A MECHANISTIC PROBE OF REGULATORY STRUCTURE IN SINGLE-CELL FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Do single-cell foundation models (scFMs) preserve the regulatory information needed to explain cellular responses to perturbation? Existing evaluations measure predictive accuracy, which cannot establish which regulatory relationships their representations retain. We introduce CLAMP, a mechanistic probe that fits a gene regulatory network through a frozen encoder and evaluates the recovered network in gene space, where its interactions can be inspected and tested against independent biological references. We apply CLAMP to four scFMs (scGPT, scPRINT, Stack and Tahoe-x1) on four Perturb-seq datasets (Norman, Replogle K562, Replogle RPE1 and X-Atlas HCT116). Fitted directly in gene space, CLAMP recovers 70–89% of an empirical ceiling, predicts unseen two-gene interventions without combinatorial training, and recovers the known lineage programs of key regulators. Regulatory recovery through the scFMs generally falls below simple baselines such as PCA and random projections. The deficit concentrates in the part of each response that identifies its regulator: the recovered networks reproduce responses shared across interventions more readily than the differences between them. A layerwise analysis shows that this specificity is already reduced at the first transformer block and is not regained with depth, and that the encoders transmit lower-variance regulatory directions more weakly than leading ones. These results suggest pretraining objectives that reward distinguishing perturbations as a way to preserve regulatory specificity. Code will be released upon acceptance.

1 INTRODUCTION

Single-cell foundation models (scFMs) are pretrained on large expression atlases and reused as general-purpose encoders, supporting tasks from cell-type annotation to *in-silico* target discovery and perturbation prediction (Theodoris et al., 2023; Cui et al., 2024). A central promise is that learning from this expression data allows scFMs to capture how genes regulate one another, supporting predictions of how cells respond to interventions. Yet their performance on perturbation prediction has not supported this promise: they underperform additive and mean baselines (Wenteler et al., 2025; Ahlmann-Eltze et al., 2025), and simple summary statistics match far more expensive learned models (Svensson et al., 2026). These findings challenge the practical value of scFMs, but leave the central question unresolved: what regulatory structure have these models learned?

A perturbation prediction is typically produced by training an additional probe on the outputs of an scFM. A poor prediction can therefore reflect a limitation of either the scFM or the probe, without revealing which regulatory relationships the scFM captures. To examine those relationships, we introduce CLAMP, a framework for evaluating the regulatory knowledge of scFMs through mechanistic network recovery. We use this mechanistic network itself as the model fitted through each frozen scFM. Its interactions can be inspected and tested, allowing us to assess the biological structure recovered rather than judging only the resulting predictions. Holding the perturbation data and mechanistic probe fixed, we compare the regulatory structure recovered through gene-space, baseline encoders and scFMs.

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.

1.1 Related Work

Our approach brings together perturbation prediction, regulatory network inference and scFM evaluation. We distinguish their goals below: predicting cellular responses, recovering the interactions underlying them, and assessing what foundation models have learned.

Regression-based perturbation prediction. A large body of work models perturbation response as a learned map from perturbation identifier and context to expression, $f_\theta : (p, x_{\mathrm{ctrl}}) \mapsto \hat{x}_p$: GEARS (Roohani et al., 2024) through a graph neural network over a Gene Ontology relational graph, CPA (Lotfollahi et al., 2023) through disentangled latent representations of perturbation, cell type and covariate effects that compose arithmetically for unseen combinations, and Attention-Pert (Bai et al., 2024) through attention over perturbation embeddings and gene context, while fine-tuned foundation models (Cui et al., 2024; Theodoris et al., 2023) apply the same framing through pretrained embeddings. All produce predicted expression profiles but no recovered regulatory model, so the structure generating the response is never made explicit and cannot itself be examined.

Recovering regulatory structure from expression. Closest in spirit is work recovering regulatory networks through dynamical models of gene expression. Cell-MNN (von Bassewitz et al., 2026) constrains a locally linearized ODE to PCA space, FLeCS (Bertin et al., 2025) keeps nonlinear kinetics but requires pseudotime to construct trajectories and fits per-condition rather than assembling a shared model, PerturbODE (Lin et al., 2026) models perturbation responses through a neural ODE and derives a regulatory network from its learned parameters, while PHOENIX (Hossain et al., 2024) fits Hill kinetics for temporal forecasting. Bicycle (Rohbeck et al., 2024) infers cyclic regulatory networks from steady-state interventional data using a linear stochastic dynamical model. These methods aim to recover regulatory dynamics from expression data. CLAMP uses mechanistic recovery as an instrument for evaluating scFMs, asking how the recovered network changes when the same fitting procedure is guided by different frozen models.

Evaluating foundation models by prediction. Regression benchmarks (Wenteler et al., 2025; Ahlmann-Eltze et al., 2025) establish that models pretrained on observational expression underperform additive and mean baselines at perturbation prediction, and recent work finds failure modes in those accuracy-based evaluations themselves (Viñas Torné et al., 2025; Miller et al., 2025), further weakening prediction accuracy as a probe of what a model has learned. These report what foundation models fail at but not why, leaving open whether the regulatory structure is present in the embedding.

Contributions. Our goal is to assess how much regulatory information can be recovered from scFMs. To this end, our contributions are threefold:

1. We introduce CLAMP, a mechanistic probe that evaluates frozen scFMs by fitting a shared regulatory network through each encoder, using implicit differentiation through perturbed steady state. The recovered networks remain in gene space, allowing their regulatory interactions to be inspected and compared directly without a learned decoder.
2. We develop a layerwise analysis combining network recovery, linearised recovery and the transmission of regulatory change to locate where recovery changes within scFMs and which differences between perturbation responses each layer preserves.
3. Across four scFMs and four Perturb-seq datasets, we find that pretrained representations generally support less recovery than simple projections, with a deficit in regulator-specific responses. We assess recovery through held-out responses, unseen two-gene combinations and independent biological references.

2 Methods

After stating the problem, we introduce the mechanistic model (Section 2.1), convert its population-level predictions into edited single cells (Section 2.2), and compare predicted and observed perturbation effects in the encoder's embedding space (Section 2.3). We then explain how the model is fitted through the frozen encoder (Section 2.4) and describe the layerwise analysis used to trace recovery changes within an encoder (Section 2.5). Appendix A goes over assumptions, scope and limitations.

Problem formulation. Given single-cell expression measurements from control cells and cells under genetic perturbations $j \in \mathcal{J}$, and a frozen encoder Enc, we seek a regulatory network A whose shared dynamics explain the responses across interventions. We jointly estimate the model parameters $\theta = (A, \mathbf{b}, \gamma, n, K)$ by comparing predicted and observed perturbation effects through Enc. The recovered network is then evaluated in gene space. Holding the mechanistic formulation fixed and varying the encoder allows regulatory recovery to be compared across models.

2.1 Recovering a gene regulatory network

Gene regulation is a dynamical process: a transcription factor's (TF) abundance influences the production of its targets, and expression settles where production balances degradation. We describe this using a simple Hill-type ODE in gene space,

$$\frac{dx_g}{dt} = b_g + \sum_{k=1}^{G} A_{gk} h(x_k) - \gamma x_g, \qquad h(x) = \frac{x^n}{K^n + x^n}. \tag{1}$$

Here G is the number of genes, x_g denotes expression in library-size-normalized, $\log(1+x)$-transformed coordinates, A_{gk} is the signed influence of gene k on gene g, b_g is a basal production parameter, and γ, n and K are shared kinetic parameters. This simplicity is deliberate: the Hill response captures saturation with few parameters, while all cross-gene influences must pass through A. Regulation therefore cannot be absorbed into additional latent variables or learned functions, but must be expressed in the network we inspect. All parameters $\theta = (A, \mathbf{b}, \gamma, n, K)$ are shared across conditions, requiring this single network to account jointly for the responses to the interventions (Appendix B derives the Hill form).

To recover this shared network from data, we use the changes in expression between control and perturbed cells: perturbing a known target and observing the response of other genes provides constraints on the regulatory network A. Crucially, a CRISPR perturbation is not a brief nudge to this system: CRISPR activation (CRISPRa) continually promotes its target's expression, while CRISPR interference (CRISPRi) continually represses it. The cells are measured while this intervention remains active, after time has elapsed for its effects to propagate through the network. We therefore model these measurements as an equilibrium reached under sustained perturbation, with the target held away from its unperturbed level (Schwanhäusser et al., 2011; Zecha et al., 2018). We represent this by *clamping* gene j: its own dynamics are replaced by the constraint $x_j = c_j$, where c_j is its observed mean expression under intervention. This fixed expression level influences the other genes through the outgoing regulatory interactions A_{ij}, while the network itself remains unchanged. The remaining genes settle where production balances decay. Setting $dx_i/dt = 0$ in Eq. 1 for these unclamped genes therefore gives

$$x_i^{*,(j)}(\theta) = \frac{1}{\gamma}\Big(b_i + \sum_k A_{ik}\, h\big(x_k^{*,(j)}\big)\Big) \;\; (i \neq j), \qquad x_j^{*,(j)} = c_j. \tag{2}$$

We solve this system by damped fixed-point iteration, reimposing the clamp at every step. To fit the network, we compare predicted and observed perturbation effects in the encoder's embedding space, and use the error to update its shared parameters. This requires translating the mechanistic model's population-level predictions into individual-cell inputs for the encoder.

2.2 From population-level predictions to individual cells

The steady states of Eq. 2 describe a population: for each gene, they give the average expression expected under an intervention. However, scFMs encode individual-cell profiles, so passing an average profile to the encoder would discard variation between cells. We therefore do not pass the model's averages to the encoder. Instead, we take real control cells and edit them so that, as a population, they display the change the network predicts. To do so, we draw a fixed set of N control cells, which we call *anchor cells*, once per dataset. From the network we derive a single quantity per gene: the fold change it predicts under intervention j, derived from its perturbed and control expression levels using the conversion to count-edit factors specified in Appendix C. The fold change is applied to the anchor cells' transcript counts: to decrease a gene, each of its transcripts is kept with probability equal to the fold change, and to increase it, new transcripts are added at random in proportion to the gene's abundance in control cells, so that edited cells remain whole-count cells

in which genes can appear and disappear. For each intervention j, this yields a set of edited anchor cells which, like the observed cells, can be passed to the encoder in its own input format. Predicted and observed perturbation effects can now be compared in the encoder's embedding space.

2.3 Comparing perturbation effects in the encoder's embedding space

For each intervention we now have two sets of cells: the perturbed cells observed in the experiment, and the anchor cells edited by the mechanistic model. We pass both, along with the control cells and the unedited anchor cells, through the encoder. In embedding space, we define the effect of an intervention as the shift of the mean embedding from control to perturbed cells. We compute this shift twice: once from the observed cells, and once from the anchor cells before and after editing,

$$\begin{aligned} \boldsymbol{\Delta}_j &= \bar{\mathbf{z}}^{(j)} - \bar{\mathbf{z}}^{(\mathrm{ctrl})} && \text{(observed)}, \\ \hat{\boldsymbol{\Delta}}_j(\theta) &= \frac{1}{N}\sum_{a=1}^{N} \mathrm{Enc}\big(\mathbf{y}_a'^{(j)}(\theta)\big) - \frac{1}{N}\sum_{a=1}^{N} \mathrm{Enc}(\mathbf{y}_a) && \text{(predicted)}, \end{aligned} \tag{3}$$

where $\bar{\mathbf{z}}^{(j)}$ and $\bar{\mathbf{z}}^{(\mathrm{ctrl})}$ are the mean embeddings of the cells observed under intervention j and of control cells, $\mathbf{y}_a$ is an anchor cell, and $\mathbf{y}_a'^{(j)}(\theta)$ is its edited version. We compare shifts, not absolute embeddings, to focus on the change associated with the intervention. The mechanistic model is fit by minimizing the squared error between predicted and observed effects,

$$\mathcal{L}_{\mathrm{pert}}(\theta) = \frac{\kappa_{\mathrm{Enc}}}{|\mathcal{J}|\, d} \sum_{j \in \mathcal{J}} \left\| \hat{\boldsymbol{\Delta}}_j(\theta) - \boldsymbol{\Delta}_j \right\|_2^2, \tag{4}$$

where d is the embedding dimension and κ_{Enc} is a scaling factor that makes the loss comparable across encoders (Appendix D). Before any fit, we check that this loss rewards the true answer over a prediction of no effect through every encoder (Appendix E). The full objective adds a control term and a sparsity penalty, $\mathcal{L} = \mathcal{L}_{\mathrm{pert}} + \lambda_{\mathrm{ctrl}}\mathcal{L}_{\mathrm{ctrl}} + \lambda_1\|A\|_1$. The control term is needed because a loss on shifts cannot locate the state from which the shifts start, while the clamp levels c_j are only meaningful relative to that state. Hence the control term anchors the unperturbed steady state of the mechanistic model to the observed control mean. The ℓ_1 penalty favors a sparse regulatory network. We fit columns of A only for directly perturbed regulators, restricting inference to regulators with direct interventional evidence.

2.4 Fitting through the frozen encoder

During fitting, the encoder remains frozen and only the mechanistic parameters θ are updated. Back-propagating the perturbation loss through the encoder tells us how changes in gene expression would reduce the embedding-space error. This signal guides the joint fit across interventions, while the recovered regulatory network remains in gene space, where its interactions can be inspected. Each intervention requires an equilibrium solve. Differentiating through all damped fixed-point iterations would retain their intermediate computations, making memory usage grow with both the number of iterations and the number of conditions. Following Bai et al. (2019) and Blondel et al. (2022), we instead differentiate the equilibrium equation implicitly: the sensitivity of a fixed point to θ follows from the equation it satisfies, without needing the path used to reach it. The forward solver therefore runs without recording an autograd graph, and a backward linear solve over the unclamped genes gives the equilibrium contribution to the parameter gradients. Appendix F gives the derivation, validity conditions and treatment of clamped coordinates and solver bounds.

2.5 Layerwise analysis of the encoder

Recovery through the full encoder measures how well its representation supports the mechanistic fit. We also examine where recovery changes within the encoder and what properties of its mapping may explain those changes. To do so, we refit at intermediate layers, refit through a linear approximation of each layer, and then measure properties of that approximation directly.

Recovery by layer. We replace the final encoder output with the representation at layer L and refit the mechanistic model using the same procedure. The resulting recovery, $\mathrm{rec}_{\mathrm{true}}(L)$ (Section 3), shows where perturbation responses become more or less recoverable through our probe.

Recovery through a linear approximation. Each encoder layer is a nonlinear function of expression. To ask how much of its recovery a linear map could already account for, we replace the layer by a local linear approximation around each anchor cell and refit the mechanistic model through it, obtaining $\mathrm{rec}_{\mathrm{lin}}(L)$. For an anchor cell with log-normalized expression $\mathbf{x}_a$, edited to $\mathbf{x}_a'^{(j)}$ under intervention j,

$$\mathrm{Enc}_L\big(\mathbf{x}_a'^{(j)}\big) \approx \mathrm{Enc}_L(\mathbf{x}_a) + J_L(\mathbf{x}_a)\big(\mathbf{x}_a'^{(j)} - \mathbf{x}_a\big), \tag{5}$$

where $J_L(\mathbf{x}_a)$ is the sensitivity of layer L at that anchor, computed as its Jacobian or, for encoders with binned inputs, from one-count finite differences (Appendix G). Comparing $\mathrm{rec}_{\mathrm{lin}}(L)$ with $\mathrm{rec}_{\mathrm{true}}(L)$ shows how recovery changes when the layer is replaced by its linear approximation. Appendix H gives the fitting details.

Transmission of regulatory change. The sensitivity matrices that define this approximation can also be examined on their own, without fitting the mechanistic model. We apply them to the observed perturbation responses in gene space and ask two questions. First, do different interventions still look different after the layer? The *separable intervention* measure removes the response common to all interventions, passes each perturbed regulator's remaining response through the layer, and measures the effective dimensionality of the resulting responses (Eq. 19). It is one if they all collapse onto a single direction. Second, which kinds of regulatory change does the layer weaken? *Per-direction transmission* measures how far the layer's output moves per unit change along each of the 48 leading principal components of the observed responses (Eq. 20). A layer that treats all directions equally scores one throughout, and a score below one means the layer attenuates that direction. The smaller components can encode differences between regulators that are less prominent in the dominant response patterns.

3 Experimental setup

Datasets and splits. We evaluate CLAMP on four Perturb-seq datasets: Norman 2019 CRISPRa in K562 (Norman et al., 2019), Replogle 2022 CRISPRi in K562 and RPE1 (Replogle et al., 2022), and X-Atlas/Orion CRISPRi in HCT116 (Huang et al., 2025), retaining 102, 58, 67 and 100 single-gene interventions, respectively. CRISPRi interventions require a measured transcription-factor target (Lambert et al., 2018), at least 25 cells and at least 30% target knockdown. X-Atlas is further restricted to the 100 strongest eligible transcriptome-wide responses. Selection is shared across encoders, with 20% of cells per intervention held out for response evaluation. All 129 Norman two-gene interventions are excluded from fitting to evaluate generalization to unseen combinations. The gene panel contains the 2,500 most variable genes plus all perturbed genes. Appendix I details preprocessing, including log-normalized expression and encoder-specific raw-count inputs.

Encoders and controls. We compare four frozen pretrained scFMs: scGPT (Cui et al., 2024), scPRINT (Kalfon et al., 2025), Stack (Dong et al., 2026) and Tahoe-x1 (Gandhi et al., 2025) using native input processing and evaluating final embeddings and intermediate layers. Binned encoders are evaluated with their native binning in the forward pass, and a differentiable relaxation is used only for gradients (Appendix J). Identity, $\mathrm{Enc}(\mathbf{x}) = \mathbf{x}$, preserves the full expression profile without learnable parameters and serves as the uncompressed recovery reference. PCA uses 512 components learned only from control cells, capturing unperturbed variation without perturbation responses. Gaussian random projections to 512 and 64 dimensions provide data-independent compression controls. Shuffling observed responses among interventions tests dependence on correct perturbation-response pairings. Randomly initializing scGPT tests the contribution of pretraining while preserving its architecture and input processing. Appendix J details the control procedures.

Fitting protocol. We fit all training interventions jointly using $N = 32$ anchor cells, beyond which recovery through Identity changes little (Appendix K), and repeat each fit with three seeds that vary the held-out cells. For each dataset, we select the learning rate using Identity and reuse it across encoders, layers and controls. All other hyperparameters are shared across datasets. Each fit uses one H200 GPU. Hyperparameters, runtimes and total compute are reported in Appendix K.

Regulatory recovery. We evaluate recovered networks in gene space using Pearson correlation across genes between predicted and observed expression changes relative to the unperturbed state, excluding directly perturbed targets whose expression is clamped. Averaging across interventions yields the single-gene intervention (SGI) score on held-out cells from fitted interventions and the

double-gene intervention (DGI) score on Norman's unseen combinations. To assess intervention specificity, we also report Top-1 retrieval: after subtracting from predicted and observed responses their mean over interventions, the fraction of predicted responses whose closest observed response by cosine similarity is the correct intervention (Appendix L).

Structural validation. On Norman, we assess networks recovered through each scFM and Identity by testing enrichment of each TF's top-25 recovered targets against biological references unused in fitting: curated hematopoietic lineage signatures, ChEA 2022 (Lachmann et al., 2010) and ENCODE TF ChIP-seq 2015 target sets (Consortium, 2012), as distributed with Enrichr (Kuleshov et al., 2016). This assesses biological agreement alongside response prediction (Appendices M and N).

4 RESULTS

4.1 REGULATORY RECOVERY ACROSS ENCODERS

Table 1 reports recovery through every encoder on the four datasets. To interpret recovery relative to an empirical recovery ceiling, we compare predictions with an *"Oracle"* that uses the training-cell mean to predict the response in held-out cells from the same intervention. Through Identity, we probe how much perturbation signal CLAMP can recover when fitted directly in gene space, showing recovery values between 70–89% of the ceiling across four datasets and establishing that the shared mechanistic model captures a substantial fraction of the reproducible response. On Norman it also predicts the 129 two-gene interventions excluded from fitting at 87% of the ceiling, approximately the same fraction as on held-out single interventions. With both targets clamped at their observed expression levels, the obtained network therefore reproduces much of the downstream response, without having been fitted to any two-gene intervention. Together, these results establish a baseline showing the mechanistic formulation can explain responses across interventions and generalize to unseen combinations.

Table 1: **Regulatory structure recovered through each encoder.** One shared gene regulatory network is fitted per dataset with the perturbation loss computed through the encoder. All metrics are then computed in gene space on held-out validation cells. SGI: Pearson correlation between predicted and observed expression shifts for single-gene interventions. DGI: the same for double-gene interventions never seen in training (Norman only). Top-1: fraction of interventions whose predicted shift, after removing the shift shared by all interventions, is closest to its own observed shift. Identity (no encoding) and PCA are positive controls, random projections are uninformed compressions at the width of PCA and at a narrower width, and the last two rows are negative controls run through the whole pipeline. Mean ± SD over 3 seeds.

	Norman K562			Replogle K562		Replogle RPE1		X-Atlas HCT116	
Encoder	SGI	DGI	Top-1	SGI	Top-1	SGI	Top-1	SGI	Top-1
Oracle									
Training-cell mean	$0.83_{\pm 0.00}$	$0.90_{\pm 0.00}$	$0.91_{\pm 0.01}$	$0.28_{\pm 0.00}$	$0.70_{\pm 0.09}$	$0.53_{\pm 0.01}$	$0.60_{\pm 0.06}$	$0.30_{\pm 0.01}$	$0.42_{\pm 0.02}$
Positive controls									
Identity	$0.72_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.63_{\pm 0.03}$	$0.25_{\pm 0.01}$	$0.43_{\pm 0.05}$	$0.44_{\pm 0.01}$	$0.25_{\pm 0.01}$	$0.21_{\pm 0.03}$	$0.19_{\pm 0.07}$
PCA (512)	$0.63_{\pm 0.03}$	$0.71_{\pm 0.03}$	$0.59_{\pm 0.01}$	$0.21_{\pm 0.01}$	$0.40_{\pm 0.03}$	$0.40_{\pm 0.02}$	$0.29_{\pm 0.05}$	$0.23_{\pm 0.01}$	$0.19_{\pm 0.02}$
Random projection (512)	$0.46_{\pm 0.03}$	$0.55_{\pm 0.02}$	$0.54_{\pm 0.05}$	$0.10_{\pm 0.00}$	$0.34_{\pm 0.03}$	$0.21_{\pm 0.01}$	$0.26_{\pm 0.05}$	$0.14_{\pm 0.03}$	$0.18_{\pm 0.03}$
Random projection (64)	$0.20_{\pm 0.01}$	$0.21_{\pm 0.01}$	$0.48_{\pm 0.03}$	$0.02_{\pm 0.01}$	$0.21_{\pm 0.05}$	$0.07_{\pm 0.01}$	$0.23_{\pm 0.08}$	$0.06_{\pm 0.01}$	$0.12_{\pm 0.01}$
Single-cell FMs									
scGPT	$0.27_{\pm 0.02}$	$0.35_{\pm 0.01}$	$0.11_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.17_{\pm 0.03}$	$0.19_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.09_{\pm 0.01}$
scPRINT	$0.27_{\pm 0.02}$	$0.35_{\pm 0.02}$	$0.24_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.23_{\pm 0.05}$	$0.16_{\pm 0.01}$	$0.28_{\pm 0.05}$	$0.08_{\pm 0.01}$	$0.10_{\pm 0.03}$
Stack-Large	$0.35_{\pm 0.01}$	$0.41_{\pm 0.01}$	$0.30_{\pm 0.05}$	$0.12_{\pm 0.01}$	$0.32_{\pm 0.04}$	$0.25_{\pm 0.01}$	$0.13_{\pm 0.03}$	$0.09_{\pm 0.02}$	$0.07_{\pm 0.03}$
Tahoe-x1 (70M)	$0.15_{\pm 0.01}$	$0.23_{\pm 0.02}$	$0.13_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.07_{\pm 0.02}$	$0.19_{\pm 0.01}$	$0.10_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.09_{\pm 0.05}$
Negative controls									
scGPT, shuffled labels	$0.04_{\pm 0.02}$	$0.05_{\pm 0.03}$	$0.01_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.02_{\pm 0.03}$	$0.08_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.02_{\pm 0.01}$
scGPT, random weights	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.01_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.02_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.01_{\pm 0.01}$

We next look at the effect of compression on regulatory signal. PCA keeps 512 directions of the control-cell variation and recovers almost as much as Identity, indicating that regulatory responses lie within ordinary cell-to-cell variation and that considerable recovery is possible after compression. A random projection of the same width keeps two thirds of Identity's recovery on Norman,

and one of width 64 a quarter. These random projections therefore show that regulatory signal can survive a degree of uninformed compression when no particular regulatory direction is affected by it. An encoder that had learned or retained regulatory structure should at least match an uninformed compression of no greater width. Instead, the four scFMs recover less than a 512-d random projection in all but three of 36 comparisons (Table 1), perform at roughly the level of a 64-d one, and remain well below Identity and PCA on almost every metric. The scFM deficits therefore cannot be explained by reduced dimensionality alone: their learned mappings and input processing make the responses less recoverable through the shared mechanistic probe.

The negative controls qualify these results: recovery through pretrained scGPT exceeds recovery than through its randomly initialized counterpart, while shuffling intervention-response pairings reduces retrieval to approximately chance. Pretraining thus does render scGPT more useful for recovery than its untrained architecture, but still below the level of a random projection. The gap to the random projections is clearest when we ask whether the recovered responses distinguish individual perturbations, rather than reproduce expression changes shared across them. A random projection to 64 dimensions keeps this specificity even when it loses correlation: on Norman it reaches an SGI of only 0.20, yet identifies the correct intervention in 48% of cases. The scFMs show the opposite pattern. Three of the four reach a higher SGI than this projection (0.27–0.35), yet none identifies more than 30% of interventions. These comparisons suggest that the pretrained representations support recovery of common response patterns more readily than the differences that distinguish individual regulators, motivating an examination of where and how this specificity becomes less recoverable within the encoders.

4.2 Locating and interpreting regulatory recovery

Figure 1 examines regulatory recovery across encoder layers and how those layers transmit observed perturbation responses. We first compare networks fitted through each layer with those fitted through its local linear approximation (panels a–d). Recovery through the original layers follows different trajectories across models: Stack-Large's DGI correlation declines from approximately 0.52 at the earliest measured block to 0.41 at the output, whereas scGPT improves from approximately 0.22 to 0.35. The final embedding is therefore not always the most informative representation for recovery. Stack-Large generally achieves comparable recovery through its original layers and their linear approximations, while Tahoe-x1 consistently recovers less through its original layers. The linearized fits show how much recovery each layer's local sensitivities support when targets and predictions share the same anchor-editing procedure.

To examine which regulatory changes the local maps preserve, we apply each layer's Jacobian to the observed perturbation responses, independently of network fitting (Figure 1e). Before applying it, we subtract the response shared by all interventions, leaving the part specific to each regulator. A model that reproduces only the shared response can still correlate positively with every intervention, so it is this specific part that distinguishes one regulator from another. The effective dimensionality of these remaining responses, which we call the number of separable interventions, is 8.4 in gene space, compared with 6.3 after a 64-d random projection and only 2.4-4.5 at the scFM outputs. Thus, the scFMs concentrate regulator-specific variation into fewer dominant directions, consistent with the reduced ability of the recovered networks to distinguish individual interventions. This reduction is already evident at the earliest measured block and persists across layers, even where recovery improves with depth.

Directional transmission reveals a related imbalance (Figure 1f). At the scFM outputs, mean transmission along regulatory directions 16–48 is only 61–69% of that along the leading five, whereas the random projection transmits these directions approximately uniformly. Lower-variance components of the observed responses therefore contribute less strongly to the embedding-space fitting signal. Together with the reduced effective rank, this suggests that dominant response patterns remain more accessible to the probe than the differences between individual regulators. Consistent with this interpretation, scPRINT and Stack-Large retain greater effective dimensionality and achieve higher retrieval on Norman than scGPT and Tahoe-x1 (Table 1). The combined analyses identify an early and persistent reduction in regulatory specificity, alongside model-dependent recovery trajectories across depth.

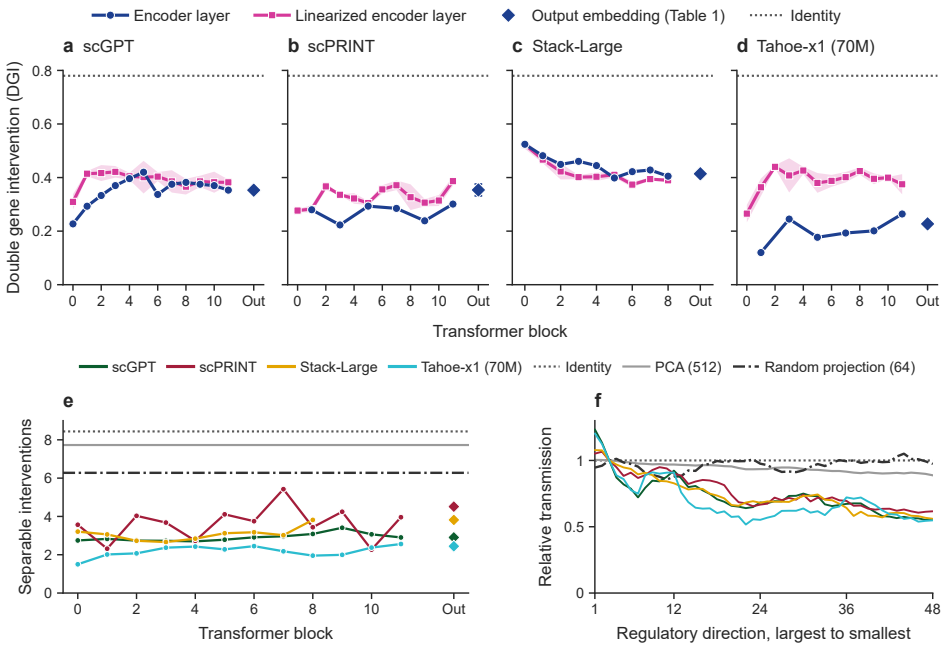

Figure 1: **Recovery by layer and transmission of regulatory change, Norman.** (a–d) DGI of the mechanistic model fitted through each transformer block and through that block's local linear approximation at the anchor cells (mean ± SD over 3 seeds). Dotted horizontal line: Identity. (e, f) We apply each layer's sensitivity matrix to the observed perturbation responses. (e) *Separable interventions*: the effective dimensionality of the intervention-specific responses, each intervention's response minus the response shared by all interventions, after the block's local linear map, by transformer block and at the output embedding. (f) *Transmission* of each of the 48 regulatory directions at the output embedding, ordered from the largest to the smallest share of the observed effects.

4.3 Structural Validation of the Recovered Networks

Finally, we assess biological agreement by testing each TF's 25 strongest recovered targets for enrichment against ChIP-seq target sets and curated lineage signatures unused during fitting (Figure 2). Because many reference sets originate from other cell types, we first establish the reference agreement detectable in the observed K562 responses. Applying the same enrichment test to the 25 genes most affected by each perturbation identifies significant enrichment for six of the 19 TFs with reference sets. We then ask whether each recovered network reproduces this agreement, using the same six TFs across encoders. Identity recovers significant enrichment for four in every seed, identifying the expected programs of erythroid differentiation for KLF1, monocyte identity for SPI1, interferon response for IRF1 and granulocyte differentiation for CEBPA. Their lineage signatures show more than 20-fold enrichment, with additional ChIP-seq support for several factors (Appendix M). The remaining factors, JUN and CEBPE, show only marginally significant enrichment in the observed responses, indicating weaker reference agreement even before network fitting.

Recovery through scFMs changes both the strongest inferred interactions and their biological agreement. Across the top-25 targets of KLF1, SPI1, IRF1 and CEBPA, Stack-Large shares 25 of 100 TF-target relationships with Identity, compared with 14 for scPRINT, five for scGPT and none for Tahoe-x1 (Figure 2a). Identity is a recovery reference, not ground truth, so this overlap complements enrichment against independent references. scGPT, scPRINT and Stack-Large each validate approximately two to three of the six factors on average, compared with four for Identity, while Tahoe-x1 validates one, matching the shuffled-label control (Figure 2c). Some biological programs thus remain recoverable through scFMs despite substantial differences in their strongest interactions. Together with the response and layerwise analyses, these results show that encoder choice affects both the specificity of recovered responses and the biological structure inferred from them.

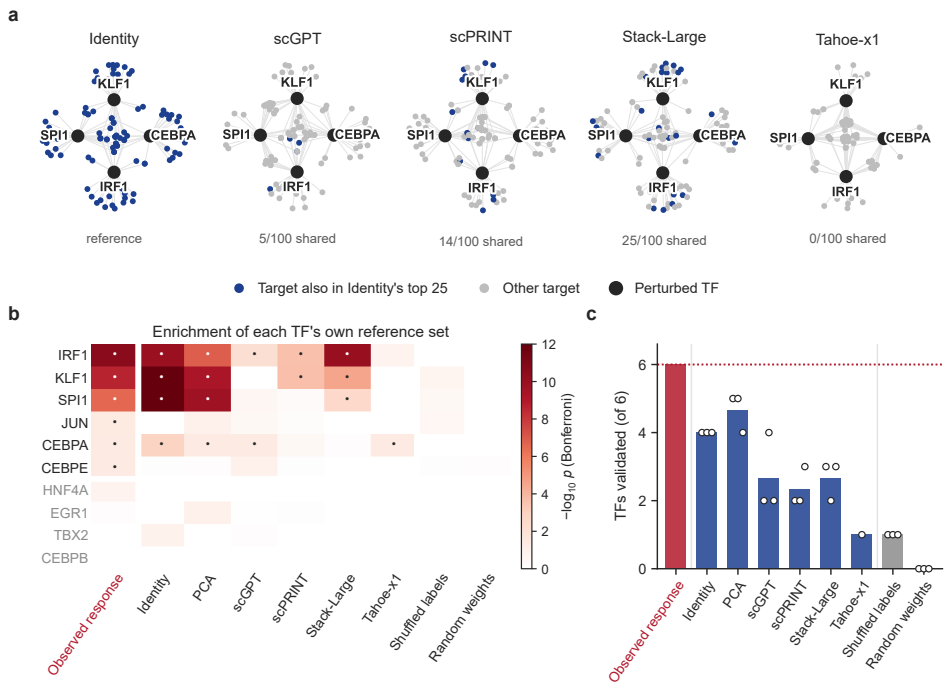

Figure 2: **Structural validation of recovered networks, Norman.** (a) Top-25 targets of KLF1, SPI1, IRF1 and CEBPA across encoders, with fixed gene positions and blue indicating targets also recovered through Identity (seed 0). (b) Enrichment of each perturbed TF's top-25 recovered targets against independent ChIP-seq (ChEA 2022, ENCODE 2015) and blood-lineage references. Color: seed mean of $-\log_{10}$ of the best own-set Bonferroni p; dots: mean $> -\log_{10} 0.05$. The first column tests the top-25 observed expression changes, identifying six TFs with significant reference agreement in K562 (black), used for network assessment. References for the remaining TFs (grey) largely derive from other cell types. (c) Number of these six TFs with significant recovered-target enrichment. Bars show seed means, white dots individual seeds, the dotted line the maximum of six.

5 Conclusion

CLAMP makes regulatory information accessible to inspection by recovering gene-space networks through frozen scFMs. Across four Perturb-seq datasets, these networks generally recover less than simple projection baselines, particularly regulator-specific effects. On Norman, regulator-specific dimensionality is reduced from the earliest measured blocks, and lower-variance regulatory directions are transmitted more weakly than dominant ones. The recovered networks also show weaker agreement with independent biological references.

These findings suggest three changes to scFM pretraining. First, objectives should reward distinguishing perturbations, not only reconstructing expression. Reconstruction losses can favor high-variance co-expression programs, allowing an encoder to lower them while attenuating regulator-specific directions. Perturbation data alone is insufficient: Tahoe-x1 was pretrained on perturbed cells yet retains the fewest separable interventions. Objectives that explicitly contrast interventions would reward preserving these directions. Second, changes should target input encoding and the first block, where reduced specificity is already evident. The four models encode expression differently yet all show this reduction, so it is not confined to one tokenization scheme. Training could encourage more uniform transmission of regulatory directions, assessed through the per-direction transmission measure. Third, both transmission readouts require no fitting, only encoder derivatives at a few dozen cells, allowing for monitoring of regulator-specific variation during pretraining. More broadly, our results suggest that regulatory inference depends on preserving small, regulator-specific response components from the first layer onward. Our framework enables testing whether pretraining that preserves these distinctions improves regulatory network recovery.

AI USE STATEMENT

In this work the research question, the mechanistic model formulation and the fitting framework were developed by the authors. We used generative AI to help as a coding assistant in implementing the framework, to assist with interpreting experimental results, refining hypotheses about the observed limitations of the evaluated representations, and drafting and editing the manuscript for clarity, concision and organization. We also used AI tools to draft and organize the Appendices, checking the particulars against the implemented code to ensure no mismatch between code and paper slipped in. We also used AI tools to assist with checking mathematical derivations, which were reviewed by the authors. Additionally, we used AI tools to create/refine scientific figures. We have not used AI tools to generate synthetic datasets, to formulate mathematical claims or provide ingredients for their proofs or to write proofs. We've reviewed all AI-assisted text and content, and have verified the claims therein. We take responsibility for the final content of this work, including all AI-assisted text and claims.

REPRODUCIBILITY STATEMENT

All components needed to reproduce our results are specified in the main text and appendix. The mechanistic model is defined in Section 2.1 and Appendix B, the count edit that turns its predictions into edited cells in Appendix C, and the full objective, including the loss scale and control term, in Appendix D, with the derivation and validity conditions of the implicit gradients in Appendix F. Appendix J documents each encoder's input format, how its layers are read and the checks it passes before use. The four public Perturb-seq datasets and their processing, including the selection of interventions, the gene panels and the cell splits, are described in Appendix I, the evaluation metrics in Appendix L, and all hyperparameters, the selection of the learning rate on Identity alone, the solver, the seeds and the compute in Appendix K. The layerwise analyses are specified in Appendices G and H, and the structural validation, including the reference sets and the enrichment test, in Appendices M and N. The appendix was written against the code and checked for consistency with it. All encoders are frozen, publicly available checkpoints, and every fit in Table 1 is repeated over three seeds. The code, configuration files and the scripts that produce every table and figure will be released upon acceptance.

REFERENCES

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

Ding Bai, Caleb N Ellington, Shentong Mo, Le Song, and Eric P Xing. AttentionPert: accurately modeling multiplexed genetic perturbations with multi-scale effects. *Bioinformatics*, 40 (Supplement_1):i453–i461, 2024.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

Paul Bertin, Joseph D Viviano, Alejandro Tejada-Lapuerta, Weixu Wang, Stefan Bauer, Fabian J Theis, and Yoshua Bengio. A scalable gene network model of regulatory dynamics in single cells. *arXiv preprint arXiv:2503.20027*, 2025.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in Neural Information Processing Systems*, 35:5230–5242, 2022.

ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.

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.

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.

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*, 2025.

Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46 (D1):D380–D386, 2018.

Stuart B Hay, Kyle Ferchen, Kashish Chetal, H Leighton Grimes, and Nathan Salomonis. The Human Cell Atlas bone marrow single-cell interactive web portal. *Experimental Hematology*, 68: 51–61, 2018.

Intekhab Hossain, Viola Fanfani, Jonas Fischer, John Quackenbush, and Rebekka Burkholz. Biologically informed NeuralODEs for genome-wide regulatory dynamics. *Genome Biology*, 25(1): 127, 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, Mohamed Ameen, Kwontae You, Christopher J Graves, Hyunsung John Kim, Adam J Litterman, Rene V Sit, Alex Blocker, and Ci Chu. X-Atlas/Orion: Genome-wide Perturb-seq datasets via a scalable fix-cryopreserve platform for training dose-dependent biological foundation models. *bioRxiv*, 2025.

Jérémie Kalfon, Jules Samaran, Gabriel Peyré, and Laura Cantini. scPRINT: pre-training on 50 million cells allows robust gene network predictions. *Nature Communications*, 16(1):3607, 2025.

Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic acids research*, 44(W1):W90–W97, 2016.

Alexander Lachmann, Huilei Xu, Jayanth Krishnan, Seth I Berger, Amin R Mazloom, and Avi Ma'ayan. ChEA: transcription factor regulation inferred from integrating genome-wide ChIP-X experiments. *Bioinformatics*, 26(19):2438–2444, 2010.

Samuel A Lambert, Arttu Jolma, Laura F Campitelli, Pratyush K Das, Yimeng Yin, Mihai Albu, Xiaoting Chen, Jussi Taipale, Timothy R Hughes, and Matthew T Weirauch. The human transcription factors. *Cell*, 172(4):650–665, 2018.

Zaikang Lin, Sei Chang, Aaron Zweig, Minseo Kang, Fabian J Theis, Elham Azizi, and David A Knowles. Interpretable neural ODEs for gene regulatory network discovery under perturbations. *International Conference on Machine Learning*, 2026.

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), 2023.

Henry E Miller, Gabriel M Mejia, Francis JA Leblanc, Bo Wang, Brendan Swain, and Lucas Paulo de Lima Camillo. Deep learning-based genetic perturbation models do outperform uninformative baselines on well-calibrated metrics. *bioRxiv*, 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.

Noa Novershtern, Aravind Subramanian, Lee N Lawton, Raymond H Mak, W Nicholas Haining, Marie E McConkey, Naomi Habib, Nir Yosef, Cindy Y Chang, Tal Shay, et al. Densely interconnected transcriptional circuits control cell states in human hematopoiesis. *Cell*, 144(2):296–309, 2011.

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, Augustin Luna, Nils Blüthgen, and Chris Sander. scPerturb: harmonized single-cell perturbation data. *Nature Methods*, 21, 2024.

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, Nika Iremadze, Florian Oberstrass, Doron Lipson, Jessica L Bonnar, Marco Jost, Thomas M Norman, and Jonathan S Weissman. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575, 2022.

Martin Rohbeck, Brian Clarke, Katharina Mikulik, Alexandra Pettet, Oliver Stegle, and Kai Ueltzhöffer. Bicycle: Intervention-based causal discovery with cycles. In *Causal Learning and Reasoning*, pp. 209–242. PMLR, 2024.

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

Stefan Schoenfelder and Peter Fraser. Long-range enhancer–promoter contacts in gene expression control. *Nature Reviews Genetics*, 20(8):437–455, 2019.

Björn Schwanhäusser, Dorothea Busse, Na Li, Gunnar Dittmar, Johannes Schuchhardt, Jana Wolf, Wei Chen, and Matthias Selbach. Global quantification of mammalian gene expression control. *Nature*, 473(7347):337–342, 2011.

Valentine Svensson, Umair Khan, Hamed Heydari, Airol A Ubas, Nicole Thomas, Daniele Merico, Hani Goodarzi, John Yu, Nima Alidoust, and Shreshth Gandhi. Back to basics: Observed statistics are sufficient to predict drug responses. *bioRxiv*, 2026.

Daniel G Tenen. Disruption of differentiation in human cancer: AML shows the way. *Nature Reviews Cancer*, 3(2):89–101, 2003.

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.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A Teichmann, Mor Nitzan, and Maria Brbić. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, 2025.

Jan-Philipp von Bassewitz, Adeel Pervez, Marco Fumero, Matthew Robinson, Theofanis Karaletsos, and Francesco Locatello. Learning explicit single-cell dynamics using ode representations. *International Conference on Learning Representations*, 2026.

Aaron Wenteler, Martina Occhetta, Nikhil Branson, Victor Curean, Magdalena Huebner, William Dee, William Connell, Siu Pui Chung, Alex Hawkins-Hooker, Yasha Ektefaie, César Miguel Valdez Córdova, and Amaya Gallagher-Syed. PertEval-scFM: Benchmarking single-cell foundation models for perturbation effect prediction. In *Forty-second International Conference on Machine Learning*, 2025.

Nicola K Wilson, Samuel D Foster, Xiaonan Wang, Kathy Knezevic, Judith Schütte, Polynikis Kaimakis, Paulina M Chilarska, Sarah Kinston, Willem H Ouwehand, Elaine Dzierzak, et al. Combinatorial transcriptional control in blood stem/progenitor cells: genome-wide analysis of ten major transcriptional regulators. *Cell stem cell*, 7(4):532–544, 2010.

Jana Zecha, Chen Meng, Daniel P Zolg, Patroklos Samaras, Mathias Wilhelm, and Bernhard Kuster. Peptide level turnover measurements enable the study of proteoform dynamics. *Molecular & Cellular Proteomics*, 17(5):974–992, 2018.

SUPPLEMENTARY MATERIAL

This supplementary material provides further implementation details deferred from the main text.

- Appendix A states the assumptions of the method, including the steady-state response to sustained interventions, and the limits of the measurement.
- Appendix B (supporting Section 2.1) derives the Hill function from cooperative TF binding and discusses the use of shared kinetic parameters.
- Appendix C (supporting Section 2.2) specifies how the anchor cells are drawn, how the fold change predicted by the mechanistic model is formed, how it is applied to the anchor cells' transcript counts, and how gradients pass through this discrete step.
- Appendix D (supporting Section 2.3) defines the scaling factor that makes the loss comparable across encoders, the control term, and the gene-space quantities that enter the fit.
- Appendix E (supporting Sections 2.2 and 2.3) checks, before any fit, that the objective rewards the true answer over a prediction of no effect through every encoder, and compares the count edit with a fold change applied to log-normalized values and with passing average profiles to the encoder.
- Appendix F (supporting Section 2.4) derives the gradients through the control and perturbed equilibria by implicit differentiation, gives the closed-form Jacobian and parameter gradients of the Hill model, and describes the treatment of clamped coordinates, the surrogate derivatives and the solver bounds.
- Appendix G (supporting Section 2.5 and Figure 1e–f) defines the separable-intervention count and the per-direction transmission, the regulatory directions they are computed along and the Jacobians at the anchor cells, and gives their values for the reference encoders.
- Appendix H (supporting Section 2.5 and Figure 1a–d) details recovery through the linearized encoder: how the linear approximation is formed at each anchor cell, how its targets are built, and what the comparison with the encoder layer itself measures.
- Appendix I (supporting Section 3) describes the four Perturb-seq datasets, the selection of interventions, the control cells, the gene panel and the cell splits.
- Appendix J (supporting Section 3) documents each encoder's native input format, the differentiable relaxation of the binning used by scGPT and Tahoe-x1, how intermediate layers are read, the two negative controls, and the checks each encoder passes before use.
- Appendix K (supporting Section 3) reports the learning rate selected for each dataset, the settings shared across all encoders and datasets, the choice of the number of anchor cells, the optimizer, initialization and solver, and the compute.
- Appendix L (supporting Section 3 and Table 1) defines the SGI, DGI and Top-1 scores, the training-cell reference row, and what the seeds vary.
- Appendix M (supporting Section 3 and Figure 2) describes the structural validation procedure, how each TF's own reference is defined, which references apply to this screen, the full enrichment results for the network recovered through Identity against lineage signatures, ChEA 2022, ENCODE 2015, MSigDB Hallmark, Reactome 2022 and TRRUST, and the validation of each encoder's network.
- Appendix N (supporting Section 3) documents the curated hematopoietic lineage signature library, with its gene lists and sources, and the filter applied to MSigDB Hallmark and Reactome 2022 before enrichment testing.

A SCOPE, ASSUMPTIONS AND LIMITATIONS

The measurement rests on four assumptions, which delimit where it applies.

1. **Steady-state response.** We model each perturbed population as an equilibrium reached under a sustained intervention (Section 2.1). This holds when the cells are profiled after the response has propagated, as in the multi-day post-infection windows of the four screens we use, given the protein turnover times of human cell lines. It would not hold for perturbations sampled within hours, such as drug treatments, where the observed state lies on a trajectory rather than at a fixed point.
2. **One deterministic equilibrium per intervention.** The mechanistic model predicts one equilibrium per intervention and is fitted to population-level effects: the count edit gives every anchor cell the same fold change, and the objective compares mean embeddings. Variation between cells within an intervention, multistability, and interventions that split a population between fates are not modelled.
3. **Regulators restricted to perturbed genes.** Only the columns of A belonging to directly perturbed genes are learned, so the network is silent on regulators that were not intervened on, and on the CRISPRi screens, where we fit a subset of the transcription factors that pass the filters of Appendix I, it is silent on the rest. This is what the data support: only a perturbed regulator carries direct interventional evidence about its outgoing edges. It also keeps the parameter count at $G \times |\mathcal{J}|$ rather than G^2.
4. **Hill kinetics with shared scalars.** The production term is a Hill function with one cooperativity, one half-maximal constant and one decay rate shared across genes (Appendix B). Real regulators differ in cooperativity and affinity and real transcripts in turnover, and the signed scalar A_{ik} compresses these into one effective coupling. The framework accepts any differentiable production term with a fixed point; Hill kinetics is the choice made here for parsimony and interpretability. We therefore do not claim that a recovered edge separates direct from indirect regulation or recovers the biophysical constants of a specific interaction. The structural validation of Appendix M is the check that the recovered network nonetheless aligns with independently measured regulatory structure rather than with a steady-state-equivalent surrogate.

The gradients through the binned encoders use surrogate derivatives (Appendix F), while the objective itself is evaluated exactly. Finally, the separable-intervention count and the per-direction transmission describe each layer locally, around the unperturbed anchor cells from which every edit starts. They state conditions that recovery requires rather than guarantee it, which is why we report them alongside recovery.

B HILL KINETICS: BIOPHYSICAL DERIVATION AND PARAMETER CHOICES

This appendix derives the Hill function from cooperative TF-promoter binding equilibria, motivates our use of shared scalar parameters, and notes the framework's generality over kinetic forms beyond Hill.

B.1 DERIVATION OF THE HILL FUNCTION

Consider a promoter with n binding sites for a transcription factor (TF) at intracellular concentration x. Under the cooperative-binding assumption, the promoter is transcriptionally active only when all n sites are simultaneously occupied by the TF. Writing the cooperative binding reaction as $\mathrm{P} + n\,\mathrm{TF} \rightleftharpoons \mathrm{P}_{\text{bound}}$, the equilibrium dissociation constant is

$$K_d = \frac{[\mathrm{P}]\,x^n}{[\mathrm{P}_{\text{bound}}]}. \tag{6}$$

The fraction of time the promoter spends in the active state is therefore

$$\frac{[\mathrm{P}_{\text{bound}}]}{[\mathrm{P}] + [\mathrm{P}_{\text{bound}}]} = \frac{x^n/K_d}{1 + x^n/K_d} = \frac{x^n}{K^n + x^n} = h(x;\, n,\, K), \tag{7}$$

where $K = K_d^{1/n}$ is the concentration at which the promoter is half-occupied. The Hill function is sigmoidal in x, with steepness controlled by n: at $n = 1$ it reduces to Michaelis-Menten saturation, and as $n \to \infty$ it approaches a hard threshold at $x = K$. We initialize $n = 2$, corresponding to dimerized cooperative binding, the most common regime in eukaryotic transcriptional regulation, and learn n jointly with the other shared scalars (Appendix K).

B.2 Per-regulator versus multi-regulator nonlinearity

When a target gene i is regulated by multiple transcription factors $\{x_j\}_{j \in \text{regs}(i)}$, two natural functional forms are available. The *per-regulator* form, used in this work,

$$\frac{dx_i}{dt} \propto \sum_j A_{ij}\, h(x_j;\, n,\, K), \tag{8}$$

applies the Hill nonlinearity to each regulator independently before summing. The *multi-regulator* form,

$$\frac{dx_i}{dt} \propto h\left(\sum_j A_{ij}\, x_j;\, n,\, K\right), \tag{9}$$

applies a single Hill activation to a linear combination of regulators. Per-regulator nonlinearity is consistent with how cooperativity is modeled in deterministic systems biology when each TF binds independently at distinct sites, and it makes the individual edge coefficients A_{ij} interpretable as marginal regulatory contributions. The multi-regulator form, used by FLeCS (Bertin et al., 2025), is a coarser approximation that compresses regulator interactions into a sigmoidal threshold on a weighted sum and obscures the contribution of any single TF to the response. Both forms are admissible in our framework without changing the recovery procedure. We adopt the per-regulator form for interpretability.

B.3 Generality over kinetic forms

The recovery framework treats the production term in Eq. 1 as a black-box differentiable function. Hill kinetics is one instantiation, chosen for biological interpretability. Other admissible parameterizations include the following.

- **Michaelis-Menten** kinetics, the $n = 1$ special case of Hill, suited to non-cooperative regulation.
- **Mass action** kinetics, $\frac{dx_i}{dt} \propto \sum_j A_{ij} x_j$, the linear special case in which each regulator contributes proportionally to its concentration.
- **Generalized Hill** with regulator-specific scalars $h(x_j;\, n_j,\, K_j)$, relaxing the shared-scalar simplification at the cost of additional parameters.
- **Neural drift**, $\frac{dx_i}{dt} \propto f_\phi(\mathbf{x})$ with f_ϕ a differentiable neural network, abandoning interpretability of individual edges in exchange for representational flexibility.

The implicit differentiation of Section 2.4 requires only that $\partial F/\partial \mathbf{x}$ and $\partial F/\partial \theta$ be computable, either in closed form (Hill, Michaelis-Menten, mass action) or via autograd through the production term (neural drift). The loss of Eq. 4 and the joint fit across interventions are independent of the kinetic form. Choosing among parameterizations is therefore a modeling decision rather than a methodological constraint.

C Editing anchor cells

Anchor cells. We draw $N = 32$ control cells uniformly at random, once per dataset, and use the same cells for every encoder, layer and seed. Anchor cell a has raw transcript counts y_{ag} over the gene panel and a size factor s_a, its total transcript count divided by the median total count, so that its log-normalized expression is $x_{ag} = \log(1 + y_{ag}/s_a)$.

Fold change. Let $\mathbf{x}^{*,(\mathrm{ctrl})}(\theta)$ denote the steady state of the network without a clamp, $\bar{x}_g^{(\mathrm{ctrl})}$ the observed mean log-normalized expression of gene g in control cells, and $\tilde{x} = e^x - 1$ the map from log-normalized expression back to the count scale. The fold change of gene g under intervention j is

$$r_{jg} = 1 + \frac{\tilde{x}_g^{*,(j)} - \tilde{x}_g^{*,(\mathrm{ctrl})}}{\tilde{\bar{x}}_g^{(\mathrm{ctrl})} + \epsilon}, \tag{10}$$

clipped to $[0, 100]$, with $\epsilon = 10^{-2}$ keeping the ratio finite for genes that are almost silent. The denominator is fixed to the observed control level rather than the network's own control state, so that the scale of the edit does not change as the network's unperturbed state moves during fitting. When the two coincide, $r_{jg} = (\tilde{x}_g^{*,(j)} + \epsilon)/(\tilde{x}_g^{*,(\mathrm{ctrl})} + \epsilon)$.

Sampling. With μ_g the mean normalized count of gene g across control cells, the edited counts are drawn independently over genes and cells as

$$y_{ag}'^{(j)} \sim \begin{cases} \mathrm{Binomial}\left(y_{ag},\, r_{jg}\right), & r_{jg} \le 1, \\ y_{ag} + \mathrm{Poisson}\left(\left(r_{jg} - 1\right) s_a\, \mu_g\right), & r_{jg} > 1, \end{cases} \tag{11}$$

and the edited cell keeps the size factor of its anchor. These are the thinning and superposition operations under which a Poisson count of rate λ becomes one of rate $r\lambda$. In a population with mean normalized count μ_g, both multiply the expected mean normalized count by r_{jg}, and $r_{jg} = 1$ leaves every cell unchanged. Real counts are over-dispersed relative to a Poisson distribution, so the correspondence is approximate.

Gradient. The draw is discrete, so we use a straight-through estimator: the forward pass uses the sampled counts, and the backward pass uses the derivative of their expectation, y_{ag} for $r_{jg} \le 1$ and $s_a \mu_g$ for $r_{jg} > 1$. For scGPT this term is rescaled to the one-count gradient of Appendix J, and for Tahoe-x1 by scales calibrated against exact one-count differences during fitting (Appendix J). The random generator is seeded by the optimization step, so that the recomputation in the backward pass reproduces the forward draw.

D Objective: scaling and regularization

Loss scaling. Embedding spaces differ in scale by more than an order of magnitude between encoders, whereas the weights of the regularization terms are fixed. Without a correction, an encoder with small embedding shifts would have a small data term relative to those terms, and would appear to recover less whatever it retains. We therefore set

$$\kappa_{\mathrm{Enc}} = \frac{\frac{1}{|\mathcal{J}|\,G} \sum_j \left\| \bar{\mathbf{x}}^{(j)} - \bar{\mathbf{x}}^{(\mathrm{ctrl})} \right\|_2^2}{\frac{1}{|\mathcal{J}|\,d} \sum_j \left\| \boldsymbol{\Delta}_j \right\|_2^2}, \tag{12}$$

the ratio of the mean squared observed effect in gene space to that in embedding space. A zero-effect prediction, $\hat{\boldsymbol{\Delta}}_j = \mathbf{0}$, then incurs the same loss through every encoder, equal to its loss through the identity encoder.

Control term. $\mathcal{L}_{\mathrm{ctrl}} = \frac{1}{G} \|\mathbf{x}^{*,(\mathrm{ctrl})}(\theta) - \bar{\mathbf{x}}^{(\mathrm{ctrl})}\|_2^2$, where $\mathbf{x}^{*,(\mathrm{ctrl})}(\theta)$ is the steady state without a clamp and $\bar{\mathbf{x}}^{(\mathrm{ctrl})}$ the observed control mean. It is computed in gene space, because a population average cannot be passed to the encoder (§2.2).

Gene-space quantities that enter the fit. The responses to interventions reach the mechanistic model only through the encoder. The gene-space measurements of perturbed cells that enter the fit are the clamp levels c_j and the single scalar κ_{Enc}; the perturbed gene is excluded from every evaluation score.

Weights. λ_{ctrl} and λ_1 are given in Appendix K.

E THE OBJECTIVE REWARDS THE TRUE ANSWER

Before any network is fitted, we check that the objective of Eq. 4 can tell the true gene-space answer apart from a prediction of no effect through every encoder. The check involves no mechanistic model and no optimization. We substitute the observed mean profiles for the steady states of Eq. 2: for each intervention j, the perturbed and control steady states are replaced by $\bar{\mathbf{x}}^{(j)}$ and $\bar{\mathbf{x}}^{(\mathrm{ctrl})}$, the fold change of Eq. 10 is formed from them, the anchor cells are edited as in Eq. 11, and the predicted effect $\hat{\boldsymbol{\Delta}}_j$ is compared with the observed effect $\boldsymbol{\Delta}_j$ exactly as during fitting. We report the ratio

$$\rho_{\mathrm{truth}} = \frac{\mathcal{L}_{\mathrm{truth}}}{\mathcal{L}_{\mathrm{zero}}}, \qquad \mathcal{L}_{\mathrm{zero}} = \frac{1}{|\mathcal{J}|\, d} \sum_{j} \|\boldsymbol{\Delta}_j\|_2^2, \tag{13}$$

where $\mathcal{L}_{\mathrm{zero}}$ is the loss of the zero-effect prediction $\hat{\boldsymbol{\Delta}}_j = \mathbf{0}$, a fixed reference independent of the network, and $\mathcal{L}_{\mathrm{truth}}$ is the loss at the observed answer. A ratio near one means the objective cannot distinguish the truth from a prediction of no effect, so a low recovery through that encoder would say nothing about the encoder. We also report the median over interventions of the cosine between $\hat{\boldsymbol{\Delta}}_j$ and $\boldsymbol{\Delta}_j$, which is insensitive to the overall scale of the predicted effect.

Table 2 gives the ratio on Norman for the count edit used throughout the paper and, for comparison, for a fold change applied directly to the anchor cells' log-normalized values, which cannot switch a gene on or off and yields fractional counts. Under the count edit the truth removes at least two thirds of the loss for every encoder. The fold change on log-normalized values is admissible for the encoders that read continuous values but fails for Tahoe-x1, whose native binning is sensitive to which genes are detected rather than to their values (Appendix J). Passing the average profile of each intervention to the encoder, in place of edited cells, gives $\rho_{\mathrm{truth}} = 14.8$ through scGPT, 8.9 through Tahoe-x1, 5.7 through Stack and 0.96 through scPRINT. The true answer then scores worse than a prediction of no effect through three of the four foundation models and only 4% better through scPRINT, so no recovery could be read through any of them. Through the two binned encoders, the predicted effect is also unrelated in direction to the observed one (median cosine -0.18 for scGPT and -0.08 for Tahoe-x1). The reason is that an encoder applied to an average profile does not return the average of its outputs over the cells (Section 2.2).

Table 2: **Loss at the true answer relative to a zero-effect prediction, Norman.** Each encoder reads its native input format. Lower ρ_{truth} is better; a ratio of one means the objective cannot reward the truth. The count edit is the construction used in every fit of the paper. Average profile: the mean profile of each intervention is passed to the encoder in place of edited cells.

	Count edit (Eq. 11)		Fold change on log values		Average profile	
Encoder	ρ_{truth}	median cosine	ρ_{truth}	median cosine	ρ_{truth}	median cosine
Identity	0.06	0.98	0.10	0.98	0.00	1.00
scGPT	0.17	0.94	0.24	0.92	14.8	−0.18
scPRINT	0.25	0.89	0.34	0.86	0.96	0.76
Stack	0.32	0.91	0.20	0.91	5.72	0.71
Tahoe-x1	0.21	0.91	0.82	0.58	8.94	−0.08

The same check is run on every dataset before its fits are read.

F GRADIENTS THROUGH THE EQUILIBRIUM

The perturbation loss depends on the mechanistic parameters θ only through the steady states: the perturbed states $\mathbf{x}^{*,(j)}(\theta)$ of Eq. 2 and, through the fold change of Eq. 10, the unclamped control state $\mathbf{x}^{*,(\mathrm{ctrl})}(\theta)$. Its gradient therefore reaches θ in two stages. The first carries the loss back to these steady states, through the encoder and the edit of the anchor cells, and produces one upstream vector for each, $\mathbf{v}^{(j)} = \partial\mathcal{L}/\partial\mathbf{x}^{*,(j)}$ and $\mathbf{v}^{(\mathrm{ctrl})} = \partial\mathcal{L}/\partial\mathbf{x}^{*,(\mathrm{ctrl})}$. The second carries each vector back to θ through its own equilibrium condition, and the contributions are summed. The control term of the objective adds a separate gradient through the same control equilibrium. We describe both, then the two places where the exact derivative is replaced by a surrogate.

F.1 From the loss to the steady state

The predicted effect of Eq. 3 is a mean of encoder outputs over edited anchor cells. Evaluating it for all interventions at once would keep every encoder activation in memory until the backward pass, so the forward pass runs the encoder in micro-batches without recording a graph and stores only the steady states. The backward pass recomputes each micro-batch with recording enabled, accumulates the vector-Jacobian product into $\mathbf{v}^{(j)}$, and discards it. Memory is bounded by the micro-batch size and the result is the same gradient unrestricted automatic differentiation would return; we verify this on a small non-linear encoder to floating-point precision. The random draws of the count edit are regenerated from the same seed in the recomputation, so the backward pass sees the cells the forward pass produced (Appendix C).

F.2 From the steady state to the parameters

Write $g(\mathbf{x}, \theta) = \big(\mathbf{b} + A\,h(\mathbf{x}; n, K)\big)/\gamma$ for the right-hand side of Eq. 2, so that the steady state satisfies $F(\mathbf{x}^*, \theta) = \mathbf{x}^* - g(\mathbf{x}^*, \theta) = \mathbf{0}$. Differentiating this identity with respect to θ gives

$$\left.\frac{\partial F}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} \frac{\partial \mathbf{x}^*}{\partial \theta} + \left.\frac{\partial F}{\partial \theta}\right|_{\mathbf{x}^*} = \mathbf{0}, \qquad \text{hence} \qquad \frac{\partial \mathbf{x}^*}{\partial \theta} = -J^{-1} \left.\frac{\partial F}{\partial \theta}\right|_{\mathbf{x}^*}, \quad J = \left.\frac{\partial F}{\partial \mathbf{x}}\right|_{\mathbf{x}^*}. \tag{14}$$

For a scalar loss with upstream gradient $\mathbf{v}$, the parameter gradient is

$$\frac{\partial \mathcal{L}}{\partial \theta} = \mathbf{v}^\top \frac{\partial \mathbf{x}^*}{\partial \theta} = -\mathbf{w}^\top \frac{\partial F}{\partial \theta}, \qquad J^\top \mathbf{w} = \mathbf{v}. \tag{15}$$

One linear solve of size $G \times G$ per intervention therefore gives the backward signal for every parameter group, whatever number of iterations the forward solver took, and nothing from the forward iteration needs to be stored.

Closed forms. With $h'(x) = \partial h / \partial x$, the Jacobian is

$$J = I - \frac{1}{\gamma}\, A \operatorname{diag}\big(h'(\mathbf{x}^*)\big), \tag{16}$$

and the Hill function admits the derivatives

$$\begin{aligned} h'(x) &= \frac{n}{x}\, h(x)\big(1 - h(x)\big), \\ \frac{\partial h}{\partial n} = h(x)\big(1 - h(x)\big) \ln \frac{x}{K}, &\qquad \frac{\partial h}{\partial K} = -\frac{n}{K}\, h(x)\big(1 - h(x)\big), \end{aligned} \tag{17}$$

so the factor $h(1-h)$ is computed once and shared. Substituting the partial derivatives of F into Eq. 15 and using the fixed-point relation $b_i + \sum_k A_{ik} h(x_k^*) = \gamma x_i^*$ for the γ term gives

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial A_{ik}} = \frac{w_i\, h(x_k^*)}{\gamma}, \qquad \frac{\partial \mathcal{L}}{\partial b_i} = \frac{w_i}{\gamma}, \qquad \frac{\partial \mathcal{L}}{\partial \gamma} = -\frac{1}{\gamma} \sum_i w_i\, x_i^*, \\ \frac{\partial \mathcal{L}}{\partial \alpha} = \frac{1}{\gamma}\, \mathbf{w}^\top A \left.\frac{\partial \mathbf{h}}{\partial \alpha}\right|_{\mathbf{x}^*} \quad \text{for } \alpha \in \{n, K\}. \end{aligned} \tag{18}$$

The gradients of the different interventions are summed, and the sparsity and control terms add their own gradients in gene space.

Clamped coordinates. Under intervention j the coordinate $x_j^{*,(j)} = c_j$ is set by the data and does not depend on θ. Its residual is $F_j = x_j - c_j$, whose row of J is the unit row $\mathbf{e}_j^\top$ and whose derivative with respect to θ is zero. We build J with this row, set $v_j = 0$ and solve Eq. 15. Because row j of J has no off-diagonal entries, the unclamped components satisfy the reduced system $J_{UU}^\top \mathbf{w}_U = \mathbf{v}_U$ exactly, where U is the set of unclamped coordinates. The component w_j is in general nonzero, but it multiplies the zero row $\partial F_j / \partial \theta$, so the parameter gradients of Eq. 18 are summed over U only. The clamped coordinate therefore enters the gradients only through the columns A_{ij} that carry its influence to the other genes. Double interventions clamp two coordinates and are treated identically.

F.3 SURROGATE DERIVATIVES

The forward pass is evaluated exactly, but at two points its derivative is replaced by a surrogate. The count edit draws integer counts, and the backward pass uses the derivative of their expectation (Appendix C). The binned encoders map a value to a bin through a piecewise-constant rule, and the backward pass uses a continuous relaxation of that rule whose forward value is the wrapper's bin: the library's own rule for Tahoe-x1, and for scGPT the library's rule with fixed rather than random tie-breaking (Appendix J). For scGPT the two are combined: the backward pass uses the change in bin position produced by one whole count. For Tahoe-x1 the relaxation is a unit slope, and the straight-through term is rescaled during fitting against exact one-count differences of the encoder. In the encoders that tokenize only detected genes, a gene absent from an anchor cell contributes no token and receives no gradient from that cell; it can still be switched on by the edit, after which it carries gradient. The vector $\mathbf{v}^{(j)}$ is therefore a surrogate for the derivative of the loss with respect to the steady state, while Eq. 15 is exact given $\mathbf{v}^{(j)}$.

F.4 SOLVER BOUNDS AND CONDITIONING

The forward solver iterates $\mathbf{x} \leftarrow \alpha\, g(\mathbf{x}, \theta) + (1 - \alpha)\, \mathbf{x}$ with damping α, bounds every coordinate to $[10^{-6}, 50]$ after each step, and reimposes the clamp; it runs a fixed number of iterations and is warm-started from the previous epoch's state (Appendix K). Where the lower bound is active, the coordinate is set by the bound rather than by θ, so its true sensitivity is zero, whereas Eq. 14 is written for the unbounded map. We therefore treat such coordinates as fixed, exactly as clamped ones: they are added to the clamped set before the solve of Eq. 15, which removes them from the parameter gradients.

Let $B = A\,\mathrm{diag}\big(h'(\mathbf{x}^*)\big)/\gamma$ and let U contain the coordinates that are neither clamped nor at the bound. The system of Eq. 15 is well posed when J is non-singular, equivalently when 1 is not an eigenvalue of B_{UU}; $\rho(B_{UU}) < 1$ is sufficient. Away from active bounds, the damped forward iteration converges locally when $\rho\big((1-\alpha)I + \alpha B_{UU}\big) < 1$, which also holds whenever $\rho(B_{UU}) < 1$. We solve the dense system directly, without regularization. During fitting we monitor an estimate of $\rho(B)$ at the unclamped control state, obtained by power iteration, and the relative fixed-point residual of the solve; these serve as diagnostics, and no step is rejected on their basis.

G SEPARABLE INTERVENTIONS AND PER-DIRECTION TRANSMISSION

The two readouts of Section 2.5 describe what a block passes on before any mechanistic model is fitted. Both are computed from the Jacobian of the block at the $N = 32$ anchor cells used by every fit, with each encoder in its native input format. This appendix specifies the regulatory directions, the Jacobians, the two readouts and the values of the reference encoders.

Regulatory directions. For each fitted intervention j we form the observed gene-space effect $\delta\mathbf{x}_j = \bar{\mathbf{x}}^{(j)} - \bar{\mathbf{x}}^{(\mathrm{ctrl})}$ over all cells of the intervention and stack the effects as the rows of $S \in \mathbb{R}^{|\mathcal{J}| \times G}$. The mean effect over interventions, $\mathbf{m} = \frac{1}{|\mathcal{J}|} \sum_j \delta\mathbf{x}_j$, is the response shared by all interventions, and the specific response of intervention j is $\mathbf{s}_j = \delta\mathbf{x}_j - \mathbf{m}$. The directions $\mathbf{v}_1, \ldots, \mathbf{v}_R$ are the $R = 48$ leading right singular vectors of the matrix $S_c = S - \mathbf{1}\mathbf{m}^\top$, centered across interventions, whose rows are the $\mathbf{s}_j$. They are then ordered by the share of the uncentered effects each carries, $w_k = \sum_j (\delta\mathbf{x}_j \cdot \mathbf{v}_k)^2$, which includes the shared response. All percentages in this appendix are fractions of the total squared effect $\|S\|_F^2 = \sum_j \|\delta\mathbf{x}_j\|_2^2$. On Norman, the 48 directions carry 95.6% of it, and the shared response, $|\mathcal{J}|\,\|\mathbf{m}\|_2^2$, carries 21%. The directions are computed once per dataset and shared by every encoder and block.

Sensitivity matrices at the anchors. For block L and anchor a, $J_L(\mathbf{x}_a)$ is the sensitivity of the block's readout to the anchor's log-normalized profile, including the encoder's input processing. For the differentiable encoders it is the Jacobian $\partial \mathrm{Enc}_L / \partial \mathbf{x}$. For scGPT and Tahoe-x1, whose native binning is piecewise constant and has no useful derivative, it is a finite-difference sensitivity matrix formed from one-count differences of the native encoder: column g is the change in the readout when one count of gene g is added to the anchor, divided by the resulting change in x_g, averaged

with the same slope for one count removed when the anchor expresses the gene. Every edited cell passes through the encoder's own forward pass, so the columns include the insertion of a token for a gene the anchor does not express and the re-binning of the other genes, and no surrogate derivative is involved. For brevity we refer to these finite-difference matrices as Jacobians throughout. The same Jacobians define the linearized encoder of Appendix H. For scPRINT the conversion from log-normalized values to counts sits inside the differentiated function, so J_L is expressed in log-normalized units for every encoder. Cells do not interact inside any encoder (Stack queries attend only to the fixed context and themselves), so for the other encoders row d of all N Jacobians is obtained from one backward pass of the sum of the d-th output over the anchors, and a block's Jacobians cost d_L backward passes; the one-count differences cost up to $2G$ forward passes per anchor and give every block at once.

Separable interventions. The image of the specific response of intervention j is $\mathbf{z}_j = \frac{1}{N}\sum_a J_L(\mathbf{x}_a)\,\mathbf{s}_j$, the first-order version of the predicted effect that the fit compares (Appendix H). The separable-intervention count is the participation ratio of the Gram matrix of these images,

$$n_{\mathrm{sep}}(L) = \frac{\left(\sum_i \mu_i\right)^2}{\sum_i \mu_i^2}, \qquad \mu_i \text{ the eigenvalues of } ZZ^\top, \quad Z = [\mathbf{z}_1 \cdots \mathbf{z}_{|\mathcal{J}|}]^\top, \tag{19}$$

with $n_{\mathrm{sep}}(L) = 0$ when all images vanish. The count is the effective dimensionality of the images: images that are parallel contribute a single dimension, and the count is one when all images lie on one line. Because the specific responses sum to zero over interventions, so do their images, and the count is at most $|\mathcal{J}| - 1$. As a ratio of moments of the eigenvalues, it does not change when the block's output is rescaled and does not depend on the units of J_L. In gene space, where $J_L = I$, it is 8.4 on Norman. Averaging over anchors before taking the Gram matrix follows the fit, which compares the anchor-averaged predicted effect; the per-anchor version, in which the Gram matrix is formed at each anchor and the ratio averaged, gives the same values to within 0.1 for every encoder.

Per-direction transmission. The gain of direction k is $g_k = \frac{1}{N}\sum_a \|J_L(\mathbf{x}_a)\mathbf{v}_k\|_2$, the length of the embedding change per unit expression change along $\mathbf{v}_k$, averaged over anchors. The transmission divides it by the mean gain of the leading five directions,

$$\tau_k(L) = \frac{g_k}{\bar{g}_{1:5}}, \tag{20}$$

so that a map with the same gain in every direction reads one throughout. Figure 1f draws a running mean of τ_k over five consecutive directions. The normalization removes the scale of the block, which grows with depth in residual networks, and the units of the Jacobian, so the curves of different blocks and encoders are directly comparable. The leading five directions carry 67% of the total squared effect, including most of the shared response; the tail of the curve, directions 16 to 48, carries 14% and contains additional variation between perturbed regulators.

Reference encoders. Identity has $J_L = I$, a separable count of 8.4 and $\tau_k = 1$ for every k; the one-count differences reproduce these values exactly when applied to Identity, which is the gate for the binned encoders. PCA on control cells keeps 7.7 separable interventions and reads 0.93 over the tail; the random projections keep 8.5 (width 512) and 6.3 (width 64) and read 0.98 over the tail, because a Gaussian map has the same expected gain along every direction. The four encoders keep between 2.4 and 4.5 separable interventions at their outputs and read between 0.61 and 0.69 over the tail directions (Table 3).

H Recovery through a linear approximation of the encoder

Section 2.5 compares recovery through an encoder layer with recovery through a local linear approximation around each anchor cell. This appendix specifies the approximation, its predicted and observed effects, and what the comparison measures.

The linearized encoder. At each of the N anchor cells, we construct a local sensitivity matrix $J_L(\mathbf{x}_a)$ of the layer's readout with respect to the anchor's log-normalized profile, as detailed in

Table 3: **Transmission at the output embedding, Norman.** Separable interventions (Eq. 19; 8.4 in gene space) and the mean per-direction transmission over the tail directions 16–48 (Eq. 20; one for a map that treats every direction alike), with Top-1 from Table 1.

Encoder	Separable interventions	Tail transmission	Top-1
Identity	8.4	1.00	0.63
PCA (512)	7.7	0.93	0.59
Random projection (512)	8.5	0.98	0.54
Random projection (64)	6.3	0.98	0.48
scGPT	2.9	0.66	0.11
scPRINT	4.5	0.69	0.24
Stack-Large	3.8	0.66	0.30
Tahoe-x1 (70M)	2.4	0.61	0.13

Appendix G. For Stack, scPRINT and the reference encoders, this is the Jacobian obtained by automatic differentiation. For scGPT and Tahoe-x1, whose native binning is discrete, we instead use one-count finite differences of the native forward pass. These matrices define a local linear approximation, rather than an exact Taylor expansion of the binned encoders. The transmission readouts and the linearized fits use the same matrices.

The predicted effect of intervention j is

$$\hat{\boldsymbol{\Delta}}_j^{\mathrm{lin}}(\theta) = \frac{1}{N}\sum_{a=1}^{N} J_L(\mathbf{x}_a)\big(\mathbf{x}_a'^{(j)}(\theta) - \mathbf{x}_a\big), \tag{21}$$

where $\mathbf{x}_a'^{(j)}(\theta)$ is the log-normalized profile of the edited anchor cell, produced by the same count edit used when fitting through the original encoder layer (Appendix C). Each anchor retains its own sensitivity matrix, so the approximation is linear in the expression change but can differ across cells. Once the matrices are formed, evaluating Eq. 21 requires only matrix products, without further encoder calls.

Targets through the same linearized encoder. We construct the targets using Eq. 21 and the same anchor-editing procedure. The observed mean profiles of the training cells replace the network's perturbed and control steady states, determining the count-edit factors applied to the anchors, and their resulting effects under Eq. 21 provide the training targets. Validation targets are constructed separately from the held-out mean profiles. The control target is zero, and the loss scale κ_{Enc} is recomputed from these targets as in Appendix D.

This construction makes the target and the prediction agree in expectation when the network reproduces the observed mean profiles. Their realized difference can remain nonzero because the count edits are sampled: each target uses a fixed draw, while predicted edits are sampled during fitting.

Fitting procedure. The mechanistic model, solver, control and sparsity terms, hyperparameters, anchor cells and cell splits are unchanged from the fit through the original encoder layer (Appendix K). The changes are the encoder map, the construction of its targets and the corresponding loss scale. We denote the resulting gene-space recovery by $\mathrm{rec}_{\mathrm{lin}}(L)$.

For a linear encoder, the approximation reproduces the encoder map exactly. This does not imply identical fits: the original fit uses mean embeddings of observed cells as targets, whereas the linearized fit uses edited anchors constructed from observed mean profiles. Even for Identity, these targets can differ, because count editing need not reproduce the observed mean log-normalized profile exactly.

What the comparison measures. Recovery through the original encoder layer, $\mathrm{rec}_{\mathrm{true}}(L)$, measures recovery when predicted and observed effects are compared through that layer. Recovery through the linearized encoder measures what the layer's local sensitivities support under targets constructed through the same anchor-editing procedure. Higher linearized recovery is consistent with limitations arising from the layer's nonlinear response to finite edits, but can also reflect the closer agreement between target and prediction construction. Conversely, higher recovery through

the original layer can indicate useful response components that the local approximation does not capture. The difference therefore does not isolate nonlinearity alone, and we interpret it alongside the transmission readouts, which characterize the local sensitivity matrices independently of mechanistic fitting. Identity provides a reference for recovery without compression; its linearized fit can differ from its original fit because of the target construction, although the encoder map itself is unchanged.

Practical notes. The sensitivity matrices are computed once and reused throughout fitting. Genes outside an encoder's vocabulary contribute zero columns. For scGPT and Tahoe-x1, one-count edits of the anchors provide sensitivities at every measured block in the same forward passes, at a cost of up to $2G$ forward passes per anchor. For the differentiable encoders, the Jacobians of a layer take d_L backward passes, jointly over the anchors (Appendix G). During fitting, the count edit uses fresh draws at each training step and a fixed draw for validation. After the sensitivity matrices have been constructed, the linearized fits require no further encoder evaluations.

I DATASETS AND PREPROCESSING

We use four Perturb-seq datasets, one CRISPR activation screen and three CRISPR interference screens, in three cell lines. Every dataset goes through the same steps in the same order, so that the mechanistic model, the anchor cells and the scores are defined identically across them. Table 4 summarizes what is retained.

Table 4: **Datasets after preprocessing.** Interventions are single-gene perturbations retained for fitting; control cells are the unperturbed cells used for the control mean, the anchor cells, PCA and the Stack context; the gene panel is the highly variable genes plus every perturbed gene.

Dataset	Cell line	Modality	Interventions	Cells per intervention, median	Control cells	Gene panel
Norman 2019 (Norman et al., 2019)	K562	CRISPRa	102 (+129 doubles)	497	11,855	2,553
Replogle 2022 (Replogle et al., 2022)	K562	CRISPRi	58	123	10,000	2,544
Replogle 2022 (Replogle et al., 2022)	RPE1	CRISPRi	67	86	10,000	2,549
X-Atlas/Orion (Huang et al., 2025)	HCT116	CRISPRi	100	104	10,000	2,504

Sources. Norman 2019 and the two Replogle 2022 screens (the K562 essential-gene screen and the RPE1 screen) are taken from scPerturb (Peidli et al., 2024) as count matrices with guide annotations. X-Atlas/Orion is released as parquet shards of sparse counts, one row per cell; we stream the HCT116 shards one at a time, keep the cells whose target is a transcription factor together with a fixed number of non-targeting control cells per shard, so that controls are spread over all batches, and write a single count matrix with the same layout as the scPerturb files. All cells of the retained interventions pass the release's guide filter.

Which interventions are fitted. On Norman, all 102 single-gene interventions with at least three cells are fitted; its 129 two-gene interventions are held out entirely. For the CRISPRi screens an intervention is retained when its target is a single gene in the Lambert et al. census of human transcription factors (Lambert et al., 2018), is measured in the expression matrix, has at least 25 cells, and is knocked down by at least 30%, the knock-down being one minus the ratio of the target's mean normalized count in perturbed cells to that in control cells. The last filter exists because the model clamps the target at its observed perturbed level: a guide that does not move its target gives a clamp equal to the control level, so the model can only predict no change for that intervention and the condition is noise for every encoder alike. On X-Atlas, $1{,}188$ interventions pass these filters, so we keep the 100 with the largest transcriptome-wide response, measured as the squared norm of the mean shift on the highly variable genes, with the target gene excluded and the sampling variance of the two means subtracted so that small conditions are not favoured by noise. Beyond the minimum-cell threshold, cell number is not used to rank interventions, because in a knock-down screen it tracks fitness and would select the interventions that do least. All selections use every cell of the intervention and never see an encoder, so the set is the same for every row of Table 1.

Control cells. All control cells are used on Norman. On the other datasets, which have tens of thousands, we draw $10{,}000$ at random once, from a fixed seed. The library size of every cell is

the sum of its raw counts over all genes of the file, taken before any gene filtering; it is the depth scPRINT conditions on and defines the size factor of the count edit.

Genes and normalization. Genes with five or fewer counts over the retained cells are removed. Counts are normalized to the median library size and $\log(1+x)$-transformed; this is the coordinate system of the mechanistic model. The gene panel is the $2{,}500$ most highly variable genes of the normalized matrix, selected on control and perturbed cells together, plus every perturbed gene that the selection did not already include, so that each clamp coordinate is in the panel; on Norman this adds 53 genes. Raw counts are kept on the same panel, cell-aligned, for the count edit and for scPRINT. Duplicate symbols keep their first occurrence. Ensembl identifiers are read from the file where present and otherwise looked up once by symbol.

Clamp levels and splits. The clamp level c_j is the mean log-normalized expression of the target over all cells of intervention j. Within each intervention, 20% of the cells (at least one) are held out; the split is drawn from the run seed, so the three seeds of every fit differ in which cells are held out and in nothing else. The two-gene interventions of Norman are split the same way with a seed offset by one, and their held-out cells supply the double-gene scores. Observed effects for fitting are computed from the training cells of each intervention and effects for scoring from the held-out cells, both relative to the mean of all control cells.

Processed data are cached. The processed container of each dataset is written once, keyed by every loader argument, the transcription-factor list and the file identity, and reused by every fit, so that no two fits of a dataset can differ in their preprocessing.

J ENCODERS: INPUT FORMATS, DIFFERENTIABLE BINNING AND CONTROLS

Every encoder is a frozen module that maps one cell to one embedding. The mechanistic model never sees the encoder's internals: it only needs the embedding of an edited anchor cell and the derivative of that embedding with respect to the cell's expression. This appendix records what each encoder receives, how the discrete steps of its input processing are made differentiable without changing the forward value of the wrapper, and how closely that forward value matches the original library, how intermediate layers are read, and how the two negative controls are built. Table 5 summarizes the input conventions.

Table 5: **Input conventions of the encoders.** Tokens: which genes enter the model. Values: what is attached to each gene. Layers: the number of transformer blocks read in the layerwise analysis. Coverage: genes of the Norman panel ($2{,}553$ genes) present in the encoder's vocabulary.

Encoder	Tokens	Values	d	Layers	Coverage
Identity	all panel genes	log-normalized expression	G	–	2,553
PCA	all panel genes	log-normalized expression	512	–	2,553
Random projection	all panel genes	log-normalized expression	512 / 64	–	2,553
scGPT	detected genes	per-cell quantile bins	512	12	2,332
scPRINT	vocabulary genes	raw counts, library size	512	12	1,850
Stack-Large	fixed 15,012-gene panel	normalized counts	1,600	9	1,558
Tahoe-x1 (70M)	detected genes	per-cell quantile bins	512	12	2,524

J.1 WHAT AN EDITED CELL BECOMES AT EACH ENCODER'S INPUT

The count edit of Appendix C yields, for every anchor cell a and intervention j, a vector of integer counts $\mathbf{y}_a'^{(j)}$ with the size factor s_a of its anchor. Each encoder then receives this cell in the form its own preprocessing expects: log-normalized expression $\log(1+y/s_a)$ for Identity, PCA, the random projections, scGPT, Stack and Tahoe-x1, and raw counts with the anchor's full-transcriptome library size for scPRINT. The observed cells go through the same conversion, so predicted and observed effects are read through one input pipeline. The binned encoders operate on ranks within a cell, so the size factor does not affect their tokens.

J.2 REFERENCE ENCODERS

Identity returns its input, so the objective reduces to a comparison of gene-space effects. PCA is fitted once per dataset on the log-normalized control cells only, centered at the control mean, and projects onto the leading 512 right singular vectors. The random projections apply a fixed Gaussian matrix with entries of variance $1/d$ to the centered profile, at widths $d = 512$ and $d = 64$; the matrix is drawn once from a fixed seed and shared across seeds and datasets of the same panel size.

J.3 SCGPT

We use the whole-human checkpoint, with genes matched to its vocabulary by symbol. The model's own embedding routine tokenizes only the genes detected in a cell, attaches to each a bin index between 1 and 50 obtained from the quantiles of that cell's detected values, and places the pad value on the class token whose output is the cell embedding. This map from a value to a bin is piecewise constant, and the library breaks ties, which in a sparse cell involve some two hundred genes sharing one count, by assigning a bin drawn at random within the tied range.

We keep this convention and make two changes that leave the forward value within the library's own variability. First, ties are broken by a fixed offset per gene, $\xi_g \sim \mathcal{U}(-0.15, 0.15)$ in log-normalized units, drawn once and applied to every cell, to observed and edited cells alike. An unedited cell then already has its tied genes spread across bins, as under the library's random rule, and an edit that changes one gene's count reorders it continuously as its value passes those of its neighbors. Second, the bin is the continuous position of the value among the cell's quantile edges,

$$p(v) = k + \frac{v - e_k}{e_{k+1} - e_k} \qquad \text{for } e_k \le v < e_{k+1}, \tag{22}$$

where $e_1 \le \cdots \le e_{50}$ are the quantiles of the cell's detected values after the offset; $\lfloor p(v) \rfloor$ is the library's bin, and the edges are treated as constants in the backward pass so that every detected gene carries a gradient. Cells embedded this way agree with the library's embedding of the same cells to a per-cell cosine of 0.9975, against 0.9979 between two draws of the library's own tie-breaking. The set of tokens is fixed by which genes are detected; the count edit changes it only by switching genes on or off, which is the discrete step the encoder itself performs. Intermediate layers are read at the class-token row of each transformer block.

Gradient through the bins. Because the count edit is discrete, the exact derivative of the expected embedding with respect to a fold change is a difference over one count. For gene g of anchor a under intervention j, write $y = y_{ag}$, $s = s_a$, $\mu = \mu_g$ and $r = r_{jg}$, let y' be the edited count, and let $\Delta^+ f$ and $\Delta^- f$ be the changes in the cell's embedding when one count of the gene is added or removed. Then

$$\frac{\partial}{\partial r}\,\mathbb{E}\big[f(y')\big] = \begin{cases} \mathbb{E}\big[\,y'\,\Delta^- f(y') + (y - y')\,\Delta^+ f(y')\,\big], & r < 1 \text{ (thinning)}, \\ s\mu\;\mathbb{E}\big[\,\Delta^+ f(y')\,\big], & r > 1 \text{ (addition)}, \end{cases} \tag{23}$$

which follows from differentiating a binomial expectation in its success probability and a Poisson expectation in its rate. Evaluating each $\Delta^{\pm} f$ exactly would take one encoder pass per gene and cell. We instead compute how far one count moves the gene's bin position, at the edited cell's own edges,

$$\delta^+ = p\big(v(y'+1)\big) - p\big(v(y')\big), \qquad \delta^- = p\big(v(y')\big) - p\big(v(y'-1)\big), \tag{24}$$

$$v(y) = \log(1 + y/s) + \xi_g,$$

with positions clipped to $[1, 50]$, and set $\Delta^{\pm} f \approx (\partial f/\partial p_g)\,\delta^{\pm}$, where $\partial f/\partial p_g$, the sensitivity of the embedding to the gene's bin position, comes from automatic differentiation. Removing a gene's last count deletes its token and is given $\delta^- = 0$; a gene absent from the edited cell receives no gradient. At $r = 1$ the two one-sided terms are summed. In practice this rescales the straight-through term of the count edit (Appendix C) entry by entry, so it needs no additional encoder pass.

On eight Norman anchors, for three interventions at two operating points and two readout directions, this gradient has a cosine of 0.67–0.90 with the exact derivative of Eq. 23, computed from one-count differences of the full encoder on the same sampled cells, and its component along the exact gradient is 0.84–0.95 of the exact gradient's length.

J.4 TAHOE-X1

We use the 70M checkpoint with genes matched by Ensembl identifier. Its embedding routine also tokenizes only detected genes and bins each value by the quantiles of the cell's detected values, but its rule is deterministic: the bin is one plus the number of quantile grades strictly below the value. Every gene tied at the lowest count receives bin 1, and a gene one count higher typically lands about twenty bins above it, so the map from a value to a bin is a staircase with few, tall steps. Observed and edited cells, which are integer-count cells, are binned exactly as the library does. For the backward pass, the zero derivative of the step function is replaced by a unit slope, and the resulting straight-through gradient of the count edit (Appendix C) is rescaled during fitting against the exact derivative of Eq. 23. At every step, for two interventions, eight anchor cells and a sample of about 32 genes, we evaluate one-count differences of the native encoder on two fresh draws of the edited cells, and set one scale per edit direction (removal, addition, none) and per level of the gene's count in the anchor (0, 1, 2, 3–6, $\geq$ 7) by least squares between the exact and straight-through derivatives, smoothed over steps. The forward value is unchanged.

Cells are grouped by token count so that each batch is rectangular and no padding is needed; this reproduces one-at-a-time encoding to 10^{-5}. Sequences are capped at $2{,}047$ gene tokens plus the class token; a cell with more detected genes keeps those with the largest values. The cell embedding is the class-token output, and intermediate layers are read at the class-token row of each of the 12 blocks, before the final normalization.

J.5 SCPRINT

scPRINT reads raw counts and conditions on a depth token computed from the cell's library size. The wrapper therefore receives integer counts on the panel together with the library size summed over all genes of the transcriptome before any gene filtering; the model applies its own $\log_2(1+x)$ transform internally. Genes are matched by Ensembl identifier. The cell embedding is the model's own cell embedding, the mean over its cell-level tokens, and intermediate layers are read by the same pooling at each block. Because counts are already the model's native input, no relaxation is needed: the edit acts on counts and the gradient passes through the model's continuous value encoder.

J.6 STACK

Stack-Large is an in-context model: it attends across a set of cells, so a cell's embedding depends on the cells it is read with. We fix a reference context of 256 control cells, drawn once per dataset from a fixed seed and detached, and embed each query cell against it with an attention mask under which context cells attend only to the context and each query attends only to the context and itself. A query's embedding is then identical to encoding it alone against the context, while many queries are processed in one batch. The model consumes a dense vector over its own panel of $15{,}012$ genes: panel genes are matched by symbol, panel columns absent from the dataset are held at zero, and dataset genes absent from the panel are dropped. It expects normalized counts and applies $\log(1+x)$ itself, so the wrapper converts the log-normalized input back to normalized counts. The embedding is the query rows of the final attention layer, flattened to $1{,}600$ dimensions, and intermediate layers apply the same readout to each of the 9 attention layers. The value encoder is linear followed by a smooth nonlinearity, so no relaxation is needed.

J.7 READING AN INTERMEDIATE LAYER

For the layerwise analysis of Section 2.5 we wrap an encoder so that its output is the readout at block L described above. The observed effects at that layer are computed from the real cells through the same wrapper and cached, so the target of the fit and its prediction live in the same layer-L space. Nothing in the mechanistic model or the objective changes.

J.8 CACHED OBSERVED EFFECTS

The observed effect of every intervention is the difference of mean embeddings of real cells (Eq. 3). These embeddings are computed once per encoder and dataset and cached, keyed by the encoder and wrapper version, the dataset and gene panel, the input format, the vocabulary mapping and,

where relevant, the layer, the reference context and the random-weight seed, so that any change to the encoding invalidates the cache.

J.9 NEGATIVE CONTROLS

Shuffled interventions. The observed effect of each intervention is replaced by that of another intervention through a fixed derangement, the same for the training and held-out cells, with the encoder, the edit, the loss scale and every hyperparameter unchanged. Only the pairing between the gene that was perturbed and the effect that was observed is destroyed; the clamp levels still belong to the true intervention. A fit through this control measures what the pipeline recovers when the responses carry no information about their interventions.

Random weights. Every module of scGPT that defines its own initialization is re-drawn from that initialization under a fixed seed, and the input processing, the soft binning and the fit are unchanged. The observed effects are recomputed through the re-initialized model, so predicted and observed effects are again read through the same encoder. This control keeps the architecture and the tokenization of scGPT and removes only what pretraining put into the weights.

J.10 CHECKS BEFORE USE

Each wrapper passes three checks before any fit is read through it: gradients reach the input from the embedding and are non-zero; encoding cells in a batch agrees with encoding them one at a time; and the objective rewards the true answer through the encoder in its native format (Appendix E).

K HYPERPARAMETERS, SOLVER AND COMPUTE

Table 6: **Fitting hyperparameters.** The loss and all of its weights are identical for every encoder and dataset. Only the learning rate differs between datasets; it was selected per dataset on the Identity encoder (held-out SGI) and then applied unchanged to all encoders, layers and controls of that dataset.

	Norman	Replogle K562	Replogle RPE1	X-Atlas HCT116
Selected on Identity, per dataset				
Learning rate (AdamW)	10^{-3}	10^{-3}	3×10^{-4}	3×10^{-4}
Shared by all datasets and encoders				
Sparsity (ℓ_1) weight λ_1	5×10^{-6}			
Control-state weight λ_{ctrl}	0.1			
Weight decay (adjacency only)	10^{-4}			
Gradient clipping (norm)	1.0			
Maximum epochs / patience	2000 / 300 epochs, on the validation perturbation loss			
Anchor cells N	32 control cells, fixed per dataset			
Applying the predicted change	count edit (Appendix C), $\epsilon = 10^{-2}$, fold change capped at 100			
Validation split	20% of the cells of every intervention			
Seeds	0, 13, 42 (cell split only)			
Data				
Interventions fitted	102	58	67	100 (largest effect)
Minimum knock-down of the target	n/a (activation)	0.3	0.3	0.3
Gene panel	2,553	2,544	2,549	2,504
Held-out double interventions	129	–	–	–

Optimizer and initialization. We use AdamW with default moment parameters and two parameter groups: the entries of A receive weight decay, while the basal rates and the log-parameterized γ, n and K do not, so that the kinetic scalars are not pulled away from their initial values by the regularizer. The columns of A for perturbed regulators are initialized from a zero-mean Gaussian of standard deviation 10^{-3}, from a seed that is the same for every run, and all other columns are masked to zero in the forward pass, so they receive no gradient and remain at zero; the mask is also reapplied after every step as a safeguard. The basal rates are initialized as γ times the control mean, so that the network without regulation has the control mean as its steady state, with

$\gamma = 0.1$, $n = 2$ and $K = 0.5$ at initialization. The three scalars are learned in log space within $\gamma, K \in [e^{-3}, e^{2}] \approx [0.05, 7.39]$ and $n \in [e^{-1}, e^{2.5}] \approx [0.37, 12.2]$.

Solver. The steady state is found by damped fixed-point iteration with damping $\alpha = 0.5$, the state bounded to $[10^{-6}, 50]$ and the clamp reimposed after every step. The solver runs 200 iterations per solve for the first 20 epochs and 50 thereafter, each solve warm-started from the same intervention's state at the previous epoch; the unclamped state has its own warm start. The backward pass is the implicit solve of Appendix F. The sparsity penalty is applied from the first epoch without a warm-up.

Selection of the learning rate. For each dataset we fitted Identity at 10^{-3} and 3×10^{-4}, with one to three seeds per rate, and kept the rate with the higher held-out SGI relative to the reference row. Replogle RPE1 and X-Atlas HCT116 preferred the smaller rate. No other hyperparameter was tuned per dataset, and none was tuned on any encoder other than Identity.

Number of anchor cells. Each intervention's predicted effect is the mean over N edited anchor cells passed through the encoder, so the cost of a fit through an scFM grows linearly with N. We selected the number of anchor cells using Identity on a small subset of Norman's double-gene interventions (seed 42). Held-out DGI was 0.70 with 8 anchors, 0.78 with 32, 0.79 with 128, 0.80 with 512 and 0.77 with 2,048: beyond 32 anchors, a 64-fold increase changes DGI by at most 0.02. We therefore use $N = 32$, which keeps fits through every scFM computationally feasible.

Compute. Every fit runs on one NVIDIA H200 GPU. The cost of a fit is dominated by the encoder, which is evaluated on N edited cells per intervention twice per step (forward, and recomputation in the backward pass). Fits through Identity and the linear baselines take 5–19 minutes per seed, through Stack-Large 16–21 minutes, through scGPT 1.3–5.7 hours, through Tahoe-x1 3.0–6.8 hours and through scPRINT 4.0–12.4 hours. The results reported here amount to approximately 500 H200 GPU-hours, and we estimate as much again for the whole project including prototypes and failed runs.

L EVALUATION

All scores are computed in gene space from the fitted mechanistic model and never through an encoder, so that networks recovered through different encoders are compared on the same quantity. This appendix defines the predicted and observed effects, the three scores of Table 1, the reference row, and what the seeds vary.

Predicted and observed effects. From the saved parameters θ we solve the unclamped steady state $\mathbf{x}^{*,(\mathrm{ctrl})}$ and, for each intervention, the clamped steady state of Eq. 2; the predicted effect is their difference. For a two-gene intervention both targets are clamped at their observed mean levels in the cells of that intervention, and the model is otherwise unchanged. The observed effect is the mean log-normalized profile of the held-out cells of the intervention minus the mean of all control cells. The coordinates of the perturbed genes are excluded from every score, because the clamp imposes their values.

SGI and DGI. For each fitted intervention we compute the Pearson correlation across genes between the predicted and the observed effect; the single-gene intervention score (SGI) is its mean over interventions. The double-gene intervention score (DGI) is the same mean over the held-out two-gene interventions of Norman, none of which is used in fitting.

Top-1 retrieval. We subtract from both the predicted and the observed effects their mean over interventions, which removes the response shared by all interventions, and remove all perturbed-gene coordinates. We then compute the cosine similarity between the predicted effect of intervention j and the observed effect of every intervention. Top-1 is the fraction of interventions for which the correct observed effect has the highest similarity; ties count against the model. Chance is $1/|\mathcal{J}|$.

Reference row: response reproducibility. The observed effect is a mean over a few held-out cells and carries sampling noise that no model can predict, so a score below one does not by itself separate

what a model misses from the noise in the held-out cells. The reference row of Table 1 separates the two: it predicts each intervention's held-out effect by the mean effect of its own training cells, with the same split, the same masking and the same retrieval rule as the fitted models. It is an unbiased but noisy estimate of the true response and therefore a slightly conservative ceiling; a noise-free ceiling can be obtained from the correlation between two disjoint held-out-sized samples of the training cells, whose square root is the correlation a perfect predictor would reach, and it lies a few points higher. The reference row shows in particular that the CRISPRi datasets, with fewer cells per intervention and weaker responses than Norman, cap SGI well below one, and that the same recovered network reads lower there for that reason.

Model selection and seeds. Each fit is stopped on the validation perturbation loss, computed through the encoder from the held-out cells, and the parameters at the best validation loss are kept; the gene-space scores are computed afterwards from those parameters. The three seeds 0, 13 and 42 change the held-out cells and nothing else: the anchor cells, the initialization of the network and the regenerated noise of the count edit are fixed across seeds. The reported standard deviation therefore measures sensitivity to the cell split.

M STRUCTURAL VALIDATION DETAILS

This appendix details the structural validation of Figure 2. We first describe how the recovered targets of each TF are compared with independent references, and how we determine which TFs can be checked at all. We then report the references for the network recovered through Identity, and finally summarize the networks recovered through each encoder.

M.1 PROCEDURE

For each TF, we take the 25 targets with the largest $|A_{g,\mathrm{TF}}|$, excluding the TF itself, whose expression is imposed by the clamp. We test their overlap with every gene set of a reference library using a hypergeometric test, with the gene panel ($2{,}553$ genes) as background. Gene sets are restricted to the panel and kept if they contain between 5 and 500 panel genes. The p-values are Bonferroni-corrected over the sets tested in each library: $m = 6$ lineage signatures, 740 ChEA 2022 sets, 816 ENCODE TF ChIP-seq 2015 sets, 49 MSigDB Hallmark sets and 574 Reactome 2022 pathways. We use the libraries distributed with Enrichr (Kuleshov et al., 2016) and compute the tests ourselves. Each library is corrected separately, because the libraries are distinct types of reference rather than one family of tests. For Hallmark and Reactome, disease and generic catch-all pathways are removed before testing (Appendix N). The lineage library contains eight hematopoietic programs; the Megakaryocyte and B cell signatures retain fewer than five panel genes and are dropped.

The own reference of a TF. The own reference sets of a TF are the ChEA and ENCODE sets named after it and, for TFs in the census of Lambert et al. (2018), the lineage signatures that contain it. The restriction to transcription factors matters because the signatures also list marker genes, such as *SLC4A1* in the erythroid program, whose targets have no reason to belong to that lineage. A TF is validated in a network when at least one of its own reference sets reaches $p_{\mathrm{bonf}} < 0.05$. Of the 102 perturbed genes on Norman, 19 have at least one own reference set.

Which references apply to this screen. A reference describes what a TF does in the cells in which it was measured, and many ChIP-seq sets were obtained in cell types unrelated to K562. To identify the references that describe these K562 cells, we apply the same test to the data themselves, without any network: for each TF, we test the 25 genes whose mean expression changes most in the cells in which the TF was perturbed. For six of the 19 TFs, these genes overlap the TF's own reference: CEBPA, CEBPE, IRF1, JUN, KLF1 and SPI1, with CEBPE and JUN only narrowly ($p_{\mathrm{bonf}} = 0.04$). For ten of the remaining 13, every reference set was obtained in another cell type, for example intestine and liver for HNF4A, prostate and breast for FOXA1, and embryonic stem cells for TBX3. The size of the response does not explain the difference: PRDM1, CEBPB and ARID1A produce some of the strongest responses in the screen. Figure 2c counts validation over the six TFs whose references apply.

Seeds. Scores are computed for every seed of every fit, and Figure 2b shows their mean. The networks drawn in Figure 2a and the tables below use seed 0.

M.2 IDENTITY: LINEAGE SIGNATURES

Table 7: Enrichment of the top-25 targets recovered through Identity against each lineage signature, for the four TFs of Figure 2a. Bold: the TF's own lineage. $m = 6$.

TF	Lineage signature	Overlap	Fold	p_{bonf}
KLF1	**Erythroid Lineage**	**11/21**	**53.5×**	**1.2e-17**
KLF1	Interferon Response	1/21	4.9×	1.0
KLF1	Myeloid Granulocyte Lineage	0/13	0.0×	1.0
KLF1	Myeloid Monocyte Lineage	0/23	0.0×	1.0
KLF1	Lymphoid T Cell Lineage	0/13	0.0×	1.0
KLF1	Cell Cycle	0/11	0.0×	1.0
SPI1	**Myeloid Monocyte Lineage**	**9/23**	**40.0×**	**7.4e-13**
SPI1	Myeloid Granulocyte Lineage	1/13	7.9×	0.72
SPI1	Erythroid Lineage	1/21	4.9×	1.0
SPI1	Lymphoid T Cell Lineage	0/13	0.0×	1.0
SPI1	Interferon Response	0/21	0.0×	1.0
SPI1	Cell Cycle	0/11	0.0×	1.0
IRF1	**Interferon Response**	**7/21**	**34.0×**	**2.2e-9**
IRF1	Erythroid Lineage	2/21	9.7×	0.10
IRF1	Myeloid Monocyte Lineage	1/23	4.4×	1.0
IRF1	Myeloid Granulocyte Lineage	0/13	0.0×	1.0
IRF1	Lymphoid T Cell Lineage	0/13	0.0×	1.0
IRF1	Cell Cycle	0/11	0.0×	1.0
CEBPA	**Myeloid Granulocyte Lineage**	**3/13**	**23.6×**	**1.3e-3**
CEBPA	Myeloid Monocyte Lineage	3/23	13.3×	7.7e-3
CEBPA	Erythroid Lineage	0/21	0.0×	1.0
CEBPA	Lymphoid T Cell Lineage	0/13	0.0×	1.0
CEBPA	Interferon Response	0/21	0.0×	1.0
CEBPA	Cell Cycle	0/11	0.0×	1.0

For each of the four TFs, the strongest lineage hit is its own lineage: KLF1 recovers 11 of the 21 panel genes of the erythroid signature ($53.5\times$, $p_{\mathrm{bonf}} = 1.2 \times 10^{-17}$), SPI1 9 of 23 monocyte genes ($40.0\times$, 7.4×10^{-13}), IRF1 7 of 21 interferon-response genes ($34.0\times$, 2.2×10^{-9}) and CEBPA 3 of 13 granulocyte genes ($23.6\times$, 1.3×10^{-3}). Every other signature has $p_{\mathrm{bonf}} \geq 0.1$, with one exception: the CEBPA targets also enrich the monocyte signature ($13.3\times$, $p_{\mathrm{bonf}} = 7.7 \times 10^{-3}$), which is expected for a regulator of both myeloid branches.

M.3 IDENTITY: CHEA 2022 CHIP-SEQ TARGETS

The own ChIP-seq set is the strongest ChEA hit for SPI1 (NB4 cells, $p_{\mathrm{bonf}} = 1.3 \times 10^{-3}$) and for IRF1 (monocytes, 1.4×10^{-5}). For KLF1 the strongest hit is the EKLF set (2.5×10^{-4}); EKLF, erythroid Krüppel-like factor, is another name for KLF1, although the naming rule of Appendix M.1 does not count it. The ChEA sets named CEBPA come from liver and are not enriched. The next hits reflect known co-regulators. Four of the SPI1 hits are members of the heptad complex that co-regulates myeloid commitment in hematopoietic progenitors (LYL1, RUNX1, TAL1 and FLI1 in HPC-7 cells) (Wilson et al., 2010), and the second IRF1 hit is IRF8, its closest paralog.

M.4 IDENTITY: ENCODE TF CHIP-SEQ TARGETS

ENCODE contains ChIP-seq in K562, the cell line of the Norman screen, for IRF1 but not for KLF1 or CEBPA. The IRF1 targets recover 12 of the 291 panel genes bound by IRF1 in K562 ($p_{\mathrm{bonf}} = 4.2 \times 10^{-3}$). The strongest KLF1 hits are erythroid co-regulators, TAL1 and GATA1 in erythroid cells, and FOSL1 in K562, although none reaches $p_{\mathrm{bonf}} < 0.05$. The own SPI1 set is its strongest

Table 8: Top five ChEA 2022 sets for the top-25 targets recovered through Identity. The cell type and species are part of each set name. Bold: sets of the TF itself. $m = 740$.

TF	ChEA set	Overlap	Fold	p_{bonf}
KLF1	EKLF 21900194 ChIP-Seq ERYTHROCYTE Mouse	9/107	8.6×	2.5e-4
KLF1	**KLF1 20508144 ChIP-Seq FETAL-LIVER-ERYTHROID Mouse**	**6/112**	**5.5×**	**0.41**
KLF1	NANOG 20526341 ChIP-Seq ESCs Human	7/166	4.3×	0.57
KLF1	MAF 26560356 Chip-Seq TH1 Human	8/231	3.5×	0.82
KLF1	TP53 22573176 ChIP-Seq HFKS Human	6/129	4.7×	0.88
SPI1	**SPI1 23547873 ChIP-Seq NB4 Human**	**14/377**	**3.8×**	**1.3e-3**
SPI1	LYL1 20887958 ChIP-Seq HPC-7 Mouse	7/89	8.0×	0.01
SPI1	RUNX1 20887958 ChIP-Seq HPC-7 Mouse	7/123	5.8×	0.09
SPI1	TAL1 20887958 ChIP-Seq HPC-7 Mouse	8/191	4.3×	0.22
SPI1	FLI1 20887958 ChIP-Seq HPC-7 Mouse	7/161	4.4×	0.47
IRF1	**IRF1 21803131 ChIP-Seq MONOCYTES Human**	**7/35**	**20.4×**	**1.4e-5**
IRF1	IRF8 22096565 ChIP-ChIP GC-B Human	6/32	19.1×	2.6e-4
IRF1	MAF 26560356 Chip-Seq TH1 Human	10/231	4.4×	0.02
IRF1	ELF3 26769127 Chip-Seq PDAC-Cell Line Human	9/194	4.7×	0.04
IRF1	TCF21 26020271 ChIP-Seq SMOOTH MUSCLE Human	9/239	3.8×	0.19
CEBPA	SMAD2/3 21741376 ChIP-Seq ESCs Human	8/188	4.3×	0.20
CEBPA	VDR 24763502 ChIP-Seq THP-1 Human	7/144	5.0×	0.24
CEBPA	SPI1 23547873 ChIP-Seq NB4 Human	11/377	3.0×	0.30
CEBPA	GATA1 22383799 ChIP-Seq G1ME Mouse	7/181	3.9×	0.95
CEBPA	ZNF652 21678463 ChIP-ChIP ZR75-1 Human	1/9	11.3×	1.0

Table 9: Top five ENCODE TF ChIP-seq 2015 sets for the top-25 targets recovered through Identity. Bold: sets of the TF itself. $m = 816$.

TF	ENCODE set	Overlap	Fold	p_{bonf}
KLF1	TAL1 G1E-ER4 mm9	6/80	7.7×	0.07
KLF1	TAL1 erythroblast mm9	6/81	7.6×	0.08
KLF1	ZNF274 K562 hg19	6/88	7.0×	0.12
KLF1	GATA1 erythroblast hg19	8/177	4.6×	0.14
KLF1	FOSL1 K562 hg19	6/92	6.7×	0.15
SPI1	**SPI1 GM12878 hg19**	**9/215**	**4.3×**	**0.09**
SPI1	EP300 CH12.LX mm9	9/258	3.6×	0.37
SPI1	BRCA1 GM12878 hg19	6/115	5.3×	0.52
SPI1	GTF2F1 K562 hg19	6/125	4.9×	0.82
SPI1	JUND MEL cell line mm9	3/29	10.6×	1.0
IRF1	MYC CH12.LX mm9	9/112	8.2×	4.0e-4
IRF1	**IRF1 K562 hg19**	**12/291**	**4.2×**	**4.2e-3**
IRF1	RCOR1 CH12.LX mm9	8/143	5.7×	0.03
IRF1	TAL1 MEL cell line mm9	9/216	4.3×	0.09
IRF1	USF2 MEL cell line mm9	7/124	5.8×	0.10
CEBPA	RAD21 K562 hg19	9/176	5.2×	0.02
CEBPA	RFX5 K562 hg19	7/129	5.5×	0.13
CEBPA	CUX1 K562 hg19	7/129	5.5×	0.13
CEBPA	POLR2A K562 hg19	12/420	2.9×	0.18
CEBPA	CTCF K562 hg19	10/295	3.5×	0.20

ENCODE hit but does not reach significance ($p_{\text{bonf}} = 0.09$); it comes from a lymphoblastoid line, less related to K562 than the NB4 set of Table 8.

M.5 Identity: MSigDB Hallmark and Reactome

The pathway libraries agree with the lineage signatures where they contain a matching program: heme metabolism for KLF1 (8 of 42 genes, $p_{\text{bonf}} = 1.2 \times 10^{-7}$), the interferon-gamma response and antigen presentation (ER-phagosome pathway) for IRF1, and transcriptional regulation of gran-

Table 10: Strongest MSigDB Hallmark and Reactome 2022 hit for the top-25 targets recovered through Identity, where one reaches $p_{\mathrm{bonf}} < 0.05$. $m = 49$ (Hallmark) and $m = 574$ (Reactome).

TF	Library	Top hit	Overlap	Fold	p_{bonf}
KLF1	Hallmark	Heme Metabolism	8/42	19.5×	1.2e-7
KLF1	Reactome	(none at $p_{\mathrm{bonf}} < 0.05$)	–	–	n.s.
SPI1	Hallmark	(none at $p_{\mathrm{bonf}} < 0.05$)	–	–	n.s.
SPI1	Reactome	(none at $p_{\mathrm{bonf}} < 0.05$)	–	–	n.s.
IRF1	Hallmark	Interferon Gamma Response	7/60	11.9×	4.7e-5
IRF1	Reactome	ER-Phagosome Pathway	5/23	22.2×	1.0e-3
CEBPA	Hallmark	(none at $p_{\mathrm{bonf}} < 0.05$)	–	–	n.s.
CEBPA	Reactome	Transcriptional Regulation Of Granulopoiesis	4/10	40.8×	8.3e-4

ulopoiesis for CEBPA (4 of 10 genes, 8.3×10^{-4}). Neither library contains a coarse monocyte program, and no hit reaches significance for SPI1. This gap motivated the curated lineage library (Appendix N).

M.6 Identity: TRRUST

TRRUST v2 curates human TF-target pairs from the literature (Han et al., 2018). Within the gene panel it annotates 16 targets of IRF1, 16 of CEBPA, 23 of SPI1 and one of KLF1. The top-25 targets recovered through Identity contain three of the IRF1 targets (*PSMB9*, *TAP1*, *TAPBP*; hypergeometric $p = 4.3 \times 10^{-4}$) and two of the CEBPA targets (*CSF3R* and *SPI1*; $p = 0.01$), one of the SPI1 targets ($p = 0.20$) and none for KLF1. TRRUST annotations are sparse for erythroid and myeloid regulation in K562, and we report them as a secondary reference.

M.7 Networks recovered through each encoder

Table 11: Validation of the six TFs whose observed response enriches their own reference set, per network. Each entry gives the number of seeds in which the TF is validated ($p_{\mathrm{bonf}} < 0.05$ for an own reference set) out of the seeds available.

Network	CEBPA	CEBPE	IRF1	JUN	KLF1	SPI1
Observed response	✓	✓	✓	✓	✓	✓
Identity	3/3	0/3	3/3	0/3	3/3	3/3
PCA	3/3	0/3	3/3	2/3	3/3	3/3
scGPT	3/3	2/3	2/3	1/3	0/3	0/3
scPRINT	1/3	0/3	3/3	0/3	3/3	0/3
Stack-Large	0/3	0/3	3/3	0/3	2/3	3/3
Tahoe-x1	3/3	0/3	0/3	0/3	0/3	0/3
Shuffled labels	0/3	0/3	0/3	1/3	1/3	1/3
Random weights	0/3	0/3	0/3	0/3	0/3	0/3

Table 11 breaks down Figure 2c by TF. Identity validates CEBPA, IRF1, KLF1 and SPI1 in every seed, and PCA the same four, with JUN in two seeds. Stack-Large validates IRF1 and SPI1 in every seed and KLF1 in two; scPRINT validates IRF1 and KLF1 in every seed and CEBPA in one; scGPT validates CEBPA in every seed, CEBPE and IRF1 in two and JUN in one, and KLF1 and SPI1 in none. Tahoe-x1 validates CEBPA only. The three scFMs that validate more than one TF therefore do so on different TFs, and only IRF1 is validated by more than one of them in every seed. Of the negative controls, the network fitted to shuffled labels validates one TF per seed, a different one in each seed (JUN, KLF1, SPI1), and the network fitted through randomly initialized scGPT none. CEBPE is validated only through scGPT, and JUN only through PCA, scGPT and one seed of the shuffled control; these are the two TFs whose references pass only narrowly on the observed responses.

N LINEAGE SIGNATURE LIBRARY AND PATHWAY LIBRARY FILTERING

This appendix documents the curated hematopoietic lineage signature library used as the primary structural validation reference in Section 3 and Table 7 and the denylist used to filter disease-context and generic catch-all pathways from MSigDB Hallmark and Reactome 2022 before enrichment testing. Both are referenced from Appendix M.

N.1 LINEAGE SIGNATURE LIBRARY

The lineage signature library consists of eight curated hematopoietic gene sets, four of which are matched-lineage signatures expected to enrich for the master transcription factors evaluated in Appendix M (erythroid for KLF1, monocyte for SPI1, granulocyte for CEBPA and interferon response for IRF1) and four of which are controls. Two of the controls, the megakaryocyte and B cell signatures, retain fewer than five panel genes on Norman and are therefore dropped before testing (Appendix M.1). The sets are intentionally compact (20–47 genes each) to focus on canonical lineage markers rather than broader expression programs, since the goal is structural validation against the most strongly lineage-defining genes rather than against the full transcriptional response of each lineage.

The eight signatures and their biological role in the validation are as follows.

ERYTHROID_LINEAGE (44 genes). Primary expected hit for KLF1. Includes the human globin cluster (HBA1, HBA2, HBB, HBG1, HBG2, HBD, HBE1, HBM, HBQ1, HBZ), heme biosynthesis enzymes (ALAS2, FECH, HMBS, ALAD, UROD, UROS, CPOX, PPOX), iron metabolism and heme catabolism genes (SLC25A37, TFRC, FTH1, FTL, BLVRB, HMOX1), erythroid transcription factors (KLF1, GATA1, NFE2, TAL1, LDB1, LMO2, EPOR), red blood cell membrane and cytoskeletal proteins (GYPA, GYPB, GYPC, EPB42, EPB41, SPTA1, SPTB, ANK1, SLC4A1, RHCE, RHD), and erythroid maturation markers (AHSP, HEMGN).

MYELOID_GRANULOCYTE_LINEAGE (34 genes). Primary expected hit for CEBPA, and an alternative myeloid signature for SPI1. Includes primary granule proteases and antimicrobials (ELANE, MPO, PRTN3, CTSG, AZU1, BPI), secondary granule contents (LCN2, LTF, CAMP, MMP8, MMP9), neutrophil-enriched calcium-binding proteins (S100A8, S100A9, S100A12), defensins (DEFA1, DEFA3, DEFA4), granulocyte receptors and surface markers (CSF3R, FCGR3B, CEACAM3, CEACAM8, FPR1, FPR2, ITGAM), the NADPH oxidase complex (CYBA, CYBB, NCF1, NCF2, NCF4), granulocytic transcription factors (CEBPA, CEBPE, GFI1), and other markers (ALOX5, PGLYRP1).

MYELOID_MONOCYTE_LINEAGE (39 genes). Primary expected hit for SPI1. Includes canonical monocyte and macrophage markers (CD14, CD68, CD163, CD86, MRC1, MARCO, MERTK), Fc receptors (FCGR1A, FCGR2A, FCGR2B, FCGR3A), macrophage and dendritic cell transcription factors (SPI1, IRF8, MAFB, KLF4), lysozyme and lysosomal enzymes (LYZ, CTSS, CTSB, CTSD, LAPTM5), the CSF1 axis (CSF1R, CSF2RA), monocyte-enriched cytoskeletal proteins (AIF1, LST1, TYROBP, CORO1A, ARPC1B, ARHGDIB), complement components (CFD, C1QA, C1QB, C1QC), inflammation effectors (TLR2, TLR4, NLRP3, FCN1), and other myeloid markers (CD33, VAMP8, PYCARD).

MEGAKARYOCYTE_LINEAGE (22 genes). Discriminative control; dropped on Norman, where it retains fewer than five panel genes. Shares GATA1 and NFE2 with the erythroid lineage, so a recovered KLF1 adjacency that reflects erythroid commitment specifically (rather than the generic erythroid–megakaryocyte progenitor program) should hit ERYTHROID_LINEAGE more strongly than MEGAKARYOCYTE_LINEAGE. Includes platelet integrins and glycoproteins (ITGA2B, ITGB3, GP1BA, GP1BB, GP9, GP5, GP6), alpha granule contents (PF4, PPBP, VWF, FGA, FGB), the thrombopoietin receptor (MPL), megakaryocytic transcription factors (FLI1, RUNX1, MEIS1), P-selectin and adhesion (SELP, TLN1), shared erythroid–megakaryocyte transcription factors (GATA1, NFE2), and other markers (TUBB1, PROS1).

LYMPHOID_T_CELL_LINEAGE (33 genes). Negative control. K562 lacks T cell lineage commitment and no enrichment is expected for any of KLF1, SPI1, CEBPA or IRF1. Includes

the T cell receptor complex (CD3D, CD3E, CD3G, CD247, TRAC, TRBC1, TRBC2), coreceptors and accessory molecules (CD2, CD4, CD7, CD8A, CD8B, CD27, CD28), T cell transcription factors (TCF7, LEF1, TBX21, GATA3, FOXP3, RUNX3, BCL11B), effector molecules (GZMA, GZMB, GZMK, PRF1, IFNG), interleukin receptors (IL7R, IL2RA, IL2RB), and T-specific signaling components (ZAP70, LCK, ITK, THEMIS).

LYMPHOID_B_CELL_LINEAGE (28 genes). Negative control; dropped on Norman, where it retains fewer than five panel genes. Includes the B cell receptor complex (CD19, MS4A1, CD22, CD79A, CD79B), immunoglobulin chains (IGHM, IGHD, IGLL1, IGKC), B cell transcription factors (PAX5, EBF1, POU2AF1, TCF3, BCL6), plasma cell terminal differentiation transcription factors (PRDM1, XBP1, IRF4), surface and signaling molecules (CD72, BANK1, BLNK, MZB1, TNFRSF13B, TNFRSF13C), and other B cell markers (VPREB1, VPREB3, FCRL1, FCRL2, FCRL5).

INTERFERON_RESPONSE (47 genes). Primary expected hit for IRF1. Combines type I and type II interferon-stimulated genes into a single signature to avoid splitting power across two highly overlapping sets. Includes canonical interferon-stimulated genes (ISG15, ISG20, IFIT1, IFIT2, IFIT3, IFIT5), the OAS family (OAS1, OAS2, OAS3, OASL), MX dynamins (MX1, MX2), antiviral IFITM proteins (IFITM1, IFITM2, IFITM3), other ISGs (RSAD2, IFI6, IFI27, IFI44, IFI44L, USP18, BST2, HERC5, HERC6, EIF2AK2), the GBP family (GBP1, GBP2, GBP4, GBP5), JAK–STAT signaling components (STAT1, STAT2, IRF1, IRF7, IRF9), antigen presentation machinery (HLA-A, HLA-B, HLA-C, TAP1, TAP2, TAPBP, PSMB8, PSMB9, PSME1, PSME2), and type II effectors (CIITA, IDO1, WARS1).

CELL_CYCLE (38 genes). Negative control. No enrichment is expected for KLF1, SPI1, CEBPA or IRF1 if the recovered adjacency captures lineage-specific regulation rather than generic proliferation. Includes cyclins (CCNA1, CCNA2, CCNB1, CCNB2, CCND1, CCND2, CCND3, CCNE1, CCNE2), cyclin-dependent kinases (CDK1, CDK2, CDK4, CDK6, CDK7), CDK inhibitors (CDKN1A, CDKN1B, CDKN2A, CDKN2B), proliferation markers (MKI67, PCNA, TOP2A), replication licensing factors (MCM2–MCM7), spindle and mitotic regulators (AURKA, AURKB, PLK1, BUB1, BUB1B, MAD2L1), and E2F transcription factors with RB1 and TFDP1.

Sources used to curate the gene lists are in (Hay et al., 2018), (Novershtern et al., 2011), the MSigDB C8 cell type signatures collection, (Schoenfelder & Fraser, 2019) for erythroid biology, and (Tenen, 2003) for myeloid transcription factor circuitry. The total number of unique genes across all eight signatures is 283. The library is provided as a tab-separated file in Enrichr format (one set per line: `<set_name>\t\t<gene1>\t<gene2>\t...`) so that it is consumed directly by the same enrichment infrastructure used for MSigDB Hallmark and Reactome 2022. Of the eight signatures, the megakaryocyte and B cell signatures retain fewer than five genes of the Norman panel and are not tested, so the Bonferroni correction on Norman is over the $m = 6$ signatures that are tested (Appendix M.1). With so few tests, the correction remains mild, and the matched-lineage hits keep most of their statistical power.

N.2 PATHWAY LIBRARY FILTERING FOR HALLMARK AND REACTOME

For the MSigDB Hallmark 2020 and Reactome 2022 enrichment runs reported in Appendix M (Table 10), we apply a denylist filter before testing to exclude two categories of pathways that are statistically real annotations but biologically inappropriate for evaluating a regulatory network recovered from K562 perturbation data.

The first category is disease and pathogen response pathways. These pathways describe the cellular response to infection or pathology rather than physiological regulation, and pathways such as "Tuberculosis Infection Response" or "Influenza Infection" are statistically enrichable for any general inflammatory or interferon program but do not describe normal regulatory biology. We exclude pathways whose names contain any of the following substrings (case-insensitive): `infection`, `infectious`, `disease`, `diseases`, `tuberculosis`, `mtb`, `mycobacterium`, `hiv`, `influenza`, `sars`, `coronavirus`, `covid`, `hepatitis`, `leishmania`, `malaria`, `plasmodium`, `carcinoma`, `neoplasm`, `leukem`, `tumor`, `tumour`, `cancer`, `syndrome`, `disorder`.

The second category is generic catch-all pathways at the top of the Reactome hierarchy. These pathways are broad superset annotations such as "Generic Transcription Pathway" or "Signal Transduction" that aggregate large numbers of genes across many specific functions. They are consistently enriched in any sufficiently broad gene list and provide no biological discrimination. We exclude pathways whose names match the following exact strings: `Generic Transcription Pathway`, `Developmental Biology`, `Signal Transduction`, `Metabolism`, `Metabolism of proteins`, `Metabolism of RNA`, `Gene expression (Transcription)`, `Gene expression`, `RNA Polymerase II Transcription`, `Cellular responses to stimuli`, `Cellular responses to stress`, `Cellular responses to external stimuli`, `Hemostasis`, `Programmed Cell Death`, `Cell Cycle`, `Cell Cycle, Mitotic`, `Mitotic Cell Cycle`, `Vesicle-mediated transport`, `Membrane Trafficking`, `Post-translational protein modification`.

The exact-name match is applied after stripping the trailing Reactome stable identifier (e.g., `R-HSA-NNNNNN`) that Enrichr appends to Reactome pathway names. We do not pattern-match on the substring `Metabolism`, because specific subpathways such as "Heme Metabolism" or "Metabolism of Lipids" are legitimate and biologically informative annotations that should be tested. The exact-string filter targets only the top-level catch-all entries.

The filter is not applied to the lineage signature library (Appendix N.1), to ChEA 2022, or to ENCODE TF ChIP-Seq 2015. The lineage signatures are curated specifically for this evaluation and contain no disease or generic-catch-all entries by construction. The ChEA and ENCODE libraries name gene sets by transcription factor and cell line rather than by pathway concept, so neither the disease substring list nor the generic-catch-all exact match list captures any entries in those libraries.