Gaining Insight into SARS-CoV-2 Infection and COVID-19 Severity Using Self-supervised Edge Features and Graph Neural Networks

Arijit Sehanobish,*[*] Neal Ravindra,*[*] David van Dijk
Internal Medicine (Cardiology) and Computer Science, Yale University
{arijit.sehanobish, neal.ravindra, david.vandijk}@yale.edu

Abstract

A molecular and cellular understanding of how SARS-CoV-2 variably infects and causes severe COVID-19 remains a bottleneck in developing interventions to end the pandemic. We sought to use deep learning (DL) to study the biology of SARS-CoV-2 infection and COVID-19 severity by identifying transcriptomic patterns and cell types associated with SARS-CoV-2 infection and COVID-19 severity. To do this, we developed a new approach to generating self-supervised edge features. We propose a model that builds on Graph Attention Networks (GAT), creates edge features using self-supervised learning, and ingests these edge features via a Set Transformer. This model achieves significant improvements in predicting the disease state of individual cells, given their transcriptome. We apply our model to single-cell RNA sequencing datasets of SARS-CoV-2 infected lung organoids and bronchoalveolar lavage fluid samples of patients with COVID-19, achieving state-of-the-art performance on both datasets with our model. We then borrow from the field of explainable AI (XAI) to identify the features (genes) and cell types that discriminate bystander vs. infected cells across time and moderate vs. severe COVID-19 disease. To the best of our knowledge, this represents the first application of DL to identifying the molecular and cellular determinants of SARS-CoV-2 infection and COVID-19 severity using single-cell omics data.

Introduction

To address the devastating impact of the Coronavirus Disease of 2019 (COVID-19), caused by infection of SARS-CoV-2, and the gap in our understanding of the molecular mechanisms of severe disease and variable susceptibility to infection, we developed a DL framework around two single-cell transcriptomic datasets that allowed us to generate hypotheses related to these biological knowledge gaps (Yan et al. 2020; Zhong et al. 2020). We rely on single-cell transcriptomic data because single-cell datasets contain rich, molecular and cellular information across a variety of cell types and conditions. We work with two publicly available single-cell datasets: one in which upper airway bronchial epithelium (airway of the lung) organoids were infected with SARS-CoV-2 over a time-course and one dataset of bronchoalveolar lavage fluid samples from patients with varying degrees of COVID-19 severity (Ravindra et al. 2020b; Liao et al. 2020). Applying machine learning to these datasets allows us to identify the molecular and cellular patterns associated with susceptibility to SARS-CoV-2 infection or severe COVID-19, highlight potential biomarkers, and suggest therapeutic targets.

Single-cell RNA sequencing (scRNA-seq) is a technology that counts the number of mRNA transcripts for each gene within a single cell (Zheng et al. 2017; Hwang, Lee, and Bang 2018; Stuart and Satija 2019). Different tissue samples or cell culture experiments can be assayed with scRNA-seq technology, allowing one to collect information spanning a variety of disease states or perturbations, with thousands of cells' gene expression measured in each experiment. Since transcript counts are correlated with gene expression, scRNA-seq yields large datasets comprising many thousands of cells' gene expression (Zheng et al. 2017). However, identifying the genes that determine an individual cell's pathophysiological trajectory or response to viral insult remains a challenge as single-cell data is noisy, sparse, and high-dimensional (Chen, Ning, and Shi 2019; Kiselev, Andrews, and Hemberg 2019). As such, we require cutting-edge DL methods to learn how to discriminate cells' controlled experimental perturbation given their transcriptome. Here, we build on previous work that uses graph neural networks (GNNs) to predict an individual cell's disease-state label (Ravindra et al. 2020a). To reduce bias and improve performance, we developed a novel DL architecture, which, to the best of our knowledge, achieves the highest, single-cell resolved prediction of disease state. Using these models, we then identify the genes and cells important for these predictions.

GNNs have been widely used and developed for predictive tasks such as node classification and link prediction (Wu et al. 2020). GNNs learn from discrete relational structure in data but the use of similarity metrics to construct graphs from feature matrices can expand the scope of GNN applications to domains where graph structured data is not readily available (Franceschi et al. 2019; Tenenbaum 2000). GNNs typically use message passing, or recursive neighborhood aggregation, to construct a new feature vector for a particular node after aggregating information from its neighbor's fea-

[*] Equal Contribution

ture vectors (Xu et al. 2018; Kipf and Welling 2017). However recent work (Seshadhri et al. 2020) has shown that these low dimensional node representations may fail to capture important structural details of a graph. Recently, edge features have been incorporated into GNNs to harness information describing different aspects of the relationships between nodes (Gong and Cheng 2018; Gao et al. 2018; Gilmer et al. 2017; Simonovsky and Komodakis 2017; Hu et al. 2019) and potentially enrich these low dimensional node embeddings. However, there are very few frameworks for creating *de novo* edge feature vectors that significantly improve the performance of GNNs. In this article, we propose a self-supervised learning framework to create new edge features that can be used to improve GNN performance in downstream node classification tasks. We hope that our framework provides useful insights into the genes and cell types that might be important determinants of COVID-19 severity and SARS-CoV-2 infection, which can guide further study.

Related Work

There is a large body of research on Graph Neural Networks. A significant amount of work has been focused on graph embedding techniques, representation learning, various predictive analyses using node features and in understanding the representational power of GNNs. There has been recent interest in using edge features to improve the performance of Graph Neural Networks (Gong and Cheng 2018; Chen et al. 2019; Abu-El-Haija, Perozzi, and Al-Rfou 2017; Gilmer et al. 2017; Simonovsky and Komodakis 2017). However, there are few frameworks to create *de novo* edge features for graphs that do not inherently contain different edge attributes.

DL, particularly GNNs, have been used in biomedical research to predict medications and diagnoses from electronic health records data (Choi et al. 2017), protein-protein and drug-protein interactions from biological networks, and molecular activity from chemical properties (Nguyen et al. 2019; Chan et al. 2019; Harikumar et al. 2019; Veličković et al. 2018). Machine learning has been applied to single-cell data for other tasks, including data de-noising, batch correction, data imputation, unsupervised clustering, and cell-type prediction (Kiselev, Andrews, and Hemberg 2019; Torroja and Sanchez-Cabo 2019; Arisdakessian et al. 2019; Alquicira-Hernandez et al. 2019; Amodio et al. 2019). However, fewer works attempt to classify the experimental label associated with each cell or to predict pathophysiological state on an individual cell basis. One recent work uses GAT models to predict the disease state of individual cells derived from clinical samples (Ravindra et al. 2020a). However, their work does not use edge features. They also do not consider multiple disease states or disease severity. Lastly, they do not account for sample-source bias (i.e., batch effects) (Stuart and Satija 2019). In this work, we use a graph-structure that balances neighbors across sample sources to reduce batch effects while preserving biological variation (Luecken et al. 2020).

Finally there has been a lot of interest in the ML community to interpret black box models. Correctly interpreting ML models can lead to new scientific discoveries and shed light on the biases inherent in the data collection process. One of the most common and popular ways to interpret machine learning models is via Shapley values (Lundberg and Lee 2017) and it's various generalizations (Michalak et al. 2013). However Shapley values require the independence of features, which is generally hard to guarantee in biological datasets. Gradient-based interpretability methods are widely used in computer vision (Sundararajan, Taly, and Yan 2018; Shrikumar, Greenside, and Kundaje 2017) and recently, GNNExplainer (Ying et al. 2019) was proposed as a general interpretability method for predictions of any GNN-based model. GNNExplainer identifies a compact sub-graph structure and a small subset of node features that play an important role in a network's prediction. It is a gradient-based method and the authors formulate it as an optimization task that maximizes the mutual information between a GNN's prediction and the distribution of possible sub-graphs. In this work, in addition to GNNExplainer, we follow the approach of (Ravindra et al. 2020a; Alaa and van der Schaar 2019) in using attention mechanisms for interpretability.

To the best of our knowledge, this is the first attempt to apply a GNN architecture to gain insight into multiple pathophysiological states at single-cell resolution, merging timepoints, severity, and disease-state prediction into a multi-label node classification task.

Models

In this section we describe our model, which consists of two components: (1) A Set Transformer and (2) Graph Attention Network (GAT) layers.

Set Transformer

We use a Set Transformer as in (Lee et al. 2018). The Set Transformer is permutation invariant so it is an ideal architecture to encode sets. The building block of our Set Transformer is the multi-head attention, as in (Vaswani et al. 2017). Given n query vectors Q of dimension d_q, a key-value pair matrix $K \in \mathbb{R}^{n_v \times d_q}$ and a value matrix $V \in \mathbb{R}^{n_v \times d_v}$ and, assuming for simplicity that $d_q = d_v = d$, the attention mechanism is a function given by the following formula:

$$\text{att}(Q, K, V) := \text{softmax}(\frac{QK^T}{\sqrt{d}})V \tag{1}$$

This multihead attention is computed by first projecting Q, K, V onto h different d_q^h, d_q^h, d_v^h dimensional vectors where, for simplicity, we take $d_q^h = d_v^h = \frac{d}{h}$ such that,

$$\text{Multihead}(Q, K, V) := \text{concat}(O_1, \cdots, O_h)W^O \tag{2}$$

where

$$O_j = \text{att}(QW_j^Q, KW_j^K, VW_j^V) \tag{3}$$

, and W_j^Q, W_j^K, W_j^V are projection operators of dimensions $\mathbb{R}^{d_q \times d_q^h}$, $\mathbb{R}^{d_q \times d_q^h}$ and $\mathbb{R}^{d_v \times d_v^h}$, respectively, and W^O is a linear operator of dimension $d \times d$. Now, given a set S, the Set Transformer Block (STB) is given the following formula:

$$STB(S) := \text{LayerNorm}(X + rFF(X)) \tag{4}$$

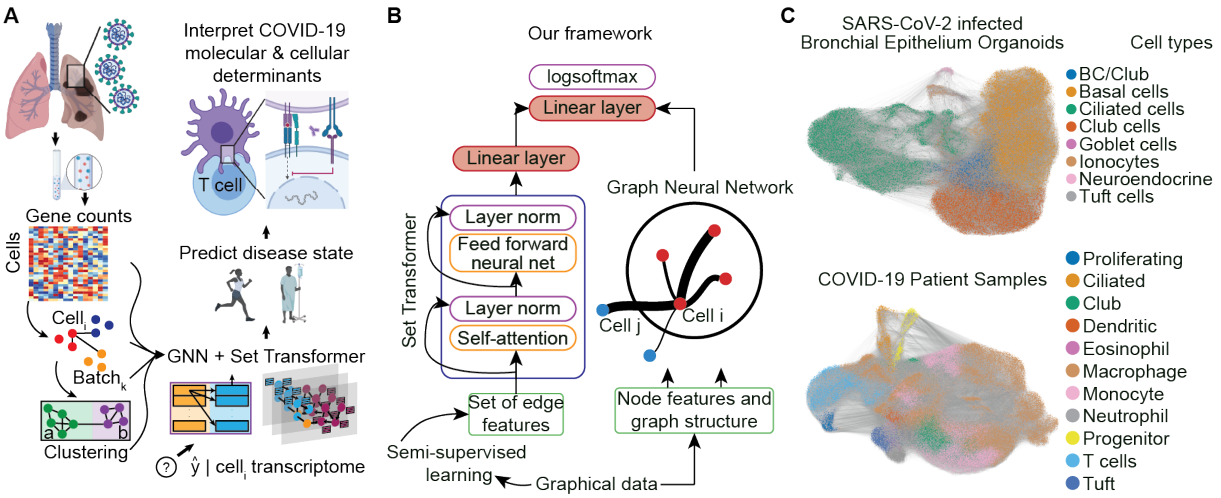

Figure 1: Our framework and datasets of interest. (A) Overview of our approach with respect to gaining molecular and cellular insights into COVID-19. (B) Our framework and model architecture, integrating edge features with GNNs via a Set Transformer. (C) Graphical data used, showing cell types for each cell and edges in a dimensionality-reduced embedding.

where

$$X = \text{LayerNorm}(S + \text{Multihead}(S, S, S)) \tag{5}$$

and rFF is a row-wise feedforward layer and LayerNorm is layer normalization (Ba, Kiros, and Hinton 2016).

Given a set of elements with input dimension d_{in}, the Set Transformer outputs a set of the same size with output dimension d_{out}. Since we will be dealing with sets of variable lengths, instead of outputting sets, we aggregate the output vectors to produce a single dense vector of dimension d_{out}. In particular, if for some set S of n elements, $\{w_1, \cdots, w_n\}$ is the output of the Set Transformer for the set S, we use learnable weights λ_j to combine the vectors via the following equation :

$$w := \sum_{j=1}^{n} \lambda_j w_j \tag{6}$$

Graph Attention Network

We use the popular Graph Attention Network (GAT) as the backbone to learn node representations and also for creating edge features based on our auxiliary tasks. We follow the exposition in (Veličković et al. 2018). The input to a GAT layer are the node features, $\mathbf{h} = \{h_1, h_2, ..., h_N\}$, where $h_i \in \mathbb{R}^F$, N is the number of nodes, and F is the number of features in each node. The layer produces a new set of node features (of possibly different dimension F') as its output, $\mathbf{h}' = \{h'_1, h'_2, ...h'_N\}$ where $h'_i \in \mathbb{R}^{F'}$. The heart of this layer is multi-head self-attention like in (Vaswani et al. 2017; Veličković et al. 2018). Self-attention is computed on the nodes,

$$a^l : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R} \tag{7}$$

where a^l is a feedforward network. Using self-attention, we can obtain attention coefficients,

$$e^l_{ij} = a^l(\mathbb{W}^l h_i, \mathbb{W}^l h_j) \tag{8}$$

where $\mathbb{W}^l$ is a linear transformation and also called the weight matrix for the head l. We then normalize these attention coefficients.

$$\alpha^l_{ij} = \text{softmax}_j(e^l_{ij}) = \frac{\exp(e^l_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e^l_{ik})} \tag{9}$$

where $\mathcal{N}_i$ is a 1-neighborhood of the node i. The normalized attention coefficients are then used to compute a linear combination of features, serving as the final output features for each corresponding node (after applying a nonlinearity, σ):

$$h^l_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha^l_{ij} \mathbb{W}^l h_j\right). \tag{10}$$

We concatenate the features of these heads to produce a new node feature, $h'_i := || \, h^l_i$.

However, for the final prediction layer, we average the representations over heads and apply a logistic sigmoid nonlinearity. Thus the equation for the final layer is:

$$h'_i = \sigma\left(\frac{1}{K} \sum_{l=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha^l_{ij} \mathbb{W}^l h_j\right). \tag{11}$$

where K is the number of heads.

Based on auxiliary tasks, our new edge features Λ_{ij} for the edge e_{ij} are created by concatenating the α^l_{ij} across all heads, i.e.

$$\Lambda_{ij} := ||^K_{l=1} \alpha^l_{ij} \tag{12}$$

Our Model

In this subsection we will describe our model that combines edge features with node features for our main node classification task. We use two GAT layers to encode the node representations. In the case of the GAT layers, we concatenate the representations obtained by different heads resulting in

Datasets	SARS-CoV-2 infected organoids	COVID-19 patients
# Nodes	54353/11646/11648	63486/13604/13605
# Node features	24714	25626
# Edges	1041226/230429/228630	2746280/703217/707529
# Edge features	18	18
# Classes	7	3
Tasks		
Main Inductive prediction	Timepoint and infection	No, Mild, or Severe Disease
Auxiliary batch metadata prediction	Culture sample ID	Patient ID
Auxiliary Louvain cluster ID prediction	Cell type	Cell type

Table 1: Dataset description showing train/val/test splits and the various experimental tasks we perform on our datasets.

a 64-dimensional node feature vector. For each node i, we construct a set $S_i := \{\Lambda_{ij} : j \in N_i\}$, where Λ_{ij} is the vector representing the edge features of the edge e_{ij} connecting the nodes i and j. We then encode this set, S_i, which we call the edge feature set attached to the node i via our Set Transformer. We use 2 heads and 1 block of the Set Transformer, outputting a 8-dimensional vector. This 8-dimensional vector is concatenated with the 64-dimensional node representation from the GAT layer. We call this new representation an enhanced node feature vector. This enhanced node feature vector is then passed through a dense layer with a logistic sigmoid non-linearity for classification. More details about the model and the training hyperparameters can be found in the Appendix. We finally note that instead of GAT layers, we can also use any message passing GNN layers in our main node classification task.

Datasets Used

We validate our model on the following scRNA-seq datasets:

- 4 human bronchial epithelial cell cultures or "organoids" that were inoculated with SARS-CoV-2 and co-cultured for 1, 2, and 3 days post-infection (Ravindra et al. 2020b).
- Bronchoalveolar lavage fluid samples from 12 patients enrolled in a study at Shenzen Third People's Hospital in Guangdong Province, China of whom 3 were healthy controls, 3 had a mild or moderate form of COVID-19 and 6 had a severe or critical COVID-19 illness (Liao et al. 2020).

Data Preprocessing : All single-cell samples were processed with the standard scRNA-seq pre-processing pipeline using Scanpy (Wolf, Angerer, and J. 2018; Satija et al. 2015). To create graphs from a cell by gene counts feature matrix, we used a batch-balanced, weighted kNN graph (Polański et al. 2019). BB-kNN constructs a kNN graph that identifies the top k nearest neighbors in each "batch", effectively aligning batches to remove sample-source bias while preserving biological variability (Luecken et al. 2020). We used annoy's method of approximate neighbor finding by calculating Euclidean distances between nodes in 50-dimensional PCA space. The PCA space is obtained by dimensionality-reduction (via principal component analysis) of the normalized and square-root transformed cell by gene counts matrix. Per "batch" we find $k = 3$ nearest neighbors, with edge weights given by the distance between corresponding nodes. An example BB-kNN graph is schematized in Figure 1A.

Single-cell Label Creation : For the COVID-19 patient dataset, all cells from each patient sample were given labels in accordance with that patient's COVID-19 severity (healthy, moderate, or severe). For the organoid dataset, cells with more than 10 transcripts aligned to the SARS-CoV-2 genome were considered to be infected. Cells in the 1, 2, and 3 days-post-infection (dpi) samples that were not infected are bystander cells. Mock is a control sample and can't be bystander or infected. The 3 timepoints were concatenated to the infection label per cell to yield 7 labels across the dataset (Mock, 1dpi-infected, 1dpi-bystander, and so on). Given the proximity of bronchoalveolar lavage fluid cells to the primary site of viral insult, we make the assumption that the transcriptomes of cells from a COVID-19 patient indicate response to disease. Thus, all cells from one patient have the same label. Similarly, we assume that all cells in an organoid culture inoculated with SARS-CoV-2 exhibit transcriptomic signatures associated with being an infected or bystander cell, distinct from mock-infected or control sample cells.

Model Performance : To generate train/test/val sets, we pooled all cells from a single dataset, then randomly assigned 70/15/15% of cells to train/test/val. We created a separate batch-balanced kNN graph for each split. To mini-batch the graphs, we used the Pytorch Geometric implementation of the ClusterData algorithm (Chiang et al. 2019). The validation set was used to select the model and the trained classifier was evaluated on the unseen test set. We evaluate the model based on node label accuracy. The negative log-likelihood loss is computed with respect to the the ground truth label of the nodes, derived from sample metadata (as described above).

Creating New Edge Features

In this section we describe our method to create new edge features.

Creating New Edge Features via Auxilliary Tasks

Predicting Louvain Clusters via GAT : We cluster our datasets using Louvain clustering (Blondel et al. 2008), and

Models	SARS-CoV-2 infected organoids	COVID-19 patients
ClusterGCN	65.43 (65.21-65.65)	89.26 (89.06-89.47)
ClusterGCN + DeepSet	79.75 (78.75-80.75)	87.2 (87.02-87.38)
ClusterGCN + Set2Set	71.65 (69.89-73.42)	88.34 (87.89-88.79)
ClusterGCN + Set Transformer	81.61 (79.34-83.87)	92.84 (91.95-93.74)
GAT	73.10 (70.93-75.27)	92.25 (91.27-93.24)
GAT + DeepSet	79.45 (77.98-80.92)	75.99 (74.8-77.68)
GAT + Set2Set	82.95 (81.75-84.15)	92.87 (92.62-93.12)
GAT + Set Transformer (Ours)	**89.8 (88.89-91.71)**	**95.12 (94.02-96.22)**
GIN + EdgeConv[1]	63.36 (62.53-64.19	89.56 (88.54-90.58)
ECC[1]	46.15 (34.72-57.59)	88.63 (86.07-91.20)

Table 2: Accuracy and 95% confidence intervals for $n = 6$ trials except for models marked with[1], where $n = 3$.

annotate these clusters as "cell types", as commonly done in single-cell analysis (Kiselev, Andrews, and Hemberg 2019). More information about these tasks, e.g., the number of clusters, can be found in the Appendix. Then, we use a 2-layer GAT with 8 attention heads in each layer to predict the cell type label. We extract the edge attention coefficients from the first layer of our trained model to use as edge features in our main node classification task. Thus we get an 8-dimensional edge feature vector by equation 12.

Predicting other Metadata Associated to our Graphs : All of our biological datasets have a batch or sample ID associated to it, i.e. some metadata that keeps track of the origin of the cell. We use the same method as before to create another 8-dimensional edge feature vector. More details and results about the auxiliary tasks can be found in the Appendix.

Creating Dataset Agnostic Features

In this subsection we quickly describe some other methods to create new edge features.

Forman-Ricci Curvature : We now use the internal geometry of the graph to create our next edge feature. We use the discrete version of the Ricci curvature as introduced by Forman (Forman 2003) and discussed in (Samal et al. 2018):

$$\begin{aligned} Ric_F(e) := \omega(e)\Bigg(&\frac{\omega(v_1)}{\omega(e)} + \frac{\omega(v_2)}{\omega(e)} \\ &- \sum_{e_{v_1}\sim v_1, e_{v_2}\sim v_2} \left[\frac{\omega(v_1)}{\sqrt{\omega(e)\omega(e_{v_1})}} + \frac{\omega(v_2)}{\sqrt{\omega(e)\omega(e_{v_2})}}\right]\Bigg) \end{aligned}$$

where e is an edge connecting the nodes v_1 and v_2, $\omega(e)$ is the weight of the edge e, $\omega(v_i)$ is the weight of the node, which we take to be 1 for simplicity, and $e_{v_i} \sim v_i$ is the set of all edges connected to v_i and *excluding* e. This is an intrinsic invariant that captures the local geometry of the graph and relates to the global property of the graph via a Gauss-Bonnet style result (Watanabe 2017). We found that our graphs are hyperbolic and most of the edges are negatively curved. As a future work, we would like to employ the methodologies introduced in (Albert, DasGupta, and Mobasheri 2014) to understand how the hyperbolicity affects higher order connectivities and the biological implications of such connectivities. We further hope that their methods would shed light on the most relevant paths between source and target nodes and to identify the most important nodes that govern these pathways.

Edge Features via Node2Vec : We use a popular embedding method called node2vec (Grover and Leskovec 2016) to embed the nodes in a 64 dimensional space. We then calculate the dot product between these node embeddings as a measure of similarity. However to be consistent with our other methods, we only compute the dot product between the nodes which share an edge. node2vec embeddings preserve the local community structure of a graph, which we expect should provide information to enhance discriminability between nodes, as previously suggested (Khosla, Setty, and Anand 2019).

Finally we concatenate all the created vectors into an 18 dimensional edge feature vector which we use in our main node classification task.

Experiments

Our main task is node classification in an inductive setting, as shown in Table 1. We compare our model and framework against popular GNN architectures like ClusterGCN (Chiang et al. 2019; Kipf and Welling 2017) and GAT (Veličković et al. 2018) as well as different set encoding frameworks like DeepSet (Zaheer et al. 2017) and Set2Set (Vinyals, Bengio, and Kudlur 2015). We also compare our model against GNN models that incorporate edge features like Graph Isomorphism Network, as modified in (Hu et al. 2019), and a Dynamic Edge Conditioned Convolution Network (ECC) (Simonovsky and Komodakis 2017; Gilmer et al. 2017). All the results shown are from the test set and our model's performance is reported in table 2. Our model outperforms the baseline models by a significant margin and also outperforms the other state-of-the-art networks and frameworks. Our processed data and code can be found at https://github.com/nealgravindra/self-supervsed_edge_feats.

Ablation Studies

In this section we sought to understand how our edge features affect model performance. A more detailed list of ablation studies can be found in the Appendix. From table 3,

Edge Feature	SARS-CoV-2 infected organoids	COVID-19 patients
Cluster label	.7137	.9211
Batch label	.8381	.9264
node2vec	.6976	.9111
Curvature	.7205	.9215
Cluster + batch label	.8449	.9689
node2vec + curvature	.6929	.9168
Cluster + batch label + node2vec	.8443	.9602
Cluster + batch label + curvature	.8438	.9605

Table 3: Ablation studies showing accuracy. Row names corresponding to the first column indicate which edge feature has been used with our model (GAT + Set Transformer).

we can see that edge features derived from the cell types and batch ID improve the model performance.

Discussion and Interpretability

We sought to gain insight into biological mechanism by extracting how our model learned to distinguish the different transcriptional signatures of SARS-CoV-2 infection dynamics and COVID-19 disease severity. We show the various aspects of model interpretability that we can glean from our model in Figure 2.

First, we extract the learned, edge attention coefficients from our Set Transformer and average these across attention heads to yield 1-dimensional edge weights. We use those edge weights to construct a new adjacency matrix. Then, with a cosine distance metric and this new adjacency matrix, we learn a new embedding of the cells (Figure 2A) using the UMAP algorithm (McInnes, Healy, and Melville 2018). In addition, to evaluate the importance of different types of edge features, we plot the average weights of the query matrix in the Set Transformer (see Appendix). Using the attention coefficients for manifold learning shows better separation of cells by cell type and label than typically used for embedding high-dimensional single-cell data (where the input for manifold learning is a cell by gene counts matrix), possibly because our model accounts for cell type variability via their edge feature representation. These embeddings may be useful in identifying unique, phenotypic subsets of cells. For example, in the new cell embedding of SARS-CoV-2 infected organoids, ciliated cells overlap with infected cells in a distinct and dense cluster. Indeed, it is thought that SARS-CoV-2 preferentially infects ciliated cells, which suggests that this type of model interpretability may have some utility (Ravindra et al. 2020b). In the new cell embedding of COVID-19 patient samples, the T cell population is mixed with cells derived from patients with mild and severe COVID-19, while a cluster predominantly comprised of cells derived from patients with severe COVID-19 is made up of macrophages and monocytes (2A, right). T cells and monocytes derived from macrophages are important in regulating the immune response and are targets for a number of therapies currently under development (Bersanelli 2020; Liao et al. 2020; Israelow et al. 2020). Furthermore, T cells are regulated by interferon signaling, which is itself a current COVID-19 therapeutic target (Meng et al. 2020). Taken together, this suggests that our model may identify cellular subsets worthy of further study to complement existing biomedical research.

After finding that T cells in severe COVID-19 patients may be hard to distinguish from healthy or mildly afflicted COVID-19 patients, we sub-clustered T cells from the test set and identified a prototypical T cell (cell nearest to cluster 10's centroid in UMAP space) in a cluster unique to COVID-19 patients with severe disease (2B). This allowed us to identify the most important features for predicting disease severity in this unique severe COVID-19 patient T cell cluster using a gradient-based approach (GNNExplainer) (Ying et al. 2019). Expected genes, i.e., genes thought to play a role in immunopathology associated with COVID-19 severity, arose in the top 20 most important genes, such as genes involved in interferon signaling and inflammation (IFNGR1, SLCO2B1). However, some novel genes also arose, for example, related to cardiac remodeling and metabolic regulation (NACA, ZNF586, PDPR, PRICKLE2, C2CD3, SGSM3, PARD6B, AL139819), which may suggest a unique response to SARS-CoV-2 infection or a cardiovascular component of severe COVID-19, the latter of which has been clinically suggested (Israelow et al. 2020; Richardson et al. 2020).

We also extract the learned weights (the matrices $\mathbb{W}^l$ for $l = 1, \cdots, 8$) from our models' first GAT layer and average them over 8 heads in order to globally investigate our model's feature saliency and indicate which genes are important in discriminating between transcriptomes of patients with varying degrees of COVID-19 severity and of lung cells with variable susceptibility to SARS-CoV-2 infection. In predicting COVID-19 severity from patient samples, our model gives high weight to genes involved in the innate immune system response to type I interferon (CCL2, CCL7, IFITM1), regulation of signaling (NUPR1, TAOK1, MTRNR2L12), a component of the major histocompatibility complex II (HLA-DQA2), which is important for developing immunity to infection, and a marker of eosinophil cells (RETN), a cell type involved in fighting parasites and a suspected source of immunopathology during COVID-19 (Israelow et al. 2020). In predicting SARS-CoV-2 infection, our model has saliency for viral transcript counts, which is encouraging, as well as genes that are involved in the inflammatory response and cell death (NFKBIA), as well as signaling (IFI27, HCLS1, NDRG1, NR1D1, TF), which may provide clues as to the dynamic regulatory response of cells in the host's lungs to SARS-CoV-2.

Conclusion

Here, we attempt to bring accurate disease state prediction to a molecular and cellular scale so that we can identify the cells and genes that are important for determining susceptibility to SARS-Cov-2 infection and severe COVID-19 illness via model interpretability. We achieved significant improvements in accuracy compared to other popular GNN

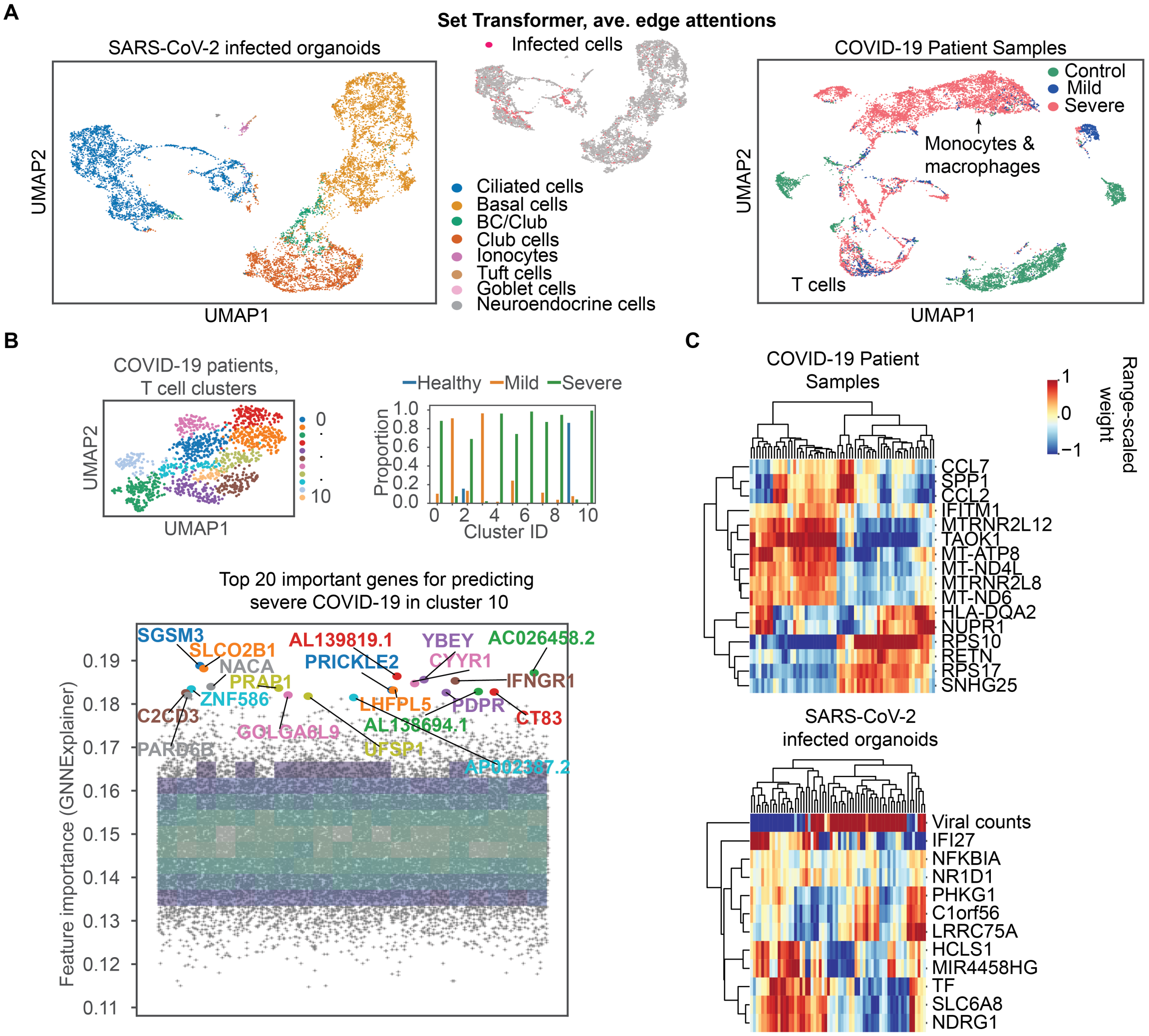

Figure 2: Model interpretability and the genes and cells important to COVID-19 severity and susceptibility to SARS-CoV-2 infection. (A) Embeddings learned from graphs constructed by averaging the edge attention coefficients across Set Transformer output dimensions, showing cell type or condition per cell. (B) Top 20 important genes for predicting a prototypical cell in a severe COVID-19 patient T cell cluster using GNNExplainer. If the proportion of points in a 20x20 grid is over 0.1, then point density is shown as color in a heatmap. (C) Top 5 important gene features for each GAT head, colored by learned weights.

architectures with our framework. Additionally, relative to vanilla GNNs, we achieve better separation of cells by cell type when visualizing the attention coefficients than possible with GATs alone. We also hypothesize that by computing edge features using the cell type and batch label, we control for these factors of variation in our main classification task and thus obtain potentially more meaningful features associated with COVID-19 than other models.

This suggests that using edge features derived from self-supervised learning can improve performance on disease-state classification from single-cell data. We used our models to gain insights into the cell tropism of SARS-CoV-2 and to elucidate the genes and cell types that are important for predicting COVID-19 severity. It is interesting that our model finds that genes involved in type I interferon signaling are important in predicting both COVID-19 severity and susceptibility to SARS-CoV-2 infection. It is suspected that dysregulation of type I interferon signaling may cause immunopathology during SARS-CoV-2 infection, leading to critical illness (Ravindra et al. 2020b; Israelow et al. 2020). Further study into the interaction partners and the subtle transcriptional differences between the genes and cells that we identified as important for prediction may provide complementary hypotheses or avenues for therapeutic intervention to mitigate the impacts of COVID-19. However, we are not medical professionals so we do *NOT* claim that interpretation of our model will bear any fruit. Rather, we hope that the approach of seeking state-of-the-art results on predicting disease states at single-cell resolution will enhance the study of biology and medicine and potentially accelerate our understanding of critical disease.

Acknowledgements

We thank Yale HPC for our use of their computing infrastructure. We thank the anonymous reviewers for helpful comments and suggestions. We also thank Mia Madel Alfajaro and Craig B. Wilen for generating the SARS-CoV-2 infected organoids dataset and sharing the data with us.

Ethical Statement

There are many caveats to our study. While we achieve good performance with our models, model interpretability in DL does not have a strong theoretical basis, and any proposed important features should merely be thought of as putative biological hypotheses. In addition, the cells in our datasets are derived from a relatively small patient population. While we attempt to limit sample-source bias by using a batch-balanced graph, we remain vulnerable to the idiosyncrasies of our samples. Thus, any putative hypotheses should only be considered meaningful after experimental validation.

References

Abu-El-Haija, S.; Perozzi, B.; and Al-Rfou, R. 2017. Learning Edge Representations via Low-Rank Asymmetric Projections. *Proceedings of the 2017 ACM Conference on Information and Knowledge Management* .

Alaa, A. M.; and van der Schaar, M. 2019. Attentive State-Space Modeling of Disease Progression. In *Advances in Neural Information Processing Systems 32*, 11338–11348. Curran Associates, Inc.

Albert, R.; DasGupta, B.; and Mobasheri, N. 2014. Topological implications of negative curvature for biological and social networks. *Physical Review E* 89(3).

Alquicira-Hernandez, J.; Sathe, A.; Ji, H. P.; Nguyen, Q.; and Powell, J. E. 2019. scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome Biology* 20(1): 264.

Amodio, M.; van Dijk, D.; Srinivasan, K.; Chen, W. S.; Mohsen, H.; Moon, K. R.; Campbell, A.; Zhao, Y.; Wang, X.; Venkataswamy, M.; Desai, A.; Ravi, V.; Kumar, P.; Montgomery, R.; Wolf, G.; and Krishnaswamy, S. 2019. Exploring single-cell data with deep multitasking neural networks. *Nature Methods* 16(11): 1139–1145.

Arisdakessian, C.; Poirion, O.; Yunits, B.; Zhu, X.; and Garmire, L. X. 2019. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome Biology* 20(1): 211.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. arXiv: 1607.06450.

Bersanelli, M. 2020. Controversies about COVID-19 and anticancer treatment with immune checkpoint inhibitors. *Immunotherapy* 12(5): 269–273.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10): P10008.

Chan, H. S.; Shan, H.; Dahoun, T.; Vogel, H.; and Yuan, S. 2019. Advancing Drug Discovery via Artificial Intelligence. *Trends in Pharmacological Sciences* 40(8): 592–604.

Chen, G.; Ning, B.; and Shi, T. 2019. Single-Cell RNA-Seq Technologies and Related Computational Data Analysis. *Frontiers in Genetics* 10: 317.

Chen, P.; Liu, W.; Hsieh, C.-Y.; Chen, G.; and Zhang, S. 2019. Utilizing Edge Features in Graph Neural Networks via Variational Information Maximization. arXiv: 1906.05488.

Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C.-J. 2019. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD ’19, 257–266. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016.

Choi, E.; Bahadori, M. T.; Song, L.; Stewart, W. F.; and Sun, J. 2017. GRAM: Graph-Based Attention Model for Healthcare Representation Learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’17, 787–795. New York, NY, USA: Association for Computing Machinery. ISBN 9781450348874.

Forman, R. 2003. Bochner’s Method For Cell Complexes And Combinatorial Ricci Curvature. *Discrete and Computational Geometry* 29: 323–374.

Franceschi, L.; Niepert, M.; Pontil, M.; and He, X. 2019. Learning Discrete Structures for Graph Neural Networks. *arXiv:1903.11960* .

Gao, Z.; Fu, G.; Ouyang, C.; Tsutsui, S.; Liu, X.; Yang, J.; Gessner, C.; Foote, B.; Wild, D.; Yu, Q.; and Ding, Y. 2018. edge2vec: Representation learning using edge semantics for biomedical knowledge discovery. arXiv:1809.02269.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *ICML’17*, 1263–1272. JMLR.

Gong, L.; and Cheng, Q. 2018. Exploiting Edge Features in Graph Neural Networks. arXiv: 1809.02709.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable Feature Learning for Networks. arXiv: 1607.00653.

Harikumar, H.; Quinn, T. P.; Rana, S.; Gupta, S.; and Venkatesh, S. 2019. A random walk down personalized single-cell networks: predicting the response of any gene to any drug for any patient.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2019. Strategies for Pre-training Graph Neural Networks. arXiv:1905.12265.

Hwang, B.; Lee, J. H.; and Bang, D. 2018. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental and Molecular Medicine* 50.

Israelow, B.; Song, E.; Mao, T.; Lu, P.; Meir, A.; Liu, F.; Madel Alfajaro, M.; Wei, J.; Dong, H.; Homer, R. J.; Ring, A.; Wilen, C. B.; and Iwasaki, A. 2020. Mouse model of SARS-CoV-2 reveals inflammatory role of type I interferon signaling. *bioRxiv* .

Khosla, M.; Setty, V.; and Anand, A. 2019. A Comparative Study for Unsupervised Network Representation Learning. IEEE Transactions on Knowledge and Data Engineering.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907.

Kiselev, V. Y.; Andrews, T. S.; and Hemberg, M. 2019. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics* 20(5): 273–282.

Lee, J.; Lee, Y.; Kim, J.; Kosiorek, A. R.; Choi, S.; and Teh, Y. W. 2018. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. arXiv: 1810.00825.

Liao, M.; Liu, Y.; Yuan, J.; Wen, Y.; Xu, G.; Zhao, J.; Cheng, L.; Li, J.; Wang, X.; Wang, F.; Liu, L.; Amit, I.; Zhang, S.; and Zhang, Z. 2020. Single-cell landscape of bronchoalveolar immune cells in patients with COVID-19. *Nature Medicine* .

Luecken, M.; Büttner, M.; Chaichoompu, K.; Danese, A.; Interlandi, M.; Mueller, M.; Strobl, D.; Zappia, L.; Dugas, M.; Colomé-Tatché, M.; and Theis, F. 2020. Benchmarking atlas-level data integration in single-cell genomics. bioRxiv.

Lundberg, S.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. arXiv: 1705.07874.

McInnes, L.; Healy, J.; and Melville, J. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426.

Meng, Z.; Wang, T.; Li, C.; Chen, X.; Li, L.; Qin, X.; Li, H.; and Luo, J. 2020. An experimental trial of recombinant human interferon alpha nasal drops to prevent coronavirus disease 2019 in medical staff in an epidemic area. medRxiv.

Michalak, T. P.; Aadithya, K. V.; Szczepanski, P. L.; Ravindran, B.; and Jennings, N. R. 2013. Efficient Computation of the Shapley Value for Game-Theoretic Network Centrality. *Journal of Artificial Intelligence Research* 46: 607–650.

Nguyen, T.; Le, H.; Quinn, T. P.; Le, T.; and Venkatesh, S. 2019. GraphDTA: Predicting drug–target binding affinity with graph neural networks. *bioRxiv* .

Polański, K.; Young, M. D.; Miao, Z.; Meyer, K. B.; Teichmann, S. A.; and Park, J.-E. 2019. BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics* 36(3): 964–965.

Ravindra, N.; Sehanobish, A.; Pappalardo, J. L.; Hafler, D. A.; and van Dijk, D. 2020a. Disease State Prediction from Single-Cell Data Using Graph Attention Networks. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL '20, 121–130. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370462.

Ravindra, N. G.; Alfajaro, M. M.; Gasque, V.; Wei, J.; Filler, R. B.; Huston, N. C.; Wan, H.; Szigeti-Buck, K.; Wang, B.; Montgomery, R. R.; Eisenbarth, S. C.; Williams, A.; Pyle, A. M.; Iwasaki, A.; Horvath, T. L.; Foxman, E. F.; van Dijk, D.; and Wilen, C. B. 2020b. Single-cell longitudinal analysis of SARS-CoV-2 infection in human bronchial epithelial cells. *bioRxiv* .

Richardson, S.; Hirsch, J. S.; Narasimhan, M.; Crawford, J. M.; McGinn, T.; Davidson, K. W.; ; and the Northwell COVID-19 Research Consortium. 2020. Presenting Characteristics, Comorbidities, and Outcomes Among 5700 Patients Hospitalized With COVID-19 in the New York City Area. *JAMA* 323(20): 2052–2059.

Samal, A.; Sreejith, R. P.; Gu, J.; Liu, S.; Saucan, E.; and Jost, J. 2018. Comparative analysis of two discretizations of Ricci curvature for complex networks. *Scientific Reports* 8(1).

Satija, R.; Farrell, J. A.; Gennert, D.; Schier, A. F.; and Regev, A. 2015. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology* 33(5): 495–502.

Seshadhri, C.; Sharma, A.; Stolman, A.; and Goel, A. 2020. The impossibility of low rank representations for triangle-rich complex networks. arXiv:2003.12635.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning Important Features Through Propagating Activation Differences. *CoRR* abs / 1704.02685.

Simonovsky, M.; and Komodakis, N. 2017. Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. arXiv:1704.02901.

Stuart, T.; and Satija, R. 2019. Integrative single-cell analysis. *Nature Reviews Genetics* 20(5): 257–272.

Sundararajan, M.; Taly, A.; and Yan, Q. 2018. Axiomatic Attribution for Deep Networks. *International Conference on Learning Represeations* .

Tenenbaum, J. B. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290(5500): 2319–2323.

Torroja, C.; and Sanchez-Cabo, F. 2019. Digitaldlsorter: Deep-Learning on scRNA-Seq to Deconvolute Gene Expression Data. *Frontiers in Genetics* 10: 978.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6000–6010. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. arXiv:1710.10903.

Vinyals, O.; Bengio, S.; and Kudlur, M. 2015. Order Matters: Sequence to sequence for sets. arXiv: 1511.06391.

Watanabe, K. 2017. Combinatorial Ricci curvature on cell-complex and Gauss-Bonnnet Theorem. arXiv:1703.08409.

Wolf, F. A.; Angerer, P.; and J., T. F. 2018. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology* 19(15): e–print.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 1–21. ArXiv: 1901.00596.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. *International Conference on Machine Learning* 5453–5462.

Yan, L.; Zhang, H.-T.; Goncalves, J.; Xiao, Y.; Wang, M.; Guo, Y.; Sun, C.; Tang, X.; Jing, L.; Zhang, M.; Huang, X.; Xiao, Y.; Cao, H.; Chen, Y.; Ren, T.; Wang, F.; Xiao, Y.; Huang, S.; Tan, X.; Huang, N.; Jiao, B.; Cheng, C.; Zhang, Y.; Luo, A.; Mombaerts, L.; Jin, J.; Cao, Z.; Li, S.; Xu, H.; and Yuan, Y. 2020. An interpretable mortality prediction model for COVID-19 patients. *Nature Machine Intelligence* 2(5): 283–288.

Ying, R.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. GNNExplainer: Generating Explanations for Graph Neural Networks. arXiv:1903.03894.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R.; and Smola, A. 2017. Deep Sets. arXiv:1703.06114.

Zheng, G. X. Y.; Terry, J. M.; Belgrader, P.; Ryvkin, P.; Bent, Z. W.; Wilson, R.; Ziraldo, S. B.; Wheeler, T. D.; McDermott, G. P.; Zhu, J.; Gregory, M. T.; Shuga, J.; Montesclaros, L.; Underwood, J. G.; Masquelier, D. A.; Nishimura, S. Y.; Schnall-Levin, M.; Wyatt, P. W.; Hindson, C. M.; Bharadwaj, R.; Wong, A.; Ness, K. D.; Beppu, L. W.; Deeg, H. J.; McFarland, C.; Loeb, K. R.; Valente, W. J.; Ericson, N. G.; Stevens, E. A.; Radich, J. P.; Mikkelsen, T. S.; Hindson, B. J.; and Bielas, J. H. 2017. Massively parallel digital transcriptional profiling of single cells. *Nature Communications* 8(1): 14049.

Zhong, J.; Tang, J.; Ye, C.; and Dong, L. 2020. The immunology of COVID-19: is immune modulation an option for treatment? *The Lancet Rheumatology* .