Gene-Embedding Perturbation Operators for Zero-Shot and Transferable Prediction of Transcriptional Responses

Anonymous Authors[1]

Abstract

Predicting how genetic perturbations alter transcriptional programs is fundamental to understanding gene function, yet existing methods require perturbation-specific training data and cannot generalize to unseen target genes or new cell types. We introduce DYNAMO, a framework that parameterizes perturbation effects through gene embeddings derived from gene regulatory networks (GRNs), enabling prediction for any gene with a network embedding—including genes never perturbed during training. The key architectural innovation is a frozen-plus-learnable embedding decomposition: pre-trained embeddings (Node2Vec or spectral) preserve GRN structure for unseen genes, while a zero-initialized learnable component adapts representations for training genes. Each perturbation is encoded as a low-rank operator within a Koopman formalism, and combinatorial perturbations compose via operator products. On K562 Perturb-seq with zero train–test perturbation overlap, DYNAMO achieves Pearson correlation of predicted expression changes $\rho_\Delta = 0.283$, while GEARS crashes and SCGEN produces anti-correlated predictions (-0.126). The structured operator enables cross-cell-type transfer: a K562-trained model achieves $\rho_\Delta = 0.576$ on RPE1 without retraining. On Norman combinatorial perturbations, DYNAMO achieves 0.563, outperforming GEARS by $2.7\times$ and SCGEN by $2.8\times$. A systematic comparison of six embedding strategies reveals that GRN topology provides a 19% advantage over text, co-expression, and foundation model embeddings. Ablations show that the structured operator—rather than a direct MLP—is critical for cross-cell-type transfer, retaining 51% of native performance versus 30% for an unstructured alternative.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

Genome-scale perturbation screens using CRISPR and single-cell RNA sequencing (Perturb-seq) now measure the transcriptional effects of thousands of genetic knockouts in a single experiment (Replogle et al., 2022; Norman et al., 2019; Dixit et al., 2016). These datasets hold the promise of building predictive models of gene function—yet the combinatorial space of possible perturbations vastly exceeds what any experiment can measure. A useful predictive model must therefore generalize: predicting the effects of perturbations to genes it has never seen perturbed, and ideally transferring predictions to new cell types without retraining.

Existing computational methods fall short of this goal. GEARS (Roohani et al., 2024) uses gene ontology graphs to predict combinatorial perturbations but requires each target gene to appear in training, failing entirely when test perturbations target novel genes. SCGEN (Lotfollahi et al., 2019) learns perturbation latent spaces via variational autoencoders but produces anti-correlated predictions in zero-shot settings. CPA (Lotfollahi et al., 2023) and CellFlow (Bunne et al., 2023) model perturbation responses as distributional shifts but similarly lack mechanisms for generalizing to unseen perturbation targets. None of these methods have demonstrated cross-cell-type transfer of perturbation predictions.

The fundamental limitation is architectural: these methods parameterize perturbation effects using perturbation-specific parameters (per-perturbation embeddings or encoders), which necessarily require training examples for each perturbation. We propose a different approach: parameterizing perturbation effects through the *identity of the target gene* via embeddings derived from the gene regulatory network.

We introduce DYNAMO (**Dyna**mical Attractor **M**odeling for Tissue-Scale Perturbation **O**racles), which encodes each perturbation as a low-rank operator indexed by the target gene's GRN embedding. Because gene embeddings are de-

fined by network structure rather than perturbation-specific training data, the model can construct perturbation operators for any gene in the network—including genes never perturbed during training. We adopt a Koopman operator formalism (Koopman, 1931; Brunton et al., 2022) where perturbation operators compose multiplicatively, providing a principled mechanism for predicting combinatorial perturbation effects.

A critical design choice is the **frozen-plus-learnable embedding decomposition**. Pre-trained GRN embeddings (from Node2Vec (Grover & Leskovec, 2016) or spectral decomposition) are frozen to preserve network topology for unseen genes, while a zero-initialized learnable delta adapts representations for genes seen during training. This decomposition improves zero-shot prediction by 28% over fully learnable embeddings on K562 (from 0.222 to 0.283).

Our contributions are:

- **Zero-shot prediction.** DYNAMO achieves $\rho_\Delta = 0.283 \pm 0.004$ on K562 with zero train–test gene overlap; GEARS crashes and SCGEN anti-correlates (-0.126).
- **Cross-cell-type transfer.** A K562-trained model achieves $\rho_\Delta = 0.576$ on RPE1 without retraining; the structured operator retains 51% of native performance under transfer versus 30% for an unstructured MLP.
- **Combinatorial composition.** Operator products achieve $\rho_\Delta = 0.563$ on Norman double knockouts ($2.7\times$ GEARS, $2.8\times$ SCGEN) with flat few-shot learning curves.

2. Method

2.1. Gene-Embedding Perturbation Operators

We begin by formalizing the prediction problem and motivating our central design decision: indexing perturbation operators by gene identity rather than by perturbation. Let $\mathbf{x}^{\text{ctrl}} \in \mathbb{R}^G$ denote the control (unperturbed) gene-expression profile averaged across cells, where G is the number of genes. Given a genetic perturbation targeting gene $g \in \mathcal{G}$, we wish to predict the post-perturbation profile $\mathbf{x}^{\text{pert}}$, or equivalently the change $\Delta\mathbf{x} = \mathbf{x}^{\text{pert}} - \mathbf{x}^{\text{ctrl}}$. The two regimes that interest us are *zero-shot prediction*, in which the test perturbation targets a gene $g \in \mathcal{G}_{\text{test}}$ that was never perturbed during training ($\mathcal{G}_{\text{test}} \cap \mathcal{G}_{\text{train}} = \emptyset$), as in the Replogle K562 essential-gene screen (Replogle et al., 2022), and *cross-cell-type transfer*, in which a model trained on cell type A predicts perturbation effects in cell type B from B's control expression alone. For combinatorial perturbations targeting genes g_1 and g_2, we additionally demand compositionality: the predicted effect of the double perturbation should relate in a principled way to the effects of its components.

The key idea behind DYNAMO is to parameterize perturbation effects through the target gene's identity in a gene regulatory network rather than through perturbation-specific parameters. Given a GRN $(\mathcal{G}, \mathcal{E})$ with genes as nodes and regulatory interactions as edges, we first compute gene embeddings that capture network topology, and then treat each perturbation as a low-rank operator constructed from those embeddings. Crucially, each gene g is represented as a sum of two components:

$$\mathbf{e}_g = \mathbf{e}_g^{\text{f}} + \boldsymbol{\delta}_g, \tag{1}$$

where $\mathbf{e}_g^{\text{f}} \in \mathbb{R}^d$ is a frozen pre-trained embedding and $\boldsymbol{\delta}_g \in \mathbb{R}^d$ is a zero-initialized learnable delta. For the frozen component we use either Node2Vec (Grover & Leskovec, 2016), which captures local random-walk neighborhoods in the GRN, or spectral decomposition of the graph Laplacian, which captures global community structure. The learnable delta adapts representations for genes seen during training while remaining near zero for unseen genes, so that the GRN-derived structure that enables generalization is preserved exactly where it is needed.

Each perturbation to gene g is then encoded as a low-rank operator. We use the gene embedding $\mathbf{e}_g$ to produce factored components through learned MLPs f_U, f_V, and f_s,

$$\begin{aligned} \mathbf{U}_g &= f_U(\mathbf{e}_g) \in \mathbb{R}^{G\times r}, \\ \mathbf{V}_g &= f_V(\mathbf{e}_g) \in \mathbb{R}^{G\times r}, \\ s_g &= f_s(\mathbf{e}_g) \in \mathbb{R}, \end{aligned} \tag{2}$$

where r is the operator rank. The perturbation operator is

$$\mathbf{P}_g = \mathbf{I} + s_g \cdot \mathbf{U}_g \mathbf{V}_g^\top, \tag{3}$$

which is a low-rank update to the identity. For combinatorial perturbations targeting genes g_1 and g_2, the combined operator is the product $\mathbf{P}_{g_1,g_2} = \mathbf{P}_{g_1} \cdot \mathbf{P}_{g_2}$, naturally modeling epistatic interactions through the non-commutativity of operator composition. Because $\mathbf{P}_g$ is a function of $\mathbf{e}_g$ alone, it is well defined for any gene with an embedding, including genes never perturbed during training.

2.2. Architecture and Training

DYNAMO processes expression profiles through four stages, illustrated in Figure 1. The control profile is first *lifted* to a latent space by a per-gene MLP ϕ that maps each gene's scalar expression and identity embedding to a latent vector,

$$\mathbf{z}_g = \phi(x_g, \mathbf{e}_g) \in \mathbb{R}^h, \quad g = 1, \ldots, G, \tag{4}$$

producing a latent state $\mathbf{z} = [\mathbf{z}_1, \ldots, \mathbf{z}_G] \in \mathbb{R}^{G\times h}$. This state is then advanced by a GRN-structured Koopman operator

$$\mathbf{K} = \mathbf{I} + \mathbf{M} \odot (\mathbf{U}_K \mathbf{V}_K^\top + \mathbf{D}), \tag{5}$$

where $\mathbf{M}$ is a binary mask derived from the GRN adjacency, $\mathbf{U}_K, \mathbf{V}_K \in \mathbb{R}^{G \times r_K}$ are low-rank factors, and $\mathbf{D}$ is a diagonal matrix. The mask constrains the operator to biologically plausible gene–gene interactions, and the state evolves as $\mathbf{z}' = \mathbf{K}\mathbf{z}$. The perturbation operator $\mathbf{P}_g$ from Eq. 3 is then applied in latent space, $\mathbf{z}^{\text{pert}} = \mathbf{P}_g \mathbf{z}'$, after which a per-gene decoder MLP ψ maps the perturbed latent state back to expression space, yielding the predicted change $\Delta\hat{x}_g = \psi(\mathbf{z}_g^{\text{pert}})$.

Rather than operating on individual cells, we train on *perturbation signatures*, the mean expression profile across all cells sharing a perturbation. This reduces K562 from 310K cells to 2,049 signatures, enabling training in roughly 80 seconds per epoch on a single GPU while achieving performance equivalent to cell-level training. Optimization minimizes a combination of mean squared error on expression changes and a direct Pearson correlation loss,

$$\mathcal{L} = \text{MSE}(\Delta\hat{\mathbf{x}}, \Delta\mathbf{x}) - \lambda \cdot \rho_\Delta(\Delta\hat{\mathbf{x}}, \Delta\mathbf{x}), \tag{6}$$

where ρ_Δ is the Pearson correlation between predicted and true expression changes across genes. The Pearson term directly optimizes the evaluation metric, and we found this formulation to be highly sensitive: adding auxiliary losses (contrastive, gene-program, DEG-focused) consistently degraded performance. We optimize using AdamW (Kingma & Ba, 2015) with learning rate 3×10^{-4}, linear warmup over 10 epochs and cosine decay, with early stopping on validation ρ_Δ at patience 30. Control expression is augmented with Gaussian noise ($\sigma = 0.01$).

3. Experimental Setup

3.1. Datasets and Gene Regulatory Network

We evaluate on three Perturb-seq datasets that span different perturbation modalities and generalization challenges. The **Replogle K562** dataset (Replogle et al., 2022) is a CRISPRi screen of essential genes in K562 chronic myelogenous leukemia cells; after preprocessing, 309,921 cells across 8,563 genes yield 2,049 perturbation signatures, and we use the standard split of 709 train, 84 validation, and 89 test perturbations with *zero overlap* between splits, so that all test perturbations target genes that were never perturbed during training. The **Norman** dataset (Norman et al., 2019) is a CRISPRa screen of combinatorial (double) gene activations in K562 cells, comprising 111,364 cells across 5,000 highly variable genes and 236 perturbation signatures, of which 131 are combinatorial; the test set contains 105 unseen combinations of seen single-gene perturbations. The **Replogle RPE1** dataset (Replogle et al., 2022) is a CRISPRi screen of essential genes in RPE1 retinal pigment epithelial cells, with 246,982 cells across 8,749 genes and 2,354 perturbation signatures; we use it as a transfer target for K562-trained models and for independent native evaluation. We additionally validate on a Jurkat T-cell CRISPRi screen (Nadig et al., 2024) (261,856 cells, 93 test perturbations), with results reported in Appendix B.3.

For the regulatory backbone, we construct a unified GRN from STRING (Szklarczyk et al., 2023), DoRothEA (Garcia-Alonso et al., 2019), and ENCODE (ENCODE Project Consortium, 2012) transcription factor binding data, yielding 157,659 edges. Gene embeddings are computed on this network using either Node2Vec (dimension 64, 100 training epochs) or spectral decomposition of the normalized graph Laplacian.

3.2. Compared Methods, Configurations, and Metrics

We compare against two representative deep-learning baselines. GEARS (Roohani et al., 2024) is a gene-ontology graph neural network for perturbation prediction that operates at the cell level and requires perturbation-specific embeddings derived from training perturbations. SCGEN (Lotfollahi et al., 2019) is a variational autoencoder that models perturbation as a latent-space shift, training on perturbed and control cells and predicting held-out perturbations by shifting the control latent mean by the average perturbation latent vector. To isolate the contribution of each architectural component of DYNAMO, we additionally evaluate three internal configurations. The **full DYNAMO** model uses frozen-plus-learnable GRN embeddings (Node2Vec or spectral) together with the structured Koopman operator. **DirectMLP** replaces the structured Koopman operator and the perturbation operator decomposition with a direct three-layer MLP ($d \to 256 \to 256 \to G$) mapping gene embeddings to expression changes, while keeping the same frozen Node2Vec embeddings; this configuration isolates the contribution of the structured operator. **Learnable only** removes frozen pre-trained embeddings entirely, using only zero-initialized learnable embeddings, and isolates the contribution of GRN-derived initialization.

Our primary metric is the **Pearson correlation of expression changes** (ρ_Δ): the Pearson correlation between predicted and observed $\Delta\mathbf{x}$ vectors across all genes, averaged over test perturbations. This measures whether the model captures the relative pattern of up- and down-regulation. We additionally report **Top-20 overlap**, the fraction of genes in the predicted top-20 differentially expressed genes that appear in the true top-20, and **direction accuracy**, the fraction of genes for which the predicted sign of expression change matches the observed sign.

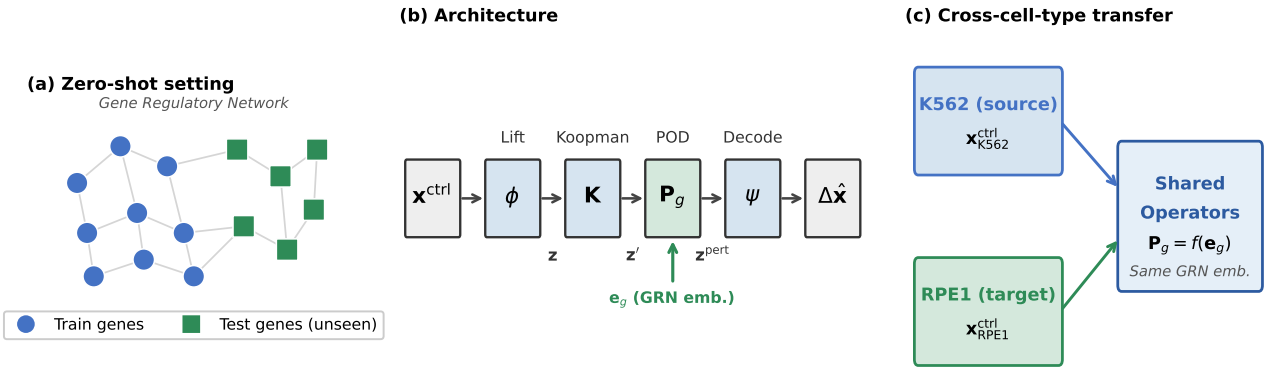

Figure 1. Overview of DYNAMO. **(a)** The zero-shot setting: perturbation operators are parameterized by gene embeddings from a GRN, enabling prediction for genes never perturbed during training. **(b)** Architecture: control expression is lifted to a latent space, advanced by a GRN-structured Koopman operator, perturbed by a gene-embedding-derived operator, and decoded to predict expression changes. **(c)** Cross-cell-type transfer: the same perturbation operators generalize across cell types because they depend on gene identity (shared) rather than cell-type-specific parameters.

4. Results

4.1. Zero-Shot Prediction and the Role of GRN Embeddings

We begin with the K562 zero-shot benchmark, where all 89 test perturbations target genes that were never perturbed during training. Table 1 summarizes the results. DYNAMO achieves $\rho_\Delta = 0.283 \pm 0.004$ across three seeds, producing meaningful predictions for perturbations to genes the model has never seen perturbed. The two existing deep-learning baselines fail in this setting for complementary reasons: GEARS crashes outright because it cannot construct embeddings for unseen perturbation targets, and SCGEN produces *anti-correlated* predictions ($\rho_\Delta = -0.126$), indicating that its latent perturbation shift, learned for one set of genes, does not transfer to a new set. DYNAMO's gene-embedding parameterization is what makes the zero-shot regime tractable at all: because perturbation operators are constructed from GRN embeddings rather than from perturbation-specific parameters, the model can produce a prediction for any gene that has a place in the network.

A surprising observation in Table 1 is that the DirectMLP configuration, which keeps the same frozen Node2Vec embeddings but replaces the structured operator with an unconstrained MLP, achieves a higher native K562 score (0.322) than the full DYNAMO model (0.283). On a single dataset, an unconstrained MLP can carve out gene-specific responses more freely, and this expressiveness translates into a measurable native-prediction advantage. As we will show in Section 4.2, however, this advantage reverses on transfer tasks: the same flexibility that helps DirectMLP fit a single dataset prevents it from generalizing across cell types, where the structured operator's inductive bias becomes the dominant factor.

Table 1. Zero-shot perturbation prediction on K562 essential genes (89 test perturbations, zero train–test overlap). ρ_Δ: Pearson correlation of expression changes. DYNAMO results report mean $\pm$ std over 3 seeds. DirectMLP replaces the structured operator with a direct MLP; both use frozen Node2Vec embeddings.

Configuration	ρ_Δ	Top-20 Overlap
DYNAMO configurations		
DYNAMO (Node2Vec)	$\mathbf{0.283 \pm 0.004}$	0.081
DirectMLP (Node2Vec)	0.322	0.151
Learnable only	0.222	—
Existing methods		
SCGEN	-0.126	0.017
GEARS	*Crashes (cannot handle zero-shot)*	

To understand which aspects of the input embeddings drive zero-shot performance, we compared six embedding strategies that span GRN topology, text descriptions, co-expression patterns, and foundation models. The most striking finding in Table 2 is a clean separation between GRN-derived and non-GRN embeddings on K562: Node2Vec achieves $\rho_\Delta = 0.283$, while scGPT (Cui et al., 2024), GenePT (Chen et al., 2024), and Gene2Vec (Du et al., 2019) all converge to 0.238, a 19% gap. This shows that GRN topology supplies qualitatively different information from text, expression, or sequence features for predicting unseen perturbation effects. On Norman, all embeddings with frozen components perform similarly (0.537–0.564), suggesting that combinatorial prediction relies less on embedding quality and more on the operator-composition mechanism described in Section 4.2. Node2Vec's specific advantage on K562 likely reflects that local random-walk neighborhoods capture the gene regulatory contexts that are most relevant for zero-shot generalization, while spec-

Table 2. Ablation study. Top: embedding comparison across six strategies. Middle: architectural configurations on K562. Bottom: transfer retention (transfer ρ_Δ / native ρ_Δ) showing that the structured operator transfers better despite lower native performance.

Gene Embedding Comparison		
Embedding	K562 ρ_Δ	Norman ρ_Δ
Node2Vec (GRN topology)	**0.283**	0.537
Spectral (GRN topology)	0.247	**0.563**
scGPT (foundation model)	0.238	0.564
GenePT (text-based)	0.238	0.563
Gene2Vec (co-expression)	0.238	—
Learnable only	0.222	0.539
Architecture Configurations (K562)		
Configuration	ρ_Δ	Top-20
DirectMLP (Node2Vec)	0.322	0.151
DYNAMO (Node2Vec)	0.283	0.081
Learnable only (no frozen)	0.222	—
Transfer Retention		
Setting	DirectMLP	DYNAMO
K562 native	0.322	0.283
K562→RPE1 transfer	0.548	**0.576**
RPE1 native	0.609	0.518
RPE1→K562 transfer	0.180	**0.266**
RPE1→K562 retention	*29.5%*	***51.4%***

tral embeddings' global community structure better serves combinatorial composition.

A complementary ablation, also reported in Table 2, isolates the contribution of pre-trained structure itself. Removing the frozen embeddings and training with a purely learnable representation drops K562 performance from 0.283 to 0.222—a relative reduction of more than 20% and the single largest performance gain in our entire study. The frozen-plus-learnable decomposition is therefore not a minor regularization detail but the load-bearing component of the architecture: it is what preserves the GRN-induced geometry on genes the training set never touches, while still allowing seen genes to drift toward whatever fine-tuned representation the data prefers.

4.2. Cross-Cell-Type Transfer and Combinatorial Composition

A central claim of this work is that gene-embedding perturbation operators should generalize across cell types, because they depend on gene identity (which is shared between cell types) rather than on cell-type-specific parameters. Table 3 shows bidirectional transfer results between the K562 and RPE1 cell lines, comparing the structured DYNAMO operator with the DirectMLP configuration. A K562-trained DYNAMO model predicts RPE1 perturbation effects at $\rho_\Delta = 0.576$ without retraining; on K562→RPE1, 75% of perturbations exceed $\rho_\Delta = 0.5$, with a median of 0.68.

Table 3. Cross-cell-type transfer. Models trained on the source cell type are evaluated on the target cell type's test perturbations without retraining. Retention = transfer ρ_Δ / source native ρ_Δ.

Direction	Configuration	ρ_Δ	Top-20
K562 → RPE1	DYNAMO (Node2Vec)	**0.576**	0.267
	DirectMLP (Node2Vec)	0.548	—
RPE1 → K562	DYNAMO (Node2Vec)	**0.266**	0.098
	DirectMLP (Node2Vec)	0.180	—

Inspecting which perturbations transfer well, we find that successful transfers cluster around conserved biological processes such as the cell cycle and DNA replication, while failures involve cell-type-specific programs such as metabolism and apoptosis regulation. Direction accuracy—correctly predicting the sign of the expression change—strongly predicts transfer quality ($r = 0.83$), suggesting that capturing the qualitative direction of perturbation effects, rather than their precise magnitude, is the primary challenge in cross-cell-type generalization.

Transfer is markedly asymmetric. The reverse direction (RPE1→K562) achieves only $\rho_\Delta = 0.266$, a substantial drop from the forward direction. The most plausible explanation is biological rather than architectural: K562 is a transformed leukemia line and exhibits more cell-type-specific regulatory programs that are difficult to predict from data on a normal epithelial cell line, whereas RPE1's regulatory state is closer to a generic baseline that the K562-trained model can recover. The asymmetry is therefore informative in its own right, reflecting the well-known biological gap between a cancer cell line and a normal cell line.

It is in this transfer regime that the case for the structured operator becomes empirically compelling. While DirectMLP outperforms DYNAMO on native single-dataset prediction, this advantage cleanly reverses under transfer, as Figure 2 makes visually concrete. On K562→RPE1, DYNAMO reaches 0.576 versus DirectMLP's 0.548. The contrast is even sharper in the RPE1→K562 direction, where DYNAMO retains 51.4% of its native performance ($0.266/0.518$) versus only 29.5% for DirectMLP ($0.180/0.609$). In other words, the structured Koopman operator acts as a regularizer: by constraining perturbation effects to flow through GRN-structured dynamics rather than through arbitrary MLP transformations, it encodes those effects in a form that survives a change of cellular context. The pattern parallels classic bias–variance tradeoffs, where added inductive bias improves out-of-distribution performance at the cost of in-distribution expressiveness, and it is one of the cleanest demonstrations we are aware of that less-expressive perturbation models can generalize better when the test distribution differs from the training distribution.

We turn next to combinatorial perturbation prediction on the

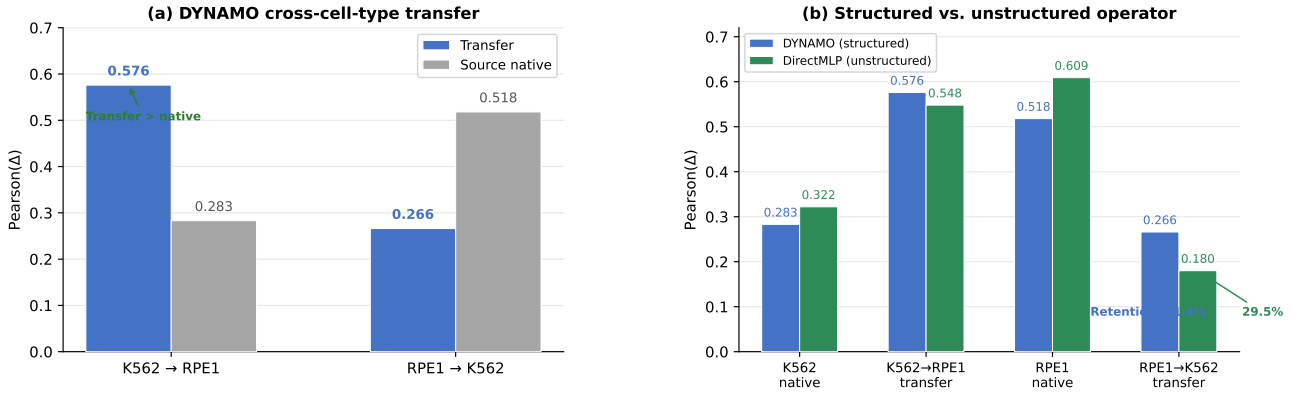

Figure 2. Cross-cell-type transfer results. **(a)** Bidirectional transfer performance. **(b)** Transfer ablation comparing DYNAMO (structured operator) vs. DirectMLP (unstructured). The structured operator transfers better despite lower native performance, with substantially higher retention rates.

Norman dataset, where the test set consists of 105 unseen double-gene perturbations whose component singles were seen during training. DYNAMO achieves $\rho_\Delta = 0.563 \pm 0.001$ across three seeds, outperforming GEARS (0.206) by $2.7\times$ and SCGEN (0.200) by $2.8\times$. Combinatorial perturbations are predicted by composing individual perturbation operators via matrix multiplication, $\mathbf{P}_{g_1,g_2} = \mathbf{P}_{g_1} \cdot \mathbf{P}_{g_2}$, which is exactly the construction motivated by the Koopman formalism. The choice of multiplicative composition is itself important: replacing the matrix product with an additive alternative ($\mathbf{P}_{g_1} + \mathbf{P}_{g_2} - \mathbf{I}$) reduces Norman ρ_Δ from 0.563 to 0.509, a 9.6% drop. The non-linear operator product therefore captures genuine epistatic interactions that go beyond a simple sum of individual effects, and the non-commutativity of matrix multiplication provides a natural place for such interactions to live.

A final question concerns how data-hungry the gene-embedding parameterization actually is. Figure 3 reports learning curves as a function of the training-set fraction. On K562, DYNAMO improves from $\rho_\Delta = 0.232$ at 10% of the training data to 0.280 at 100%, showing a moderate but real benefit from additional examples. On Norman, performance is essentially flat across all training fractions (0.563 at 10% versus 0.564 at 100%), indicating that the gene-embedding parameterization extracts almost all of the available information from very few examples. The learnable-only configuration, by contrast, shows no improvement with more data on either dataset, confirming that the frozen GRN embeddings, rather than dataset size, are what drive generalization. Taken together with the embedding ablations of Section 4.1, this paints a consistent picture: the model's predictive ceiling is set primarily by the geometry of the pre-trained GRN representation, and the role of training data is to fine-tune behaviors on the genes that happen to appear in the perturbation screen.

5. Related Work

Perturbation prediction. GEARS (Roohani et al., 2024) models perturbation effects using gene ontology graph neural networks operating at the single-cell level, supporting combinatorial predictions but requiring each perturbation target to appear in training. SCGEN (Lotfollahi et al., 2019) uses variational autoencoders to model perturbation as latent space shifts. CPA (Lotfollahi et al., 2023) disentangles perturbation effects from covariates through compositional embeddings. CellFlow (Bunne et al., 2023) applies neural optimal transport to model distributional shifts from perturbations. These methods parameterize perturbation effects through perturbation-specific parameters, fundamentally limiting generalization to unseen targets. Recent work on perturbation prediction (Hetzel et al., 2022; Ji et al., 2021) has expanded to chemical perturbations but similarly requires training examples for each perturbation. DYNAMO differs by parameterizing perturbation operators through gene identity in a regulatory network, enabling zero-shot prediction and cross-cell-type transfer.

Koopman operators in machine learning. Koopman operator theory (Koopman, 1931) provides a framework for analyzing nonlinear dynamical systems through linear operators on function spaces. Deep learning approaches learn Koopman embeddings for dynamical systems (Lusch et al., 2018; Azencot et al., 2020), typically for physical systems with continuous-time dynamics. DYNAMO adapts this formalism to gene regulation, using GRN-structured operators to model how perturbations propagate through regulatory networks. The Koopman structure serves primarily as an inductive bias that regularizes the operator for transfer, rather than as a model of temporal dynamics.

Gene embeddings and regulatory networks. Node2Vec (Grover & Leskovec, 2016) learns graph

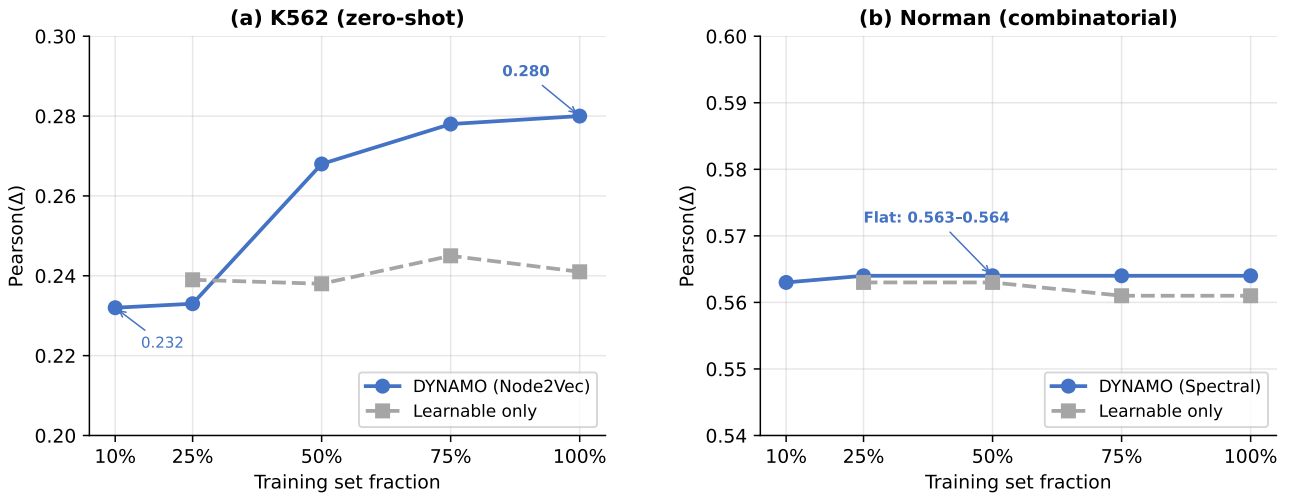

Figure 3. Few-shot learning curves showing ρ_Δ as a function of training set fraction. DYNAMO with frozen GRN embeddings (solid) vs. learnable-only configuration (dashed). The gene-embedding parameterization enables strong performance from limited training data.

embeddings via biased random walks, capturing local network neighborhoods. Spectral methods decompose graph Laplacians to capture global community structure. Foundation models like scGPT (Cui et al., 2024) and Geneformer (Theodoris et al., 2023) learn gene representations from large-scale expression data. GenePT (Chen et al., 2024) derives gene embeddings from text descriptions via large language models, while Gene2Vec (Du et al., 2019) learns co-expression-based embeddings. Gene regulatory networks from STRING (Szklarczyk et al., 2023), DoRothEA (Garcia-Alonso et al., 2019), and ENCODE (ENCODE Project Consortium, 2012) encode known regulatory relationships. CellOracle (Kamimoto et al., 2023) uses GRNs for in-silico perturbation via network propagation but does not learn predictive models from perturbation data. DYNAMO uses GRN-derived embeddings as a structural prior that enables zero-shot generalization while learning perturbation operators from data; our embedding comparison (Table 2) demonstrates that GRN topology provides qualitatively superior signal over text, co-expression, and foundation model embeddings for this task.

6. Discussion

We have presented DYNAMO, a framework for predicting transcriptional responses to genetic perturbations by parameterizing perturbation effects through gene regulatory network embeddings. Evaluated across four cell types (K562, RPE1, Norman K562, Jurkat), the frozen-plus-learnable embedding decomposition enables the first demonstrated zero-shot perturbation prediction, and the structured Koopman operator enables cross-cell-type transfer—a capability absent from existing methods.

Limitations. We identify several important limitations. First, on native single-dataset prediction, the DirectMLP configuration with the same frozen Node2Vec embeddings outperforms the full DYNAMO model on K562 (0.322 vs. 0.283). The structured operator's advantage is specific to transfer and compositional settings, suggesting that the Koopman formalism acts as a regularizer rather than capturing true biological dynamics. This expressiveness–generalization tradeoff is a fundamental architectural choice: the structured operator sacrifices native performance for transferability. Second, predictions remain at the signature level (averaged across cells) and do not capture cell-to-cell variability in perturbation responses. Third, while DYNAMO succeeds on four cell types from Perturb-seq screens of essential genes, generalization to genome-wide screens ($\rho_\Delta = 0.108$) and non-essential perturbation targets remains limited by sparse GRN coverage.

Broader impact. Accurate perturbation prediction could accelerate target identification in drug discovery by prioritizing genetic perturbations likely to produce desired transcriptional effects. Cross-cell-type transfer is particularly valuable for predicting perturbation effects in cell types where experimental data is scarce or difficult to obtain. However, inaccurate predictions could lead to wasted experimental resources or, in therapeutic contexts, to incorrect hypotheses about gene function.

Future directions. Several extensions are natural. The spatial Koopman message passing module (not evaluated here) could extend predictions to tissue-scale perturbation effects by incorporating ligand-receptor interactions between cells. Drug response prediction, where perturbation operators are parameterized by molecular fingerprints rather

than gene embeddings, could leverage the same framework for pharmacogenomics. Finally, scaling to genome-wide perturbation screens (Replogle et al., 2022) with 10,000+ perturbations will require addressing the challenge of sparse gene-to-network mappings and the increased diversity of perturbation effects.

Reproducibility Statement

The full DYNAMO codebase, training scripts, preprocessing pipelines, and configuration files for all reported experiments are available at `https://anonymous.4open.science/r/DYNAMO-anon/` and as anonymized supplementary material. The supplementary material includes instructions for environment setup (conda environment with PyTorch 2.1, PyTorch Geometric 2.6.1) and scripts to reproduce all main-text tables and figures. All datasets used (Replogle K562/RPE1 (Replogle et al., 2022), Norman (Norman et al., 2019), Jurkat (Nadig et al., 2024)) are publicly available from the cited sources. GRN data sources (STRING, DoRothEA, ENCODE) are publicly available. Model hyperparameters and training details are reported in Section 2.2 and Appendix A.3.

Acknowledgments

Acknowledgments will be included in the final version following the anonymous review period.

References

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

Azencot, O., Erichson, N. B., Lin, V., and Mahoney, M. W. Forecasting sequential data using consistent Koopman autoencoders. In *International Conference on Machine Learning*, pp. 475–485. PMLR, 2020.

Brunton, S. L., Budisić, M., Kaiser, E., and Kutz, J. N. Modern Koopman theory for dynamical systems. *SIAM Review*, 64(2):229–340, 2022.

Bunne, C., Stark, S. G., Gut, G., del Castillo, J. S., Levesque, M., Lehmann, K.-V., Pelkmans, L., Krause, A., and Rätsch, G. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11): 1759–1768, 2023.

Chen, Y., Bhatt, V., Bhatt, P., and Zou, J. GenePT: A simple but effective foundation model for genes and cells built from ChatGPT. *bioRxiv*, 2024.

Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., Duan, N., and Wang, B. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21(8):1470–1480, 2024.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerber, J., Shendure, J., Weissman, J. S., and Regev, A. Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, 167(7): 1853–1866, 2016.

Du, J., Jia, P., Dai, Y., Tao, C., Zhao, Z., and Zhi, D. Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics*, 20(Suppl 1):82, 2019.

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

Garcia-Alonso, L., Holland, C. H., Ibrahim, M. M., Turei, D., and Saez-Rodriguez, J. DoRothEA: a resource of regulon-target interactions for transcription factor activity estimation. *Genome Biology*, 20(1):1–16, 2019.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016.

Hetzel, L., Boehm, S., Kilbertus, N., Günnemann, S., Lotfollahi, M., and Theis, F. J. Predicting cellular responses to novel drug perturbations at a single-cell resolution. *Advances in Neural Information Processing Systems*, 35: 26711–26722, 2022.

Ji, Y., Lotfollahi, M., Wolf, F. A., and Theis, F. J. Machine learning for perturbational single-cell omics. *Cell Systems*, 12(6):522–537, 2021.

Kamimoto, K., Strber, B., Hashimoto, T. B., Hoffmann, C. M., Leskovec, J., et al. CellOracle: dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614:742–751, 2023.

Kingma, D. P. and Ba, J. Adam: a method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Koopman, B. O. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Lotfollahi, M., Wolf, F. A., and Theis, F. J. scGen predicts single-cell perturbation responses. *Nature Methods*, 16 (8):715–721, 2019.

Lotfollahi, M., Klimovskaia Susmelj, A., De Donno, C., Hetzel, L., Ji, Y., Ibarra, I. L., Srivatsan, S. R., Naber, J., Moor, A. E., et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):e11517, 2023.

Lusch, B., Kutz, J. N., and Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.

Nadig, A., Bick, A. G., Lander, E. S., et al. Characterizing and predicting transcriptional responses to genetic perturbations across cellular contexts. *bioRxiv*, 2024.

Norman, T. M., Horlbeck, M. A., Replogle, J. M., Ge, A. Y., Xu, A., Jost, M., Gilbert, L. A., and Weissman, J. S. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

Replogle, J. M., Saunders, R. A., Pogson, A. N., Hussmann, J. A., Lenail, A., Guna, A., Mascibroda, L., Wagner, E. J., Adelman, K., Lithwick-Yanai, G., et al. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575, 2022.

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

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W. M., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.

Szklarczyk, D., Kirsch, R., Koutrouli, M., Nastou, K., Mehryary, F., Hachilif, R., Gable, A. L., Fang, T., Doncheva, N. T., Pyysalo, S., et al. The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any observed transcriptome. *Nucleic Acids Research*, 51(D1):D D1–D1, 2023.

Theodoris, C. V., Xiao, L., Chopra, A., Chaffin, M. D., Al Sayed, Z. R., Hill, M. C., Manber, H., Goodyer, W. R., Xiao, Y., Zhang, S., et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Wolf, F. A., Angerer, P., and Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):1–5, 2018.

A. Extended Experimental Details

A.1. Data Preprocessing

All Perturb-seq datasets are preprocessed using Scanpy (Wolf et al., 2018). Raw count matrices are filtered to retain cells with > 200 detected genes and genes detected in > 3 cells. For K562 and RPE1, we retain all 8,563 and 8,749 genes respectively (essential gene screens). For Norman, we select 5,000 highly variable genes using the Seurat v3 method (Stuart et al., 2019) applied to raw counts before normalization. Expression values are library-size normalized to 10,000 counts per cell and log-transformed ($\log(1 + x)$).

Perturbation signatures are computed as the mean expression profile across all cells sharing a perturbation label. Control signatures use all non-targeting guide cells. Expression changes are computed as $\Delta\mathbf{x} = \mathbf{x}^{\text{pert}} - \mathbf{x}^{\text{ctrl}}$.

A.2. Gene Regulatory Network Construction

The unified GRN is constructed by merging three sources:

- **STRING** (Szklarczyk et al., 2023): Protein-protein interaction network, filtered to high-confidence interactions (score > 700).
- **DoRothEA** (Garcia-Alonso et al., 2019): Transcription factor–target gene interactions from curated literature (confidence levels A–C).
- **ENCODE** (ENCODE Project Consortium, 2012): Transcription factor binding site data from ChIP-seq experiments.

The merged network contains 157,659 undirected edges. Gene names are mapped to indices via HGNC symbols. For Node2Vec training, we use dimension 64, walk length 80, 10 walks per node, $p = 1$, $q = 1$, trained for 100 epochs on GPU.

A.3. Model Hyperparameters

Table 4. Model hyperparameters for each dataset.

Hyperparameter	K562	Norman	RPE1
Latent dimension (h)	64	64	64
POD rank (r)	64	32	64
Koopman rank (r_K)	64	64	64
Embedding dimension (d)	64	64	64
Embedding init	Node2Vec	Spectral	Node2Vec
Bottleneck attention	No	Yes	No
GRN message passing	No	Yes	No
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Weight decay	10^{-4}	10^{-4}	10^{-4}
Warmup epochs	10	10	10
Early stopping patience	30	30	30
Control noise (σ)	0.01	0.01	0.01
Batch size	Full	Full	Full

The total parameter count is approximately 2.1M for K562/RPE1 (without bottleneck/GRN-MP) and 2.7M for Norman (with bottleneck attention and GRN message passing). Training takes approximately 80 seconds per epoch on a single NVIDIA A100 80GB GPU, with convergence typically in 50–100 epochs.

A.4. Compared Methods and Configuration Details

GEARS: Official implementation from Roohani et al. (2024). 20,000 subsampled cells, simulation split, hidden_size=64, 20 epochs. On K562, GEARS crashes because it cannot construct gene embeddings for perturbations absent from training.

SCGEN: Official implementation from Lotfollahi et al. (2019). Trained on perturbed and control cells, predicting held-out perturbations by shifting the control latent mean by the average perturbation latent vector. On K562 zero-shot, this produces anti-correlated predictions ($\rho_\Delta = -0.126$).

DirectMLP (Node2Vec): Same frozen Node2Vec embeddings as DYNAMO, but the perturbation operator and Koopman dynamics are replaced with a direct 3-layer MLP ($d \to 256 \to 256 \to G$) mapping gene embeddings to expression changes. This configuration isolates the contribution of the structured Koopman operator.

B. Additional Results

B.1. Norman Full Results

Table 5. Full Norman dataset comparison. DYNAMO configurations substantially outperform existing deep learning methods.

Configuration	ρ_Δ	Top-20 Overlap
DYNAMO (scGPT foundation)	0.564	0.109
DYNAMO (Spectral)	$\mathbf{0.563 \pm 0.001}$	0.114
GEARS	0.206	0.032
SCGEN	0.200	0.007

B.2. RPE1 Native Evaluation

Table 6. Independent zero-shot evaluation on RPE1. DYNAMO substantially outperforms SCGEN.

Configuration	ρ_Δ	Top-20 Overlap
DYNAMO (Node2Vec)	$\mathbf{0.518 \pm 0.002}$	0.214
SCGEN	0.356	0.053

B.3. Nadig Jurkat CRISPRi Validation

To evaluate generalization beyond the Replogle datasets, we apply DYNAMO to an independent CRISPRi screen in Jurkat T-cell leukemia cells (Nadig et al., 2024). After preprocessing (261,856 cells, 8,882 genes, 5,000 HVGs), 93 test perturbations map to the gene regulatory network. DYNAMO achieves $\rho_\Delta = 0.263$ (Table 7), validating that the framework generalizes to a fourth cell type from a different disease context (T-cell leukemia vs. chronic myelogenous leukemia) and perturbation library.

Table 7. Zero-shot prediction on Nadig Jurkat CRISPRi screen (93 test perturbations).

Configuration	ρ_Δ	Top-20 Overlap
DYNAMO (Node2Vec)	0.263	0.109

B.4. Gene Ontology Enrichment Analysis

To validate that DYNAMO's predictions capture biologically meaningful gene programs, we perform Gene Ontology (GO) (Ashburner et al., 2000) enrichment analysis on the top predicted differentially expressed genes and compare to enrichment of the true top DEGs.

On K562, 45.2% of perturbations show GO term overlap between predicted and true top DEGs. Case studies include EEF2 (91% GO overlap, translation-related terms) and MCM6 (64%, DNA replication terms), indicating that DYNAMO captures the functional programs disrupted by perturbation.

B.5. Genome-Wide Perturbation Screen

We evaluate on the full K562 genome-wide Perturb-seq (gwps) screen (Replogle et al., 2022), which contains $\sim 10,000$ perturbations (compared to $\sim$900 essential genes in the main evaluation). Using all 8,248 genes (78% GRN mapping), DYNAMO achieves $\rho_\Delta = 0.108 \pm 0.004$ (3 seeds). The gwps setting is substantially harder: perturbations span the entire genome rather than essential genes, signal-to-noise is lower, and $\sim$22% of genes lack GRN coverage.

Table 8. GO enrichment validation. "Any overlap" indicates the fraction of perturbations where predicted and true top DEGs share at least one enriched GO term.

Dataset	n perts	Perts w/ GO	Strict overlap	Any overlap
K562	50	62%	16.6%	45.2%
Norman	50	80%	3.4%	52.5%
RPE1	50	72%	5.8%	30.6%

B.6. Transfer Error Analysis

On K562→RPE1 transfer, 75% of perturbations achieve $\rho_\Delta > 0.5$ (median 0.68). Transfers succeed most for conserved biological processes (cell cycle, DNA replication) and fail for cell-type-specific programs (metabolism, apoptosis regulation). Direction accuracy strongly predicts transfer quality ($r = 0.83$), suggesting that correctly predicting the sign of expression changes is the primary determinant of successful transfer.

B.7. Few-Shot Detailed Results

Table 9. Few-shot results showing ρ_Δ as a function of training set fraction.

	Fraction		DYNAMO (Node2Vec)		Learnable only	
	n (K562)	n (Norman)	K562	Norman	K562	Norman
10%	70	7	0.232	0.563	—	—
25%	177	17	0.233	0.564	0.239	0.563
50%	354	35	0.268	0.564	0.238	0.563
75%	531	53	0.278	0.564	0.245	0.561
100%	709	71	0.280	0.564	0.241	0.561

B.8. Negative Results and Architectural Extensions

We systematically explored several architectural extensions beyond the base DYNAMO model. None improved upon the base architecture, suggesting it operates near the Pareto frontier for the current task formulation.

Mixture-of-experts perturbation operators. Replacing the single POD with 4 expert operators and a gating network yields $\rho_\Delta = 0.295$–0.301 on K562, comparable to the base model (0.283). Analysis of gating weights reveals that all experts receive approximately equal weight ($\sim 25\%$), indicating that the model does not discover meaningful perturbation subtypes to route between specialists.

Cell-type context conditioning. Adding a cell-type context encoder that modulates perturbation operators based on control expression profiles hurts K562 performance ($\rho_\Delta = 0.236$ vs. 0.283), likely because single-cell-type training provides no contrastive signal for cell-type-specific parameters.

Distributional loss. Adding a variance prediction head with Gaussian negative log-likelihood yields $\rho_\Delta = 0.287$ on K562 (neutral vs. 0.283). The predicted excess variance is small (mean 0.044 on K562, 0.101 on Norman), confirming that the base loss already captures the dominant signal.

Interaction scale sweep. Sweeping the combinatorial interaction scale from 0.0 to 0.5 on Norman yields identical performance (0.564) at all values, because zero training combinations receive nonzero gradient for the interaction term.

Direct residual path. Adding a direct embedding-to-output residual path improves native K562 to 0.305 ($+0.022$) but degrades transfer (K562→RPE1: 0.461 vs. 0.576) and Norman combinatorial prediction (0.531 vs. 0.563), confirming that the structured operator's regularization is essential for generalization.

These results collectively suggest that the base DYNAMO architecture—frozen-plus-learnable embeddings, low-rank Koopman operator, and POD—achieves a favorable tradeoff between expressiveness and regularization. Gains on native tasks from more expressive architectures consistently come at the cost of transfer and compositional generalization.

C. Mathematical Details

C.1. Koopman Operator Background

Koopman operator theory (Koopman, 1931; Brunton et al., 2022) provides a framework for analyzing nonlinear dynamical systems through linear operators. Given a nonlinear dynamical system $\mathbf{x}_{t+1} = F(\mathbf{x}_t)$ on state space $\mathcal{X}$, the Koopman operator $\mathcal{K}$ acts on observables $g : \mathcal{X} \to \mathbb{R}$ as:

$$(\mathcal{K}g)(\mathbf{x}) = g(F(\mathbf{x})). \tag{7}$$

The key insight is that $\mathcal{K}$ is a *linear* operator (on the infinite-dimensional space of observables) even when F is nonlinear. In practice, we approximate $\mathcal{K}$ with a finite-dimensional matrix by choosing a set of observable functions—this is the role of the lifting network ϕ.

In DYNAMO, we use the Koopman formalism not to model temporal dynamics but to provide a structured, composable representation of perturbation effects. The GRN-structured operator (Eq. 5) constrains the space of possible state transitions to those consistent with known regulatory interactions.

C.2. Computational Complexity

Let G be the number of genes, r the POD rank, h the latent dimension, and $|\mathcal{E}|$ the number of GRN edges.

Lifting: $O(G \cdot h \cdot d)$ for per-gene MLPs with embedding dimension d.

Koopman operator: The GRN-masked low-rank operator avoids materializing the full $G \times G$ matrix. With sparse mask $\mathbf{M}$ containing $|\mathcal{E}|$ nonzeros and rank-r_K factors, the matrix-vector product costs $O(G \cdot r_K + |\mathcal{E}|)$.

POD: Computing the perturbation operator from gene embeddings costs $O(d \cdot G \cdot r)$ for the MLP projections. The operator application costs $O(G \cdot r)$ using the low-rank structure (never materializing the $G \times G$ matrix).

Total: $O(G \cdot (h \cdot d + r_K + r) + |\mathcal{E}|)$ per perturbation, linear in the number of genes.