From Transcriptome-wide Prediction to Target Gene Discovery: Improving Virtual Cell Models with scGPT

Anrui Wang
2023533015
wangar2023@shanghaitech.edu.cn

Jiawen Dai
2023533132
daijw2023@shanghaitech.edu.cn

Yiting Qi
2023533043
qiyt2023@shanghaitech.edu.cn

Abstract

Predicting cellular responses to genetic or drug perturbations is a fundamental challenge in computational biology, with significant implications for drug discovery and functional genomics. Traditional virtual cell modeling, exemplified by the Virtual Cell Challenge (VCC), formulates this task as transcriptome-wide regression. However, transcriptome-wide evaluation suffers from high dimensionality, noisy measurements, and metrics dominated by non-informative genes, limiting its biological interpretability. In this work, we reformulate the task as target gene identification, ranking candidate genes by their likelihood of being true perturbation targets. We implement a refined evaluation framework for the VCC benchmark, re-splitting 150 known gene perturbations into train, validation, and test sets, and assess the performance of STATE and scGPT models. Furthermore, we develop scGPT-based models tailored for target gene discovery and extend our evaluation to drug perturbations using the Tahoe-100M dataset. Our study highlights both the limitations of transcriptome-wide metrics and the promise of target-focused modeling, providing insights for more biologically meaningful virtual cell predictions.

1 Introduction

1.1 Phenotypic screening creates an inverse problem

Phenotypic drug discovery has historically been a cornerstone of therapeutic development, uniquely capable of identifying compounds that rescue disease states in a physiologically relevant context without requiring prior knowledge of a specific molecular target. However, this “target-agnostic” success comes with a fundamental **inverse problem**: while the therapeutic effect—the phenotype—is observed, the underlying molecular mechanism of action (MoA) or target often remains unknown, necessitating a complex and laborious target deconvolution process. Unlike target-based approaches that proceed from cause to effect, phenotypic screening presents us with the effect, requiring us to solve for the cause.

High-throughput transcriptomic profiling has emerged as a critical bridge in this gap, offering a high-dimensional, information-rich “fingerprint” of cellular states that reflects the downstream consequences of upstream perturbations. By defining the biological phenotype as a computable vector, it becomes possible to systematically map the causal relationship between a perturbation (whether genetic or chemical) and its resulting cellular state. This capability transforms the biological question into a data-driven inference task: if we can quantify the phenotypic shift, can we identify the perturbation that caused it?

Recent advances in artificial intelligence have begun to address this by proposing the construction of “Virtual Cells”—high-fidelity simulators learned directly from large-scale biological data. As highlighted by Roohani et al. [RHT$^+$25], these systems are expected to learn the fundamental relationship between cell state and function, with the primary intent of predicting the consequences of perturbations—such as gene knockdowns or drug treatments—across diverse cell contexts. While much of the current effort focuses on this **forward prediction**, the ultimate utility of AI Virtual Cells (AIVCs) extends beyond mere simulation to “closing the loop” for discovery. Bunne et al. [BRR$^+$24] articulate a vision where AIVCs serve as engines for **in silico experimentation**, capable of simulating the effects of varying interventions to propose potential causal factors behind observed phenotypes. Although computation alone may not fully resolve all causal links, AIVCs offer the critical potential to drastically **“reduce the space of possible hypotheses”**, thereby accelerating the identification of underlying mechanisms and new drug targets. This positions the Virtual Cell not just as a simulator, but as a computational partner for target deconvolution in phenotypic drug discovery.

1.2 The community’s asymmetry: optimizing forward metrics vs answering actionable questions

The field currently faces a critical asymmetry: while immense effort is poured into optimizing forward prediction metrics, empirical gains remain marginal. A systematic benchmark across diverse datasets reveals that complex deep learning models do not consistently outperform simpler alternatives, with performance often heavily confounded by perturbation effect sizes rather than architectural superiority [LYF$^+$24]. More strikingly, recent rigorous evaluations serve as a wake-up call, demonstrating that state-of-the-art models for perturbation prediction do not yet consistently outperform simple linear baselines or even trivial mean predictions [AE]. This suggests that the singular pursuit of minimizing reconstruction error (MSE) may not inherently lead to the actionable causal discovery required for drug development.

1.3 Reframing: invert the causal chain to make models actionable

To bridge the gap between model performance and biological discovery, we propose a fundamental reframing of the Virtual Cell’s utility: shifting the focus from **forward simulation** (predicting the expression profile of a known perturbation) to **reverse retrieval** (inferring the causal perturbation of an observed profile). This is not an arbitrary task invention but a rigorous operationalization of the "in silico reverse perturbation prediction" paradigm already emerging in foundational models. For instance, **scGPT** [CWM$^+$24] explicitly formalizes this as a top-K retrieval task, demonstrating that foundation models can learn to rank potential genetic drivers based on their alignment with a target cell state. By adopting this retrieval-based formulation, we align the model’s output directly with the decision-making logic of drug discovery: generating a ranked list of probable targets for experimental validation.

This inversion transforms the Virtual Cell from a descriptive simulator into a prescriptive engine for **therapeutic target identification**. This utility has been notably demonstrated by **Geneformer** [TXC$^+$23], which leveraged transfer learning on large-scale data to move beyond mere expression forecasting, successfully prioritizing candidate therapeutic targets and key network regulators for cardiomyopathy. However, reliable reverse inference requires models to possess true mechanistic reasoning rather than simple pattern matching, particularly when dealing with the combinatorial complexity of biological systems. As evidenced by **GEARS** [RHL24], incorporating biological inductive biases—such as gene regulatory graphs—is essential for generalizing to unseen perturbations and combinatorial sets that lack direct experimental training data. Similarly, achieving robust target discovery requires models (like the transformer-based STATE [AGB$^+$]) to model distributions across **sets of cells**, enabling mechanistic generalization to novel cellular contexts where training data may be sparse or absent. By closing this causal loop, we move from optimizing abstract error metrics to answering the actionable question: *what perturbation caused this phenotype?*

1.4 Contributions

In this work, we close the loop between forward simulation and reverse inference, presenting a unified framework that repurposes the Virtual Cell from a predictive simulator into a prescriptive engine for target discovery. Our primary contributions are as follows:

1. **Rigorous Evaluation of Forward Generalization:** We establish a strict evaluation protocol for forward perturbation prediction using the **Virtual Cell Challenge (VCC)** dataset [RHT+25]. Unlike standard benchmarks that often rely on random cell-splits, we enforce **perturbation-level splits** to test true mechanistic generalization. By evaluating state-of-the-art models (including scGPT and STATE) against biologically grounded metrics (DES, PDS) rather than simple reconstruction error, we validate their capacity to capture causal perturbation-response maps beyond mere data fitting.
2. **A Unified Paradigm for Reverse Target Retrieval:** We formally define the task of **in silico reverse perturbation prediction** as a retrieval and ranking problem. We demonstrate that this paradigm—which takes a transcriptomic phenotype as query to retrieve the most probable causal agents—can be effectively generalized across different perturbation modalities. This validates the feasibility of using deep perturbation models to bridge the gap between phenotypic screening readouts and target identification, aligning drug-induced signatures with genetic ground truths.
3. **Decoupling Geometric and Semantic Inference:** We uncover a critical **"Geometry vs. Semantics" trade-off** in phenotypic retrieval. Our comparative analysis reveals that while simple heuristic baselines (e.g., PCA-KNN) excel at "head" precision for strong phenotypes via geometric similarity, pre-trained foundation models (e.g., scGPT) demonstrate superior robustness and "tail" recall. This finding suggests that foundation models uniquely capture the semantic context required for deciphering complex or subtle biological signals that lie beyond simple geometric proximity.

Author Contributions

- **Anrui Wang**: Implemented scGPT experiments to fulfill the Virtual Cell Challenge (VCC) requirements, evaluated model performance, adapted scGPT to leverage the improved VCC evaluation metrics, and created scGPT-based models specifically for target gene identification, enabling a more biologically meaningful assessment of predictions.
- **Jiawen Dai**: Reimplemented a more biologically informed set of evaluation metrics for the Virtual Cell Challenge (VCC), conducted baseline experiments for VCC models, and extracted relevant data from the Tahoe-100M dataset.
- **Yiting Qi**: Implemented STATE experiments to fulfill the VCC requirements and evaluate performance, conducted studies on the Tahoe-100M dataset, and developed baseline models for target gene prediction on the Tahoe-100M dataset.

2 Forward validation as a credibility scaffold

2.1 Forward prediction on VCC under perturbation-level splits

To establish a rigorous baseline for virtual cell capabilities, we first benchmarked state-of-the-art models on the **Virtual Cell Challenge (VCC)** dataset. Unlike standard evaluations that often rely on random cell splits—which risk simplifying the task to cellular interpolation—we enforced a strict **perturbation-level split**. In this setting, the models were tasked with predicting the transcriptional effects of genetic perturbations that were entirely unseen during training, thereby testing true mechanistic generalization rather than pattern memorization.

We evaluated two representative model architectures: **STATE**, which leverages cell-set modeling to capture distributional shifts across cellular contexts, and **scGPT**, a foundational model that utilizes large-scale pre-training to learn generalizable gene-gene interactions. By systematically comparing these models alongside linear baselines, we defined the current performance boundary for forward prediction (Figure 2). This "forward validation" step serves as a necessary credibility scaffold; it confirms that the underlying embeddings capture sufficient biological signal to support the more complex inference tasks required for target discovery, without assuming that forward simulation is the endpoint of the modeling effort. All implementation details, including data splits and hyperparameter configurations, are detailed in Supplementary Note 1.

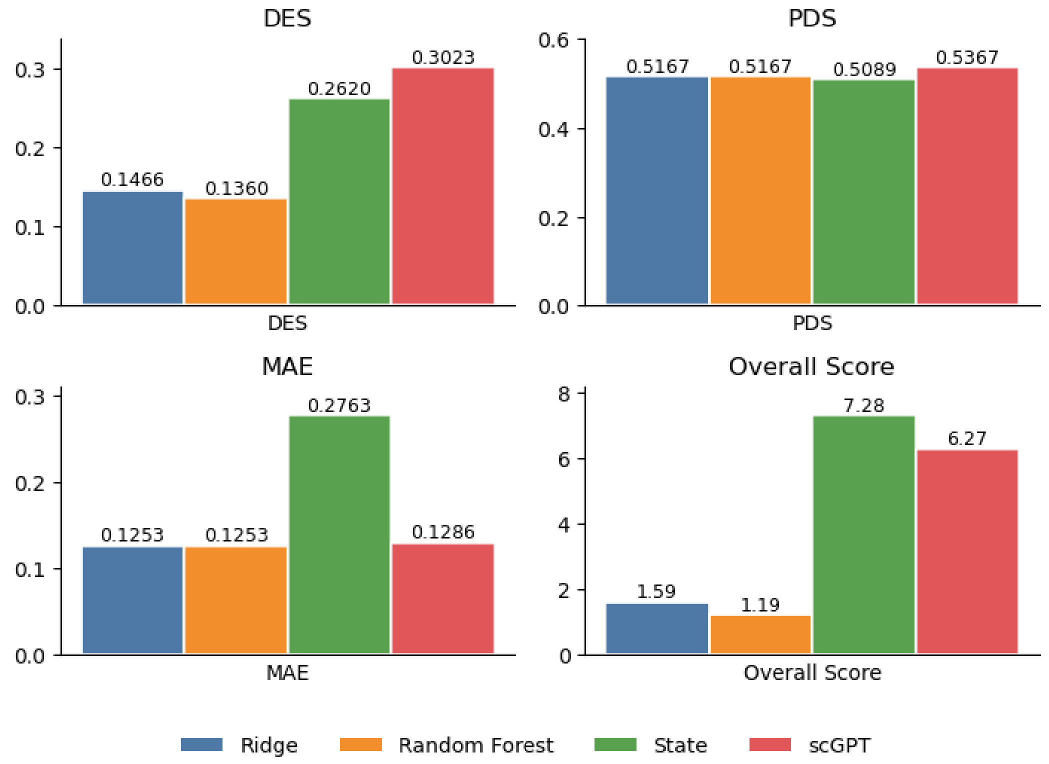

Figure 1: **Evaluation Metrics and Scoring Framework.** The Virtual Cell Challenge utilizes a composite scoring system derived from three complementary metrics to assess forward prediction performance. (A) Differential Expression Score (DES): Measures the model's fidelity in recovering biologically interpretable 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).

2.2 Why we evaluate biology, not just numbers

In evaluating these models, we depart from a reliance on raw reconstruction errors, which can be misleading in transcriptomic data. Conventional metrics like Mean Absolute Error (MAE) are frequently dominated by the high baseline expression of invariant genes, potentially awarding high scores to trivial "mean-prediction" baselines while failing to distinguish true biological replicates from unrelated perturbations. Instead, as illustrated in Figure 2, we prioritize metrics that assess the **perturbation delta** (δ)—the specific causal shift in gene expression induced by the intervention relative to controls. We focus on two biologically interpretable metrics to serve as our primary credibility scaffold:

- **Differential Expression Score (DES):** This metric evaluates the model's ability to recover the specific set of differentially expressed genes (DEGs). By measuring the overlap (precision and recall) between predicted and ground-truth DEGs, DES verifies whether the model faithfully recapitulates the biologically interpretable effects of a perturbation, rather than merely fitting global statistical distributions.

- **Perturbation Discrimination Score (PDS):** Adapted here to utilize cosine similarity, PDS assesses the directional fidelity of the predicted shift. This ensures that the model correctly ranks the true perturbation as the most similar candidate among all possibilities. Unlike L1-based rankings, this approach emphasizes the alignment of up- or down-regulation vectors rather than signal magnitude, mitigating the penalty on "attenuated" predictions often seen in single-cell modeling.

This evaluation philosophy aligns with the "delta-based" validation advocated in recent foundational studies, ensuring that high performance reflects the capture of causal regulatory signals rather than the preservation of static cell-type identity.

Metric Analysis and Failure Modes The selection of these metrics addresses specific failure modes in predictive modeling (Figure 2D). 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 dampened 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.

2.3 Takeaway: forward is necessary but not sufficient

While forward benchmarks verify that a model can simulate *effects*, therapeutic discovery necessitates inferring *causes* from observed effects. This fundamental asymmetry motivates an explicit inversion of the perturbation–phenotype mapping, moving from verifying simulation accuracy to validating retrieval capability.

3 Discussion

4 Conclusion

References

[AE] Constantin Ahlmann-Eltze. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines.

[AGB$^+$] 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.

[BRR$^+$24] 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, Srigokul 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].

[CWM$^+$24] 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.

[LYF+24] 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.

[RHL24] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, June 2024.

[RHT+25] 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.

[TXC+23] 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.

Appendix A Supplementary Note 1: Methods

Appendix A.1 Transcriptome-wide Perturbation Prediction

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.

Appendix A.1.1 Evaluation metrics: PDS, MAE, DES

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 $k \in \{1, \ldots, N\}$, 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 $\text{nPDS}_{\text{rank}} = \frac{1}{N^2}\sum_{k=1}^N R_k$.

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 $\log_2$ fold change for each gene g using a pseudocount of 1:

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

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.

Differential Expression Score (DES). The Differential Expression Score (DES) measures how accurately a model recovers perturbation-induced differential gene expression. For each perturbation $k \in \{1, \ldots, N\}$, 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 $G_{k,\mathrm{pred}}$ and a ground-truth DE gene set $G_{k,\mathrm{true}}$, with sizes $n_{k,\mathrm{pred}} = |G_{k,\mathrm{pred}}|$ and $n_{k,\mathrm{true}} = |G_{k,\mathrm{true}}|$, respectively. If $n_{k,\mathrm{pred}} \leq n_{k,\mathrm{true}}$, 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,

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

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

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

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

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

$$S = \frac{DES_{\text{scaled}} + PDS_{\text{scaled}} + MAE_{\text{scaled}}}{3} \times 100$$

Appendix A.1.2 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.

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.

Model architecture. The ST module takes as input control cell expression profiles, perturbation embeddings, and batch embeddings to account for technical variability (Figure 2). 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.

Appendix A.1.3 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.

Model Initialization and Parameter Freezing. We initialize the model using pre-trained scGPT checkpoints (Figure 3). 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.

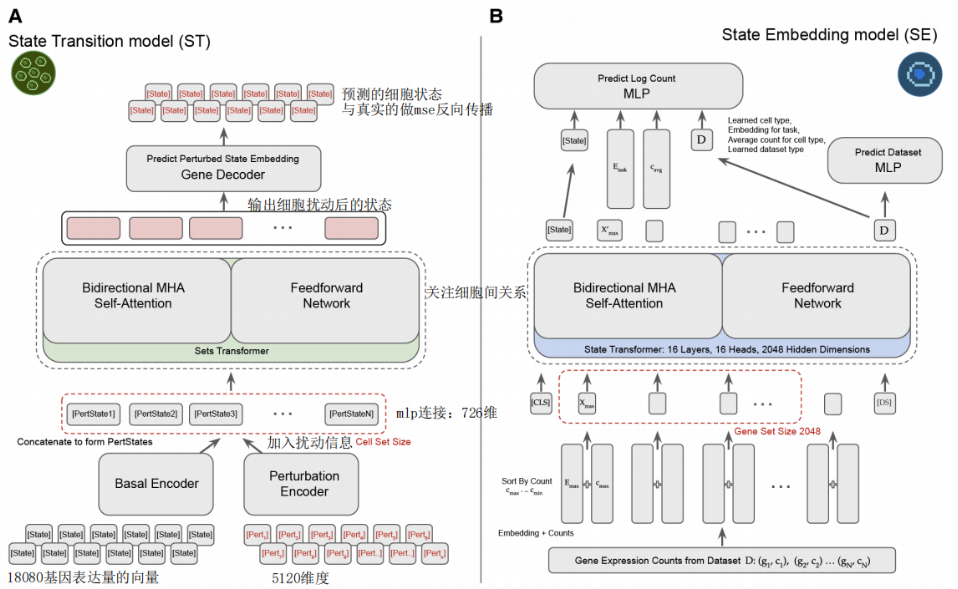

Figure 2: The architecture of the STATE model. The State Transition (ST) module integrates control expression, perturbation embeddings, and batch information to predict post-perturbation transcriptomic states.

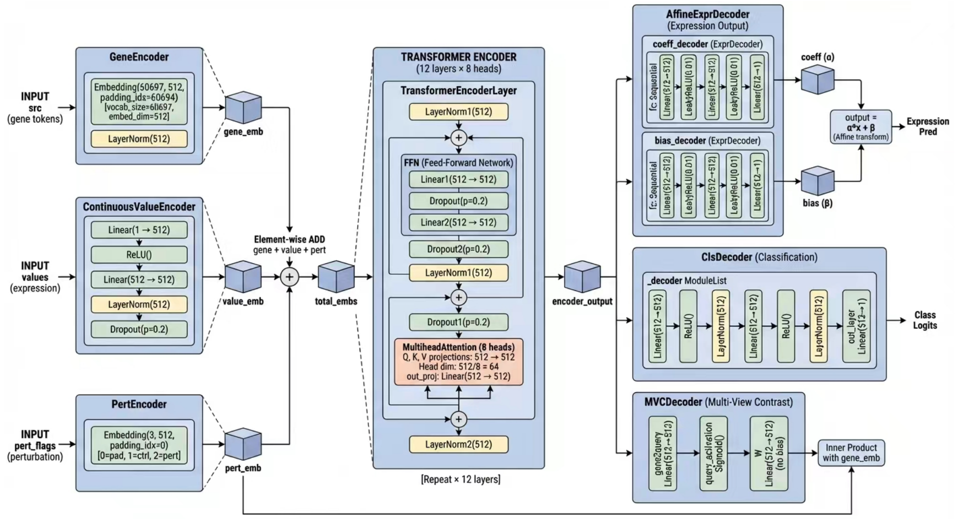

Figure 3: Overview of the scGPT model adaptation for perturbation prediction. The model is initialized with pre-trained weights, with core modules frozen to retain biological knowledge, while the perturbation encoder and decoders are fine-tuned.

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 $\lambda_{dir} = 0.05$.

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.

Appendix A.2 Target Gene Identification

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}:

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

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).
- $f(x_{\text{abn}}, p)$: 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.

Evaluation Metrics. To quantitatively assess the performance of the reverse perturbation prediction models, we employ a suite of ranking-based metrics. Let $\mathcal{C}_{test}$ denote the set of test perturbation conditions. For each condition $c \in \mathcal{C}_{test}$, 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.

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

where 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.

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

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.

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

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).

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

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.).

Appendix A.2.1 Baseline Models

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 $\mathbf{s}_q$ 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.

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.

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.

Appendix A.2.2 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.

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 $f(\mathbf{x}) : \mathbb{R}^G \to \mathbb{R}^G$ 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.

Architecture and Input Processing. The model architecture builds upon the scGPT transformer backbone. 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: $\mathbf{e}_{ref} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{e}_{i,k}^{\text{ctrl}}$.
- **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 $[\mathbf{h}_i; \mathbf{g}_j]$.

Table 1: 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

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:

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

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

- **Pairwise Ranking Loss ($\mathcal{L}_{\text{rank}}$):** 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:

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

  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 ($\mathcal{L}_{\text{bce}}$):** 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.

Appendix B Supplementary Note 2: Experimental Details

Appendix B.1 Transcriptome-wide Perturbation Prediction

Appendix B.1.1 Experimental Setup

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.

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.

Appendix B.1.2 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 1.

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.

Appendix B.2 Target Gene Identification

Appendix B.2.1 Experimental Setup

In our experiments, we instantiate these tasks (Section Appendix A.2) 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.

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.

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.

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.

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.

Appendix B.2.2 Results

The quantitative results for the Reverse Genetic Perturbation Identification task are summarized in Table 2. Interestingly, robust baselines operating on pseudobulk profiles (PCA+kNN and Random Forest) outperform the scGPT single-cell fine-tuning approach in top-ranking metrics, such as MRR (0.3602 vs 0.1975). This indicates that for immediate precision (Hit@1, Hit@5), the aggregated signal used by pseudobulk methods is cleaner and more discriminative than noisy single-cell logits. However, as the retrieval window expands (Hit@20, Hit@40), scGPT demonstrates superior recall, significantly overtaking the baselines (Recall@40 of 0.5152 vs 0.3684). This suggests that while scGPT successfully identifies valid perturbation signals, it tends to rank them lower in the candidate list, likely due to the difficulty of optimizing rank-based objectives directly on high-variance single-cell data.

Table 2: 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