

From Simulation to Inference: A Deep Learning Framework for the Inverse Problem in Phenotypic Discovery

Anrui Wang
wangar2023@shanghaitech.edu.cn
2023533015

Jiawen Dai
2023533132

Yetong Teng
2025212157

Yanzhi Yin
2025213248

Abstract—The inference of latent causal factors from high-dimensional observational data constitutes a fundamental inverse problem in phenotypic drug discovery. While generative “Virtual Cell” models have demonstrated fidelity in forward simulation, current benchmarks reveal that optimizing for reconstruction metrics (e.g., MSE) does not inherently capture the causal structure necessary for mechanistic insight. To address this, we reframe the modeling objective from forward generation to *in silico* reverse perturbation retrieval, transforming the Virtual Cell from a descriptive simulator into a prescriptive inference engine. We propose a deep learning framework that utilizes control-matched delta embeddings and pairwise ranking objectives to map observed disease phenotypes back to their perturbation sources. Our evaluation uncovers a governing Geometry vs. Semantics trade-off in the representation space: while linear geometric baselines excel at retrieving targets with dominant, high-magnitude signals, pre-trained foundation models uniquely capture the semantic dependencies required to identify complex, low-signal targets that lie beyond simple Euclidean proximity. This establishes a new paradigm where the utility of biological foundation models is defined not by their generative fidelity, but by their capacity to close the causal loop and support rigorous hypothesis generation for target deconvolution.

I. INTRODUCTION

The ambition to construct “Digital Twins” of cellular biology—computational systems capable of simulating complex life processes—has increasingly converged with advances in generative deep learning. In the domain of phenotypic drug discovery, this manifests as a fundamental inverse problem: identifying the latent molecular drivers (causes) that generate observed high-dimensional transcriptomic states (effects). Historically, phenotypic screening has succeeded by observing therapeutic rescues in a target-agnostic context, yet this empirical success masks a computational bottleneck. While the downstream effect—the phenotypic state vector—is observable, the upstream mechanism of action remains a latent variable, necessitating a laborious deconvolution process to solve for the causal perturbation.

High-throughput transcriptomic profiling offers a bridge across this inference gap, providing information-rich “fingerprints” that allow us to define biological phenotypes as computable vectors. Theoretically, this transforms the biological question into a data-driven manifold learning task: if we can quantify the shift in cellular state space, can we retrieve the perturbation embedding that induced it? Recent efforts have

attempted to address this by training “Virtual Cells”—high-fidelity simulators learned from large-scale data [1]. These systems aim to approximate the function mapping a cell state and a perturbation to a future state, effectively serving as engines for *in silico* experimentation [2]. By simulating the effects of varying interventions, such models hold the potential to drastically reduce the hypothesis space for target deconvolution, positioning the Virtual Cell not merely as a simulator, but as a computational partner for causal inference.

Despite this promise, the field faces a critical asymmetry in objective optimization. Immense computational effort is currently directed toward optimizing forward prediction metrics, yet empirical gains remain marginal. Systematic benchmarks reveal that complex deep neural networks do not consistently outperform simpler linear alternatives, with performance often confounded by statistical noise rather than driven by architectural superiority [3]. More rigorous evaluations suggest that the singular pursuit of minimizing reconstruction error (MSE) may be insufficient for capturing the causal structure required for discovery, as state-of-the-art models frequently fail to surpass trivial mean-prediction baselines [4].

To bridge the gap between generative fidelity and inference utility, we propose a fundamental reframing of the Virtual Cell’s objective: shifting from forward simulation (generating the expression profile of a known perturbation) to reverse retrieval (inferring the causal perturbation of an observed profile). We operationalize this paradigm shift by treating transcriptomic phenotypes as queries in a retrieval and ranking task, a formulation that aligns the model’s output with the decision-making logic of therapeutic discovery. In this work, we present a unified framework that validates the semantic richness of single-cell embeddings through two complementary phases. First, we establish a “credibility scaffold” by rigorously evaluating forward perturbation prediction on the Virtual Cell Challenge (VCC) dataset [1], prioritizing biologically grounded metrics (DES, PDS) to ensure models capture directional causal signals. Second, we invert the verified backbone to perform *in silico* reverse perturbation prediction, benchmarking foundation models against geometric baselines. Our investigation uncovers a critical “Geometry vs. Semantics” trade-off: while heuristic baselines like PCA-KNN excel at retrieving signals that are geometrically reachable in Eu-

clidean space, pre-trained Transformer models such as scGPT uniquely capture the semantic context required to identify complex, low-signal targets that lie beyond simple similarity.

AUTHOR CONTRIBUTIONS

- **Anrui Wang:** Implemented scGPT experiments for both forward and reverse tasks, adapted the architecture to leverage improved VCC evaluation metrics, and developed scGPT-based models specifically for target gene identification to assess biologically meaningful predictions.
- **Jiawen Dai:** Implemented STATE experiments to fulfill VCC requirements, designed and implemented a biologically informed set of evaluation metrics, and curated relevant data from the Tahoe-100M dataset.
- **Yetong Teng:** Conducted benchmark studies on baseline models for the VCC forward prediction task and evaluated comparative model performance.
- **Yanzhi Yin:** Developed and optimized baseline models for target gene prediction on both the Norman and Tahoe-100M datasets.

II. RELATED WORK

The development of AI Virtual Cells (AIVCs) sits at the intersection of generative modeling, representation learning, and high-dimensional biology. Early conceptualizations of the Virtual Cell emphasized the capacity to learn fundamental relationships between cell state and function from massive, unlabelled datasets [1]. Bunne et al. [2] expanded this vision, articulating a framework where AIVCs function as differentiable simulators capable of predicting the consequences of perturbations—such as gene knockdowns or drug treatments—across diverse cellular contexts. These foundational works established the forward simulation task as the primary training objective, utilizing reconstruction losses to force models to internalize cellular dynamics.

However, the efficacy of deep learning architectures in this domain has recently faced scrutiny. The challenge of modeling single-cell transcriptomics lies in the sparsity and high dimensionality of the feature space, where true biological signals are often overwhelmed by technical noise. Li et al. [3] demonstrated through systematic benchmarking that complex deep learning models often struggle to outperform simpler machine learning alternatives when evaluated on standard reconstruction metrics. Reinforcing this observation, Ahlmann-Eltze [4] provided evidence that state-of-the-art perturbation models do not yet consistently surpass linear baselines, suggesting that architectures optimized solely for minimizing Mean Squared Error (MSE) may prioritize global statistical fit over the capture of subtle, causal regulatory mechanisms.

Response to these limitations has driven the exploration of specialized architectures and alternative problem formulations. Geneformer [5] pioneered the application of transfer learning to this space, leveraging attention mechanisms to prioritize candidate therapeutic targets beyond mere expression forecasting. To address the combinatorial complexity of biolog-

ical systems, models like GEARS [6] integrated biological inductive biases—specifically gene regulatory graphs—to improve generalization to unseen perturbation sets. Similarly, the Transformer-based STATE model [7] introduced cell-set distribution modeling to enable mechanistic generalization across novel cellular contexts. Building on these architectural advances, scGPT [8] explicitly formalized the reverse inference challenge as a top-K retrieval task, demonstrating that foundation models can learn to rank potential drivers based on their alignment with a target cell state. Our work extends this trajectory, moving beyond architectural design to rigorously evaluate the trade-offs between geometric interpolation and semantic reasoning in the context of the inverse problem.

III. METHODOLOGY

To construct a Virtual Cell capable of both high-fidelity simulation and causal reasoning, we propose a two-stage deep learning framework. First, we formalize the Forward Problem to validate that the model’s latent space captures true biological perturbation signals. Second, we freeze this validated backbone and address the Inverse Problem via a specialized retrieval architecture and ranking objective.

A. Phase I: The Forward Problem (Generative Simulation)

1) *Problem Formulation:* The core task of Transcriptome-wide Perturbation Prediction serves as the generative phase: predicting the transcriptomic consequence of a defined intervention. Formally, we aim to learn a function f :

$$\text{given } (x, p) \implies \hat{y} = f(x, p)$$

where x denotes the pre-perturbation control state (baseline distribution), p represents the known perturbation embedding (e.g., gene knockout or drug), and \hat{y} is the predicted post-perturbation state vector optimized to minimize divergence from the observed profile y .

2) *Evaluation Metrics as Optimization Proxies:* To ensure the model learns causal mechanisms rather than statistical noise, we utilize specific metrics that define our optimization landscape:

- **Perturbation Discrimination Score (PDS):** To enforce directional fidelity, PDS computes the cosine similarity between the predicted expression shift ($\hat{\delta}_k$) and the true shift (δ_k). This rewards models that align the angle of the regulation vector, mitigating penalties for magnitude attenuation common in single-cell data.
- **Differential Expression Score (DES):** This metric validates the biological sparsity of the prediction by measuring the overlap (precision and recall) between the predicted and ground-truth sets of differentially expressed genes.

3) *Deep Generative Architectures:* We benchmark two distinct deep learning paradigms for this task:

1. **Cell-Set Modeling (STATE).** We implement the STATE model’s State Transition (ST) module. As illustrated in Figure 1, this architecture utilizes a Transformer backbone that

ingests control expression profiles, ESM2-based protein embeddings for perturbations, and batch embeddings. By projecting these inputs into a shared hidden space, the model learns to shift cellular state distributions in response to perturbation tokens.

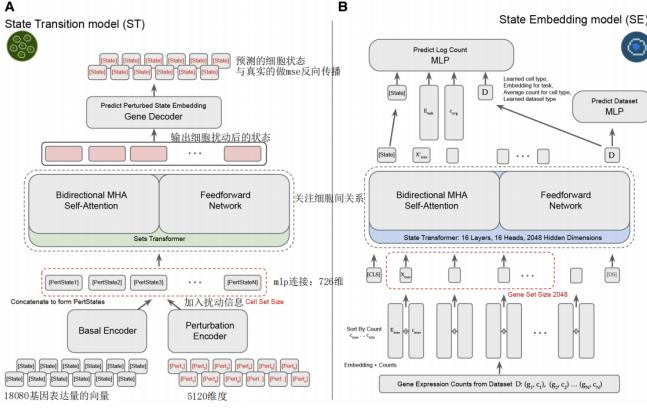


Fig. 1. The architecture of the STATE model. The State Transition (ST) module integrates control expression, perturbation embeddings, and batch information to predict post-perturbation transcriptomic states.

2. Foundation Model Adaptation (scGPT). We employ a transfer learning strategy using the scGPT foundation model (Figure 2). To prevent catastrophic forgetting, we adopt a partial freezing strategy: the core gene token encoders and transformer layers are frozen, preserving pre-trained gene-gene interactions. We fine-tune only the perturbation encoder and the affine expression decoder.

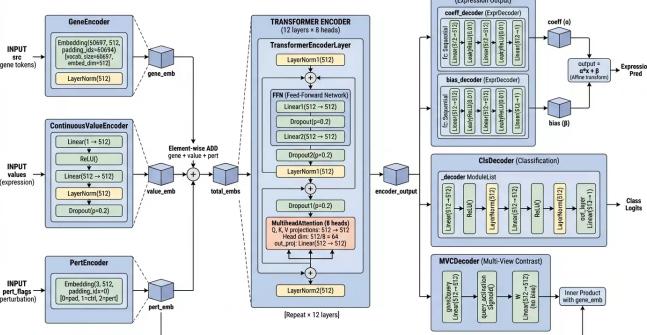


Fig. 2. Overview of the scGPT model adaptation for perturbation prediction. The model is initialized with pre-trained weights, with core modules frozen to retain biological knowledge, while the perturbation encoder and decoders are fine-tuned.

Crucially, we optimize a Composite Loss Function to enforce biological realism:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{sw1}} \mathcal{L}_{\text{sw1}} + \lambda_{\text{proto}} \mathcal{L}_{\text{proto}} + \lambda_{\text{rank}} \mathcal{L}_{\text{de_rank}} + \lambda_{\text{dir}} \mathcal{L}_{\text{dir}}$$

This objective combines Sliced Wasserstein distance for distribution alignment, ProtoInfoNCE for contrastive learning of perturbation effects, and auxiliary ranking losses to enforce correct gene regulation directionality.

B. Phase II: The Inverse Problem (Reverse Retrieval)

1) Problem Formulation: The actionable goal in therapeutic discovery is the inverse: given an observed disease phenotype y , identify the causal perturbation p . We operationalize this as a retrieval task:

$$\hat{p} = g(x, y) = \arg \min_p d(f(x, p), y)$$

We reject the "Forward Model-Based Retrieval" (Route A) approach, which requires exhaustive simulation of all combinatorial possibilities, as it suffers from prohibitive computational explosion. Instead, we pursue Route B: Direct Inverse Mapping, learning a parameterized function g_θ that directly maps (control, disease) pairs to ranked perturbation candidates.

2) Geometric vs. Semantic Baselines: To benchmark the necessity of deep learning, we compare against geometric baselines. We implement PCA+kNN and Random Forest regressors on pseudobulk profiles. These models rely on Euclidean or Cosine distance in the raw or linearly projected feature space, effectively retrieving perturbations based on "Geometric Reachability"—i.e., phenotypes with strong, high-magnitude signatures.

3) Deep Discriminative Architecture: scGPT Reverse: To capture "Semantic Reachability"—subtle, non-linear signals lost in pseudobulk aggregation—we propose a single-cell discriminative fine-tuning strategy for scGPT.

A key innovation in our approach is the isolation of the perturbation signal from cell-specific noise (e.g., cell cycle). As shown in Figure 3, we compute a delta embedding \mathbf{h}_i for each cell by subtracting a reference embedding derived from matched control cells:

$$\mathbf{h}_i = \mathbf{e}_i - \mathbf{e}_{\text{ref}}$$

This difference vector \mathbf{h}_i is then projected via a gene-scoring head to compute similarity scores s_{ij} against all candidate gene embeddings \mathbf{g}_j .

Pairwise Ranking Objective. To optimize the retrieval ranking directly, we employ a composite loss $\mathcal{L} = \lambda_{\text{rank}} \mathcal{L}_{\text{rank}} + \lambda_{\text{bce}} \mathcal{L}_{\text{bce}}$. The core component is the Pairwise Ranking Loss:

$$\mathcal{L}_{\text{rank}} = \mathbb{E}_{p,n} [\max(0, m - (s_{i,p} - s_{i,n}))]$$

This margin-based loss forces the model to score the true causal gene p higher than mined negative samples n , explicitly optimizing the metric of interest (Rank) rather than reconstruction error.

IV. EXPERIMENTS

We structured our evaluation in two distinct phases to mirror the "Simulation-to-Inference" framework. First, we benchmarked the forward simulation capabilities to validate that our deep learning backbones capture true biological signals rather than statistical noise. Second, we froze these validated encoders to assess their performance on the downstream, actionable task of reverse perturbation retrieval.

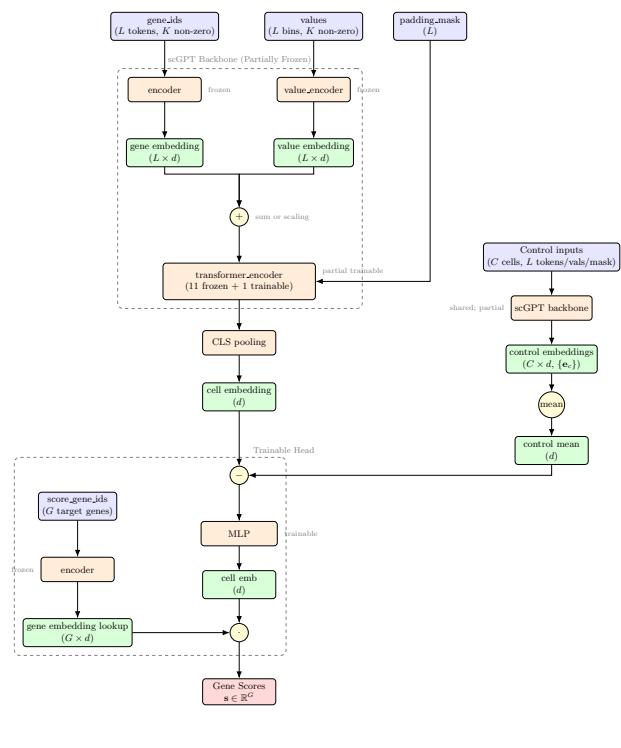


Fig. 3. Architecture of the scGPT-based reverse perturbation prediction model. The model processes perturbed cells through the scGPT encoder to generate cell embeddings. A control-matched delta embedding strategy is employed, where reference control embeddings are subtracted from perturbed embeddings to isolate perturbation-specific signals. The resulting delta embedding is then projected through a gene-scoring head that computes similarity scores against all candidate gene embeddings to rank potential perturbation targets.

A. Phase I: Forward Validation (VCC Dataset)

1) *Experimental Setup:* To evaluate the generative fidelity of the Virtual Cell, we utilized the ARC Institute VCC competition dataset, comprised of single-cell RNA-seq data for H1 human embryonic stem cells under 150 distinct genetic perturbations [1].

Generalization Protocol: A critical flaw in standard benchmarks is the use of random cell splits, which reduces the task to cellular interpolation. To rigorously test mechanistic generalization, we enforced a Perturbation-Level Split. We re-partitioned the dataset such that 20% of perturbation genes (30 genes) were held out entirely as a test set. This ensures that the model must predict the transcriptomic effects of genetic interventions it has *never seen during training*, forcing it to rely on learned gene-gene interaction dynamics rather than memorization.

2) *Results and Analysis:* We compared the deep learning models (STATE and fine-tuned scGPT) against linear (Ridge Regression) and non-linear (Random Forest) baselines. The quantitative results are presented in Table I.

TABLE I
PERFORMANCE COMPARISON OF FORWARD PREDICTION MODELS ON VCC. DEEP MODELS (STATE, SCGPT) OUTPERFORM BASELINES ON BIOLOGICAL METRICS (DES, PDS) DESPITE HIGHER RECONSTRUCTION ERROR (MAE).

Model	PDS	MAE	DES	Overall
Cell-mean baseline	0.5167	0.1258	0.1075	0.00
Ridge Regression	0.5167	0.1253	0.1466	1.59
Random Forest	0.5167	0.1253	0.1360	1.19
STATE Model	0.5367	0.1286	0.3023	7.28
scGPT Finetune	0.5089	0.2763	0.2620	6.27

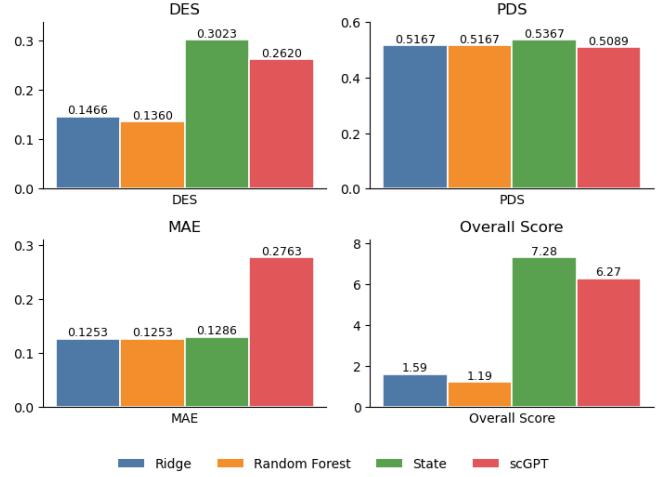


Fig. 4. Evaluation Metrics and Scoring Framework. The Virtual Cell Challenge utilizes a composite scoring system derived from three complementary metrics to assess forward prediction performance. (A) Differential Expression Score (DES): Measures the model's fidelity in recovering biologically interpretable signals by calculating the overlap (precision and recall) between predicted and ground-truth differentially expressed genes (DEGs). (B) Perturbation Discrimination Score (PDS): A ranking-based metric that evaluates whether a predicted pseudo-bulk profile is more similar to its corresponding ground truth than to other perturbations. To mitigate biases toward vectors with small norms, PDS utilizes cosine similarity to emphasize directional alignment over magnitude. (C) Mean Absolute Error (Top-2k MAE): Captures the global reconstruction accuracy by computing the element-wise error across the top 2,000 highly variable genes. (D) Overall Score: A weighted aggregation of the individual metrics used to rank model performance, enforcing a balance between global fit (MAE), directional correctness (PDS), and biological specificity (DES).

a) *Metric Analysis and Failure Modes:* The selection of these metrics addresses specific failure modes in predictive modeling (Figure 4D). While MAE provides a necessary baseline for global statistical fit, it remains largely insensitive to subtle, biologically meaningful shifts. Consequently, DES serves as the strictest test of biological utility, ensuring the model identifies the precise gene sets driving the phenotype. Furthermore, the adaptation of PDS addresses the "attenuation" phenomenon, where models correctly predict the direction of regulation but with damped magnitude compared to ground truth. By prioritizing vector direction (cosine similarity) over Euclidean distance, we reward models that capture the correct regulatory mechanism even if the signal strength is conservative.

To further investigate these failure modes, we selected

MED13 from the test set for detailed analysis, as it exhibited relatively high performance scores (DES, PDS, MAE) compared to other genes. However, visual inspection reveals a critical limitation (Figure 5). While the ground truth demonstrates a significant downregulation of MED13 expression following its knockout, the STATE model predicts a distribution centered around zero change ($\Delta \approx 0$), resembling a normal distribution. We hypothesize that STATE tends to predict a conservative normal distribution around zero for unseen perturbations. This behavior likely stems from the model being trained solely on H1 cell line data with a limited set of gene perturbations. Lacking exposure to diverse cell types or combinatorial perturbations, the model adopts a conservative strategy on unknown perturbations, predicting near-control states. Furthermore, the STATE architecture does not explicitly enforce a zero-expression constraint for the target gene, further contributing to this “attenuation” artifact.

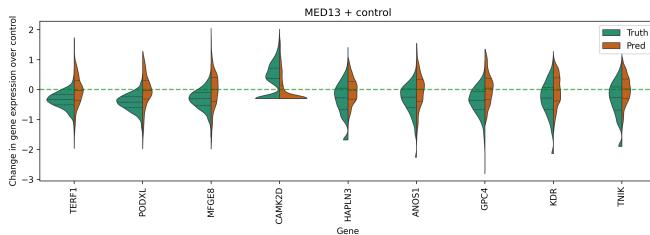


Fig. 5. Analysis of Prediction Failure on MED13. Despite achieving high global metrics, the STATE model fails to predict the specific downregulation of the target gene MED13. The predicted expression change follows a normal distribution centered at zero (conservative prediction), contrasting with the true downregulation observed in the ground truth. This highlights the model’s tendency to revert to a mean-zero shift when encountering unseen perturbations, a phenomenon not fully captured by aggregate metrics like MAE.

B. Phase II: Reverse Perturbation Retrieval (Norman Dataset)

Having validated that the embeddings capture causal directionality, we proceeded to the primary task: Target Gene Identification.

1) *Experimental Setup:* We utilized the Norman et al. Perturb-seq dataset, comprising 91,205 cells across 236 unique perturbation conditions, to benchmark performance in a realistic drug discovery context where causal mechanisms remain obscure.

To rigorously evaluate generalization, we implemented a Compositional Split strategy that partitions the data based on perturbation complexity rather than simple random sampling. Within this framework, 15% of single-gene perturbations were strictly held out to assess the model’s capacity to identify novel drivers. Extending this logic to combinatorial interactions, we stratified double-gene perturbations into three distinct tiers: “Seen2,” encompassing pairs where both constituents were observed separately during training; “Seen1,” where only one gene was known; and “Seen0,” representing completely novel combinations. This tiered approach challenges the model to deconstruct complex, unseen phenotypes into their constituent

genetic drivers, moving beyond retrieval to true mechanistic inference.

2) *Results: The “Geometry vs. Semantics” Trade-off:* We benchmarked the scGPT retrieval framework against pseudobulk geometric baselines (PCA+kNN, Random Forest). The results, summarized in Table II, reveal a striking performance crossover that illuminates the distinct utility of deep learning in this domain.

TABLE II
PERFORMANCE COMPARISON OF SCGPT AND BASELINE MODELS ON THE NORMAN DATASET FOR REVERSE PERTURBATION PREDICTION. BEST RESULTS ARE BOLDED.

Metric	scGPT	PCA+kNN	Random Forest	XGBoost
MRR	0.1975	0.3602	0.3232	0.3323
Exact Hit@1	0.0000	0.0000	0.0000	0.0000
Relevant Hit@1	0.1039	0.3158	0.2105	0.2368
Recall@1	0.0520	0.1579	0.1053	0.1184
Exact Hit@5	0.0526	0.1053	0.0789	0.0789
Relevant Hit@5	0.2922	0.4211	0.4737	0.4211
Recall@5	0.1631	0.2632	0.2763	0.2500
Exact Hit@10	0.0963	0.1053	0.1316	0.1053
Relevant Hit@10	0.4155	0.4211	0.5000	0.5000
Recall@10	0.2492	0.2632	0.3158	0.3026
Exact Hit@20	0.2123	0.1053	0.1842	0.1842
Relevant Hit@20	0.5731	0.4211	0.5000	0.5526
Recall@20	0.3927	0.2632	0.3421	0.3684
Exact Hit@40	0.3476	0.1053	0.1842	0.1842
Relevant Hit@40	0.6828	0.4211	0.5263	0.5526
Recall@40	0.5152	0.2632	0.3553	0.3684

Phenomenon: The Cross-over Effect. Figure 6 illustrates the “Geometry vs. Semantics” trade-off observed in the target identification task. The bars represent Relevant Hit@K (the probability of retrieving at least one correct driver), while the dashed lines track Recall@K (the fraction of ground-truth targets recovered). At strictly local retrieval windows (K=1 to K=10), shallow baselines such as PCA-KNN (light blue) and Random Forest (red) outperform scGPT, validating their efficiency in identifying “geometrically reachable” targets with strong phenotypic signatures. However, a decisive performance crossover occurs as the window expands: at K=40, scGPT (dark blue) surpasses all baselines, achieving a Recall of approximately 0.52 compared to 0.26 for PCA-KNN. This trend demonstrates that while baselines excel at high-precision “head” retrieval, scGPT possesses the “semantic reachability” necessary to retrieve the “long tail” of complex or subtle perturbation signals that simpler models miss.

Explanation: Geometric vs. Semantic Reachability. To interpret this divergence, we introduce two distinct modes of perturbation retrieval. Geometric Reachability characterizes perturbations with strong, high-variance phenotypic signatures. In these cases, the perturbed state lies structurally close to the target definition in the raw gene expression space. Simple geometric neighbors (KNN) are sufficient to lock onto these targets. Conversely, Semantic Reachability characterizes perturbations with weak signals, combinatorial effects, or context-dependent regulation. Here, the raw expression distance is noisy or misleading. Retrieving these targets requires “semantic” inference—leveraging learned gene regulatory networks

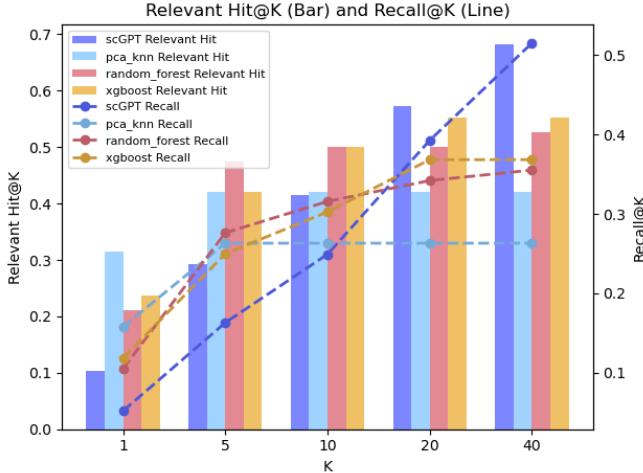


Fig. 6. The “Cross-over” Effect in Reverse Perturbation Retrieval on the Norman Dataset. Comparison of foundation model (scGPT) performance against geometric (PCA-KNN) and ensemble (Random Forest, XGBoost) baselines across varying retrieval window sizes (K).

and context embeddings to map a subtle phenotype back to its driver, a capability unique to deep generative models.

V. FROM GENES TO DRUGS: THE LIMITS OF UNIMODAL INVERSION ON TAHOE

A. The Vanishing Signal: Why Drug Deconvolution is an Ill-Posed Inverse Problem

While the reverse retrieval of genetic perturbations (Section III) proved successful, extending this framework to pharmacological perturbations (drugs) reveals a fundamental escalation in the complexity of the inverse function. In Deep Learning terms, CRISPR-mediated interventions represent a shallow causal graph: the perturbation (gene knockout) acts directly on the transcriptomic state definition. The mapping $f : \text{Perturbation} \rightarrow \text{Expression}$ is relatively direct and invertible.

In contrast, small-molecule drugs operate on a deep causal graph with a hidden layer: the proteome. Drugs bind to proteins, not genes. The transcriptomic signature observed in scRNA-seq (Y) is merely a downstream echo of the initial drug-protein binding event (X), mediated by complex signaling cascades (Z).

$$X_{\text{drug}} \xrightarrow{\text{Binding}} Z_{\text{protein}} \xrightarrow{\text{Signaling}} Y_{\text{mRNA}}$$

This creates a significant Information Bottleneck. By the time the signal reaches the observable transcriptomic layer, it is attenuated by temporal delays and confounded by off-target toxicity (noise). Consequently, the inverse mapping $P(X|Y)$ becomes ill-posed: multiple different drug inputs (X) can collapse into indistinguishable transcriptomic outputs (Y), making unique retrieval mathematically impossible without additional constraints. The assumption that a drug’s target can be retrieved simply as the “most altered feature” in the output vector is biologically naive and computationally insufficient.

B. Benchmarking on Noisy Labels: The Tahoe-100M Pilot

To empirically map the boundaries of our retrieval architecture, we conducted an exploratory benchmark using the Tahoe-100M dataset. This task represents a classic case of Learning with Noisy Labels. The “ground truth” targets provided in public databases are often binary and context-independent, failing to account for whether the target protein is actually expressed or functional in the specific cell line assayed.

We curated a high-confidence subset of 52 drugs with verifiable targets to serve as a test bed. Our analysis reveals a stark reality: standard expression-based baselines (Differential Expression ranking and Logistic Regression) performed close to random guessing.

We posit that this performance plateau is not a failure of model capacity, but a Data Limitation inherent to the unimodal formulation. The observed feature space (scRNA-seq) likely does not contain the *sufficient statistics* required to distinguish between the targets. The model is effectively being asked to classify inputs that are conditionally independent of the labels given the missing latent variable (protein activity). Therefore, we frame the current state of Tahoe-100M not as a failure of the architecture, but as a proof-of-concept that large-scale perturbation atlases currently lack the label density required to train precise reverse-engineering models.

C. Architectural Implications: Towards Multimodality and Graph Priors

The failure of unimodal retrieval on pharmacological data implies that future architectures must transcend simple transcriptomic inversion to bridge the “Inference Gap.” Central to this evolution is Latent Variable Modeling with Priors, where the unobserved protein layer is treated as a latent variable Z rather than a hidden black box. By integrating Graph Neural Networks (GNNs) grounded in Protein-Protein Interaction (PPI) networks, models can exploit structural inductive biases that constrain the search space, ensuring that potential targets are topologically “reachable” from observed expression changes. Beyond topological constraints, the information loss inherent in the $X \rightarrow Y$ transition necessitates Multimodal Contrastive Learning; effectively recovering causal signals requires fusing orthogonal data streams—such as proteomics or morphological profiles—to triangulate drivers and resolve the ambiguity plaguing unimodal RNA-seq. Because pharmacological agents inherently exhibit promiscuity, accurate modeling further demands a shift toward Soft-Label Training Objectives that reframe the challenge as Multi-Label Learning. Consequently, evaluation metrics must expand from strict “Hit@1” precision to capture broader “Hit@Pathway” or “Hit@Mechanism” standards, reflecting the reality that a drug’s footprint typically manifests as the modulation of an entire functional module rather than a single isolated gene.

VI. DISCUSSION

A. The Generative-Inverse Duality in Foundation Models

We argue that the defining challenge of the “Virtual Cell” is not merely the construction of a high-fidelity generative

simulator, but the effective inversion of that simulator to solve ill-posed inference problems. While the current zeitgeist in biological deep learning prioritizes the forward mapping $P(Y|X)$ —predicting cellular states from perturbations—our work demonstrates that the ultimate utility of these architectures lies in their invertibility, or $P(X|Y)$. By reframing the objective from forward simulation to Reverse Perturbation Retrieval, we transform the model from a descriptive generator into a prescriptive inference engine. This shift mirrors the evolution seen in computer vision, where the focus has expanded from pure image generation (GANs/Diffusion) to the semantic interpretation and inversion of latent representations for discovery.

B. The Metric Gap: Reconstruction vs. Semantic Retrieval

Our findings expose a critical misalignment between standard model optimization objectives and downstream utility. The community’s reliance on reconstruction metrics, such as Mean Squared Error (MSE), implicitly assumes that minimizing the Euclidean distance in gene expression space equates to learning the causal structure of the system. However, our benchmarks reveal that MSE often prioritizes the preservation of global, high-variance statistical features (noise) over the subtle, low-variance signals that define causality.

This observation resolves the apparent paradox where complex Deep Learning models fail to outperform linear baselines in forecasting tasks. We posit that this “Linear vs. Deep” debate is a byproduct of the evaluation landscape. When the objective is shifted to Reverse Retrieval—a task requiring the disentanglement of non-linear manifolds—the “Geometry vs. Semantics” trade-off becomes clear. Linear models (PCA) rely on geometric reachability and excel at interpolation within the convex hull of training data. In contrast, Transformer-based foundation models (scGPT) capture “semantic reachability,” leveraging learned attention mechanisms to retrieve targets that are geometrically distant but mechanistically linked. Thus, the value of Deep Learning lies not in curve fitting, but in its capacity for semantic inference where simple geometric heuristics fail.

C. Toward Hybrid Neuro-Symbolic Architectures

The “Cross-over Effect” observed in our retrieval experiments—where linear models dominate precision at low K while Transformers dominate recall at high K —suggests that the optimal architecture is not monolithic. Instead, we propose a Functional Decomposition of the inference task. Future systems should likely adopt a hybrid design: a “Wide and Deep” approach where geometric modules handle the high-precision screening of dominant, linear effects, while deep generative backbones are reserved for deciphering the “long tail” of complex, non-linear regulatory landscapes.

To support this semantic reasoning, mere data scale is insufficient. Future architectures must integrate strong Inductive Biases into the Transformer backbone. As demonstrated by GEARS and Geneformer, incorporating prior knowledge structures—such as gene regulatory graphs or cell-state transition

probabilities—is essential for constraining the search space and enabling zero-shot generalization to unseen combinatorial perturbations.

D. Limitations: The Information Bottleneck

Validating the invertibility of single-cell models establishes a proof-of-concept, yet scaling this to industrial application faces the “Information Bottleneck” of the underlying causal graph. The most formidable challenge is the Inference Gap in pharmacological perturbations. Unlike genetic interventions (shallow causal depth), drugs operate through a deep, unobserved protein layer. The transcriptomic signal observed in scRNA-seq is a downstream compression of this event, often rendering the inverse mapping ill-posed due to information loss and polypharmacology.

Addressing this requires transitioning from unimodal learning to Multimodal Fusion. Future benchmarks must integrate orthogonal data streams (e.g., proteomics) to triangulate the latent causal variables, effectively adding constraints to the inverse problem. Furthermore, our analysis of the Tahoe-100M dataset highlights the peril of Learning with Noisy Labels. The low performance of all models on this dataset suggests that the signal-to-noise ratio in current large-scale atlases is insufficient for precise supervision. Finally, the discrepancy between scGPT’s high Recall and moderate Mean Reciprocal Rank (MRR) indicates a need for specialized Ranking Loss Functions. Future work must focus on “Reranking” objectives—potentially using contrastive margins or list-wise optimization—to refine broad semantic retrieval into precise top- k predictions.

REFERENCES

- [1] Yusuf H. Roohani, Tony J. Hua, Po-Yuan Tung, Lexi R. Bounds, Feiqiao B. Yu, Alexander Dobin, Noam Teyssier, Abhinav Adduri, Alden Woodrow, Brian S. Plosky, Reshma Mehta, Benjamin Hsu, Jeremy Sullivan, Chiara Ricci-Tam, Nianzhen Li, Julia Kazaks, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Hani Goodarzi, and Dave P. Burke. Virtual Cell Challenge: Toward a Turing test for the virtual cell. *Cell*, 188(13):3370–3374, June 2025.
- [2] Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B. Burkhardt, Andrea Califano, Jonah Cool, Abby F. Dernburg, Kirsty Ewing, Emily B. Fox, Matthias Haury, Amy E. Herr, Eric Horvitz, Patrick D. Hsu, Viren Jain, Gregory R. Johnson, Thomas Kalil, David R. Kelley, Shana O. Kelley, Anna Kreshuk, Tim Mitchison, Stephani Otte, Jay Shendure, Nicholas J. Sofroniew, Fabian Theis, Christina V. Theodoris, Srivikul Upadhyayula, Marc Valer, Bo Wang, Eric Xing, Serena Yeung-Levy, Marinka Zitnik, Theofanis Karaletsos, Aviv Regev, Emma Lundberg, Jure Leskovec, and Stephen R. Quake. How to Build the Virtual Cell with Artificial Intelligence: Priorities and Opportunities, October 2024. arXiv:2409.11654 [q-bio].
- [3] Lanxiang Li, Yue You, Yunlin Fu, Wenyu Liao, Xueying Fan, Shihong Lu, Ye Cao, Bo Li, Wenle Ren, Jiaming Kong, Shuangjia Zheng, Jizheng Chen, Xiaodong Liu, and Luyi Tian. A Systematic Comparison of Single-Cell Perturbation Response Prediction Models, December 2024.
- [4] Constantin Ahlmann-Eltze. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines.
- [5] Christina V. Theodoris, Ling Xiao, Anant Chopra, Mark D. Chaffin, Zeina R. Al Sayed, Matthew C. Hill, Helene Mantineo, Elizabeth M. Brydon, Zexian Zeng, X. Shirley Liu, and Patrick T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, June 2023.
- [6] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, June 2024.
- [7] Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, Chiara Ricci-Tam, Christopher Carpenter, Vishvak Subramanyam, Aidan Winters, Sravya Tirukkovular, Jeremy Sullivan, Brian S. Plosky, Basak Eraslan, Nicholas D. Youngblut, Jure Leskovec, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Alexander Dobin, Dave P. Burke, Hani Goodarzi, and Yusuf H. Roohani. Predicting cellular responses to perturbation across diverse contexts with State.
- [8] 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, August 2024.

APPENDIX

A. Transcriptome-wide Perturbation Prediction

a) *Problem Formulation*.: The core task of the Transcriptome-wide Perturbation Prediction serves as the **Forward Problem**: predicting the transcriptomic consequence of a defined intervention. Formally:

$$\text{given } (x, p) \Rightarrow \hat{y} = f(x, p)$$

where:

- x denotes the *pre-perturbation cell state*, serving as the baseline. This is empirically represented by the distribution of control cells within the same experimental batch or cell type. For instance, in scGPT benchmarks on the Norman dataset, x is approximated by the mean expression vector of all control cells ($1 \times M$ genes).
- p represents the *known perturbation* (e.g., gene knockout, CRISPRi, or drug). This is typically encoded as a discrete gene identifier or a multidimensional perturbation embedding.
- \hat{y} is the *predicted post-perturbation state*. The model is optimized to minimize the divergence between \hat{y} and the experimentally observed expression profile y .

1) Evaluation metrics: PDS, MAE, DES:

a) *Perturbation Discrimination Score (PDS)*.: The Perturbation Discrimination Score (PDS) measures whether the predicted gene expression changes induced by different perturbations preserve the correct global directionality across all genes. For each perturbation $k \in \{1, \dots, N\}$, we first construct pseudobulk expressions by averaging all cells under the same perturbation, yielding the ground-truth pseudobulk $y_k \in \mathbb{R}^G$, the predicted pseudobulk $\hat{y}_k \in \mathbb{R}^G$, and the control (NTC) pseudobulk $y_{\text{ntc}} \in \mathbb{R}^G$. Perturbation-induced expression changes are then defined as $\delta_k = y_k - y_{\text{ntc}}$ and $\hat{\delta}_k = \hat{y}_k - y_{\text{ntc}}$. For each predicted perturbation $\hat{\delta}_k$, we compute cosine similarities to all true perturbation deltas $\{\delta_j\}_{j=1}^N$ as

$$S_{k,j} = \frac{\hat{\delta}_k \cdot \delta_j}{\|\hat{\delta}_k\|_2 \|\delta_j\|_2}.$$

These similarities are ranked, and the rank of the correct perturbation k is denoted by R_k . The final score is reported as the mean rank $\text{PDS}_{\text{rank}} = \frac{1}{N} \sum_{k=1}^N R_k$, or its normalized version $\text{nPDS}_{\text{rank}} = \frac{1}{N^2} \sum_{k=1}^N R_k$.

b) *MAE on Top 2000 Genes by Ground-Truth Fold Change*.: To focus evaluation on biologically meaningful signals, we additionally compute the mean absolute error (MAE) restricted to the genes with the largest true perturbation effects. For each perturbation k , we start from the raw (unnormalized) average counts $c_k \in \mathbb{R}^G$ and control counts $c_{\text{ntc}} \in \mathbb{R}^G$, and compute the absolute log₂ fold change for each gene g using a pseudocount of 1:

$$\text{LFC}_{k,g} = |\log_2(c_{k,g} + 1) - \log_2(c_{\text{ntc},g} + 1)|.$$

Genes are ranked by $\text{LFC}_{k,g}$ in descending order, and the top 2000 genes form the index set Ω_k . Using log1p-normalized

pseudobulk expressions y_k and \hat{y}_k (the same representation as in PDS), the MAE is computed only over these selected genes and then averaged across perturbations:

$$\text{MAE}_{\text{top2k}} = \frac{1}{N} \sum_{k=1}^N \left(\frac{1}{2000} \sum_{g \in \Omega_k} |\hat{y}_{k,g} - y_{k,g}| \right).$$

This metric emphasizes prediction accuracy on genes that exhibit the strongest true responses to each perturbation.

c) *Differential Expression Score (DES)*.: The Differential Expression Score (DES) measures how accurately a model recovers perturbation-induced differential gene expression. For each perturbation $k \in \{1, \dots, N\}$, we first perform differential expression analysis between perturbed and control cells for both the predicted and ground-truth data using the Wilcoxon rank-sum test with tie correction, and identify significantly differentially expressed (DE) genes by controlling the false discovery rate at level $\alpha = 0.05$ with the Benjamini-Hochberg procedure. This yields a predicted DE gene set $G_{k,\text{pred}}$ and a ground-truth DE gene set $G_{k,\text{true}}$, with sizes $n_{k,\text{pred}} = |G_{k,\text{pred}}|$ and $n_{k,\text{true}} = |G_{k,\text{true}}|$, respectively. If $n_{k,\text{pred}} \leq n_{k,\text{true}}$, the DES for perturbation k is defined as the size of the intersection between the predicted and true DE gene sets, normalized by the size of the true set,

$$\text{DES}_k = \frac{|G_{k,\text{pred}} \cap G_{k,\text{true}}|}{n_{k,\text{true}}}.$$

If $n_{k,\text{pred}} > n_{k,\text{true}}$, to avoid overpenalizing predictions that overestimate differential expression, we construct a reduced predicted set $\tilde{G}_{k,\text{pred}}$ by selecting the $n_{k,\text{true}}$ genes with the largest absolute log fold changes (relative to control cells) from $G_{k,\text{pred}}$, and compute

$$\text{DES}_k = \frac{|\tilde{G}_{k,\text{pred}} \cap G_{k,\text{true}}|}{n_{k,\text{true}}}.$$

The final DES is obtained by averaging DES_k across all perturbations.

d) *Overall score*: We make a scale of each metric based on the cell-mean baseline model and calculate the overall score:

$$S = \frac{\text{DES}_{\text{scaled}} + \text{PDS}_{\text{scaled}} + \text{MAE}_{\text{scaled}}}{3} \times 100$$

2) *cell-mean model*: This baseline predicts gene expression by returning the global mean expression profile computed from all perturbed cells in the training data. It ignores the input perturbation and cell-specific information, and simply broadcasts the pre-computed mean expression vector to all samples at inference time.

3) *ridge regression*: We apply a linear ridge-regression model in a PCA-reduced gene expression space. Pseudo-bulk mean expression profiles are computed for each perturbation, and a regularized linear mapping is learned to predict genome-wide expression changes from the perturbed gene, with the global perturbed mean added back to obtain the final expression profile.

4) *random forest regression*: The preprocessing and PCA representation are identical to the ridge regression baseline. A Random Forest regressor is used in place of the linear model to predict perturbation effects in the latent space, which are then mapped back to gene space to generate the final expression profile.

5) *STATE Implementation*: We implement the STATE model following the official open-source implementation released by the authors. In this work, we focus on the *State Transition (ST)* module of STATE, which is responsible for predicting perturbation-induced transcriptomic responses given control cells and perturbation conditions.

a) *Perturbation features*.: Perturbation identities are represented using pretrained protein embeddings. Specifically, gene perturbations are encoded using ESM2-based perturbation feature vectors provided by the official STATE repository. These embeddings allow the model to incorporate prior biological information about gene identity without directly accessing expression-level supervision.

b) *Model architecture*.: The ST module takes as input control cell expression profiles, perturbation embeddings, and batch embeddings to account for technical variability (Figure 1). All inputs are projected into a shared hidden space and processed by a Transformer backbone, which models how perturbations induce shifts in cellular state distributions. The output representations are mapped back to the gene expression space to generate predicted perturbed transcriptomes.

6) *scGPT Finetune*: To adapt the scGPT foundation model for the specific task of predicting cellular responses to perturbations, we employ a transfer learning strategy that balances the retention of generalized biological knowledge with the flexibility to learn condition-specific dynamics.

a) *Model Initialization and Parameter Freezing*.: We initialize the model using pre-trained scGPT checkpoints (Figure 2). To mitigate catastrophic forgetting and ensure computational efficiency, we adopt a partial freezing strategy. The core representation learning modules—specifically the gene token encoder, the continuous value encoder, and the central transformer encoder layers—are frozen. This preserves the model’s pre-learned understanding of gene-gene interactions and cellular states.

Concurrently, we designate specific components as trainable to capture perturbation-specific features. A dedicated embedding layer (`pert_encoder`) is optimized to represent distinct perturbation flags. The output assembly, specifically the affine expression decoder (`ExprDecoder`), remains trainable. This module utilizes coefficient and bias heads to reconstruct gene expression values. If the model is configured for explicit zero-probability modeling, the corresponding zero-prob heads are also optimized.

b) *Composite Loss Function*.: The fine-tuning process minimizes a composite loss function designed to align predicted expression profiles with ground truth data while preserving biologically relevant differential expression signals. For each training batch, the total loss \mathcal{L}_{total} is calculated as a weighted sum of four specific objectives.

The Sliced Wasserstein-1 distance (\mathcal{L}_{sw1}) is employed as the Distribution Alignment component to minimize the distributional discrepancy between the predicted and actual gene expression profiles. A ProtoInfoNCE loss (\mathcal{L}_{proto}) is applied to pseudobulk deltas for Contrastive Learning, encouraging the model to distinguish between different perturbation effects in the latent space. When a Differential Expression (DE) gene map is available, two auxiliary losses are included: \mathcal{L}_{de_rank} , which penalizes errors in the ranking of DE genes to prioritize biologically significant changes, and \mathcal{L}_{dir} , which enforces the correct directionality of regulation (up-regulation vs. down-regulation).

In this study, the default weighting coefficients are set to $\lambda_{sw1} = 0.60$, $\lambda_{proto} = 0.25$, $\lambda_{rank} = 0.10$, and $\lambda_{dir} = 0.05$.

c) *Optimization Procedure*.: Optimization is performed using Automatic Mixed Precision (AMP) combined with gradient scaling to maximize training throughput and numerical stability. To prevent overfitting, we implement an early stopping mechanism that monitors a validation metric (`overall_score`). Training is halted if this score fails to improve within a specified patience window, and the checkpoint corresponding to the best validation performance is saved for inference.

B. Target Gene Identification

a) *Problem Formulation*.: While forward prediction tests a model’s understanding of causality, the actionable goal in drug discovery is the inverse: inferring the mechanism that explains an observed phenotype. We formulate this through two related inverse problems:

1. Reverse Perturbation Prediction (Inverse Problem).

This task asks: *given an observed cellular response, what perturbation caused it?* Mathematically, given a control state x and a perturbed state y , we seek the perturbation \hat{p} that best explains the transition:

$$\text{given } (x, y) \Rightarrow \hat{p} = g(x, y)$$

Ideally, this is the perturbation that minimizes the discrepancy between the forward model’s prediction and the observation:

$$\hat{p} = g(x, y) = \arg \min_p d(f(x, p), y)$$

2. Target Discovery (Therapeutic Goal). This extends the inverse framework to therapeutic intervention. Here, we aim to transform a disease state x_{abn} into a desired healthy state $y_{desired}$:

$$p^* = g(x_{abn}, y_{desired}) = \arg \min_p d(f(x_{abn}, p), y_{desired})$$

Here:

- x_{abn} : The abnormal (e.g., disease-associated) cell state.
- $y_{desired}$: The target state (e.g., healthy control or a specific functional profile).
- p : The candidate perturbation/target (e.g., a gene KO from a discrete set).
- $f(x_{abn}, p)$: The forward intervention effect model.

- $d(\cdot, \cdot)$: A distance metric measuring how close the intervention brings the cell to the desired state.

We explore two methodological approaches to solve these problems:

Route A: Forward Model-Based Retrieval. This approach directly utilizes the forward simulator f . For every candidate perturbation $p \in \mathcal{P}$: 1. Predict the outcome $\hat{y}_p = f(x_{\text{abn}}, p)$. 2. Compute the distance to the target $d(\hat{y}_p, y_{\text{desired}})$. 3. Rank candidates by minimizing this distance. Here, g is implicitly defined by the optimization over f .

A critical limitation of the route A framework employed by original scGPT paper is its reliance on a pre-defined, closed candidate set composed of enumerated gene combinations. While this approach is feasible for a restricted subspace—such as 20 genes yielding 210 distinct single- and double-gene perturbation classes—it suffers from prohibitive combinatorial explosion when applied to larger gene pools or when the specific number of target genes is unknown. As the search space expands to accommodate genome-wide candidates or higher-order interactions, the number of potential combinatorial targets increases exponentially, rendering the exhaustive generation and retrieval of candidate profiles computationally intractable. Consequently, due to these scalability constraints and the infeasibility of defining a discrete label space for unknown perturbation complexities, we exclude this combinatorial classification route from our methodology.

Route B: Direct Inverse Mapping. This approach learns a parameterized inverse function g_θ (e.g., a classifier or regressor) to directly predict the optimal perturbation:

$$p^* \approx g_\theta(x_{\text{abn}}, y_{\text{desired}})$$

Instead of explicitly modeling the transition, the model learns the mapping from (state, target) pairs to perturbations directly from data. This absorbs the forward dynamics and distance metric into the model weights.

b) Evaluation Metrics.: To quantitatively assess the performance of the reverse perturbation prediction models, we employ a suite of ranking-based metrics. Let $\mathcal{C}_{\text{test}}$ denote the set of test perturbation conditions. For each condition $c \in \mathcal{C}_{\text{test}}$, let \mathcal{G}_c^* be the set of ground-truth target genes, and let \mathcal{G}_c^K represent the set of the top- K genes identified by the model based on the predicted scores.

1. Mean Reciprocal Rank (MRR). The Mean Reciprocal Rank evaluates the model’s ability to place the correct perturbation targets at the top of the ranking list. Since a perturbation may involve multiple target genes (e.g., in combinatorial perturbations), we define the reciprocal rank based on the *highest-ranked* (best) true target.

$$\text{MRR} = \frac{1}{|\mathcal{C}_{\text{test}}|} \sum_{c \in \mathcal{C}_{\text{test}}} \max_{g \in \mathcal{G}_c^*} \frac{1}{\text{rank}(g)}$$

where $\text{rank}(g)$ denotes the position of gene g in the predicted descending order list (1-indexed). A higher MRR indicates that the first correct target appears earlier in the candidate list.

2. Exact Hit@K. Exact Hit@K is a stringent metric that measures the proportion of test cases where the model successfully retrieves the *entire* set of target genes within the top- K predictions. This metric is particularly relevant for identifying complete combinatorial pairs.

$$\text{Exact Hit@K} = \frac{1}{|\mathcal{C}_{\text{test}}|} \sum_{c \in \mathcal{C}_{\text{test}}} \mathbb{I}(\mathcal{G}_c^* \subseteq \mathcal{G}_c^K)$$

where $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition is true and 0 otherwise.

3. Relevant Hit@K. Relevant Hit@K assesses the model’s capacity to discover *at least one* correct driver gene within the top- K candidates. This metric reflects the practical utility of the model in experimental screening, where identifying a subset of drivers is often sufficient for further validation.

$$\text{Relevant Hit@K} = \frac{1}{|\mathcal{C}_{\text{test}}|} \sum_{c \in \mathcal{C}_{\text{test}}} \mathbb{I}(\mathcal{G}_c^* \cap \mathcal{G}_c^K \neq \emptyset)$$

4. Recall@K. Recall@K quantifies the fraction of ground-truth target genes that are successfully retrieved within the top- K predictions. Unlike Relevant Hit@K, which is binary, Recall@K penalizes the model for missing parts of a combinatorial perturbation (e.g., finding only 1 gene of a 2-gene pair).

$$\text{Recall@K} = \frac{1}{|\mathcal{C}_{\text{test}}|} \sum_{c \in \mathcal{C}_{\text{test}}} \frac{|\mathcal{G}_c^* \cap \mathcal{G}_c^K|}{|\mathcal{G}_c^*|}$$

This metric provides a granular view of the model’s retrieval completeness, particularly as K increases (e.g., Recall@1, Recall@5, Recall@10, etc.).

1) Baseline Models:

a) PCA + kNN Baseline.: To establish a retrieval-based baseline, we first aggregate single-cell gene expression vectors into pseudobulk profiles for each perturbation condition. This yields a feature vector $\mathbf{x}_c \in \mathbb{R}^G$ over G genes, associated with a multi-hot label vector $\mathbf{y}_c \in \{0, 1\}^G$ derived from the specific perturbation targets. To prevent trivial mappings (i.e., data leakage), we optionally mask the expression values of target genes within \mathbf{x}_c . We apply Principal Component Analysis (PCA) to the training profiles, projecting them into a latent space $\mathbf{z}_c = \mathbf{W}^\top \mathbf{x}_c \in \mathbb{R}^d$. For a given query condition q , we retrieve the k nearest neighbors in the PCA space ($\mathcal{N}_k(q)$) using Cosine or Euclidean similarity. The final gene relevance scores s_q are computed via a weighted aggregation of the neighbors’ labels. Hyperparameters, including the latent dimension d and neighborhood size k , are optimized based on the Mean Reciprocal Rank (MRR) of the validation set before final evaluation on held-out test conditions.

b) Random Forest Baseline.: Utilizing the same pseudobulk representations \mathbf{x}_c as input, we employ a multi-output Random Forest regressor to predict continuous per-gene relevance scores $\hat{\mathbf{y}}_c \in \mathbb{R}^G$ (or a restricted subset of candidate targets). Each tree in the ensemble learns non-linear splits on the input features, and the final prediction constitutes an average over T trees. Target vectors are constructed as multi-hot encodings of the condition labels, with optional input masking applied to

ensure generalization. Hyperparameters such as the number of estimators T and maximum tree depth are selected via validation MRR. The predicted score vectors are subsequently ranked to compute Top-K metrics on the test split.

c) XGBoost Baseline.: We further benchmark performance using Gradient Boosted Decision Trees (GBDT) implemented via XGBoost in a multi-output regression setting. The model predicts per-gene scores by optimizing a squared-error objective function $\mathcal{L} = \sum_c \|\mathbf{y}_c - \hat{\mathbf{y}}_c\|_2^2$ through an additive training process, where η represents the learning rate and f_m represents the m -th tree. This approach captures non-linear gene-score relationships while managing model complexity through depth constraints and subsampling. As with other baselines, inputs \mathbf{x}_c are subject to masking strategies to prevent leakage. Hyperparameters are tuned on validation MRR, and the final evaluation employs the same ranking-based metrics as the PCA+kNN and Random Forest baselines.

2) Fine-tuning scGPT for Discriminative Perturbation Prediction: This section details the methodology for the discriminative fine-tuning strategy ("Route B1"), which leverages the pre-trained scGPT backbone to identify perturbation targets at single-cell resolution. Unlike the pseudobulk baselines, this approach models the perturbation likelihood for every gene within individual cells, utilizing a control-aware embedding strategy to isolate perturbation-specific signals.

a) Task Formulation.: The reverse perturbation prediction is formulated as a ranking problem. Given a perturbed single-cell expression profile \mathbf{x} and its associated condition context, the model learns a scoring function $f(\mathbf{x}) : \mathbb{R}^G \rightarrow \mathbb{R}^G$ that outputs a vector of logits $\hat{\mathbf{p}}$. These logits represent the likelihood of each gene being a target of the perturbation. The ground truth is represented as a multi-hot vector $\mathbf{y} \in \{0, 1\}^G$, where $y_j = 1$ if gene j is a target in the condition string (e.g., "GeneA+GeneB"), and 0 otherwise.

b) Architecture and Input Processing.: The model architecture builds upon the scGPT transformer backbone (Figure 3). Single-cell expression counts are preprocessed via the standard scGPT pipeline: counts are normalized, log-transformed, and quantile-binned. Non-zero gene expressions are tokenized into gene-value pairs and prepended with a [CLS] token, yielding a sequence input for the encoder.

To mitigate confounding biological variation (e.g., cell cycle or batch effects) and emphasize the perturbation signal, we employ a control-matched delta embedding strategy:

- **Encoder Embedding:** The perturbed cell i is embedded by the scGPT encoder to obtain a latent representation \mathbf{e}_i .
- **Reference Aggregation:** We sample K matched control cells sharing the same batch and cell-type metadata. Their embeddings are averaged to form a reference baseline: $\mathbf{e}_{ref} = \frac{1}{K} \sum_{k=1}^K \mathbf{e}_{i,k}^{ctrl}$.
- **Delta Representation:** A perturbation-specific embedding \mathbf{h}_i is computed by subtracting the reference signal:

$$\mathbf{h}_i = \mathbf{e}_i - \mathbf{e}_{ref}$$

The gene-scoring head projects this delta embedding \mathbf{h}_i into the gene embedding space. Let \mathbf{g}_j denote the pre-trained embedding for gene j . The score s_{ij} for gene j in cell i is computed via a learnable projection \mathbf{W} and a dot product:

$$s_{ij} = \langle \mathbf{W}\mathbf{h}_i, \mathbf{g}_j \rangle$$

Alternatively, a multi-layer perceptron (MLP) can be applied to the concatenation $[\mathbf{h}_i; \mathbf{g}_j]$.

c) Optimization Objective.: The training objective is a composite loss function designed to rank true perturbation targets higher than non-targets while maintaining calibration. The total loss \mathcal{L} is a weighted sum:

$$\mathcal{L} = \lambda_{rank} \mathcal{L}_{rank} + \lambda_{bce} \mathcal{L}_{bce}$$

In this study, we utilize weights $\lambda_{rank} = 0.7$ and $\lambda_{bce} = 0.1$. The components are defined as follows:

- **Pairwise Ranking Loss (\mathcal{L}_{rank}):** This loss enforces that the scores of ground-truth target genes ($p \in \mathcal{P}$) are higher than those of sampled negative genes ($n \in \mathcal{N}$) by a margin m :

$$\mathcal{L}_{rank} = \mathbb{E}_{p,n} [\max(0, m - (s_{i,p} - s_{i,n}))]$$

Negative samples \mathcal{N} are drawn using a mix of random sampling and "hard negative" mining (high-scoring non-targets) to facilitate robust learning.

- **Auxiliary Binary Cross-Entropy (\mathcal{L}_{bce}):** To stabilize training and provide probabilistic calibration, we apply BCE loss on the positive set plus a subset of sampled negatives. This avoids gradient dominance from the vast majority of non-target genes (class imbalance) while guiding the model to predict correct multi-hot labels.

C. Transcriptome-wide Perturbation Prediction

1) Experimental Setup:

a) Data preprocessing.: We use the ARC Institute VCC competition support dataset (competition_support_set), which provides preprocessed single-cell RNA-seq data for H1 human embryonic stem cells. Gene expression values are transformed using a log1p normalization, which stabilizes variance and reduces the dominance of highly expressed genes. Control cells corresponding to non-targeting perturbations are retained and used as a reference state.

b) Data Split.: Instead of using the original VCC split, we re-split 150 gene perturbations into training, validation, and test sets at the perturbation level to evaluate generalization to unseen perturbations. Specifically, 64% of perturbation genes are used for training, 16% for validation, and the remaining 20% (30 genes) form a held-out test set. Control (non-targeting) cells are included in all splits to provide a consistent reference state. Importantly, there is no overlap of perturbation genes between the training, validation, and test sets, ensuring that models are evaluated on entirely unseen perturbation targets.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MODELS.

Model	PDS	MAE	DES	Overall
Cell-mean baseline	0.5167	0.1258	0.1075	0.00
Ridge Regression	0.5167	0.1253	0.1466	1.59
Random Forest Regression	0.5167	0.1253	0.1360	1.19
State Model	0.5367	0.1286	0.3023	7.28
scGPT finetune	0.5089	0.2763	0.2620	6.27

2) *Main Results*: We select Ridge Regression and Random Forest Regression as representative baseline models, as their overall scores fall in the range of 1 to 2. Among all methods, the State Model achieves the best overall performance, with substantial improvements in both DES and PDS. The scGPT fine-tuned model also shows competitive performance in DES and overall score, but suffers from a significantly higher MAE. Overall, more complex models tend to improve DES and PDS, but these gains do not consistently translate into lower MAE. The detailed results are reported in Table III.

The aim of the Virtual Cell Challenge (VCC) was to predict post-perturbation single-cell gene expression in the H1 cell based on 150 available perturbations. Though the dataset itself is single-cell, the competition metrics PDS (Perturbation Discrimination Score) and MAE (Mean Absolute Error) were calculated on the pseudobulk gene expression profiles. The third competition metric, DES (Differential Expression Score), in theory, is a more challenging metric as it does consider the single-cell expression distributions and uses standard differentially expressed gene algorithms to infer differentially expressed genes.

Conceptually, DES evaluates a model’s ability to faithfully reproduce biologically interpretable perturbation effects, focusing on whether differentially expressed genes are correctly identified and whether the most influential genes are properly prioritized. As such, DES provides a direct measure of how well a model captures single-cell-level perturbation signals.

Under these metrics, our model learns meaningful perturbation patterns at the single-cell level, as evidenced by non-trivial DES scores. However, the predicted expression changes for individual cells are often small or close to zero, causing the learned effects to be attenuated when aggregated into pseudobulk representations ($\hat{\delta}_k$). This attenuation partially explains the relatively low PDS scores observed.

Furthermore, single-cell data are inherently noisy, which can interfere with effective model training. Since our model is optimized at the single-cell level rather than directly on pseudo-bulk, this mismatch likely further contributes to the lower performance on PDS.

D. Target Gene Identification

1) *Experimental Setup*: In our experiments, we instantiate these tasks (Section B) as follows:

- **Norman Dataset (Genetic Perturbations)**: We define the task as *Reverse Genetic Perturbation Identification*. The input is the perturbed expression profile (with optional control baseline), and the output is a ranking

of potential perturbation genes. Ground truth labels are derived from Perturb-seq metadata, and performance is evaluated using ranking metrics (Hit@K, MRR, NDCG).

- **Tahoe Dataset (Drug Perturbations)**: We define the task as *Reverse Drug Target Identification*. The input is the drug-treated vs. DMSO control profile (or delta signature). The output is a ranking of the drug’s target gene. By focusing on the single-target subset, we ensure a clean label space for evaluation.

a) *Dataset and Preprocessing*.: We evaluated our method using the Norman et al. Perturb-seq dataset, pre-processed via the GEARS framework. The dataset comprises $N = 91,205$ cells and $G = 5,045$ log-normalized Highly Variable Genes (HVGs). Gene symbols were mapped to the pre-trained scGPT vocabulary to ensure compatibility. While control cells (unperturbed) were retained to serve as references for computing perturbation-specific shifts (e.g., delta embeddings), the “control” class itself was excluded from the target label space.

b) *Condition Filtering and Canonicalization*.: To ensure robust statistical modeling, we applied quality control filters to the perturbation conditions. Conditions were first canonicalized to enforce lexicographic order (e.g., treating $A + B$ and $B + A$ as identical) and to remove redundant control strings. We subsequently filtered out conditions with insufficient cell coverage, requiring a minimum of 50 cells for single-gene perturbations and 30 cells for combinatorial (double) perturbations. This process yielded a curated set of valid conditions for splitting.

c) *Compositional Splitting Strategy*.: We employed a strict condition-level splitting strategy to prevent data leakage, ensuring that no perturbation condition appeared in more than one data split (train/validation/test). To rigorously test the model’s ability to generalize to novel biological contexts, we adopted a “compositional split” design:

- **Unseen Single Genes**: A fraction of single-gene perturbations (15%) was held out as “unseen.”
- **Combinatorial Logic**: Combinatorial perturbations were categorized based on the training visibility of their constituent genes. “Seen2” doubles (where both genes are present in the training set as singles) were stratified by gene frequency and cell count to ensure balanced distribution across splits. Conversely, “Seen1” (one gene unseen) and “Seen0” (both genes unseen) combinations were exclusively allocated to the test set to evaluate zero-shot generalization.

d) *Dataset Statistics*.: The final filtered dataset consisted of 236 unique perturbation conditions across 83,803 cells. Using a fixed random seed (42), the data was partitioned as follows:

- **Training Set**: 147 conditions (56,580 cells), comprising 81 single perturbations and 66 “Seen2” doubles.
- **Validation Set**: 23 conditions (6,560 cells).
- **Test Set**: 66 conditions (20,712 cells).

To facilitate detailed performance analysis, the test set was further stratified into four generalization tiers: 15 unseen

single perturbations, 15 "Seen2" doubles, 31 "Seen1" doubles, and 5 "Seen0" doubles. This split was persisted and applied uniformly across all benchmarked models (scGPT, PCA+kNN, Random Forest, XGBoost) to ensure a standardized evaluation protocol.