GRAPE: Heterogeneous Graph Representation Learning for Genetic Perturbation with Coding and Non-Coding Biotype

Changxi Chi[1,2], **Jun Xia**[1,2], **Jingbo Zhou**[1,2], **Jiabei Cheng**[3], **Chang Yu**[2] and **Stan Z. Li**[2]
[1]Zhejiang University, Hangzhou
[2]AI Lab, Research Center for Industries of the Future, Westlake University
[3]Department of Automation, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University
{chichangxi, xiajun}@westlake.edu.cn

Abstract

Predicting genetic perturbations enables the identification of potentially crucial genes prior to wet-lab experiments, significantly improving overall experimental efficiency. Since genes are the foundation of cellular life, building gene regulatory networks (GRN) is essential to understand and predict the effects of genetic perturbations. However, current methods fail to fully leverage gene-related information, and solely rely on simple evaluation metrics to construct coarse-grained GRN. More importantly, they ignore functional differences between biotypes, limiting the ability to capture potential gene interactions. In this work, we leverage pretrained large language model and DNA sequence model to extract features from gene descriptions and DNA sequence data, respectively, which serve as the initialization for gene representations. Additionally, we introduce gene biotype information for the first time in genetic perturbation, simulating the distinct roles of genes with different biotypes in regulating cellular processes, while capturing implicit gene relationships through graph structure learning (GSL). We propose **GRAPE**, a heterogeneous graph neural network (HGNN) that leverages gene representations initialized with features from descriptions and sequences, models the distinct roles of genes with different biotypes, and dynamically refines the GRN through GSL. The results on publicly available datasets show that our method achieves state-of-the-art performance. The code for reproducing the results can be seen at the link: https://github.com/ChangxiChi/GRAPE.

1 Introduction

Single-cell genetic perturbation refers to the process of knocking out specific genes in single cells using CRISPR technology [Barrangou and Doudna, 2016; Lino *et al.*, 2018], thereby altering their normal physiological state or behavior. Despite significant advances in single-cell technology and high-throughput screening, it remains impractical to perform perturbation experiments on all genes within the vast genome of a cell. Accurate prediction of the effects of untested genetic perturbations helps reveal the specific roles of target genes in cellular biological processes and uncovers the regulatory relationships and network structures between genes. Using predictive models, potentially important genes can be identified before experiments, improving the efficiency of the utilization of experimental resources.

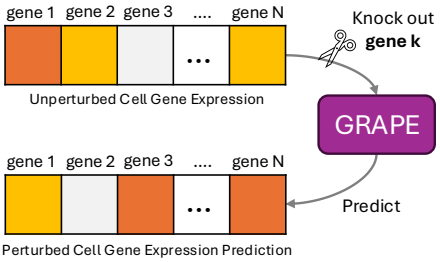

Figure 1: Schematic diagram of genetic perturbation.

Generative models have been widely applied in perturbation modeling, with existing approaches focusing on modeling the latent distribution of data to generate gene expression changes under various perturbations. SAMS-VAE [Bereket and Karaletsos, 2024] enhances model interpretability by using sparse global latent variables to disentangle the specific effects of perturbations. GraphVCI [Wu *et al.*, 2022], on the other hand, leverages graph convolutional networks to model gene regulatory networks (GRN) and predict gene expression under counterfactual perturbations. In contrast, GEARS [Roohani *et al.*, 2022] constructs a graph representation learning model based on gene pathways, ultimately producing prediction based on gene embeddings.

However, these methods have several limitations. **1. Lack of Biological Information:** They do not fully leverage the rich biological information available in genes, such as detailed gene descriptions and DNA sequences, which provide valuable insights into the functional roles of genes. **2. Failure to Capture Implicit Gene Interactions:** Relying on simplistic evaluation metrics, these methods fail to capture the complex nature of gene regulation, where genes interact in diverse and multifaceted ways, hindering the accurate modeling and prediction of gene behavior across various biological con-

texts. **3. Ignoring Biotype-Specific Gene Functions:** Coding genes (c-genes) are transcribed into mRNA and translated into proteins, while non-coding genes (nc-genes) are transcribed into RNA but do not produce proteins. Biotype refers to the classification of genes into coding and non-coding categories based on their functional roles, with the functional dimensions of genes varying across biotypes [Dykes and Emanueli, 2017; Böhmdorfer and Wierzbicki, 2015; Magistri *et al.*, 2012; Pankiewicz *et al.*, 2021]. These methods overlook these differences, limiting their ability to model gene interactions effectively.

To address these issues, we propose **GRAPE** (Heterogeneous **G**raph **R**epresentation Le**A**rning for Genetic **PE**rturbation) to predict genetic perturbation, which utilizes pre-trained large language model and DNA sequence model for gene feature initialization. Furthermore, **GRAPE** constructs a GRN as heterogeneous graph to capture the distinct roles of different biotypes and automatically learn the interactions between genes via graph structure learning. Experiments on publicly available datasets show that **GRAPE** outperforms existing methods.

The main contributions of our work are as follows:

- **Perspective:** We introduce **GRAPE**, the first model to incorporate cellular gene biotype information (coding and non-coding genes), to model gene regulatory networks as heterogeneous graphs. This approach captures the distinct functional roles of genes from different biotypes through heterogeneous graph neural networks.
- **Algorithm:** We initialize gene representations by leveraging pre-trained models [Kenton and Toutanova, 2019; Nguyen *et al.*, 2024] to extract multi-modal features of genes (textual descriptions and DNA sequences), which are then semantically aligned and fused to generate richer and more comprehensive representations. Besides, in constructing the gene interaction network, we propose a novel graph structure learning (GSL) method to capture implicit gene relationships.
- **Experiments:** We demonstrate the superiority of **GRAPE** over existing methods on publicly datasets, using a range of comprehensive evaluation metrics.

2 Related Work

2.1 Graph Structure Learning (GSL) and Heterogeneoous Graph Neural Network (HGNN)

GSL optimizes the graph's structure to enhance information flow, improve node representations.[Franceschi *et al.*, 2019] jointly learn the graph structure and GCN parameters; SLAPS ([Fatemi *et al.*, 2021]) uses self-supervision to better infer graph structures for GNNs; HGSL([Zhao *et al.*, 2021]) generating relation subgraphs based on feature similarity, propagation, and semantics. HGNNs learn relationships between different node types and generate node representations. HetGNN [Zhang *et al.*, 2019] uses a random walk strategy and two aggregation modules to generate embeddings; HAN [Wang *et al.*, 2019] uses hierarchical attention mechanisms at both node and semantic levels; SeHGNN [Shi, 2022] reduces complexity by using a lightweight mean aggregator and avoids excessive attention mechanisms. Despite their individual advantages, there remains room for improvement in capturing more complex and implicit relationships within heterogeneous graphs.

2.2 Genetic Perturbation Model

Gene perturbation is a essential direction in single-cell perturbation research. The Variational Autoencoder (VAE) framework and Graph Neural Networks (GNNs) have emerged as the dominant approaches in this filed of study. scGen [Lotfollahi *et al.*, 2019] is the first to use deep learning methods to model single-cell perturbations, utilizing variational autoencoders to capture cellular responses to perturbations in single-cell gene expression; GEARS [Roohani *et al.*, 2022] introduces ontology modality to model gene-gene interaction relationships; CellOracle [Bunne *et al.*, 2023] leverages gene-regulatory networks from single-cell multi-omics data to simulate transcription factor perturbations; SAMS-VAE [Bereket and Karaletsos, 2024] incorporates sparse global latent variables to enhance model interpretability; GraphVCI [Wu *et al.*, 2022] propose a novel graph-based Bayesian causal inference framework to predict gene expression responses. These methods have several limitations. They fail to fully leverage available biological information, which provide valuable insights into gene functions. Moreover, relying on simplistic evaluation metrics, they cannot capture the complex nature of gene regulation, hindering accurate predictions. Additionally, overlooking the functional differences between genes of different biotypes hinders their ability to effectively model gene interactions.

3 Preliminaries

In this section, we expound some basic concepts and give some definitions of our model.

Definition 1. Heterogeneous Graph We introduce an undirected heterogeneous graph $G = (N, R, X)$, where N is node set, along with node type mapping function $\varphi : N \rightarrow T = \{\text{c-gene}, \text{nc-gene}\}$, R is edge set, along with edge type mapping function $\phi : R \rightarrow E = \{r_{cc}, r_{cn}, r_{nn}\}$, X is node feature matrix.

Definition 2. Model Input and Output The model takes as input gene textual descriptions and DNA sequence information, along with the perturbation conditions. As illustrated in Fig.1, the task of our model is to predict the change in single-cell expression levels under a given perturbation condition (e.g., knockdown of gene k).

4 Method

In this section, we introduce the proposed model GRAPE. Figure.2.A illustrates the workflow for initializing gene features using sequence and textual descriptions by DNA sequence model [Nguyen *et al.*, 2024] nad LLM [Kenton and Toutanova, 2019], respectively. Figure.2.B presents the methodology for constructing gene interaction networks and performing heterogeneous graph representation learning.

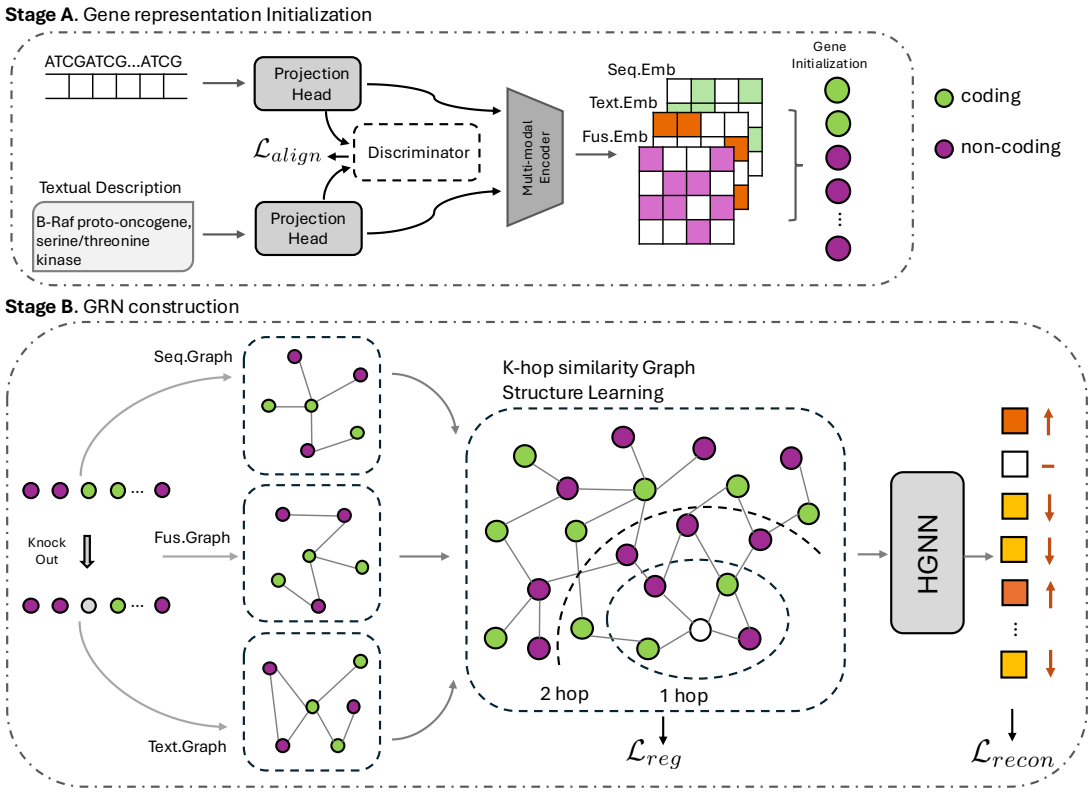

Figure 2: Overview of the **GRAPE**.

4.1 Gene Representation Initialization

Existing methods often initialize gene representations based on gene indices, which lack biological prior information and interpretability, potentially increasing the cost of model convergence. Therefore, we propose multi-modal biological prior information for representation initialization, enabling a more effective capture of the intrinsic gene characteristics. Building upon the pre-trained models [Nguyen *et al.*, 2024; Kenton and Toutanova, 2019], we align gene sequences with their textual descriptions to effectively integrate and capture the shared semantic information between them.

Formally, we extract DNA sequence features using [Nguyen *et al.*, 2024] and textual description features using [Kenton and Toutanova, 2019], resulting in the multi-modal features matrix $X_m = [x_m^1, x_m^2, ..., x_m^N] \in R^{N \times d_m}$, where $m \in \{\texttt{seq}, \texttt{text}\}$ denotes either the sequence or the textual description modality, with a dimensionality of d_m. Our goal is to ensure that the semantic integrity of each modality is preserved while achieving a unified feature representation $Z_m = f(X_m) = [z_m^1, z_m^2, ..., z_m^N]^T \in R^{N \times q_m}$ after the alignment, where f represents the transformation process.

Specifically, we aim to align the features extracted from different modalities for the same gene, while ensuring that cross-modal features from different genes are pushed apart. We achieve this using contrastive learning [Oord *et al.*, 2018], and the corresponding formula is as follows:

$$\mathcal{L}_{align} = -\sum_{i=1}^{N} \frac{exp^{(D(z_S^i, z_D^i)/\mathcal{T})}}{\sum_{j=1}^{N} exp^{(D(z_S^i, z_D^j)//\mathcal{T})}} \tag{1}$$

where D is the discriminator function that estimates the cosine similarity between the features after transformation. This indirect contrastive learning approach helps mitigate the disruption of the original semantics of each modality during alignment. After modality alignment, we obtain the multi-modal fusion features through a multi-layer perceptron (MLP) Ψ:

$$Z_{\texttt{fus}} = \Psi(Z_{\texttt{seq}} || Z_{\texttt{text}}) \tag{2}$$

4.2 Gene Regulation Network Construction

Based on the multi-modal features previously extracted, we can construct a graph for each modality m, where $m \in M = \{\texttt{seq}, \texttt{text}, \texttt{fus}\}$, according to the similarity of the characteristics between genes, thus revealing gene associations from different perspectives. Specifically, each gene is treated as a node in the graph, with multiple modality-specific representations. The edge between node z_m^i and node z_m^j is obtained by:

$$G_m[i,j] = \begin{cases} \mathbb{S}_m(z_m^i, z_m^j), & \text{if } \mathbb{S}_m(z_m^i, z_m^j) \geq \epsilon^m \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where ϵ^m is used to control the sparsity of the graph corresponding to the m modality. $\mathbb{S}_m$ is a multi-head similarity

module that computes multiple similarity scores using different heads with independent parameters. The definition is as follow:

$$\mathbb{S}_m(z_m^i, z_m^j) = \frac{1}{T}\sum_{t=1}^{T} cos(z_m^i W_m^t, z_m^j W_m^t) \tag{4}$$

where W_m^t denotes the learnable parameters of the t-th similarity head in the m modality. After the above graph construction steps and edge filtering, we obtain the gene interaction network structures G_m from the perspective of each modality. To aggregate the Gene Regulatory Networks (GRNs) G_{agg} from different modalities, we can express the aggregation as the mean across all modalities:

$$G_{agg} = \frac{1}{|M|}\sum_{m\in M} G_m \tag{5}$$

However, graphs constructed using simple similarity-based methods tend to capture only coarse-grained neighbor relationships, making it challenging to comprehensively model the complex and latent interactions between nodes. In real-world scenarios, genes often exhibit latent interactions across multiple scales, necessitating a more flexible and holistic mechanism to model their associations. To address this, we introduce the K-hop GCN aggregation mechanism, which aggregates the results from different K-hop convolutional layers to integrate multi-scale structural information, enabling a more effective representation of the intricate gene interactions. In practice, we aggregate the node relationships across different hop distances as follows:

$$H^{(l+1)} = \mathrm{GCN}(G_{agg}, H^{(l)}) \tag{6}$$

$$\mathbb{G}^k[i,j] = \mathbb{S}^k(H_i^k, H_j^k) \tag{7}$$

when $l = 0$, the feature matrix $H^{(l)}$ is typically initialized as the initial node fusion representations $Z_{\texttt{fus}}$ (Eq. (2)), with GCN serving as the backbone for feature aggregation [Kipf and Welling, 2016].

Then the K-hop structure is defined as:

$$\mathbb{G} = \Phi(\mathbb{G}^1, \mathbb{G}^2, \ldots, \mathbb{G}^k) \tag{8}$$

$$\mathbb{E} = \{(i,j)|\mathbb{G}(i,j) \geq \epsilon\} \tag{9}$$

where Φ denotes a channel attention layer. The edge set $\mathbb{E}$ are selected to capture the most significant relationships between nodes. To maintain the structural consistency between the original graph G_{agg} and the k-hop graph $\mathbb{G}$, the regularization (Eq. (10)) is proposed, ensuring that the model does not deviate from the local structural information of the original graph while learning higher-order graph structures.

$$\mathcal{L}_{reg} = ||\mathbb{G} - G_{agg}||_2 \tag{10}$$

4.3 Heterogeneous Graph Representation Learning

As mentioned earlier, c-genes and nc-genes regulate the cellular state at different level [Dykes and Emanueli, 2017; Böhmdorfer and Wierzbicki, 2015; Magistri *et al.*, 2012; Pankiewicz *et al.*, 2021]. Therefore, it is necessary to construct a gene interaction network using a heterogeneous graph based on the gene biotype. To simplify the problem, we regard the heterogeneous graph as an undirected graph.

Based on the graph construction before, we map the nodes using φ and the edges using ϕ as defined in Definition.1. Given the different types of associations between genes, we design a multi-head heterogeneous graph attention network (HGAT) to aggregate these features:

$$P^{(l+1)} = \frac{1}{|E|}\sum_{r\in E}(\frac{1}{H}\sum_{h=1}^{H}\mathrm{GAT}_r^h(\{e|e\in\mathbb{E}\wedge\phi(e)=r\}, P^l)) \tag{11}$$

where H represents number of head, and $P^{(0)} = Z_{\texttt{fus}}$ is initialized as the fusion gene embeddings with perturbation information, which will be elaborated on in the following section. The GAT used for feature aggregation can be found in [Brody *et al.*, 2021; Veličković *et al.*, 2017].This framework enables the model to actively learn the weights of different types of edges by leveraging the attention mechanism. In the context of a heterogeneous graph, where various edge types represent distinct gene interactions, the model can assign independent parameters to each type of relationship. By doing so, it allows the model to prioritize the most relevant interactions for each specific task, rather than treating all the edges equally.

4.4 Genetic Perturbation Prediction

During training process, **GRAPE** initializes gene embeddings Z_m and constructs gene regulation network $\mathbb{G}$ at first. Given a gene perturbation condition c, which biologically corresponds to the knockout of a specific gene , we incorporate this perturbation information into the original input in multi-head HGAT to simulate its effects. Specifically, this involves masking the representations of the corresponding knockout genes in $Z_{\texttt{fus}}$, ultimately resulting in the final output $P_c = \{p_c^1, p_c^1, \ldots, p_c^N\} \in R^{N\times d}$ of Eq. (11).

Considering that each gene has its own unique expression pattern [Roohani *et al.*, 2022], we apply an independent linear transformation to each gene to obtain the final output under perturbation c:

$$\hat{g}_c^i = W_i \odot p_c^i + b_i \in R \tag{12}$$

where $\odot$ denotes Hadamard Product. Ultimately, the prediction of gene expression can be obtained by:

$$\tilde{g}_c = \hat{g}_c + ctrl \tag{13}$$

where $ctrl$ represents the expectation of gene expression levels from the control group cells (non-knockout).

To optimize the **GRAPE**'s performance, we compute the mean squared error (MSE) between the predicted and true gene expression values under condition c:

$$\mathcal{L}_{mse} = \frac{1}{N}\sum_{i=1}^{N}(g_c^i - \tilde{g}_c^i)^2 \tag{14}$$

where $g_c \in R^{N\times 1}$ represents the observed gene expression value sampled under perturbation condition c. Moreover, to control the direction of gene expression changes, we use the Huber loss function, which combines MSE's sensitivity to

		RMSE(↓)	DAC(↑)			PCC_delta(↑)		
			All	DE20	DE40	All	DE20	DE40
Adamson	**GRAPE (Ours)**	**0.0754** $\pm$ 0.0006	**0.6948** $\pm$ 0.0031	**0.8863** $\pm$ 0.0029	**0.9363** $\pm$ 0.0023	**0.7461** $\pm$ 0.0070	**0.7828** $\pm$ 0.0124	**0.8010** $\pm$ 0.0110
	GEARS	0.0812 $\pm$ 0.0008	0.6557 $\pm$ 0.0024	0.7881 $\pm$ 0.0038	0.7876 $\pm$ 0.0027	0.5514 $\pm$ 0.0064	0.6109 $\pm$ 0.0059	0.6429 $\pm$ 0.0084
	GraphVCI	0.4031 $\pm$ 0.0064	0.5315 $\pm$ 0.0184	0.6114 $\pm$ 0.0233	0.4807 $\pm$ 0.0112	0.1647 $\pm$ 0.0201	0.4138 $\pm$ 0.0306	0.4405 $\pm$ 0.0291
	sams-VAE	0.6511 $\pm$ 0.0113	0.4241 $\pm$ 0.0218	0.6659 $\pm$ 0.0351	0.5625 $\pm$ 0.0377	0.1441 $\pm$ 0.0195	0.4155 $\pm$ 0.0432	0.4313 $\pm$ 0.0319
Norman	**GRAPE (Ours)**	**0.0482** $\pm$ 0.0003	**0.7583** $\pm$ 0.0012	**0.7352** $\pm$ 0.0014	**0.7556** $\pm$ 0.0011	**0.4742** $\pm$ 0.0052	**0.5452** $\pm$ 0.0152	**0.5581** $\pm$ 0.0107
	GEARS	0.0702 $\pm$ 0.0007	0.5201 $\pm$ 0.0050	0.6319 $\pm$ 0.0057	0.6361 $\pm$ 0.0047	0.4277 $\pm$ 0.0036	0.5012 $\pm$ 0.0044	0.5417 $\pm$ 0.0043
	GraphVCI	0.5572 $\pm$ 0.0187	0.2795 $\pm$ 0.0127	0.3395 $\pm$ 0.0183	0.3664 $\pm$ 0.0174	0.1202 $\pm$ 0.0147	0.3205 $\pm$ 0.0340	0.2423 $\pm$ 0.0286
	sams-VAE	0.5025 $\pm$ 0.0170	0.3742 $\pm$ 0.0209	0.4963 $\pm$ 0.0402	0.5361 $\pm$ 0.0343	0.1248 $\pm$ 0.0117	0.3066 $\pm$ 0.0258	0.2490 $\pm$ 0.0274

Table 1: Performance comparison across different models on Adamson and Norman datasets, evaluated by RMSE, Direction Accuracy Coefficient (DAC), and Pearson Correlation Coefficient (PCC) metrics.

small errors with MAE's robustness to large errors, reducing the impact of outliers and ensuring more stable training in noisy data.

$$\mathcal{L}_{dir} = \frac{1}{N}\sum_{i=1}^{N} Huber(tanh(g_i - ctrl_i), tanh(\tilde{g}_i - ctrl_i)) \tag{15}$$

$$Huber(a,b) = \begin{cases} \frac{1}{2}(a-b)^2, & \text{if } |a-b| \le \delta \\ \delta(|a-b| - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \tag{16}$$

The reconstruction objective of the model for gene expression values is represented as:

$$\mathcal{L}_{recon} = \mathcal{L}_{mse} + \mathcal{L}_{dir} \tag{17}$$

4.5 Training Steps

During model training, we optimize the above objectives simultaneously, and the final training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{reg} + \mathcal{L}_{align} \tag{18}$$

here, $\mathcal{L}_{align}$is specifically optimized during the pre-training phase before the main model training.

4.6 Analysis of Time Complexity

To simplify the analysis, we assume the feature dimension has no impact and treat computations as $O(1)$. The computational cost of the model consists of four main components: multi-head attention $O(H_1 \cdot N^2)$, K-hop similarity ($O(K \cdot m)$), channel attention $O(N^2)$, and feature extractions from multi-head HGAT $O(L \cdot H_2 \cdot (|\mathbb{E}| + N))$. Here, N denotes the number of genes, H_1 represents head number in Eq. (4), m stands for the number of edges in the result of Eq. (3), with L and H_2 referring to the layer depth and the number of attention heads in Eq. (11), respectively.

5 Results and Discussion

To demonstrate the effectiveness of our method, we conduct extensive experiments on public available datasets. Adamson dataset [Adamson *et al.*, 2016] contains data from 87 types of single-gene perturbations, with a total of $68,603$ single-cell RNA sequencing results. Norman dataset [Norman *et al.*, 2019] includes results from both single-gene and double-gene perturbations. We only select experimental data for single-gene perturbations, with over 100 types of single-gene perturbations in total. In each dataset, hundreds of single-cell RNA sequencing data can be obtained for each type of genetic perturbation.

5.1 Experiment Settings

Gene text descriptions, sequences, and biotypes can all be accessed through the API provided by Ensemble (Click here to visit the tutorial). In the training process of **GRAPE**, we randomly select 70% of gene perturbation conditions for the training set and use the remaining for testing, then select single-cell data according to these conditions. The number of heads in both the multi-head similarity module (Eq. 4) and the multi-head HGAT module (Eq. 11) is set to 4. The model is trained using the AdamW [Loshchilov, 2017] optimizer with a learning rate of 0.001 and a batch size of 64. For evaluation, we adopt multiple metrics, including Direction Accuracy Coefficient (DAC), Root Mean Square Error (RMSE), and Pearson Correlation Coefficient delta (PCC_delta), to assess the model's prediction accuracy and biological relevance (refer to the github for further details). All our method and its competitors are conducted using four Nvidia A100 GPUs.

5.2 GRAPE outperform existing methods

Table 1 demonstrates that **GRAPE** significantly outperforms existing gene perturbation methods ([Roohani *et al.*, 2022; Wu *et al.*, 2022; Bereket and Karaletsos, 2024]) across all evaluated metrics. While Root Mean Square Error (RMSE) measures the overall prediction error, DAC and PCC_delta provide deeper biological insights. DAC evaluates the consistency between predicted and true directions of gene expression changes, which is crucial for understanding gene regulatory effects and ensuring accurate biological interpretations, as incorrect directional predictions can lead to misleading conclusions. PCC_delta, on the other hand, quantifies the consistency in the change of gene expression values between the predicted and true values under perturbation conditions. Together, these metrics offer a comprehensive assessment of model accuracy and relevance in biological contexts.

GRAPE showcases superior performance compared to GEARS [Roohani *et al.*, 2022], primarily because it goes beyond GEARS' simplified gene interaction network by accounting for the diverse relationships across different biotype genes. **GRAPE** actively uncovers latent connections between genes, providing a deeper understanding of gene regulatory dynamics. In contrast, generative models [Wu *et al.*, 2022; Bereket and Karaletsos, 2024] perform poorly due to their inability to effectively capture the semantic meaning of gene perturbation conditions. This limitation hinders their ability to model biologically relevant features, resulting in less

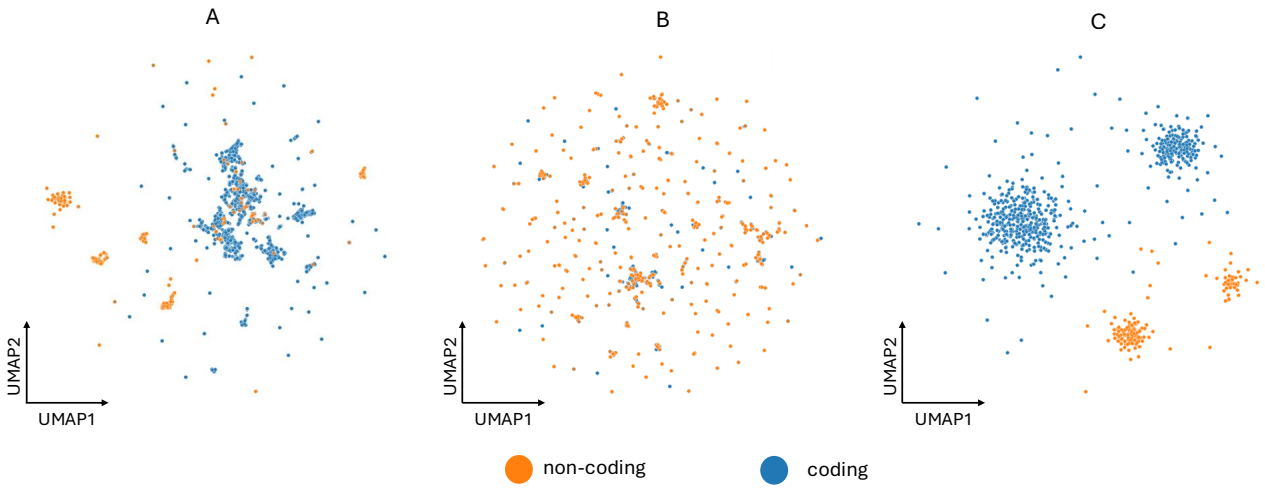

Figure 3: The UMAP visualization of gene representations. Subfigures A and B show the gene text modality representations and gene sequence modality representations extracted by the pre-trained LLM and DNA sequence model, respectively. Subfigure C illustrates the representations extracted by **GRAPE** after training.

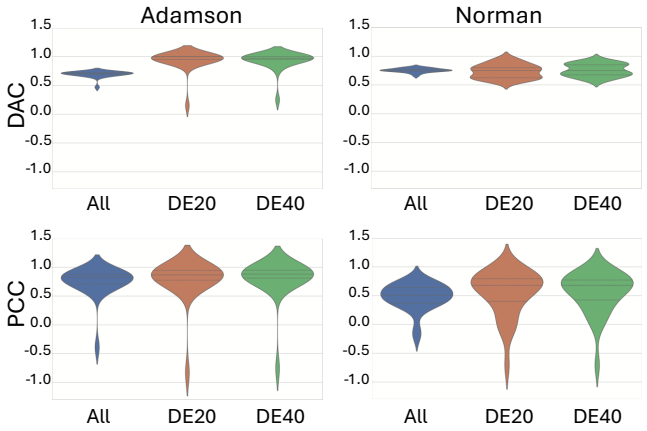

Figure 4: The picture shows the distribution of predicted values across samples.

accurate predictions. Moreover, **GRAPE** leverages multi-modal information to initialize gene representations, providing a more comprehensive and biologically meaningful characterization of genes.

To further evaluate the model's performance, we visualize the distribution of DAC and PCC values across samples (Fig.4). These plots provide a detailed view of the prediction accuracy at the sample level, highlighting the variability and consistency of the model's predictions. By focusing on these key metrics, the violin plots offer an intuitive representation of how well the model captures the directionality and correlation of gene expression changes under different perturbations.

5.3 GRAPE Enhances Gene Representation with Biotype-Aware Heterogeneous Graphs

To evaluate **GRAPE** leveraging biotype knowledge, we first visualize gene features extracted from pre-trained LLM [Kenton and Toutanova, 2019] and DNA sequence model [Nguyen *et al.*, 2024], focusing on two modalities: gene textual descriptions (Fig.3.A) and gene sequences (Fig.3.B). For this, we employe Uniform Manifold Approximation and Projection (UMAP) to visualize the high-dimensional gene features in a lower-dimensional space. The visualizations of Fig.3.A and Fig.3.B reveal that the clustering of gene representations did not align with the biotype labels. Despite the inherent complexity and richness of these features, the unsupervised clustering approach failed to exhibit any clear, biotype-specific grouping. This suggests that the pre-trained models, although capable of extracting meaningful features, does not sufficiently capture the underlying biological relevance of gene biotypes on its own.

In contrast, after training the **GRAPE**, which incorporates biotype-aware heterogeneous graphs into the representation learning process (Eq. (11)), the extracted gene features exhibited a distinct biotype-based clustering pattern (Fig.3.C). The representations of genes that belong to the same biotype formed tight and well-separated clusters, indicating that the model successfully learned to embed gene biotype information into the feature space. This outcome highlights the efficacy of **GRAPE** in improving gene representation by dynamically adjusting the gene regulatory network and then extracting features through HGAT that incorporates biotype-aware information.

Constructing a heterogeneous graph network effectively represents the complex relationships between genes and simulates interactions between different biotype genes, thereby enhancing gene representation through HGAT. On the other hand, when biotype is directly concatenated as a feature to

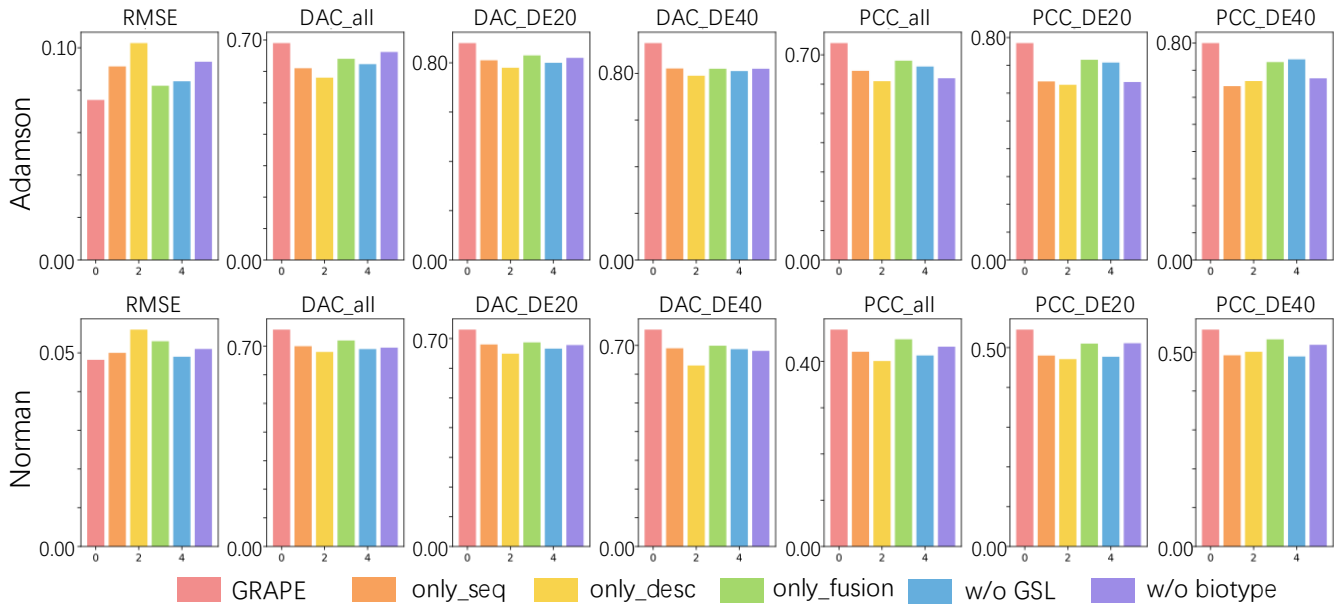

Figure 5: Ablation Study Results.

the gene representation, the model may fail to fully explore the underlying connections between genes based on their biotype. This approach only provides a singular biotype feature, lacking interactions with other gene attributes and fails to leverage the diversity of nodes and edges in the GRN. As a result, the model may not capture the biological significance of genes in their multidimensional relationships. Additionally, directly incorporating the biotype label as a feature could lead to over-reliance on the biotype information, potentially neglecting other crucial biological features, which could in turn reduce the model's generalization ability.

5.4 Ablation Study

To further evaluate the effectiveness of **GRAPE**, we compare it with the following methods through an ablation study. 1)**only_seq**: Excludes the textual description modality features, using only the DNA sequence features in the model. 2)**only_desc**: Excludes the DNA sequence modality features, using only the textual description features in the model. 3)**only_fusion**: Uses the fusion modality, combining both desc and seq features into a unified representation, excluding the individual modalities. 4)**w/o GSL**: Excludes the K-hop similarity graph structure learning process. 5)**w/o biotype**: Ignores biotype information, converting the GRN into a homogeneous graph, and uses a homogeneous GNN instead of HGNN. The results are shown in Figure.5.

The experimental results show that **GRAPE** outperforms methods relying on individual modalities. This indicates that integrating multiple modalities allows the model to capture a broader range of information, leveraging complementary insights from each modality for improved performance and more robust feature extraction. Additionally, the performance of **no_GSL** is less than ideal, highlighting the significant role that graph structure learning plays in capturing the hidden relationships between genes. Without graph structure learning, the model fails to fully leverage the intricate connections and dependencies inherent in the gene interactions, which are essential for accurate representation and prediction.

More importantly, the results clearly demonstrate the effectiveness of introducing biotype information. By leveraging HGNN, the model is able to capture and simulate the complex, multi-dimensional regulatory relationships between genes. This approach not only allows for a more nuanced understanding of gene interactions across different biological categories, but also enhances the model's ability to predict gene behavior with greater accuracy.

6 Conclusion

In this work, we present **GRAPE**, a heterogeneous graph neural network (HGNN) that integrates gene biotype information for predicting genetic perturbations. By leveraging pre-trained large language model (LLM) and DNA sequence model to extract features from gene textual descriptions and DNA sequences, we initialize gene representations in a way that captures both sequence-based and textual semantic information. Furthermore, for the first time, we incorporate gene biotype data to simulate the distinct roles of different biotypes in regulating cellular processes. Through graph structure learning (GSL), **GRAPE** dynamically refines the gene regulatory network (GRN), effectively capturing implicit gene relationships. Our method outperforms existing approaches on publicly available datasets, achieving state-of-the-art performance in predicting genetic perturbations. If sufficient data is available, it can enable a zero-shot model on top of **GRAPE**, automatically constructing gene regulatory networks (GRN) and inferring gene interactions to predict genetic perturbation results without human annotations.

Acknowledgments

This work was supported by National Science and Technology Major Project (No. 2022ZD0115101), National Natural Science Foundation of China Project (No. 623B2086), National Natural Science Foundation of China Project (No. U21A20427), Project (No. WU2022A009) from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University, Project (No. WU2023C019) from the Westlake University Industries of the Future Research Funding.

Contribution Statement

Changxi Chi and Jun Xia contributed equally to this work. Stan Z. Li supervised the project.

References

[Adamson *et al.*, 2016] Britt Adamson, Thomas M Norman, Marco Jost, Min Y Cho, James K Nuñez, Yuwen Chen, Jacqueline E Villalta, Luke A Gilbert, Max A Horlbeck, Marco Y Hein, et al. A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882, 2016.

[Barrangou and Doudna, 2016] Rodolphe Barrangou and Jennifer A Doudna. Applications of crispr technologies in research and beyond. *Nature biotechnology*, 34(9):933–941, 2016.

[Bereket and Karaletsos, 2024] Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. *Advances in Neural Information Processing Systems*, 36, 2024.

[Böhmdorfer and Wierzbicki, 2015] Gudrun Böhmdorfer and Andrzej T Wierzbicki. Control of chromatin structure by long nonco ding rna. *Trends in cell biology*, 25(10):623–632, 2015.

[Brody *et al.*, 2021] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

[Bunne *et al.*, 2023] Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768, 2023.

[Dykes and Emanueli, 2017] Iain M Dykes and Costanza Emanueli. Transcriptional and post-transcriptional gene regulation by long non-coding rna. *Genomics, Proteomics and Bioinformatics*, 15(3):177–186, 2017.

[Fatemi *et al.*, 2021] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. Slaps: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems*, 34:22667–22681, 2021.

[Franceschi *et al.*, 2019] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *International conference on machine learning*, pages 1972–1982. PMLR, 2019.

[Kenton and Toutanova, 2019] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota, 2019.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Lino *et al.*, 2018] Christopher A Lino, Jason C Harper, James P Carney, and Jerilyn A Timlin. Delivering crispr: a review of the challenges and approaches. *Drug delivery*, 25(1):1234–1257, 2018.

[Loshchilov, 2017] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[Lotfollahi *et al.*, 2019] Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 16(8):715–721, 2019.

[Magistri *et al.*, 2012] Marco Magistri, Mohammad Ali Faghihi, Georges St Laurent, and Claes Wahlestedt. Regulation of chromatin structure by long noncoding rnas: focus on natural antisense transcripts. *Trends in genetics*, 28(8):389–396, 2012.

[Nguyen *et al.*, 2024] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36, 2024.

[Norman *et al.*, 2019] Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Pankiewicz *et al.*, 2021] Katarzyna Pankiewicz, Piotr Laudański, and Tadeusz Issat. The role of noncoding rna in the pathophysiology and treatment of premature ovarian insufficiency. *International journal of molecular sciences*, 22(17):9336, 2021.

[Roohani *et al.*, 2022] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Gears: Predicting transcriptional outcomes of novel multi-gene perturbations. *BioRxiv*, pages 2022–07, 2022.

[Shi, 2022] Chuan Shi. Heterogeneous graph neural networks. *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 351–369, 2022.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019.

[Wu *et al.*, 2022] Yulun Wu, Robert A Barton, Zichen Wang, Vassilis N Ioannidis, Carlo De Donno, Layne C Price, Luis F Voloch, and George Karypis. Predicting cellular responses with variational causal inference and refined relational information. *arXiv preprint arXiv:2210.00116*, 2022.

[Zhang *et al.*, 2019] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 793–803, 2019.

[Zhao *et al.*, 2021] Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song, and Yanfang Ye. Heterogeneous graph structure learning for graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4697–4705, 2021.