Gaining insight into SARS-CoV-2 infection and COVID-19 severity using self-supervised edge features and Graph Neural Networks

Arijit Sehanobish [*1] **Neal Ravindra** [*1] **David van Dijk** [1]

Abstract

Graph Neural Networks (GNN) have been extensively used to extract meaningful representations from graph structured data and to perform predictive tasks such as node classification and link prediction. In recent years, there has been a lot of work incorporating edge features along with node features for prediction tasks. In this work, we present a framework for creating new edge features, via self-supervised and unsupervised learning which we then use along with node features for node classification tasks. We validate our work on two biological datasets comprising of single-cell RNA sequencing data of *in vitro* SARS-CoV-2 infection and human COVID-19 patients. We demonstrate that our method achieves better performance over baseline Graph Attention Network (GAT) and Graph Convolutional Network (GCN) models. Furthermore, given the attention mechanism on edge and node features, we are able to interpret the cell types and genes that determine the course and severity of COVID-19, contributing to a growing list of potential disease biomarkers and therapeutic targets.

1. Introduction

Graph neural networks (GNN) have been widely used and developed for predictive tasks such as node classification and link prediction (1) and have been shown to learn from any sparse and discrete relational structure in data (2). GNNs typically use message passing, or recursive neighborhood aggregation, to construct a new feature vector for a particular node by collecting its neighbor's feature vectors (3; 4). However, most GNN schemes do not use edge features in learning new representations of graphical data. Recently, edge features have been incorporated into GNNs to harness information describing different aspects of the relationships between nodes (5; 6). However, there are very few frameworks for creating *de novo* edge feature vectors in a domain agnostic manner. In this article, using Graph Attention Networks, we propose a self-supervised learning framework to create new edge features which can be used to improve GNN performance in downstream node classification tasks.

Given 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 the disease, we sought to focus our efforts on COVID-19 datasets that can generate hypotheses related to these gaps (7; 8). Our focus on single-cell transcriptomic data relating to COVID-19 was motivated by recent work showing that Graph Attention Networks are effective at predicting disease states on an individual cell-to-cell basis (9). Single-cell RNA sequencing (scRNA-seq) is a technology that yields large datasets comprising many thousands of cells' gene expression in a variety of conditions (10; 11; 12). However, identifying factors important for determining an individual cell's pathophysiological trajectory or response to viral insult remains a challenge as single-cell data is noisy, sparse, and multi-dimensional (13; 14). We hope that our framework's performance could extract useful insights into the genes and cell types that might be important determinants of COVID-19 severity and SARS-CoV-2 infection.

2. Related works

There is a wealth of research on Graph Neural Networks. A significant amount of work has been focused on graph embedding techniques, representation learning and various predictive analyses using node features. There has been recent interest in using edge features to improve the performance of Graph Neural Networks (5; 15; 16). In this work we use an unique multi–tasking approach in creating these edge features.

Since our work focuses on biological applications, it becomes necessary to be able to interpret the results of our

[*] Equal contribution [1] Internal Medicine (Cardiology) and Computer Science, Yale University, New Haven, CT, USA. Correspondence to: Arijit Sehanobish <arijit.sehanobish@yale.edu>, Neal Ravindra <neal.ravindra@yale.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020. Copyright 2020 by the author(s).

network to inform further study of biology and medicine. One of the most common and popular ways to interpret machine learning models is via Shapley values (17) and it's various generalizations (18). However Shapley values require independence of features which is generally hard to guarantee. There is an excellent library (19) that allows us to interpret PyTorch models. In this work we follow the approach of (9; 20) in using attention mechanisms for interpretability. Thus even though set2set (21) is a popular mechanism to encode sets and has been previously used in the graph domain (22; 23), our view is that it is hard to interpret the hidden state of a LSTM. The transformer model (24; 25), on the other hand, allows us to interpret the results by looking at their attention heads.

Even though GNNs have been extensively used in biomedical research to predict medications, diagnoses, and outcomes from graphical representations of electronic health records (26), protein-protein and drug-protein interactions and molecular activity (27; 28; 29; 30) there is only one recent work that uses GAT models to predict the disease state of individual cells derived from clinical samples (9). However they do not consider multiple disease states or severity nor do they account for the confounding bias of batch effects. Here, we use the information contained within the dataset to construct edge features and a graph-structure that is balanced across batches, which reduces the bias of cell source while preserving biological variation (31). To the best of our knowledge, this is the first attempt to apply a GNN model to gain insight into multiple pathophysiological states, merging time-points, severity, and disease-state prediction into a multi-label node classification task from single-cell data.

3. Our Model

In this subsection we describe our model which consists of two components: (1) Set Transformer and (2) GAT layers.

3.1. Set Transformer

We use a Set Transformer as in (25). The Set Transformer is permutation invariant so it is an ideal architecture to encode sets. The building block of our Set Transformer is the multihead attention, as in (24). 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 for simplicity, let $d_q = d_v = d$, the multihead attention is computed by first projecting Q, K, V onto h different d_q^h, d_q^h, d_v^h dimensional vectors. For simplicity, let $d_q^h = d_v^h = \frac{d}{h}$, where h is the number of heads.

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

$$O_j := \text{softmax}\left(\frac{(QW_j^Q)(KW_j^K)^T}{\sqrt{\frac{d}{h}}}\right)VW_j^V$$

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:

$$STB(S) := \text{LayerNorm}(X + rFF(X)) \tag{1}$$

where $X = \text{LayerNorm}(S + \text{Multihead}(S, S, S))$. rFF is a row-wise feedforward layer and LayerNorm is layer normalization (32).

A Set Transformer takes as input a 3d tensor of the form [batch, seq-length, input-dim] and outputs 3d tensor of the form [batch, seq-length, output-dim], i.e. it outputs sets of the same size as the input sets. If, for a batch b_i, the set transformer outputs a set of the form $\{w_{i1}, ...w_{ij}\}$, we modify the output of the transformer to a fixed length vector

$$w_i := \sum_j \lambda_j w_{ij} \tag{2}$$

where λ_j are learnable weights. This step is necessary since our downstream tasks require vectors of fixed length.

3.2. Graph Attention Network

We also use the popular Graph Attention Network (GAT) (30). The input to a GAT layer are the node features and it outputs a new set of node features (of possibly different dimension). The heart of this layer is multi-head self-attention like in (24; 30). If h_i is the feature vector of node i, then self-attention is computed on the nodes

$$e_{ij}^l = a^l(\mathbb{W}^l h_i, \mathbb{W}^l h_j) \tag{3}$$

where $\mathbb{W}^l$ is a linear transformation called the weight matrix for the head l, a^l is a feedforward network outputting a scalar indexed by the head l. We then normalize these attention coefficients.

$$\alpha_{ij}^l = \text{softmax}_j(e_{ij}^l) = \frac{\exp(e_{ij}^l)}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik}^l)} \tag{4}$$

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 the features corresponding to them, to serve as the final output features for every node (after applying a nonlinearity, σ):

$$h_i^l = \sigma\Bigg(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^l \mathbb{W}^l h_j\Bigg). \tag{5}$$

We concatenate the features of these heads to produce a new node feature, $h_i' := || \, h_i^l$. However, for the final prediction layer, we average the representations over the heads and apply a logistic sigmoid non-linearity. Thus the equation for the final layer is : $h_i' = \sigma\left(\frac{1}{K}\sum_{l=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^l \mathbb{W}^l h_j\right)$, where K is the number of heads. We create new edge features Λ_{ij} for the node e_{ij} by concatenating the α_{ij}^l across all heads resulting in a K-dimensional edge feature, i.e.

$$\Lambda_{ij} := ||_{l=1}^{K}\alpha_{ij}^{l} \tag{6}$$

3.3. Our model

In this section we will describe our model that combines edge features, obtained as described above, with node features for our main node classification tasks. 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 a 64-dimensional node feature vector. For each node i, we construct a set $S_i := \{e_{ij} : j \in N_i\}$, where e_{ij} is the vector representing the edge features of the edge connecting 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 modified Set Transformer. We use 2 heads and 1 block of Set Transformer outputting a 8-dimensional vector. This 8-dimensional vector is concatenated with the 64-dimensional node representation. 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. Note that instead of GAT layers, we can also use GCN layers. Thus our method can be used by a variety of message passing architectures.

4. Experiments

Table 1. Dataset description showing train/val/test splits.

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

We validate our model on the following scRNA-seq datasets: (**I**) 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 (33); (**II**) 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 (34). Table 1 gives a summary of our datasets.

Creating new edge features : We cluster our datasets using Louvain clustering (35), and annotate these clusters as "cell types," as commonly done in single-cell analysis (14). 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 as edge features. Thus we get an 8-dimensional edge feature vector by equation 6. All of our biological datasets have a batch 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. We also use the Forman-Ricci curvature (36) as a measure of local geometry of the graph. Finally we use node2vec (37) to embed the nodes in 16-dimensional space and then we calculate the dot product between the node embeddings, which share an edge, as a measure of similarity. We concatenate the above vectors to create a 18 dimensional edge feature vector.

Table 2. Results of inductive tasks on single-cell datasets showing accuracy and 95% confidence intervals.

Models	SARS-CoV-2 infected organoids	COVID-19 patients	P (Welch's t-test)
GCN	65.43 (65.21-65.65)	89.26 (89.06-89.47)	-
GCN + Edge Features (Ours)	**81.61 (79.34-83.87)**	**92.84 (91.95-93.74)**	< 0.001
GAT	73.10 (70.93-75.27)	92.25 (91.27-93.24)	-
GAT + Edge Features (Ours)	**82.95 (81.75-84.15)**	**95.12 (94.02-96.22)**	< 0.001

5. Discussion

We achieved significant improvements using self-supervised edge features when comparing our model to the two popular GNN architectures, GCN and GAT. This suggests that using edge features derived from self-supervised learning and local graph information, with no requirement for handcrafted edge features, can improve graph neural network performance on challenging node classification tasks. Our models are simple, easy to train and can be used with various graph neural network architectures. Additionally, our edge feature creation framework is applicable to any graphical data. Given the excellent performance of our GAT with added edge features at predicting the disease-state at single-cell level, we sought to use the insight learned from our model to better understand SARS-CoV-2 infection dynamics and COVID-19 disease severity on an individual gene and cell basis. We show the various aspects of model interpretability that we can glean from our model in Figure 1. First, we extract the learned, edge attentions from our Set Transformer and average the coefficients across heads and dimensions to construct a new adjacency matrix. Then, using a cosine distance metric, we learn a new embedding of the cells (1A. In addition, to evaluate the importance of different types of edge features, we plot the average weights

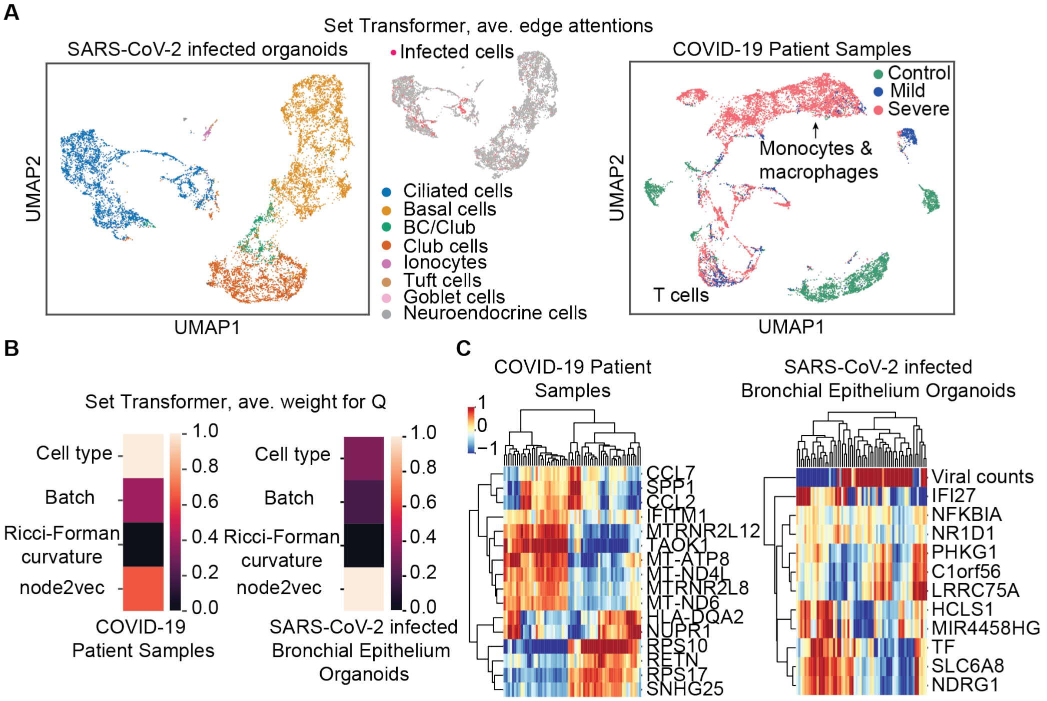

Figure 1. Model interpretability to generate hypotheses for genes and cells important to COVID-19 severity. (**A**) Embedding learned from graphs extracted from average edge attentions across Set Transformer output dimensions, showing cell type or condition per cell. (**B**) Relative importance of crafted edge features in disease state prediction tasks, averaged across the query layer from the Set Transformer. (**C**) Top 5 important gene features for each GAT head, colored by normalized, learned weights.

of the query layer in the set transformer (1B). Learning a manifold in this way shows better segregation of cell types than typically used for embedding high-dimensional single-cell data, possibly because our models give weight to cell type labels via their edge feature representation. In addition, we also extract the learned weights from our models' first GAT layer in order to investigate our model's feature saliency with respect to gene importance. 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, which are involved in fighting parasites (RETN). In predicting SARS-CoV-2 infection, our model finds saliency in counts of viral transcript, which is encouraging, as well as genes that are involved in inflammatory response and cell death (NFKBIA) and signaling (IFI27, HCLS1, NDRG1, NR1D1, TF), which may provide clues as to the dynamic regulatory response of cells in the lung's airways to SARS-CoV-2. The learned embedding for the SARS-CoV-2 infected organoids dataset highlights that our model segregates infected ciliated cells, which is the reported SARS-CoV-2 cell tropism, validating our models' interpretability (33). In predicting COVID-19 severity, it is interesting that our model learns to mix macrophages and monocytes in a predominantly severe patient cell cluster while cells derived from mild and severe COVID-19 patients are mixed in a T cell cluster. Monocytes derived from macrophages are thought to be enriched in severe COVID-19 cases and T cells are proposed targets for immune checkpoint therapy of COVID-19, despite lack of understanding as to the transcriptional differences between mild and severe COVID-19 illness (38; 34; 39). Lastly, it is interesting that our models find that genes involved in type I interferon signaling are important in predicting both COVID-19 severity and susceptibility to SARS-CoV-2 infection. Interferon signaling is involved in pro-inflammatory immune responses and it is suspected that type I interferon signaling may cause immunopathology during SARS-CoV-2 infection, leading to critical illness (33; 39). Further study into the interaction partners and the subtle transcriptional differences between the cells and cell types that we identified 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 study of biology and medicine and potentially accelerate our understanding of critical diseases.

References

[1] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2020. arXiv: 1901.00596.

[2] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning Discrete Structures for Graph Neural Networks. *arXiv:1903.11960*, May 2019.

[3] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation Learning on Graphs with Jumping Knowledge Networks. *International Conference on Machine Learning (ICML)*, pages 5453–5462, 2018.

[4] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[5] Liyu Gong and Qiang Cheng. Exploiting edge features in graph neural networks, 2018.

[6] Zheng Gao, Gang Fu, Chunping Ouyang, Satoshi Tsutsui, Xiaozhong Liu, Jeremy Yang, Christopher Gessner, Brian Foote, David Wild, Qi Yu, and Ying Ding. edge2vec: Representation learning using edge semantics for biomedical knowledge discovery, 2018.

[7] Li Yan, Hai-Tao Zhang, Jorge Goncalves, Yang Xiao, Maolin Wang, Yuqi Guo, Chuan Sun, Xiuchuan Tang, Liang Jing, Mingyang Zhang, Xiang Huang, Ying Xiao, Haosen Cao, Yanyan Chen, Tongxin Ren, Fang Wang, Yaru Xiao, Sufang Huang, Xi Tan, Niannian Huang, Bo Jiao, Cheng Cheng, Yong Zhang, Ailin Luo, Laurent Mombaerts, Junyang Jin, Zhiguo Cao, Shusheng Li, Hui Xu, and Ye Yuan. An interpretable mortality prediction model for COVID-19 patients. *Nature Machine Intelligence*, 2(5):283–288, 2020.

[8] Jixin Zhong, Jungen Tang, Cong Ye, and Lingli Dong. The immunology of COVID-19: is immune modulation an option for treatment? *The Lancet Rheumatology*, 2020.

[9] Neal Ravindra, Arijit Sehanobish, Jenna L. Pappalardo, David A. Hafler, and David van Dijk. Disease state prediction from single-cell data using graph attention networks. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL 20, page 121130, New York, NY, USA, 2020. Association for Computing Machinery.

[10] Grace X. Y. Zheng, Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, Tobias D. Wheeler, Geoff P. McDermott, Junjie Zhu, Mark T. Gregory, Joe Shuga, Luz Montesclaros, Jason G. Underwood, Donald A. Masquelier, Stefanie Y. Nishimura, Michael Schnall-Levin, Paul W. Wyatt, Christopher M. Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D. Ness, Lan W. Beppu, H. Joachim Deeg, Christopher McFarland, Keith R. Loeb, William J. Valente, Nolan G. Ericson, Emily A. Stevens, Jerald P. Radich, Tarjei S. Mikkelsen, Benjamin J. Hindson, and Jason H. Bielas. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):14049, April 2017.

[11] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental and Molecular Medicine*, 50, 2018.

[12] Tim Stuart and Rahul Satija. Integrative single-cell analysis. *Nature Reviews Genetics*, 20(5):257–272, MAY 2019.

[13] Geng Chen, Baitang Ning, and Tieliu Shi. Single-cell rna-seq technologies and related computational data analysis. *Frontiers in Genetics*, 10:317, 2019.

[14] Vladimir Yu Kiselev, Tallulah S. Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, May 2019.

[15] Pengfei Chen, Weiwen Liu, Chang-Yu Hsieh, Guangyong Chen, and Shengyu Zhang. Utilizing edge features in graph neural networks via variational information maximization, 2019.

[16] Sami Abu-El-Haija, Bryan Perozzi, and Rami Al-Rfou. Learning edge representations via low-rank asymmetric projections. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Nov 2017.

[17] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[18] T. P. Michalak, K. V. Aadithya, P. L. Szczepanski, B. Ravindran, and N. R. Jennings. Efficient computation of the shapley value for game-theoretic network centrality. *Journal of Artificial Intelligence Research*, 46:607650, Apr 2013.

[19] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Jonathan Reynolds, Alexander Melnikov, Natalia Lunova, and Orion Reblitz-

Richardson. Pytorch captum. https://github.com/pytorch/captum, 2019.

[20] Ahmed M. Alaa and Mihaela van der Schaar. Attentive state-space modeling of disease progression. In *Advances in Neural Information Processing Systems 32*, pages 11338–11348. Curran Associates, Inc., 2019.

[21] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets, 2015.

[22] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML17, page 12631272. JMLR.org, 2017.

[23] Han Altae-Tran, Bharath Ramsundar, Aneesh S. Pappu, and Vijay Pande. Low data drug discovery with one-shot learning, 2016.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS17, page 60006010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[25] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks, 2018.

[26] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun. 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, page 787795, New York, NY, USA, 2017. Association for Computing Machinery.

[27] Thin Nguyen, Hang Le, Thomas P. Quinn, Thuc Le, and Svetha Venkatesh. GraphDTA: Predicting drug-target binding affinity with graph neural networks. *bioRxiv*, 2019.

[28] H.C. Stephen Chan, Hanbin Shan, Thamani Dahoun, Horst Vogel, and Shuguang Yuan. Advancing Drug Discovery via Artificial Intelligence. *Trends in Pharmacological Sciences*, 40(8):592–604, 2019.

[29] Haripriya Harikumar, Thomas P. Quinn, Santu Rana, Sunil Gupta, and Svetha Venkatesh. A random walk down personalized single-cell networks: predicting the response of any gene to any drug for any patient, 2019.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[31] Md Luecken, M Bttner, K Chaichoompu, A Danese, M Interlandi, Mf Mueller, Dc Strobl, L Zappia, M Dugas, M Colom-Tatch, and Fj Theis. Benchmarking atlas-level data integration in single-cell genomics. *bioRxiv*, 2020.

[32] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[33] Neal G. Ravindra, Mia Madel Alfajaro, Victor Gasque, Jin Wei, Renata B. Filler, Nicholas C. Huston, Han Wan, Klara Szigeti-Buck, Bao Wang, Ruth R. Montgomery, Stephanie C. Eisenbarth, Adam Williams, Anna Marie Pyle, Akiko Iwasaki, Tamas L. Horvath, Ellen F. Foxman, David van Dijk, and Craig B. Wilen. Single-cell longitudinal analysis of sars-cov-2 infection in human bronchial epithelial cells. *bioRxiv*, 2020.

[34] Mingfeng Liao, Yang Liu, Jing Yuan, Yanling Wen, Gang Xu, Juanjuan Zhao, Lin Cheng, Jinxiu Li, Xin Wang, Fuxiang Wang, Lei Liu, Ido Amit, Shuye Zhang, and Zheng Zhang. Single-cell landscape of bronchoalveolar immune cells in patients with COVID-19. *Nature Medicine*, 2020.

[35] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, Oct 2008.

[36] Robin Forman. Bochner's method for cell complexes and combinatorial ricci curvature. *Discrete and Computational Geometry*, 29:323–374, 2003.

[37] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016.

[38] Melissa Bersanelli. Controversies about covid-19 and anticancer treatment with immune checkpoint inhibitors. *Immunotherapy*, 12(5):269–273, 2020.

[39] Benjamin Israelow, Eric Song, Tianyang Mao, Peiwen Lu, Amit Meir, Feimei Liu, Mia Madel Alfajaro, Jin Wei, Huiping Dong, Robert J Homer, Aaron Ring, Craig B Wilen, and Akiko Iwasaki. Mouse model of sars-cov-2 reveals inflammatory role of type i interferon signaling. *bioRxiv*, 2020.

[40] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks, 2019.

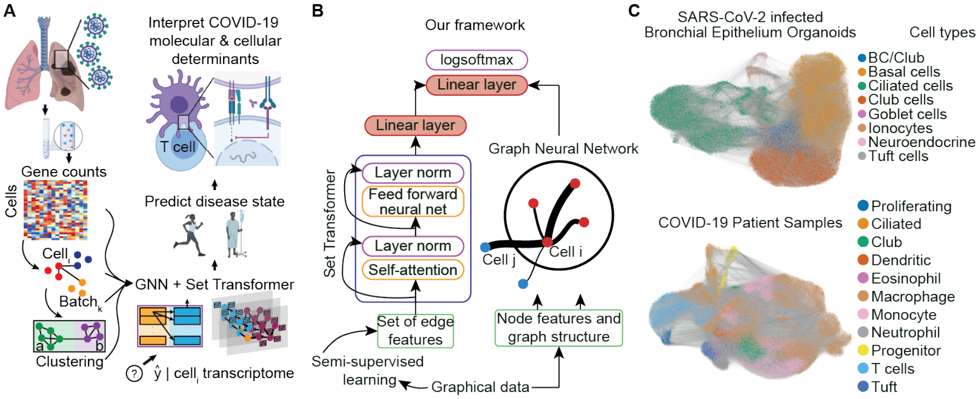

Figure 2. Our framework and datasets of interest. (**A**) Overview of our approach with respect to gaining molecular and cellular insights into COVID-19. (**B**) Our framework and models' architecture, integrating edge features with GNNs via a Set Transformer. (**C**) Graphical data used, showing cell types for each cell and edges in a node-feature, dimension-reduced embedding.

A. Model Figure

The figure 2 shows (A) our approach to the problem, (B) an overview of the model description, (C) a low dimensional embedding of our graphs.

B. Auxillary tasks

Table 3. Experimental tasks

Task	SARS-CoV-2 infected organoids	COVID-19 patients
Louvain cluster ID	Cell type	Cell type
Batch or node metadata	Culture sample ID	Patient ID
Inductive prediction	Timepoint and infection	No, Mild, or Severe Disease

Table 4. Number of labels for auxiliary tasks

Task	SARS-CoV-2 infected organoids	COVID-19 patients
Cell type	8	10
Batch	4	12

C. Hyperparameters and Training details

Table 5. Default hyperparameters used in the experiments

	Graph Attention Network	Graph Convolution Network
Number of layers	2	2
Hidden_size	8	256
Attention Heads	8	N/A
Optimizer	Adagrad	Adagrad
weight_decay	.0005	.0005
Batch size	256	256
Dropout	.5	.4
Slope in LeakyRelu	.2	.2
Training Epochs	1000	1000
Early stopping	100	100

For auxiliary tasks and for training our models, we break our graph into 5000 subgraphs using the ClusterData function in PyTorch Geometric library and then minibatched the subgraphs using the ClusterData function. These algorithms are originally introduced in (40). We used a single block of Set Transformer with input dimension 18, output dimension 8 and 2 heads.