VCLMU: MECHANISM-CENTRIC VIRTUAL CELL WORLD MODELING FOR PERTURBATION RESPONSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting cellular responses to genetic perturbations is a central capability of virtual cell models and an important foundation for computational modeling of biological perturbations. Many existing perturbation models achieve accurate prediction of observed responses without explicitly representing the internal state transitions or biological mechanisms underlying them. We introduce VCLMU, a mechanism-centric world model for virtual cells. VCLMU brings the object-centric principle of world modeling to cellular systems by representing internal cellular state as a collection of Latent Mechanism Units (LMUs),which represent latent biological mechanisms corresponding to objects in an object-centric world model. Each LMU has a reusable identity and a cell-specific state. Its identity is biologically grounded in multimodal gene evidence, while its state is obtained from the current cellular observation to capture cell-specific mechanism states. Given the current cellular state and a perturbation action, the world model predicts the perturbation-induced transition of each active LMU, then aggregates the resulting mechanism states and decodes them into the transcriptional response. Built on the same LMU state representation, VCLMU supports both forward perturbation response prediction and inverse perturbation inference. The forward model predicts the perturbed cellular response from the current state and perturbation action, whereas the inverse model infers the perturbation action from the observed state transition. VCLMU is trained through two-stage pretraining. The first stage uses approximately 165K pseudo-bulk perturbation profiles to learn stable population-level responses for individual perturbation conditions and establish robust perturbation dynamics. The second stage further pretrains on large-scale, gene-aligned single-cell perturbation data to capture cell-specific heterogeneity and stochastic variation beyond the population mean. We evaluate VCLMU through forward response prediction, inverse perturbation inference, mechanistic interpretation, and ablation studies, demonstrating its ability to generalize to unseen perturbations while learning biologically structured latent dynamics.

1 INTRODUCTION

Predicting cellular responses to genetic perturbations is a central capability of virtual cell models and an important foundation for computational understanding and prediction of cellular behavior. Advances in high-throughput single-cell perturbation profiling now enable systematic measurement of transcriptional responses to genetic perturbations across large cell populations (Dixit et al., 2016; Adamson et al., 2016; Norman et al., 2019; Replogle et al., 2022), with these datasets increasingly consolidated into harmonized public resources (Peidli et al., 2024). Such data have driven rapid progress in perturbation response prediction, pretrained transcriptomic models, and large-scale virtual cell models, while also motivating systematic evaluation of generalization to unseen perturbations (Roohani et al., 2025). Reliable virtual cell models could enable large numbers of candidate perturbations to be screened computationally before experimental testing, reducing experimental cost and resource use, decreasing reliance on large-scale animal studies, and supporting the study of disease-associated cellular mechanisms, therapeutic targets, and biological perturbation mechanisms. More broadly, a capable virtual cell should not only predict how a cell responds to a given perturbation, but also help identify perturbations capable of producing a desired cellular change.

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.

Despite rapid progress in perturbation response prediction, many existing models remain focused primarily on the observed transcriptional outcome. Given a current cellular state and a genetic perturbation, these models typically predict the perturbed transcriptional state directly, while the biological mechanisms through which the perturbation acts and jointly gives rise to the final response remain implicit in the input-output mapping. As a result, even accurate prediction of the perturbed expression profile may not reveal which biological mechanisms are influenced or how their changes combine to produce the observed response, limiting mechanistic interpretability. Advancing virtual cell models therefore requires not only accurate prediction of transcriptional outcomes, but also more explicit representation and modeling of perturbation-induced changes at the mechanism level.

Explicitly modeling perturbation-induced changes at the mechanism level not only improves interpretation of the resulting transcriptional response, but also provides a basis for inferring which perturbations could have produced an observed cellular change. A complete virtual cell model should therefore support both forward perturbation response prediction and inverse perturbation inference. Forward prediction simulates the cellular response induced by a given genetic perturbation, whereas inverse inference identifies perturbations that could account for an observed cellular change, supporting perturbation identification, target discovery, and candidate prioritization. Although these tasks describe opposite directions of the same relationship between perturbations and cellular responses, they are typically treated as separate problems, without a shared state representation and dynamics framework that connects them.

Unifying forward perturbation prediction and inverse perturbation inference within a shared state representation and dynamics framework requires a model that explicitly captures the relationship among the current state, perturbation action, and resulting state change. World modeling provides a suitable framework for this goal. Rather than directly mapping inputs to final observations, a world model represents the current state of a system and learns how an action transforms that state to generate a new observation (Ha & Schmidhuber, 2018; Hafner et al., 2019b;a; 2025). In a virtual cell, a genetic perturbation can be viewed as an action on cellular state, with the transcriptional response arising from the resulting state transition. Under this formulation, forward prediction and inverse inference become two directions of the same perturbation dynamics and can share a common state representation and latent dynamics.

Cellular responses to genetic perturbations often involve coordinated changes across multiple biological mechanisms, and different mechanisms may respond differently to the same perturbation. Representing the cell with a single global latent state makes these mechanism-specific states and responses difficult to distinguish explicitly. Object-centric world modeling offers a structured alternative by decomposing system state into multiple units, each with its own identity and state, and modeling how these units change under an action. We bring this principle to virtual cells through Latent Mechanism Units (LMUs), which serve as latent biological units corresponding to objects in an object-centric world model. Each LMU has a reusable identity and a cell-specific state. Its identity is biologically grounded in multimodal gene evidence, while its state is instantiated from the current cellular observation. This representation organizes cellular state into a collection of LMUs with distinct biological associations and current states, providing a structured basis for modeling mechanism-specific perturbation dynamics.

Building on this LMU representation, we develop VCLMU to explicitly model mechanism-level state transitions induced by genetic perturbations. Given the current cellular state and a perturbation action, the model predicts the state transition of each active LMU separately, allowing different biological units to respond differently to the same perturbation and making their individual responses explicitly distinguishable in the latent state space. This factorized transition structure enables the model to represent a perturbation as a coordinated change across multiple biological units rather than as a single undifferentiated shift in cellular state. To capture single-cell variation that is not fully determined by the current state and perturbation, these LMU transitions are modeled stochastically. The transitioned LMU states are then aggregated and decoded into the perturbed transcriptional response. Using the same LMU state representation, VCLMU supports both forward perturbation response prediction and inverse perturbation inference, linking perturbation actions and cellular state transitions within a unified state space. To learn perturbation dynamics that are both stable and cell-specific, we adopt two-stage pretraining: the first stage uses approximately 165K pseudo-bulk perturbation profiles to learn stable population-level response patterns, while the second stage continues pretraining on large-scale, gene-aligned single-cell perturbation data to capture cell-specific heterogeneity and stochastic variation.

Our main contributions are:

- We formulate the virtual cell as an object-centric world model, treating genetic perturbations as actions on cellular state and using a shared LMU state representation to support forward perturbation response prediction and inverse perturbation inference.
- We introduce Latent Mechanism Units (LMUs), which decompose cellular state into latent biological units with reusable identities and cell-specific states, and explicitly model their mechanism-specific stochastic state transitions under genetic perturbations.
- We develop a two-stage pretraining strategy from pseudo-bulk perturbation profiles to gene-aligned single-cell perturbation data, and evaluate the learned perturbation dynamics through forward prediction, inverse inference, mechanistic interpretation, and ablation studies, assessing generalization to unseen perturbations and LMU interpretability.

2 Related Work

Virtual cell models and forward perturbation response prediction. Single-cell perturbation response prediction has become one of the most developed tasks in virtual cell modeling. Early approaches learned mappings from unperturbed states to perturbed expression distributions using generative latent-variable models, including scGen (Lotfollahi et al., 2019), trVAE (Lotfollahi et al., 2020), and scPreGAN (Wei et al., 2022). Subsequent methods introduced richer perturbation representations and structured latent effects. CPA models compositional perturbations and covariates (Lotfollahi et al., 2023), chemCPA incorporates chemical representations for unseen compounds (Hetzel et al., 2022), SAMS-VAE learns sparse perturbation-specific latent mechanism shifts (Bereket & Karaletsos, 2023), and CellOT formulates perturbation response as transport between unpaired control and perturbed cell populations (Bunne et al., 2023). For generalization to unseen genetic perturbations, GEARS incorporates gene interaction structure (Roohani et al., 2024), biolord learns disentangled state transformations (Piran et al., 2024), and PerturbNet maps perturbation representations to distributions of cellular states (Yu et al., 2025). More recent methods exploit pretrained gene representations, biological knowledge, and single-cell foundation models, including Scouter (Zhu & Li, 2026), TxPert (Wenkel et al., 2026), and PertAdapt (Bai et al., 2026). In parallel, large-scale transcriptomic pretraining has produced models such as Geneformer (Theodoris et al., 2023), scGPT (Cui et al., 2024), scFoundation (Hao et al., 2024), and AIDO.Cell (Ho et al., 2024), several of which support perturbation modeling downstream. Recent virtual cell models increasingly train directly on large perturbation collections, including STATE (Adduri et al., 2025), Tahoe-x1 (Gandhi et al., 2025), Stack (Dong et al., 2026), X-Cell (Wang et al., 2026), PerturbDiff (Yuan et al., 2026), SCALE (Chen et al., 2026), and PerturbGen (Chi Hao Ly et al., 2026). These approaches expand model scale, response distributions, and cross-context generalization, but primarily remain focused on forward perturbation response prediction, mapping a cellular context and perturbation to a perturbed transcriptomic observation or response distribution.

World models and object-centric world models. World models learn latent states together with action-conditioned dynamics to describe how a system changes under actions, rather than predicting outcomes only in observation space (Ha & Schmidhuber, 2018; Hafner et al., 2019b;a; 2025). They typically encode high-dimensional observations into compact latent states, learn how actions transform those states, and use predicted future states for generation, planning, or control. Object-centric world models further decompose global state into multiple entities with their own identities, states, and dynamics, allowing different components of a complex system to be modeled separately and to respond differently to the same action (Locatello et al., 2020; Villar-Corrales et al., 2023; Villar-Corrales & Behnke, 2025; Daniel et al., 2026). World-model formulations have recently begun to appear explicitly in virtual cell modeling, including VCWorld, GeneJEPA, AlphaCell, Lingshu-Cell, Chreode, and CellOS. These approaches represent cellular state and perturbation-induced transitions through biological reasoning, predictive representation learning, continuous latent dynamics, or generative modeling (Wei et al., 2026; Litman et al., 2025; Chuai et al., 2026; Zhang et al., 2026a; Qiu et al., 2026; Zhou et al., 2026). Together, they move virtual cell modeling beyond direct prediction of final transcriptomic responses toward explicit modeling of how perturbations transform cellular state. VCLMU differs by adopting an object-centric formulation that factorizes cellular state into latent biological units and explicitly models perturbation-induced stochastic transitions at the LMU level.

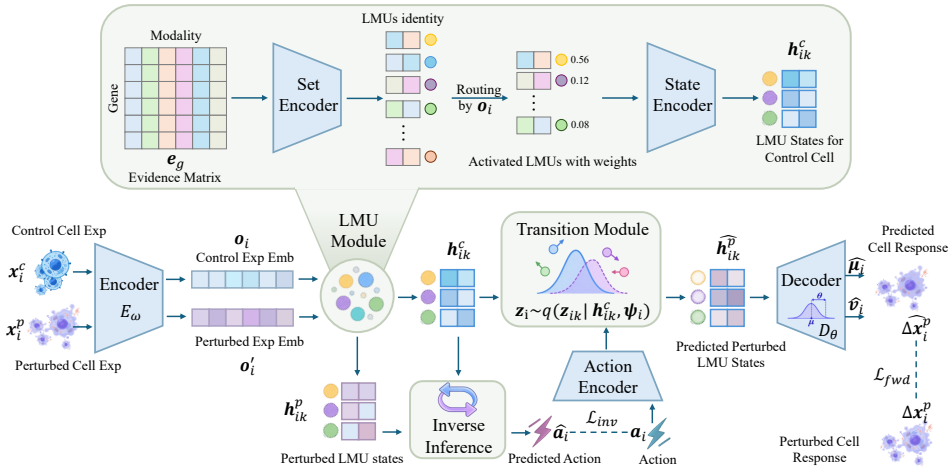


Figure 1: VCLMU overview. Gene evidence $\mathbf{e}_g$ defines reusable LMU identities (Eq. 1). The control reference $\mathbf{x}_i^c$ is encoded by E_ω to select active LMUs and determine their states $\mathbf{h}_{ik}^c$ (Eqs. 2–3). The action embedding $\boldsymbol{\psi}_i$ induces stochastic LMU state transitions (Eq. 4), which are decoded by D_θ into the predicted response (Eq. 6). During training, the observed perturbed profile $\mathbf{x}_i^p$ is encoded into $\mathbf{h}_{ik}^p$ to provide transition targets and support inverse inference (Eq. 8), but it is not used for forward prediction.

Inverse perturbation inference. Compared with forward perturbation response prediction, identifying a genetic perturbation from cellular states before and after perturbation remains underexplored. Heimdall explicitly formulates reverse perturbation prediction as paired-cell classification, using a control cell and its corresponding perturbed cell to predict the gene perturbation identity (Haber et al., 2025). This formulation is closely related to the inverse task considered in our work. GREmLN further studies reverse perturbation prediction by predicting perturbation labels from perturbed expression profiles, although it does not explicitly condition on the corresponding control state (Zhang et al., 2026b). Earlier work including scGPT and CellFM also evaluated reverse perturbation prediction, but primarily through forward-model-based candidate search, where predicted responses under candidate perturbations are matched against an observed perturbed state to identify the likely perturbation (Cui et al., 2024; Zeng et al., 2025). Together, these studies demonstrate that transcriptomic observations can support inference of genetic perturbations, but direct paired-state inverse modeling remains limited. VCLMU instead places inverse perturbation inference and forward response prediction within the same LMU state representation, directly inferring the perturbation action from the encoded perturbation-induced state change.

3 Method

3.1 Problem Formulation

Each sample i consists of an observed perturbed expression profile $\mathbf{x}_i^p \in \mathbb{R}^G$ over G genes and the index a_i of the perturbed gene. $\mathbf{x}_i^p$ is a single perturbed cell, except in Stage 1 pretraining, where it is a pseudo-bulk profile (Section 3.6). Because a cell cannot be profiled both before and after perturbation, its unperturbed state is approximated by a control reference $\mathbf{x}_i^c \in \mathbb{R}^G$, the mean of training control cells from the same cellular context (Appendix B). We write x_{ig}^c and x_{ig}^p for the expression of gene g in $\mathbf{x}_i^c$ and $\mathbf{x}_i^p$. The forward task predicts the response $\Delta\mathbf{x}_i = \mathbf{x}_i^p - \mathbf{x}_i^c$ from $(\mathbf{x}_i^c, a_i)$, and the inverse task identifies a_i from $(\mathbf{x}_i^c, \mathbf{x}_i^p)$. Each gene g has a multimodal evidence embedding $\mathbf{e}_g \in \mathbb{R}^d$, which fuses functional annotations, pathways, interaction networks, protein sequence, and expression resources (Appendix C), together with a gene-regulatory-network (GRN) neighborhood (Kamimoto et al., 2023). The evidence supplies multimodal biological information about each gene, while the response model itself is learned end-to-end. Rather than fitting $(\mathbf{x}^c, a) \mapsto \Delta\mathbf{x}$ directly, VCLMU follows the structure of a world model (Ha & Schmidhuber, 2018; Hafner et al., 2019a): it encodes the control profile into a set of mechanism states, changes each state under

the perturbation, and decodes the changed states into the response (Fig. 1). The inverse task reads the perturbation from the same mechanism states.

3.2 Cellular State as a Set of Latent Mechanism Units

VCLMU represents a cell with K Latent Mechanism Units (LMUs), of which only a sparse subset $\mathcal{A}_i$ of K_a units is active for a given observation, following slot-based representations and sparse routing (Locatello et al., 2020; Shazeer et al., 2017). Each LMU has an identity shared across all cells and a state specific to one observation.

LMU k owns a set $\mathcal{C}_k$ of C candidate genes, the genes currently assigned to it. Its identity combines a learnable query $\mathbf{q}_k$ with a permutation-invariant summary SetEnc (Lee et al., 2019) of the evidence of those genes:

$$\boldsymbol{\ell}_k = \mathrm{LayerNorm}(\alpha \mathbf{q}_k + (1-\alpha)\,\mathrm{SetEnc}\{\mathbf{e}_g : g \in \mathcal{C}_k\})\,, \tag{1}$$

where $\alpha \in (0,1)$ is learned. The query lets a unit fit programs supported by the data, and the evidence term ties the unit to a concrete set of genes, through which the unit is later interpreted (Appendix D.4). Candidate sets are refreshed periodically: $\mathcal{C}_k$ is reset to the C genes whose evidence embeddings are closest to the unit's identity, with a retention bonus for current members (exact rule in Appendix A.2). Candidate sets may overlap, so genes are not hard-partitioned across units.

An observation encoder summarizes the control profile as $\mathbf{o}_i^c = E_\omega(\mathbf{x}_i^c)$. The K_a units whose identities best match this summary are activated, and a softmax over the active units gives their activity weights:

$$\mathcal{A}_i = \mathrm{Top}_{K_a}\Big[\cos(W_o\mathbf{o}_i^c, \bar{\boldsymbol{\ell}}_k) + \tfrac{c_{\mathrm{ucb}}}{\sqrt{N_k+1}}\Big]_{k=1}^{K}, \qquad w_{ik}^c = \mathrm{softmax}_{k\in\mathcal{A}_i}\big(\cos(W_o\mathbf{o}_i^c, \bar{\boldsymbol{\ell}}_k)/\tau\big), \tag{2}$$

where $\bar{\boldsymbol{\ell}}_k$ is an exponential moving average of $\boldsymbol{\ell}_k$, τ is a learned temperature, and $c_{\mathrm{ucb}}/\sqrt{N_k+1}$ is an upper-confidence-bound (UCB) style exploration bonus: N_k counts how often unit k has been selected, so rarely used units receive a small boost (Appendix A.3). The state of an active unit summarizes how its own candidate genes are expressed in the observation:

$$\mathbf{h}_{ik}^c = f_{\mathrm{state}}\big(\boldsymbol{\ell}_k, \{(\mathbf{e}_g, x_{ig}^c)\}_{g\in\mathcal{C}_k}, \mathbf{o}_i^c\big)\,, \quad k \in \mathcal{A}_i. \tag{3}$$

Concretely, f_{state} adds an embedding of each candidate gene's expression to its evidence, pools these gene tokens with the same SetEnc, fuses the result with the identity $\boldsymbol{\ell}_k$, adds a gated amount of the global summary $\mathbf{o}_i^c$, and scales the output according to the unit's activity weight (Appendix A.3). The identity therefore describes what a unit represents, while its state describes how the unit's genes are expressed in the current observation. Routing depends only on the control profile, so the active set is a property of the starting cell, not of the outcome. We stack the active states into $\mathbf{H}_i^c \in \mathbb{R}^{K_a\times d}$. During training, the same encoder and the same control-selected indices $\mathcal{A}_i$ encode the observed perturbed profile $\mathbf{x}_i^p$ into $\mathbf{H}_i^p$, with activity weights w_{ik}^p recomputed from $\mathbf{o}_i^p = E_\omega(\mathbf{x}_i^p)$, so corresponding rows of $\mathbf{H}_i^c$ and $\mathbf{H}_i^p$ describe the same LMU before and after the perturbation.

3.3 Perturbation as an Action on Mechanism States

A perturbation is encoded once into an action embedding $\boldsymbol{\psi}_i = f_a(a_i, \{\mathbf{e}_g\}, \mathrm{GRN})$, which attends from the evidence of the target gene $\mathbf{e}_{a_i}$ to the evidence of its strongest GRN neighbors, so the action carries both the gene and its regulatory context (Appendix A.4). The same action is then applied to every active LMU. Because each unit receives the same action but starts from its own state, one perturbation can strongly change some mechanisms and leave others untouched, a pattern that a single global latent state does not represent explicitly. Transitions are residual and stochastic. A per-unit latent $\mathbf{z}_{ik}$ is drawn from a learned conditional Gaussian and enters the update:

$$\widehat{\mathbf{h}}_{ik}^p = \mathbf{h}_{ik}^c + F_\theta(\mathbf{h}_{ik}^c, \boldsymbol{\psi}_i, \mathbf{z}_{ik}), \qquad \mathbf{z}_{ik} \sim q(\mathbf{z}_{ik} \mid \mathbf{h}_{ik}^c, \boldsymbol{\psi}_i). \tag{4}$$

The residual form means the model predicts a change to each mechanism rather than re-deriving the cell from scratch. The variation of latent variable $\mathbf{z}_{ik}$ represents variability that the control state and the action do not determine, and q is regularized toward the prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ by a Kullback–Leibler (KL) penalty $\mathcal{L}_{\mathrm{KL}}$ (Kingma & Welling, 2013). Neither q nor F_θ receives the observed outcome, so

it cannot leak into the forward path. Stacking $\widehat{\mathbf{h}}^p_{ik}$ into $\widehat{\mathbf{H}}^p_i$, we supervise the transition by matching the predicted change of each unit to the observed one,

$$\mathcal{L}_{\text{dyn}} = \text{SmoothL1}\Big(\widehat{\mathbf{H}}^p_i - \mathbf{H}^c_i, \text{sg}[\mathbf{H}^p_i - \mathbf{H}^c_i]\Big), \tag{5}$$

where the stop-gradient sg (Van Den Oord et al., 2017) blocks gradients through the target, so the encoder supplies the target change but is not trained to make it easier to match.

3.4 Decoding the Predicted State

Mechanisms are not independent, so a one-layer Transformer first lets the predicted LMU states interact. Each gene then attends to these states, using its own evidence embedding as the query (Vaswani et al., 2017), and outputs a mean and a variance for its response:

$$(\widehat{\boldsymbol{\mu}}_i, \widehat{\mathbf{v}}_i) = D_\theta\Big(\{\widehat{\mathbf{h}}^p_{ik}\}_{k\in\mathcal{A}_i}, \{\mathbf{e}_g\}_{g=1}^G\Big). \tag{6}$$

For a given draw of the transition latents $\mathbf{Z}_i = (\mathbf{z}_{ik})_{k\in\mathcal{A}_i}$, these outputs define a factorized Gaussian over the response, $p(\Delta\mathbf{x}_i \mid \mathbf{x}^c_i, a_i, \mathbf{Z}_i) = \prod_{g=1}^G \mathcal{N}(\Delta x_{ig}; \widehat{\mu}_{ig}, \widehat{v}_{ig})$, and the point prediction is $\widehat{\mathbf{x}}^p_i = \mathbf{x}^c_i + \widehat{\boldsymbol{\mu}}_i$. The same gene evidence thus defines the LMU identities in Eq. 1 and reads them out gene by gene here, which keeps the latent units tied to genes at both ends of the model.

3.5 Inverse Inference from Observed State Changes

The inverse task uses the same LMU states and does not run the transition network. Given a control and an observed perturbed profile, both are encoded through the control-selected units as in Section 3.2, and each unit contributes a token that describes how it changed:

$$\mathbf{r}_{ik} = f_{\text{inv}}([\mathbf{h}^c_{ik}; \mathbf{h}^p_{ik}; \mathbf{h}^p_{ik} - \mathbf{h}^c_{ik}; \mathbf{h}^p_{ik} \odot \mathbf{h}^c_{ik}]), \quad k \in \mathcal{A}_i, \tag{7}$$

where $[\cdot;\cdot]$ denotes concatenation and $\odot$ the elementwise product. The tokens are attention-pooled into $\bar{\mathbf{r}}_i$ and scored against every gene through its evidence embedding:

$$s_{ig} = \exp(\xi)\cos(\bar{\mathbf{r}}_i, W_{\text{inv}}\mathbf{e}_g), \qquad \mathcal{L}_{\text{inv}} = \text{CE}(\mathbf{s}_i, a_i), \qquad \widehat{a}_i = \arg\max_g s_{ig}, \tag{8}$$

where ξ is a learned logit scale and CE is cross-entropy over the G genes. The perturbation a_i serves only as the target, never as an input. During training, $\mathcal{L}_{\text{inv}}$ also encourages the encoded state changes to retain perturbation-specific information.

3.6 Learning Objective and Two-Stage Pretraining

Supervising only the decoded response would leave the mechanism changes underdetermined, since many different sets of LMU changes decode to the same profile. The objective therefore combines the response likelihood with supervision on the LMU states and a few regularizers, jointly constraining both the observable perturbation response and the latent transition that produces it:

$$\mathcal{L} = \mathcal{L}_{\text{fwd}} + \lambda_{\text{dyn}}\mathcal{L}_{\text{dyn}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{obs}}\mathcal{L}_{\text{obs}} + \lambda_{\text{inv}}\mathcal{L}_{\text{inv}} + \mathcal{R}_{\text{LMU}}. \tag{9}$$

Here $\mathcal{L}_{\text{fwd}}$ is the negative log-likelihood of $\Delta\mathbf{x}_i$ under the Gaussian of Section 3.4, evaluated at the sampled latents $\mathbf{Z}_i$, $\mathcal{L}_{\text{dyn}}$ is Eq. 5, and $\mathcal{L}_{\text{KL}}$ is the KL penalty on $q(\mathbf{z}_{ik} \mid \mathbf{h}^c_{ik}, \boldsymbol{\psi}_i)$ averaged over the active units. The reconstruction loss $\mathcal{L}_{\text{obs}}$ maps the activity-weighted average of the observed perturbed states back to $\mathbf{x}^p_i$ with a linear layer, so the target in Eq. 5 stays tied to the observed profile rather than drifting to an arbitrary code. $\mathcal{L}_{\text{inv}}$ is the inverse loss of Eq. 8. The regularizer $\mathcal{R}_{\text{LMU}}$ keeps LMUs distinct and used: it penalizes queries that drift from their evidence-based initialization, penalizes pairs of similar identities, and keeps the entropy of the activity weights w^c_{ik} near a moderate target, so that neither a single unit dominates nor all units share activity equally (Appendix A.6).

We pretrain in two stages on perturbation-disjoint splits (Appendices A.7 and B). Stage 1 uses approximately 165K pseudo-bulk profiles, each aggregating 200 cells from the same perturbation and cellular context, which gives low-noise targets for learning stable population-level responses. Stage 2 continues on single-cell profiles aligned to the same gene vocabulary, freezes the evidence projections, and disables $\mathcal{L}_{\text{dyn}}$ and $\mathcal{L}_{\text{obs}}$ while keeping the forward and inverse objectives. This stage refines the learned perturbation dynamics with cell-specific heterogeneity that is suppressed by pseudo-bulk aggregation. At forward inference the perturbed profile is unavailable: the model routes and encodes $\mathbf{x}^c_i$, encodes a_i, sets each $\mathbf{z}_{ik}$ to its conditional mean, and returns $\widehat{\mathbf{x}}^p_i = \mathbf{x}^c_i + \widehat{\boldsymbol{\mu}}_i$. Inverse inference encodes $\mathbf{x}^c_i$ and $\mathbf{x}^p_i$ and returns $\widehat{a}_i$ from Eq. 8.

Table 1: Forward perturbation prediction performance averaged across ten evaluation datasets. Values report macro mean ± cross-dataset standard deviation. Higher is better. The best result is shown in bold and the second best is underlined.

	GenePert	Geneformer-FT	scFoundation-FT	scDFM	LPM	Tahoe-x1-FT
Jaccard@20	0.0155 ± 0.0189	0.0622 ± 0.0765	0.0634 ± 0.0774	0.0091 ± 0.0100	0.0287 ± 0.0227	0.0193 ± 0.0137
ΔPCC	0.0036 ± 0.0077	$\underline{0.3669 \pm 0.1788}$	0.3655 ± 0.1775	0.0227 ± 0.0162	0.0268 ± 0.0737	0.0494 ± 0.0552

	CPA	GEARS	scGPT-ZS	scGPT-FT	VCLMU-S1-FT	**VCLMU-S2-FT**
Jaccard@20	0.0397 ± 0.0476	0.0558 ± 0.0618	0.0409 ± 0.0522	0.0406 ± 0.0436	$\underline{0.1238 \pm 0.1331}$	$\mathbf{0.1648 \pm 0.1499}$
ΔPCC	0.0894 ± 0.0474	0.2648 ± 0.2097	0.0980 ± 0.0511	0.2360 ± 0.2003	0.3498 ± 0.2062	$\mathbf{0.3992 \pm 0.1959}$

Table 2: Ablation study averaged across ten evaluation datasets. Values report macro mean ± cross-dataset standard deviation. Higher is better. The best result is shown in bold and the second best is underlined.

	w/o LMU	w/o Dynamics	w/o Transition	w/o GRN	**VCLMU-S2-FT**
Jaccard@20	0.1388 ± 0.1296	$\underline{0.1486 \pm 0.1406}$	0.1362 ± 0.1287	0.1475 ± 0.1341	$\mathbf{0.1648 \pm 0.1499}$
ΔPCC	0.3536 ± 0.1945	0.3617 ± 0.1781	0.3422 ± 0.1969	$\underline{0.3803 \pm 0.1935}$	$\mathbf{0.3992 \pm 0.1959}$

4 EXPERIMENTS

4.1 BASELINE

For forward perturbation prediction, we compare VCLMU with a broad set of perturbation prediction and pretrained single-cell models, including GenePert (Chen & Zou, 2024), Geneformer-FT (Theodoris et al., 2023), scFoundation-FT (Hao et al., 2024), scDFM (Yu et al., 2026), LPM (Miladinovic et al., 2025), Tahoe-x1-FT (Gandhi et al., 2025), CPA (Lotfollahi et al., 2023), GEARS (Roohani et al., 2024), scGPT-ZS, and scGPT-FT (Cui et al., 2024). Geneformer, scFoundation, Tahoe-x1, and scGPT are evaluated with downstream fine-tuning where applicable, while scGPT-ZS evaluates zero-shot transfer. We additionally report VCLMU-S1-FT to isolate the effect of Stage 2 single-cell pretraining. For inverse perturbation inference, we compare VCLMU with CPA-FC, GEARS-FC, scGPT-FC, Geneformer-FT, scFoundation-FT, and TxPert-FT. FC denotes forward-consistency retrieval, where candidate perturbations are ranked by agreement between their predicted responses and the observed cellular state transition. All methods are evaluated on the same perturbation-disjoint splits, with predictions aligned to a common evaluation gene space.

4.2 FORWARD

We first evaluate forward response prediction on held-out genetic perturbations across ten datasets. Table 1 reports the macro average across datasets for Jaccard@20 and $\Delta\mathrm{PCC}$, measuring recovery of strongly responsive genes and overall perturbation-induced expression changes, respectively. VCLMU-S2-FT achieves the highest performance on both metrics, reaching 0.1648 Jaccard@20 and 0.3992 $\Delta\mathrm{PCC}$. For Jaccard@20, this represents a substantial improvement over the strongest non-VCLMU baseline, scFoundation-FT at 0.0634, indicating that VCLMU more accurately identifies which genes respond most strongly to previously unseen perturbations. The improvement is also consistent at the expression level, where VCLMU-S2-FT exceeds Geneformer-FT, the strongest external baseline in $\Delta\mathrm{PCC}$, with 0.3992 versus 0.3669. Stage 2 single-cell pretraining provides a clear additional gain over Stage 1 alone. Under the same downstream fine-tuning protocol, VCLMU-S1-FT obtains 0.1238 Jaccard@20 and 0.3498 $\Delta\mathrm{PCC}$, while VCLMU-S2-FT improves these values to 0.1648 and 0.3992, respectively. The larger relative gain in Jaccard@20 suggests that single-cell pretraining particularly improves recovery of perturbation-specific response structure, while the simultaneous increase in $\Delta\mathrm{PCC}$ shows that this gain does not come at the expense of global transcriptional accuracy. Together, these results indicate that combining population-level perturbation pretraining with subsequent single-cell pretraining improves generalization to unseen perturbations at both the response-gene and whole-expression-profile levels. Complete per-dataset results for Jaccard@20 and $\Delta\mathrm{PCC}$ are reported in Appendix Tables 7 and 8.

Table 3: Inverse perturbation retrieval performance averaged across ten evaluation datasets. The macro means are taken from the benchmark summary, and the reported cross-dataset standard deviations are computed from the ten displayed dataset-level three-seed means. Higher is better for all metrics. The best result is shown in bold and the second best is underlined.

Method	Hit@1	Hit@5	Hit@20	MRR	NDCG@20
CPA-FC	0.0063 ± 0.0059	0.0209 ± 0.0167	0.1117 ± 0.0652	0.0306 ± 0.0160	0.3513 ± 0.0494
GEARS-FC	0.0052 ± 0.0050	0.0299 ± 0.0250	0.1455 ± 0.1030	0.0357 ± 0.0182	0.3547 ± 0.0438
scGPT-FC	$\underline{0.0843 \pm 0.0573}$	$\underline{0.1536 \pm 0.0906}$	$\underline{0.2993 \pm 0.1374}$	$\underline{0.1350 \pm 0.0725}$	$\underline{0.5436 \pm 0.0954}$
Geneformer-FT	0.0031 ± 0.0037	0.0330 ± 0.0193	0.1169 ± 0.0783	0.0314 ± 0.0155	0.3408 ± 0.0549
scFoundation-FT	0.0040 ± 0.0040	0.0377 ± 0.0260	0.1165 ± 0.0848	0.0335 ± 0.0185	0.3600 ± 0.0442
TxPert-FT	0.0153 ± 0.0104	0.0558 ± 0.0383	0.1924 ± 0.1017	0.0518 ± 0.0246	0.3784 ± 0.0532
VCLMU-FT	$\mathbf{0.1645 \pm 0.0827}$	$\mathbf{0.2249 \pm 0.0874}$	$\mathbf{0.3274 \pm 0.0907}$	$\mathbf{0.2043 \pm 0.0819}$	$\mathbf{0.6544 \pm 0.1129}$

4.3 Ablation Study

We further evaluate the contribution of the main VCLMU components in Table 2. Removing any individual component reduces performance relative to the full VCLMU-S2-FT model on both Jaccard@20 and $\Delta\mathrm{PCC}$, indicating that the improvement arises from the combined design rather than a single component. Removing the transition module produces the largest degradation on both metrics, highlighting the importance of explicitly modeling perturbation-induced changes in LMU states. Removing LMU factorization also causes a substantial decrease, supporting the benefit of decomposing cellular state into multiple structured mechanism representations instead of using a single unfactorized latent state. The smaller but consistent drops from removing mechanism dynamics or GRN information further suggest that stochastic state variation and regulatory context provide complementary information. Notably, removing GRN information affects Jaccard@20 more strongly than $\Delta\mathrm{PCC}$, suggesting that regulatory context is particularly useful for recovering the identities of strongly responsive genes. Overall, the ablations support the complementary roles of LMU factorization, transition modeling, mechanism dynamics, and GRN-informed action representations.

4.4 Inverse

We next evaluate whether the perturbation can be recovered from an observed cellular state transition. Table 3 reports macro-average retrieval performance across ten evaluation datasets, while complete per-dataset results are provided in Appendix Table 6. VCLMU-FT achieves the highest macro-average performance across all five retrieval metrics, revealing complementary strengths of the learned representation. Hit@1 measures exact perturbation identification, and the stronger average performance indicates improved discrimination of perturbation identity. Hit@5 and Hit@20 evaluate whether the true perturbation can be narrowed to a small candidate set, where VCLMU also performs best on average, suggesting that its latent representation preserves a more informative neighborhood of plausible actions. MRR is sensitive to the exact rank of the correct perturbation rather than a fixed cutoff, and the higher value indicates that the true action is more consistently placed near the top of the ranking. NDCG@20 further reflects the quality of the top-ranked candidate list, showing that the advantage extends beyond exact recovery to the organization of plausible perturbation candidates. Together, these results indicate that the LMU representation retains fine-grained perturbation-specific information rather than only generic cellular response similarity, supporting recovery of the perturbation most compatible with an observed cellular state transition.

4.5 LMU Interpretability

We next examine whether learned LMUs capture structured perturbation-specific response programs. Across 1,101 held-out perturbations, 86.6% are significantly associated with at least one LMU after filtering ubiquitous response genes and removing the across-perturbation median response. To test whether these associations could arise from selecting the best match among many arbitrary gene modules, we construct 50 random module libraries matched to the learned LMUs in both module number and gene-set size and repeat the same analysis. Only $0.67 \pm 0.27\%$ of perturbations show significant associations under this matched null, while learned LMUs also exhibit substantially greater response-gene overlap and enrichment (Table 4). None of the 50 random

Table 4: Matched random module validation of perturbation-specific LMU associations. Random results are reported as mean ± standard deviation over 50 independently sampled module libraries.

Modules	Significant perturbations	Median overlap	Median enrichment
Random modules	$0.67 \pm 0.27\%$	3.06 ± 0.24	$7.61 \pm 0.60\times$
Learned LMUs	$\mathbf{86.65\%}$	$\mathbf{7.00}$	$\mathbf{19.87\times}$

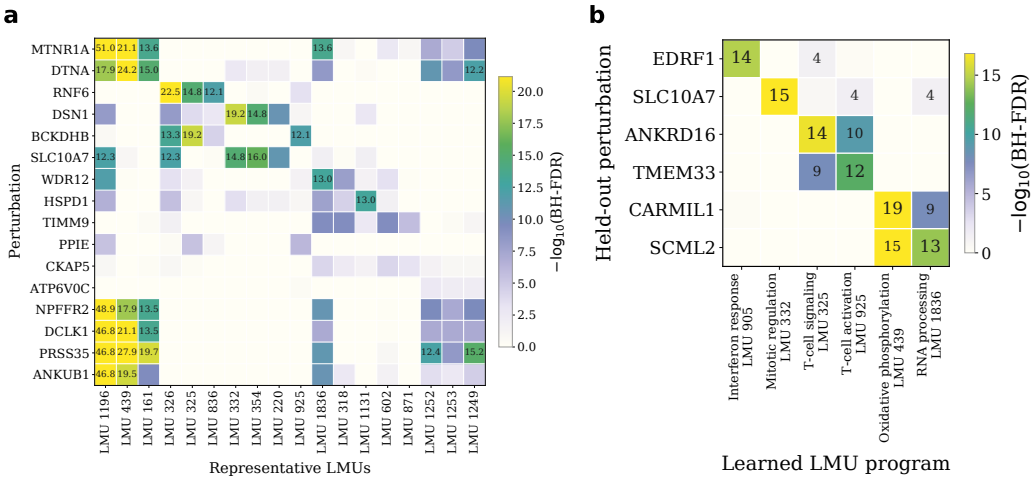


Figure 2: **LMU interpretability. (a)** Perturbation-specific response gene enrichment across learned LMUs for held-out perturbations; color indicates $-\log_{10}(\text{BH-FDR})$. **(b)** Six representative significant perturbation-to-LMU associations with biologically coherent overlapping response genes; numbers indicate the number of overlapping genes.

libraries reaches the learned LMU result on any of these measures, indicating that the observed associations reflect structure learned by VCLMU rather than chance overlap with arbitrary gene groupings. The learned associations further reveal biologically distinct response programs (Fig. 2). Different perturbations preferentially align with different LMUs, and representative associations recover programs related to interferon response, mitotic regulation, T-cell signaling, T-cell activation, oxidative phosphorylation, and RNA processing. For example, the association between EDRF1 and LMU 905 contains *IFI44*, *IFI44L*, *IFI6*, *IFIH1*, *IFIT3*, *ISG15*, *MX1*, *OAS1*, *OAS2*, and *RSAD2*. These genes include canonical interferon-stimulated genes together with OAS and MX antiviral effectors, forming a coherent interferon-response program. Other representative LMUs capture distinct functional axes involving cell-cycle regulation, T-cell signaling and activation, mitochondrial oxidative phosphorylation, and RNA processing, showing that VCLMU organizes perturbation responses into functionally distinguishable latent units.

4.6 Conclusion

We introduced VCLMU, a mechanism-centric virtual cell world model inspired by object-centric world modeling that represents cellular state as a collection of Latent Mechanism Units and explicitly models how genetic perturbations transform their states. By grounding reusable LMU identities in multimodal gene evidence while instantiating cell-specific LMU states from cellular observations, VCLMU provides a structured latent space for perturbation dynamics beyond direct endpoint prediction. The shared LMU state representation supports both forward perturbation response prediction and inverse perturbation inference, while two-stage pretraining combines population-level perturbation structure with single-cell heterogeneity. Across held-out genetic perturbations, VCLMU improves response prediction while learning diverse LMUs aligned with distinct biological response programs. Together, these results support mechanism-structured world modeling as a promising direction for virtual cells that both predict perturbation outcomes and represent the latent biological changes connecting perturbation actions to cellular responses.

AI USE STATEMENT

Generative AI was used only to improve the writing and check consistency in the manuscript. All ideas, experiments, and results were produced by the authors. The authors take full responsibility for the final manuscript and reported results.

REFERENCES

The gene ontology resource: enriching a gold mine. *Nucleic acids research*, 49(D1):D325–D334, 2021.

Britt Adamson, Thomas M Norman, Marco Jost, Min Y Cho, James K Nuñez, Yuwen Chen, Jacqueline E Villalta, Luke A Gilbert, Max A Horlbeck, Marco Y Hein, et al. A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882, 2016.

Abhinav K Adduri, Dhruv Gautam, Beatrice Bevilacqua, Mohsen Naghipourfar, Alishba Imran, Rohan Shah, Noam Teyssier, Rishi Verma, Christopher Carpenter, Basak Eraslan, et al. Predicting cellular responses to perturbation across diverse contexts with state. *Cell*, 2025.

Ding Bai, Le Song, and Eric P Xing. Pertadapt: unlocking single-cell foundation models for genetic perturbation prediction via condition-sensitive adaptation. *Bioinformatics*, 42(Supplement_1): btag307, 2026.

Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. *Advances in Neural Information Processing Systems*, 36:1–12, 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.

Shuizhou Chen, Lang Yu, Kedu Jin, Songming Zhang, Hao Wu, Wenxuan Huang, Sheng Xu, Quan Qian, Qin Chen, Lei Bai, et al. Scale: Scalable conditional atlas-level endpoint transport for virtual cell perturbation prediction. *bioRxiv*, pp. 2026–03, 2026.

Yiqun Chen and James Zou. Genepert: Leveraging genept embeddings for gene perturbation prediction. *bioRxiv*, pp. 2024–10, 2024.

Kevin Chi Hao Ly, Adib Miraki Feriz, Tomoya Isobe, Amirhossein Vahidi, Delshad Vaghari, Anthony Rostron, Mariana Quiroga Londoño, Nicole Mende, MS Vijayabaskar, Marie Moullet, et al. Predicting how perturbations reshape cellular trajectories with perturbgen. *bioRxiv*, pp. 2026–03, 2026.

Hyunghoon Cho, Bonnie Berger, and Jian Peng. Compact integration of multi-network topology for functional analysis of genes. *Cell systems*, 3(6):540–548, 2016.

Guohui Chuai, Xiaohan Chen, Xingbo Yang, Cheng Zhang, Kairu Qu, Yiheng Wang, Wannian Li, Jingya Yang, Duanmiao Si, Feiyang Xing, et al. Towards building a world model to simulate perturbation-induced cellular dynamics by alphacell. *bioRxiv*, pp. 2026–03, 2026.

GTEx Consortium. The gtex consortium atlas of genetic regulatory effects across human tissues. *Science*, 369(6509):1318–1330, 2020.

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.

Tal Daniel, Carl Qi, Dan Haramati, Amir Zadeh, Chuan Li, Aviv Tamar, Deepak Pathak, and David Held. Latent particle world models: Self-supervised object-centric stochastic dynamics modeling. *arXiv preprint arXiv:2603.04553*, 2026.

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

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

Jingcheng Du, Peilin Jia, Yulin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. Gene2vec: distributed representation of genes based on co-expression. *BMC genomics*, 20(Suppl 1):82, 2019.

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

Shreshth Gandhi, Farnoosh Javadi, Valentine Svensson, Umair Khan, Matthew G Jones, John Yu, Daniele Merico, Hani Goodarzi, and Nima Alidoust. Tahoe-x1: Scaling perturbation-trained single-cell foundation models to 3 billion parameters. *bioRxiv*, pp. 2025–10, 2025.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Ellie Haber, Shahul Alam, Nicholas Ho, Renming Liu, Evan Trop, Shaoheng Liang, Muyu Yang, Spencer Krieger, and Jian Ma. Heimdall: A modular framework for tokenization in single-cell foundation models. *bioRxiv*, 2025.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

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.

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

Nicholas Ho, Caleb N Ellington, Jinyu Hou, Sohan Addagudi, Shentong Mo, Tianhua Tao, Dian Li, Yonghao Zhuang, Hongyi Wang, Xingyi Cheng, et al. Scaling dense representations for single cell with transcriptome-scale context. *biorxiv*, pp. 2024–11, 2024.

Ann C Huang, Tsung-Han S Hsieh, Jiang Zhu, Jackson Michuda, Ashton Teng, Soohong Kim, Elizabeth M Rumsey, Sharon K Lam, Ikenna Anigbogu, Philip Wright, et al. X-atlas/orion: Genome-wide perturb-seq datasets via a scalable fix-cryopreserve platform for training dose-dependent biological foundation models. *bioRxiv*, pp. 2025–06, 2025.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo. The molecular signatures database hallmark gene set collection. *Cell systems*, 1(6): 417–425, 2015.

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

Elon Litman, Tyler Myers, Vinayak Agarwal, Ekansh Mittal, Orion Li, Ashwin Gopinath, and Timothy Kassis. Genejepa: A predictive world model of the transcriptome. *bioRxiv*, pp. 2025–10, 2025.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

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

Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian J Theis, and F Alexander Wolf. Conditional out-of-distribution generation for unpaired data using transfer vae. *Bioinformatics*, 36 (Supplement_2):i610–i617, 2020.

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):MSB202211517, 2023.

Marija Milacic, Deidre Beavers, Patrick Conley, Chuqiao Gong, Marc Gillespie, Johannes Griss, Robin Haw, Bijay Jassal, Lisa Matthews, Bruce May, et al. The reactome pathway knowledgebase 2024. *Nucleic acids research*, 52(D1):D672–D678, 2024.

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

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.

Stefan Peidli, Tessa D Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J Schumacher, Jake P Taylor-King, Debora S Marks, et al. scperturb: harmonized single-cell perturbation data. *Nature Methods*, 21(3):531–540, 2024.

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

Mufan Qiu, Genhui Zheng, Yinuo Xu, Ruichen Zhang, Ying Ding, Qi Long, and Tianlong Chen. Chreode: A cell world model for one-step temporal dynamics and perturbation prediction. *arXiv preprint arXiv:2605.28111*, 2026.

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

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

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.

Michael Schubert, Bertram Klinger, Martina Klünemann, Anja Sieber, Florian Uhlitz, Sascha Sauer, Mathew J Garnett, Nils Blüthgen, and Julio Saez-Rodriguez. Perturbation-response genes reveal signaling footprints in cancer gene expression. *Nature communications*, 9(1):20, 2018.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, et al. The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic acids research*, 51(D1):D638–D646, 2023.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

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.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Angel Villar-Corrales and Sven Behnke. Playslot: Learning inverse latent dynamics for controllable object-centric video prediction and planning. *arXiv preprint arXiv:2502.07600*, 2025.

Angel Villar-Corrales, Ismail Wahdan, and Sven Behnke. Object-centric video prediction via decoupling of object dynamics and interactions. In *2023 IEEE International conference on image processing (ICIP)*, pp. 570–574. IEEE, 2023.

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

Xiajie Wei, Jiayi Dong, and Fei Wang. scpregan, a deep generative model for predicting the response of single-cell expression to perturbation. *Bioinformatics*, 38(13):3377–3384, 2022.

Zhijian Wei, Runze Ma, Zichen Wang, Zhongmin Li, Shuotong Song, and Shuangjia Zheng. Vcworld: a biological world model for virtual cell simulation. In *International Conference on Learning Representations*, volume 2026, pp. 72795–72823, 2026.

Frederik Wenkel, Wilson Tu, Cassandra Masschelein, Hamed Shirzad, Liam Hodgson, Ihab Bendidi, Cian Eastwood, Shawn T Whitfield, Craig Russell, Yassir El Mesbahi, et al. Txpert: using multiple knowledge graphs for prediction of transcriptomic perturbation effects. *Nature Biotechnology*, pp. 1–8, 2026.

Zihan Xu, Andras Sziraki, Jasper Lee, Wei Zhou, and Junyue Cao. Dissecting key regulators of transcriptome kinetics through scalable single-cell rna profiling of pooled crispr screens. *Nature Biotechnology*, 42(8):1218–1223, 2024.

Chenglei Yu, Chuanrui Wang, Bangyan Liao, and Tailin Wu. scdfm: Distributional flow matching model for robust single-cell perturbation prediction. *arXiv preprint arXiv:2602.07103*, 2026.

Hengshi Yu, Weizhou Qian, Yuxuan Song, and Joshua D Welch. Perturbnet predicts single-cell responses to unseen chemical and genetic perturbations. *Molecular systems biology*, 21(8):960–982, 2025.

Xinyu Yuan, Xixian Liu, Ya Shi Zhang, Zuobai Zhang, Hongyu Guo, and Jian Tang. Perturbdiff: Functional diffusion for single-cell perturbation modeling. *arXiv preprint arXiv:2602.19685*, 2026.

Yuansong Zeng, Jiancong Xie, Ningyuan Shangguan, Zhuoyi Wei, Wenbing Li, Yun Su, Shuangyu Yang, Chengyang Zhang, Jinbo Zhang, Nan Fang, et al. Cellfm: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells. *Nature Communications*, 16(1):4679, 2025.

Han Zhang, Guo-Hua Yuan, Chaohao Yuan, Tingyang Xu, Tian Bian, Hong Cheng, Wenbing Huang, Deli Zhao, and Yu Rong. Lingshu-cell: A generative cellular world model for transcriptome modeling toward virtual cells. *arXiv preprint arXiv:2603.25240*, 2026a.

Mingxuan Zhang, Vinay Swamy, Rowan Cassius, Léo Dupire, Charilaos Kanatsoulis, Evan Paull, Mohammed AlQuraishi, Theofanis Karaletsos, and Andrea Califano. Gremln: A cellular graph structure aware transcriptomics foundation model. *bioRxiv*, pp. 2025–07, 2026b.

Qi Zhou, Yuan Le, Xiaoning Qi, Hanshuo Lu, Shaole Chang, Yunjia Wu, Haoran Wang, Ruoyao Ran, and Xin Li. Cellos: Learning a world model of cellular state through joint embedding prediction. *bioRxiv*, pp. 2026–06, 2026.

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.

Ronghui Zhu, Emma Dann, Jun Yan, Justine Reyes Retana, Ryunosuke Goto, Reese C Guitche, Lillian Brixi, Mineto Ota, Austin Hartman, Theodore L Roth, et al. Genome-scale perturb-seq in primary human cd4+ t cells maps context-specific regulators of t cell programs and human immune traits. *Cell*, 2025.

A Additional Method Details

A.1 Notation and gene evidence

We describe all quantities for a single sample i consisting of a control reference and its associated perturbed profile. A superscript c denotes the control state, a superscript p denotes the perturbed state, and a hat denotes a model prediction. The prediction and auxiliary losses below are per-sample and are averaged over the mini-batch. LMU identity regularizers average over units or sampled unit pairs, while the activity-entropy regularizer is computed over the mini-batch. The operator Top_n returns the indices of the n largest scores, norm denotes ℓ_2 normalization, sg denotes stop-gradient, and $\odot$ denotes elementwise product. All functions written $f_\bullet$ and all matrices written $W_\bullet$ are learned. The observation encoder is denoted by E_ω, and LN denotes LayerNorm. Brackets $[\cdot;\cdot]$ denote concatenation.

$\mathbf{x}_i^c, \mathbf{x}_i^p \in \mathbb{R}^G$ are the control reference and perturbed profile, a_i is the index of the perturbed gene, and $\Delta\mathbf{x}_i = \mathbf{x}_i^p - \mathbf{x}_i^c$. The model contains K LMUs, of which a sparse subset $\mathcal{A}_i$ of size K_a is active for sample i. Each LMU maintains C candidate genes. Bold lower-case letters denote vectors, and $\mathbf{H}_i^c, \mathbf{H}_i^p, \widehat{\mathbf{H}}_i^p \in \mathbb{R}^{K_a \times d}$ stack the corresponding LMU states. Table 5 summarizes the symbols used throughout.

Table 5: Notation used in the main text and appendix.

Symbol	Meaning
G, g	number of genes, gene index
K, k	number of LMUs in the bank, LMU index
K_a, $\mathcal{A}_i$	number of active LMUs for sample i, their index set
C, $\mathcal{C}_k$	candidate genes per LMU, candidate set of LMU k
d, d_z	model width, latent transition width
$\mathbf{x}_i^c$, $\mathbf{x}_i^p$, $\widehat{\mathbf{x}}_i^p$	control profile, observed perturbed profile, predicted perturbed profile
$\Delta\mathbf{x}_i$, $\widehat{\Delta\mathbf{x}}_i$	observed response $\mathbf{x}_i^p - \mathbf{x}_i^c$, predicted response
a_i, $\boldsymbol{\psi}_i$	perturbed gene (action), action embedding
$\mathcal{G}_L(a_i)$	the L highest-weight GRN neighbors of gene a_i
$\mathbf{e}_g^{(m)}$, $\widetilde{\mathbf{e}}_g^{(m)}$, $\mathbf{e}_g$	raw, projected, and fused multimodal evidence for gene g
$\mathcal{M}_g$, m	available modalities for gene g, modality index
$\mathbf{q}_k$, $\mathbf{q}_k^0$	learnable LMU query, its initialization anchor
$\boldsymbol{\ell}_k$, $\bar{\boldsymbol{\ell}}_k$	current LMU identity, EMA prototype
$\mathbf{o}_i^c$, $\mathbf{o}_i^p$	observation summaries $E_\omega(\mathbf{x}_i^c)$ and $E_\omega(\mathbf{x}_i^p)$
w_{ik}^c, w_{ik}^p	LMU activity weights from $\mathbf{x}_i^c$ and $\mathbf{x}_i^p$
$\mathbf{u}_k$, $\mathbf{b}_k$	pooled candidate-gene features, identity–expression fusion
$\mathbf{h}_{ik}^c$, $\mathbf{h}_{ik}^p$, $\widehat{\mathbf{h}}_{ik}^p$	control, observed perturbed, and predicted perturbed LMU state
$\mathbf{z}_{ik}$, $\boldsymbol{\mu}_{ik}$, $\boldsymbol{\sigma}_{ik}^2$	transition latent, its conditional mean and variance
$\mathbf{Z}_i$	transition latents of all active LMUs, $(\mathbf{z}_{ik})_{k\in\mathcal{A}_i}$
$\mathbf{r}_{ik}$, $\bar{\mathbf{r}}_i$, s_{ig}	per-LMU inverse token, its pooled summary, inverse score of gene g
$\boldsymbol{\delta}_{ik}$, $\Delta\mathbf{h}_{ik}$	raw and FiLM-modulated LMU state update
$\widehat{\mu}_{ig}$, $\widehat{v}_{ig}$	predicted per-gene response mean and variance
N_k	cumulative usage count of LMU k

Let $\mathbf{e}_g^{(m)} \in \mathbb{R}^{d_m}$ be the raw modality-m feature for gene g, and let $\mathcal{M}_g$ be its set of available modalities. We project each available modality to the model width and average the projections:

$$\widetilde{\mathbf{e}}_g^{(m)} = \mathrm{LN}_m\left(W_m \mathbf{e}_g^{(m)}\right), \qquad \mathbf{e}_g = W_e\left(\frac{1}{|\mathcal{M}_g|}\sum_{m\in\mathcal{M}_g}\widetilde{\mathbf{e}}_g^{(m)}\right). \tag{10}$$

All modality projections and W_e are shared across genes. The implementation supports control-expression statistics, control co-expression PCs, Gene Ontology, Hallmark and PROGENy memberships, Gene2vec, STRING-PPI, Reactome, protein-language-model embeddings, and GTEx tissue expression. The evidence module supports per-gene modality masks. Unavailable modalities are

omitted from the average in Eq. 10. Any evidence or GRN feature estimated from perturbational expression is fitted using the training split only.

The shared set encoder used below is a Transformer encoder followed by masked mean pooling and LayerNorm:

$$\mathrm{SetEnc}(\{\mathbf{t}_j\}_{j=1}^n) = \mathrm{LN}\left(\frac{1}{n}\sum_{j=1}^{n}\mathrm{Transformer}(\{W_s\mathbf{t}_j\}_{j=1}^n)_j\right). \tag{11}$$

A.2 LMU IDENTITIES AND CANDIDATE-GENE REFRESH

Each LMU has a learnable query $\mathbf{q}_k$ and an initialization anchor $\mathbf{q}_k^0$ obtained from gene-evidence centroids. The implementation permits deterministic tile means or K-means initialization. The reported training entry point uses deterministic tile means. Given candidate set $\mathcal{C}_k$, its current identity is

$$\boldsymbol{\ell}_k = \mathrm{LN}\left(\alpha\mathbf{q}_k + (1-\alpha)\,\mathrm{SetEnc}\{\mathbf{e}_g : g \in \mathcal{C}_k\}\right), \qquad \alpha = \mathrm{sigmoid}(\alpha_0). \tag{12}$$

For the LMUs selected by routing in each step, an exponential moving average (EMA) prototype is updated by

$$\bar{\boldsymbol{\ell}}_k \leftarrow c_{\mathrm{ema}}\bar{\boldsymbol{\ell}}_k + (1-c_{\mathrm{ema}})\,\mathrm{sg}[\boldsymbol{\ell}_k]. \tag{13}$$

Candidate membership is refreshed periodically along the direction

$$\mathbf{n}_k = \mathrm{norm}\left(\tfrac{1}{2}\,\mathrm{norm}(\mathbf{q}_k) + \tfrac{1}{2}\,\mathrm{norm}(\bar{\boldsymbol{\ell}}_k)\right), \tag{14}$$

giving

$$\mathcal{C}_k = \mathrm{Top}_C\left[\cos(\mathbf{e}_g, \mathbf{n}_k) + c_{\mathrm{loy}}\,\mathbf{1}\{g \in \mathcal{C}_k^{\mathrm{old}}\}\right]_{g=1}^{G}. \tag{15}$$

The retention bonus c_{loy} favors existing candidates and reduces abrupt membership replacement. Candidate sets may overlap, so the model does not impose a hard partition of genes. In distributed training, EMA statistics and usage counts are synchronized before every refresh.

A.3 CONTROL-BASED ROUTING AND STATE CONSTRUCTION

The action-independent observation encoder produces $\mathbf{o}_i^c = E_\omega(\mathbf{x}_i^c)$ for the control reference and $\mathbf{o}_i^p = E_\omega(\mathbf{x}_i^p)$ for the observed perturbed profile. Routing uses the control reference only:

$$\mathcal{A}_i = \mathrm{Top}_{K_a}\left[\cos\big(\mathrm{norm}(W_o\mathbf{o}_i^c), \mathrm{norm}(\bar{\boldsymbol{\ell}}_k)\big) + \frac{c_{\mathrm{ucb}}}{\sqrt{N_k+1}}\right]_{k=1}^{K}, \tag{16}$$

$$w_{ik}^c = \frac{\exp\big(\cos(W_o\mathbf{o}_i^c, \bar{\boldsymbol{\ell}}_k)/\tau\big)}{\sum_{j\in\mathcal{A}_i}\exp\big(\cos(W_o\mathbf{o}_i^c, \bar{\boldsymbol{\ell}}_j)/\tau\big)}, \quad k \in \mathcal{A}_i, \tag{17}$$

where N_k is the cumulative usage count of LMU k and $\tau = 0.01 + 0.99\,\mathrm{sigmoid}(\tau_0)$. The exploration bonus $c_{\mathrm{ucb}}/\sqrt{N_k+1}$ favors rarely used LMUs and affects only the discrete top-K_a selection. Gradients flow through the soft activity weights. For the observed perturbed profile, w_{ik}^p is recomputed from $\mathbf{o}_i^p$ over the fixed set $\mathcal{A}_i$.

Writing $\mathbf{y}$ for either profile, we combine each gene's evidence with its expression and pool over the candidate set:

$$\mathbf{t}_g(\mathbf{y}) = \mathbf{e}_g + \phi(y_g), \qquad \mathbf{u}_k(\mathbf{y}) = \mathrm{SetEnc}\{\mathbf{t}_g(\mathbf{y}) : g \in \mathcal{C}_k\}. \tag{18}$$

Here $\phi : \mathbb{R} \to \mathbb{R}^d$ maps a gene-expression value to an expression embedding. Identity and expression are fused before adding global context through a learned gate:

$$\mathbf{b}_k(\mathbf{y}) = f_{\mathrm{fuse}}\left([\boldsymbol{\ell}_k; \mathbf{u}_k(\mathbf{y})]\right), \tag{19}$$

$$\mathbf{h}_{ik}(\mathbf{y}) = \rho_{ik}(\mathbf{y})\, f_{\mathrm{mix}}\left([\mathbf{b}_k(\mathbf{y}); \eta\, E_\omega(\mathbf{y})]\right), \tag{20}$$

$$\rho_{ik}(\mathbf{y}) = \mathrm{clip}\left(K_a\, w_{ik}(\mathbf{y}),\, 0.25,\, 4\right). \tag{21}$$

Together, f_{fuse}, f_{mix}, and the activity scaling ρ_{ik} constitute the state encoder f_{state} of Eq. 3. The global-context gate is $\eta = \mathrm{sigmoid}(\eta_0)$ with η_0 initialized to -3, so the state initially depends primarily on LMU-specific candidate-gene expression. The same encoder and active indices define $\mathbf{h}_{ik}^c = \mathbf{h}_{ik}(\mathbf{x}_i^c)$ and $\mathbf{h}_{ik}^p = \mathbf{h}_{ik}(\mathbf{x}_i^p)$.

A.4 PERTURBATION ENCODER AND STOCHASTIC TRANSITION

For the configured GRN-aware action encoder, let $\mathcal{G}_L(a_i)$ be the L highest-weight GRN neighbors of perturbed gene a_i. Multi-head attention $\mathrm{MHA}(Q, K, V)$ uses queries Q, keys K, and values V. The action representation is

$$\mathbf{c}_i = \mathrm{MHA}\left(\mathbf{e}_{a_i}, \{\mathbf{e}_g\}_{g\in\mathcal{G}_L(a_i)}, \{\mathbf{e}_g\}_{g\in\mathcal{G}_L(a_i)}\right), \qquad \boldsymbol{\psi}_i = f_{\mathrm{act}}([\mathbf{e}_{a_i}; \mathbf{c}_i]). \tag{22}$$

This implements f_a in the main text. A simpler ablation uses $\boldsymbol{\psi}_i = \mathbf{e}_{a_i}$.

For each active LMU, a conditional Gaussian supplies a transition latent. Its mean and log-variance are produced by f_{prior}, which parameterizes $q(\mathbf{z}_{ik} \mid \mathbf{h}^c_{ik}, \boldsymbol{\psi}_i)$ in Eq. 4. Feature-wise linear modulation (FiLM) then scales and shifts the resulting update according to the control state and action:

$$\left[\boldsymbol{\mu}_{ik}; \log \boldsymbol{\sigma}^2_{ik}\right] = f_{\mathrm{prior}}([\mathbf{h}^c_{ik}; \boldsymbol{\psi}_i]), \tag{23}$$

$$\mathbf{z}_{ik} = \boldsymbol{\mu}_{ik} + \boldsymbol{\sigma}_{ik} \odot \boldsymbol{\epsilon}_{ik}, \qquad \boldsymbol{\epsilon}_{ik} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{24}$$

$$\boldsymbol{\delta}_{ik} = f_{\mathrm{trans}}([\mathbf{h}^c_{ik}; \boldsymbol{\psi}_i; \mathbf{z}_{ik}]), \tag{25}$$

$$[\boldsymbol{\gamma}_{ik}; \boldsymbol{\beta}_{ik}] = f_{\mathrm{film}}([\mathbf{h}^c_{ik}; \boldsymbol{\psi}_i]), \tag{26}$$

$$\Delta\mathbf{h}_{ik} = (1 + \tanh \boldsymbol{\gamma}_{ik}) \odot \boldsymbol{\delta}_{ik} + c_{\mathrm{res}} \tanh \boldsymbol{\beta}_{ik}, \qquad \widehat{\mathbf{h}}^p_{ik} = \mathbf{h}^c_{ik} + \Delta\mathbf{h}_{ik}. \tag{27}$$

These blocks jointly implement F_θ of Eq. 4. The implementation uses $c_{\mathrm{res}} = 0.1$ to limit a deterministic additive bypass around $\mathbf{z}_{ik}$. During training, $\mathbf{z}_{ik}$ is sampled. At validation and forward inference, $\mathbf{z}_{ik} = \boldsymbol{\mu}_{ik}$. The KL term is

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{K_a} \sum_{k\in\mathcal{A}_i} \frac{1}{2} \sum_{j=1}^{d_z} \left[\mu^2_{ikj} + \sigma^2_{ikj} - \log \sigma^2_{ikj} - 1\right]. \tag{28}$$

This is a KL-regularized conditional transition. There is no target-conditioned variational posterior.

A.5 OBSERVATION DECODER AND INVERSE MODEL

A one-layer Transformer first permits interactions among predicted LMUs. Gene evidence is reused as decoder queries through W_q:

$$\widetilde{\mathbf{H}}^p_i = \mathrm{Transformer}_{\mathrm{LMU}}(\widehat{\mathbf{H}}^p_i), \tag{29}$$

$$\mathbf{d}_{ig} = \mathrm{LN}\left(\mathrm{MHA}(W_q\mathbf{e}_g, \widetilde{\mathbf{H}}^p_i, \widetilde{\mathbf{H}}^p_i) + W_q\mathbf{e}_g\right), \tag{30}$$

$$[\widehat{\mu}_{ig}; \log \widehat{v}_{ig}] = f_{\mathrm{gene}}([\mathbf{d}_{ig}; W_q\mathbf{e}_g]). \tag{31}$$

Together these blocks form the decoder D_θ of Eq. 6. For fixed transition latents $\mathbf{Z}_i = (\mathbf{z}_{ik})_{k\in\mathcal{A}_i}$, the decoder defines

$$p(\Delta\mathbf{x}_i \mid \mathbf{x}^c_i, a_i, \mathbf{Z}_i) = \prod_{g=1}^{G} \mathcal{N}(\Delta x_{ig} \mid \widehat{\mu}_{ig}, \widehat{v}_{ig}), \tag{32}$$

where the means and variances depend on $\mathbf{Z}_i$ through the transition and decoder. Averaging this distribution over the transition latents gives $p(\Delta\mathbf{x}_i \mid \mathbf{x}^c_i, a_i)$, which need not be a factorized Gaussian. Training evaluates the following conditional negative log-likelihood at sampled latents, rather than the exact negative log-likelihood of that marginal distribution:

$$\mathcal{L}_{\mathrm{fwd}} = \frac{1}{2G} \sum_{g=1}^{G} \left[\log \widehat{v}_{ig} + \frac{(\Delta x_{ig} - \widehat{\mu}_{ig})^2}{\widehat{v}_{ig}} + \log(2\pi)\right]. \tag{33}$$

The inverse model predicts the perturbed gene from the control and observed perturbed LMU states. It uses both states, their difference, and their elementwise product to form per-LMU tokens

$$\mathbf{r}_{ik} = f_{\mathrm{inv}}\left([\mathbf{h}^c_{ik}; \mathbf{h}^p_{ik}; \mathbf{h}^p_{ik} - \mathbf{h}^c_{ik}; \mathbf{h}^p_{ik} \odot \mathbf{h}^c_{ik}]\right), \tag{34}$$

attention-pools them into $\bar{\mathbf{r}}_i$, and computes

$$s_{ig} = \exp(\xi) \cos(\bar{\mathbf{r}}_i, W_{\mathrm{inv}}\mathbf{e}_g), \qquad \mathcal{L}_{\mathrm{inv}} = \mathrm{CE}(\mathbf{s}_i, a_i), \tag{35}$$

where ξ is a learnable logit scale, $\mathbf{s}_i = (s_{i1}, \ldots, s_{iG})$, and CE denotes cross-entropy. The true perturbation identity a_i is used only as the classification target, not as an input to the inverse head.

A.6 AUXILIARY SUPERVISION AND LMU REGULARIZATION

A decoded response alone does not determine a unique set of LMU changes. The following losses provide additional constraints on the observed states and predicted transitions. Because the active indices are shared, dynamics supervision matches corresponding LMUs:

$$\mathcal{L}_{\mathrm{dyn}} = \mathrm{SmoothL1}\left(\widehat{\mathbf{H}}_i^p - \mathbf{H}_i^c, \mathrm{sg}[\mathbf{H}_i^p - \mathbf{H}_i^c]\right). \tag{36}$$

The stop-gradient blocks derivatives through the observed-change target. The shared encoder can still receive gradients through the predicted change and the other objectives. To retain expression information in the observed perturbed states,

$$\bar{\mathbf{h}}_i^p = \sum_{k \in \mathcal{A}_i} w_{ik}^p \mathbf{h}_{ik}^p, \qquad \mathcal{L}_{\mathrm{obs}} = \mathrm{SmoothL1}\left(W_{\mathrm{out}}(\bar{\mathbf{h}}_i^p), \mathbf{x}_i^p\right). \tag{37}$$

The LMU regularizer is

$$\mathcal{R}_{\mathrm{LMU}} = \lambda_{\mathrm{div}}\mathcal{R}_{\mathrm{div}} + \lambda_{\mathrm{anc}}\mathcal{R}_{\mathrm{anc}} + \lambda_{\mathrm{act}}\mathcal{R}_{\mathrm{act}}, \tag{38}$$

with

$$\mathcal{R}_{\mathrm{div}} = \mathbb{E}_{k \neq l}\left[\max\{0, \cos(\boldsymbol{\ell}_k, \boldsymbol{\ell}_l) - c_{\mathrm{div}}\}^2\right], \tag{39}$$

$$\mathcal{R}_{\mathrm{anc}} = \frac{1}{K}\sum_{k=1}^{K}\left[1 - \cos(\mathbf{q}_k, \mathbf{q}_k^0)\right], \tag{40}$$

$$H_i = -\sum_{k \in \mathcal{A}_i} w_{ik}^c \log(w_{ik}^c + 10^{-8}), \qquad \mathcal{R}_{\mathrm{act}} = \left(\mathbb{E}_i[H_i] - H_\star\right)^2, \tag{41}$$

where $\mathbb{E}_i[H_i]$ is the mini-batch mean of the per-cell activity entropy. The diversity expectation is estimated from sampled pairs of active LMUs using their current identities $\boldsymbol{\ell}_k$. Unlike plain entropy minimization, $\mathcal{R}_{\mathrm{act}}$ targets a moderate entropy $H_\star$ and therefore discourages both one-LMU collapse and uniform activity.

Combining the terms gives the objective of Eq. 9,

$$\mathcal{L} = \mathcal{L}_{\mathrm{fwd}} + \lambda_{\mathrm{dyn}}\mathcal{L}_{\mathrm{dyn}} + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{obs}}\mathcal{L}_{\mathrm{obs}} + \lambda_{\mathrm{inv}}\mathcal{L}_{\mathrm{inv}} + \mathcal{R}_{\mathrm{LMU}}, \tag{42}$$

where λ_{KL} and, when used, λ_{inv} follow linear warm-up schedules that scale them from 0 to their final values, and all per-sample terms are averaged over the mini-batch.

A.7 TWO-STAGE TRAINING AND INFERENCE

We first form perturbation-disjoint data splits before any response-derived preprocessing or model fitting. All quantities estimated from expression data, including response-derived preprocessing, control statistics, and any expression-derived GRN or evidence features, are fitted using training data only. The external biological resources in Appendix C do not depend on the expression data.

Stage 1: pseudo-bulk pretraining. For each perturbed pseudo-bulk profile, $\mathbf{x}_i^c$ is the training control mean for its context, not a measurement of the same cell before perturbation. The model is optimized end-to-end, including evidence projections, set encoder, LMU queries, dynamics, and decoder. Candidate memberships are refreshed periodically from synchronized EMA prototypes. The current configuration jointly uses forward prediction, dynamics alignment, observed-state reconstruction, inverse prediction, KL regularization, and LMU regularization.

Stage 2: single-cell pretraining. The stage-1 checkpoint is restored before recomputing evidence-derived quantities. Single-cell profiles are log-transformed and aligned one-to-one to the stage-1 gene order. Only shared genes contribute to observation losses. The control reference is estimated from training control cells. Evidence projections are then frozen, and the remaining model continues pretraining at a lower learning rate. In the current configuration, $\lambda_{\mathrm{dyn}} = \lambda_{\mathrm{obs}} = 0$, while the forward and inverse objectives remain active.

At forward inference, the observed perturbed profile is absent: the model selects LMUs from $\mathbf{x}_i^c$, encodes a_i, sets $\mathbf{z}_{ik} = \boldsymbol{\mu}_{ik}$, predicts $\widehat{\mathbf{H}}_i^p$, and returns $\widehat{\mathbf{x}}_i^p = \mathbf{x}_i^c + \widehat{\boldsymbol{\mu}}_i$. The observed-state, reconstruction, and inverse branches are not required. Evaluating the nonlinear decoder at the latent means is a deterministic point-prediction rule, not an exact average over transition samples.

Inverse inference instead requires both $\mathbf{x}_i^c$ and $\mathbf{x}_i^p$. It encodes them using the control-selected LMUs and predicts $\widehat{a}_i = \arg\max_g s_{ig}$ from Eq. 35. It shares the observation encoder, LMU state encoder, and gene evidence with forward prediction, but does not run the forward transition network or receive a_i as an input.

B DATA

Pseudo-bulk corpus. Stage-1 training uses approximately 165K pseudo-bulk genetic perturbation profiles constructed from publicly available datasets comprising more than 10 million single-cell perturbation profiles, including X-Atlas-Orion (Huang et al., 2025), Marson (Zhu et al., 2025), and the Virtual Cell Challenge 2025 collection (Roohani et al., 2025). We use only gene perturbation experiments (e.g., CRISPR knockout, CRISPR interference, and CRISPR activation) and exclude chemical perturbation data. Pseudo-bulk profiles are generated by randomly aggregating 200 cells from the same perturbation condition within each cellular context, improving the signal-to-noise ratio while preserving perturbation-specific and context-specific biological responses. Expression profiles are normalized and log-transformed over a shared gene vocabulary. For each perturbed pseudo-bulk profile, the control reference $\mathbf{x}_i^c$ is defined as the mean expression profile of the corresponding training control cells from the same cellular context.

Single-cell data. Stage 2 continues pretraining on gene-aligned single-cell genetic perturbation data derived from the same resources, including X-Atlas-Orion (Huang et al., 2025), Marson (Zhu et al., 2025), and the Virtual Cell Challenge 2025 collection (Roohani et al., 2025). Profiles are library-size normalized and log-transformed. Only genes shared between Stage 1 and Stage 2 contribute to the observation losses. For each dataset, the control reference is estimated as the mean expression profile of the corresponding training control cells.

We evaluate both forward perturbation prediction and inverse perturbation retrieval on ten genetic perturbation datasets: Adamson (Adamson et al., 2016), Norman (Norman et al., 2019), Replogle K562 and RPE1 (Replogle et al., 2022), Tian CRISPRa and CRISPRi (Tian et al., 2021), Xu (Xu et al., 2024), and three Frangieh settings, Control, Coculture, and IFN (Frangieh et al., 2021). All evaluations use perturbation-disjoint splits in which held-out perturbation identities are excluded from downstream training.

Gene evidence. Each gene carries a multimodal evidence embedding built from co-expression, Gene Ontology, MSigDB Hallmark, PROGENy, Gene2Vec, STRING Mashup, Reactome, ESM2, and GTEx representations (Appendix C). Features estimated from perturbational expression are fitted using the training split only.

C BIOLOGICAL EVIDENCE REPRESENTATIONS

To incorporate complementary biological evidence, we augment each gene with a collection of representations capturing transcriptional co-regulation, functional annotation, signaling pathways, interaction networks, protein sequence, and tissue-specific expression. These biological evidence representations combine information from curated biological resources with training-derived expression statistics. All representations are constructed at the human gene level, aligned to the shared gene vocabulary used throughout training, and standardized prior to model training. Each evidence modality m provides the raw feature $\mathbf{e}_g^{(m)}$ used in Eq. 10.

Among the nine representations in Table 9, only the co-expression representation (m_1) is estimated from the training data. All remaining representations are obtained from external biological resources. PCA is used to construct the co-expression representation, while truncated SVD is applied to the Gene Ontology and Reactome annotation matrices to obtain compact embeddings.

D EXPERIMENTAL DETAILS

D.1 PERTURBATION-DISJOINT DATA SPLITS

Following the perturbation-disjoint evaluation setting used in prior perturbation-response benchmarks, we split each evaluation dataset by perturbation identity rather than by individual cells. For each random seed, 25% of the genetic perturbations are held out for testing, and all cells associated with a held-out perturbation are assigned to the test set. The remaining perturbations are used for downstream model development, ensuring that no test perturbation appears as an intervention during downstream model fitting. This setting evaluates generalization to held-out perturbation identities rather than to new cells from perturbations already observed during training. We repeat the perturbation-level split across three random seeds and report performance aggregated over the resulting test sets. For downstream evaluation, VCLMU-FT initializes from the Stage-2 pretrained checkpoint and is further fine-tuned using only the downstream training split of each evaluation dataset. Held-out perturbations are never used during downstream fine-tuning.

D.2 BASELINE IMPLEMENTATION DETAILS

All baselines use the same perturbation-disjoint train, validation, and test splits as VCLMU. For each dataset and split seed, forward methods predict the post-perturbation expression profile for each held-out perturbation, and all forward metrics are computed centrally from the resulting predicted and observed condition-level profiles. For inverse retrieval, every method receives the same candidate perturbation set and produces one score for each candidate gene. Genes that cannot be represented by a model because of checkpoint or vocabulary restrictions are treated as unscoreable.

GenePert. We implement the BaseMLP baseline from GenePert (Chen & Zou, 2024). Each perturbation gene is represented by its fixed 1536-dimensional GenePT text embedding. A two-layer MLP, with architecture $1536 \to 128 \to G$, maps the perturbation representation to an expression response over the evaluation gene space. The model is trained independently for each dataset and split using Adam and MSE, with model selection based on validation-set cosine similarity. The forward prediction is obtained by adding the predicted response to the training control mean.

Geneformer-FT. We use the frozen Geneformer-V2-104M encoder (Theodoris et al., 2023). Cells are rank-value encoded and genes outside the Geneformer vocabulary are excluded. For a candidate perturbation gene, an in-silico perturbation is applied to sampled control cells by removing the corresponding gene token for knockdown perturbations or moving it to the highest-ranked position for activation perturbations. The resulting encoder-space shift is used as the perturbation representation. The Geneformer encoder remains frozen, while a three-layer MLP with hidden dimension 512 is trained to map this shift to the observed expression response using MSE. The head is trained with Adam using learning rate 10^{-3} and weight decay 10^{-4} for up to 300 epochs, with model selection on held-out validation perturbations. Forward prediction is given by the predicted response plus the training control mean.

scFoundation-FT. We use the frozen scFoundation cell encoder (Hao et al., 2024) over its native $19{,}264$-gene panel. Candidate perturbations are simulated directly in the input expression representation, setting the target gene to zero for knockdown perturbations and to the per-cell maximum for activation perturbations. The resulting change in the frozen cell representation defines an in-silico perturbation feature. As with Geneformer-FT, only a three-layer MLP prediction head is trained, using MSE and the same optimization and validation procedure. The scFoundation encoder itself remains frozen.

scDFM. We use scDFM (Yu et al., 2026) as a conditional flow-matching baseline and train it from scratch for each dataset and split. The model uses a 256-dimensional hidden representation with eight attention heads and four Transformer layers. Perturbation identity is provided through the model's gene encoder. Training uses conditional flow matching with Adam and a cosine learning-rate schedule for up to $8{,}000$ optimization steps, with validation every 500 steps and selection of the best validation checkpoint. At inference, the post-perturbation profile is generated by solving the learned flow with a 12-step RK4 ODE solver.

LPM. We train the Large Perturbation Model (LPM) (Miladinovic et al., 2025) independently on each dataset. LPM represents perturbations and readout genes using separate embedding tables and maps them through a two-layer MLP with hidden dimension 512 and embedding dimension 32. We follow the published perturbation-prediction configuration with AdamW, learning rate 0.002, batch size $5{,}000$, and 50 training epochs. At inference, each perturbation is crossed with all readout genes and the resulting predictions are assembled into the full post-perturbation expression profile.

Tahoe-x1-FT. We use the released Tahoe-x1 70M encoder (Gandhi et al., 2025) as a frozen representation model. Cells are represented using the model's native nonzero-gene tokenization and expression binning. For each candidate perturbation, the target gene token is removed from sampled control-cell contexts, and the resulting change in the representations of the remaining genes defines an in-silico perturbation feature. A prediction MLP with hidden dimension 256 is trained on top of the frozen encoder for 10 epochs. The predicted response is added to the control mean to obtain the forward prediction.

CPA. We use the official CPA implementation (Lotfollahi et al., 2023) with a separately trained checkpoint for each dataset and split. At inference, sampled control cells are assigned the candidate perturbation and passed through the perturbation-conditioned CPA decoder. Predicted post-perturbation profiles are averaged across sampled cells to obtain a condition-level prediction.

GEARS. We use the official GEARS implementation (Roohani et al., 2024) with hidden dimension 64 and a separately trained checkpoint for each dataset and split. For each candidate gene, GEARS predicts the corresponding post-perturbation expression profile using its standard graph-conditioned prediction procedure. Predictions are restricted to perturbations represented in the model's gene-graph vocabulary.

scGPT-ZS and scGPT-FT. We use the `TransformerGenerator` architecture from scGPT (Cui et al., 2024). For scGPT-ZS, the encoder, value encoder, and Transformer encoder are initialized from the pretrained `scGPT_human` checkpoint without dataset-specific optimization. For scGPT-FT, the complete perturbation-prediction checkpoint is fine-tuned independently on each downstream training split. For a candidate perturbation, we sample 100 control cells, mark the corresponding gene with the perturbation indicator, predict the resulting profiles, and average the predictions to obtain the condition-level post-perturbation profile.

VCLMU-S1-FT. VCLMU-S1-FT uses the Stage 1 pseudo-bulk pretrained checkpoint and applies the same downstream fine-tuning and evaluation protocol as VCLMU-S2-FT. The comparison therefore isolates the effect of Stage 2 single-cell pretraining while keeping the downstream dataset, split, architecture, and optimization procedure fixed.

D.3 Inverse Baseline Implementation

Inverse perturbation inference is evaluated as retrieval over the candidate perturbation genes available for each dataset and split. For test condition i, each baseline produces a score s_{ig} for every candidate gene g, where larger values indicate that g is more compatible with the observed cellular state transition.

Forward-consistency retrieval. CPA-FC, GEARS-FC, and scGPT-FC reuse the corresponding trained forward models without introducing an additional inverse prediction head. For each candidate perturbation g, the forward model predicts a response

$$\widehat{\Delta \mathbf{x}}_{ig} = \widehat{\mathbf{x}}^{p}_{ig} - \mathbf{x}^{c}_{i}.$$

The candidate is then scored by its agreement with the observed response,

$$s_{ig} = \cos\left(\widehat{\Delta \mathbf{x}}_{ig}, \Delta \mathbf{x}_{i}\right), \qquad \Delta \mathbf{x}_{i} = \mathbf{x}^{p}_{i} - \mathbf{x}^{c}_{i}.$$

Candidates are ranked in descending order of s_{ig}. This procedure evaluates whether the forward model can identify the perturbation whose predicted response is most consistent with the observed state transition.

Geneformer-FT and scFoundation-FT. For Geneformer-FT and scFoundation-FT, candidate-specific in-silico perturbation features are computed using the frozen encoders as described in Appendix D.2. The trained MLP head maps each candidate feature to a predicted expression response. Each candidate is then ranked by cosine similarity between this predicted response and the observed perturbation response, using the same retrieval protocol as the forward-consistency baselines.

TxPert-FT. We use TxPert (Wenkel et al., 2026) with its STRING-graph GATv2 perturbation encoder. The perturbation component is initialized from the released K562 checkpoint, while the dataset-specific control encoder and decoder are initialized for the evaluation gene space. The model is optimized using reconstruction losses for the control and perturbed profiles together with an expression-difference loss weighted by 10. We use AdamW with learning rate 10^{-4} for the graph component and 10^{-3} for the remaining parameters, train for up to 60 epochs, and select the checkpoint with the highest validation cosine similarity. For inverse retrieval, candidate g is scored by the cosine similarity between its predicted expression response and the observed response. Candidates outside the TxPert STRING vocabulary are treated as unscoreable.

VCLMU. VCLMU uses its native inverse head rather than forward-consistency retrieval. The observed control and perturbed profiles are encoded into corresponding LMU states, and their state changes are pooled and scored directly against the evidence representation of every candidate perturbation gene, as described in Section 3.5. Thus, VCLMU predicts all candidate scores in a single inverse pass without running the forward transition separately for every candidate.

Ablation definitions. **w/o LMU** removes LMU factorization and replaces the collection of mechanism states with a single latent state. **w/o Dynamics** removes the stochastic mechanism-dynamics component while retaining the deterministic state-update pathway. **w/o Transition** removes the explicit LMU state-transition module and predicts the response directly from the encoded control state and perturbation representation. **w/o GRN** removes GRN-neighbor information from the perturbation action encoder and uses only the target-gene evidence representation.

D.4 LMU Interpretability Analysis

Perturbation-specific response gene sets. We perform the interpretability analysis on held-out perturbations with at least 10 observations. For each perturbation p, we first compute its mean expression response relative to the corresponding control,

$$\Delta \mathbf{x}_p = \overline{\mathbf{x}}_p^{\mathrm{pert}} - \overline{\mathbf{x}}_p^{\mathrm{ctrl}}.$$

This yields 1,101 held-out perturbations over an original gene universe of $G = 16{,}203$ genes.

Rather than directly using the most differentially responsive genes, we first remove genes that respond ubiquitously across many perturbations. For each perturbation, we identify the 200 genes with the largest absolute response, excluding the directly perturbed gene. We then calculate how frequently each gene appears among these top-response sets across perturbations and remove genes appearing in more than 20% of perturbations. This leaves an effective gene universe of $G_{\mathrm{eff}} = 15{,}894$ genes.

We next subtract the shared gene-wise median response across perturbations. Specifically, for each gene g, we compute

$$c_g = \mathrm{median}_p \, \Delta x_{pg},$$

and define the residual perturbation response as

$$R_{pg} = \Delta x_{pg} - c_g.$$

For each perturbation, the 100 genes with the largest absolute residual response form its perturbation-specific response gene set S_p.

LMU enrichment analysis. Each learned LMU k is associated with a candidate gene set C_k obtained from the pretrained checkpoint. For each held-out perturbation and LMU pair, we calculate the observed overlap

$$O_{pk} = |S_p \cap C_k|.$$

We then test whether this overlap is greater than expected by chance using a one-sided hypergeometric test,

$$p_{pk} = P(X \geq O_{pk}),$$

where

$$X \sim \text{Hypergeom}\left(G_{\text{eff}}, |C_k|, |S_p|\right).$$

For each perturbation independently, the resulting p values across all 2,048 LMUs are corrected using the Benjamini–Hochberg procedure. An association is considered significant when the corrected value satisfies $q_{pk} < 0.05$.

In Fig. 2a, each heatmap entry represents the association between one held-out perturbation and one LMU. Color indicates enrichment significance as $-\log_{10}(\text{BH-FDR})$, while the number inside each cell indicates the actual number of genes shared between the perturbation-specific response set and the corresponding LMU candidate gene set.

Representative biological programs. Fig. 2b uses the same enrichment analysis and highlights six representative significant perturbation-to-LMU associations. The numbers shown in the figure again denote the number of overlapping genes.

The biological labels shown for these examples, including interferon response, mitotic regulation, T-cell signaling, T-cell activation, oxidative phosphorylation, and RNA processing, are assigned after inspecting the overlapping genes for each representative association. These labels are therefore biological interpretations of the observed LMU-response associations rather than predefined LMU identities or pathway annotations.

For example, the overlap between EDRF1 and LMU 905 contains *IFI44*, *IFI44L*, *IFI6*, *IFIH1*, *IFIT3*, *ISG15*, *MX1*, *OAS1*, *OAS2*, and *RSAD2*. These genes strongly overlap with established interferon-response signatures, including the MSigDB Hallmark Interferon Alpha Response and Interferon Gamma Response gene sets, and contain several canonical interferon-stimulated genes such as *ISG15*, *MX1*, and members of the *OAS* family. We therefore annotate this representative association more generally as an interferon-response program. The remaining representative labels are assigned in the same manner based on the functional coherence of their overlapping response genes.

Matched random module validation. To determine whether significant perturbation-to-LMU associations could arise simply from selecting the strongest match among many arbitrary gene modules, we construct 50 independently sampled random module libraries. Each random library contains 2,048 modules, matching the number of learned LMUs, and each random module is sampled from the same filtered gene universe while preserving the size of the corresponding learned LMU candidate set.

For each random library, we repeat the complete enrichment analysis using the same perturbation-specific response sets, including the one-sided hypergeometric test, Benjamini–Hochberg correction across modules, and best-module selection. Across the 50 random libraries, only $0.67 \pm 0.27\%$ of perturbations are significantly associated with at least one random module, compared with 86.65% for the learned LMUs. The median best-module overlap is 3.06 ± 0.24 genes for random modules and 7.00 genes for learned LMUs, while the median enrichment is $7.61 \pm 0.60\times$ and $19.87\times$, respectively.

D.5 Evaluation Metrics

We evaluate perturbation-response predictions independently for each held-out perturbation condition. For a test perturbation p, let $\mathbf{x}_p \in \mathbb{R}^G$ denote the observed condition-level expression profile and $\widehat{\mathbf{x}}_p \in \mathbb{R}^G$ denote the corresponding model prediction over the same G genes. All control reference profiles are constructed exclusively from the training split, while ground-truth differential-expression sets are used only for evaluation. We report two complementary metrics: ΔPCC for global response prediction and Jaccard@20 for recovery of response-gene identities.

Control-referenced perturbation response. We first construct a common control reference using all control cells in the training split. Let $\mathcal{X}_{\text{ctrl}}$ denote the set of training control profiles. The training

control mean is

$$\bar{\mathbf{x}}_{\mathrm{ctrl}} = \frac{1}{|\mathcal{X}_{\mathrm{ctrl}}|} \sum_{\mathbf{x} \in \mathcal{X}_{\mathrm{ctrl}}} \mathbf{x}. \tag{43}$$

For each held-out perturbation p, the observed and predicted perturbation-induced expression changes are defined relative to the same control reference:

$$\Delta \mathbf{x}_p = \mathbf{x}_p - \bar{\mathbf{x}}_{\mathrm{ctrl}}, \qquad \widehat{\Delta \mathbf{x}}_p = \widehat{\mathbf{x}}_p - \bar{\mathbf{x}}_{\mathrm{ctrl}}. \tag{44}$$

Using the same training-derived control reference for both observed and predicted profiles ensures that the evaluation measures agreement in the perturbation-induced response under a common baseline.

Global perturbation-response correlation. We use Pearson correlation across all evaluated genes to measure agreement between the observed and predicted perturbation-response profiles:

$$\Delta\mathrm{PCC}_p = \mathrm{PCC}\left(\Delta \mathbf{x}_p, \widehat{\Delta \mathbf{x}}_p\right). \tag{45}$$

This metric evaluates whether the model reproduces the direction and relative magnitude of gene-expression changes induced by a held-out perturbation across the full transcriptional response.

Top-20 response-gene recovery. ΔPCC evaluates global agreement between the observed and predicted response profiles, but it does not directly measure whether the model identifies the same genes as the strongest responders. We therefore additionally report Jaccard@20. We therefore additionally report Jaccard@20.

Genes are ranked by the absolute magnitude of their predicted perturbation-induced change relative to the training control mean, giving the predicted top-20 response set

$$\widehat{\mathcal{D}}_p = \mathrm{Top}_{20}\left\{\left|\widehat{\Delta x}_{pg}\right|\right\}_{g=1}^{G}. \tag{46}$$

We then compare the predicted response genes with the ground-truth top-20 non-dropout DE genes using the Jaccard index:

$$\mathrm{Jaccard@20}_p = \frac{\left|\mathcal{D}_p \cap \widehat{\mathcal{D}}_p\right|}{\left|\mathcal{D}_p \cup \widehat{\mathcal{D}}_p\right|}. \tag{47}$$

Jaccard@20 is therefore complementary to ΔPCC: the latter measures global agreement between the observed and predicted perturbation-response profiles across all evaluated genes, whereas Jaccard@20 evaluates whether the model can recover the identities of the most responsive genes from its own predicted response.

Aggregation. Both metrics are computed independently for each held-out perturbation condition. Condition-level scores are then averaged across the held-out perturbations within each split, and the evaluation is repeated over the three perturbation-disjoint random splits described above. We report the resulting mean performance across the three splits, with variability summarized by the corresponding standard deviation.

D.6 Inverse Retrieval Metrics

For each test condition i, let r_i denote the one-indexed rank of the true perturbed gene after sorting all candidate genes by decreasing retrieval score, with $r_i = 1$ indicating exact top-ranked recovery. Let N denote the number of test conditions.

Hit@k. Hit@k measures the fraction of test conditions for which the true perturbation appears among the top k candidates:

$$\mathrm{Hit@}\,k = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[r_i \le k].$$

We report $k \in \{1, 5, 20\}$. Because each test condition has exactly one relevant perturbation gene, Hit@k, Recall@k, and the benchmark's single-target Precision@k are equivalent.

Mean reciprocal rank. MRR emphasizes the exact position of the true perturbation near the top of the candidate ranking:

$$\mathrm{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i}.$$

NDCG@k. With one relevant perturbation per test condition, the discounted gain of a successful retrieval at rank r_i is

$$\mathrm{DCG}_i = \frac{1}{\log_2(r_i+1)}.$$

Following the benchmark implementation, NDCG@k is computed over test conditions for which the true perturbation is retrieved within the top k:

$$\mathrm{NDCG@}\,k = \begin{cases} \dfrac{1}{|\mathcal{H}_k|}\displaystyle\sum_{i\in\mathcal{H}_k}\frac{1}{\log_2(r_i+1)}, & |\mathcal{H}_k| > 0, \\ 0, & |\mathcal{H}_k| = 0, \end{cases}$$

where

$$\mathcal{H}_k = \{i : r_i \le k\}.$$

We report NDCG@20 in the main results.

Mean and median rank. As additional ranking diagnostics, we compute

$$\mathrm{MeanRank} = \frac{1}{N}\sum_{i=1}^{N} r_i, \qquad \mathrm{MedianRank} = \mathrm{median}_i(r_i),$$

where lower values indicate that the true perturbation is ranked closer to the top of the candidate list.

Chance retrieval rate. For a candidate set containing C perturbation genes, the expected Hit@1 of uniform random ranking is

$$\mathrm{ChanceHit@1} = \frac{1}{C}.$$

We use this quantity as a dataset-specific reference floor because the number of candidate perturbations varies across datasets.

All inverse metrics are first computed independently for each dataset and split seed. Per-dataset results are reported as the mean and standard deviation across three seeds. Macro-average results in the main text are then obtained by averaging the displayed dataset-level means across the ten evaluation datasets.

Table 6: Per-dataset inverse perturbation retrieval performance. Results are reported as mean ± standard deviation over three seeds. Higher is better for all metrics. The best result is shown in bold and the second best is underlined.

Dataset	Method	Hit@1 ↑	Hit@5 ↑	Hit@20 ↑	MRR ↑	NDCG@20 ↑
Adamson	CPA-FC	0.0000 ± 0.0000	0.0000 ± 0.0000	0.2063 ± 0.0594	0.0379 ± 0.0078	0.2741 ± 0.0177
	GEARS-FC	0.0000 ± 0.0000	0.0317 ± 0.0224	0.3492 ± 0.0594	0.0574 ± 0.0098	0.2937 ± 0.0240
	scGPT-FC	0.0794 ± 0.0224	**0.2222 ± 0.0224**	**0.4921 ± 0.0809**	0.1835 ± 0.0031	0.4948 ± 0.0563
	Geneformer-FT	0.0000 ± 0.0000	0.0635 ± 0.0449	0.2381 ± 0.0778	0.0535 ± 0.0152	0.3284 ± 0.0000
	scFoundation-FT	0.0000 ± 0.0000	0.0952 ± 0.0389	0.2698 ± 0.0224	0.0642 ± 0.0109	0.3595 ± 0.0000
	TxPert-FT	0.0317 ± 0.0449	0.1587 ± 0.0809	0.4127 ± 0.0449	0.1084 ± 0.0386	0.4063 ± 0.0885
	VCLMU-FT	**0.1429 ± 0.0389**	0.1905 ± 0.0389	0.4127 ± 0.0594	**0.1909 ± 0.0355**	**0.5489 ± 0.0512**
Fr-Ctrl	CPA-FC	0.0000 ± 0.0000	0.0079 ± 0.0112	0.1032 ± 0.0625	0.0226 ± 0.0053	0.2870 ± 0.0332
	GEARS-FC	0.0079 ± 0.0112	0.0159 ± 0.0112	0.0873 ± 0.0405	0.0318 ± 0.0123	0.3564 ± 0.0477
	scGPT-FC	0.1270 ± 0.0297	0.1746 ± 0.0224	0.3095 ± 0.0194	0.1638 ± 0.0217	0.6019 ± 0.0771
	Geneformer-FT	0.0079 ± 0.0112	0.0397 ± 0.0112	0.0873 ± 0.0112	0.0354 ± 0.0102	0.4098 ± 0.0000
	scFoundation-FT	0.0079 ± 0.0112	0.0397 ± 0.0112	0.0873 ± 0.0112	0.0354 ± 0.0102	0.4098 ± 0.0000
	TxPert-FT	0.0000 ± 0.0000	0.0238 ± 0.0194	0.1190 ± 0.0389	0.0272 ± 0.0073	0.2943 ± 0.0336
	VCLMU-FT	**0.2222 ± 0.0405**	**0.2778 ± 0.0112**	**0.3730 ± 0.0224**	**0.2574 ± 0.0280**	**0.7427 ± 0.0853**
Fr-Cocu	CPA-FC	0.0079 ± 0.0112	0.0159 ± 0.0112	0.0794 ± 0.0405	0.0278 ± 0.0126	0.3498 ± 0.0713
	GEARS-FC	0.0159 ± 0.0112	0.0476 ± 0.0194	0.1111 ± 0.0112	0.0417 ± 0.0109	0.4228 ± 0.0717
	scGPT-FC	0.1905 ± 0.0891	**0.3095 ± 0.1010**	**0.4762 ± 0.1010**	**0.2647 ± 0.0874**	0.6241 ± 0.0579
	Geneformer-FT	0.0079 ± 0.0112	0.0397 ± 0.0112	0.0952 ± 0.0000	0.0367 ± 0.0087	0.4080 ± 0.0000
	scFoundation-FT	0.0079 ± 0.0112	0.0397 ± 0.0112	0.0873 ± 0.0112	0.0375 ± 0.0020	0.4419 ± 0.0000
	TxPert-FT	0.0238 ± 0.0000	0.0556 ± 0.0224	0.1349 ± 0.0594	0.0527 ± 0.0083	0.4632 ± 0.0526
	VCLMU-FT	**0.2063 ± 0.1323**	0.2937 ± 0.1265	0.4206 ± 0.1071	0.2601 ± 0.1207	**0.6621 ± 0.1744**
Fr-IFN	CPA-FC	0.0079 ± 0.0112	0.0397 ± 0.0297	0.1111 ± 0.0112	0.0340 ± 0.0109	0.3581 ± 0.0768
	GEARS-FC	0.0000 ± 0.0000	0.0397 ± 0.0112	0.1111 ± 0.0405	0.0312 ± 0.0082	0.3494 ± 0.0293
	scGPT-FC	0.1190 ± 0.0514	0.2143 ± 0.0583	0.3730 ± 0.0786	0.1754 ± 0.0534	0.5514 ± 0.0428
	Geneformer-FT	0.0079 ± 0.0112	0.0397 ± 0.0112	0.0873 ± 0.0112	0.0354 ± 0.0102	0.4098 ± 0.0000
	scFoundation-FT	0.0079 ± 0.0112	0.0397 ± 0.0112	0.0873 ± 0.0112	0.0354 ± 0.0102	0.4098 ± 0.0000
	TxPert-FT	0.0079 ± 0.0112	0.0397 ± 0.0112	0.1825 ± 0.0297	0.0429 ± 0.0146	0.3405 ± 0.0507
	VCLMU-FT	**0.3651 ± 0.0297**	**0.3968 ± 0.0405**	**0.4603 ± 0.0594**	**0.3905 ± 0.0332**	**0.8741 ± 0.0423**
Norman	CPA-FC	0.0185 ± 0.0131	0.0458 ± 0.0249	0.1920 ± 0.1148	0.0557 ± 0.0243	0.3893 ± 0.0802
	GEARS-FC	0.0090 ± 0.0127	0.0921 ± 0.0569	**0.2581 ± 0.0895**	0.0693 ± 0.0176	0.3592 ± 0.0337
	scGPT-FC	0.0000 ± 0.0000	0.0281 ± 0.0233	0.2530 ± 0.1326	0.0468 ± 0.0086	0.3170 ± 0.0520
	Geneformer-FT	0.0000 ± 0.0000	0.0564 ± 0.0410	0.1858 ± 0.0369	0.0469 ± 0.0128	0.3417 ± 0.0000
	scFoundation-FT	0.0093 ± 0.0131	0.0553 ± 0.0601	0.2143 ± 0.1371	0.0554 ± 0.0345	0.3414 ± 0.0000
	TxPert-FT	0.0095 ± 0.0135	0.0561 ± 0.0240	0.2514 ± 0.0872	0.0566 ± 0.0224	0.3376 ± 0.0424
	VCLMU-FT	**0.1138 ± 0.1237**	**0.1421 ± 0.1460**	0.2344 ± 0.1393	**0.1481 ± 0.1279**	**0.5426 ± 0.1951**
K562	CPA-FC	0.0000 ± 0.0000	0.0022 ± 0.0018	0.0140 ± 0.0055	0.0041 ± 0.0007	0.3000 ± 0.0144
	GEARS-FC	0.0000 ± 0.0000	0.0022 ± 0.0018	0.0081 ± 0.0010	0.0039 ± 0.0006	0.3372 ± 0.0503
	scGPT-FC	0.0250 ± 0.0092	0.0338 ± 0.0105	0.0609 ± 0.0105	0.0354 ± 0.0078	0.6126 ± 0.0750
	Geneformer-FT	0.0000 ± 0.0000	0.0029 ± 0.0010	0.0132 ± 0.0018	0.0042 ± 0.0004	0.3202 ± 0.0000
	scFoundation-FT	0.0000 ± 0.0000	0.0029 ± 0.0010	0.0132 ± 0.0018	0.0042 ± 0.0004	0.3202 ± 0.0000
	TxPert-FT	0.0162 ± 0.0058	0.0441 ± 0.0062	0.1116 ± 0.0122	0.0372 ± 0.0062	0.4362 ± 0.0291
	VCLMU-FT	**0.1681 ± 0.0535**	**0.2254 ± 0.0682**	**0.2779 ± 0.0720**	**0.1985 ± 0.0604**	**0.7580 ± 0.0387**
RPE1	CPA-FC	0.0009 ± 0.0013	0.0019 ± 0.0013	0.0085 ± 0.0023	0.0052 ± 0.0009	0.4462 ± 0.1666
	GEARS-FC	0.0057 ± 0.0040	0.0085 ± 0.0040	0.0283 ± 0.0080	0.0119 ± 0.0044	0.4175 ± 0.0735
	scGPT-FC	0.0274 ± 0.0071	0.0368 ± 0.0129	0.0680 ± 0.0160	0.0387 ± 0.0087	**0.6086 ± 0.0415**
	Geneformer-FT	0.0009 ± 0.0013	0.0057 ± 0.0061	0.0208 ± 0.0053	0.0066 ± 0.0031	0.3258 ± 0.0000
	scFoundation-FT	0.0000 ± 0.0000	0.0019 ± 0.0013	0.0113 ± 0.0000	0.0046 ± 0.0001	0.3022 ± 0.0000
	TxPert-FT	0.0104 ± 0.0027	0.0255 ± 0.0040	0.0774 ± 0.0107	0.0239 ± 0.0024	0.4126 ± 0.0212
	VCLMU-FT	**0.0674 ± 0.0453**	**0.1015 ± 0.0673**	**0.1441 ± 0.0826**	**0.0865 ± 0.0538**	0.6067 ± 0.1234
CRISPRa	CPA-FC	0.0133 ± 0.0189	0.0400 ± 0.0000	0.1867 ± 0.0189	0.0502 ± 0.0136	0.3585 ± 0.0512
	GEARS-FC	0.0000 ± 0.0000	0.0133 ± 0.0189	0.2533 ± 0.0754	0.0428 ± 0.0113	0.2809 ± 0.0172
	scGPT-FC	0.0400 ± 0.0000	**0.1200 ± 0.0566**	**0.2933 ± 0.0499**	0.1000 ± 0.0150	0.4327 ± 0.0499
	Geneformer-FT	0.0000 ± 0.0000	0.0400 ± 0.0327	0.2533 ± 0.0189	0.0448 ± 0.0054	0.2993 ± 0.0000
	scFoundation-FT	0.0000 ± 0.0000	0.0533 ± 0.0499	0.2267 ± 0.0189	0.0468 ± 0.0094	0.3126 ± 0.0000
	TxPert-FT	0.0133 ± 0.0189	0.0533 ± 0.0499	**0.2933 ± 0.0499**	0.0629 ± 0.0226	0.3475 ± 0.0852
	VCLMU-FT	**0.0667 ± 0.0499**	**0.1200 ± 0.0327**	**0.2933 ± 0.0499**	**0.1160 ± 0.0421**	**0.4772 ± 0.0858**
CRISPRi	CPA-FC	0.0072 ± 0.0102	0.0290 ± 0.0102	0.1159 ± 0.0271	0.0374 ± 0.0116	0.3736 ± 0.0443
	GEARS-FC	0.0072 ± 0.0102	0.0145 ± 0.0102	0.1014 ± 0.0102	0.0314 ± 0.0112	0.3442 ± 0.0771
	scGPT-FC	0.1014 ± 0.0410	0.2029 ± 0.0271	**0.3333 ± 0.0623**	0.1623 ± 0.0321	0.5801 ± 0.0849
	Geneformer-FT	0.0000 ± 0.0000	0.0290 ± 0.0102	0.0942 ± 0.0271	0.0259 ± 0.0013	0.3406 ± 0.0000
	scFoundation-FT	0.0000 ± 0.0000	0.0290 ± 0.0102	0.0942 ± 0.0271	0.0259 ± 0.0013	0.3406 ± 0.0000
	TxPert-FT	0.0072 ± 0.0102	0.0217 ± 0.0177	0.0942 ± 0.0271	0.0307 ± 0.0121	0.3247 ± 0.0761
	VCLMU-FT	**0.1325 ± 0.0493**	**0.2079 ± 0.0550**	0.3111 ± 0.0196	**0.1723 ± 0.0450**	**0.6265 ± 0.1159**
Xu	CPA-FC	0.0067 ± 0.0094	0.0267 ± 0.0094	0.1000 ± 0.0589	0.0309 ± 0.0111	0.3765 ± 0.0427
	GEARS-FC	0.0067 ± 0.0094	0.0333 ± 0.0094	0.1467 ± 0.0471	0.0357 ± 0.0074	0.3856 ± 0.1209
	scGPT-FC	0.1333 ± 0.0094	0.1933 ± 0.0189	0.3333 ± 0.0340	0.1792 ± 0.0126	0.6129 ± 0.0554
	Geneformer-FT	0.0067 ± 0.0094	0.0133 ± 0.0189	0.0933 ± 0.0680	0.0249 ± 0.0119	0.2242 ± 0.0000
	scFoundation-FT	0.0067 ± 0.0094	0.0200 ± 0.0163	0.0733 ± 0.0411	0.0257 ± 0.0109	0.3619 ± 0.0000
	TxPert-FT	0.0333 ± 0.0189	0.0800 ± 0.0000	0.2467 ± 0.0094	0.0755 ± 0.0102	0.4207 ± 0.0354
	VCLMU-FT	**0.1600 ± 0.0163**	**0.2933 ± 0.0411**	**0.3467 ± 0.0249**	**0.2228 ± 0.0179**	**0.7049 ± 0.0080**

Table 7: Per-dataset Jaccard@20 performance for forward perturbation prediction. Results are reported as mean ± standard deviation over three seeds. Higher is better. The best result is shown in bold and the second best is underlined. Scouter results are available for nine of the ten datasets.

(a) Adamson, Fr-Ctrl, Fr-Cocu, Fr-IFN, and Norman

Method	Adamson	Fr-Ctrl	Fr-Cocu	Fr-IFN	Norman
GenePert	0.0613 ± 0.0265	0.0055 ± 0.0121	0.0023 ± 0.0087	0.0068 ± 0.0165	0.0360 ± 0.0327
Geneformer-FT	0.2525 ± 0.1020	0.0074 ± 0.0139	0.0031 ± 0.0102	0.0150 ± 0.0328	0.1217 ± 0.0848
scFoundation-FT	0.2525 ± 0.1020	0.0074 ± 0.0139	0.0033 ± 0.0115	0.0150 ± 0.0328	0.1336 ± 0.0849
Scouter	0.2201 ± 0.1095	0.0039 ± 0.0098	0.0309 ± 0.1397	0.0305 ± 0.1252	0.1207 ± 0.0955
scDFM	0.0330 ± 0.0251	0.0028 ± 0.0081	0.0028 ± 0.0081	0.0033 ± 0.0086	0.0040 ± 0.0094
LPM	0.0750 ± 0.0305	0.0082 ± 0.0161	0.0018 ± 0.0066	0.0130 ± 0.0249	0.0433 ± 0.0347
Tahoe-x1-FT	0.0335 ± 0.0274	0.0093 ± 0.0160	0.0035 ± 0.0094	0.0093 ± 0.0168	0.0450 ± 0.0293
CPA	0.1650 ± 0.0870	0.0060 ± 0.0160	0.0080 ± 0.0190	0.0110 ± 0.0210	0.0470 ± 0.0490
GEARS	0.2090 ± 0.0670	0.0190 ± 0.0240	0.0210 ± 0.0650	0.0350 ± 0.0900	0.1180 ± 0.0790
scGPT-ZS	0.1790 ± 0.0950	0.0060 ± 0.0160	0.0080 ± 0.0200	0.0100 ± 0.0210	0.0460 ± 0.0480
scGPT-FT	0.1520 ± 0.0760	0.0110 ± 0.0160	0.0130 ± 0.0150	0.0120 ± 0.0180	0.0670 ± 0.0470
VCLMU-S1-FT	0.1370 ± 0.0080	0.0060 ± 0.0020	0.0060 ± 0.0030	0.0050 ± 0.0030	0.0860 ± 0.0060
VCLMU w/o LMU	0.1730 ± 0.0020	0.0080 ± 0.0010	0.0190 ± 0.0040	0.0090 ± 0.0040	0.1250 ± 0.0060
VCLMU w/o Dynamics	0.1940 ± 0.0040	0.0080 ± 0.0030	0.0210 ± 0.0030	0.0050 ± 0.0030	0.1380 ± 0.0070
VCLMU w/o Transition	0.1870 ± 0.0070	0.0060 ± 0.0010	0.0170 ± 0.0010	0.0020 ± 0.0050	0.1240 ± 0.0030
VCLMU w/o GRN	0.2010 ± 0.0040	0.0060 ± 0.0010	0.0200 ± 0.0030	0.0040 ± 0.0030	0.1340 ± 0.0080
VCLMU-S2-FT	0.2170 ± 0.0170	0.0080 ± 0.0020	0.0240 ± 0.0020	0.0160 ± 0.0010	0.1420 ± 0.1030

(b) K562, RPE1, CRISPRa, CRISPRi, and Xu

Method	K562	RPE1	CRISPRa	CRISPRi	Xu
GenePert	0.0153 ± 0.0218	0.0084 ± 0.0161	0.0125 ± 0.0190	0.0035 ± 0.0094	0.0036 ± 0.0095
Geneformer-FT	0.0272 ± 0.0327	0.0315 ± 0.0522	0.0882 ± 0.0835	0.0294 ± 0.0372	0.0465 ± 0.0456
scFoundation-FT	0.0270 ± 0.0324	0.0321 ± 0.0527	0.0857 ± 0.0829	0.0306 ± 0.0374	0.0464 ± 0.0441
Scouter	0.0587 ± 0.0734	0.0456 ± 0.0624	0.0428 ± 0.0575	0.0163 ± 0.0231	–
scDFM	0.0184 ± 0.0266	0.0138 ± 0.0232	0.0028 ± 0.0112	0.0026 ± 0.0090	0.0079 ± 0.0137
LPM	0.0354 ± 0.0395	0.0191 ± 0.0293	0.0356 ± 0.0456	0.0090 ± 0.0163	0.0469 ± 0.0466
Tahoe-x1-FT	0.0342 ± 0.0393	0.0184 ± 0.0289	0.0150 ± 0.0228	0.0073 ± 0.0144	0.0176 ± 0.0334
CPA	0.0560 ± 0.0630	0.0270 ± 0.0380	0.0240 ± 0.0340	0.0080 ± 0.0160	0.0450 ± 0.0440
GEARS	0.0240 ± 0.0280	0.0340 ± 0.0470	0.0550 ± 0.0640	0.0110 ± 0.0170	0.0320 ± 0.0320
scGPT-ZS	0.0560 ± 0.0630	0.0280 ± 0.0380	0.0160 ± 0.0290	0.0090 ± 0.0160	0.0510 ± 0.0510
scGPT-FT	0.0460 ± 0.0570	0.0380 ± 0.0480	0.0180 ± 0.0240	0.0090 ± 0.0170	0.0400 ± 0.0360
VCLMU-S1-FT	0.2730 ± 0.0190	0.3540 ± 0.0130	0.0510 ± 0.0120	0.2880 ± 0.0540	0.0320 ± 0.0050
VCLMU w/o LMU	0.3320 ± 0.0030	0.3380 ± 0.0170	0.0930 ± 0.0110	0.2510 ± 0.0280	0.0400 ± 0.0030
VCLMU w/o Dynamics	0.3470 ± 0.0070	0.3680 ± 0.0150	0.0880 ± 0.0090	0.2760 ± 0.0330	0.0410 ± 0.0220
VCLMU w/o Transition	0.3110 ± 0.0020	0.3550 ± 0.0090	0.1060 ± 0.0050	0.2210 ± 0.0520	0.0330 ± 0.0010
VCLMU w/o GRN	0.3050 ± 0.0010	0.3640 ± 0.0100	0.1150 ± 0.0080	0.2830 ± 0.0260	0.0430 ± 0.0090
VCLMU-S2-FT	0.3610 ± 0.0090	0.4140 ± 0.0100	0.1260 ± 0.0200	0.2930 ± 0.0420	0.0470 ± 0.0040

Table 8: Per-dataset ΔPCC performance for forward perturbation prediction. Results are reported as mean ± standard deviation over three seeds. Higher is better. The best result is shown in bold and the second best is underlined.

(a) Adamson, Fr-Ctrl, Fr-Cocu, Fr-IFN, and Norman

Method	Adamson	Fr-Ctrl	Fr-Cocu	Fr-IFN	Norman
GenePert	0.0165 ± 0.0226	0.0023 ± 0.0147	-0.0016 ± 0.0140	0.0051 ± 0.0143	0.0174 ± 0.0186
Geneformer-FT	0.5975 ± 0.4517	0.1571 ± 0.1487	$\underline{0.1519 \pm 0.1401}$	0.1238 ± 0.1655	0.4888 ± 0.2504
scFoundation-FT	0.5975 ± 0.4520	0.1571 ± 0.1487	$\underline{0.1519 \pm 0.1402}$	0.1238 ± 0.1655	0.4710 ± 0.3062
scDFM	-0.0118 ± 0.0312	0.0249 ± 0.0227	0.0225 ± 0.0253	0.0229 ± 0.0278	0.0266 ± 0.0218
LPM	-0.0229 ± 0.0425	0.1374 ± 0.1673	0.1438 ± 0.1564	0.1080 ± 0.1869	-0.0252 ± 0.0373
Tahoe-x1-FT	0.1846 ± 0.0938	0.0796 ± 0.1038	0.0496 ± 0.0785	0.0530 ± 0.1101	0.0557 ± 0.0310
CPA	0.0434 ± 0.0755	0.0827 ± 0.0747	0.0849 ± 0.0790	0.0747 ± 0.0868	0.1205 ± 0.0769
GEARS	0.6530 ± 0.3281	0.0558 ± 0.0723	0.0620 ± 0.1142	0.0374 ± 0.1418	0.4473 ± 0.2196
scGPT-ZS	0.0485 ± 0.0791	0.0881 ± 0.0793	0.0902 ± 0.0834	0.0791 ± 0.0915	0.1290 ± 0.0824
scGPT-FT	0.6210 ± 0.3070	0.0370 ± 0.0530	0.0660 ± 0.0580	0.0410 ± 0.0620	0.2840 ± 0.1910
VCLMU-S1-FT	0.6780 ± 0.0520	0.1720 ± 0.0070	0.1030 ± 0.0150	0.1380 ± 0.0040	0.5080 ± 0.0370
VCLMU w/o LMU	0.6930 ± 0.0540	$\underline{0.1730 \pm 0.0240}$	0.1500 ± 0.0320	0.1260 ± 0.0050	0.5120 ± 0.0040
VCLMU w/o Dynamics	0.6640 ± 0.0390	0.1680 ± 0.0620	0.1410 ± 0.0160	0.1470 ± 0.0050	0.5360 ± 0.0020
VCLMU w/o Transition	0.6320 ± 0.0560	0.1660 ± 0.0250	0.0080 ± 0.0800	$\underline{0.1570 \pm 0.0030}$	0.5330 ± 0.0060
VCLMU w/o GRN	$\underline{0.6940 \pm 0.0590}$	0.1670 ± 0.0290	0.1490 ± 0.0330	0.1490 ± 0.0080	$\mathbf{0.5430 \pm 0.0340}$
VCLMU-S2-FT	$\mathbf{0.7040 \pm 0.0500}$	$\mathbf{0.1770 \pm 0.0090}$	$\mathbf{0.1540 \pm 0.0190}$	$\mathbf{0.1590 \pm 0.0060}$	$\underline{0.5370 \pm 0.0400}$

(b) K562, RPE1, CRISPRa, CRISPRi, and Xu

Method	K562	RPE1	CRISPRa	CRISPRi	Xu
GenePert	0.0012 ± 0.0132	-0.0017 ± 0.0123	-0.0063 ± 0.0152	0.0029 ± 0.0168	0.0002 ± 0.0279
Geneformer-FT	0.3132 ± 0.1630	$\underline{0.5369 \pm 0.3247}$	0.5021 ± 0.1862	0.4943 ± 0.1315	0.3034 ± 0.1191
scFoundation-FT	$\underline{0.3141 \pm 0.1621}$	0.5367 ± 0.3249	0.4998 ± 0.1981	0.4974 ± 0.1237	0.3053 ± 0.1229
scDFM	0.0356 ± 0.0340	0.0453 ± 0.0332	0.0373 ± 0.0308	0.0158 ± 0.0323	0.0076 ± 0.0344
LPM	0.0089 ± 0.0254	-0.0234 ± 0.0294	0.0159 ± 0.0451	-0.0396 ± 0.0570	-0.0352 ± 0.1324
Tahoe-x1-FT	0.0104 ± 0.0259	-0.0085 ± 0.0584	0.0473 ± 0.0270	0.0240 ± 0.0386	-0.0018 ± 0.0640
CPA	0.1187 ± 0.1057	0.1113 ± 0.0796	0.1862 ± 0.1016	0.0497 ± 0.1118	0.0220 ± 0.0737
GEARS	0.2122 ± 0.1130	0.5006 ± 0.2941	0.3030 ± 0.1315	0.2309 ± 0.0494	0.1457 ± 0.0927
scGPT-ZS	0.1247 ± 0.1108	0.1171 ± 0.0837	0.2115 ± 0.1132	0.0551 ± 0.1234	0.0369 ± 0.1130
scGPT-FT	0.1750 ± 0.1540	0.5210 ± 0.2910	0.2760 ± 0.1110	0.2350 ± 0.0530	0.1040 ± 0.1830
VCLMU-S1-FT	0.1770 ± 0.0120	0.4700 ± 0.3130	0.5080 ± 0.0230	$\underline{0.5170 \pm 0.0370}$	0.2270 ± 0.0030
VCLMU w/o LMU	0.1940 ± 0.1450	0.4890 ± 0.3240	0.4880 ± 0.1400	0.4340 ± 0.0250	0.2770 ± 0.0140
VCLMU w/o Dynamics	0.3010 ± 0.1260	0.4250 ± 0.3620	0.4280 ± 0.0360	0.4920 ± 0.0420	$\underline{0.3150 \pm 0.0150}$
VCLMU w/o Transition	0.2750 ± 0.1760	0.4430 ± 0.3390	0.5050 ± 0.0040	0.4230 ± 0.0550	0.2800 ± 0.0330
VCLMU w/o GRN	0.2940 ± 0.1240	0.4700 ± 0.2450	$\underline{0.5230 \pm 0.0280}$	0.5130 ± 0.0080	0.3010 ± 0.0340
VCLMU-S2-FT	$\mathbf{0.3330 \pm 0.1350}$	$\mathbf{0.5420 \pm 0.3250}$	$\mathbf{0.5450 \pm 0.0350}$	$\mathbf{0.5190 \pm 0.0440}$	$\mathbf{0.3220 \pm 0.0130}$

Table 9: Biological evidence representations used to construct the multimodal gene evidence embedding.

Evidence	Source	Description	Dim.
m_1	Training co-expression	Principal components of the training-set gene–gene co-expression matrix, capturing major transcriptional covariance patterns.	≤ 256
m_2	Gene Ontology (gen, 2021)	Gene–term association matrix compressed using truncated SVD to summarize biological process, molecular function, and cellular component annotations.	512
m_3	MSigDB Hallmark (Liberzon et al., 2015)	Binary membership in the Hallmark gene sets, representing approximately 50 curated biological programs.	~ 50
m_4	PROGENy (Schubert et al., 2018)	Association with the 14 PROGENy signaling pathways, providing pathway-level functional information.	14
m_5	Gene2Vec (Du et al., 2019)	Pretrained gene embeddings capturing functional similarity learned from gene co-expression.	200
m_6	STRING + Mashup (Szklarczyk et al., 2023; Cho et al., 2016)	Protein interaction network embedding capturing both local and global network topology.	800
m_7	Reactome (Milacic et al., 2024)	Gene–pathway association matrix compressed using truncated SVD to summarize Reactome pathway memberships.	200
m_8	ESM2-3B (Lin et al., 2023)	Protein language model embedding encoding evolutionary and structural information from protein sequences.	2560
m_9	GTEx v8 (Consortium, 2020)	RNA-seq expression profiles across 54 human tissues, providing tissue-specific biological context.	54