RAPTORGRAPH: GRAPH-BASED PATHWAY MODELING FOR CAUSAL DISCOVERY IN SINGLE-CELL PERTURBATIONS

Yeremia Gunawan Adhisantoso[1], Stephanie Kristin Schröder[1], Maximilian Greß[1], Mikel Hernaez[2,3,4], Jan Voges[1,5,6,7] *

[1] Institute of Information Processing, Leibniz University Hannover, Germany
[2] Computational Biology Program, CIMA University of Navarra, CCUN, IdiSNA, Spain
[3] Data Science and Artificial Intelligence Institute (DATAI), University of Navarra, Spain
[4] Centro de Investigación Biomédica en Red de Cáncer (CIBERONC), Madrid, Spain
[5] L3S Research Center, Leibniz University Hannover, Germany
[6] Peter L. Reichertz Institute for Medical Informatics of the TU Braunschweig and the MHH
[7] Lower Saxony Center for Artificial Intelligence and Causal Methods in Medicine (CAIMed)

ABSTRACT

Experiments involving perturbation of single cells are central to understanding cellular mechanisms and accelerating therapeutic discovery; however, the space of combinatorial perturbations is intractably large. Causal representation learning has shown great promise for predicting unseen combinations of perturbations, but existing methods often suffer from mean collapse and intervention spillover, which violate the theoretical requirements for identifiability guarantees. We introduce RAPTORGraph, an end-to-end VAE framework that addresses these issues through: i) a preconditioned *GraphPathway* encoder that enforces intervention-guided mappings to causal meta-pathways, enabling the clean, single-node latent interventions needed for causal identifiability, and ii) optimal-transport alignment that aligns control and perturbed populations to stabilize conditional generation. Empirical results on Perturb-seq datasets demonstrate that RAPTORGraph improves the trade-off between reconstruction fidelity and distributional matching, and yields improved performance on predicting non-additive combinatorial effects. Finally, we show that RAPTORGraph recovers biologically meaningful latent programs and a causal graph over meta-pathways, providing an interpretable bridge between generative quality and mechanistic insight.

1 INTRODUCTION

Understanding cellular responses to genetic perturbations is fundamental to disease biology and therapeutic discovery (Dixit et al., 2016; Plenge et al., 2013; Norman et al., 2019). However, the combinatorial explosion across ∼25,000 genes makes exhaustive experimentation infeasible, motivating computational models to predict these responses. Existing predictive models face a trade-off between generative quality and causal interpretability. Recent methods like GEARS (Roohani et al., 2024) and scGPT (Cui et al., 2024) achieve strong performance but act as "black boxes", failing to provide mechanistic insights or testable biological hypotheses. While causal representation learning (CRL) approaches like DiscrepancyVAE (dVAE) (Zhang et al., 2023) and SENA (de la Fuente et al., 2025) aim to recover underlying causal factors, they suffer from *intervention spillover*: dense interventional encoders allow sparse gene interventions to influence multiple latent variables, violating identifiability requirements. Furthermore, the destructive nature of scRNA-seq leads to *mean collapse*, where models trained on random pairings of control and perturbed cells regress to the population mean response, eroding cellular diversity.

*Correspondence to: Mikel Hernaez <mhernaez@unav.es> and Jan Voges <voges@tnt.uni-hannover.de>

To address these issues, we present **R**esponse **A**nalysis of **P**erturbed **T**ranscript**O**mes using Inte**R**pretable **Graph** (RAPTORGraph), a framework that ensures causal identifiability through a structured encoder design. Our primary innovation, the GraphPathway layer, enforces sparse mappings from genes to learned interpretable causal factors, which we term causal meta-pathways (causal latent factors). This allows for clean, single-node latent manipulations required for valid causal inference and mechanistic transparency. We complement this with an optimal transport (OT) preprocessing step to mitigate mean collapse by establishing biologically grounded pairings. We benchmark RAPTORGraph on Norman-CPA (Norman et al., 2019) and Replogle2020 (Replogle et al., 2020) datasets, demonstrating its ability to predict complex genetic interactions while uncovering biologically accurate programs such as cell-cycle arrest mediated by *EGR1* and *JUN*.

2 Background

2.1 In silico high-throughput screening models

A key limitation in learning transcriptional responses to unseen genetic perturbations is the destructive nature of scRNA-seq, which prevents longitudinal measurement of the same cell. While paired measurements are infeasible, representation learning can leverage independent observations from control and perturbed populations to predict outcomes of novel perturbations. Existing models utilize varied architectures: Kamimoto et al. (2023) infer linear gene networks, CPA (Lotfollahi et al., 2023) employs latent vector arithmetic, GEARS (Roohani et al., 2024) uses GNNs with prior knowledge, CellOT (Bunne et al., 2023) learns optimal transport maps, and scGPT (Cui et al., 2024) utilizes transformers. However, these non-causal "black box" models lack the mechanistic transparency needed to uncover the causal drivers of transcriptional change. While CRL frameworks like dVAE (Zhang et al., 2023) and SENA (de la Fuente et al., 2025) introduce causal structure, they typically yield either uninterpretable latents or rely on rigid, pre-defined pathways.

2.2 Structural Causal Models for Latent Discovery

We model the underlying data-generating process using a structural causal model (SCM) (Pearl, 2009), assuming high-dimensional samples $\mathbf{x} \in \mathbb{R}^n$ are generated by unobserved causal factors $\mathbf{u} \in \mathbb{R}^d$, where $d \ll n$. The SCM defines these relationships via structural assignments $\mathrm{u}_i = f_i(\mathrm{Pa}_{\mathcal{G}}(\mathrm{u}_i), \mathbf{z}_i)$ for $i = 1, \ldots, d$, where f_i is a structural function, $\mathrm{Pa}_{\mathcal{G}}(\cdot)$ are causal parents under graph $\mathcal{G}$, and $\mathbf{z}_i$ is independent noise. This system induces a directed acyclic graph (DAG) $\mathcal{G}$, implying the joint distribution factorizes as $\mathrm{p}(\mathbf{u}) = \prod_{i=1}^{d} \mathrm{p}(\mathrm{u}_i \mid \mathrm{Pa}_{\mathcal{G}}(\mathrm{u}_i))$. We observe $\mathbf{x}$ through an unknown mixing function $\mathbf{x} = g^*(\mathbf{u})$.

2.3 Identifiability from Targeted Interventions

Identifiability—the unique recovery of $\mathbf{u}$ from $\mathbf{x}$—is theoretically intractable from observational data alone (Pearl, 2009; Khemakhem et al., 2020). Modern CRL achieves this by using interventional data to break statistical symmetries (see Fig. 1). This depends on three requirements established by Schölkopf et al. (2021): i) an acyclic causal structure (DAG), ii) access to atomic interventions targeting each latent variable at least once (Zhang et al., 2023; von Kügelgen et al., 2023; de la Fuente et al., 2025), and iii) faithfulness. Under these requirements, the true latent causal graph is provably recoverable up to a well-defined equivalence class (Zhang et al., 2023).

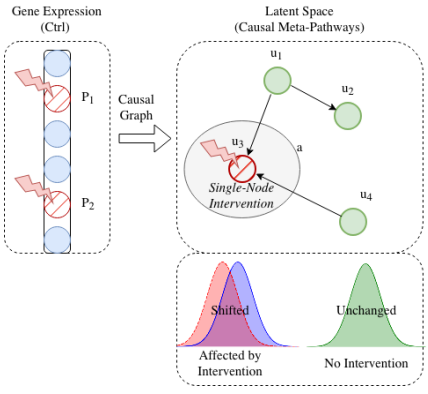


Figure 1: An intervention in gene expression $\mathbf{x}$ shifts the conditional distribution $p^I(\mathrm{u}_i \mid \mathrm{Pa}_{\mathcal{G}}(\mathrm{u}_i))$ of a single causal latent factor u_i, while leaving unrelated factors unaffected.

Requirement 1 (Acyclic Causal Structure). The assumption that causal structures form DAGs is a foundational principle in causal discovery (Pearl, 2009).

Requirement 2 (Atomic Interventions) Causal structure recovery requires access to interventional data that targets each individual latent variable at least once. This

requirement of atomic interventions is a key assumption
in modern CRL (Zhang et al., 2023; von Kügelgen et al., 2023; de la Fuente et al., 2025).

Requirement 3 (Faithfulness). CRL also relies on the faithfulness assumption, where conditional independencies in the data are a consequence of the causal graph structure Zhang et al. (2023).

Following Zhang et al. (2023), their main theorem provides a formal guarantee for identifiability:

Theorem 1 (Full DAG Identifiability, Zhang et al. (2023))**.** *Assume a SCM with a latent DAG $\mathcal{G}$ and an invertible mixing function. If the set of interventions is atomic (targeting each latent variable at least once), and the assumptions of faithfulness hold, then the latent DAG $\mathcal{G}$ and the intervention assignments are identifiable up to permutation, scaling, and translation (the CD-equivalence class).*

The reliance on a set of atomic interventions, where each intervention targets a unique latent factor at least once, is the cornerstone of modern CRL-based prediction of genetic perturbations (Zhang et al., 2023; de la Fuente et al., 2025).

2.4 The Intervention Spillover Problem

While CRL theory provides a robust foundation for identifiability, it encounters a practical hurdle in modern deep learning architectures: *intervention spillover*. Standard interventional encoders f_θ (e.g., MLPs) typically possess a dense Jacobian matrix, which incorrectly transforms a sparse single-gene intervention into a multi-variable distributional shift in the latent space. This "intervention spillover" violates the atomic intervention requirement and confounds downstream causal discovery (see Sec. B.1). RAPTORGraph resolves this by enforcing sparse mapping via preconditioning, ensuring each intervention targets a unique causal meta-pathway by design.

The Dilemma of Prior Knowledge in Causal Discovery. Existing models span a spectrum from purely exploratory (e.g., dVAE) to confirmatory approaches (e.g., SENA) that leverage predefined biological pathways (Herrmann et al., 2024; de la Fuente et al., 2025). However, confirmatory strategies are limited by poor knowledge transfer due to the high cell-type and cell-line specificity of biological pathways (Schneider et al., 2017; Gamazon et al., 2018; Hekselman & Yeger-Lotem, 2020; Walter, 2019). Our framework bridges this gap by learning interpretable causal structures from data without relying on rigid, context-specific pathway definitions.

3 Methods

RAPTORGraph is an end-to-end framework designed to learn causal relationships from interventional single-cell data (Perturb-seq data). It consists of two novel modules: a preconditioned GraphPathway Encoder (Sec. 3.1) that learns a sparse mapping from genes to a set of learned interpretable causal latent factors (hereafter referred to as causal meta-pathways) and a DAGMA-based DAG module (Sec. 3.2) that infers the causal graph between the learnt causal meta-pathways. The proposed design directly addresses the Intervention Spillover Problem ensuring, by construction, that interventions on genes target a unique causal meta-pathway (i.e., latent causal factor). We refer to Sec. C.1 for a detailed description of the model.

3.1 GraphPathway Encoder for Deconfounding through Preconditioning

To address the Intervention Spillover Problem and enforce the theoretical requirement for single-causal-factor interventions, we introduce a novel encoder architecture, the GraphPathway Encoder, which features a deconfounding approach through preconditioning. This approach requires a specific ordering of the input data. Let k be the number of known perturbed genes in the experiment. Without loss of generality, we assume the input gene vector $\mathbf{x} \in \mathbb{R}^n$ is sorted such that the first k genes are those perturbed, followed by the remaining $n - k$ unperturbed genes.

Preconditioning Module (Fig. 2-A). We define d causal meta-pathways such that $d \geq k$, where k represents the number of known perturbed genes. The encoder's primary weight matrix $\boldsymbol{W}$ is structured as a block matrix that enforces a sparse, one-to-one mapping for perturbed genes while allowing dense interactions for the unperturbed components (see Sec. C.2). This diagonal structure ensures each perturbed gene influences exactly one causal meta-pathway, while the critical zero block in $\boldsymbol{W}$ explicitly prevents intervention spillover. The remaining learnable dense submatrices

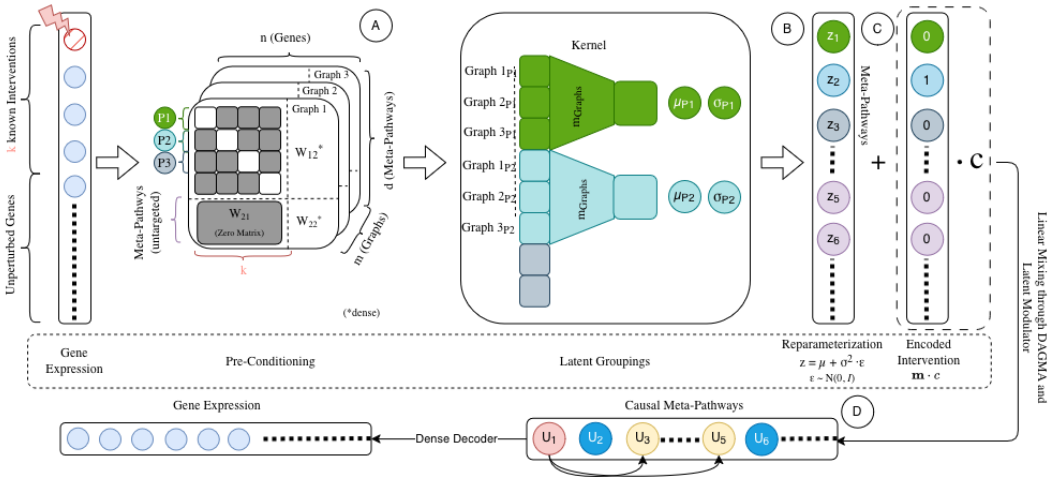


Figure 2: **The GraphPathway Encoder Architecture of RAPTORGraph. A**: A preconditioned linear layer maps control gene expression ($\mathbf{z}_{\text{obs}}$) to latent factors, enforcing sparse, one-to-one connections for perturbed genes to prevent signal spillover. **B**: The model learns multiple parallel subgraph representations to capture complex, non-linear dependencies. **C**: A standard VAE head generates stochastic latent variables ($\mathbf{z}$). **D**: The Latent Modulator applies atomic perturbations (Δ) to these latents, shifting the state from observed ($\mathbf{z}_{\text{obs}}$) to interventional ($\mathbf{z}_{\text{int}}$). Both states are propagated through a learned DAG via the DAGMA layer to yield final causal representations ($\mathbf{u}$), which are then reconstructed into predicted gene expression profiles ($\hat{\mathbf{x}}_{\text{obs}}$, $\hat{\mathbf{x}}_{\text{int}}$) by a dense decoder.

provide the model flexibility to discover complex gene-gene relationships across all unperturbed genes. Crucially, this preconditioned design allows the intervention mask $\boldsymbol{m}$ to be predefined and fixed, satisfying the fundamental structural requirements for identifiability.

Latent Grouping via an Interaction Block (Fig. 2-B). To model the complex non-linear gene-gene dynamics of biological systems, our architecture extends this preconditioning approach by learning multiple distinct subgraph representations for each causal meta-pathway. An intermediate *Interaction Block* then processes these subgraph activations, allowing the model to capture higher-order dependencies by learning unique interaction patterns among the subgraphs within each learned causal meta-pathway (Fig. 2-**B**). This multi-head design allows RAPTORGraph to disentangle the multifaceted effects of a single perturbation, where one gene may simultaneously modulate multiple distinct downstream programs. By learning these patterns directly from data, the model avoids the over-simplification of traditional linear pathway models.

Causal Meta-Pathway Inference via Variational Autoencoder (VAE) (Fig. 2-C). Next, an aggregation layer combines these processed features to compute the VAE parameters ($\mu_{\mathbf{z}}$ and $\sigma_{\mathbf{z}}$) for the causal meta-pathway distributions (Fig. 2-**C**). Thus, the final stochastic representations of the causal meta-pathways are then obtained using the reparameterization trick $\mathbf{z} = \mu_{\mathbf{z}} + \sigma_{\mathbf{z}} \odot \epsilon; \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Causal Meta-Pathway Inference via DAGMA (Fig. 2-D, Sec. 3.2). The model then represents the downstream biological consequences of genetic interventions through the activities of causal meta-pathways. Specifically, a DAGMA layer propagates the VAE-learned (independent) causal meta-pathway activities through a learned directed acyclic graph, yielding the final causally dependent causal meta-pathway activities.

As a final step, a non-linear dense decoder maps these causal states back to the high-dimensional gene expression space to predict the post-perturbation (or unperturbed/control) gene expression. For a comprehensive architectural description, we refer the reader to Sec. C.2.

3.2 Deep Causal Graph Learning with DAGMA

The GraphPathway Encoder with preconditioning eliminates the need for permutation-based discovery. However, fixed ordering makes arbitrary acyclicity constraints (e.g., upper-triangularity) unsuitable, as

they restrict causal hierarchies in ways that may conflict with true biological relationships. To address this, we adapt the DAGMA framework (Bello et al., 2022), which reformulates the combinatorial search for a DAG into a continuous optimization problem by minimizing a data-fit score subject to a differentiable acyclicity constraint. The DAGMA objective $\mathcal{L}_{\text{DAGMA}}(\boldsymbol{A})$ minimizes the negative log-likelihood of the latent causal factors under a linear structural model:

$$\mathcal{L}_{\text{DAGMA}}(\boldsymbol{A}) = \frac{1}{2d} \log\det\left((\boldsymbol{I}-\boldsymbol{A})^\top(\boldsymbol{I}-\boldsymbol{A})\right) + \frac{1}{2}\text{tr}\left((\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\Sigma}_{\mathbf{z}}(\boldsymbol{I}-\boldsymbol{A})^\top\right). \tag{1}$$

where $\boldsymbol{\Sigma}_{\mathbf{z}}$ is the covariance matrix of the detached latent factors $\mathbf{z}_{\text{obs}}$. Acyclicity is enforced via the log-determinant characterization $h_s(\boldsymbol{A}) = -\log\det(s\boldsymbol{I} - \boldsymbol{A}\circ\boldsymbol{A}) + d\log s = 0$, where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is the weighted adjacency matrix, $s > 0$ is a scaling scalar, and $\circ$ represents the Hadamard product. We implement a modified DAGMA approach designed to learn the invariant, baseline causal structure of the biological system. To ensure stability and manage the trade-off between reconstruction fidelity and acyclicity, we introduce two critical regularizations. First, we employ a path-following schedule where the score weight μ follows a decreasing schedule (see Sec. C.3), allowing the model to prioritize feature learning early in training before strictly enforcing the graph constraint. Second, we isolate gradient flow by computing the DAGMA-associated loss $\mathcal{L}_{\text{DAGMA}}$ on detached latent representations. This ensures the acyclicity constraint optimizes the graph structure without distorting the encoder's representation learning. A detailed description of the DAGMA layer, its objective, theoretical background, and implementation can be found in Sec. C.3.

3.3 Optimal Transport for Deterministic Interventional Learning

A fundamental challenge in single-cell interventional modeling is the lack of direct one-to-one mappings between control and perturbed cells, creating a counterfactual gap. Standard approaches often address this by randomly pairing control and interventional samples during training. However, this random pairing matches cells with poor counterfactuals, providing inconsistent gradient signals that force the model to minimize loss by predicting the average of the target distribution, a phenomenon we term *mean collapse* (see Fig. S5). To resolve this, we employ a Wasserstein-based OT framework to establish a deterministic, globally optimal mapping. By minimizing the total transport cost between the control and perturbed populations, OT pairs each cell with its most biologically similar counterpart. Unlike random pairing, which assumes an "average" counterfactual, OT establishes a high-resolution alignment that respects the transcriptomic manifold. This alignment is particularly critical for capturing the responses of rare cell states or transitional populations that are otherwise lost in population-level averages. This preserves heterogeneous subpopulation structures and ensures stable, structure-preserving representations for causal discovery. Mathematical details are provided in Sec. E.

3.4 Training Objective

The model is trained end-to-end by minimizing a composite objective function, $\mathcal{L}_{\text{total}}$, which balances four specialized components to ensure robust representation learning and causal discovery:

$$\mathcal{L}_{\text{total}} = \alpha_{\text{rec}}\mathcal{L}_{\text{MSE}} + \beta_{\text{KL}}\mathcal{L}_{\text{KL}} + \gamma_{\text{MMD}}\mathcal{L}_{\text{MMD}} + \delta_{\text{DAGMA}}\mathcal{L}_{\text{DAGMA}}, \tag{2}$$

where the scalar hyperparameters $\alpha_{\text{rec}}, \beta_{\text{KL}}, \gamma_{\text{MMD}}$, and δ_{DAGMA} balance each term's contribution. Specifically, we employ a reconstruction loss ($\mathcal{L}_{\text{MSE}}$) to maintain high-fidelity control unperturbed cell reconstruction; a KL divergence ($\mathcal{L}_{\text{KL}}$) to regularize the causal meta-pathway manifold; an interventional prediction loss ($\mathcal{L}_{\text{MMD}}$) to align distributions of predicted perturbed gene expressions with those of the experimental ground truth; and a causal graph loss ($\mathcal{L}_{\text{DAGMA}}$) to enforce structural acyclicity and data fit (see Sec. G.4 for the formal definition). A comprehensive derivation and discussion of each term is provided in Sec. G.

3.5 Evaluation Setup, Metrics and Datasets

To ensure a consistent and fair benchmark across all evaluated models, we use Maximum Mean Discrepancy (MMD) to assess the similarity between predicted and true gene expression distributions. The MMD is calculated per type of perturbation (single or combinations thereof) and then averaged to provide an overall measure of distribution similarity. Alongside objective metric comparison, we use Precision@10 to evaluate non-additive interactions of combinatorial perturbations (see Sec. G).

For evaluation we use two datasets: First, the Perturb-seq single-cell RNA-seq dataset from Norman et al. (2019) and as processed in the CPA study (Lotfollahi et al., 2023) and refer to it as the Norman-CPA dataset. It profiles 105 gene perturbations in K562 cells, including both single and double perturbations. In total, 284 conditions across $\sim$108,000 cells were measured, of which 131 represent unique gene–gene combinations and the remaining 153 represent single perturbations. For each perturbation, associated control cells were also profiled. We utilize the entire set of single-perturbations during training and evaluate the subsequent methods on the double-perturbations.

Additionally, we used the CRISPRi single-cell RNA sequencing (scRNA-seq) dataset curated by Replogle et al. (2020) and as processed in the PerturBase study (Wei et al., 2024). We refer to it as the Replogle2020 dataset. This dataset profiles 82 gene perturbations in K562 cells, including both single (44), double perturbations (37), and control. The 82 conditions were measured across $\sim$22,740 cells.

4 EXPERIMENTS

We evaluate RAPTORGraph on two Perturb-seq scRNA-seq datasets: the Norman-CPA dataset (Norman et al., 2019; Lotfollahi et al., 2023), profiling 105 gene perturbations ($\sim$108,000 cells), and the Replogle2020 dataset (Replogle et al., 2020; Wei et al., 2024), profiling 82 perturbations ($\sim$22,740 cells). We assess generative performance using MMD and Mean Squared Error (MSE), and evaluate non-additive interaction prediction via Precision@10 (see Sec. G and Sec. F for details). We benchmark against four state-of-the-art models: SENA, scGPT, dVAE, and GEARS.

Ablation Studies. First, to validate our hyperparameter selection, we conducted a comprehensive ablation study on the VAE regularization parameter (β) and the DAGMA score weights (μ, λ_1). Our analysis reveals a critical trade-off between distribution matching and the preservation of biological heterogeneity. Detailed results of the full hyperparameter sweep and the rationale for the selected configuration are provided in Sec. H.6.

4.1 RECONSTRUCTION AND GENERATIVE CAPABILITIES

We first measured the MSE of control cells to evaluate basal state reconstruction and the MMD for double-gene perturbations to assess global transcriptional shifts. We prioritized the Norman-CPA dataset for comprehensive benchmarking due to baseline implementation constraints, while using Replogle2020 to demonstrate generalizability through a comparison with scGPT.

Table 1: Results using the Norman-CPA dataset are averaged over 3 runs. Bold: best results; Italics: second best. Values are mean $\pm$ variance.

Metric	SENA	scGPT	dVAE	GEARS	RAPTORGraph (ours)
MMD	0.522 ± 0.003	0.176 ± 0.001	$\mathbf{0.076 \pm 0.000}$	–*	*0.112 ± 0.001*
MSE	$\mathbf{0.043 \pm 0.000}$	0.066 ± 0.000	0.062 ± 0.000	–*	*0.045 ± 0.000*

*We excluded the GEARS model because it behaves in a fundamentally different way that is incompatible with our evaluation framework, making a fair and direct comparison impossible (see Sec. G.8).

The results in Table 1 highlight a crucial trade-off between reconstruction fidelity (MSE) and distributional accuracy (MMD). RAPTORGraph achieves a strong overall balance, demonstrating highly competitive performance across both reconstruction and generative capabilities.

The additional benchmark using the Repogle2020 dataset (shown in Table 2 and restricted to a direct comparison between RAPTORGraph and scGPT) further demonstrates our model's robust performance and superior reconstruction fidelity on independent biological data.

4.2 PREDICTION OF NON-ADDITIVE GENETIC PERTURBATIONS.

This complex and biologically significant benchmark assesses a model's ability to predict complex interaction effects between single-gene perturbations that are beyond simple linear combinations (Table 3). Specifically, we compared each model based on their ability to predict five distinct interaction types, using Precision@10 to quantify how effectively each model could identify true interactions within its top 10 predictions, a direct measure of its utility for experimental discovery (see Sec. G.5).

Table 2: Results using the Replogle2020 dataset are averaged over 3 runs. Values are mean ± variance. Bold: best results.

Metric	scGPT	RAPTORGraph (ours)
MMD	0.138 ± 0.000	$\mathbf{0.137 \pm 0.000}$
MSE	0.069 ± 0.000	$\mathbf{0.054 \pm 0.000}$

RAPTORGraph is among the top performers in almost all metrics, yielding the best overall performance across all methods.

Table 3: Comparison of non-additive genetic interaction prediction. Precision@10 scores for five baseline models across distinct genetic interaction subtypes on the Norman et al. dataset. Higher is better. Bold: best results; Italics: second best.Values are mean ± variance over 3 runs.

Metric	dVAE	SENA	GEARS	scGPT	RAPTORGraph (ours)
Synergy	0.000 ± 0.000	0.233 ± 0.153	*0.333 ± 0.115*	0.100 ± 0.000	$\mathbf{0.367 \pm 0.058}$
Suppression	0.000 ± 0.000	0.167 ± 0.115	*0.333 ± 0.115*	$\mathbf{0.400 \pm 0.000}$	0.267 ± 0.153
Neomorphism	*0.333 ± 0.153*	*0.333 ± 0.115*	0.133 ± 0.058	0.100 ± 0.000	$\mathbf{0.433 \pm 0.058}$
Redundancy	$\mathbf{0.933 \pm 0.115}$	0.467 ± 0.058	0.567 ± 0.115	0.100 ± 0.000	*0.800 ± 0.100*
Epistasis	*0.567 ± 0.153*	$\mathbf{0.800 \pm 0.100}$	0.533 ± 0.115	0.400 ± 0.000	0.433 ± 0.115

OT Ablation Study. We further investigated the contribution of OT to this performance. Our ablation study reveals that OT preprocessing is a key driver of the model's ability to predict complex non-additive interactions, particularly boosting redundancy and epistasis. Importantly, OT-based pairing significantly outperforms a random pairing baseline. Note that this benefit comes with negligible computational overhead ($\approx 2.2\%$ of total training time). Detailed analyses of these empirical benefits and the computational cost are provided in Sec. H.5 and Sec. H.3, respectively.

4.3 Reverse Perturbation Analysis

While predicting genetic interactions demonstrates forward predictive power, a more profound test is its ability to reverse-engineer the genetic cause of an observed phenotype. We evaluated the models on their capacity to identify biologically influential genes using the Relevant (Gene Overlap) Hit Rate @ K. RAPTORGraph demonstrates exceptional performance, consistently ranking among the top models, highlighting its utility for hypothesis generation and experimental design. Detailed results and metrics for this task are provided in Sec. H.1.

4.4 RAPTORGraph learns biologically meaningful causal meta-pathways

To confirm that RAPTORGraph learns biologically meaningful causal structures, we examined the functional identity of the learned causal meta-pathways using Gene Set Enrichment Analysis (GSEA) against the MSigDB "Hallmark" gene sets (Liberzon et al., 2015). By identifying significantly enriched biological processes associated with each learned latent factor, we assign verifiable biological labels to the nodes of our graph (see Sec. G.7 for methodology).

The GSEA analysis reveals that RAPTORGraph captures distinct and biologically accurate programs (Fig. 3). We highlight three such programs, verified against the known functions of their corresponding gene perturbations: **Differentiation-Induced Arrest**: The causal meta-pathway linked to *EGR1* displays massive negative enrichment for "E2F Targets" and "G2-M Checkpoint", accurately capturing its role as a tumor suppressor that drives megakaryocytic differentiation and cell cycle exit in K562 cells (Ma et al., 2019). **Stress-Induced Growth Arrest**: The causal meta-pathway linked to the AP-1 subunit *JUN* similarly shows strong negative enrichment for these cell cycle programs, reflecting its role as a central stress-response mediator (Shaulian & Karin, 2002; Shaulian et al., 2000). **P53-Like Tumor Suppression**: The causal meta-pathway associated with *TP73* shows significant negative enrichment for "G2-M Checkpoint", correctly identifying it as a functional homolog of *TP53* that triggers checkpoint signaling (Kaghad et al., 1997). These findings confirm that the nodes in our learned DAG represent specific, identifiable biological mechanisms, allowing the edges learned

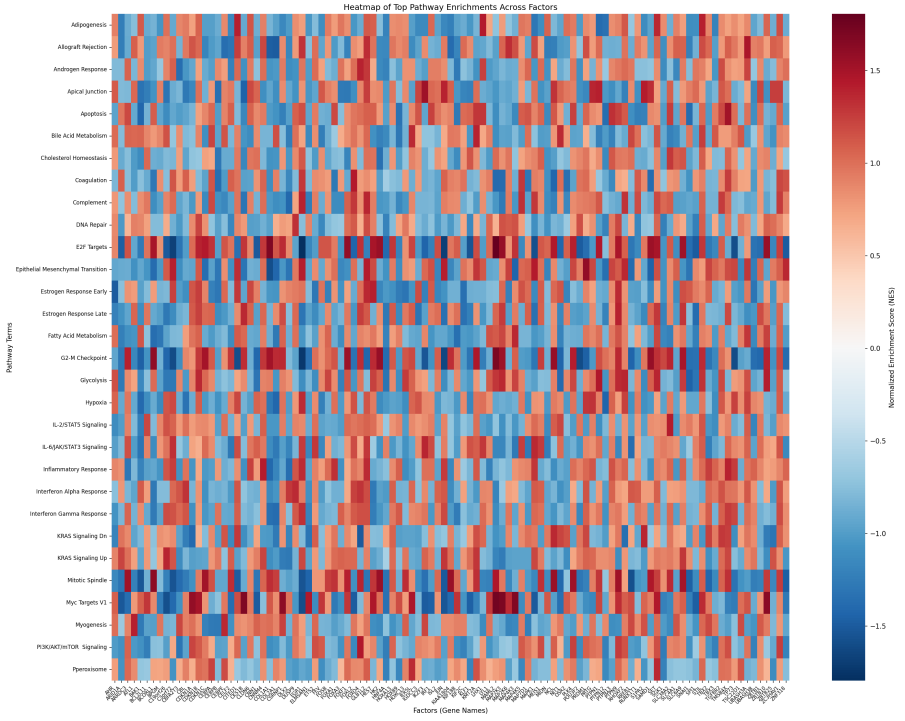


Figure 3: **Top MSigDB Hallmark Pathway Enrichments.** Columns denote architecturally-constrained causal meta-pathways (u_i); color reflects Normalized Enrichment Score (NES). This strong label-pathway concordance confirms the successful encoding of targeted biological functions.

by the DAGMA module to be interpreted as causal regulatory links between verified processes. A detailed description of the methodology and additional validation is provided in Sec. H.2.

5 Conclusion

Developing interpretable *in silico* models is crucial for therapeutic discovery, yet current approaches face two key challenges: intervention spillover, where dense encoders create entangled latent signals that confound causal discovery, and mean collapse, where the absence of one-to-one cell pairings erodes cellular diversity. We introduce RAPTORGraph, a framework that directly resolves these issues. To counteract intervention spillover, it combines a preconditioned GraphPathway layer, enforcing the sparse mappings required for clean single-node interventions, with a DAGMA layer for robust DAG discovery. This architectural design eliminates the artifactual entanglement of input genes with latent pathways, while the subsequent DAG module explicitly learns the legitimate biological causal relationships and correlations between the resulting sparse pathway activities. To prevent mean collapse, RAPTORGraph globally pairs control and perturbed populations through OT. RAPTORGraph not only achieves a superior balance between reconstruction fidelity and distributional accuracy compared to state-of-the-art methods, but also demonstrates a superior capability to predict complex, non-additive genetic interactions, and can reverse-engineer genetic drivers to generate testable hypotheses. RAPTORGraph bridges predictive power and mechanistic insight, enabling not only the prediction of a perturbation's effect but also an understanding of the causal drivers behind it.

Limitations and Future Work. Despite its advantages, our framework assumes a principle of minimal transcriptomic change via OT, which may not hold for drastic cellular reprogramming or differentiation. While our current evaluation focuses on transcriptomics, the framework's modular architecture is generalizable to other single-cell modalities. Future research will extend RAPTOR-Graph to multi-omic integration, incorporating chromatin accessibility and proteomic data to uncover cell-type-specific regulatory mechanisms and discover novel therapeutic targets across diverse clinical contexts.

BROADER IMPACT STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning by improving the interpretability and accuracy of causal discovery in biological systems. By enabling more precise *in silico* predictions of genetic perturbations, our framework potentially accelerates therapeutic discovery and reduces the need for exhaustive experimental screening. While the primary focus is on basic research in cellular biology, we recognize that improved causal modeling of biological data could eventually inform clinical decision-making; thus, we emphasize the importance of independent experimental validation before applying these findings in medical contexts. There are no direct negative societal consequences of this work anticipated at this stage.

ACKNOWLEDGMENTS

The authors acknowledge the financial support by the Ministry of Science and Culture of Lower Saxony through the *zukunft.niedersachsen* program of the Volkswagen Foundation via the CAIMed (Lower Saxony Center for Artificial Intelligence and Causal Methods in Medicine) project under project number ZN4257, and by the German Federal Ministry of Education and Research (BMBF) in the framework of the P4D (Personalisierte, prädiktive, präzise und präventive Medizin zur Verbesserung der Früherkennung, Diagnostik, Therapie und Prävention depressiver Erkrankungen) project under project number 01EK2204F. MH: Ramon y Cajal fellowship (RYC2021-033127-I) and DL2CURE project (PID2023-151980OB-I00) funded by MCIN/AEI/10.13039/501100011033 and "NextGenerationEU"/PRTR; Palatchi Foundation; AECC Proyectos generales 2025 (PRYGN259200HERN). Views expressed herein are solely those of the author(s) and do not necessarily reflect the views of the German Federal Government, the State of Lower Saxony, nor the granting authorities.

REFERENCES

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: Learning DAGs via M-matrices and a log-determinant acyclicity characterization. *Advances in Neural Information Processing Systems*, 35:8226–8239, 2022. doi:10.48550/arXiv.2209.08037. URL https://doi.org/10.52202/068431-0598.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768, 2023. doi:10.1038/s41592-023-01969-x. URL https://doi.org/10.1101/2021.12.15.472775.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21(8):1470–1480, 2024. doi:10.1038/s41592-024-02201-0. URL https://doi.org/10.1101/2023.04.30.538439.

Jesus de la Fuente, Robert Lehmann, Carlos Ruiz-Arenas, Jan Voges, Irene Marin-Goñi, Xabier Martinez-de Morentin, David Gomez-Cabrero, Idoia Ochoa, Jesper Tegner, Vincenzo Lagani, et al. Interpretable causal representation learning for biological data in the pathway space. *arXiv preprint arXiv:2506.12439*, 2025. doi:10.48550/arXiv.2506.12439. URL https://arxiv.org/abs/2506.12439.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, 167(7):1853–1866, 2016. doi:10.1016/j.cell.2016.11.038. URL https://doi.org/10.3410/f.727114584.793554173.

Eric R Gamazon, Ayellet V Segrè, Martijn Van De Bunt, Xiaoquan Wen, Hualin S Xi, Farhad Hormozdiari, Halit Ongen, Anuar Konkashbaev, Eske M Derks, François Aguet, et al. Using an atlas of gene regulation across 44 human tissues to inform complex disease-and trait-associated variation. *Nature Genetics*, 50(7):956–967, 2018. doi:10.1038/s41588-018-0154-4. URL https://doi.org/10.1038/s41588-018-0154-4.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. doi:10.5555/2188385.2188410. URL https://jmlr.org/papers/v13/gretton12a.html.

Idan Hekselman and Esti Yeger-Lotem. Mechanisms of tissue and cell-type specificity in heritable traits and diseases. *Nature Reviews Genetics*, 21(3):137–150, 2020. doi:10.1038/s41576-019-0200-9. URL https://doi.org/10.1038/s41576-019-0200-9.

Moritz Herrmann, F. Julian D. Lange, Katharina Eggensperger, Giuseppe Casalicchio, Marcel Wever, Matthias Feurer, David Rügamer, Eyke Hüllermeier, Anne-Laure Boulesteix, and Bernd Bischl. Position: Why we must rethink empirical research in machine learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18228–18247. PMLR, 2024. doi:10.48550/arXiv.2405.01931.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*, 2016. doi:10.48550/arXiv.1611.01144. URL https://arxiv.org/abs/1611.01144.

Mourad Kaghad, Helene Bonnet, Annie Yang, Laurent Creancier, Jean-Christophe Biscan, Valent Alexandre, Adrian Minty, Pascale Chalon, Jean-Michel Lelias, Xavier Dumont, Pascual Ferrara, Frank McKeon, and Daniel Caput. Monoallelically Expressed Gene Related to p53 at 1p36, a Region Frequently Deleted in Neuroblastoma and Other Human Cancers. *Cell*, 90(4):809–819, August 1997. doi:10.1016/S0092-8674(00)80540-1. URL https://doi.org/10.1016/s0092-8674(00)80540-1.

Kenji Kamimoto, Blerta Stringa, Christy M Hoffmann, Kunal Jindal, Lilianna Solnica-Krezel, and Samantha A Morris. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614(7949):742–751, 2023. doi:10.1038/s41586-022-05688-9. URL https://doi.org/10.1101/2020.02.17.947416.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020. doi:10.48550/arXiv.1907.04809. URL https://proceedings.mlr.press/v108/khemakhem20a.html.

Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P. Mesirov, and Pablo Tamayo. The Molecular Signatures Database Hallmark Gene Set Collection. *Cell Systems*, 1(6):417–425, December 2015. doi:10.1016/j.cels.2015.12.004. URL https://doi.org/10.1016/j.cels.2015.12.004.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, Jay Shendure, Jose L McFaline-Figueroa, Pierre Boyeau, F Alexander Wolf, Nafissa Yakubova, Stephan Günnemann, Cole Trapnell, David Lopez-Paz, and Fabian J Theis. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):e11517, 2023. doi:10.15252/msb.202211517. URL https://doi.org/10.1101/2021.04.14.439903.

Wenjuan Ma, Fang Liu, Lingyan Yuan, Chuan Zhao, and Che Chen. Emodin and AZT synergistically inhibit the proliferation and induce the apoptosis of leukemia K562 cells through the EGR1 and the Wnt/β-catenin pathway. *Oncology Reports*, November 2019. ISSN 1021-335X, 1791-2431. doi:10.3892/or.2019.7408. URL http://www.spandidos-publications.com/10.3892/or.2019.7408.

Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019. doi:10.1126/science.aax4438. URL https://doi.org/10.3410/f.736387820.793564518.

Judea Pearl. *Causality*. Cambridge University Press, 2009. doi:10.1017/cbo9780511803161. URL https://doi.org/10.1017/cbo9780511803161.

Robert M Plenge, Edward M Scolnick, and David Altshuler. Validating therapeutic targets through human genetics. *Nature Reviews Drug Discovery*, 12(8):581–594, 2013. doi:10.1038/nrd4051. URL https://doi.org/10.1038/nrd4051.

Jiangtao Ren, Zhou Liang, and Shaofeng Hu. Multiple kernel learning improved by MMD. In *International Conference on Advanced Data Mining and Applications*, pp. 63–74. Springer, 2010. doi:10.1007/978-3-642-17313-4_7. URL https://doi.org/10.1007/978-3-642-17313-4_7.

Joseph M. Replogle, Thomas M. Norman, Albert Xu, Jeffrey A. Hussmann, Jin Chen, J. Zachery Cogan, Elliott J. Meer, Jessica M. Terry, Daniel P. Riordan, Niranjan Srinivas, Ian T. Fiddes, Joseph G. Arthur, Luigi J. Alvarado, Katherine A. Pfeiffer, Tarjei S. Mikkelsen, Jonathan S. Weissman, and Britt Adamson. Combinatorial single-cell CRISPR screens by direct guide RNA capture and targeted sequencing. *Nature Biotechnology*, 38(8):954–961, August 2020. doi:10.1038/s41587-020-0470-y. URL https://doi.org/10.1038/s41587-020-0470-y.

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

Günter Schneider, Marc Schmidt-Supprian, Roland Rad, and Dieter Saur. Tissue-specific tumorigenesis: context matters. *Nature Reviews Cancer*, 17(4):239–253, 2017. doi:10.1038/nrc.2017.5. URL https://doi.org/10.1038/nrc.2017.5.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. doi:10.1109/JPROC.2021.3058327. URL https://ieeexplore.ieee.org/document/9363920.

Eitan Shaulian and Michael Karin. AP-1 as a regulator of cell life and death. *Nature Cell Biology*, 4 (5):E131–E136, May 2002. doi:10.1038/ncb0502-e131. URL https://doi.org/10.1038/ncb0502-e131.

Eitan Shaulian, Martin Schreiber, Fabrice Piu, Michelle Beeche, Erwin F Wagner, and Michael Karin. The Mammalian UV Response. *Cell*, 103:897–908, December 2000. doi:10.1016/S0092-8674(00)00193-8. URL https://doi.org/10.1016/s0092-8674(00)00193-8.

Julius von Kügelgen, Michel Besserve, Liang Wendong, Luigi Gresele, Armin Kekić, Elias Bareinboim, David Blei, and Bernhard Schölkopf. Nonparametric identifiability of causal representations from unknown interventions. *Advances in Neural Information Processing Systems*, 36:48603–48638, 2023. doi:10.48550/arXiv.2306.02235. URL https://doi.org/10.52202/075280-2110.

Nils G Walter. Biological pathway specificity in the cell—does molecular diversity matter? *Bioessays*, 41(8):1800244, 2019. doi:10.1002/bies.201800244. URL https://doi.org/10.1002/bies.201800244.

Zhiting Wei, Duanmiao Si, Bin Duan, Yicheng Gao, Qian Yu, Zhenbo Zhang, Ling Guo, and Qi Liu. PerturBase: a comprehensive database for single-cell perturbation data analysis and visualization. *Nucleic Acids Research*, 53:D1099–D1111, October 2024. doi:10.1093/nar/gkae858. URL https://doi.org/10.1101/2024.02.03.578767.

Jiaqi Zhang, Kristjan Greenewald, Chandler Squires, Akash Srivastava, Karthikeyan Shanmugam, and Caroline Uhler. Identifiability guarantees for causal disentanglement from soft interventions. *Advances in Neural Information Processing Systems*, 36:50254–50292, 2023. doi:10.48550/arXiv.2307.01207. URL https://doi.org/10.52202/075280-2186.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. DAGs with NO TEARS: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018. doi:10.48550/arXiv.1803.01422. URL https://papers.nips.cc/paper/2018/hash/e347c51419ffb23ca37c160963d91bb2-Abstract.html.

Xiaofeng Zhu, Kim-Han Thung, Ehsan Adeli, Yu Zhang, and Dinggang Shen. Maximum mean discrepancy based multiple kernel learning for incomplete multimodality neuroimaging data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 72–80. Springer, 2017. doi:10.1007/978-3-319-66179-7_9. URL https://doi.org/10.1007/978-3-319-66179-7_9.

A NOMENCLATURE AND MATHEMATICAL NOTATION

In this section, we provide a comprehensive list of the mathematical notations used throughout our work, organized by category. This serves as a centralized reference to ensure clarity and consistency.

A.1 GENERAL VARIABLES AND DATA STRUCTURES

We begin by defining the fundamental variables and data structures used to represent observed and latent spaces.

Symbol	Description
$\mathbb{R}$	The set of real numbers.
$\mathbf{x}$	A vector of observed variables (e.g., gene expression).
$\mathbf{z}$	A vector of general latent variables in the learned representation space.
$\mathbf{e}$	A vector of general noise terms (Gaussian/unspecified).
$\boldsymbol{X}$	A matrix representing a set of observed data samples.
$\boldsymbol{I}$	The identity matrix.

A.2 CAUSAL MODELS AND GRAPHS

Next, we define the components of the Structural Causal Model (SCM) that describes the underlying data-generating process.

Symbol	Description
$\mathcal{G}$	A DAG representing the causal structure.
$\boldsymbol{A}$	The weighted adjacency matrix of the causal graph $\mathcal{G}$.
$\boldsymbol{B}$	A general adjacency matrix.
$\boldsymbol{W}$	A general weight matrix.
$\mathrm{Pa}_{\mathcal{G}}(\cdot)$	A function that returns the parent nodes of a given node in $\mathcal{G}$.
$\bar{\mathcal{G}}$	The transitive closure of the graph $\mathcal{G}$.
$\mathbf{u}$	The true, unobserved latent causal variables in the SCM.
u	A single latent causal variable.
$\mathbf{z}$	A vector of exogenous noise terms for the SCM.
f_i	The structural causal function for a single variable u_i.
$\mathrm{Tr}(\cdot)$	The trace function for a matrix.

A.3 MODEL FUNCTIONS AND REPRESENTATIONS

We denote the core functions of our model, which learn to map between the observed and latent spaces, as follows.

Symbol	Description
g^*	The true, unknown data-generating (mixing) function.
g	The learned decoder network.
f^*	The ideal, oracle encoder function.
f_θ	The practical, parameterized encoder network.
$\hat{\mathbf{z}}$	The learned latent representation, output of the encoder ($f_\theta(\mathbf{x}) = \mu_{\mathbf{z}}$).
$\hat{\mathbf{x}}$	A single data sample generated by the model's decoder.
$\hat{\boldsymbol{X}}$	A matrix representing a batch of data samples generated by the model's decoder.
a	The composed function $f_\theta \circ g^*$.
$\tilde{\mathbf{z}}$	The true latent representation produced by the f^*.
$\Delta\tilde{\mathbf{z}}$	The ideal, single-node change in the latent space.
$\Delta\mathbf{z}$	The approximated (dense) change in the latent space.
$\mathcal{N}$	The normal (Gaussian) distribution.

A.4 Intervention-Specific Variables

Here, we list the variables specifically related to the modeling of interventions.

Symbol	Description
q_{int}	The intervention encoder network.
$\boldsymbol{i}$	The one-hot vector specifying the target of an intervention.
$\boldsymbol{c}$	A general condition vector input to an intervention network.
$\boldsymbol{m}$	The mask vector specifying an intervention's target(s).
c	The scalar strength of a latent intervention.
$\mathbf{z}_{\text{obs}}$	Latents from an observational (control) sample.
$\mathbf{z}_{\text{int}}$	Latents from an interventional sample.
$P_Z^{(i)}$	The interventional distribution over latent variables.
I_k	The set of intervened variables in environment k.
$\boldsymbol{t}_i$	The targets of an intervention for a variable i.
$\boldsymbol{u}$	An auxiliary variable providing conditioning information (e.g., environment index).

A.5 Identifiability and Transformations

We define symbols used in the context of identifiability proofs and the resulting transformation classes.

Symbol	Description
$\sim_T$	Equivalence class for a transformation T.
$\sim_C$	Equivalence class for a transformation C.
$\mathbf{A}$	The matrix component of an affine transformation.
$\mathbf{c}$	The vector component of an affine transformation.
e	A scalar factor in a scaling transformation.
b	An offset in a shift transformation.

A.6 Probability, Losses, and Functions

We define the probability functions, loss functions, and mathematical operators used to train our models and enforce constraints.

Symbol	Description
$\mathrm{P}(\cdot)$	PMF (for discrete variables).
$\mathrm{p}(\cdot)$	PDF (for continuous variables).
$\mathrm{F}(\cdot)$	CDF.
p_{data}	The true, underlying data-generating distribution (PDF).
p_{model}	The learned distribution of the model (PDF).
$\mathbb{E}[\cdot]$	The expectation function.
$\mathcal{L}_{\text{DAGMA}}$	Loss function related to the causal graph structure.
$\mathcal{L}_{\text{MMD}}$	Loss function based on the MMD.
h	A function used to enforce graph acyclicity.
S	A score function for graph discovery.
$\text{MMD}(\cdot, \cdot)$	The MMD function, which computes the distance between two distributions.
$\text{MMD}(\cdot, \cdot)$	The MMD function, which computes the distance between two distributions.
$k(\cdot, \cdot)$	A kernel function used in MMD computation.

A.7 Hyperparameters

We list the key hyperparameters that control our model's training and behavior.

Symbol	Description
λ_{score}	Regularization hyperparameter for a graph score.
σ_{MMD}	The bandwidth hyperparameter for the Gaussian RBF kernel.
β	The weights for the convex combination in multi-kernel MMD.
L	The number of kernels used in the MMD calculation.
m	The number of graphs.
i	The number of samples in a batch of observed data.
j	The number of samples in a batch of predicted data.

A.8 Downstream Analysis Metrics

We define variables used in the downstream evaluation tasks, such as the genetic interaction analysis.

Symbol	Description
Δ	The perturbation effect vector (mean of perturbed minus mean of control).
$\Delta(A+B)_{\text{naive}}$	The naive additive effect vector, defined as $\Delta A + \Delta B$.
$\|\cdot\|_2$	The L2 norm of a vector.
$\cos(\cdot,\cdot)$	The cosine similarity between two vectors.
$\bar{\mathbf{x}}$	The mean expression profile of a cell population.
N	The number of cells for a given perturbation condition.
f_M	The prediction function for a given model M.
Synergy Ratio	The score used to measure synergy and suppression.
Neomorphism Score	The score used to measure neomorphism.
Redundancy Score	The score used to measure redundancy.
Epistasis Score	The score used to measure epistasis.

A.9 Dimensionality and Spaces

Finally, we list symbols for dimensionality and the properties of the data spaces.

Symbol	Description
n	The dimensionality of the observed data space.
d	The dimensionality of the latent space.
d	The number of learned pathways, equivalent to d.
p	The degree of a polynomial mixing function.
$\mathcal{Z}$	The support of the latent distribution.
$\mathcal{X}$	The support of the observed distribution.
n	The number of genes in the expression profile, equivalent to n.

A.10 Acronyms and Abbreviations

List of Acronyms

CDF Cumulative Distribution Function. 14

CRL causal representation learning. 1–3, 17, 19, 22, 29

DAG directed acyclic graph. 2–5, 7, 8, 13, 20, 22, 30

dVAE DiscrepancyVAE. 1–3, 6, 7, 19, 33

EMD Earth Mover's Distance. 34, 35

GRBF Gaussian Radial Basis Function. 28

MMD Maximum Mean Discrepancy. 5–7, 14, 15, 20, 22, 28, 29, 32, 35–38

MSE Mean Squared Error. 6, 7, 20, 28–30, 35–37

OT optimal transport. 2, 5, 7, 8, 25, 34–37

PDF Probability Density Function. 14

PMF Probability Mass Function. 14

RAPTORGraph Response Analysis of Perturbed Transcriptomes using Interpretable Graph. 2–4, 6–8, 20, 33

RKHS Reproducing Kernel Hilbert Space. 28

SCM structural causal model. 2, 3, 13, 17, 19

scRNA-seq single-cell RNA sequencing. 6, 19, 35

VAE Variational Autoencoder. 4, 6, 20–22, 29

B MATHEMATICAL PROOFS AND THEORETICAL BACKGROUND

This appendix provides a detailed theoretical examination of the core challenges in causal representation learning that motivate our proposed architecture. We begin by formally defining and proving the incompatibility of dense encoders via the Intervention Spillover Problem, and then discuss the role of prior knowledge in differentiating causal modeling strategies.

B.1 THE INTERVENTION SPILLOVER PROBLEM

This section provides the formal theoretical background for the *Intervention Spillover Problem*. We formally define this problem, arguing that it is a direct consequence of the inherent properties of dense encoder architectures commonly used in causal representation learning (CRL). We then prove the fundamental incompatibility between these architectures and the requirements for causal identifiability.

Definition 2 (The Intervention Spillover Problem)**.** The *Intervention Spillover Problem* is the phenomenon where a targeted single-variable intervention in a high-dimensional observation space is transformed into a dense, multi-node signal in the lower-dimensional latent space. This issue arises as a direct consequence of using a practical, dense encoder architecture, which inevitably corrupts the intervention's sparsity and creates a fundamental incompatibility with CRL methods that require sparse, single-node latent interventions for identifiability.

B.1.1 MATHEMATICAL PRELIMINARIES AND SETUP

While the problem described is general, this analysis formalizes it within the context of a *linear SCM* for clarity. Let the gene expression space be $\mathbb{R}^n$ and the latent space be $\mathbb{R}^d$, with $d \ll n$. Let $f_\theta : \mathbb{R}^n \to \mathbb{R}^d$ be the practical, differentiable encoder network we train; it maps an observation vector $\mathbf{x}$ to an *approximated* latent vector $\mathbf{z} = f_\theta(\mathbf{x})$. The local behavior of this encoder is described by its Jacobian matrix $\boldsymbol{J}_\theta(\mathbf{x}) \in \mathbb{R}^{d \times n}$.

We model the latent space with a linear SCM, where causal relationships are defined by an adjacency matrix $\boldsymbol{A}$. An intervention on a single gene creates a post-intervention vector $\mathbf{x}_{\text{int}}$ from a corresponding control vector $\mathbf{x}_{\text{obs}}$. This experimental paradigm is characteristic of modern pooled CRISPR screens (e.g., Perturb-seq), where the genetic target of each intervention is known by design.

B.1.2 FORMAL ASSUMPTIONS

Assumption 3 (Ideal Causal Encoder and Single-Node Intervention Target)**.** An *Ideal Causal Encoder*, f^*, is an unknown oracle function that perfectly disentangles the effect of any intervention. The key property is that the *true* latent change, $\Delta\tilde{\mathbf{z}} = f^*(\mathbf{x}_{\text{int}}) - f^*(\mathbf{x}_{\text{obs}})$, corresponds to a single-node perturbation. This ideal latent change is defined as $\Delta\tilde{\mathbf{z}} = \boldsymbol{m} \cdot c$, where c is the latent intervention strength and the target mask $\boldsymbol{m} \in \{0, 1\}^d$ is *one-hot*:

$$\sum_{i=1}^{d} m_i = 1 \tag{S1}$$

In a *closed-world experimental setting*, where the targeted gene is known *a priori*, the behavior of the f^* must fulfill the single-node intervention requirement. The following lemma formalizes this ideal mapping.

Lemma 4 (Equivalence in Ideal Latent Space)**.** *For an Ideal Causal Encoder f^*, operating in a closed-world setting, the latent representation of the post-intervention sample is equivalent to the latent representation of the control sample plus the ideal single-node latent perturbation. Formally:*

$$f^*(\mathbf{x}_{int}) = f^*(\mathbf{x}_{obs}) + \boldsymbol{m} \cdot c \tag{S2}$$

Proof. The proof follows directly from the definition of the ideal latent change in theorem 3. We start with the identity $\Delta\tilde{\mathbf{z}} = f^*(\mathbf{x}_{\text{int}}) - f^*(\mathbf{x}_{\text{obs}})$. Substituting the single-node perturbation gives $f^*(\mathbf{x}_{\text{int}}) - f^*(\mathbf{x}_{\text{obs}}) = \boldsymbol{m} \cdot c$. Rearranging the terms yields the desired result. □

Assumption 5 (Practical Model Approximation and Dense Jacobian)**.** Our *Practical Model* uses an encoder $f_\theta(\mathbf{x})$ and a perturbation inference network q_{int}. The underlying encoder network has a *dense Jacobian matrix* $\boldsymbol{J}_\theta(\mathbf{x})$.

Remark 6 (Architectural Bias vs. Learned Function). It is important to distinguish between the architecture of the encoder and the function it learns. While it is theoretically possible for a densely-connected network to learn an effectively sparse Jacobian, the architecture itself possesses a strong inductive bias towards dense mappings. Standard training objectives provide no direct incentive for learning a sparse function.

B.1.3 Main Result: The Incompatibility

Lemma 7 (Incompatibility of the Practical Model)**.** *Given the practical model's reliance on a dense encoder network (theorem 5), there is no guarantee that an intervention on a single gene will produce a single-node latent intervention target vector $\boldsymbol{m}$ as required by the ideal model in theorem 3.*

Proof. The proof relies on the direct conflict between the output of the practical encoder and the requirements of the ideal model in a closed-world setting. In this setting, we have access to both the control sample $\mathbf{x}_{\text{obs}}$ and the post-intervention sample $\mathbf{x}_{\text{int}}$. Our trained practical encoder, f_θ, is a known deterministic function (e.g., a trained MLP).

When we process these known samples, the encoder produces an approximated latent change:

$$\Delta\mathbf{z} = f_\theta(\mathbf{x}_{\text{int}}) - f_\theta(\mathbf{x}_{\text{obs}}) \tag{S3}$$

Due to the dense Jacobian of f_θ (theorem 5), each component of the latent vector $\mathbf{z}$ is a function of all input components in $\mathbf{x}$. Consequently, a change in a single gene will result in a change across nearly all latent dimensions. The resulting vector $\Delta\mathbf{z}$ is therefore dense.

The objective for a conditional network q_{int} is to learn to predict this latent change. For the model to perfectly capture the intervention effect, the learned perturbation $(\boldsymbol{m} \cdot c)$ must match the observed latent change. This implies minimizing the loss:

$$\mathcal{L} = ||\Delta\mathbf{z} - (\boldsymbol{m} \cdot c)||^2 \tag{S4}$$

To achieve a loss of zero, the learned perturbation $(\boldsymbol{m} \cdot c)$ must be equal to the dense vector $\Delta\mathbf{z}$. This forces the learned target mask $\boldsymbol{m}$ to be dense, which directly contradicts the one-hot requirement for a single-node intervention as defined in Eq. (S1). Therefore, the practical model's architecture is fundamentally misaligned with the goal of identifying single-node latent interventions. □

B.1.4 Implications for Causal Discovery in Dense Encoders

This mathematical reality creates a cascade of problems for causal discovery models that rely on a learned intervention network to identify the latent target of a perturbation. The core of the issue is the need to make a discrete, single-target selection. This is typically addressed using a temperature-scaled softmax function, a mechanism that mimics the Gumbel-Softmax trick for differentiable categorical sampling (Jang et al., 2016). Given the output logits, $\boldsymbol{o}$, from an intervention network, the probability for each target is computed as:

$$\pi_i = \frac{\exp(\boldsymbol{o}_i/\tau)}{\sum_{k=1}^{d} \exp(\boldsymbol{o}_k/\tau)} \tag{S5}$$

where $\tau > 0$ is a temperature hyperparameter that controls the sharpness of the output distribution.

As the temperature $\tau \to 0$, the softmax output approaches a discrete, one-hot vector (a proxy for a Dirac delta function), representing a single, "hard" choice. However, this creates a fundamental contradiction when applied to the output of a dense encoder. The encoder produces a dense latent change vector, $\Delta\mathbf{z}$, where the evidence suggests a multi-node perturbation. To satisfy the model's objective of selecting a single target, a low temperature τ must be used to force the softmax into a "winner-take-all" state. This forces the model to ignore the dense evidence and declare a single winner, which can lead to it learning spurious, causally unfaithful mappings.

The temperature-scaled softmax is therefore caught in an architectural dissonance: it is a mechanism forced to make a discrete choice that fundamentally misrepresents the dense, continuous signal it receives from the encoder.

Furthermore, in the closed-world setting of a single-cell RNA sequencing (scRNA-seq) experiment, the mapping from a perturbed gene to its corresponding latent variable is known beforehand. Therefore, the task of "exploratory perturbation" (i.e., discovering which latent variable was the target) is not actually necessary. The true challenge is to learn the *scale* or *power* of the intervention's effect on the causal representation, not its target, a task for which these architectures are ill-suited.

B.2 The Role of Prior Knowledge in Causal Modeling

The theoretical gaps in existing models can be understood by positioning them on a spectrum between two complementary modes of scientific inquiry: exploratory and confirmatory research (Herrmann et al., 2024). Exploratory research is concerned with hypothesis generation; it involves analyzing data to uncover novel patterns, structures, and relationships without a strong pre-existing theory. In contrast, confirmatory research is concerned with hypothesis testing; it begins with a specific, pre-formulated hypothesis and uses data to determine if that hypothesis is supported or refuted. This distinction helps clarify the intended purpose and inherent trade-offs of different causal models. For instance, DiscrepancyVAE (dVAE) is primarily exploratory, aiming to discover latent variables and their causal graph from scratch (Zhang et al., 2023). In contrast, SENA takes a more targeted approach, acting as a confirmatory model for pathways while exploring perturbation effects; it **confirms** latent variables as known biological pathways and explores which pathway is targeted by a perturbation (de la Fuente et al., 2025).

The approach by de la Fuente et al. (2025) highlights the close relationship between SCM and CRL. By assuming its causal variables (the biological pathways) are known *a priori*, it operates like a traditional SCM. However, because it must still *learn* a representation of how to map raw gene expression data onto these variables and then discover the causal graph between them, it firmly resides within the CRL paradigm. It represents a specific, constrained form of CRL where prior knowledge heavily guides the representation learning process. This confirmatory design choice is motivated by the goal of enhancing the biological interpretability of the learned causal model.

However, this reliance on predefined pathways introduces significant limitations, primarily stemming from the challenge of knowledge transfer. Biological pathways exhibit strong cell-type and cell-line specificity, meaning that mechanisms characterized in one cellular context often do not generalize to another (Schneider et al., 2017; Gamazon et al., 2018; Hekselman & Yeger-Lotem, 2020). This context dependence poses a fundamental barrier to transferring pathway insights across different *in vitro* models and ultimately to patient tissues (Walter, 2019). Consequently, the findings from confirmatory models are difficult to apply to less-researched cell types, limiting the scalability and broader applicability of these approaches. Furthermore, this confirmatory strategy may not be suitable for smaller-scale experiments that only incorporate a limited set of genes, making it difficult to map observations to comprehensive pathway definitions.

C Model Architecture

This section provides a comprehensive overview of the core architectural components of our model, detailing both the novel GraphPathway Encoder and the deep causal graph learning framework.

C.1 Overview of RAPTORGraph framework

The architecture of the **R**esponse **A**nalysis of **P**erturbed **T**ranscript**O**mes using Inte**R**pretable **Graph** (RAPTORGraph) framework, illustrated in Fig. S4, is designed as an end-to-end system for learning causal relationships from interventional single-cell data. The framework is composed of three primary modules: a preconditioned encoder for sparse representation learning, a DAG discovery layer for identifying causal structure, and a decoder for reconstructing cellular states.

The input to the model is the high-dimensional gene expression data. The first layer of the encoder is our novel preconditioned GraphPathway Encoder, which addresses the Intervention Spillover Problem. By leveraging prior biological knowledge, it enforces a sparse mapping from the thousands of observed genes ($\mathbb{R}^n$) to a lower-dimensional set of learned meta-pathways ($\mathbb{R}^d$). This ensures that a single-gene intervention activates only a sparse, well-defined set of learned meta-pathways in the latent space, fulfilling the requirements for valid downstream causal inference.

The resulting latent representations, $\mathbf{z}$, are then passed to the DAGMA-based DAG discovery module. This layer operates directly on the latent variables to learn the causal relationships *between* the learned meta-pathways. Its primary output is a directed acyclic graph ($\mathcal{G}$), represented by its adjacency matrix, which models the flow of biological influence among the learned meta-pathways.

Finally, the decoder network reconstructs the original gene expression profile from the latent representation. This ensures that the latent variables, in addition to being causally structured, also retain sufficient information to accurately capture the full cellular state. The model is trained end-to-end by optimizing a composite loss function that balances these different objectives (see Fig. S4).

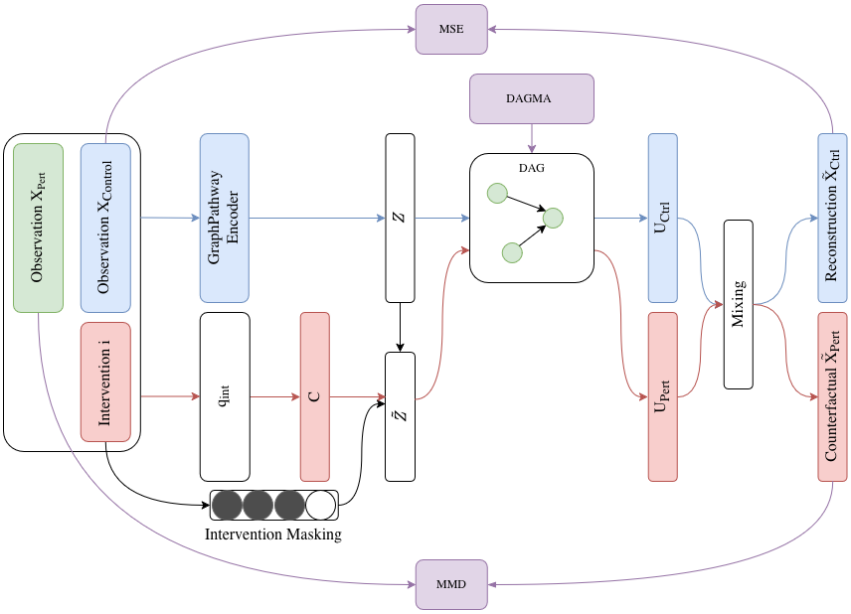


Figure S4: Detailed schematic of the RAPTORGraph framework. The preconditioned GraphPathway Encoder enforces a sparse mapping from the high-dimensional gene expression space ($\mathbb{R}^n$) to the learned meta-pathways ($\mathbb{R}^d$). The DAGMA-based DAG operates on the latent representations to learn a directed acyclic graph ($\mathcal{G}$) representing the causal relationships between the learned meta-pathways. The model is trained end-to-end using a composite loss function comprised of several terms: a reconstruction loss (Mean Squared Error (MSE)), a distributional loss (MMD) on interventional predictions, a KL Divergence on the Variational Autoencoder (VAE) latent variables, and an acyclicity constraint from the DAGMA layer.

C.2 GRAPHPATHWAY ENCODER ARCHITECTURE

The GraphPathway Encoder is a custom neural network layer designed to solve the Intervention Spillover Problem through a principled architectural design. It combines a preconditioned weight matrix to deconfound known perturbation effects with a multi-subgraph structure to capture complex, non-linear gene interactions.

C.2.1 DECONFOUNDING THROUGH PRECONDITIONING

The foundational component of the encoder is its preconditioned weight matrix, which enforces the single-node intervention assumption required for causal identifiability. As described in Sec. 3.1, the layer's primary weight matrix $\boldsymbol{W}$ is structured as a block matrix with a diagonal top-left submatrix. This structure guarantees that each known perturbed gene has a direct, isolated influence on exactly one latent pathway, while the zero-block explicitly prevents its effect from spilling over into other pathways. By explicitly blocking this spillover, the design transforms the encoder into a deterministic modulator for known interventions, allowing the intervention mask $\boldsymbol{m}$ to be fixed by the architecture rather than being a learned (and potentially confounded) variable.

C.2.2 MODELING NON-LINEAR INTERACTIONS WITH SUBGRAPHS

To move beyond a purely linear model and capture the complex, non-linear dynamics of biological processes, the architecture learns m distinct representations for each pathway. This is achieved by first applying a masked linear layer that maps the input genes $\mathbf{x}$ to an intermediate representation. This layer's weight matrix is derived by repeating the base preconditioned mask m times, creating m parallel, similarly conditioned subgraphs that are initialized differently to learn diverse feature mappings. The initial activation vector, $\mathbf{s}^{(g)}$, for each subgraph g is computed as:

$$\mathbf{s}^{(g)} = \rho(\boldsymbol{W}^{(1,g)}\mathbf{x} + \mathbf{b}^{(1,g)}) \quad \text{for } g = 1, \ldots, m \tag{S6}$$

where each layer-specific weight matrix $\boldsymbol{W}^{(1,g)}$ has the same block-diagonal structure.

An intermediate *Interaction Block* then processes these subgraph activations to model higher-order dependencies. This block applies a series of pathway-specific linear transformations and non-linear activations to the set of m activations within each pathway; this enables the model to learn complex interactions between the different feature mappings before the final aggregation step.

Finally, the processed subgraph activations are aggregated into the final pathway representations using a grouped 1D convolution. This operation has a kernel size of m and is configured with d groups, making it equivalent to applying a separate, pathway-specific linear layer to the set of m subgraph activations for each pathway. This learns an optimal linear combination of the non-linear subgraph features to produce the final parameters for the latent distributions, $\mu_{\mathbf{z}}$ and $\sigma_{\mathbf{z}}$.

C.2.3 IMPLEMENTATION DETAILS OF GRAPHPATHWAY

The specific implementation used in our model is the 'InteractingGraphPathwayLayer'. This layer encapsulates the preconditioning, multi-subgraph mapping, and interaction blocks into a single module. To model complex, non-linear biological processes, each pathway learns its own unique set of interaction weights within the interaction block. This design provides greater expressive power, as it allows the model to capture distinct interaction dynamics for each biological process. When configured for a variational autoencoder, the output channel dimension of the final grouped convolution is doubled, allowing the layer to directly produce both the mean $\mu_{\mathbf{z}}$ and variance $\sigma_{\mathbf{z}}$ for each latent pathway.

C.3 DEEP CAUSAL GRAPH LEARNING WITH DAGMA

This subsection details the DAGMA framework for causal discovery, the rationale for its adaptation within our deep learning model, and the key implementation principles required for stable and effective graph learning in a VAE context.

C.3.1 THEORETICAL BACKGROUND: FROM NOTEARS TO DAGMA

A fundamental challenge in causal discovery is the combinatorial nature of learning a DAG, as the number of possible graphs grows super-exponentially with the number of variables. A breakthrough was achieved by NOTEARS (Zheng et al., 2018), which reformulated this discrete problem into a continuous optimization problem amenable to gradient-based methods. The core innovation was a differentiable function, $h(\boldsymbol{A})$, whose value is zero if and only if the weighted adjacency matrix $\boldsymbol{A}$ represents a DAG. The original NOTEARS constraint is based on the matrix exponential:

$$h(\boldsymbol{A}) = \mathrm{Tr}(e^{\boldsymbol{A}\circ\boldsymbol{A}}) - d = 0 \tag{S7}$$

where $\mathrm{Tr}(\cdot)$ is the trace of the matrix and $\boldsymbol{A} \circ \boldsymbol{A}$ is the element-wise (Hadamard) product. This function cleverly sums all cycles of all possible lengths in the graph; it equals zero only if no cycles exist.

DAGMA (Bello et al., 2022) builds upon this continuous optimization framework but introduces a new, log-determinant-based acyclicity characterization that often leads to faster and more stable convergence. The DAGMA acyclicity constraint is:

$$h_s(\boldsymbol{A}) = -\log\det(s\boldsymbol{I} - \boldsymbol{A}\circ\boldsymbol{A}) + d\log s = 0 \tag{S8}$$

where $s > 0$ is a scaling scalar. This function is based on the principle that a matrix corresponds to a DAG if and only if all eigenvalues of its squared adjacency matrix are zero. The log-determinant acts as a barrier, approaching infinity as the graph's structure approaches a cycle, while being minimized at zero exclusively for DAGs.

C.3.2 THE DAGMA OPTIMIZATION FRAMEWORK

The full optimization problem is to minimize a score function that measures how well the graph fits the data, subject to the acyclicity constraint. For a linear model with latent variables $\mathbf{z}$, the score function $S(\boldsymbol{A})$ is typically the mean squared error of reconstruction:

$$S(\boldsymbol{A}) = \frac{1}{2n}\|\mathbf{z} - \mathbf{z}\boldsymbol{A}\|_F^2 \tag{S9}$$

The original DAGMA paper proposes a path-following approach that solves a sequence of unconstrained optimization problems:

$$\min_{\boldsymbol{A}}(\mu \cdot S(\boldsymbol{A}) + h_s(\boldsymbol{A})) \tag{S10}$$

where μ is a hyperparameter that is gradually decreased towards zero during training. As $\mu \to 0$, the penalty on violating acyclicity becomes infinitely strong, forcing the final solution for $\boldsymbol{A}$ to be a DAG.

C.3.3 RATIONALE FOR A MODIFIED DAGMA IN CAUSAL REPRESENTATION LEARNING

Integrating DAGMA into a VAE for interventional CRL requires careful design to ensure the model learns a meaningful causal graph. Our implementation is guided by two core principles.

Principle 1: Learning the Invariant Graph from Observational Data. A foundational assumption is that the causal graph $\boldsymbol{A}$ represents an invariant, underlying mechanism of the system. Therefore, the DAGMA loss, which encourages the learning of this structure, must be computed exclusively on the latent representations of observational data ($\mathbf{z}_{\text{obs}}$). This allows the model to learn the fundamental causal rules of the system's natural dynamics. The interventional data and their latent representations ($\mathbf{z}_{\text{int}}$) are used to compute a separate discrepancy loss (e.g., MMD). This second loss teaches the model the consequences of breaking the causal rules, evaluating how well the learned graph functions as a simulator for interventions. Applying the graph loss to interventional latents would create a conflicting objective, as an intervention by definition breaks the natural mechanism the graph is supposed to represent.

Principle 2: Isolating Gradient Flow via Input Detachment. The most critical design principle for stability is isolating the gradient flow between the VAE's encoder and the DAGMA layer. This is achieved by detaching the latent variable tensor ($\mathbf{z}_{\text{obs}}$) before it is used in the score calculation. This

design choice has two key benefits. First, it mimics the original DAGMA algorithm, which operates on a fixed dataset. Second, and more importantly, it creates a separation of concerns: the encoder learns to produce a rich representation, and the DAGMA layer learns the graph of that representation. Without detaching, gradients from the DAGMA score would flow back to the encoder, encouraging it to learn a trivial, simplistic latent space that is easy to fit, thereby destroying the quality of the representation.

C.3.4 IMPLEMENTATION DETAIL OF DAGMA

Several practical considerations are crucial for the stable application of DAGMA in a deep learning context.

Weight Initialization and Invertibility. To ensure stability from the start of training, the weights of the adjacency matrix $\boldsymbol{A}$ are initialized from a uniform distribution in the range $[-0.1, 0.1]$. This provides the optimizer with small, non-zero weights as a starting point. This initialization also ensures that $(\boldsymbol{I} - \boldsymbol{A})$ is likely well-conditioned and invertible, a property critical for the forward pass of the linear DAGMA model. The forward pass solves the structural equation $\mathbf{z} = \mathbf{z}\boldsymbol{A} + \mathbf{z}$ for $\mathbf{z}$, which requires computing $\mathbf{z}(\boldsymbol{I} - \boldsymbol{A})^{-1}$. To handle potential numerical instability, our implementation first attempts to solve the linear system directly. If this fails due to a singular or ill-conditioned matrix, it robustly falls back to using the Moore-Penrose pseudo-inverse, guaranteeing a stable forward pass throughout training.

Input Normalization. A common failure mode is the collapse of the graph weights in $\boldsymbol{A}$ to a trivial near-zero solution. This is prevented by applying a normalization layer (e.g., 'LayerNorm') to the latent inputs ($\mathbf{z}_{\text{obs}}$ and $\mathbf{z}_{\text{int}}$) before they enter the DAGMA layer. This ensures the score term has a consistent magnitude, providing a strong and stable gradient signal that encourages the model to learn a non-trivial graph.

Numerical Precision. The acyclicity constraint $h_s(\boldsymbol{A})$ is theoretically guaranteed to be non-negative. However, due to floating-point limitations, it can sometimes become a very small negative number. To handle this, the calculated value is clamped at zero, which respects the theoretical constraint while gracefully handling practical numerical artifacts.

D Training Configurations and Analysis Setup

D.1 Model Architectures and Training Hyperparameters

To establish a fair and reproducible benchmark, we standardized key architectural and training hyperparameters for all models, primarily based on the default configurations provided in their respective publications and source code. This approach ensures that performance differences can be attributed to the models' core methodologies rather than variations in setup. Table S4 details the specific architectural parameters, such as latent dimension size and the number of transformer layers, while Table S5 outlines the training parameters used for each model, including optimizer, learning rate, and batch size. Notably, gradient clipping was employed exclusively for the scGPT model, a standard practice for ensuring numerical stability in large transformer-based architectures.

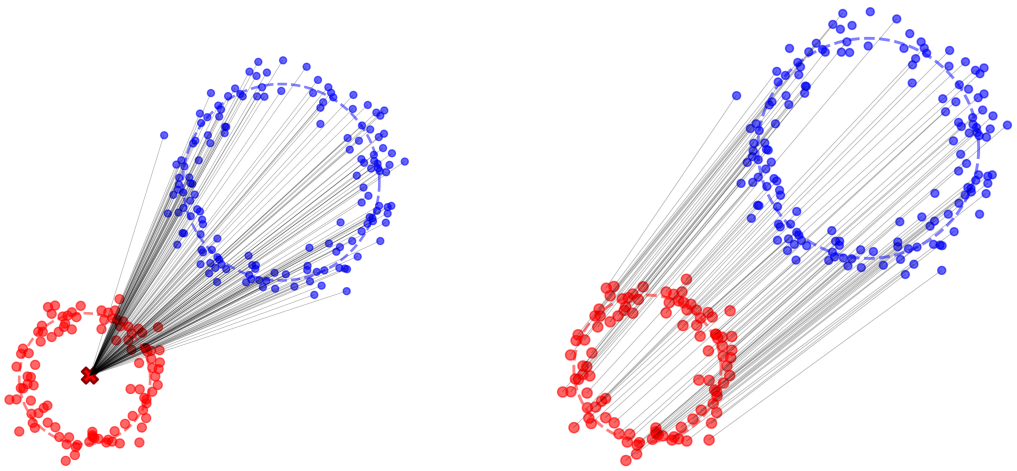
For a complete list of hyperparameters and reproducibility instructions, please refer to the 'configs/' directory in our code repository and the accompanying 'README.md'.

Table S4: Detailed Model Architectures

Hyperparameter	**dVAE**	**SENA**	**scGPT**	**RAPTORGraph (Ours)**
Latent Dimension (z_{dim})	105	105	N/A	109
Embedding Size ($embsize$)	N/A	N/A	512	N/A
Transformer Layers ($nlayers$)	N/A	N/A	12	N/A
Attention Heads ($nheads$)	N/A	N/A	8	N/A
Feed-Forward Dimension (d_{hid})	N/A	N/A	512	904
Hidden Layers (Encoder)	1 (128 units)	1 (128 units)	N/A	N/A
Activation Functions	LeakyReLU	LeakyReLU	GELU	SiLU

Table S5: Detailed Training Hyperparameters

Hyperparameter	**dVAE**	**SENA**	**scGPT**	**RAPTORGraph (Ours)**
Optimizer	Adam	Adam	AdamW	AdamW
Learning Rate	1×10^{-3}	1×10^{-3}	1×10^{-4}	1×10^{-4}
Training Batch Size	32	32	16	256
Evaluation Batch Size	32	32	32	256
Epochs	100	100	30	100



(a) Random pairing (b) Optimal Transport

Figure S5: **Comparison of random pairing vs. Optimal Transport.** (a) Random pairing collapses the source distribution towards the target mean (mean collapse). (b) Optimal Transport pairing preserves the underlying population structure and biological heterogeneity.

E Rationale For Global Pairing through Optimal Transport

As discussed in Sec. 3, random pairing of control and perturbed cells leads to mean collapse. Here, we detail the mechanism behind this phenomenon and the mathematical formulation of our optimal transport (OT) solution.

E.1 Mechanism of Mean Collapse

For a given interventional sample (e.g., cell i, in which gene A was perturbed), a randomly selected observational sample (e.g., cell j) from the batch is highly unlikely to be an appropriate counterfactual. The biological state of cell j may differ substantially from that of cell i prior to the perturbation. Consequently, the model must learn a transformation from an often noisy and biologically irrelevant initial state. Across thousands of training iterations, a single interventional sample is paired with hundreds of different random controls. Each pairing provides a distinct and noisy gradient signal. The optimizer seeks to minimize the loss on average across these inconsistent pairings and converges on a simple solution: predicting the mean of the target distribution. The random noise from the poor counterfactuals averages to zero, leaving only the central tendency as a stable learning signal. Ultimately, the objective function effectively becomes an expectation over these random pairings. The model learns that the most reliable strategy to achieve a low expected loss is to ignore the specifics of the randomly chosen input and produce an average output. The optimization landscape is thus smoothed in a manner that makes the 'mean prediction' an attractive local minimum.

Rather than creating noisy one-to-one random pairs, OT considers the entire population of observational and interventional latents. It computes an optimal flow of probability mass from the observational to the interventional distribution. This flow represents the most efficient method of morphing one entire point cloud into the other. The flow implicitly creates meaningful pairings. Each interventional sample is matched with the observational sample(s) that are closest in gene expression space, providing the best possible counterfactuals within the given populations. The theoretical properties of OT ensure that it robustly preserves the inherent heterogeneous subpopulation structures by minimizing the ground movement cost, a feature critical for reliable counterfactual prediction.

E.2 MATHEMATICAL FORMULATION

Let $\mu_s = \sum_{i=1}^{n} \frac{1}{n} \delta_{\mathbf{z}_{\text{obs}}^{(i)}}$ and $\nu_t = \sum_{j=1}^{m} \frac{1}{m} \delta_{\mathbf{z}_{\text{int}}^{(j)}}$ be the empirical measures of the source (control) and target (perturbed) latent distributions. The Optimal Transport problem seeks a transport plan $\gamma \in \mathbb{R}_{+}^{n \times m}$ that minimizes the total cost:

$$\min_{\gamma \in \Pi(\mu_s, \nu_t)} \langle \gamma, \mathbf{C} \rangle_F \tag{S11}$$

where $\mathbf{C}_{ij} = \|\mathbf{z}_{\text{obs}}^{(i)} - \mathbf{z}_{\text{int}}^{(j)}\|_2^2$ is the squared Euclidean cost matrix, and $\Pi(\mu_s, \nu_t)$ is the set of joint probability measures with marginals μ_s and ν_t. The resulting optimal plan γ^* provides a probabilistic coupling. To obtain a deterministic mapping for training, we use the barycentric projection map, ensuring each control cell is paired with its optimal counterfactual representation in the perturbed domain.

F Data and Evaluation Details

F.1 Training and Test Data Partitioning

To create a robust benchmark for evaluating the generalization capabilities of each model, we partitioned the Norman et al. (2019) dataset based on the type of perturbation. The training dataset was composed of the complete set of control cells (i.e., those with non-targeting guides) and the complete set of cells from all available **single-gene perturbation** conditions. This approach ensures that the models are trained on the fundamental effects of individual gene knockdowns. The test dataset, conversely, consisted exclusively of cells from all **double-gene perturbation** conditions. These combinatorial perturbations were entirely held out from the training process. This training/test split was designed to specifically assess each model's ability to perform out-of-distribution prediction, where the primary task is to predict the effects of unseen combinatorial perturbations based on the learned effects of their individual constituent perturbations.

G Evaluation Metrics and Loss Functions

In this section, we provide detailed definitions for the key metrics used to quantitatively assess model performance. To ensure a comprehensive evaluation, we employed a suite of metrics targeting different aspects of model capability. We used the MSE for evaluating the reconstruction of basal cell states, the MMD for evaluating the accuracy of out-of-distribution predictions for perturbed states, and a set of non-additive interaction scores to assess the model's ability to capture complex biological phenomena. The calculation of these metrics was standardized across all benchmarked tools.

G.1 Mean Squared Error

The MSE was used to measure each model's ability to faithfully reconstruct the gene expression profile of control (unperturbed) cells, with lower MSE indicating superior reconstruction. To ensure unbiased evaluation, the control population was randomly partitioned into two non-overlapping halves: an input set and a ground truth set. The evaluation proceeded in a batch-wise manner, where a batch of cells from the input set was fed to the model to generate reconstructions. The MSE was then calculated between these reconstructed cells and a corresponding batch from the ground truth set. The final reported MSE is the mean of the scores from all batches. For a single batch of i cells with n genes, the MSE is calculated as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{i \cdot n} \sum_{i=1}^{i} \sum_{j=1}^{n} (\mathbf{x}_{ij} - \hat{\mathbf{x}}_{ij})^2 \tag{S12}$$

where $\mathbf{x}_{ij}$ denotes the expression level of the jth gene in the ith true control cell, whereas $\hat{\mathbf{x}}_{ij}$ represents the predicted expression of the jth gene in the ith reconstructed control cell.

G.2 Maximum Mean Discrepancy

The MMD (Gretton et al., 2012) is a metric used to measure the distance between two probability distributions based on the distance between their mean embeddings in a Reproducing Kernel Hilbert Space (RKHS). It was used to measure the dissimilarity between the distribution of model-predicted gene expression profiles and the distribution of true, experimentally observed profiles for a given perturbation.

G.2.1 The Kernel Function

The MMD relies on a kernel function, $k(\cdot, \cdot)$, which is a symmetric, positive definite function that computes a notion of similarity between two samples. A common choice for the kernel, and the one used in our work, is the Gaussian Gaussian Radial Basis Function (GRBF) kernel:

$$k(\mathbf{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{y}\|^2}{2\sigma_{\text{MMD}}^2}\right) \tag{S13}$$

where σ_{MMD} is the bandwidth hyperparameter.

G.2.2 Formal Definitions of MMD

The squared MMD between the true data distribution p_{data} and the model's learned distribution p_{model} is defined in its population form as:

$$\begin{aligned} \text{MMD}^2(p_{\text{data}}, p_{\text{model}}) = \; & \mathbb{E}_{\mathbf{x},\mathbf{x}' \sim p_{\text{data}}}\big[k(\mathbf{x}, \mathbf{x}')\big] \\ & - 2\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}, \hat{\mathbf{x}} \sim p_{\text{model}}}\big[k(\mathbf{x}, \hat{\mathbf{x}})\big] \\ & + \mathbb{E}_{\hat{\mathbf{x}},\hat{\mathbf{x}}' \sim p_{\text{model}}}\big[k(\hat{\mathbf{x}}, \hat{\mathbf{x}}')\big] \end{aligned} \tag{S14}$$

In practice, we use batches of data and compute the unbiased empirical estimator. Given a batch of i true samples $\{\mathbf{x}_i\}_{i=1}^{i} \sim p_{\text{data}}$ from matrix $\boldsymbol{X}$ and a batch of j predicted samples $\{\hat{\mathbf{x}}_j\}_{j=1}^{j} \sim p_{\text{model}}$ from matrix $\hat{\boldsymbol{X}}$, the estimator is:

$$\begin{aligned}\text{MMD}_u^2(\boldsymbol{X}, \hat{\boldsymbol{X}}) = &\frac{1}{i(i-1)} \sum_{i \neq j}^{i} k(\mathbf{x}_i, \mathbf{x}_j) \\ &+ \frac{1}{j(j-1)} \sum_{i \neq j}^{j} k(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) \\ &- \frac{2}{ij} \sum_{i=1}^{i} \sum_{j=1}^{j} k(\mathbf{x}_i, \hat{\mathbf{x}}_j)\end{aligned} \tag{S15}$$

G.2.3 Multi-Kernel Implementation

MMD (Ren et al., 2010; Zhu et al., 2017) generalizes the standard MMD by employing a convex combination of L distinct kernels (Gretton et al., 2012). The squared MMD is defined as a weighted sum of individual squared MMDs:

$$\text{MMD}^2(p_{\text{data}}, p_{\text{model}}) = \sum_{l=1}^{L} \beta_l \cdot \text{MMD}_{k_l}^2(p_{\text{data}}, p_{\text{model}}) \tag{S16}$$

where $\beta_l \geq 0$ are the weights assigned to each kernel.

G.2.4 Rationale for Using MMD

The choice of MMD over MSE for evaluating interventional predictions is motivated by two key factors. First, MMD captures holistic distributional shifts, including changes in variance, skewness, and modality, whereas MSE is only sensitive to the mean. This is critical for modeling the heterogeneous biological response to a perturbation. Second, state-of-the-art CRL models for interventional data often do not produce one-to-one pairings between predicted and ground-truth samples, making MSE computation impossible. MMD is a two-sample test that compares two sets of samples directly, which aligns perfectly with this setting.

G.2.5 Hyperparameter Selection for MMD

The performance of the MMD test is highly sensitive to the kernel bandwidth, σ_{MMD}. To establish a fair and consistent benchmark across all models, it was necessary to standardize this parameter. We noted that different models, such as DiscrepancyVAE and SENA (de la Fuente et al., 2025), use different default values. The implementations often define a fix_sigma parameter, which relates to the standard kernel variance by fix_sigma $= 2\sigma_{\text{MMD}}^2$. To standardize, we aligned with the value used in SENA. Our benchmark, therefore, uses a MMD implementation with a base fix_sigma set to 200.0 for all evaluations, which corresponds to a variance σ_{MMD}^2 of 100.0. This approach generates a series of kernels with varying bandwidths derived from this base value, ensuring the metric's scale and sensitivity were consistent and robust.

G.3 KL Divergence

The Kullback-Leibler (KL) divergence term is a standard component of the VAE framework used to regularize the learned latent space. It measures the information lost when the learned posterior distribution, $q(\mathbf{z}|\mathbf{x})$, is used to approximate a predefined prior distribution, $p(\mathbf{z})$. In our model, we assume an isotropic Gaussian prior, $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The KL divergence term, $\mathcal{L}_{\text{KL}}$, encourages the encoder to produce latent representations that are both smooth and well-structured, preventing the model from assigning each input sample to a single, isolated point in the latent space (a form of overfitting). The term is calculated as:

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) = -\frac{1}{2} \sum_{j=1}^{d} \left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right) \tag{S17}$$

where μ_j and σ_j are the mean and standard deviation produced by the encoder for the j-th latent pathway.

G.4 Causal Graph Loss

The causal graph structure between learned meta-pathways is identified by optimizing the DAGMA-based objective function (Bello et al., 2022). The loss function, $\mathcal{L}_{\text{DAGMA}}$, is computed on the latent representations of observational data to learn the invariant causal rules of the biological system. The objective integrates three key components: a data-fit score that measures how well the graph explains the observed dependencies, an L1 penalty to encourage structural sparsity, and a differentiable acyclicity constraint to ensure the resulting graph is a DAG. The full causal graph loss is defined as:

$$\mathcal{L}_{\text{DAGMA}} = \mu \left(S(\boldsymbol{A}; \boldsymbol{X}) + \lambda_1 \|\boldsymbol{A}\|_1 \right) + h(\boldsymbol{A}) \tag{S18}$$

where $S(\boldsymbol{A}; \boldsymbol{X})$ is the data-fit score (typically the MSE of linear structural assignments in the latent space), $\|\boldsymbol{A}\|_1$ is the L1 norm of the adjacency matrix $\boldsymbol{A}$ controlled by sparsity hyperparameter λ_1, and $h(\boldsymbol{A})$ is the log-determinant acyclicity penalty. The hyperparameter μ controls the trade-off between the data-fit score and the acyclicity constraint, following a path-following schedule during training as detailed in Sec. C.3.2.

G.5 Non-Additive Genetic Interaction Analysis

To evaluate the models' ability to predict non-additive effects, we designed a genetic interaction analysis inspired by the Precision@10 metric introduced in the GEARS paper (Roohani et al., 2024). While the original study used a robust regression model, we opted for a more direct naive additive baseline, defined as the vector sum of single-perturbation effects over control ($\Delta_A + \Delta_B$). We calculated five interaction scores by comparing the predicted double-perturbation effect (Δ_{AB}) to this baseline. This unified metric was applied to all models to ensure a fair comparison.

The analysis follows a multi-step process for each double-gene perturbation. First, "effect vectors" (Δ) are calculated for single and double perturbations by subtracting the mean gene expression profile of control cells from the mean profile of perturbed cells. This is done for both ground truth data and model predictions. The naive additive baseline assumes the double perturbation effect is a linear sum of individual effects:

$$\Delta(A+B)_{\text{naive}} = \Delta A + \Delta B \tag{S19}$$

By comparing the true or predicted $\Delta(A+B)$ to the individual and naive additive vectors, we compute raw scores for five distinct genetic interaction subtypes.

- *Synergy* and *Suppression* are measured using a ratio of magnitudes:

$$\text{Synergy Ratio} = \frac{\|\Delta(A+B)\|_2}{\|\Delta(A+B)_{\text{naive}}\|_2} \tag{S20}$$

- *Neomorphism* measures changes in the direction of the effect:

$$\text{Neomorphism Score} = 1 - \cos\left(\Delta(A+B), \Delta(A+B)_{\text{naive}}\right) \tag{S21}$$

- *Redundancy* measures functional overlap:

$$\text{Redundancy Score} = \min\left(\cos\left(\Delta(A+B), \Delta A\right), \cos\left(\Delta(A+B), \Delta B\right) \right) \tag{S22}$$

- *Epistasis* measures dominance:

$$\text{Epistasis Score} = \left| \cos\left(\Delta(A+B), \Delta A\right) - \cos\left(\Delta(A+B), \Delta B\right) \right| \tag{S23}$$

Final performance is measured using Precision@10. For each interaction type, perturbations are ranked by their predicted score, and the top 10 are selected. These are compared against a ground truth set of interactors, defined as those whose true score falls in the top 25% (or bottom 25% for suppression) for that category. The Precision@10 score is the fraction of correct predictions in the top 10.

G.6 REVERSE PERTURBATION ANALYSIS

To assess the models' understanding of genotype-phenotype relationships, we implemented an in silico reverse perturbation prediction task, inspired by the analysis in the scGPT study (Cui et al., 2024). This task challenges a model to infer the causal genetic perturbation that induced a given transcriptomic state.

The analysis is conducted on a specific subset of 20 genes from the Norman et al. (2019) dataset, creating a controlled combinatorial search space. For each unique perturbation condition p, we establish a ground truth mean expression profile, $\bar{\mathbf{x}}_p$, by averaging the expression vectors of all cells for that condition:

$$\bar{\mathbf{x}}_p = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{x}_{p,i} \tag{S24}$$

where N_p is the number of cells under perturbation p, and $\mathbf{x}_{p,i} \in \mathbb{R}^n$ is the expression profile of the i-th cell. The central component is the creation of a comprehensive in silico prediction database restricted to the 20-gene subset. For each model, we generate a predicted expression profile $\hat{\mathbf{x}}_{A+B}$ for every unique combination of two genes, A and B. This is achieved by feeding the model the ground truth mean control profile, $\bar{\mathbf{x}}_{\text{ctrl}}$, and conditioning it on the desired double perturbation. The prediction function f for a given model is expressed as:

$$\hat{\mathbf{x}}_{A+B} = f(\bar{\mathbf{x}}_{\text{ctrl}}, \text{pert} = A + B) \tag{S25}$$

This results in a key-value database where each key is a double-perturbation identifier from the 20-gene subset, and the value is the corresponding predicted mean expression vector in $\mathbb{R}^n$.

With the prediction database established, we query it using the ground truth mean expression profiles of the experimentally observed double perturbations from the 20-gene subset. For each true double perturbation p_{true} present in the test set, its ground truth mean profile $\bar{\mathbf{x}}_{p_{\text{true}}}$ is used as a query vector. We then calculate the Euclidean distance between this query vector and every predicted vector $\hat{\mathbf{x}}_{p_{\text{pred}}}$ in the database:

$$d(\bar{\mathbf{x}}_{p_{\text{true}}}, \hat{\mathbf{x}}_{p_{\text{pred}}}) = \sqrt{\sum_{j=1}^{n} (\bar{\mathbf{x}}_{p_{\text{true},j}} - \hat{\mathbf{x}}_{p_{\text{pred},j}})^2} \tag{S26}$$

The perturbations in the database are subsequently ranked based on their proximity to the query vector, from the smallest to the largest Euclidean distance. This yields a ranked list of predicted perturbations that are most likely to have caused the observed cellular state.

To evaluate the ranking performance, we employ the "Hit Rate @ K" metric, which measures the frequency of successful retrievals within the top K predictions. We define two variants of this metric to capture different aspects of prediction accuracy.

Correct Hit Rate @ K: This is a stringent metric that measures the proportion of queries where the exact true perturbation is correctly identified within the top K ranked predictions. It is formally defined as:

$$\text{Correct Hit Rate @ K} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbb{I}(\text{rank}(p_q) \leq K) \tag{S27}$$

where $\mathcal{Q}$ is the set of all query perturbations, p_q is the true perturbation corresponding to query q, and $\mathbb{I}$ is the indicator function, which is 1 if the condition is met and 0 otherwise.

Relevant Hit Rate @ K: This is a more lenient metric that assesses whether the model can identify at least one of the correct genetic components of a combinatorial perturbation. A prediction is considered "relevant" if its set of perturbed genes has a non-empty intersection with the set of genes in the true perturbation. The metric is defined as:

$$\text{Relevant Hit Rate @ K} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbb{I}(\exists \hat{p} \in \text{TopK}(q) \text{ s.t. genes}(\hat{p}) \cap \text{genes}(p_q) \neq \emptyset) \tag{S28}$$

where $\text{TopK}(q)$ is the set of top K predicted perturbations for query q, and $\text{genes}(p)$ is the set of constituent genes in perturbation p. This metric rewards models that can correctly identify influential genes, even if the exact combination is not perfectly predicted. Together, these two metrics provide a comprehensive evaluation of each model's ability to reverse-engineer the mapping from cellular phenotype back to genetic cause.

G.7 Gene Set Enrichment Analysis

To validate the semantic meaning of the learned Directed Acyclic Graph, we performed a systematic Gene Set Enrichment Analysis. We employed an in-silico perturbation strategy to define the biological identity of each latent variable:

1. Latent Perturbation: For each causal latent factor $\mathbf{z}_i$ associated with a known gene perturbation, we artificially activated the factor in the latent space by setting the intervention vector Δ such that the target component $\Delta_i = 1$ and all other components $\Delta_{j \neq i} = 0$.
2. Counterfactual Decoding: We decoded this perturbed latent state back to the gene expression space $\mathbb{R}^n$ to generate a batch of "counterfactual" expression profiles.
3. Differential Analysis: We computed the mean differential expression vector $\delta = \bar{x}_{int} - \bar{x}_{ctrl}$ relative to the control baseline.
4. Enrichment: This vector was used to create a ranked gene list, which was analyzed using the `gseapy` framework against the MSigDB Hallmark 2020 gene set collection.

This process allows us to independently verify if the latent factor $\mathbf{z}_i$ actually encodes the biological function of its assigned gene g_i, assigning a verifiable "biological label" to the nodes of our learned graph.

G.8 Rationale for the Exclusion of GEARS from the Benchmark

In designing this benchmark, our goal was to establish a fair and methodologically consistent framework for comparing models that predict the full distribution of single-cell gene expression profiles following perturbation. While we considered including other prominent models such as GEARS, we ultimately excluded it from the final comparison due to fundamental incompatibilities between its predictive paradigm and our standardized evaluation framework. Primary reasons for its exclusion include: (i) its prediction of a single population-average vector rather than a cell distribution, which is incompatible with our MMD metric; and (ii) its non-equivalent handling of the control state, where it uses the precomputed mean of true control cells as a baseline rather than learning to reconstruct them. Given these significant differences, we concluded that a fair and direct comparison between GEARS and the other models within our established framework was not possible.

H EXTENDED RESULTS

H.1 EXTENDED RESULTS: REVERSE PERTURBATION ANALYSIS

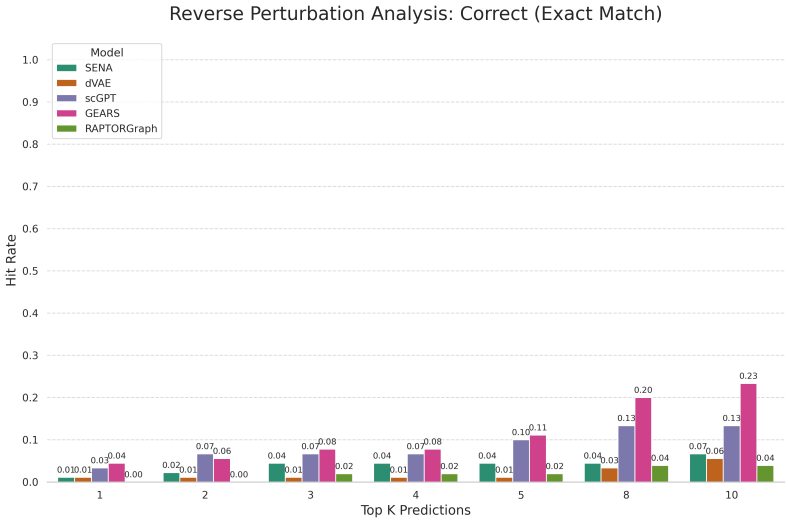


Figure S6: Hit Rate @ K (correct) for the reverse perturbation task. The metric queries where the exact true perturbation is correctly identified within the top K ranked predictions. Results are averaged over 3 runs.

Table S6: Hit Rate @ K (relevant) for the reverse perturbation task. The metric measures the fraction of queries where at least one correct gene was identified in the top K predictions. Higher values are better. Values are mean ± variance over 3 runs.

k	dVAE	SENA	GEARS	scGPT	RAPTORGraph (ours)
1	0.333 ± 0.115	0.244 ± 0.019	**0.344 ± 0.051**	0.233 ± 0.000	0.294 ± 0.059
3	0.556 ± 0.051	0.444 ± 0.069	0.578 ± 0.051	0.533 ± 0.000	**0.608 ± 0.122**
5	0.733 ± 0.088	0.600 ± 0.058	0.722 ± 0.077	0.700 ± 0.000	**0.784 ± 0.090**

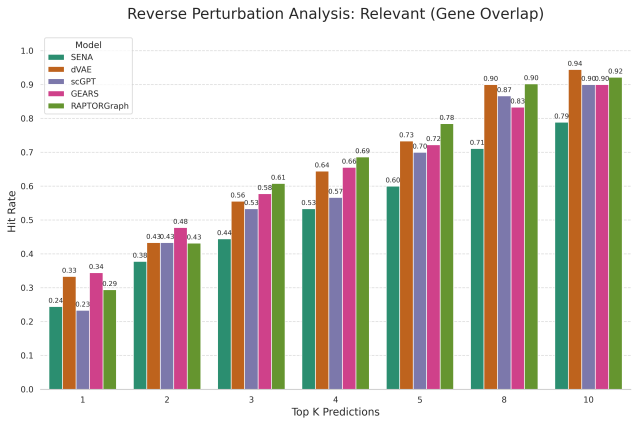


Figure S7: Hit Rate @ K (relevant) for the reverse perturbation task. The metric measures the fraction of queries where at least one correct gene was identified in the top K predictions. Higher values are better. Results are averaged over 3 runs.

While predicting genetic interactions demonstrates forward predictive power, a more profound test is its ability to reverse-engineer the genetic cause of an observed phenotype. To this end, we challenged a model to identify the specific genetic perturbation responsible for a given cellular gene expression profile. The analysis was performed on a controlled combinatorial space of 20 transcription factors from the Norman et al. (2019) dataset. For each model, we generated a database of predicted expression profiles for all possible double-gene perturbations within this subset. We then used the ground truth mean expression profiles from experimentally observed double perturbations as queries, ranking the database entries by similarity to identify the most likely causal perturbation.

We first evaluated the models on the highly stringent task of identifying the exact causal gene combination, quantified by the Correct (Exact Match) Hit Rate @ K (Fig. S6). However, a more practical measure of a model's utility to guide experimental genetic perturbation designs is its ability to retrieve perturbations containing at least one of the correct genes. We thus evaluated each model's capacity to identify biologically influential genes using the Relevant (Gene Overlap) Hit Rate @ K (see Fig. S7 and Table S6).

H.2 Extended Results: Biological Validation Details

Detailed GSEA methodology and further validation examples are provided below.

H.3 Extended Results: Computational Complexity of Optimal Transport Preprocessing

We evaluated the computational cost of the proposed OT pairing strategy compared to a random baseline. While random pairing scales linearly ($O(N)$), the OT approach involves solving the Earth Mover's Distance (EMD) problem, which typically exhibits a worst-case time complexity of $O(N^3 \log N)$.

To mitigate this, the OT pairing problem is formulated by taking the entire control population ($N_{ctrl} = 8,907$) and pairing it independently with each specific intervened population (i.e., cells sharing the same perturbation label, N_k in the hundreds). This effectively segments the total perturbed dataset ($N = 108,497$) into many smaller, manageable OT problems, significantly reducing the N in the $O(N^3 \log N)$ complexity for each individual transport plan and enabling efficient parallelization.

We quantified the overhead of OT preprocessing on a workstation equipped with an Intel Core i9-9900K CPU (8 cores, 16 threads) and 64 GB RAM. The OT preprocessing added approximately 34 seconds to the baseline execution time (Random: 1m 44s vs. OT: 2m 18s). Detailed profiling confirmed that the process is compute-bound but highly efficient:

- **Parallelization**: The process achieved a peak CPU utilization of 1140%, confirming that the strategy effectively saturates available cores (≈ 11.4 active threads).
- **Memory**: Peak memory usage was modest at 2.4 GB.

To contextualize this cost, we benchmarked the total training time for a standard 100-epoch experiment on an NVIDIA RTX 3090 GPU (≈ 25.8 minutes). The 34-second OT overhead represents only 2.2% of the total experimental runtime. Given that OT is critical for preserving causal identifiability and preventing mean collapse, this negligible computational cost presents a highly favorable trade-off.

H.4 Extended Results: Empirical Analysis of Loss Functions for Sparse Single-Cell Causal Inference

Training causal inference models on scRNA-seq data presents unique challenges due to two fundamental properties of the data: high sparsity and the lack of paired ground truth observations (unpaired control and perturbed populations). To address this, we utilize OT as a preprocessing step to infer latent couplings, combined with MMD or reconstruction losses.

To rigorously demonstrate the necessity of this approach compared to baselines (e.g., Random Pairing or MMD-only objectives), we conducted an empirical ablation study using synthetic data. This controlled environment allows us to overcome the "missing pair" problem inherent in real biological data, where the lack of ground truth counterfactuals makes it impossible to definitively validate causal alignment. The code to reproduce these results is available in our source code.

We utilized synthetic data for this analysis for three critical reasons:

1. **Unobservable Counterfactuals:** In real scRNA-seq, measuring a cell destroys it, making it impossible to observe the same cell in both control and perturbed states (x_i and y_i). Synthetic data allows us to fabricate ground truth pairs, providing access to latent ground truth to definitively quantify causal accuracy.
2. **Controlled Sparsity Sweep:** Real data has fixed, high sparsity. Synthetic data allows us to sweep sparsity from 0% to 99% to empirically observe the phase transition where reconstruction metrics degrade.
3. **Signal Isolation:** It allows us to isolate the mathematical properties of the loss functions from biological noise and batch effects.

We simulated a dataset of 1000 samples with 5000 features (genes).

- **Control Population (X):** Generated from $\mathcal{N}(0, 1)$ and masked to achieve target sparsity levels ranging from 0% to 99%.
- **Perturbation:** A systematic shift was added to the first 50 features ($+2.0$) to simulate a biological effect.
- **Ground Truth (Y):** The perturbed state for each control cell, preserving the sparse structure.

We evaluated three hypothetical model outputs against the ground truth Y, representing the outcomes of different training strategies:

1. **Causal/Paired Model (Ideal):** The ground truth Y with added Gaussian noise ($\sigma = 0.1$). *Representation:* This proxies the outcome of a model trained with OT-based pairing. By recovering the latent pairs ($x_i \to y_i$), the model learns the correct cell-specific trajectories.
2. **Population/Generative Model (Unpaired):** The ground truth population Y with sample indices randomly permuted. This proxies the equilibrium state of a model trained with MMD only. Such a model perfectly matches the target distribution $P(Y)$ (minimizing the MMD loss) but fails to learn the causal mapping ($f(x_i) \neq y_i$), effectively becoming a generative model of the population rather than a causal predictor.
3. **Mode Collapse (Average):** The mean vector $\bar{Y}$ repeated for all samples. This represents the failure mode of Random Pairing (MSE). When training on random pairs, the model minimizes variance by predicting the population mean.

We evaluated three metrics in this extended results:

- **MSE:** Standard point-wise reconstruction loss.
- **MMD:** Distributional distance using a multi-scale Gaussian kernel.
- **OT:** Exact EMD (Wasserstein-2).

H.4.1 Results

We evaluated the metrics across varying sparsity levels (see Table S7) to identify the failure modes of standard losses.

Table S7: Metric Performance Across Sparsity Regimes. Lower is better. Bold indicates the metric's "preferred" model.

Sparsity	Metric	Causal/Paired (OT+MMD)	Population/Generative (MMD Only)	Mode Collapse (MSE Failure)	Verdict
0% (Dense)	**MSE**	**0.0100**	1.9947	1.0000	**Success.**
	MMD	**−0.0061**	**−0.0062**	2.3127	**Ambiguity.**
50% (Medium)	**MSE**	**0.0100**	0.9987	0.4989	**Degradation.**
	MMD	−0.0059	**−0.0062**	2.3119	**Failure.**
90% (High)	**MSE**	**0.0100**	0.1997	0.0997	**Weakness.**
	MMD	−0.0022	**−0.0062**	2.3066	**Failure.**
99% (Extreme)	**MSE**	0.0100	0.0201	**0.0101**	**Failure.**
	MMD	0.1648	**−0.0061**	2.2505	**Failure.**
	OT	49.95	**0.00**	50.26	**Failure.**

The key finding is in the "Population/Generative" column. Across all sparsity levels, MMD assigns a near-perfect score (≈ 0.0) to the Population Model. This implies that a model trained with MMD as the sole objective has no incentive to learn the correct causal mapping. It can achieve zero loss simply by generating a realistic population that is causally scrambled (i.e., Control Cell A is mapped to Perturbed State B). The result is that the model learns the *distribution* but loses the *cell identity*.

Even if we attempted to force pairing using standard reconstruction (MSE), the high sparsity (99%) causes MSE to favor the Mode Collapse (0.0101) over the correct structure (0.0100). This empirically demonstrates that training on random pairs forces the model to converge to the population mean, which on sparse data is indistinguishable from a valid prediction in terms of error magnitude.

The empirical results provide definitive proof that standard approaches are insufficient for sparse, unpaired causal inference:

1. **MMD-Only Training** leads to Causal Scrambling (Permutation Invariance).
2. **Random Pairing / MSE Training** leads to Mode Collapse (Convergence to Mean).

This justifies the necessity of **OT preprocessing**. By solving for the optimal coupling matrix π that minimizes transport cost, we explicitly enforce the Principle of Minimal Action, assuming that cells undergo the smallest necessary transcriptomic change. This effectively infers the latent pairing that MSE assumes exists, resolving the identifiability crisis that MMD ignores and avoiding the collapse mode that Random Pairing induces.

H.5 Extended Results: Impact of Optimal Transport on Downstream Tasks

While the primary motivation for integrating OT is to prevent mean collapse and preserve population structure during training, its ultimate value lies in improving performance on downstream biological tasks. We conducted an ablation study comparing the standard OT-based pairing against a Random Pairing baseline to isolate the impact of this preprocessing step on the model's ability to predict non-additive genetic interactions.

We evaluated both models on the "Genetic Interaction" benchmark (Precision @ 10). As shown in Table S8, the inclusion of OT leads to a marked improvement in identifying complex interaction types, specifically Redundancy, Epistasis, and Suppression.

The results indicate that while Random Pairing can achieve competitive performance on simpler metrics (like Neomorphism), it falls short in capturing the subtle dependencies required to identify Redundancy and Epistasis. The OT pairing, by matching cells based on their distributional similarity, likely preserves the underlying biological signal that defines these interactions, preventing them from being washed out by the noise of random assignment. This validates the hypothesis that OT is not just a theoretical necessity for causal identifiability but a practical enhancer of model utility for complex biological discovery.

Table S8: Ablation Study: Impact of Optimal Transport on Genetic Interaction Prediction (Precision @ 10). The model trained with OT preprocessing consistently outperforms the Random Pairing baseline in detecting complex interaction types (Redundancy, Epistasis, Suppression), demonstrating that OT helps preserve the fine-grained causal structure required for these tasks.

Interaction Type	**Random Pairing**	**Optimal Transport**
Neomorphism	0.4	0.4
Redundancy	0.7	**0.8**
Epistasis	0.8	**0.9**
Synergy	**0.5**	0.4
Suppression	0.2	**0.3**

H.6 Extended Results: Ablation Study on β-VAE and DAGMA Regularization

The primary goal of this ablation study is to rigorously assess the sensitivity of the RAPTORGraph model to its key regularization parameters, specifically the β parameter of the β-VAE (β) and the weights of the DAGMA loss (μ and λ_1). Our core hypotheses for this study were: 1. The criticality of β for preventing mode collapse, where we investigated if β is the most crucial hyperparameter for maintaining a regularized and informative latent space, looking for evidence of KL divergence approaching zero and its detrimental effects on learning meaningful representations. 2. The hierarchy of hyperparameter importance, testing the assertion that β is of primary importance, followed by the DAGMA weights, in terms of their impact on downstream performance.

The study was conducted using the RAPTORGraph model on the Norman et al. 2019 dataset. For the hyperparameter sweep, β was varied across $\{0.0, 0.1, 0.4, 0.8, 1.2, 1.6, 2.0, 2.5, 3.0\}$ to observe its impact on KL divergence and to test for mode collapse. Additionally, μ and λ_1 were varied around default values to analyze their influence on the causal graph structure, specifically $\mu \in \{0.1, 0.27, 0.5\}$ and $\lambda_1 \in \{0.02, 0.05, 0.1\}$. The evaluation metrics included: a) KL Divergence as the primary metric to monitor for signs of mode collapse; b) Prediction Variance (Pred. Var.), representing the variance of the reconstructed gene expression for *control* cells, used to detect mean collapse; and c) MMD to evaluate the quality of predictions for *intervened or perturbed* cells. Notably, MSE is not explicitly used for mode collapse detection in this context, as variance directly assesses the diversity of generated outputs, which MSE alone might obscure.

To better visualize the dominant effect of β on the latent space, Table S9 summarizes all experiments. The goal is to demonstrate that β is the primary factor driving the KL divergence towards zero regardless of the DAGMA hyperparameter settings, indicating posterior collapse.

The consolidated table makes the trend clear. For a given β value (e.g., 0.4), the KL Divergence remains consistently low (≈ 0.004) across all variations of μ and λ_1. However, changing β from 0.1 to 0.4, and then to 0.8 and higher, causes the KL Divergence to drop by orders of magnitude. This demonstrates that β is the determining factor for the degree of regularization and subsequent posterior collapse. The DAGMA weights have a negligible effect on the KL Divergence itself, reinforcing the conclusion that an appropriate β must be selected first before fine-tuning the other parameters.

Our argument is to choose $\beta = 0.4$ (approx. 0.447 in our final model) to balance the trade-off between distribution matching (MMD) and preserving biological heterogeneity. The KL Divergence at this range ($\approx 1 \times 10^{-3}$) is chosen to be similar to baselines like dVAE and SENA, ensuring the model captures meaningful biological variability rather than regressing to the population mean. While $\beta = 0.1$ preserves more variance, it suffers from poor stability and high MMD. Conversely, $\beta = 0.8$ achieves competitive MMD but suffers from an order-of-magnitude drop in prediction variance (1×10^{-4} vs 1×10^{-3}), indicating severe over-smoothing. We select $\beta = 0.4$ as the optimal operating point, minimizing MMD while retaining significantly higher variance than the fully collapsed regimes. The bolded row in Table S9 represents this chosen configuration.

Table S9: Impact of β-VAE and DAGMA Regularization on Model Performance. **Pred. Var.** denotes the variance of the predicted cell states (or variance of predicted control cells), where a lower value may indicate mean collapse. Lower MMD indicates better distribution matching. The bold row indicates the selected best configuration.

β	μ	λ_1	**KL Divergence**	**Pred. Var.**	**MMD**
0.0	0.27	0.05	0.0181	0.0181	1.929
0.1	0.1	0.02	0.059	0.0059	0.268
0.1	0.1	0.05	0.058	0.0067	0.257
0.1	0.1	0.1	0.054	0.0073	0.161
0.1	0.27	0.02	0.061	0.0051	0.401
0.1	0.27	0.05	0.060	0.0062	0.318
0.1	0.27	0.1	0.0615	0.0068	0.209
0.1	0.5	0.02	0.0609	0.0048	0.510
0.1	0.5	0.05	0.0604	0.0054	0.450
0.1	0.5	0.1	0.0572	0.0058	0.319
0.4	0.1	0.02	0.0040	0.0010	0.120
0.4	0.1	0.05	0.0039	0.0010	0.117
0.4	0.1	0.1	0.0040	0.0009	0.117
0.4	0.27	0.02	0.0040	0.0009	0.133
0.4	**0.27**	**0.05**	**0.0039**	**0.0009**	**0.127**
0.4	0.27	0.1	0.0040	0.0008	0.121
0.4	0.5	0.02	0.0040	0.0007	0.203
0.4	0.5	0.05	0.0040	0.0007	0.196
0.4	0.5	0.1	0.0040	0.0007	0.209
0.8	0.1	0.02	0.0009	0.0002	0.136
0.8	0.1	0.05	0.0009	0.0002	0.132
0.8	0.1	0.1	0.0009	0.0002	0.127
0.8	0.27	0.02	0.0009	0.0002	0.147
0.8	0.27	0.05	0.0009	0.0002	0.146
0.8	0.27	0.1	0.0009	0.0002	0.143
0.8	0.5	0.02	0.0008	0.0001	0.215
0.8	0.5	0.05	0.0009	0.0001	0.227
0.8	0.5	0.1	0.0009	0.0001	0.206
1.2	0.27	0.05	0.0003	0.00005	0.137
1.6	0.27	0.05	0.0002	0.00002	0.144
2.0	0.27	0.05	0.0001	0.00001	0.155
2.5	0.27	0.05	0.0001	0.000007	0.171
3.0	0.27	0.05	0.0000	0.000004	0.167

I LARGE LANGUAGE MODEL (LLM) USAGE DISCLOSURE

We utilized Google Gemini 2.5 Pro to assist with proofreading and summarizing related work. All content was verified by the authors.