightly at full data. These results suggest that most of the benefit is obtained by 75 percent coverage and that IGWM remains usable when training observations are limited.

4.6 Sensitivity to Graph Sparsity

The sparse editor retains a fixed number of candidate edges for each state and intervention. We vary this budget over $K \in \{64, 128, 256, 512, 1024\}$ while keeping the candidate graph, architecture, training schedule, and evaluation split unchanged. This experiment tests whether the reported performance depends on one narrowly tuned sparsity level. Performance remains stable across the tested budgets in Fig. 4c. Mean MSE ranges from 0.00729 to 0.00821, DEG F1 ranges from 0.5280 to 0.5376, and direction accuracy ranges from 0.9684 to 0.9832. No single value dominates every metric. $K = 512$ gives the lowest MSE, $K = 256$ gives the highest DEG F1, and $K = 128$ gives the highest direction accuracy. We use $K = 512$ in the main model because it gives the strongest complete-profile accuracy while retaining a small active graph. The limited variation across settings indicates that the main conclusion is not sensitive to an exact TopK choice.

4.7 Sequential Intervention Prediction

A transition model can be applied directly to a combined action or repeatedly to an ordered sequence. For each held-out pair (A, B), we compare direct prediction with the combined action $A + B$, the rollout that applies A and then B, and the reverse rollout that applies B and then A. All three

modes use the same trained IGWM-sparse checkpoint and 16 Euler steps per transition. The model receives no sequential supervision during training. Direct prediction is substantially more accurate than sequential rollout. As shown in Fig. 4d, direct prediction obtains an MSE of 0.00743 ± 0.00140, while A then B and B then A obtain 0.05384 ± 0.01836 and 0.03250 ± 0.00999. Direct DEG F1 is 0.5348, compared with 0.4464 and 0.4324 for the two rollout orders. Direction accuracy decreases from 0.9776 for direct prediction to about 0.941 for both sequential modes. The MSE between the two rollout orders is 0.04542, and B then A is closer to direct prediction than A then B, with consistency errors of 0.01723 and 0.04250 respectively.

The PCA projections in Fig. 5 make the same pattern visible at the state level. Direct prediction has the smallest endpoint distance in all six displayed combinations. Some sequential paths preserve the observed direction, but their response magnitude is often too large, particularly for A then B. The transition operator can therefore be executed repeatedly, but errors accumulate and the two orders can diverge. These results do not yet support a strong claim of accurate sequential composition without additional consistency training or supervision.

4.8 Analysis of Intervention-Conditioned Graph Editing

We finally examine whether sparse edge selection changes with the intervention and the state. For each test combination, we record the 512 selected candidate edges at the control state and after the predicted transition. State Jaccard compares the selected sets before and after the transition for the same action. Action Jaccard compares sets produced by different actions at the same control state. We also measure the fraction of candidate edges selected at least once across the tested actions. Fig. 6 shows both broadly reused edges and edges whose selection frequency changes across interventions. Each effective graph contains exactly 512 edges, and the union of active edges covers 4.44 ± 3.25 percent of the candidate set. State Jaccard is 0.758 ± 0.328, while action Jaccard is 0.754 ± 0.343. Their similar means and large variation indicate mixed evidence. The editor is capable of producing condition-specific patterns, but the selected graphs also share a substantial core and are not uniformly distinct across actions or states. We therefore interpret this analysis as evidence of adaptive graph use rather than recovery of a unique causal network.

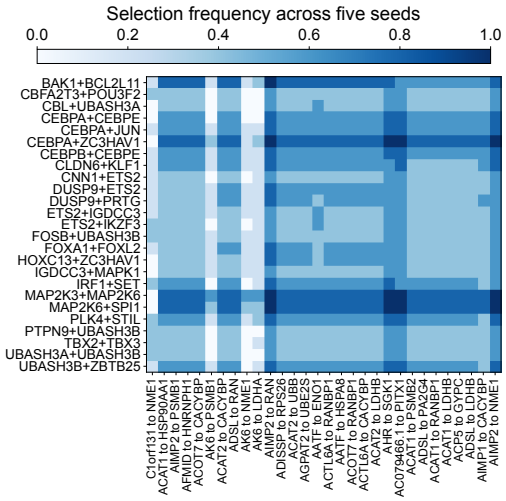

Figure 6: Selection frequency of the 30 most intervention-variable candidate edges. Rows are intervention combinations, columns are directed candidate edges, and darker cells indicate more frequent selection.

5 Conclusion and Future Work

In this work, we designed IGWM, a structured world model that represents interventions as reusable transition operators with sparse, context dependent gene interactions. Learning from unpaired endpoint measurements connects compositional prediction to explicit transition modeling. On Norman19 and VCC-H1, IGWM-sparse improves all five mean metrics over CPA and GEARS under unseen combination and observed-intervention cell holdout tasks, respectively. Comparisons among graph variants support conditional weighting and sparse selection, although dense weighting yields slightly higher VCC-H1 direction accuracy. Sequential rollout remains less accurate than direct prediction, and the selected graphs should not be interpreted as identified causal mechanisms. Future work should evaluate IGWM across cell types, tissues, and larger perturbation spaces, incorporate temporal and multimodal data, and integrate stronger biological priors. Uncertainty modeling could identify unreliable predictions, while active intervention selection could guide informative experiments.

6 AI USE STATEMENT

Generative AI tools were used to improve the readability and language of the manuscript. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final content.

7 ETHICS STATEMENT

This study uses released Norman19 and VCC-H1 single-cell perturbation data and does not collect new human participant data or personal information. We follow the original terms for data access and use. IGWM is developed for scientific research and has not been validated for clinical decision making.

8 REPRODUCIBILITY STATEMENT

We describe IGWM in Section 3 and the experimental protocol in Section 4.1. Appendix A provides graph construction, architecture, preprocessing, and training settings. Appendix B states and proves the theoretical properties.

REFERENCES

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. doi: 10.1038/s41592-025-02772-6. URL https://doi.org/10.1038/s41592-025-02772-6.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 4603–4623, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

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. doi: 10.1038/s41592-023-01969-x. URL https://doi.org/10.1038/s41592-023-01969-x.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21(8):1470–1480, 2024. doi: 10.1038/s41592-024-02201-0. URL https://doi.org/10.1038/s41592-024-02201-0.

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. doi: 10.1038/s43588-024-00698-1. URL https://doi.org/10.1038/s43588-024-00698-1.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2. URL https://doi.org/10.1038/s41586-025-08744-2.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 21(8):1481–1491, 2024. doi: 10.1038/s41592-024-02305-7. URL https://doi.org/10.1038/s41592-024-02305-7.

Graham Heimberg, Tony Kuo, Daryle J. DePianto, Omar Salem, Tobias Heigl, Nathaniel Diamant, Gabriele Scalia, Tommaso Biancalani, Shannon J. Turley, Jason R. Rock, Héctor Corrada Bravo, Josh Kaminker, Jason A. Vander Heiden, and Aviv Regev. A cell atlas foundation model for scalable search of similar human cells. *Nature*, 638(8052):1085–1094, 2025. doi: 10.1038/s41586-024-08411-y. URL https://doi.org/10.1038/s41586-024-08411-y.

Leon Hetzel, Simon Boehm, Niki Kilbertus, Stephan Günnemann, Mohammad Lotfollahi, and Fabian J. Theis. Predicting cellular responses to novel drug perturbations at a single-cell resolution. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26711–26722, 2022. doi: 10.52202/068431-1937. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/aa933b5abc1be30baece1d230ec575a7-Abstract.html.

Luni Hu, Hua Qin, Yilin Zhang, Yi Lu, Ping Qiu, Zhihan Guo, Lei Cao, Wenjian Jiang, Yixin Shen, Qianqian Chen, Yanbang Shang, Tianyi Xia, Ziqing Deng, Xun Xu, Hansheng Zhao, Shuangsang Fang, Yuxiang Li, and Yong Zhang. RegFormer: a single-cell foundation model powered by gene regulatory hierarchies. *Nature Communications*, 17(1):6432, 2026. doi: 10.1038/s41467-026-72198-x. URL https://doi.org/10.1038/s41467-026-72198-x.

Guillaume Huguet, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows for trajectory inference. In *Advances in Neural Information Processing Systems*, volume 35, pp. 29705–29718, 2022. doi: 10.52202/068431-2154. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/bfc03f077688d8885c0a9389d77616d0-Abstract-Conference.html.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1gax6VtDB.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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

Sneha Mitra, Rohan Malik, Wilfred Wong, Afsana Rahman, Alexander J. Hartemink, Yuri Pritykin, Kushal K. Dey, and Christina S. Leslie. Single-cell multi-ome regression models identify functional and disease-associated enhancers and enable chromatin potential analysis. *Nature Genetics*, 56(4):627–636, 2024. doi: 10.1038/s41588-024-01689-8. URL https://doi.org/10.1038/s41588-024-01689-8.

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

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Zoe Piran, Niv Cohen, Yedid Hoshen, and Mor Nitzan. Disentanglement of single-cell data with biolord. *Nature Biotechnology*, 42(11):1678–1683, 2024. doi: 10.1038/s41587-023-02079-x. URL https://doi.org/10.1038/s41587-023-02079-x.

Xiaoning Qi, Lianhe Zhao, Chenyu Tian, Yueyue Li, Zhen-Lin Chen, Peipei Huo, Runsheng Chen, Xiaodong Liu, Baoping Wan, Shengyong Yang, and Yi Zhao. Predicting transcriptional responses to novel chemical perturbations using deep generative model for drug discovery. *Nature Communications*, 15(1):9256, 2024. doi: 10.1038/s41467-024-53457-1. URL https://doi.org/10.1038/s41467-024-53457-1.

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

Yusuf H. Roohani, Tony J. Hua, Po-Yuan Tung, Lexi R. Bounds, Feiqiao B. Yu, Alexander Dobin, Noam Teyssier, Abhinav Adduri, Alden Woodrow, Brian S. Plosky, Reshma Mehta, Benjamin Hsu, Jeremy Sullivan, Chiara Ricci-Tam, Nianzhen Li, Julia Kazaks, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Hani Goodarzi, and Dave P. Burke. Virtual Cell Challenge: Toward a Turing test for the virtual cell. *Cell*, 188(13):3370–3374, 2025. doi: 10.1016/j.cell.2025.06.008. URL https://doi.org/10.1016/j.cell.2025.06.008.

Yanay Rosen, Maria Brbić, Yusuf Roohani, Kyle Swanson, Ziang Li, and Jure Leskovec. Toward universal cell embeddings: integrating single-cell RNA-seq datasets across species with SATURN. *Nature Methods*, 21(8):1492–1500, 2024. doi: 10.1038/s41592-024-02191-z. URL https://doi.org/10.1038/s41592-024-02191-z.

Yanay Rosen, Yusuf Roohani, Ayush Agrawal, Leon Samotorčan, Stephen R. Quake, and Jure Leskovec. Universal cell embedding provides a foundation model for cell biology. *Nature*, 656 (8126):183–191, 2026. doi: 10.1038/s41586-026-10689-z. URL https://doi.org/10.1038/s41586-026-10689-z.

Christina V. Theodoris, Ling Xiao, Anant Chopra, Mark D. Chaffin, Zeina R. Al Sayed, Matthew C. Hill, Helene Martineau, Elizabeth M. Brydon, Zexian Zeng, X. Shirley Liu, and Patrick T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023. doi: 10.1038/s41586-023-06139-9. URL https://doi.org/10.1038/s41586-023-06139-9.

Shushan Toneyan and Peter K. Koo. Interpreting cis-regulatory interactions from large-scale deep neural networks. *Nature Genetics*, 56(11):2517–2527, 2024. doi: 10.1038/s41588-024-01923-3. URL https://doi.org/10.1038/s41588-024-01923-3.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A. Teichmann, Mor Nitzan, and Maria Brbić. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, 44(6):1050–1059, 2026. doi: 10.1038/s41587-025-02777-8. URL https://doi.org/10.1038/s41587-025-02777-8.

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

Xiaodong Yang et al. GeneCompass: deciphering universal gene regulatory mechanisms with a knowledge-informed cross-species foundation model. *Cell Research*, 34(12):830–845, 2024. doi: 10.1038/s41422-024-01034-y. URL https://doi.org/10.1038/s41422-024-01034-y.

Hengshi Yu, Weizhou Qian, Yuxuan Song, and Joshua D. Welch. Perturbnet predicts single-cell responses to unseen chemical and genetic perturbations. *Molecular Systems Biology*, 21(8):

960–982, 2025. doi: 10.1038/s44320-025-00131-3. URL https://doi.org/10.1038/s44320-025-00131-3.

Yuansong Zeng, Jiancong Xie, Ningyuan Shangguan, Zhuoyi Wei, Wenbing Li, Yun Su, Shuangyu Yang, Chengyang Zhang, Jinbo Zhang, Nan Fang, Hongyu Zhang, Yutong Lu, Huiying Zhao, Jue Fan, Weijiang Yu, and Yuedong Yang. CellFM: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells. *Nature Communications*, 16(1): 4679, 2025. doi: 10.1038/s41467-025-59926-5. URL https://doi.org/10.1038/s41467-025-59926-5.

Ouyang Zhu and Jun Li. Scouter predicts transcriptional responses to genetic perturbations with large language model embeddings. *Nature Computational Science*, 6(1):21–28, 2026. doi: 10.1038/s43588-025-00912-8. URL https://doi.org/10.1038/s43588-025-00912-8.

Yazdan Zinati, Abdulrahman Takiddeen, and Amin Emad. GRouNdGAN: GRN-guided simulation of single-cell RNA-seq data using causal generative adversarial networks. *Nature Communications*, 15(1):4055, 2024. doi: 10.1038/s41467-024-48516-6. URL https://doi.org/10.1038/s41467-024-48516-6.

CONTENTS

A Implementation and Experimental Details **15**

A.1 Candidate Graph Construction 15

A.2 Network Architecture and Message Normalization 15

A.3 Dataset Preprocessing and Splits 15

A.4 Training and Inference Settings 15

B Proofs of the Theoretical Properties **16**

B.1 Proof of Proposition 1 16

B.2 Proof of Proposition 2 16

B.3 Proof of Proposition 3 17

B.4 Proof of Proposition 4 17

B.5 Proof of Proposition 5 17

A Implementation and Experimental Details

A.1 Candidate Graph Construction

We first construct a candidate graph $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E}_0)$ from control cells only. The node set $\mathcal{V} = \{1, \dots, G\}$ contains one node for each gene, and the candidate edge set $\mathcal{E}_0$ contains directed gene pairs. Let N_0 be the number of control cells and let $\boldsymbol{X}^0 \in \mathbb{R}^{N_0 \times G}$ be their expression matrix. For genes i and j, we compute the Pearson correlation

$$\rho_{ij} = \frac{\sum_{n=1}^{N_0}(X_{ni}^0 - \mu_i)(X_{nj}^0 - \mu_j)}{\sqrt{\sum_{n=1}^{N_0}(X_{ni}^0 - \mu_i)^2}\sqrt{\sum_{n=1}^{N_0}(X_{nj}^0 - \mu_j)^2}}, \tag{9}$$

where X_{ni}^0 is the expression of gene i in control cell n, and $\mu_i = N_0^{-1}\sum_{n=1}^{N_0} X_{ni}^0$ is the mean control expression of gene i. For every source gene i, we retain directed edges to the r genes with the largest correlations, where $r = 12$ in our experiments. This construction uses no treated samples and therefore cannot reveal the response of a withheld intervention.

The candidate graph encodes possible routes of interaction rather than a complete causal network. IGWM assumes mechanistic locality, meaning that only a small subset of candidate edges is needed for one transition. It also assumes intervention dependence, meaning that the useful subset can change with the state and action. The graph editor described next turns these assumptions into a conditional sparse graph.

A.2 Network Architecture and Message Normalization

The state encoder f_x and edge network s_ϕ are two-layer multilayer perceptrons with GELU activations. The action encoder is the bias-free linear map $\boldsymbol{W}_a\boldsymbol{a}$. The velocity network F_θ is a two-layer multilayer perceptron with GELU activation: its first layer maps the concatenated state, message, state encoding, action encoding, and scalar time to $2H$ features, and its final layer outputs G velocities. We use $H = 128$.

For minibatch size B, the normalization uses $\overline{g}_{ij} = B^{-1}\sum_{b=1}^{B} g_{ij}^{(b)}$ and $d_j = \max(1, \sum_{i:(i,j)\in\mathcal{E}_0} \overline{g}_{ij})$, as in Eq. (4). This normalization depends on the minibatch during training and uses the single-state gates during single-state inference. The denominator is bounded below by one to avoid division by zero and amplification from a small incoming gate mass. The population flow result concerns per-sample fields and does not remove this implementation dependence.

A.3 Dataset Preprocessing and Splits

We use Norman19 (Norman et al., 2019) and VCC-H1 (Roohani et al., 2025). On Norman19, we retain the 2,000 genes with the highest variance in control cells and all perturbation targets, yielding 2,058 genes. The main benchmark tests 25 held-out double perturbations whose component singles appear in training, while other eligible conditions remain available for fitting. Qualitative response and gene-level analyses use a stricter setting trained only on controls and singles. For VCC-H1, we use the released processed H1 data and evaluate 110 single-gene intervention conditions. We retain up to 100 cells per intervention, reserve 20% of eligible intervention conditions for validation, and hold out 20% of cells within each remaining condition for testing. Thus, VCC-H1 measures prediction for observed interventions, not unseen targets or combinations. We use the provided log-transformed expression and select the 2,000 highest-variance control genes plus intervention targets. Within each dataset, all methods share expression, gene order, controls, and the fixed split.

For VCC-H1, the processed file is `arch1_processed_complete.h5ad`, distributed through scPertEval from the Virtual Cell Challenge public training release. Our split is constructed from this release and is not the official challenge leaderboard split. The fixed split seed is 7.

A.4 Training and Inference Settings

For both datasets, we fix the data split and repeat each experiment across five random seeds. IGWM has a hidden dimension of 128 and is trained for 100 epochs with batch size 128 and AdamW at a

learning rate of 3×10^{-4}. Inference uses 16 Euler steps. The candidate graph retains the 12 most correlated available neighbors per gene, and the sparse editor selects $K = 512$ edges per state and intervention. CPA and GEARS use their public configurations and train for at most 100 epochs. Each run uses one NVIDIA A100 80 GB GPU. The consecutive training seeds are 7, 8, 9, 10, and 11. We use Gaussian interpolation noise $\sigma = 0.01$, gate regularization weight $\lambda = 10^{-4}$, and gradient norm clipping at one. The gate regularizer acts through the straight-through surrogate in sparse mode, despite its constant forward value.

For an ordered sequence of S interventions $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_S)$, where S is the sequence length, inference composes $\widehat{T}_{\boldsymbol{a}_S} \circ \cdots \circ \widehat{T}_{\boldsymbol{a}_1}$. Each transition uses 16 Euler steps and recomputes its graph from the current state. No sequential loss or supervision is added to the main training objective.

B Proofs of the Theoretical Properties

The statements in Section 3 concern the population objective or a fixed learned vector field. They do not assert recovery of a causal gene network, statistical generalization to unseen actions, or global convergence of the nonconvex training procedure. For the ODE statements, the required bounds are assumed on a region containing the trajectories under consideration. The hard TopK editor can change discontinuously at score ties, so the smoothness assumptions need not hold across those boundaries.

B.1 Proof of Proposition 1

Proposition 1 (Local stability of graph editing). *Fix an action $\boldsymbol{a}$ and a state $\boldsymbol{x}$. Suppose the gap between the Kth and $(K+1)$th largest edge scores at $\boldsymbol{x}$ is $\gamma > 0$, and each score is L_s-Lipschitz near $\boldsymbol{x}$. The selected edge set is unchanged at any state $\boldsymbol{x}'$ in that neighborhood with $\|\boldsymbol{x}' - \boldsymbol{x}\|_2 < \gamma/(2L_s)$, with no distance restriction within the neighborhood when $L_s = 0$.*

Proof. Let $S(\boldsymbol{x})$ be the indices of the K largest edge scores $p_e(\boldsymbol{x}, \boldsymbol{a})$, with a fixed rule for ties, and let $\gamma = p_{(K)}(\boldsymbol{x}, \boldsymbol{a}) - p_{(K+1)}(\boldsymbol{x}, \boldsymbol{a}) > 0$. Suppose $\delta = \|\boldsymbol{x}' - \boldsymbol{x}\|_2$ and $|p_e(\boldsymbol{x}', \boldsymbol{a}) - p_e(\boldsymbol{x}, \boldsymbol{a})| \le L_s\delta$ for all candidate edges. For any $e \in S(\boldsymbol{x})$ and $f \notin S(\boldsymbol{x})$,

$$p_e(\boldsymbol{x}', \boldsymbol{a}) - p_f(\boldsymbol{x}', \boldsymbol{a}) \ge p_e(\boldsymbol{x}, \boldsymbol{a}) - p_f(\boldsymbol{x}, \boldsymbol{a}) - 2L_s\delta \ge \gamma - 2L_s\delta. \tag{10}$$

If $\delta < \gamma/(2L_s)$, every previously selected score remains larger than every unselected score, so $S(\boldsymbol{x}') = S(\boldsymbol{x})$. When $L_s = 0$, the scores are constant throughout the stated neighborhood. This result concerns the forward edge set and does not characterize the surrogate gradient in Eq. (3). □

B.2 Proof of Proposition 2

Proposition 2 (Local stability of the transition). *Fix an action $\boldsymbol{a}$. Suppose the exact trajectories from $\boldsymbol{x}$ and $\boldsymbol{y}$ stay in a region away from TopK switches where $\boldsymbol{V}_{\boldsymbol{a}}(t, \cdot)$ is L_x-Lipschitz. Then $\|T_{\boldsymbol{a}}(\boldsymbol{x}) - T_{\boldsymbol{a}}(\boldsymbol{y})\|_2 \le e^{L_x}\|\boldsymbol{x} - \boldsymbol{y}\|_2$.*

Proof. Let $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ solve $\dot{\boldsymbol{z}} = \boldsymbol{V}_{\boldsymbol{a}}(t, \boldsymbol{z})$ with initial values $\boldsymbol{x}$ and $\boldsymbol{y}$. The integral form of the ODE and the assumed Lipschitz condition give

$$\|\boldsymbol{x}(t) - \boldsymbol{y}(t)\|_2 \le \|\boldsymbol{x} - \boldsymbol{y}\|_2 + L_x \int_0^t \|\boldsymbol{x}(s) - \boldsymbol{y}(s)\|_2 \, ds. \tag{11}$$

Gronwall's inequality yields $\|\boldsymbol{x}(t) - \boldsymbol{y}(t)\|_2 \le e^{L_x t}\|\boldsymbol{x} - \boldsymbol{y}\|_2$. Evaluating at $t = 1$ proves the claim. The argument applies only while both paths remain in the region where $\boldsymbol{V}_{\boldsymbol{a}}$ is Lipschitz. A TopK switch may violate that condition, so the proposition does not assert global stability across all possible states. □

B.3 Proof of Proposition 3

Proposition 3 (Population flow target). *Assume $\mathbb{E}\|\boldsymbol{x}_1 - \boldsymbol{x}_0\|_2^2 < \infty$. Over all square-integrable measurable per-sample fields, the minimizer of the unregularized population squared velocity loss is $\boldsymbol{v}^*(\boldsymbol{x}, t, \boldsymbol{a}) = \mathbb{E}[\boldsymbol{x}_1 - \boldsymbol{x}_0 \mid \boldsymbol{x}_t = \boldsymbol{x}, t, \boldsymbol{a}]$ almost everywhere.*

Proof. Let $\boldsymbol{Y} = \boldsymbol{x}_1 - \boldsymbol{x}_0$ and $Z = (\boldsymbol{x}_t, t, \boldsymbol{a})$. Assume $\mathbb{E}\|\boldsymbol{Y}\|_2^2 < \infty$ and define $\boldsymbol{m}(Z) = \mathbb{E}[\boldsymbol{Y} \mid Z]$. For any measurable field $\boldsymbol{v}(Z)$ with finite squared risk, conditional expectation gives

$$\mathbb{E}\|\boldsymbol{v}(Z) - \boldsymbol{Y}\|_2^2 = \mathbb{E}\|\boldsymbol{v}(Z) - \boldsymbol{m}(Z)\|_2^2 + \mathbb{E}\|\boldsymbol{m}(Z) - \boldsymbol{Y}\|_2^2. \tag{12}$$

The cross term vanishes because $\boldsymbol{v}(Z) - \boldsymbol{m}(Z)$ is measurable with respect to Z and $\mathbb{E}[\boldsymbol{m}(Z) - \boldsymbol{Y} \mid Z] = \boldsymbol{0}$. The second term does not depend on $\boldsymbol{v}$, so the first term is minimized by $\boldsymbol{v} = \boldsymbol{m}$ almost everywhere. The factor $1/G$ in Eq. (7) does not change the minimizer. The argument uses the interpolation distribution in Eq. (6), including its Gaussian noise, and does not require observed pairs of control and treated cells. It applies to an unrestricted per-sample field and the unregularized population loss. A finite parameterization, minibatch-dependent normalization, the gate regularizer, and straight-through gradients may change the attained solution. □

B.4 Proof of Proposition 4

Proposition 4 (Euler approximation error)**.** *Suppose $\boldsymbol{V}_{\boldsymbol{a}}$ is L_x-Lipschitz on a region containing an exact trajectory and its Euler iterates, and the total time derivative of $\boldsymbol{V}_{\boldsymbol{a}}(t, \boldsymbol{x}(t))$ along that trajectory is bounded by M. Then $\|\widehat{T}_{\boldsymbol{a}}(\boldsymbol{x}) - T_{\boldsymbol{a}}(\boldsymbol{x})\|_2 \le \frac{M}{2L}\frac{e^{L_x}-1}{L_x} = O(L^{-1})$, where the fraction is one when $L_x = 0$.*

Proof. Put $h = 1/L$, $t_\ell = \ell h$, and $e_\ell = \|\boldsymbol{x}(t_\ell) - \boldsymbol{x}^{(\ell)}\|_2$. Integrating the exact path over one step and using the bound on its total velocity derivative gives a local truncation error of at most $Mh^2/2$. The Lipschitz condition then implies

$$e_{\ell+1} \le (1 + L_x h)e_\ell + \frac{Mh^2}{2}, \qquad e_0 = 0. \tag{13}$$

Summing the geometric recurrence gives $e_L \le \frac{Mh}{2L_x}((1 + L_x h)^L - 1)$ when $L_x > 0$. Since $(1 + L_x h)^L \le e^{L_x}$, the claimed bound follows. If $L_x = 0$, the recurrence directly yields $e_L \le Mh/2$. This is a discretization bound for a fixed learned field; it does not bound the difference between the learned and biological dynamics. □

B.5 Proof of Proposition 5

Proposition 5 (Sequential error propagation)**.** *Let $T^*_{\boldsymbol{a}_1}$ and $T^*_{\boldsymbol{a}_2}$ be reference transitions, with $T^*_{\boldsymbol{a}_2}$ being C_2-Lipschitz. If $\|\widehat{T}_{\boldsymbol{a}_i}(\boldsymbol{z}) - T^*_{\boldsymbol{a}_i}(\boldsymbol{z})\|_2 \le \varepsilon_i$ for $i \in \{1, 2\}$ on the relevant states, then $\|\widehat{T}_{\boldsymbol{a}_2}(\widehat{T}_{\boldsymbol{a}_1}(\boldsymbol{x})) - T^*_{\boldsymbol{a}_2}(T^*_{\boldsymbol{a}_1}(\boldsymbol{x}))\|_2 \le \varepsilon_2 + C_2\varepsilon_1$.*

Proof. Insert $T^*_{\boldsymbol{a}_2}(\widehat{T}_{\boldsymbol{a}_1}(\boldsymbol{x}))$ between the two terms being compared. By the triangle inequality,

$$\begin{aligned}
&\|\widehat{T}_{\boldsymbol{a}_2}(\widehat{T}_{\boldsymbol{a}_1}(\boldsymbol{x})) - T^*_{\boldsymbol{a}_2}(T^*_{\boldsymbol{a}_1}(\boldsymbol{x}))\|_2 && (14)\\
&\quad \le \|\widehat{T}_{\boldsymbol{a}_2}(\widehat{T}_{\boldsymbol{a}_1}(\boldsymbol{x})) - T^*_{\boldsymbol{a}_2}(\widehat{T}_{\boldsymbol{a}_1}(\boldsymbol{x}))\|_2 + \|T^*_{\boldsymbol{a}_2}(\widehat{T}_{\boldsymbol{a}_1}(\boldsymbol{x})) - T^*_{\boldsymbol{a}_2}(T^*_{\boldsymbol{a}_1}(\boldsymbol{x}))\|_2 && (15)\\
&\quad \le \varepsilon_2 + C_2\varepsilon_1. && (16)
\end{aligned}$$

The first bound applies to the predicted intermediate state, which is why the approximation assumption must include that state. The result concerns an ordered reference rollout and gives no equality with $T^*_{\boldsymbol{a}_1+\boldsymbol{a}_2}(\boldsymbol{x})$. □