scGraPhT: Merging Transformers and Graph Neural Networks for Single-Cell Annotation

Emirhan Koç, *Student Member, IEEE*, Emre Kulkul, Gülara Kaynar, Tolga Çukur, *Senior Member, IEEE*, Murat Acar, and Aykut Koç, *Senior Member, IEEE*

Abstract—**The invention of single-cell RNA sequencing (scRNA-seq) has enabled transcriptomic examination of cells on an individual basis, uncovering cell-to-cell phenotypic heterogeneity within isogenic cell populations. Inevitably, cell type annotation has emerged as a fundamental, albeit challenging task in scRNA-seq data analysis, which involves identifying and characterizing cells based on their unique molecular profiles. Recently, deep learning techniques with their data-driven priors have shown significant promise in this task. On the one hand, task-agnostic transformers pre-trained on large-scale biological databases capture generalizable representations but cannot characterize intricate relationships between genes and cells. Contrarily, task-specific graph neural networks (GNNs) trained on target datasets can characterize entity relationships, but they can suffer from poor generalizability. Furthermore, existing GNNs focus on either homogeneous or heterogeneous relationships, failing to capture the full cellular complexity. Here, we propose scGraPhT, a unified transformer–graph model that combines pre-trained transformer embeddings of scRNA-seq data with a multilayer GNN to capture cell-cell, cell-gene, and gene-gene relationships. Different from previous GNNs, scGraPhT examines both homogeneous and heterogeneous relationships through subgraph layers to offer a more comprehensive assessment. Since the graph construction uses transformer-derived embeddings, scGraPhT does not require costly training procedures and can also be adapted to leverage any transformer-based single-cell annotation method, such as scGPT or scBERT. Demonstrations on three scRNA-seq benchmark datasets indicate that scGraPhT outperforms state-of-the-art annotation methods without compromising efficiency. Utilizing Grad-CAM, we demonstrate how the GNN and transformer components complement each other to enhance performance. We share our source codes and datasets for reproducibility.**

***Index Terms*—Graph neural networks (GNNs), transformers, foundation models, scRNA sequencing, cell type annotation.**

I. Introduction

SINGLE-CELL RNA-sequencing (scRNA-seq) is a state-of-the-art technique that allows the examination of gene expression profiles at the individual cell level, which paves the way for significant discoveries in research fields such as pathology, immunology, cancer, genomics, and regenerative medicine [1], [2], [3], [4], [5], [6], [7], [8], [9]. Contrary to traditional bulk sequencing, which collects RNA transcript information from a large population of cells and produces an average read-out, scRNA-seq enables the transcriptomic analysis of individual cells [10], [11]. In this regard, scRNA-seq offers a higher resolution view of mRNA composition in cells, facilitating characterization of phenotypic heterogeneity across single cells in populations and within tissues, tracking of cellular differentiation, and unraveling the complexities of cellular responses to environmental stimuli [12], [13], [14], [15].

Cell type annotation is a systematic process of identifying and characterizing cells within a heterogeneous biological sample based on their unique molecular and phenotypic profiles to elucidate the type of each cell. It is a fundamental step in scRNA-seq data analysis and constitutes the basis of further downstream analyses [16], [17], [18], [19]. In its simplest form, the task of annotating cells consists of two steps. First, cells are grouped into clusters based on their gene expression profiles using unsupervised clustering [20], [21], [22]. The resulting clusters typically consist of cells with similar gene expression profiles and common or closely related marker genes -those that are distinctly and highly expressed in a specific cluster compared to others [17], [23], [24], [25]. Thus, the clustering step enables estimation of the number of distinct cell types [26], [27]. Afterward, the identified marker gene candidates are manually inspected or cross-checked against literature and cell marker databases to assign cell type labels to the detected clusters [28], [29], [30]. However, performing manual annotation is challenging due to the lack of comprehensive information on marker genes for specific cell subtypes [17], [18]. As such, despite their broad availability, utilization of such unsupervised clustering-based methods can be labor-intensive.

Received 27 August 2024; revised 2 March 2025; accepted 20 May 2025. Date of publication 26 May 2025; date of current version 9 June 2025. The work of Tolga Çukur was supported in part by BAGEP 2017 and in part by GEBIP 2015. The work of Murat Acar is supported by TUBITAK 1001 under Grant 123Z270. The work of Aykut Koç is supported by BAGEP 2023. This work was supported by Turkcell İletişim Hizmetleri A.Ş. through the 5G and Beyond Joint Graduate Support Program coordinated by Information and Communication Technologies Authority and TUBITAK 1001 under Grant 124E179. The associate editor coordinating the review of this article and approving it for publication was Dorina Thanou. Tolga Çukur, Murat Acar, and Aykut Koç have co-supervised this work. *(Corresponding authors: Tolga Çukur; Murat Acar; Aykut Koç).*

Emirhan Koç, Emre Kulkul, Tolga Çukur, and Aykut Koç are with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Türkiye, and also with National Magnetic Resonance Research Center (UMRAM), 06800 Ankara, Türkiye (e-mail: cukur@ee.bilkent.edu.tr; aykut.koc@bilkent.edu.tr).

Gülara Kaynar is with the Department of Computer Engineering, Koç University, 34450 Istanbul, Türkiye.

Murat Acar is with the Department of Basic Medical Sciences, School of Medicine, Koç University, 34450 Istanbul, Türkiye, and also with the Koç University Research Center for Translational Medicine, School of Medicine, Koç University, 34450 Istanbul, Türkiye (e-mail: macar@ku.edu.tr).

Available at https://github.com/koc-lab/scGraPhT

This article has supplementary downloadable material available at https://doi.org/10.1109/TSIPN.2025.3573591, provided by the authors.

Digital Object Identifier 10.1109/TSIPN.2025.3573591

This labor-intensive nature poses significant challenges when applying these methodologies in large-scale analyses involving large populations or multiple experimental sessions.

With the unprecedented expansion of scRNA-seq atlases, including the Human Cell Atlas [31], [32], [33], automated annotation methods have been developed to simplify the cell annotation process [25]. These methods often rely on classification algorithms trained on reference datasets and employ transfer learning strategies for their application to new single-cell datasets [17]. For instance, ACTINN [34] utilizes a neural network model with three hidden layers, trained on reference datasets, to classify cell types in query datasets. Kiselev et al. [35] developed scmap, which projects cells from scRNA-seq data onto reference cell types using nearest neighbor classifiers. De et al. [36] proposed CHETAH that leverages a hierarchical classification approach where cells are iteratively classified at multiple levels of resolution; at each level, cells are grouped into progressively finer clusters, allowing for a detailed and accurate annotation of cell types. These earlier models are often trained on task-specific, compact datasets from a limited variety of tissues, constraining their ability to capture generalizable representations and struggle to scale well to larger datasets. It is worth indicating that these approaches constitute fundamental milestones and are crucial for understanding the rapid advancements in this field.

To address the challenge of producing generalizable representations, pre-trained transformer models originating from the domain of natural language processing (NLP) have recently been adopted for a variety of downstream applications in scRNA-seq analysis, including cell type annotation. These models are trained on extensive datasets, enabling them to capture universal, dense representations of entities such as cells or genes from scRNA-seq data [37]. Transformers are commonly subjected to self-supervised learning to build task-agnostic representations, making them effective for a diverse array of tasks, including cell-type annotation. Their ability to generalize across different datasets and tasks, coupled with fine-tuning on smaller labeled datasets, enhances annotation accuracy and reduces the need for manual curation [38]. Recently, transformer-based models, including TOSICA [39], scBERT [40], and scGPT [41] have been introduced in the scRNA-seq domain, and they have been extensively employed for cell type annotation [42], [43], [44], [45], [46], [47]. Note that existing transformer models take gene expression profiles as input, i.e., a cell-by-gene count matrix showing the relative abundance of specific genes for each cell. Similar to how words are taken as tokens in NLP models, genes are taken as tokens in scRNA-seq models. As such, existing transformers primarily encode gene-gene interactions via self-attention mechanisms. While this approach yields promising performance in annotation tasks, a significant limitation is that it does not consider broader relationships, such as cell-cell and cell-gene interactions, which can also provide valuable information for annotation.

Graph Neural Networks (GNNs) are inherently capable of capturing diverse relationships through their nodes and edges, making them well-suited for modeling gene-gene, cell-gene, and cell-cell interactions. In this context, GNNs emerge as a compelling method for cell type annotation, as they can more effectively capture both global and local relationships. GNNs can primarily be divided into two groups based on the types of entities represented on their nodes. The first group consists of homogeneous graphs, where all nodes represent the same type of entity, such as a social network graph where each node represents a person and edges represent interactions between them. The second group comprises heterogeneous graphs, where nodes represent different types of entities, as in the case of a graph where some nodes represent authors and others represent books, with edges indicating which authors contributed to the writing of which books. These examples can be applied across various fields where graph-based modeling approaches are relevant [48], [49], [50], [51], [52], [53], [54], [55], [56], [57]. Most of the existing GNN-based approaches for cell type annotation predominantly rely on homogeneous graph structures, where nodes represent either cells or genes, and the edges encode their interactions. For instance, methods like HNNVAT [58], sigGCN [59], scAGN [60], and [61] utilize graph nodes that represent either cell or gene features. On the other hand, scDeepSort [62] presents a heterogeneous cell-gene graph network where both cells and genes are represented as nodes.

While GNN-based methods demonstrated promising results, several challenges still prevent them from reaching their full potential. First, similar to conventional deep learning models, GNNs are trained on task-specific datasets, which limits their generalizability in node feature representation. Second, most existing methods rely on fixed graph representations, where subsequent graph layers replicate the structure of the preceding ones. Third, both homogeneous and heterogeneous graphs also exhibit inherent shortcomings on their own. For instance, homogeneous gene-gene graphs lack the ability to capture cell-level relationships. Even though homogeneous cell-cell graphs take into account the interactions between cells, they are not sufficient in capturing the gene-level interactions [63]. Despite integrating cell- and gene-level relationships, heterogeneous graphs may still struggle to fully capture the intricate and context-specific interactions between these levels, leading to potential gaps in biological representation. Therefore, more comprehensive GNN models that effectively capture both homogeneous and heterogeneous interactions in complex biological systems are direly needed.

In this study, we introduce scGraPhT, a transformer-based graph neural network model for cell type annotation designed to overcome the limitations of task-agnostic transformer models and task-specific GNN models. scGraPhT incorporates heterogeneous and homogeneous graphs in a unified architecture. It represents genes and cells as nodes and constructs subgraphs, such as homogeneous connections between either genes or cells, as well as heterogeneous connections between cells and genes. The subgraph layers construct pathways, which are designed as multi-layered subgraphs to capture a more diverse range of relationships. Inspired by the representative capabilities of transformer models, scGraPhT also utilizes a large-scale pre-trained foundation model and combines it with the graph's ability to represent globally intertwined relationships. We present a range of training schemes and graph construction strategies and evaluate their performance on three benchmark scRNA-seq datasets commonly used in the field for cell-type annotation. Our model demonstrates superior performance compared to the previously published state-of-the-art and conventional methods. We also provide the analysis of performance enhancements using Grad-CAM [64], a visual explainability technique that identifies the key features driving our model's predictive power.

Our main contributions can be summarized as follows. We propose a cell-type annotation framework that

- constructs homogeneous and heterogeneous subgraphs to better represent the transcriptomic interactions within and across cells,
- offers a flexible and adaptable graph construction strategy,
- merges subgraph layers to form pathways designed as multi-layered subgraphs that capture a broader array of relationships,
- does not require exhaustive pre-training stages and can leverage existing transformer-based foundation models as a starting point,
- outperforms existing works in cell type annotation performance on common benchmark datasets,
- provides insights into the complementary nature of GNN- and transformer-based models for performance improvement.

The rest of the manuscript is organized as follows. In Section II, the related works utilizing state-of-the-art models, including transformers and GNNs, are provided. In Section III, our proposed model scGraPhT is presented. In Section IV, the experimental procedures, datasets with their properties, model performances, and inference time comparison of the models are provided. Sections V and VI present a discussion of our findings, outline potential limitations, and propose possible solutions. Finally, we conclude our manuscript in Section VII.

II. Related Works

In this section, we present the existing literature in three subsections. First, we provide preliminary works on automated methods for cell type annotation within the scRNA-seq domain. Then, we cover the recent transformer-based models used for cell-type annotation with scRNA-seq data. In the last subsection, we focus on GNN-based methods for cell type annotation using scRNA-seq data.

A. *Conventional Cell Type Annotation Methods*

Prior to the emergence of sophisticated deep learning (DL) models, various tools were developed for annotating cells, including those based on probabilistic models, correlation analyses, and simple machine learning (ML) techniques. [65] is a robust framework for the probabilistic modeling of gene expression in single cells. Utilizing stochastic optimization and deep neural networks (DNN), they effectively consolidate information across similar cells and genes, accurately approximating the underlying expression distributions. [66] proposes to annotate cell types using correlation-based similarity measures by comparing individual cells in the test datasets to those in reference datasets. Examples of supervised ML algorithms for cell type annotation include [67] and [68], which are based on the Random Forest algorithm [69]. Similarly, [70] introduces a logistic regression framework that utilizes stochastic gradient descent for optimization. Additionally, some methods [71], [72] employ shallow multilayer perceptrons (MLPs) for cell-type annotation tasks. While the conventional approaches are computationally efficient and typically require fewer parameters, they are less effective at capturing and interpreting intricate relationships within the data, as demonstrated by transformer- and graph-based approaches. In particular, unlike transformers, they lack the ability to generate rich, transferable data representations that can be effectively utilized by other models. In contrast to traditional task-specific models, our proposed method, scGraPhT, leverages a large-scale transformer architecture pre-trained on a vast dataset in a task-agnostic manner. This approach enables the model to generate generalizable representations, allowing it to adapt effectively to the cell-type annotation task with newly introduced datasets.

B. *Transformer-Based Cell Type Annotation Methods*

Among the recent transformer-based models, TOSICA is a supervised cell type annotation model trained to capture cell embeddings for pancreatic and brain cells in humans and mice [39]. Inspired by the BERT pre-training and fine-tuning paradigm [73], scBERT is a pioneering model that is first pre-trained on vast amounts of unlabeled scRNA-seq data from various human tissues and then fine-tuned for cell type annotation task. scBERT leverages information from gene embeddings for classification [40]. Another large-scale transformer-based pre-trained model is CellPLM, which employs cell embeddings for the cell type annotation task. It is trained on extensive scRNA-seq datasets as well as spatially-resolved transcriptomic (SRT) data [74]. Finally, scGPT is a foundation model pre-trained on scRNA-seq datasets collected from diverse human cell types and designed for single-cell transcriptomic analysis using a generative pre-training approach similar to scGPT in natural language generation (NLG) [75]. It utilizes the scGPT-based transformer architecture to learn representations of cells and genes and utilizes cell embeddings for cell type annotation [41].

Even though transformer-based models yield promising results, they exhibit limitations in modeling and capturing relationships among different genes and cells. We present scGraPhT as a transformer-based hybrid GNN model transferring embeddings from the transformer model to a GNN model that can dynamically construct and tailor the relationships between different genes and cells.

C. *GNN-Based Cell Type Annotation Methods*

GNNs are powerful deep learning techniques that are used in the analysis of scRNA-seq data, enabling the integration of complex cellular relationships and accurately encoding the homogeneous and heterogeneous interactions within biological entities [12]. To represent these interactions, several methods employ heterogeneous graph structures to encode relationships between cells and genes across cells from multiple reference datasets and within separate homogeneous graphs for cells and genes

[62], [63], [76], [77]. These heterogeneous graph structures can capture the wide array of relationships in a biological system, enhancing the model's ability to accurately annotate cell types. However, they may require more computational resources due to the complexity of modeling various interactions within a single graph structure. In contrast, most of the cell type annotation models utilize homogeneous graph structures based on either cell-cell or gene-gene interactions [58], [60], [61], [63], [78]. These models are simpler and more computationally efficient, focusing on uniform relationships within a single type of entity (i.e. cell-cell or gene-gene). However, these models may lack the expressiveness needed to capture the diverse interactions present in biological systems that are complex and dynamic by nature.

Our method, scGraPhT, addresses these drawbacks by introducing an adaptive graph construction approach that utilizes subgraph layers to concurrently capture both homogeneous and heterogeneous interactions. These subgraphs model cell-cell, gene-gene, and cell-gene interactions, and the overall assembled graph can model the combination of these subgraphs.

III. scGraPhT

In this section, we present the details of our proposed method, scGraPhT. After delineating the problem formulation in Section III-A, we present the overall model architecture in Section III-B. We analytically describe the transformer- and graph-based components of scGraPhT in Sections III-C and III-D, respectively. We present model variants based on different strategies to integrate transformer and graph components in Section III-E. Lastly, we present computational complexity analysis of our model's GNN component in Section III-F.

A. Problem Formulation

We formulate that scRNA-seq data is represented by the gene expression matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, where N denotes the number of cells and M denotes the number of unique genes. We represent each cell and gene by their corresponding embedding vectors. For the i-th cell c_i and the j-th unique gene g_j; $\mathbf{v}_i \in \mathbb{R}^D$ and $\mathbf{u}_j \in \mathbb{R}^D$ denote their D-dimensional embedding vectors, respectively. The matrix $\mathbf{C} \in \mathbb{R}^{N \times D}$ and $\mathbf{G} \in \mathbb{R}^{M \times D}$ denote the cell and gene embedding matrices, respectively. Cells with the expression matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ in scRNA-seq data may or may not have a corresponding cell type label during training stage, so we use $\mathcal{C}_{L_{train}}$ and $\mathcal{C}_{L_{test}}$ to denote the set of labeled and unlabeled cell indices, respectively. For a total number of K classes in a scRNA-seq data, $\mathbf{y}_i \in \mathbb{R}^K$ denotes the 1-hot representation of c_i's target class, and $\mathbf{Y} \in \mathbb{R}^{N \times K}$ denotes the target class matrix. For the cell type annotation task, we denote the model output as $\mathbf{P} \in \mathbb{R}^{N \times K}$, which is the classification probability distribution of N cells over K classes. Please refer to Table I for a summary of variable notations.

TABLE I
Summary of Notation

Notation	Meaning
N	Number of cells
M	Number of genes
D	Embedding dimension
c_i	i-th cell, $i \in \{1, 2, \ldots, N\}$
g_j	j-th gene, $j \in \{1, 2, \ldots, M\}$
K	Number of classes
$\mathcal{C}_{L_{train}}$	Labeled cell indices
$\mathcal{C}_{L_{test}}$	Unlabeled cell indices
$\mathbf{Y} \in \mathbb{R}^{N \times K}$	Matrix of target classes
$\mathbf{v}_i \in \mathbb{R}^D$	D-dimensional embedding of c_i
$\mathbf{u}_j \in \mathbb{R}^D$	D-dimensional embedding of g_j
$\mathbf{I}_n \in \mathbb{R}^{n \times n}$	Any identity matrix of size n
$\mathbf{X} \in \mathbb{R}^{N \times M}$	Gene expression matrix
$\mathbf{C} \in \mathbb{R}^{N \times D}$	Cell embedding matrix
$\mathbf{G} \in \mathbb{R}^{M \times D}$	Gene embedding matrix
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Any adjacency matrix with n nodes
$\mathbf{A_{CC}} \in \mathbb{R}^{N \times N}$	Cell-cell adjacency matrix
$\mathbf{A_{CG}} \in \mathbb{R}^{N \times M}$	Cell-gene adjacency matrix
$\mathbf{A_{GC}} \in \mathbb{R}^{M \times N}$	Gene-cell adjacency matrix
$\mathbf{A_{GG}} \in \mathbb{R}^{M \times M}$	Gene-gene adjacency matrix
$\mathbf{P} \in \mathbb{R}^{N \times K}$	Cell type probability distribution
$\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$	Any trainable weight matrix
$\mathbf{S} \in \{\mathbf{A_{CC}}, \mathbf{A_{GG}}, \mathbf{A_{CG}}, \mathbf{A_{GC}}\}$	Subgraph matrices
$\mathbf{E} \in \{\mathbf{C}, \mathbf{G}, \mathbf{I}_N, \mathbf{I}_M\}$	Cell and gene embedding matrices

B. Overall Model Architecture

A schematic of scGraPhT's model architecture that comprises a large-scale pre-trained transformer, GNN, and integration modules is displayed in Fig. 1. Given the scRNA-seq data with the expression matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, which is composed of a collection of cells $\{c_i\}_{i=1}^N$ and unique genes $\{g_j\}_{j=1}^M$, scGraPhT first employs the scGPT [41], a generative pre-trained model tailored for scRNA-seq analysis, to produce embedding matrices $\mathbf{C}_{\text{scGPT}} \in \mathbb{R}^{N \times D}$ and $\mathbf{G}_{\text{scGPT}} \in \mathbb{R}^{M \times D}$ for the cells and genes, respectively. These embedding matrices provide a rich representation of cell and gene entities that can show a high degree of generalization. However, as scGPT architecture is subjected to task-agnostic pre-training, they can be suboptimal in capturing interactions between entities. Thus, scGraPhT does not rely solely on transformer-based predictions of cell type.

Instead, scGraPhT employs a GNN module to construct a graph that sensitively captures homogeneous and heterogeneous interactions among cells and genes, where graph nodes are initialized with transformer-driven embeddings.

Finally, cell type predictions from scGPT and GNN modules are fused with an interpolation coefficient λ within the integration module as follows:

$$\mathbf{P}_{\text{scGraPhT}} = \lambda \mathbf{P}_{\text{GNN}} + (1 - \lambda) \mathbf{P}_{\text{scGPT}}, \tag{1}$$

where $\lambda \in [0, 1]$. As such, scGraPhT combines the contextual sensitivity of large-scale pre-trained scGPT models, along with the global sensitivity of GNNs to entity interactions in order to achieve a more comprehensive characterization of scRNA-seq data and thereby improve accuracy in cell type annotation.

C. scGPT Module: Cell and Gene Embeddings

The scGPT module functions in two primary stages. In the first stage, the gene expression matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ is transformed into input gene embeddings $\hat{\mathbf{X}} \in \mathbb{R}^{N \times \hat{M} \times D}$ through an embedding layer for each cell, where D is the embedding dimension. This transformation comprises both pre-processing of the expression matrix and mapping into embedding space. It should be noted that $\hat{M}$ represents the total sequence length, comprising both the number of gene tokens and special tokens,

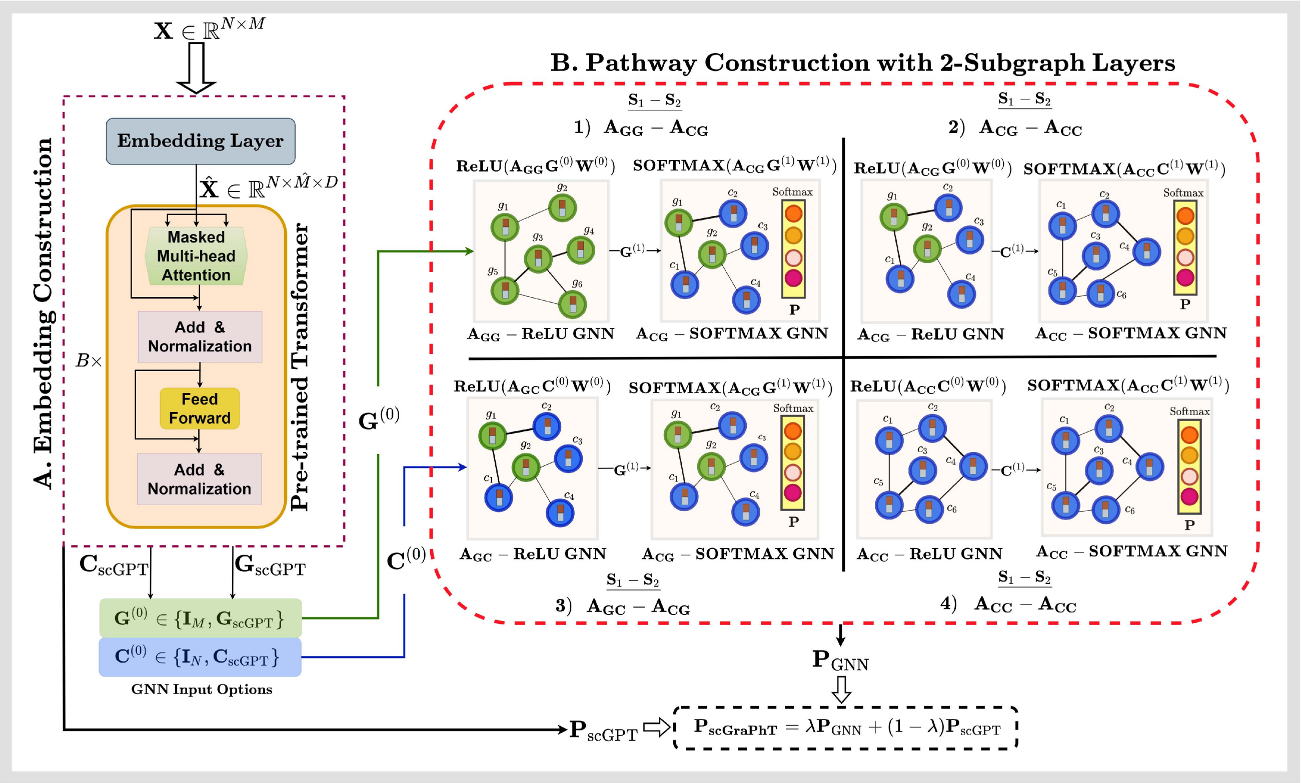
$\mathbf{X} \in \mathbb{R}^{N\times M}$
A. Embedding Construction
Embedding Layer
$\hat{\mathbf{X}} \in \mathbb{R}^{N\times\hat{M}\times D}$
Masked Multi-head Attention
Add & Normalization
Feed Forward
Add & Normalization
$B\times$
Pre-trained Transformer
$\mathbf{C}_{\text{scGPT}}$
$\mathbf{G}_{\text{scGPT}}$
$\mathbf{G}^{(0)} \in \{\mathbf{I}_M, \mathbf{G}_{\text{scGPT}}\}$
$\mathbf{C}^{(0)} \in \{\mathbf{I}_N, \mathbf{C}_{\text{scGPT}}\}$
GNN Input Options
$\mathbf{G}^{(0)}$
$\mathbf{C}^{(0)}$
B. Pathway Construction with 2-Subgraph Layers
$\mathbf{S}_1 - \mathbf{S}_2$
1) $\mathbf{A}_{\mathbf{GG}} - \mathbf{A}_{\mathbf{CG}}$
$\text{ReLU}(\mathbf{A}_{\mathbf{GG}}\mathbf{G}^{(0)}\mathbf{W}^{(0)})$
$\text{SOFTMAX}(\mathbf{A}_{\mathbf{CG}}\mathbf{G}^{(1)}\mathbf{W}^{(1)})$
$\mathbf{A}_{\mathbf{GG}}$ – ReLU GNN
$\mathbf{A}_{\mathbf{CG}}$ – SOFTMAX GNN
2) $\mathbf{A}_{\mathbf{CG}} - \mathbf{A}_{\mathbf{CC}}$
$\text{ReLU}(\mathbf{A}_{\mathbf{CG}}\mathbf{G}^{(0)}\mathbf{W}^{(0)})$
$\text{SOFTMAX}(\mathbf{A}_{\mathbf{CC}}\mathbf{C}^{(1)}\mathbf{W}^{(1)})$
$\mathbf{A}_{\mathbf{CG}}$ – ReLU GNN
$\mathbf{A}_{\mathbf{CC}}$ – SOFTMAX GNN
3) $\mathbf{A}_{\mathbf{GC}} - \mathbf{A}_{\mathbf{CG}}$
$\text{ReLU}(\mathbf{A}_{\mathbf{GC}}\mathbf{C}^{(0)}\mathbf{W}^{(0)})$
$\text{SOFTMAX}(\mathbf{A}_{\mathbf{CG}}\mathbf{G}^{(1)}\mathbf{W}^{(1)})$
$\mathbf{A}_{\mathbf{GC}}$ – ReLU GNN
$\mathbf{A}_{\mathbf{CG}}$ – SOFTMAX GNN
4) $\mathbf{A}_{\mathbf{CC}} - \mathbf{A}_{\mathbf{CC}}$
$\text{ReLU}(\mathbf{A}_{\mathbf{CC}}\mathbf{C}^{(0)}\mathbf{W}^{(0)})$
$\text{SOFTMAX}(\mathbf{A}_{\mathbf{CC}}\mathbf{C}^{(1)}\mathbf{W}^{(1)})$
$\mathbf{A}_{\mathbf{CC}}$ – ReLU GNN
$\mathbf{A}_{\mathbf{CC}}$ – SOFTMAX GNN
g_1 g_2 g_3 g_4 g_5 g_6
c_1 c_2 c_3 c_4 c_5 c_6
$\mathbf{G}^{(1)}$
$\mathbf{C}^{(1)}$
Softmax
P
$\mathbf{P}_{\text{GNN}}$
$\mathbf{P}_{\text{scGPT}}$
$\mathbf{P}_{\text{scGraPhT}} = \lambda\mathbf{P}_{\text{GNN}} + (1-\lambda)\mathbf{P}_{\text{scGPT}}$

Fig. 1. The overall architecture of our scGraPhT model with two-layer pathways is illustrated. $\mathbf{S}_1 - \mathbf{S}_2$ indicates that the pathways are constructed by sequentially integrating two consecutive subgraph layers, $\mathbf{S}_1$ and $\mathbf{S}_2$. The process begins with the input of cell and gene embeddings, which are first processed through an initial subgraph layer. These embeddings are then passed to a second subgraph layer for further processing and classification. An integration module, which combines the transformer and GNN modules, can be incorporated in the final stage, depending on the model variant.

such as [CLS], which summarizes gene information into a single cell representation, and [PAD], which ensures consistent input lengths across sequences.

In the second stage, these embeddings are fed into a transformer model, which consists of multiple blocks of masked multi-head self-attention (MMHSA) layers and feedforward neural networks (FFN). Given the input embeddings $\hat{\mathbf{X}}$, the embeddings are first divided into h heads for multi-head self-attention, resulting in:

$$\hat{\mathbf{X}}_i \in \mathbb{R}^{N\times\hat{M}\times\frac{D}{h}} \quad \forall i \in \{1, 2, \ldots, h\}. \tag{2}$$

Next, for each head, the transformer computes three sets of vectors: queries ($\mathbf{Q}_i$), keys ($\mathbf{K}_i$), and values ($\mathbf{V}_i$). These are obtained by linear transformations specific to each head:

$$\mathbf{Q}_i = \hat{\mathbf{X}}_i\mathbf{W}_{Q_i}, \quad \mathbf{K}_i = \hat{\mathbf{X}}_i\mathbf{W}_{K_i}, \quad \mathbf{V}_i = \hat{\mathbf{X}}_i\mathbf{W}_{V_i}, \tag{3}$$

where $\mathbf{W}_{Q_i}, \mathbf{W}_{K_i}, \mathbf{W}_{V_i} \in \mathbb{R}^{\frac{D}{h}\times\frac{D}{h}}$ are the learnable weight matrices for each head. The attention mechanism computes the attention scores for each head as follows:

$$\text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \text{SOFTMAX}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{\frac{D}{h}}}\right)\mathbf{V}_i. \tag{4}$$

The calculated h attention scores are concatenated for improved representations and linearly transformed:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O, \tag{5}$$

where each head is defined as $\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$, and $\mathbf{W}_O \in \mathbb{R}^{D\times D}$ is the output weight matrix. The result of the multi-head self-attention is then processed through a residual connection and layer normalization:

$$\mathbf{X}_{LN}^1 = \text{LayerNorm}(\hat{\mathbf{X}} + \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})). \tag{6}$$

Next, the output of the attention layer $\mathbf{X}_{LN}^1$ is passed through an FFN, which consists of a linear transformation followed by a ReLU activation function:

$$\text{FFN}(\mathbf{X}_{LN}^1) = \text{ReLU}(\mathbf{X}_{LN}^1\mathbf{W}_{\text{FFN}}), \tag{7}$$

where $\mathbf{W}_{FFN} \in \mathbb{R}^{D\times D}$ are learnable parameters. The output of the FFN is again processed through a residual connection and layer normalization:

$$\mathbf{X}_{LN}^2 = \text{LayerNorm}(\mathbf{X}_{LN}^1 + \text{FFN}(\mathbf{X}_{LN}^1)). \tag{8}$$

These operations are iterated over consecutive transformer blocks, wherein the output of one block is fed as the input for the subsequent block. Let $\mathbf{H}^0 = \hat{\mathbf{X}} \in \mathbb{R}^{N\times\hat{M}\times D}$. The entire process can be summarized as follows:

$$\mathbf{H}^l = \text{TransformerBlock}_l(\mathbf{H}^{l-1}) \quad \forall l \in \{1, 2, \ldots, B\}, \tag{9}$$

where $\mathbf{H}^B \in \mathbb{R}^{N \times \hat{M} \times D}$ is the output of final transformer block, B is the number of transformer blocks. It should be noted that scGPT uses accelerated self-attention in its transformer blocks, implemented through FlashAttention [41], [79]. The first entry of $\mathbf{H}^B$ in the second dimension corresponds to the [CLS] embeddings, which are identical to the cell embedding matrix $\mathbf{C}_{\text{scGPT}}$ for all cells. Similarly, the remaining entries, excluding the [PAD] embeddings, are pooled to construct the gene embedding matrix $\mathbf{G}_{\text{scGPT}}$. Ultimately, the [CLS] embeddings are mapped to the classification layer. The formal definition of this mapping is given as:

$$\mathbf{P}_{\text{scGPT}} = \text{SOFTMAX}(\mathbf{C}_{\text{scGPT}} \mathbf{W}_{\text{scGPT}}), \tag{10}$$

where $\mathbf{W}_{\text{scGPT}}$ denotes the set of MLP blocks that map the cell embeddings to the cell type probabilities $\mathbf{P}_{\text{scGPT}} \in \mathbb{R}^{N \times K}$. Further details on the scGPT module can be found in [41].

D. GNN Module: Layer and Pathway Construction

Graph construction inherently involves the specification of an adjacency matrix that governs interactions among graph nodes. In scGraPhT, two possible node types are considered to correspond to cell (c_i) and gene (g_j) entities. Here, we propose four distinct types of adjacency matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$, where n denotes the number of nodes. Heterogeneous graphs can be constructed based on $\mathbf{A_{CG}} \in \mathbb{R}^{N \times M}$ that allows cells to aggregate gene information, or $\mathbf{A_{GC}} \in \mathbb{R}^{M \times N}$ that allows genes to aggregate cell information. Meanwhile, homogeneous graphs can be constructed based on $\mathbf{A_{CC}} \in \mathbb{R}^{N \times N}$ that allows cells to aggregate cell information, or $\mathbf{A_{GG}} \in \mathbb{R}^{M \times M}$ that allows genes to aggregate gene information. To capture both homogeneous and heterogeneous interactions, we propose to implement scGraPhT based on a mixture of subgraph layers derived from $\{\mathbf{A_{CC}}, \mathbf{A_{GG}}, \mathbf{A_{CG}}, \mathbf{A_{GC}}\}$. More explicitly, we build the entire GNN component of our scGraPhT by constructing pathways—multi-layered subgraphs formed through various combinations of these subgraphs.

In this section, we first define subgraph layers derived from different types of adjacency matrices. We then illustrate how these subgraph layers can be combined to produce *pathways* that are sensitive to different types of entity interactions.

1) Subgraph Layers: Constructing subgraph layers involves selecting a subgraph matrix $\mathbf{S}$ from the set $\{\mathbf{A_{CC}}, \mathbf{A_{GG}}, \mathbf{A_{CG}}, \mathbf{A_{GC}}\}$, an initial embedding matrix $\mathbf{E}$ from $\{\mathbf{C}, \mathbf{G}, \mathbf{I}_N, \mathbf{I}_M\}$, and an activation function ϕ from $\{\text{ReLU}, \text{Tanh}, \text{Sigmoid}, \text{SOFTMAX}\}$. Based on the graph convolution framework, a subgraph layer based on the above selections can be expressed as:

$$\mathbf{E}^{(l+1)} = \phi_l \left(\mathbf{S}^{(l)} \mathbf{E}^{(l)} \mathbf{W}^{(l)} \right), \tag{11}$$

where $\mathbf{W}^{(l)}$ denote network weights in layer l. Note that the inner dimensions of $\mathbf{S}^{(l)}$ and $\mathbf{E}^{(l)}$ must match, which restricts the selectable options for $\mathbf{E}^{(l)}$ given $\mathbf{S}^{(l)}$. For instance, when $\mathbf{S}^{(l)} = \mathbf{A_{GC}} \in \mathbb{R}^{M \times N}$, $\mathbf{E}^{(l)}$ can be either $\mathbf{I}_N \in \mathbb{R}^{N \times N}$ or $\mathbf{C} \in \mathbb{R}^{N \times D}$, and the subgraph layer maps from cells onto gene embeddings. Below, we summarize the proposed definitions of the adjacency matrices employed in scGraPhT:

- $\mathbf{A_{CG}} = \mathbf{X} \in \mathbb{R}^{N \times M}$ captures the expression patterns of genes within individual cells.
- $\mathbf{A_{GC}} = \mathbf{X}^T \in \mathbb{R}^{M \times N}$ captures the expression patterns of individual genes across cells.
- $\mathbf{A_{CC}}(i, j) = \frac{\mathbf{c}_i^T \mathbf{c}_j}{\|\mathbf{c}_i\|_2 \|\mathbf{c}_j\|_2}$ captures the similarities between gene expression patterns of pairs of cells (c_i, c_j) where $\mathbf{c}_k \in \mathbb{R}^M$ is k-th row vector of $\mathbf{X}$, $\forall k \in \{1, 2, \ldots, N\}$.
- $\mathbf{A_{GG}}(i, j) = \frac{\mathbf{g}_i^T \mathbf{g}_j}{\|g_i\|_2 \|\mathbf{g}_j\|_2}$ captures the similarities between cell expression patterns of pairs of genes (g_i, g_j) where $\mathbf{g}_k \in \mathbb{R}^N$ is k-th column vector of $\mathbf{X}$, $\forall k \in \{1, 2, \ldots, M\}$.

2) GNN Pathways: By integrating these subgraph layers in alignment with the input embeddings, we construct pathways represented as multi-layered graphs formed through the sequential aggregation of consecutive subgraphs. In scGraPhT, four different pathways are considered based on a combination of the abovementioned subgraph layers. These pathways are formed by connecting two consecutive subgraph layers, $\mathbf{S}_1$ and $\mathbf{S}_2$. It is important to note that any depth of these layers can be constructed, provided the input embeddings and final outputs of the subgraphs remain compatible. The principles of these pathways are described below:

- *Pathway 1 (P1):* $\mathbf{A_{GG}} - \mathbf{A_{CG}}$

Takes gene embeddings as input and produces hidden gene representations. Subsequently, final cell representations are generated:

$$\mathbf{G}^{(1)} = \text{ReLU}(\mathbf{A_{GG}} \mathbf{G}^{(0)} \mathbf{W}^{(0)}), \tag{12}$$

$$\mathbf{C}^{(2)} = \text{ReLU}(\mathbf{A_{CG}} \mathbf{G}^{(1)} \mathbf{W}^{(1)}). \tag{13}$$

- *Pathway 2 (P2):* $\mathbf{A_{GC}} - \mathbf{A_{CG}}$

Takes cell embeddings as input and produces hidden gene representations. Subsequently, final cell representations are generated:

$$\mathbf{G}^{(1)} = \text{ReLU}(\mathbf{A_{GC}} \mathbf{C}^{(0)} \mathbf{W}^{(0)}), \tag{14}$$

$$\mathbf{C}^{(2)} = \text{ReLU}(\mathbf{A_{CG}} \mathbf{G}^{(1)} \mathbf{W}^{(1)}). \tag{15}$$

- *Pathway 3 (P3):* $\mathbf{A_{CG}} - \mathbf{A_{CC}}$

Takes gene embeddings as input and produces hidden cell representations. Subsequently, final cell representations are generated:

$$\mathbf{C}^{(1)} = \text{ReLU}(\mathbf{A_{CG}} \mathbf{G}^{(0)} \mathbf{W}^{(0)}), \tag{16}$$

$$\mathbf{C}^{(2)} = \text{ReLU}(\mathbf{A_{CC}} \mathbf{C}^{(1)} \mathbf{W}^{(1)}). \tag{17}$$

- *Pathway 4 (P4):* $\mathbf{A_{CC}} - \mathbf{A_{CC}}$

Takes cell embeddings as input and produces hidden cell representations. Subsequently, final cell representations are generated:

$$\mathbf{C}^{(1)} = \text{ReLU}(\mathbf{A_{CC}} \mathbf{C}^{(0)} \mathbf{W}^{(0)}), \tag{18}$$

$$\mathbf{C}^{(2)} = \text{ReLU}(\mathbf{A_{CC}} \mathbf{C}^{(1)} \mathbf{W}^{(1)}). \tag{19}$$

Each pathway produces a final set of cell representations $\mathbf{C}^{(2)}$. To obtain cell type probabilities $\mathbf{P}_{\text{GNN}} \in \mathbb{R}^{N\times K}$ from the output, a linear layer followed by a softmax layer is employed, which is common across all pathways. The formal definition of this process is defined as follows:

$$\mathbf{P}_{\text{GNN}} = \text{SOFTMAX}(\mathbf{C}^{(2)}\mathbf{W}^{(2)}), \tag{20}$$

where $\mathbf{W}^{(2)} \in \mathbb{R}^{d_{out}\times K}$ denotes the linear layer that maps the node feature size to the number of classes, and d_{out} is the node feature size of $\mathbf{C}^{(2)} \in \mathbb{R}^{N\times d_{out}}$.

E. Training Procedures

In Section III-D, we described four distinct GNN pathways to construct scGraPhT variants, each variant capturing different relationships between cells and genes. Here, we consider four distinct training schemes to incorporate scGPT within scGraPhT: scGraPhT$_{GO}$, scGraPhT$_{EO}$, scGraPhT$_{EL}$, and scGraPhT$_{JT}$. It should be noted that all training schemes utilize a transductive setting.

1. scGraPhT$_{GO}$: is a "graph only" variant that takes gene and cell embeddings as one-hot vectors instead of utilizing embeddings from the scGPT module. Specifically, $\mathbf{C}^{(0)} = \mathbf{I}_N$ and $\mathbf{G}^{(0)} = \mathbf{I}_M$. This variant is trained via optimization over GNN weights:

$$\underset{W_{\text{GNN}}}{\text{argmax}} \sum_{t\in\mathcal{C}_{L_{train}}} \sum_{k=1}^{K} \mathbf{Y}_{tk} \ln \mathbf{P}_{\text{GNN},tk}. \tag{21}$$

2. scGraPhT$_{EO}$: is an "embedding only" variant that utilizes embeddings from the scGPT module. Specifically, we focus on cell embeddings, with $\mathbf{C}^{(0)} = \mathbf{C}_{\text{scGPT}}$ and $\mathbf{G}^{(0)} = \mathbf{I}_M$. This variant is again trained with (21).
3. scGraPhT$_{EL}$: is an "embedding & logit" variant that utilizes both the cell embeddings and the prediction logits from the scGPT module. Specifically, $\mathbf{C}^{(0)} = \mathbf{C}_{\text{scGPT}}$ and $\mathbf{G}^{(0)} = \mathbf{I}_M$. In this variant, the scGPT module provides pre-saved and fixed logits. Thus, the scGPT weights are not included in the optimization process, which adopts the integration layer in (1) as follows:

$$\underset{W_{\text{GNN}}}{\text{argmax}} \sum_{t\in\mathcal{C}_{L_{train}}} \sum_{k=1}^{K} \mathbf{Y}_{tk} \ln \mathbf{P}_{\text{scGraPhT},tk}. \tag{22}$$

4. scGraPhT$_{JT}$: is a "joint training" variant that extends EL, where the scGPT module also gets optimized during training, which allows $\mathbf{C}_{\text{scGPT}}$ to be updated. The classification probability of the hybrid model is computed with (1). The optimization process is done over GNN and scGPT weights simultaneously:

$$\underset{(W_{\text{GNN}},W_{\text{scGPT}})}{\text{argmax}} \sum_{t\in\mathcal{C}_{L_{train}}} \sum_{k=1}^{K} \mathbf{Y}_{tk} \ln \mathbf{P}_{\text{scGraPhT},tk}. \tag{23}$$

TABLE II
FORWARD COMPUTATIONAL COMPLEXITY OF GNNs IN SCGRAPHT FOR EACH PATHWAY

Pathway	First Layer	Second Layer	Total Forward Computation
P1	$\mathcal{O}(M^2h)$	$\mathcal{O}(MK(N+h))$	$\mathcal{O}(M^2h + MK(N+h))$
P2	$\mathcal{O}(Nh(M+D)$	$\mathcal{O}(MK(N+h))$	$\mathcal{O}(Nh(M+D) + MK(N+h))$
P3	$\mathcal{O}(M^2h + NMh))$	$\mathcal{O}(N^2K + NKh)$	$\mathcal{O}(N^2K + Mh(M+N) + NKh)$
P4:	$\mathcal{O}(N^2h + NDh)$	$\mathcal{O}(N^2K + NKh)$	$\mathcal{O}(N^2(K+h) + Nh(D+K))$

For all model variants, model performance was evaluated in the test stage via the accuracy metric:

$$Accuracy = \frac{\sum_{i\in\mathcal{C}_{L_{test}}} \mathbb{I}\left(\hat{y}_i = y_i\right)}{|\mathcal{C}_{L_{test}}|}, \tag{24}$$

where $\hat{y}_i = \text{argmax}_{(k\in\{1,2,\ldots,K\})}\mathbf{P}_{\text{scGraPhT},ik}$ is the predicted label of cell in i-th node, and y_i denotes the true label of cell in i-th node, and $\mathbb{I}(\hat{y}_i = y_i)$ is the indicator function that equals 1 if the predicted label $\hat{y}_i$ matches the true label y_i, and 0 otherwise.

F. Computational Complexity Analysis

We analyze the computational complexity of the GNN component in scGraPhT for each pathway during a single forward pass. Specifically, we present the layer-wise computational costs along with the total cost for two-layered pathways. In our analysis, M, N, D, h, and K represent the number of genes, the number of cells, the embedding dimension, the embedding dimension of any node between two layers, and the number of cell types, respectively. In all datasets, it is assumed that $N > M > D > h > K$. Table II presents the forward computational complexity of GNNs in scGraPhT, where the dominant term depends on N, given order of dimensions. In $P3$ and $P4$, which involve cell-cell interactions ($\mathbf{A}_{\text{CC}}$), the computational cost is primarily dominated by quadratic terms in N, making these pathways the most expensive. $P1$ and $P2$, which involve gene-gene ($\mathbf{A}_{\text{GG}}$) and gene-cell ($\mathbf{A}_{\text{GC}}$) interactions, have lower complexity terms in N. As a result, they remain relatively less expensive than $P3$ and $P4$ when N is significantly large, such as in datasets with millions of cell samples, as in the Human Cell Atlas. Apart from the time complexity, the memory requirement is another critical concern for growing graphs. In particular, the use of the adjacency matrix in pathways involving $\mathbf{A}_{\text{CC}}$ requires memory proportional to N^2. This quadratic scaling implies a considerable amount of storage capacity for large-scale datasets comprising millions of cells.

IV. EXPERIMENTS AND RESULTS

We first outline the datasets used for training and evaluating our model scGraPhT in Section IV-A. Next, we describe the experimental setup in Section IV-B. In Section IV-C, we present the results of our proposed method and its comparison with the state-of-the-art methods. Finally, Section IV-D presents our ablation studies.

A. Datasets

We evaluate our approach on three datasets: Multiple Sclerosis (MS), Human Pancreas (Pancreas), and Myeloid, which

TABLE III
DATASETS

	MS	Pancreas	Myeloid
Train Cells	7,844	10,600	9,748
Test Cells	13,468	4,218	3,430
Genes	3,000	3,000	3,000
Classes	18	14	21

TABLE IV
TRAINING HYPERPARAMETERS

	Min.	Max.	Step Size	Setup
Lambda (λ)	0.0	1.0	+0.1	0.7
LR-GNN	10^{-5}	10^{-3}	×10	10^{-4}
LR-scGPT	10^{-6}	10^{-4}	×10	10^{-5}
Dropout-GNN	0.1	0.4	+0.1	0.2
Dropout-scGPT	0.1	0.4	+0.1	0.2

were curated from the scGPT repository [41]. All training and test samples were pre-filtered following the protocol in [41], including selection of the top 3,000 highly variable genes and mitigation of scale differences via value binning. Dataset attributes are summarized in Table III.

1) MS: Originally released in [80], the MS dataset contains brain tissue cell types such as interneurons, excitatory neurons, glial and precursor cells, endothelial cells, and phagocytes. The training set contains nine healthy control individuals, whereas the test set contains 12 MS disease individuals. This setting enables the assessment of out-of-distribution generalization performance.

2) Pancreas: This dataset is comprised of five scRNA-seq datasets of human pancreas cells. The training set contains Baron [81] and Muraro [82], and the test set contains Xin [83], Segerstolpe [84], and Lawlor [85] datasets. The cell types primarily include pancreatic cells such as alpha, beta, delta, PP, PSC, and ductal cells. Immune cells such as mast, dendritic, B, and T cells are also present.

3) Myeloid: The Myeloid dataset [86] is available in the Gene Expression Omnibus database under the access tag GSE154763. The cell types all belong to the immune system and include various macrophage, monocyte, and conventional dendritic cell subtypes. The training set contains six individuals with different cancer types, while the test set contains three individuals with distinct cancer types not observed in the training set, which is another instance of out-of-distribution data.

B. Experimental Setup

Our design presents four different paths and four different training schemes due to the size-matching condition of pathways in a two-layered graph as discussed in Section III-D. We conducted experiments for each training strategy across each path simultaneously, resulting in a total of 16 different model variants per dataset. For all model variants, including the competing transformer baselines, we conducted model evaluations on test sets 10 times using a randomly created array of fixed seeds for a fair comparison. The collected scores were averaged, and standard deviations were computed.

All variants, except for scGraPhT$_{JT}$, train only the graph model, meaning that only the GNN weights $\mathbf{W}_{\text{GNN}}$ are updated. Compared to transformer models, which utilize computationally heavy multi-head attention mechanisms, this small computational complexity of GNNs allows for training without mini-batching while avoiding memory issues. Therefore, we perform a single forward pass each epoch, significantly reducing training time except for scGraPhT$_{JT}$. For the scGraPhT$_{JT}$ variant, mini-batch sizes were selected as 16 and ran for 25 epochs. The other variants were run for 3,000 iterations without mini-batching. Additionally for the scGraPhT$_{JT}$ variant, the standard learning rate (LR) 10^{-4} of the transformer was dropped to 10^{-5}.

To analyze the effects of different layer depths, we conducted additional experiments on single-layered and three-layered graphs. For the single-layer model, the size-matching validity only holds for $\mathbf{A_{CG}}$ and $\mathbf{A_{CC}}$, out of which $\mathbf{A_{CG}}$ was utilized. For the three-layer model, there exist 2^3 valid pathway combinations, out of which $\mathbf{A_{CG}} - \mathbf{A_{GC}} - \mathbf{A_{CG}}$ was utilized. The reasons for these choices are explained in Section IV-D.

Model hyperparameters were empirically selected based on performance on the validation set, which comprises 10% of the training data, as listed in Table IV. The selected parameters were used in all experiments. Experiments were conducted on an RTX 4090 24GB GPU via the PyTorch framework.

C. Results

We compare our method scGraPhT against 4 state-of-the-art transformer-based scRNA-seq analysis models: scBERT [40], TOSICA [39], scGPT [41], and CellPLM [74]. Apart from state-of-the-art deep learning approaches, we also compare our method against three conventional approaches: ACTINN [34], SingleCellNet [68], and CellTypist [70]. As evaluation metrics, we use accuracy and macro-F1 (mF1) scores for model performance and inference times for model complexity. We produced the results of the competing methods by strictly adhering to the guidelines provided in their original published papers and shared code repositories. To ensure a fair comparison, each method was executed 10 times on the same device, using an identical array of seeds. In Table V, we present the performance of scGraPhT alongside baseline methods. The results indicate that our proposed method significantly outperforms the existing approaches across nearly all datasets and metrics except one case. We also visually inspect our model's effectiveness using a Uniform Manifold Approximation and Projection (UMAP) plot illustrated in Fig. 2, where high-dimensional cell-type clusters are projected onto a two-dimensional space.

We now discuss the training variants and the pathway variants independently to assess their individual importance in our proposed method, starting with the training settings. Primarily, the scGraPhT$_{GO}$ variant generally underperforms against the transformer baselines and other training settings, which confirms the significance of having contextualized processing of the model through transformers in training. To uncover the effect of $\mathbf{P}_{\text{scGPT}}$, we examine the scGraPhT$_{EO}$ and scGraPhT$_{EL}$ variants in conjunction, which only differ with the inclusion of scGPT predictions. Notably, scGraPhT$_{EL}$ outperforms scGraPhT$_{EO}$ in most

TABLE V
ACCURACY AND MACRO-F1 SCORES ARE PRESENTED

METHOD	**DATASET**					
	MS		**Pancreas**		**Myeloid**	
	Accuracy (%) ↑	**mF1 (%) ↑**	**Accuracy (%) ↑**	**mF1 (%) ↑**	**Accuracy (%) ↑**	**mF1 (%) ↑**
ACTINN*	-	62.80±0.012	-	70.50±0.005	-	-
SingleCellNet*	-	63.70±0.001	-	73.90±0.006	-	-
CellTypist	76.05±0.002	62.28±0.004	97.17±0.002	66.29±0.04	61.12±0.004	32.56±0.002
scBERT	75.09±5.36	40.92±10.3	97.33±0.61	66.61±2.20	54.71±1.72	31.14±1.20
CellPLM	87.88±0.01	**76.59±0.01**	96.30±0.01	74.39±0.01	62.84±0.01	33.77±0.01
TOSICA	69.37±1.68	58.78±2.27	96.45±0.35	65.25±2.78	47.35±3.91	27.27±1.05
scGPT	84.96±1.17	71.13±2.68	96.41±1.07	72.82±4.49	63.47±3.00	34.16±1.41
$\underline{P1 : \mathbf{A_{GG}} - \mathbf{A_{CG}}}$						
scGraPhT$_{GO}$	83.51±0.92	70.00±1.62	94.85±2.62	$\underline{75.89\pm3.49}$	$\underline{65.58\pm1.58}$	$\underline{\mathbf{38.05\pm1.03}}$
scGraPhT$_{EO}$	83.51±0.92	70.00±1.62	94.85±2.62	$\underline{75.89\pm3.49}$	$\underline{65.58\pm1.58}$	$\underline{\mathbf{38.05\pm1.03}}$
scGraPhT$_{EL}$	$\underline{86.35\pm0.92}$	$\underline{72.25\pm1.09}$	$\underline{97.46\pm0.23}$	$\underline{74.48\pm0.74}$	$\underline{67.86\pm0.84}$	$\underline{38.02\pm0.90}$
scGraPhT$_{JT}$	$\underline{85.97\pm0.66}$	$\underline{73.23\pm0.55}$	96.30±0.41	$\underline{73.73\pm2.55}$	62.01±1.07	$\underline{37.42\pm0.48}$
$\underline{P2 : \mathbf{A_{GC}} - \mathbf{A_{CG}}}$						
scGraPhT$_{GO}$	81.03±1.30	68.69±0.76	89.16±12.8	67.47±16.0	$\underline{65.39\pm1.15}$	$\underline{34.80\pm0.42}$
scGraPhT$_{EO}$	$\underline{87.55\pm0.45}$	$\underline{73.99\pm0.45}$	$\underline{97.66\pm0.31}$	**80.14±1.87**	$\underline{67.97\pm1.20}$	$\underline{37.29\pm1.01}$
scGraPhT$_{EL}$	**88.17±0.41**	$\underline{74.30\pm0.31}$	**97.76±0.10**	$\underline{74.60\pm0.73}$	**68.08±0.59**	$\underline{37.86\pm0.78}$
scGraPhT$_{JT}$	$\underline{86.70\pm0.42}$	$\underline{72.90\pm0.51}$	$\underline{97.27\pm0.39}$	$\underline{72.90\pm1.49}$	$\underline{64.82\pm1.00}$	$\underline{36.73\pm1.80}$
$\underline{P3 : \mathbf{A_{CG}} - \mathbf{A_{CC}}}$						
scGraPhT$_{GO}$	79.96±0.64	69.66±0.44	84.59±3.17	61.70±1.08	55.94±1.30	31.45±0.59
scGraPhT$_{EO}$	79.96±0.64	69.66±0.44	84.59±3.17	61.70±1.08	55.94±1.30	31.45±0.59
scGraPhT$_{EL}$	82.65±0.52	$\underline{71.60\pm0.30}$	91.24±2.27	65.73±1.33	59.77±1.21	32.88±0.54
scGraPhT$_{JT}$	$\underline{85.21\pm0.81}$	$\underline{74.30\pm0.52}$	$\underline{97.61\pm0.44}$	$\underline{74.18\pm1.84}$	61.20±1.01	33.22±0.57
$\underline{P4 : \mathbf{A_{CC}} - \mathbf{A_{CC}}}$						
scGraPhT$_{GO}$	75.61±0.21	66.47±0.33	60.32±4.57	49.10±3.01	50.90±0.47	30.58±0.55
scGraPhT$_{EO}$	80.77±0.70	70.00±0.40	93.24±2.72	63.23±3.36	54.72±1.79	30.96±0.76
scGraPhT$_{EL}$	84.73±0.55	$\underline{72.81\pm0.40}$	$\underline{97.36\pm0.18}$	$\underline{73.64\pm0.52}$	62.78±1.22	33.92±0.57
scGraPhT$_{JT}$	$\underline{85.91\pm0.41}$	$\underline{73.29\pm0.57}$	$\underline{97.06\pm0.25}$	$\underline{75.50\pm1.07}$	$\underline{64.86\pm1.16}$	$\underline{35.91\pm1.02}$

The best model performances are highlighted in bold, and the scores from our method that outperform scGPT are underlined. Arrows indicate that higher scores are superior. (*) Indicates results taken from [74] to be included here for comparison purposes.

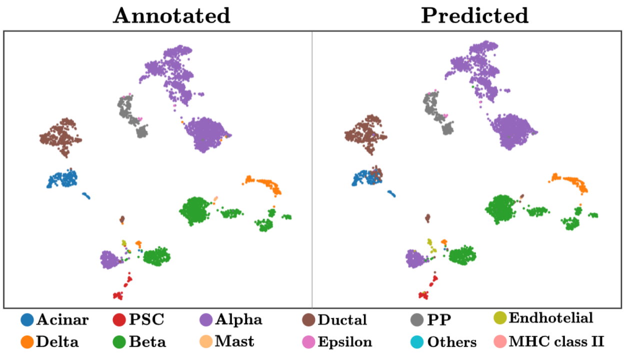

Fig. 2. Label-annotated and predicted cell types for the Pancreas dataset. Each color represents a different cell type in Pancreas tissue. Projection onto two-dimensional space was performed with UMAP.

cases by capturing the complementary aspects of the predictions, yielding better results from the interplay. In pathway variants, $P2$ demonstrates exceptional performance against other methods. Since heterogeneous subgraphs are directly based on the gene expression matrix in our formulation, the results show the importance of utilizing inherent interactions present in the data. A detailed interpretation of the subgraph connections will be provided in Section V.

To analyze the effect of the integration layer introduced in (1), we initialize $\lambda = 0.0$ and increment until $\lambda = 1.0$, which corresponds to utilizing only scGPT and only GNN predictions, respectively. For this analysis, we investigate all pathways under the scGraPhT$_{EL}$ variant. The results are displayed in Fig. 3, where the scGPT baseline acts as a comparative benchmark. A close examination reveals that:

1) For $\lambda \in [0.7, 0.9]$, where pathway $P2$ achieves optimal performance across all datasets, the GNN-based component contributes $70\% - 90\%$ of the weight to each prediction. This underscores the importance of incorporating global information in refining cell representations.
2) Increasing λ from 0.9 to 1.0 (the case where only GNN-based components of scGraPhT are taken into account in overall predictions), a decrease in accuracy across all pathways and datasets is observed. This suggests that scGPT predictions start complementing information potentially missed by the GNN module for contribution factors as low as $(1 - \lambda) = 0.1$.
3) There is no universal choice of λ that maximizes performance across all pathways since the peak of each pathway occurs at a different point. We extend this discussion in Section V to account for the importance of datasets on the choice of λ.

We also examine the inference times of the best-performing pathway $P2$. It should be noted that running scGPT once allows the embeddings and logits to be saved for later retrieval by our models. Thus, our approach requires a single complete run of scGPT beforehand for each dataset. Therefore, for fairness, our graph models' results are added to the baseline inference time for evaluation. Nevertheless, we note that any subsequent run of scGraPhT does not need to rerun scGPT, benefiting from the modular nature of our models. In Table VI, we demonstrate

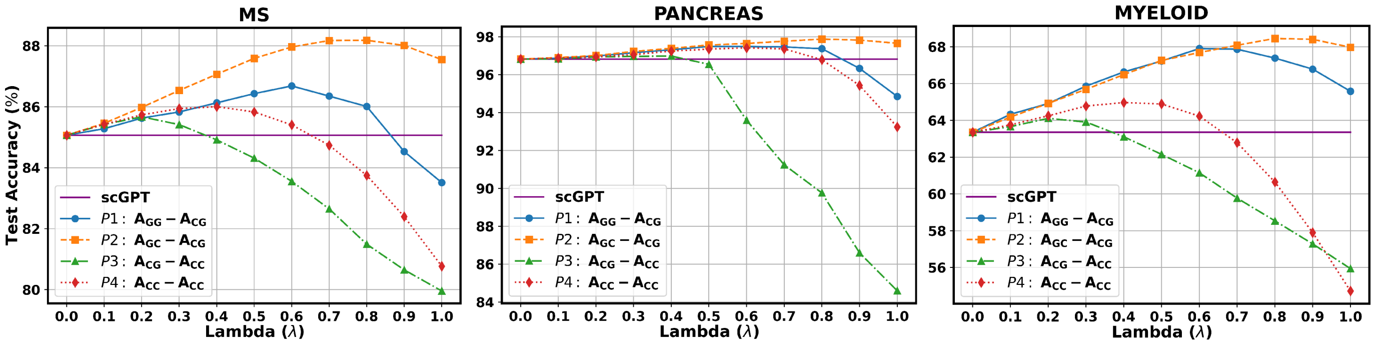

Fig. 3. Accuracy of scGraPhT with varying λ under the scGraPhT$_{EL}$ variant with all four pathways. The straight line and $\lambda = 0.0$ point correspond to the fine-tuned scGPT baseline. $\lambda = 1.0$ point corresponds to predictions based on merely GNN-based contributions.

TABLE VI
INFERENCE TIMES (IN SECOND) FOR QUERYING CELLS IN TEST PORTION ON AN NVIDIA RTX 4090 24GB GPU

METHOD	DATASET					
	MS		Pancreas		Myeloid	
	Duration (s) ↓	Accuracy (%) ↑	Duration (s) ↓	Accuracy (%) ↑	Duration (s) ↓	Accuracy (%) ↑
scBERT	416.09	75.09	341.93	97.33	176.16	54.71
CellPLM	9.29	87.88	9.36	96.30	10.09	62.84
TOSICA	5.05	69.37	2.19	96.45	1.96	47.35
scGPT	16.43	84.96	6.25	96.41	4.35	63.47
$\underline{\mathrm{A_{GC} - A_{CG}}}$						
scGraPhT$_{GO}$	0.032	81.03	0.026	89.16	0.025	65.39
scGraPhT$_{EO}$	16.43 + 0.028	87.55	6.25 + 0.022	97.66	4.35 + 0.023	67.97
scGraPhT$_{EL}$	16.43 + 0.037	88.17	6.25 + 0.028	97.76	4.35 + 0.026	68.08
scGraPhT$_{JT}$	16.43 + 55.72	86.70	6.25 + 19.90	97.27	4.35 + 14.10	64.82

Down arrows indicate that lower scores are better, while up arrows indicate that higher scores are better.

TABLE VII
COMPARISON OF MODEL PERFORMANCES BETWEEN SCBERT AND SCBERT$_{EL}$, WITH THE LATTER INTEGRATING OUR APPROACH INTO THE ORIGINAL SCBERT

	MS		Pancreas		Myeloid	
	Accuracy (%) ↑	mF1 (%) ↑	Accuracy (%) ↑	mF1 (%) ↑	Accuracy (%) ↑	mF1 (%) ↑
scBERT	75.09±5.36	40.92±10.25	97.33±0.61	66.61±2.20	54.71±1.72	31.14±1.20
scBERT$_{EL}$:						
$\mathrm{A_{GG} - A_{CG}}$	$\underline{76.26\pm0.98}$	$\underline{59.55\pm1.22}$	97.22±0.47	$\underline{\mathbf{76.93\pm1.73}}$	$\underline{\mathbf{59.18\pm1.01}}$	$\underline{\mathbf{33.92\pm1.00}}$
$\mathrm{A_{GC} - A_{CG}}$	$\underline{79.84\pm2.56}$	$\underline{62.45\pm1.49}$	$\underline{\mathbf{97.95\pm0.37}}$	$\underline{72.74\pm2.57}$	$\underline{56.41\pm1.38}$	$\underline{33.55\pm0.82}$
$\mathrm{A_{CG} - A_{CC}}$	$\underline{\mathbf{81.60\pm0.35}}$	$\underline{\mathbf{70.21\pm0.24}}$	91.52±2.42	65.19±1.80	$\underline{54.96\pm1.02}$	$\underline{32.87\pm0.53}$
$\mathrm{A_{CC} - A_{CC}}$	$\underline{78.32\pm0.57}$	$\underline{65.25\pm1.12}$	94.76±1.02	$\underline{70.70\pm1.80}$	$\underline{55.70\pm0.91}$	$\underline{33.12\pm0.63}$

The best model performances are emboldened, and the scores from hybrid methods outperforming scBERT are underlined.

that our models operate in the order of milliseconds, adding negligible changes to overall durations except for the scGraPhT$_{JT}$ variant, which is naturally more demanding due to joint training.

In the main experiments reported above, the transformer module in scGraPhT was implemented based on scGPT, given its generally high performance and fast inference times on most scRNA-seq datasets. That said, it is worth noting that other transformer architectures could also be adopted in scGraPhT modularly. To demonstrate this point, we also integrated our method with scBERT, creating scBERT$_{EL}$, where both embeddings and logits from scBERT were utilized in the GNN module of scGraPhT. Similar to the main results based on scGPT, we find that our hybrid transformer-GNN approach enables significant performance improvements over scBERT across all datasets. The results are detailed in Table VII.

D. Ablation Study

As an ablation study, we conduct a set of experiments to examine the effect of different layer depths in our proposed models. For reference, we utilize a single-layered and a three-layered model. Since our analysis indicated that the heterogeneous subgraphs might be more favorable, we set the layers as $\mathrm{A_{CG}}$ and $\mathrm{A_{CG} - A_{GC} - A_{CG}}$ for the new models, respectively. Table VIII illustrates that a two-layered composition is optimal for our annotation task, which may indicate that a one-layer aggregation may potentially fail to capture sufficient cell-gene information, whereas appending beyond two layers may cause over-aggregation, harming cell distinctiveness. However, we note that this optimal depth might not be suitable in all contexts. For instance, a multi-tissue dataset where the same cell types are sampled across different tissues is a scenario absent in the benchmark datasets. Since intra-tissue sample pairs of the same cell type often exhibit more similar profiles compared to inter-tissue pairs where technical noise tends to be more prevalent [87], extensive information sharing between nodes can propagate undesired variations, potentially harming cell distinctiveness. In such a case, adopting a single-layer GNN might be preferable to avoid excessive aggregation.

V. DISCUSSIONS

A primary motivation for building scGraPhT rests on the hypothesis that transformers and GNNs, due to their distinct

TABLE VIII
MODEL PERFORMANCE IN DIFFERENT LAYER DEPTHS

METHOD	DATASET					
	MS		Pancreas		Myeloid	
	Accuracy (%) ↑	mF1 (%) ↑	Accuracy (%) ↑	mF1 (%)	Accuracy (%) ↑	mF1 (%) ↑
$\mathrm{A_{CG}}$						
scGraPhT$_{GO}$	77.29±0.35	63.66±0.39	97.75±0.12	74.41±2.04	61.75±0.38	33.25±0.20
scGraPhT$_{EO}$	77.29±0.35	63.66±0.39	97.75±0.12	74.41±2.04	61.75±0.38	33.25±0.20
scGraPhT$_{EL}$	82.18±0.22	68.33±0.37	97.77±0.05	75.73±1.01	64.31±0.30	34.38±0.17
scGraPhT$_{JT}$	83.81±0.43	70.90±0.38	**98.07±0.28**	74.41±2.94	63.20±0.44	33.93±0.20
$\mathrm{A_{GC} - A_{CG}}$						
scGraPhT$_{GO}$	81.03±1.30	68.69±0.76	89.16±12.3	67.47±16.0	65.39±1.15	34.80±0.42
scGraPhT$_{EO}$	87.55±0.45	73.99±0.45	97.66±0.31	**80.14±1.87**	67.97±1.20	37.29±1.01
scGraPhT$_{EL}$	**88.17±0.41**	**74.30±0.31**	97.76±0.10	74.60±0.73	**68.08±0.59**	37.86±0.78
scGraPhT$_{JT}$	86.70±0.42	72.90±0.51	97.27±0.39	72.90±1.49	64.82±1.00	36.73±1.80
$\mathrm{A_{CG} - A_{GC} - A_{CG}}$						
scGraPhT$_{GO}$	75.71±3.36	57.60±2.48	96.57±0.56	69.56±2.30	62.86±2.21	38.15±2.18
scGraPhT$_{EO}$	75.71±3.36	57.60±2.48	96.57±0.56	69.56±2.30	62.86±2.21	38.15±2.18
scGraPhT$_{EL}$	84.05±0.79	65.62±1.51	97.59±0.22	70.69±1.25	66.54±1.05	**38.95±0.39**
scGraPhT$_{JT}$	84.72±1.07	68.47±1.48	96.44±1.17	72.22±2.61	62.01±1.26	37.35±0.60

Arrows indicate that higher scores are superior. The best scores are emboldened.

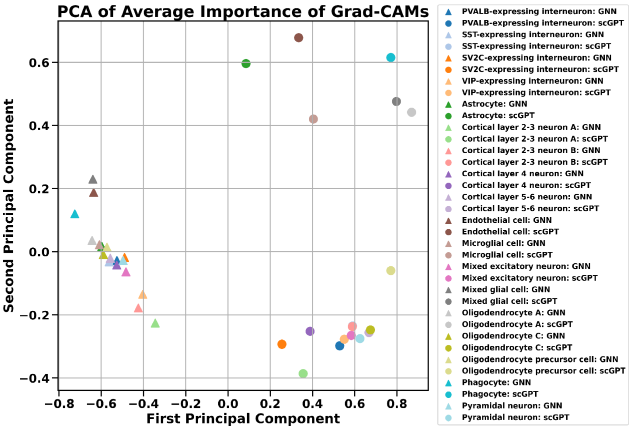

Fig. 4. PCA analysis of scGPT and GNN Grad-CAMs on the MS dataset. The triangles and circles depict GNN and scGPT results, respectively.

architectures, emphasize different aspects of cells and genes —transformers excel in contextualized processing, while GNNs are better at global analysis. By integrating these two complementary approaches, scGraPhT aims to achieve better performance in identifying more nuanced cell types, capturing subtleties that may be missed by either model in isolation. To offer support for this hypothesis and to make our approach more interpretable, we use Grad-CAM [64], a visual explainability method, on the scGPT and GNN modules separately, where the latter uses scGraPhT$_{EO}$ to benefit only from cell embeddings. The top 1% of importance scores returned by GradCAM are retained to identify the most critical entries of cell embeddings for distinguishing cell types, which result in an importance profile for each cell under each module. Afterward, principal component analysis (PCA) is performed on these importance profiles to assess the similarities/dissimilarities among different cell types and, more importantly, between scGPT and GNN modules, as depicted in Fig. 4.

Fig. 4 highlights a significant result: the importance profiles for the GNN module and the importance profiles for the scGPT module form spatially segregated clusters in the PCA space. This outcome indicates that, on average, for each cell type, the transformer and graph modules focus on distinct attributes of the cell embeddings during cell type predictions. Having validated the strength of the scGraPhT model, this result offers a mechanistic understanding of our approach, explaining how it outperforms state-of-the-art methods through its use of the two modules in a complementary manner. To ensure the consistency of this analysis, we ran our scGraPhT$_{EO}$ variant four more times with different seeds on the MS dataset. As anticipated, the clustering pattern was preserved across all trials. The resulting figures are provided in *the Supplementary Material*.

While a precise functional interpretation of gene-level contributions onto PCA axes is challenging, some clustering among different cell types is also evident from observing Fig. 4. In terms of segregation between respective GPT and GNN importance profiles, the PC1 axis appears to primarily capture a contrast between glial-immune cells (Oligodendrocytes, Glial Cells, and Phagocytes) with higher segregation versus neuronal-vascular cells (Cortical Neurons, Astrocytes, Interneurons, and Endothelial Cells) with lower segregation. PC2 axis appears to capture a contrast between glial/vascular cells (Microglia, Endothelial Cells, Astrocytes, Phagocytes, and Oligodendrocytes) with higher segregation versus neuronal cells (Cortical Neurons, Interneurons, and Pyramidal Neurons) with lower segregation.

Due to factors intrinsic to the scRNA-seq method, datasets usually suffer from mRNA count drop-outs and batch-to-batch differences. Our approach is no different from other state-of-the-art approaches with similar downstream analysis goals in terms of its performance being inherently constrained by the quality of the sequencing data. For example, an inspection of the results shown in Table V shows that cell-cell and gene-gene interaction layers generally underperform against heterogeneous subgraphs, which we attribute to the absence of inherent connections stemming directly from the data. This problem is further emphasized by our method of computing similarities between nodes, which results in a matrix where all entries are non-zero. Such non-sparsity necessitates the use of thresholding techniques such as k-nearest neighbors (KNN) or hard thresholding to limit connections and, consequently, the number of aggregations. However, these methods require manual cut-off value selections that lack flexibility and prevent generalization across different datasets. In future studies, an adaptive thresholding mechanism tailored to transcriptomics can potentially be adopted to overcome this potential generalization issue.

Another example of the scRNA-seq data-related performance constraints can be provided when the inherently stochastic nature of gene expression processes is considered. Due to stochasticity, gene-specific mRNA counts are usually different, even among isogenic cells grown in the same environment. Furthermore, the total number of mRNA molecules transcribed from a gene can be very low, increasing the impact of even minute differences in single-cell mRNA counts. Therefore, with such inherent and unavoidable variability, transcriptomics data alone might be insufficient in faithfully capturing cell-cell or gene-gene similarities. As additional information, knowledge graphs that model interactions between entities can be incorporated into training. Such information is generally static since they are based on known biological pathways, co-expressions, and functional relations. Although it may introduce higher computational complexities, cell-cell interaction [88] or gene-gene interaction [89] knowledge graphs can refine the homogeneous subgraphs based on experimentally validated interactions.

In our analyses, we observed that the parameter λ that governs how predictions from scGPT and GNN modules are fused is pathway-dependent, while the optimal lambda values for a given pathway are fairly consistent across datasets. However, the optimal λ values may change for other datasets that show distinct distributions from those examined here. To account for this potential data-driven variability, the selection of λ can be performed using an adaptive learning-based approach. In this regard, one approach is gradient-based optimization [90], which integrates λ directly into the training loss, allowing it to be optimized via backpropagation alongside other model parameters. As discussed in [90], gradient-based methods can be further enhanced by incorporating Bayesian optimization techniques for more robust hyperparameter learning. Another effective strategy involves leveraging meta-learning frameworks [91], which aim to learn an optimal λ that generalizes well across a variety of tasks or datasets. Additionally, reinforcement learning [92] can be employed to dynamically adjust λ based on a reward function that reflects performance improvements, offering a flexible and adaptive approach to hyperparameter optimization. These approaches could achieve an optimal balance between our modules while implicitly addressing the data-induced variations. Interestingly, if our approach were to be tailored to another downstream task, these methods could also possibly account for the distinct nature of the new task, where the optimal interplay between our modules might vary greatly from the trends observed herein.

In addition to the discussed data constraints, many benchmark scRNA-seq datasets suffer from cell type imbalances, which can affect the results for various downstream tasks [93]. Hence, this imbalance situation requires special attention and further analysis. To this end, we provide confusion matrices for scGPT and scGraPhT in cross-comparison to examine the success of our approach in identifying rare cell types. The results indicate that the scGraPhT shows cleaner segregation among distinct cell types compared to scGPT. Thus, the nuanced cell representations in scGraPhT help mitigate biases in the annotation of rare cell types. The comparative results are presented in *the Supplementary Material.*

In this study, we present a computationally efficient framework by utilizing pre-trained transformer-based embeddings. Additionally, we discuss the computational complexity of the GNN component in Section III-F and present the results in Table II. Our findings reveal that as the dataset size increases from tens of thousands to millions of samples, the computational cost of a single forward pass grows by a factor of 100 for $P1$ and $P2$ and 10,000 for $P3$ and $P4$. In terms of storage, all pathways impose significant memory demands, posing scalability challenges. However, it is important to note that scaling GNNs to large datasets is a well-documented challenge in the field and remains an active area of research. Therefore, this limitation is inherent to GNNs rather than specific to our approach. To further mitigate these challenges, distributed training can be employed, particularly for scGraPhT$_{JT}$, where the transformer and GNN components are trained in parallel on separate devices. Additionally, distributing graph computations, sparse matrix representations, and gradient checkpointing [94] can further optimize memory efficiency and computational performance.

VI. Limitations and Implications

One limitation of this study concerns an inevitable reality associated with scRNAseq experimental pipelines or output: missing entries or zero expression values in the gene expression matrix. We construct the adjacency matrices of our graphs from these expression matrices. In this context, significant data loss or sparsity in the original expression matrix may pose challenges in determining the degree of connections between nodes. As a solution, the missing or zero values in the expression matrix can be enriched at an earlier stage using specialized imputation models tailored for this purpose [95], [96].

Another limitation concerns the large datasets containing millions of cells. As discussed previously, the size of the dataset poses a challenge not only for our model but also for general GNN studies. However, as outlined, there are several strategies available to mitigate the computational and memory burden associated with handling such large-scale data.

Lastly, the presence of rare cell types gives rise to the issue of class imbalance in the dataset. While scGPT does not directly address this challenge, and we have not specifically focused on it, scGraPhT enhances the classification of rare cell types compared to scGPT, as previously discussed. Potential biases from class imbalance can be mitigated through imbalance-aware training techniques [97], both within the transformer framework and the GNN model we have developed. Furthermore, data augmentation strategies may also be employed to alleviate the impact of rare class types [98].

VII. Conclusion

We proposed scGraPhT, a merged transformer-graph model that draws parallels from the NLP and GNN domains to scRNA-seq analysis to enhance the distinctiveness of cell representations, leading to better results in cell-type annotation tasks. As scGraPhT is devised to capture distinct interaction types between cell and gene entities, it can aggregate global information in both homogeneous and heterogeneous settings. In addition to the contextualized processing of cells within the transformer model, which considers only intra-cell relations, global information passing facilitates cell representations to benefit from inter-cell dependencies that capture a broader context. Given that the complementary nature of the transformer and graph models was a central hypothesis of our approach, we validated it through explainability methods [64], rendering our approach and the outcomes more interpretable.

To examine the effects of the interplay between the transformer and graph models on single-cell annotation, we proposed four different training schemes that adopt distinct settings. Our experimental results illustrate that the proposed graph layers improve the annotation performance of existing transformer models, with only a marginal increase in required inference times in the order of milliseconds. In addition to the computational simplicity, we showcased that our merged transformer-graph framework is highly modular, highlighting its potential to enhance annotation capabilities for other foundation models pre-trained on single-cell transcriptomics data.

References

[1] M. M. Fansler et al., "Quantifying 3'UTR length from scRNA-seq data reveals changes independent of gene expression," *Nature Commun.*, vol. 15, no. 1, 2024, Art. no. 4050.

[2] A. Eisele et al., "Gene-expression memory-based prediction of cell lineages from scRNA-seq datasets," *Nature Commun.*, vol. 15, no. 1, 2024, Art. no. 2744.

[3] Q. Zhu et al., "Single cell multi-omics reveal intra-cell-line heterogeneity across Human cancer cell lines," *Nature Commun.*, vol. 14, no. 1, 2023, Art. no. 8170.

[4] M. Nishide et al., "Single-cell analysis in rheumatic and allergic diseases: Insights for clinical practice," *Nature Rev. Immunol.*, vol. 24, pp. 781–797, 2024.

[5] J. K. Ocasio et al., "scRNA-seq in medulloblastoma shows cellular heterogeneity and lineage expansion support resistance to SHH inhibitor therapy," *Nature Commun.*, vol. 10, no. 1, 2019, Art. no. 5829.

[6] Y. Dong et al., "scRNA-seq of gastric cancer tissues reveals differences in the immune microenvironment of primary tumors and metastases," *Oncogene*, vol. 43, no. 20, pp. 1549–1564, 2024.

[7] A. Ruta et al., "Single-cell transcriptomics in tissue engineering and regenerative medicine," *Nature Rev. Bioeng.*, vol. 2, no. 2, pp. 101–119, 2024.

[8] A. Garrido-Trigo et al., "Macrophage and neutrophil heterogeneity at single-cell spatial resolution in human inflammatory bowel disease," *Nature Commun.*, vol. 14, no. 1, 2023, Art. no. 4506.

[9] K. Sun et al., "scRNA-seq of gastric tumor shows complex intercellular interaction with an alternative T cell exhaustion trajectory," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 4943.

[10] P. V. Kharchenko, "The triumphs and limitations of computational methods for scRNA-seq," *Nature Methods*, vol. 18, no. 7, pp. 723–732, 2021.

[11] Y. Kashima et al., "Single-cell sequencing techniques from individual to multiomics analyses," *Exp. Mol. Med.*, vol. 52, no. 9, pp. 1419–1427, 2020.

[12] K. Lazaros et al., "Graph neural network approaches for single-cell data: A recent overview," *Neural Comput. Appl.*, vol. 36, pp. 9963–9987, 2024.

[13] D. A. Skelly et al., "Single-cell transcriptional profiling reveals cellular diversity and intercommunication in the mouse heart," *Cell Reports*, vol. 22, no. 3, pp. 600–610, 2018.

[14] D. T. Paik et al., "Single-cell RNA sequencing in cardiovascular development, disease and medicine," *Nature Rev. Cardiol.*, vol. 17, no. 8, pp. 457–473, 2020.

[15] Y. Yang et al., "Tracing immune cells around biomaterials with spatial anchors during large-scale wound regeneration," *Nature Commun.*, vol. 14, no. 1, 2023, Art. no. 5995.

[16] W. Hou et al., "Assessing GPT-4 for cell type annotation in single-cell RNA-seq analysis," *Nature Methods*, vol. 21, pp. 1462–1465, 2024.

[17] S. G. Riva, B. Myers, P. Cazzaniga, and A. Tangherloni, "A deep learning pipeline for the automatic cell type assignment of scRNA-seq data," in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol.*, 2022, pp. 1–8.

[18] S. Jia et al., "scDeepInsight: A supervised cell-type identification method for scRNA-seq data with deep learning," *Brief. Bioinf.*, vol. 24, no. 5, pp. 1–31, 2023.

[19] L. Chen et al., "Single-cell RNA-seq data semi-supervised clustering and annotation via structural regularized domain adaptation," *Bioinform.*, vol. 37, no. 6, pp. 775–784, 2021.

[20] F. Buettner et al., "Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells," *Nature Biotechnol.*, vol. 33, no. 2, pp. 155–160, 2015.

[21] B. Tasic et al., "Adult mouse cortical cell taxonomy revealed by single cell transcriptomics," *Nature Neurosci.*, vol. 19, no. 2, pp. 335–346, 2016.

[22] B. Yu et al., "scGMAI: A Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder," *Brief. Bioinf.*, vol. 22, no. 4, pp. 1–10, 2021.

[23] X. Shao et al., "scCATCH: Automatic annotation on cell types of clusters from single-cell RNA sequencing data," *Iscience*, vol. 23, no. 3, 2020, Art. no. 100882..

[24] X. Dong, S. Chowdhury, U. Victor, X. Li, and L. Qian, "Semi-supervised deep learning for cell type identification from single-cell transcriptomic data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 2, pp. 1492–1505, Mar./Apr. 2023.

[25] Y. Zhai et al., "Generalized cell type annotation and discovery for single-cell RNA-seq data," in *Proc AAAI Conf. Artif. Intell.*, 2023, pp. 5402–5410.

[26] T. Kim et al., "Impact of similarity metrics on single-cell RNA-seq data clustering," *Brief. Bioinf.*, vol. 20, no. 6, pp. 2316–2326, 2019.

[27] W. Zhang, Y. Li, and X. Zou, "SCCLRR: A robust computational method for accurate clustering single cell RNA-seq data," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 1, pp. 247–256, Jan. 2021.

[28] A. Gribov et al., "SEURAT: Visual analytics for the integrated analysis of microarray data," *BMC Med. Genomic.*, vol. 3, pp. 1–6, 2010.

[29] F. A. Wolf et al., "SCANPY: Large-scale single-cell gene expression data analysis," *Genome Biol.*, vol. 19, pp. 1–5, 2018.

[30] J. -H. Choi et al., "scTyper: A comprehensive pipeline for the cell typing analysis of single-cell RNA-seq data," *BMC Bioinf.*, vol. 21, pp. 1–8, 2020.

[31] O. Rozenblatt-Rosen et al., "The human cell Atlas: From vision to reality," *Nature*, vol. 550, no. 7677, pp. 451–453, 2017.

[32] O. Rozenblatt-Rosen et al., "Building a high-quality human cell atlas," *Nature Biotechnol.*, vol. 39, no. 2, pp. 149–153, 2021.

[33] A. Regev et al., "The Human cell Atlas," *eLife*, vol. 6, pp. 1–30, 2017.

[34] F. Ma et al., "ACTINN: Automated identification of cell types in single cell RNA sequencing," *Bioinform.*, vol. 36, no. 2, pp. 533–538, 2020.

[35] V. Y. Kiselev et al., "scmap: Projection of single-cell RNA-seq data across data sets," *Nature Methods*, vol. 15, no. 5, pp. 359–362, 2018.

[36] J. K. De Kanter et al., "CHETAH: A selective, hierarchical cell type identification method for single-cell RNA sequencing," *Nucleic Acids Res.*, vol. 47, no. 16, pp. 1–13, 2019.

[37] H. Bian et al., "scMulan: A multitask generative pre-trained language model for single-cell analysis," in *Proc. Int. Conf. Res. Comput. Mol. Biol.*, 2024, pp. 479–482.

[38] F. He et al., "Parameter-efficient fine-tuning enhances adaptation of single cell large language model for cell type identification," *bioRxiv*, 2024, doi: 10.1101/2024.01.27.577455.

[39] J. Chen et al., "Transformer for one stop interpretable cell type annotation," *Nature Commun.*, vol. 14, no. 1, 2023, Art. no. 223.

[40] F. Yang et al., "scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data," *Nature Mach. Intell.*, vol. 4, no. 10, pp. 852–866, 2022.

[41] H. Cui et al., "scGPT: Toward building a foundation model for single-cell multi-omics using generative AI," *Nature Methods*, vol. 21, pp. 1470–1480, 2024.

[42] D. Levine et al., “Cell2Sentence: Teaching large language models the language of biology,” in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 27299–27325.

[43] W. Connell et al., “A single-cell gene expression language model,” 2022, *arXiv:2210.14330*.

[44] T. Song et al., “TransCluster: A cell-type identification method for single-cell RNA-seq data using deep learning based on transformer,” *Front. Genet.*, vol. 13, pp. 1–10, 2022.

[45] S.Amara-Belgadi et al., “BioFormers: A scalable framework for exploring biostates using transformers,” *bioRxiv*, 2023, doi: 10.1101/2023.11.29.569320.

[46] H. Shen et al., “Generative pretraining from large-scale transcriptomes for single-cell deciphering,” *Iscience*, vol. 26, no. 5, 2023, Art. no. 106536.

[47] J. Gong et al., “xTrimoGene: An efficient and scalable representation learner for single-cell RNA-seq data,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 69391–69403.

[48] S. Kwak, N. Geroliminis, and P. Frossard, “Traffic signal prediction on transportation networks using spatio-temporal correlations on graphs,” *IEEE Trans Signal Inf. Process. Netw.*, vol. 7, pp. 648–659, 2021.

[49] M. Xu, W. Dai, C. Li, J. Zou, H. Xiong, and P. Frossard, “Graph neural networks with lifting-based adaptive graph wavelets,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 63–77, 2022.

[50] M. Ye, V. Stankovic, L. Stankovic, and G. Cheung, “Robust deep graph based learning for binary classification,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 7, pp. 322–335, 2021.

[51] Y. Lin et al., “BertGCN: Transductive text classification by combining GNN and BERT,” *Assoc Comput. Linguistics*, 2021, pp. 1456–1462.

[52] R. Ragesh et al., “HeteGCN: Heterogeneous graph convolutional networks for text classification,” in *Proc. ACM Int. Conf. Web Search Data Mining*, 2021, pp. 860–868.

[53] L. Yao et al., “Graph convolutional networks for text classification,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7370–7377.

[54] A. C. Aras, T. Alikaaifoglu, and A. Koc, “Graph receptive transformer encoder for text classification,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 10, pp. 347–359, 2024.

[55] A. C. Aras, T. Alikasifoglu, and A. Koç, “Text-RGNNs: Relational modeling for heterogeneous text graphs,” *IEEE Signal Process. Lett.*, vol. 31, pp. 1955–1959, 2024.

[56] E. R. Bonet et al., “Explaining graph neural networks with topology-aware node selection: Application in air quality inference,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 499–513, 2022.

[57] T. Alikaşifoğlu et al., “VISPool: Enhancing transformer encoders with vector visibility graph neural networks,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2024, pp. 2547–2556.

[58] K. Wang et al., “Adversarial dense graph convolutional networks for single-cell classification,” *Bioinformatics*, vol. 39, no. 2, pp. 1–7, 2023.

[59] T. Wang et al., “Single-cell classification using graph convolutional networks,” *BMC Bioinf.*, vol. 22, pp. 1–23, 2021.

[60] R. Bhadani et al., “Attention-based graph neural network for label propagation in single-cell omics,” *Genes*, vol. 14, no. 2, 2023, Art. no. 506.

[61] H.-B. Li et al., “Improving classification and data imputation for single-cell transcriptomics with graph neural networks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022.

[62] X. Shao et al., “scDeepSort: A pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network,” *Nucleic Acids Res.*, vol. 49, no. 21, pp. 1–13, 2021.

[63] R. Yang et al., “scbiGNN: Bilevel graph representation learning for cell type classification from single-cell RNA sequencing data,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2023.

[64] R. R. Selvaraju et al., “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” *Int. J. Comput. Vis.*, vol. 128, pp. 336–359, 2020.

[65] R. Lopez et al., “Deep generative modeling for single-cell transcriptomics,” *Nature Methods*, vol. 15, no. 12, pp. 1053–1058, 2018.

[66] D. Aran et al., “Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage,” *Nature Immunol.*, vol. 20, no. 2, pp. 163–172, 2019.

[67] P. Cahan et al., “CellNet: Network biology applied to stem cell engineering,” *Cell*, vol. 158, no. 4, pp. 903–915, 2014.

[68] Y. Tan et al., “SingleCellNet: A computational tool to classify single cell RNA-Seq data across platforms and across species,” *Cell Syst.*, vol. 9, no. 2, pp. 207–213, 2019.

[69] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, “Improved random forest for classification,” *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4012–4024, Aug. 2018.

[70] C. Domínguez Conde et al., “Cross-tissue immune cell analysis reveals tissue-specific features in humans,” *Science*, vol. 376, no. 6594, 2022, Art. no. eabl5197.

[71] P. Xie et al., “SuperCT: A supervised-learning framework for enhanced characterization of single-cell transcriptomic profiles,” *Nucleic Acids Res.*, vol. 47, no. 8, pp. 1–12, 2019.

[72] T. S. Johnson et al., “LAmbDA: Label ambiguous domain adaptation dataset integration reduces batch effects and improves subtype detection,” *Bioinformatics*, vol. 35, no. 22, pp. 4696–4706, 2019.

[73] J. Devlin et al., “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[74] H. Wen et al., “CellPLM: Pre-training of cell language model beyond single cells,” in *Proc. Int. Conf. Learn. Representations*, 2024.

[75] J. Achiam et al., “GPT-4 technical report,” 2023, *arXiv:2303.08774*.

[76] M. Yuan et al., “scMRA: A robust deep learning method to annotate scRNA-seq data with multiple reference datasets,” *Bioinformatics*, vol. 38, no. 3, pp. 738–745, 2022.

[77] Q. Song et al., “scGCN is a graph convolutional networks algorithm for knowledge transfer in single cell omics,” *Nature Commun.*, vol. 12, no. 1, 2021, Art. no. 3826.

[78] Q. Yin et al., “scGraph: A graph neural network-based approach to automatically identify cell types,” *Bioinformatics*, vol. 38, no. 11, pp. 2996–3003, 2022.

[79] T. Dao et al., “FlashAttention: Fast and memory-efficient exact attention with IO-awareness,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 16344–16359.

[80] L. Schirmer et al., “Neuronal vulnerability and multilineage diversity in multiple sclerosis,” *Nature Commun.*, vol. 573, pp. 75–82, 2019.

[81] M. Baron et al., “A single-cell transcriptomic map of the Human and mouse pancreas reveals inter-and intra-cell population structure,” *Cell Syst.*, vol. 3, no. 4, pp. 346–360, 2016.

[82] M. J. Muraro et al., “A single-cell transcriptome atlas of the human pancreas,” *Cell Syst.*, vol. 3, no. 4, pp. 385–394, 2016.

[83] Y. Xin et al., “RNA sequencing of single human islet cells reveals type 2 diabetes genes,” *Cell Metab.*, vol. 24, no. 4, pp. 608–615, 2016.

[84] Å. Segerstolpe et al., “Single-cell transcriptome profiling of Human pancreatic Islets in health and type 2 diabetes,” *Cell Metab.*, vol. 24, no. 4, pp. 593–607, 2016.

[85] N. Lawlor et al., “Single-cell transcriptomes identify Human islet cell signatures and reveal cell-type–specific expression changes in type 2 diabetes,” *Genome Res.*, vol. 27, no. 2, pp. 208–222, 2017.

[86] S. Cheng et al., “A pan-cancer single-cell transcriptional atlas of tumor infiltrating myeloid cells,” *Cell*, vol. 184, no. 3, pp. 792–809, 2021.

[87] J. Qie et al., “Integrated proteomic and transcriptomic landscape of macrophages in mouse tissues,” *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 7389.

[88] J. Y. Noh et al., “CCIDB: A manually curated cell–cell interaction database with cell context information,” *Database*, vol. 2023, pp. 1–8, 2023.

[89] M. B. Guebila et al., “GRAND: A database of gene regulatory network models across human conditions,” *Nucleic Acids Res.*, vol. 50, no. D1, pp. 1–12, 2022.

[90] D. Maclaurin et al., “Gradient-based hyperparameter optimization through reversible learning,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2113–2122.

[91] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, “Meta-learning in neural networks: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2022.

[92] X. Dong, J. Shen, W. Wang, L. Shao, H. Ling, and F. Porikli, “Dynamical hyperparameter optimization via deep reinforcement learning in tracking,” *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 43, no. 5, pp. 1515–1529, May 2021.

[93] H. Maan et al., “Characterizing the impacts of dataset imbalance in single-cell data integration,” *Nature Biotechnol.*, vol. 42, pp. 1899–1908, 2024, doi: 10.1038/s41587-023-02097-9.

[94] V. T. Chakaravarthy et al., “Efficient scaling of dynamic graph neural networks,” in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2021, pp. 1–15.

[95] R. Viñas et al., “Hypergraph factorization for multi-tissue gene expression imputation,” *Nature Mach. Intell.*, vol. 5, no. 7, pp. 739–753, 2023.

[96] W. V. Li et al., “An accurate and robust imputation method scimpute for single-cell RNA-seq data,” *Nature Commun.*, vol. 9, no. 1, 2018, Art. no. 997.

[97] K. Cao et al., “Learning imbalanced datasets with label-distribution-aware margin loss,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1567–1578.

[98] M. Marouf et al., “Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks,” *Nature Commun.*, vol. 11, no. 1, 2020, Art. no. 166.

Emirhan Koç (Student Member, IEEE) received the B.Sc. degree in electronics engineering from SabancıUniversity, Istanbul, Türkiye, in 2021. He is currently working toward the M.Sc. degree in electrical and electronics engineering with Bilkent University, Ankara, Türkiye. His research focuses on combining signal processing methods with deep learning models.

Emre Kulkul is currently working toward the undergraduation degree in electrical and electronics engineering with Bilkent University, Ankara, Türkiye. His research interests include graph neural networks and signal processing over graphs, with a focus on their potential applications in transcriptomics and epigenetics.

Gülara Kaynar is currently working toward the undergraduation degree in computer engineering with Koç University, Istanbul, Türkiye. In addition to her primary field of study, she is pursuing a minor in Molecular Biology and Genetics. Her academic interests lie at the intersection of computer science and biology.

Tolga Çukur (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2009. He was a Postdoctoral Fellow with the University of California, Berkeley, Berkeley, CA, USA, from 2010 to 2013. He is currently a Professor with the Department of Electrical and Electronics Engineering, Department of Neuroscience, and National Magnetic Resonance Research Center, Bilkent University, Ankara, Türkiye. His research interests include novel imaging and machine learning techniques for biomedical applications. He was the recipient of TUBA-GEBIP Outstanding Young Scientist Award, BAGEP Young Scientist Award, IEEE Turkey Research Encouragement Award, Science Heroes Association Young Scientist of the Year Award, METU Parlar Foundation Research Incentive Award, and TUSEB Incentive Award. Dr. Çukur is a Fellow of ISMRM. He is on the Editorial Boards of IEEE Transactions on Medical Imaging, IEEE Journal of Biomedical and Health Informatics, IEEE Open Journal of Engineering in Medicine and Biology, *Magnetic Resonance in Medicine*, *MAGMA*, and *Frontiers in Neuroscience*.

Murat Acar received the B.S. degree from Bogazici University, Istanbul, Türkiye, and the Ph.D. degree in physics from the Massachusetts Institute of Technology, Cambridge, MA, USA, where his studies focused on feedback regulation and genetic noise in gene networks. Prof. Acar joined CalTech as a Postdoctoral Fellow with the Center for Biological Circuit Design, and studied dosage compensation in genetic circuits. From 2012 to 2022, he was a Full-Time Faculty Member with the Department of Molecular Cellular and Developmental Biology, Yale University, New Haven, CT, USA, and the Yale Systems Biology Institute. He was promoted to the rank of Associate Professor in 2017. He was the recipient of the 2013 Ellison Medical Foundation New Scholar in Aging Award and the 2014 NIH New Innovator Award. Applying experimental/computational systems biology approaches, machine learning and deep learning methods, the ACAR Lab is interested in uncovering the genetic, epigenetic and pharmaceutical modulators of cellular aging and organismal lifespan, discovering/predicting disease markers/propensities.

Aykut Koç (Senior Member, IEEE) received the B.S. degree from Bilkent University, Ankara, Türkiye, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2005 and 2007, respectively. He is currently a Faculty Member with Electrical and Electronics Engineering, Bilkent University. He has authored or coauthored more than 90 research papers and one book chapter and issued five patents. His research interests include machine learning and signal processing, extending into natural language and graph signal processing. Dr. Koç is an Associate Editor for IEEE Signal Processing Letters,IEEE Transactions on Signal and Information Processing over Networks, and IEEE Transactions on Circuits and Systems for Video Technology. He is the Chair of the IEEE Signal Processing Society Turkiye Chapter. He was the recipient of the 2023 Science Academy Young Scientists Award (BAGEP).