N-ACT: An Interpretable Deep Learning Model for Automatic Cell Type and Salient Gene Identification

A. Ali Heydari [1 2 ‡] **Oscar A. Davalos** [2 3 ‡] **Katrina K. Hoyer** [2 4] **Suzanne S. Sindi** [1 2]

Abstract

Single-cell RNA sequencing (scRNAseq) is rapidly advancing our understanding of cellular composition within complex tissues and organisms. A major limitation in most scRNAseq analysis pipelines is the reliance on manual annotations to determine cell identities, which are time consuming, subjective, and require expertise. Given the surge in cell sequencing, supervised methods–especially deep learning models–have been developed for automatic cell type identification (ACTI), which achieve high accuracy and scalability. However, all existing deep learning frameworks for ACTI lack interpretability and are used as "black-box" models. We present *N-ACT* (Neural-Attention for Cell Type identification): the first-of-its-kind interpretable deep neural network for ACTI utilizing neural attention to detect salient genes for use in cell-types identification. We compare N-ACT to conventional annotation methods on two previously manually annotated data sets, demonstrating that N-ACT accurately identifies marker genes and cell types in an unsupervised manner, while performing comparably on multiple data sets to current state-of-the-art model in traditional supervised ACTI.

1. Introduction

Single-cell RNA-sequencing (scRNAseq) technologies allow for measuring transcriptome-wide gene expression at the single-cell level. In contrast to bulk-RNA sequencing, scRNAseq can elucidate dynamic expression patterns between different cellular populations, providing a tremendous advantage when studying organisms as well as delineating intra-population heterogeneities (Erfanian et al., 2022).

[‡]Equal contribution [1]Department of Applied Mathematics, University of California [2]Health Sciences Research Institute [3]Quantitative and Systems Biology Graduate Program [4]Department of Molecular and Cell Biology, University of California, Merced, USA. Correspondence to: Suzanne Sindi <ssindi@ucmerced.edu>.

The 2022 ICML Workshop on Computational Biology.

Accurate identification of cell types in scRNAseq studies remains a challenging and time-consuming task (Pasquini et al., 2021). Cell type annotation is often performed manually by experts – a long, laborious, and subjective process (Pasquini et al., 2021; Clarke et al., 2021). To mitigate these challenges, researchers have developed automatic cell type identification (ACTI) pipelines, including deep learning (DL) models such as ACTINN (Ma & Pellegrini, 2019), which learn from datasets that have *already been annotated from the same (or similar) populations* (see (Pasquini et al., 2021) for a review on ACTI). However, the supervision required for these approaches limits their utility for studies which lack prior knowledge of tissue- or sample-specific cell types. Such models can not provide the much-needed biological interpretability on the algorithm's decision-making needed to translate scRNAseq findings to inform experimental design.

In this work, we introduce *Neural-Attention for Cell Type identification (N-ACT)*: An interpretable *unsupervised* DL model that employs *attention* to detect salient genes, and applies this information to identify specific cell-types. Our results on multiple datasets show that N-ACT accurately predicts preliminary annotations with no prior knowledge about the system, providing a valuable complementary framework to experimental studies and computational pipelines.

2. Methods and Approach

N-ACT's framework consists of three stages (shown in Fig. 1): (I) Assigning labels (if no labels are available) and identifying highly-variable genes (HVGs), (II) a deep neural network (NN), which is the core of N-ACT, and (III) interpretation. In this section, we present stage II, i.e. N-ACT's DL core, and later describe stages (I) and (III) in Section 4. N-ACT's DL core has three modules: (1) *Additive attention module*, responsible for learning the importance of each gene; (2) *Multi-headed projection module* tasked with learning a set of non-linear operators mapping attention outputs to the last layer, *i.e.* (3) *Task module*, which is designed to fulfill a specific downstream task.

2.1. Additive Attention

Attention (or neural-attention) is a recent DL mechanism that has transformed computer vision and natural language

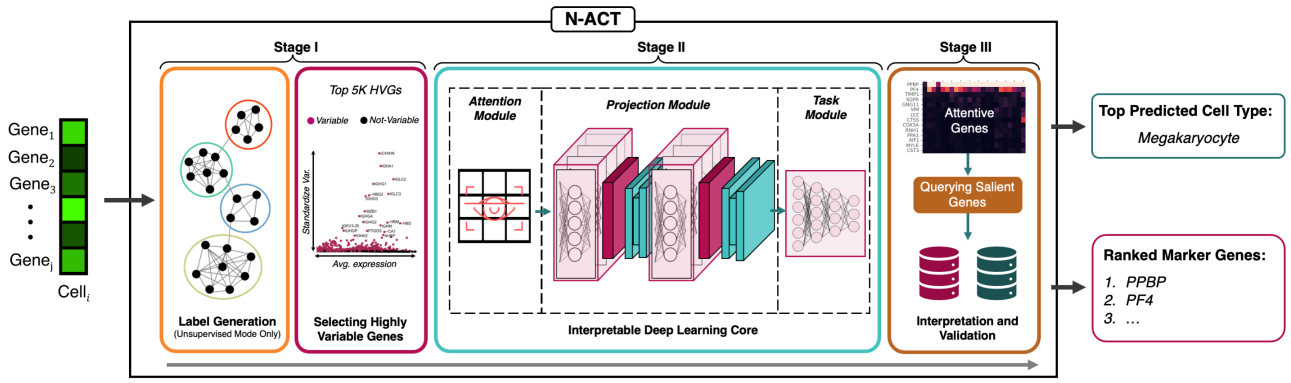

Figure 1: **N-ACT, an Interpretable Model for Unsupervised/Supervised Cell-Type Identification.** N-ACT consists of flexible stages and modules that can be modified for different objectives. We focus on unsupervised ACTI, which consists of the following stages: *(Stage I)* Generating labels through graph-based clustering followed by identifying five thousand highly-variable genes for efficient training (standard practice in scRNAseq pipelines), *(Stage II)* Training the N-ACT DL core to predict generated (or true) labels, with the goal of identifying salient genes to predict the correct cell types (labels), and *(Stage III)* Interpreting model predictions by extracting attention values, thus constructing a ranked list of "attentive genes" used to compare to existing literature (referred to as "Querying Salient Genes" in the figure) and thus predicting cell types. We describe each component in more detail in the Appendix.

processing research (refer to (Chaudhari et al., 2021) for a review on attention in DL). Attention networks aim to mimic the way humans understand "context" in sentences or details in images by focusing on a subset of significant features for a given objective. The use of attention-based NN for scRNAseq analysis is still in its infancy, with only a few successful applications to date. To identify salient genes, we use an additive attention module (Bahdanau et al., 2015) in a feed-forward NN (similar to (Raffel & Ellis, 2015)) aiming to learn the optimal weighting (*importance*) of all genes for each cell, given a downstream task.

The first step in the DL core (attention) is used to calculate a gene-score matrix (weighted version of scRNAseq count matrix), representing expression data in later layers. These importance scores enable gene prominence quantification for the downstream task, allowing for interpretation of model decision making. Given a gene expression matrix $X \in \mathbb{R}^{C\times N}$, where C and N denote the number of cells and genes, respectively, we define the gene-score matrix Γ and the attention weights A as shown in Eq. (1):

$$\Gamma = A \odot X, \text{where } A_{i,j} = \frac{e^{L_{i,j}}}{\sum_{j=1}^{N} e^{L_{i,j}}}, \qquad (1)$$

with $L = NN(X)$ denoting a linear NN[1]. The learned operator A is leveraged after training to identify important (or "attentive") genes for interpretability.

2.2. Multi-Headed Projections

Our *Projection* mechanisms are intermediate layers between the attention layer and the downstream task module. The goal in using projection modules is to strike a balance between model capacity and efficiency: Too much capacity (*e.g.* too many non-linear layers) could lead to significant over-fitting, while insufficient capacity prevents the model from learning the correct representations. We design the projection blocks to be *multi-headed* (consisting of $h \in \mathbb{N}$ separate linear operators), a concept shown by (Vaswani et al., 2017) as effective in learning different representations. Such design allows efficient consideration of different gene subsets and improves model performance without the need for numerous non-linear layers. N-ACT consists of k projection blocks, each consisting of h heads. Outputs from each head is then concatenated and inputted to a point-wise feed-forward network (equivalent to 1×1 convolution layer) with a Rectified Linear Unit (ReLU) (Nair & Hinton, 2010) that adds non-linearity. Through careful ablation studies, we found that $k = 2$ projection blocks with $h = 10$ heads provides the appropriate balance of accuracy and efficiency. Details on projection blocks and relevant ablation studies are provided in Appendix F.

2.3. Task Module and Architecture Choices

The last stage of our DL core is the *task module*, which can be adjusted based on desired objectives. Given our ACTI goal, we chose a non-linear mapping between the projection block's output and the labels (either provided [supervised] or generated in the earlier stages [unsupervised]).

Our task module connects the projections to the number of labels, followed by Leaky ReLU activation (Xu et al., 2015) (depicted in Appendix F). N-ACT minimizes a standard cross entropy loss (Appendix C) using the Adam gradient-based optimizer (Kingma & Ba, 2014) at a learning rate $\alpha = 10^{-4}$ for 50 epochs. Additional information on N-ACT's training scheme is provided in Appendix F.

[1] $NN : \mathbb{R}^{C\times N} \to \mathbb{R}^{C\times N}$ is a linear operator of the form $NN(X) = XW + B$, with input X and biases $B \in \mathbb{R}^{C\times N}$ and weights $W \in \mathbb{R}^{N\times N}$.

3. Datasets Studied

We tested the model's ability to learn cell-types on four datasets (two for supervised and two for unsupervised ACTI). Results are presented on different datasets for each learning setting to showcase N-ACT's versatility and utility for different systems and species (similar results were achieved on all datasets in all tasks). Data were minimally pre-processed (only quality-controlled) and divided roughly 85%:15% for training and testing using "balanced split" (described in (Heydari et al., 2022)). A brief description of each dataset is provided below, with more details on pre-processing and the data presented in Appendix B.

Datasets for supervised training: (1) ***Mouse HDF*** (PubMed ID: 34548614) consists of scRNAseq of murine aortic cells of mice on a normal diet versus mice on a high-fat diet, resulting in 24K cells and 10 annotated populations after processing. (2) ***Immune CSF*** (PubMed ID: 33382973) profiles single-cells from cerebrospinal fluid (CSF) of immune CSF, viral encephalitis, and non-inflammatory and autoimmune neurological disease. Cells were isolated from 31 patients, resulting in 80K cells (70K cells after processing) and 15 annotated populations.

Datasets for unsupervised training: (1) ***COVID PBMC*** (PubMed ID: 33357411) profiles the transcriptional immune dysfunction triggered in moderate and severe COVID-19 patients using scRNAseq. Peripheral blood mononuclear cells (PBMC) were isolated from 20 patients and were sequenced resulting in 69K cells (64K cells after processing) with 9 cell populations. (2) ***Immune cSCC*** (Pub Med ID: 32579974) consists of scRNAseq from healthy skin and cutaneous squamous cell carcinoma (cSCC) tumors. 10 patients with cSCC tumors had healthy skin and tumor cells sequenced, resulting in 48K cells (47K cells after processing) and 14 cell populations.

4. Results

In this section, we provide the results of using N-ACT for datasets described in Section 3. Standard evaluation tools were used to measure model performance with each metric detailed in Appendix C.

4.1. Supervised ACTI Performance

We benchmark N-ACT against the current state-of-the-art supervised model, ACTINN, though the goal of this work is to generate unsupervised annotations. To show the importance of our novel architecture in effective attention utilization, we added the same feed-forward attention module to ACTINN (denoted by "ACTINN+ATTN"), which significantly hindered the model's performance (Table 1). Given the comparable performance of our model to the state-of-the-art DL algorithm, our results show that N-ACT can effectively learn supervised ACTI. Moreover, the poor performance of ACTINN + ATNN highlights the importance of appropriate architectures needed for effective use of attention for interpretability. We next consider unsupervised cell-type annotation on two previously annotated datasets.

Table 1: **Benchmarking N-ACT on Supervised ACTI**. Although the main goal of N-ACT is interpretable ACTI in an unsupervised manner, our model can be used in a supervised setting as well, while still providing biological interpretability. *W-F1*: Weighted F1 score, *NW-F1*: Non-Weighted F1 score, *Interp*: Interpretability

MODEL	W-F1	NW-F1	INTERP?
MOUSE HDF			
ACTINN	**0.9703**	0.9677	NO
ACTINN+ATTN	0.8759	0.7438	YES
N-ACT (OURS)	0.9681	**0.9712**	YES
IMMUNE CSF			
ACTINN	**0.9357**	0.8898	NO
ACTINN+ATTN	0.7528	0.2413	YES
N-ACT (OURS)	0.9285	**0.8963**	YES

4.2. Unsupervised ACTI Performance

Unsupervised label generation: Given that the cell types are not known *a priori*, labels must be generated before training the DL core. To do so, we choose to perform unsupervised clustering using the Leiden algorithm (Traag et al., 2019), a standard scRNAseq clustering technique in many pipelines (Heydari & Sindi, 2022), allowing label generation without supervision. All results shown in this section follow this approach. However, given the requisite comparison against the true annotations for this work, we ensure the number of clusters generated by our model is equal to the number of annotated populations from the original publication, without enforcing any additional constraints.

Finding salient genes: To identify *attentive* (salient) genes in each population, we leverage the attention scores calculated by our model: First, we calculate the mean attention score per cell type for each gene. To analyze the performance of our model in assigning importance to various genes, we investigate the correlation between the mean attention scores for all genes in each cell type (Fig. 2**(A),(C)**). We find a low correlation for those populations that are not closely related (*e.g.* dendritic and endothelial cells in Fig. 2**(A)**), and a high correlation between those that are related (*e.g.* $CD4^+$ and $CD8^+$ T cells in Fig. 2**(C)**). These results indicate that N-ACT has accurately learned to assign importance to the same gene sets across similar populations, as desired.

Next, we use mean attention scores per cluster and selected the top 100 genes with the highest average values, constructing an object that includes the top 100 genes for each cell per cluster. Using this object, we calculate term frequency (TF)-inverse document frequency (IDF) for each gene in the object. TF-IDF is a standard natural language processing technique that weights the importance of a word in a document (see Appendix C for more information). In this

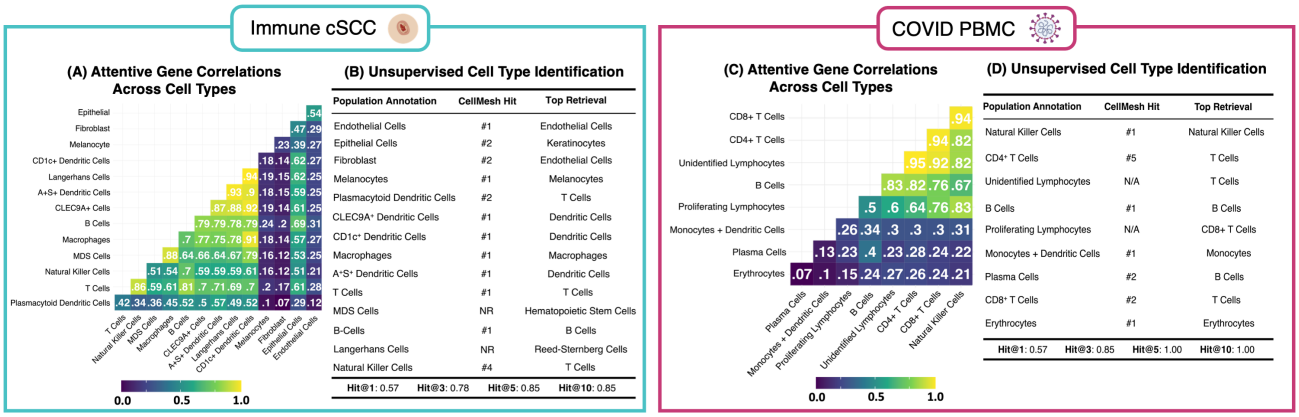

Figure 2: **Evaluating N-ACT's Unsupervised ACTI**. **[(A), (C)]:** To evaluate the interpretability and the performance of the attentive genes, we calculated the Pearson correlations of mean attention values in each population (per gene) for all cell types in **(A)** Immune cSCC and **(C)** COVID PBMC. **[(B), (D)]:** To assess N-ACT's predictive capabilities, we queried the TF-IDF-ranked attentive genes from CellMeSH and recorded the placement when the predicted cell type matched the actual annotation (called a "hit"). To interpret our model's mistakes, we also retrieved the first prediction from CellMeSH, which we include as "Top Retrieval".

formulation, each row of the object (containing top genes for all cells in a cluster) is a document used to calculate TF and IDF. The TF-IDF values for each gene were then multiplied by the original attention values, providing the final saliency scores. TF-IDF normalization down-weights common housekeeping genes that frequently appear in each cell population but are less useful in identifying cell-types. Lastly, we re-rank genes based on these weighted scores and select the top 25 as the attentive genes, which we use for cell-type identification.

Identifying cell types: Once attentive genes are identified, various techniques for finding their corresponding cell-types can be applied. To showcase the accuracy and automation capabilities of our model, we queried the attentive genes using CellMeSH (Mao et al., 2021), a probabilistic cell type querying tool that uses a database built from indexed literature to map marker genes to probable cell types (CellMeSH details provided in Appendix D). It is important to note that the bias in each database can affect results, and that other specialized databases or methods can further improve the identification process (see Appendix G). Fig. 2**(B)** and 2**(D)** present the accuracy of cell-type prediction using N-ACT-identified salient genes for Immune cSCC and for COVID PBMC, respectively. These results demonstrate that N-ACT accurately identifies attentive genes that are known markers for the underlying populations without any prior knowledge of the system or species.

5. Conclusions and Discussion

In this work, we presented N-ACT, the first-of-its-kind interpretable DL model for ACTI. We show that N-ACT effectively identifies cell-types, in a supervised and, more importantly, unsupervised manner. N-ACT is a first attempt at providing interpretability in this context, and we believe our improvements and developments reduce subjectivity while significantly minimizing the time needed for annotating scRNAseq datasets. Optimizing the scRNAseq annotation process will accelerate translational and basic research by enabling scientists to focus on the underlying biological questions. Our results demonstrate that N-ACT accurately identifies salient genes that are known markers for the underlying populations, without prior knowledge of the system or species. Moreover, the interpretability of our framework is useful for predicting the correct cell-types, and for better understanding the data when there is ambiguity (*e.g.* Appendix G) or when the model makes mistakes. As such, N-ACT provides a powerful tool for facilitating discovery, even if its top prediction is ultimately incorrect. Using our model, cells can be assigned cell-types by new users without prior expertise in the given system, within minutes. From these conclusions, we hypothesize that attention can be further utilized to identify unique relationships between different genes and cells, which would not otherwise be apparent. Despite successful application of DL in scRNAseq space, most DL models are not interpretable. Biologically-interpretable DL models, such as N-ACT, can provide crucial information on the algorithm's decision making, while assisting scientists in understanding underlying complex biological networks.

Code and Data Accessibility

All source code and reproducibility/tutorial notebooks, alongside download links to trained models and datasets, are available at `https://github.com/SindiLab/NACT`.

Acknowledgements

The authors were supported from the National Institutes of Health (R15-HL146779 & R01-GM126548), National Science Foundation (NSF) (DMS-1840265) and University of California (UC) Office of the President and UC Merced COVID-19 Seed Grant. Computational resources were supported by the NSF, Grant No. ACI-2019144 and in part through NSF awards CNS-1730158, ACI-1540112 and ACI-1541349. Due to space constraints, complementary citations are included in the Appendix.

References

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Chaudhari, S., Mithal, V., Polatkan, G., and Ramanath, R. An attentive survey of attention models. *ACM Trans. Intell. Syst. Technol.*, 12(5), oct 2021. ISSN 2157-6904. doi: 10.1145/3465055. URL https://doi.org/10.1145/3465055.

Clarke, Z. A., Andrews, T. S., Atif, J., Pouyabahar, D., Innes, B. T., MacParland, S. A., and Bader, G. D. Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods. *Nature Protocols*, 16(6):2749–2764, 2021. doi: 10.1038/s41596-021-00534-0. URL https://doi.org/10.1038/s41596-021-00534-0.

Erfanian, N., Heydari, A. A., Iañez, P., Derakhshani, A., Ghasemigol, M., Farahpour, M., Nasseri, S., Safarpour, H., and Sahebkar, A. Deep learning applications in single-cell omics data analysis. *bioRxiv*, 2022. doi: 10.1101/2021.11.26.470166. URL https://doi.org/10.1101/2021.11.26.470166.

Heydari, A. A. and Sindi, S. S. Deep learning in spatial transcriptomics: Learning from the next next-generation sequencing. *bioRxiv*, 2022. doi: 10.1101/2022.02.28.482392. URL https://doi.org/10.1101/2021.11.26.470166.

Heydari, A. A., Davalos, O. A., Zhao, L., Hoyer, K. K., and Sindi, S. S. ACTIVA: realistic single-cell RNA-seq generation with automatic cell-type identification using introspective variational autoencoders. *Bioinformatics*, 38(8):2194–2201, 02 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac095. URL https://doi.org/10.1093/bioinformatics/btac095.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Ma, F. and Pellegrini, M. ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics*, 36(2):533–538, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz592. URL https://doi.org/10.1093/bioinformatics/btz592.

Mao, S., Zhang, Y., Seelig, G., and Kannan, S. CellMeSH: probabilistic cell-type identification using indexed literature. *Bioinformatics*, 38(5):1393–1402, 12 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab834. URL https://doi.org/10.1093/bioinformatics/btab834.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Pasquini, G., Rojo Arias, J. E., Schäfer, P., and Busskamp, V. Automated methods for cell type annotation on scrna-seq data. *Computational and Structural Biotechnology Journal*, 19:961–969, 2021. ISSN 2001-0370. doi: https://doi.org/10.1016/j.csbj.2021.01.015. URL https://www.sciencedirect.com/science/article/pii/S2001037021000192.

Raffel, C. and Ellis, D. P. W. Feed-forward networks with attention can solve some long-term memory problems, 2015. URL https://arxiv.org/abs/1512.08756.

Traag, V. A., Waltman, L., and van Eck, N. J. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019. doi: 10.1038/s41598-019-41695-z. URL https://doi.org/10.1038/s41598-019-41695-z.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Xu, B., Wang, N., Chen, T., and Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Appendix and Supplementary Material

Appendix A. Computational Environment

Model development and testing was performed in `Python` (v 3.9.7) and data processing was performed in `R` (v. 4.1.2) (detailed in section B). Our models were developed and tested on A100 Nvidia GPUs, and data pre-processing on a 14-inch Macbook Pro with Apple M1 Max and 64 GB RAM. Data I/O was done in `Scanpy` (v. 1.7.0) (Wolf et al., 2018). The DL core of our model was developed in `PyTorch` (v. 1.9.1); however, developing/testing N-ACT on A100 GPUs required installation of a specific version of `PyTorch` (`torch==1.9.0+cu111`), which is provided in N-ACT's package repository. A complete list of requirements of `Python` packages is also available in N-ACT's GitHub repository: `https://github.com/SindiLab/NACT`.

Appendix B. Datasets Studied and Pre-Processing Workflow

B. 1. Pre-Processing

All data (count matrices and manual annotations) are publicly available from NCBI gene expression omnibus (GEO) and the Broad Institute Single Cell Portal (SCP), with links provided below. Datasets were processed using the `Seurat` package (v. 4.1.0) in `R` (Butler et al., 2018). Manual annotations were merged with count matrices using a variety of `tidyverse` (v. 1.3.1) functions, and subsequently added to `Seurat` object as metadata. Data filtering consisted of the standard practice of removing cells with fewer than 200 expressed genes and removing genes present in fewer than 3 cells. Next, we retained cells with less than 10% mitochondrial reads to mitigate cellular debris. Lastly, cell types containing less than 100 cells were removed and excluded from the dataset.

After filtering, we identified the top 5,000 highly variable genes (HVG's) for each dataset (HVG procedure is detailed in Appendix C. 3). To minimize the biological and technical effects in each dataset based on patient and/or biological conditions such as normal versus disease state, we utilized `Harmony` (v 0.1.0) to perform integration (Korsunsky et al., 2019). Resulting Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) plots for each dataset with original annotations are provided in Fig. A.3. `SeuratDisk` (v. 0.0.0.9019) was used to convert the Seurat data object into an AnnData object compatible with `Scanpy`. To perform clustering and generate cell labels (in the unsupervised case), we used `Scanpy`'s pipeline for clustering (consisting of dimensionality reduction using principal component analysis (PCA), followed by Leiden clustering). As mentioned in the main manuscript, we found Leiden resolutions that led to the same number of clusters as the annotated populations in order to compare our predictions to the ground truth labels.

B. 2. Datasets

To evaluate N-ACT capabilities, we chose four large datasets relevant to immune researchers in light of the current pandemic. Three datasets are comprised of human cells, and one dataset consists of mouse cells. The three human datasets were chosen due to the distinct disease conditions being evaluated in the original studies. These included two COVID viral infection studies and human cSCC cancer study. We included a mouse dataset to demonstrate our model can be effectively used on non-human and non-immune datasets. All datasets were generated using the 10x Genomics platform (see (Goodwin et al., 2016; Heydari & Sindi, 2022) for a review of next-generation scRNAseq). A summary for each dataset is provided in subsequent subsections (see Fig. A.3).

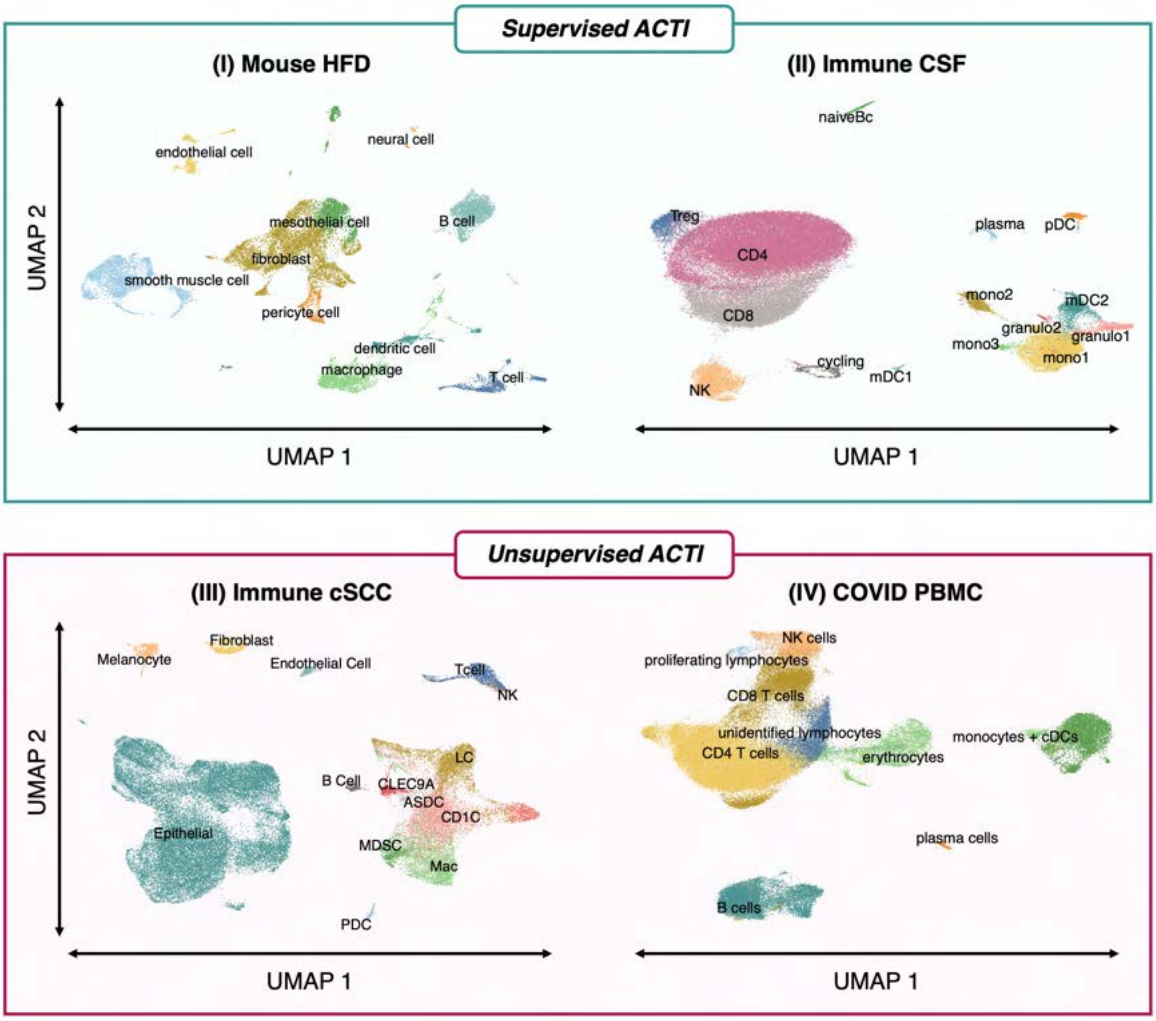

Figure A.3: A UMAP-reduced plot of cell populations present in each dataset investigated in this work.

B. 2.1. MOUSE-HDF

Mouse-HDF (Kan et al., 2021) [SCP1361] consists of scRNAseq of aortic cells in mice given a normal diet and mice given a high-fat diet (HFD), resulting in 24K cells (24K cells after processing). The authors identified 27 clusters, for 10 different cell populations.

Download Link: https://singlecell.broadinstitute.org/single_cell/study/SCP1361/single-cell-transcriptome-analysis-reveals-cellular-heterogeneity-in-the-ascending-aorta-of-normal-and-high-fat-diet-mice

B. 2.2. IMMUNE-CSF

Immune-CSF (Heming et al., 2021) [GSE163005] profiles single cells in cerebrospinal fluid (CSF) of Neuro-COVID, non-inflammatory, autoimmune neurological diseases, and viral encephalitis patients. Cells isolated from 31 patients: 8 COVID patients, 9 non-inflammatory, 9 autoimmune, 5 viral encephalitis resulting in a total 80K cells. After processing the dataset contains 70K cells with 15 populations.

Download Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE163005

B. 2.3. COVID PBMC

COVID-PBMC (Yao et al., 2021) [GSE163005] consists of the evaluation of the transcriptional immune dysfunction triggered during moderate and severe COVID-19 patients using scRNA-seq. Peripheral blood mononuclear cells (PBMC) were isolated from 20 patients and were sequenced. Patients included in the study ranged from healthy ($n = 3$), moderate COVID ($n = 5$), acute respiratory distress syndrome (ARDS-Severe, $n = 6$), and recovering ARDS-Recovering ($n = 6$), resulting in 69K cells. After pre-processing the dataset, we retained 64K cells. The authors identified 9 populations in the dataset.

Download Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE154567

B. 2.4. IMMUNE-CSCC

Immune-cSCC (Ji et al., 2020) [GSE144236] evaluates scRNAseq of normal skin and patient cutaneous squamous cell carcinoma (cSCC) tumors. Patients (n=10) with the cSCC tumors had normal skin and tumor cells sequenced, resulting in 48K cells. We retained 47K cells after pre-processing. The authors identified 7 major cell populations. The myeloid cell population (CD14+Hi) is composed of various subpopulations.

Download Link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE144236

Appendix C. : Definition of Standard Mathematical Expressions and Evaluation Metrics

C. 1. Cross Entropy Loss

The downstream task was to predict the correct labels (original labels or generated labels), with the learning objective being a standard cross-entropy loss:

$$H(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{j=1}^{k} y_j \log(\hat{y}_j) + (1 - y_j) \log(1 - \hat{y}_j), \tag{A.2}$$

where $\mathbf{y}$ and $\hat{\mathbf{y}}$ denote the original/generated labels and model predicted labels, respectively. In this formulation, $\mathbf{y}$ is a one-hot encoded vector of the cell types; that is

$$y_j = \begin{cases} 1 & \text{if cell type is } j \\ 0 & \text{otherwise.} \end{cases}$$

C. 2. Formulation of TF-IDF for Cell Type Querying

In this work, we leverage attention scores generated for each gene in all clusters. First, we calculate the average attention score for each gene per cluster. Next, take the top 100 genes with the highest attention scores per cluster and create an object containing only gene names. Using the top 100 genes per cluster object, we calculate the term frequency(TF)-inverse document frequency(IDF) for each gene in the matrix. TF-IDF is a standard tool in natural language processing (NLP) often used to weight the importance of words appearing in documents, as shown in (A.3).

$$TF(g, P) = \frac{f_P(g)}{|P|} \tag{A.3a}$$

$$IDF(g, D) = -\log p(g|D) = \log\left(\frac{|D|}{|\{P \in D, g \in P\}|}\right) \tag{A.3b}$$

$$TF\text{-}IDF(g, P, D) = TF(g, P) \cdot IDF(g, D) \tag{A.3c}$$

where f_P is the raw count of term (gene) g in the document (population) P, with a corpus of documents D. We consider each row in the matrix to be a document in which we calculate gene frequencies, and then calculate the IDF, which down-weights more common genes in the matrix. We multiply both TF and IDF for each gene to obtain the TF-IDF score. The generated TF-IDF score down-weights common genes in the top 100 by multiplying the TF-IDF scores by the average attention scores. Once the attention scores have been weighted, the top 25 genes per cluster are submitted to CellMeSH (CITE: PMID: 34893819), which generates a prediction cell type.

C. 3. Selecting Highly Variable Genes

To select Highly Variable Genes (HVGs), we utilized `Seurat`'s `FindVariableFeatures()` function, described in (Stuart et al., 2019). In this step, the aim is to calculate a measure of single-cell dispersion while including the mean expression. Step one learns a mean-variance relationship from the data, which is then used to calculate an expected standard deviation for each gene. To do so, the mean μ_g and variance σ_g of each gene is computed from raw data. Next, a curve fitting using a 2^{nd} degree polynomial is applied to learn $f(\mu_g) = \hat{\sigma}_g$, providing a regularized estimator of variance given the mean of a gene g. Using this, we perform feature transformations without removing higher-than-expected variations. That is, given the raw counts X_{ig} for gene g in cell i, the mean raw value for gene g, μ_g, and σ_g, the expected standard deviation of gene g from the global fit, we have the transformed count value of gene g as:

$$z_{ig} = \frac{X_{ig} - \mu_g}{\sigma_g}, \tag{A.4}$$

for all gene g across all cell i. Finally, variance across the new standard feature values is calculated and used to rank the genes. Although the convention is to select 2,000 HVGs, we chose to select 5,000 HVGs to increase the complexity and truly test model interpretability.

C. 4. Hit@k

Cell type retrieval position is measured using N-ACT attentive genes for each cell population using *Hit@k*. Hit@k is a metric measuring retrieval of a target value among the top k retrievals. In our model, a "hit" is the published cell type annotation (ground truth label) existing in the database. Rankings for $k = \{1, 3, 5, 10\}$ are reported in Fig. 2, and in Fig. A.4.

C. 5. F1 Score

F1 score is a standard metric for evaluating a classifier. F1 score is the harmonic mean of precision and recall, which is shown in Eq. (A.5),

$$F1 = 2\left(\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}\right). \tag{A.5}$$

For more information on weighted and non-weighted F1 score, see `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`.

Appendix D. CellMeSH Cell Type Predictions

CellMeSH (Mao et al., 2021) is a probabilistic method that generates a cell type prediction based on a gene list. CellMeSH aims to alleviate two common issues with querying gene lists from the literature: 1) publication bias, which relates to specific genes/cell types that are studied more often than other types, and therefore have more literature associated with them; and 2) Noise present in gene/cell-type mapping and the corresponding database. CellMesh uses a database built from the National Library of Medicine (NLM) MEDLINE indexed records called Medical Subject Headings. In building the CellMeSH database, Mao et al. use TF-IDF to reduce publication bias thus addressing publication bias.

To address the second issue, CellMeSH utilizes a probabilistic querying method. Given gene query list Q, CellMeSH assumes a probabilistic model for a query genes g being obtained from a cell-type C, shown in Eq. (A.6):

$$P(g|C) = \begin{cases} \alpha \cdot w_C(g) & g \in Q \cap C \\ (1-\alpha) \cdot \frac{1}{N_g - K_C} & g \in Q \cap \overline{C} \end{cases} \tag{A.6}$$

with $w_C(g)$ being the adjusted weight of gene g in cell type C (using TF-IDF), N_g, K_C denoting the total number of genes and total number genes with non-zero weight in C. α is a parameter which aims to help with noise in the database. For each candidate cell type C, CellMeSH calculates a log-likelihood score:

$$L(Q|C) = \log P(Q|C) = \sum_g \log P(g|C), \tag{A.7}$$

which are then used to rank C based on their values. Lastly, CellMeSH uses a maximum likelihood-based estimation to generate predictions. In this framework, the top cell type, C^*, can be found as:

$$C^* = \text{argmax}_C \log L(Q|C). \tag{A.8}$$

Appendix E. : Main Manuscript Results (Larger Figures)

COVID PBMC

Population Annotation	CellMesh Hit	Top Retrieval
Natural Killer Cells	#1	Natural Killer Cells
CD4+ T Cells	#5	T Cells
Unidentified Lymphocytes	N/A	T Cells
B Cells	#1	B Cells
Proliferating Lymphocytes	N/A	CD8+ T Cells
Monocytes + Dendritic Cells	#1	Monocytes
Plasma Cells	#2	B Cells
CD8+ T Cells	#2	T Cells
Erythrocytes	#1	Erythrocytes
Hit@1: 0.57	**Hit@3**: 0.85	**Hit@5**: 1.00	**Hit@10**: 1.00

Immune cSCC

Population Annotation	CellMesh Hit	Top Retrieval
Endothelial Cells	#1	Endothelial Cells
Epithelial Cells	#2	Keratinocytes
Fibroblast	#2	Endothelial Cells
Melanocytes	#1	Melanocytes
Plasmacytoid Dendritic Cells	#2	T Cells
CLEC9A+ Dendritic Cells	#1	Dendritic Cells
CD1c+ Dendritic Cells	#1	Dendritic Cells
Macrophages	#1	Macrophages
A+S+ Dendritic Cells	#1	Dendritic Cells
T Cells	#1	T Cells
MDS Cells	NR	Hematopoietic Stem Cells
B-Cells	#1	B Cells
Langerhans Cells	NR	Reed-Sternberg Cells
Natural Killer Cells	#4	T Cells
Hit@1: 0.57	**Hit@3**: 0.78	**Hit@5**: 0.85	**Hit@10**: 0.85

Figure A.4: (For the ease of the reader) An enlarged version of Hit@K results, initially presented in Fig. 2 of the main manuscript.

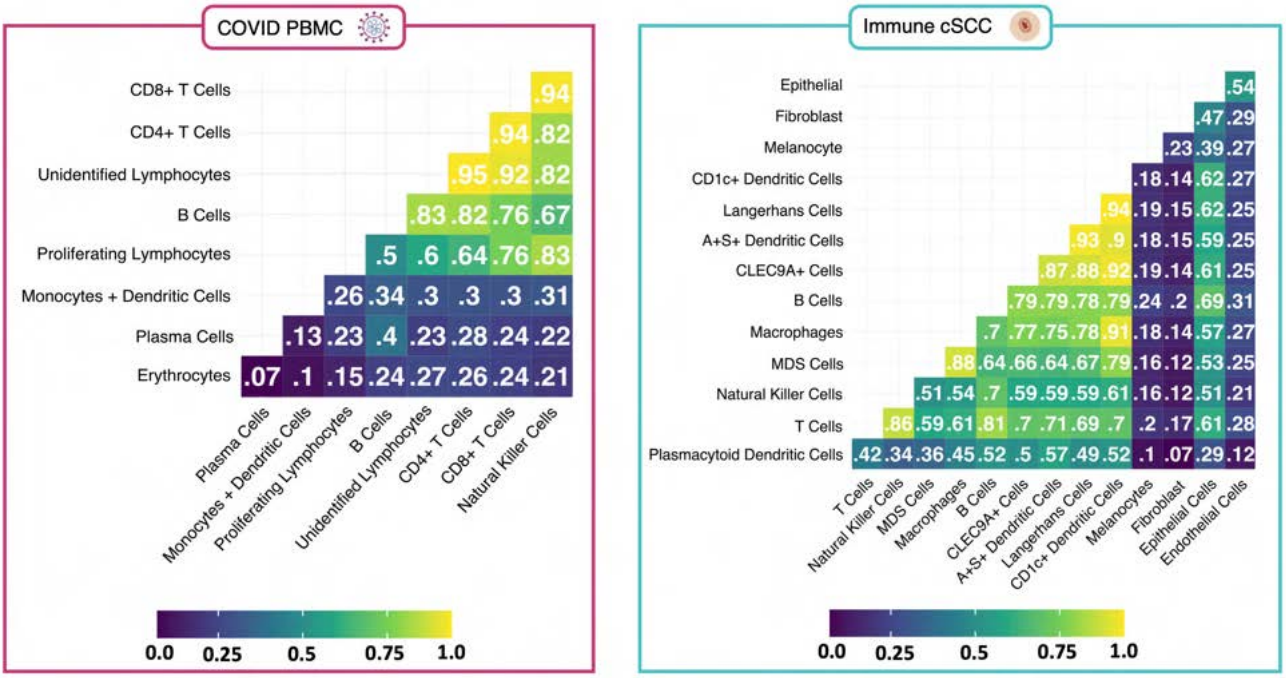

Figure A.5: (For the ease of the reader) An enlarged version of correlation results presented in Fig. 2 of the main manuscript.

Appendix F. N-ACT Architecture Details

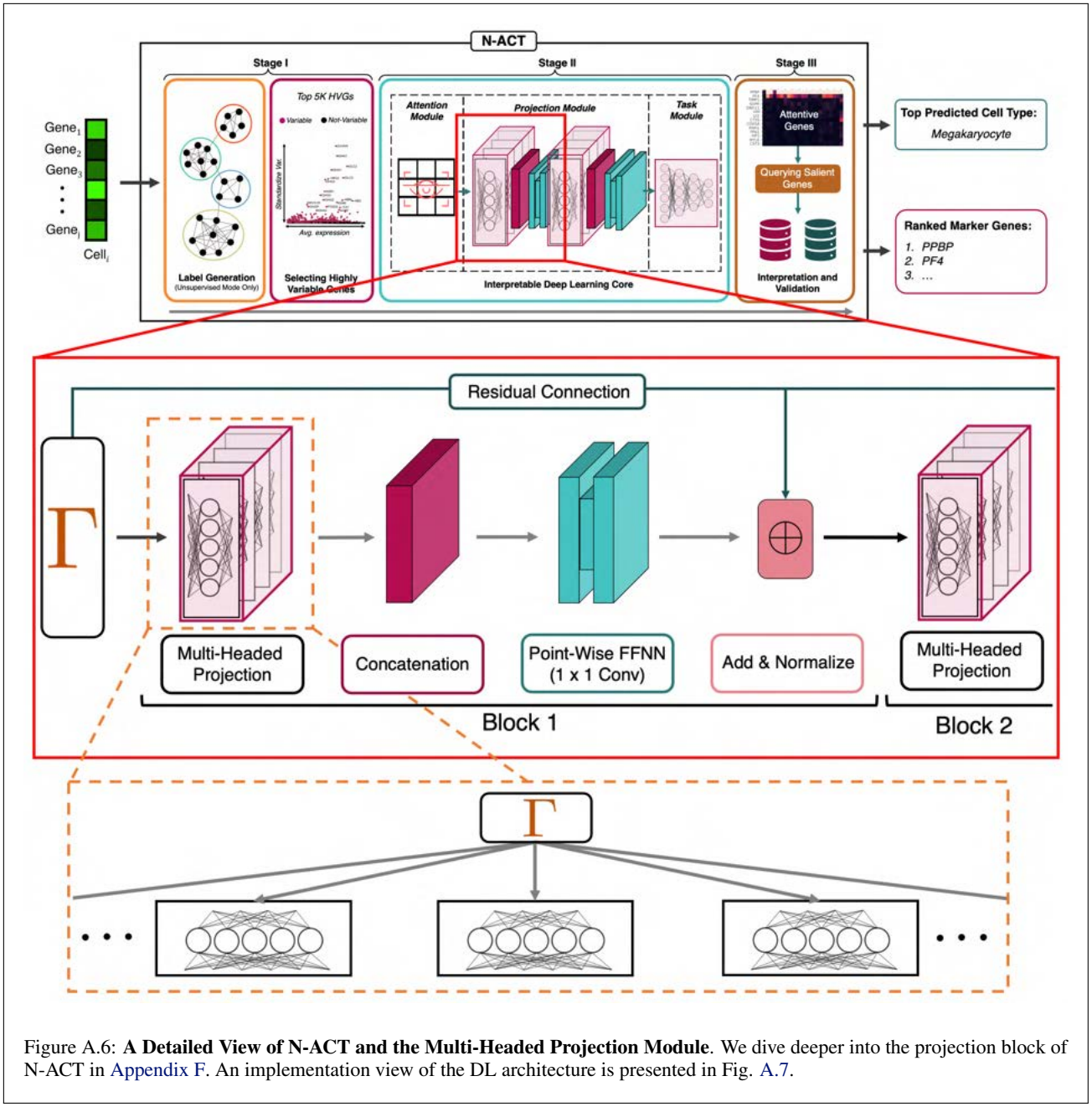

Figure A.6: **A Detailed View of N-ACT and the Multi-Headed Projection Module**. We dive deeper into the projection block of N-ACT in Appendix F. An implementation view of the DL architecture is presented in Fig. A.7.

F. 1. Projection Block

The idea behind the N-ACT projection block is to learn various representations for different gene subsets in each cell. Projection block design was inspired by the multi-head attention architecture presented in (Vaswani et al., 2017); however, the projections are not multi-head attention mechanisms. One way to think about the multi-head projection block is to view it as a set of h linear projections, with each $l_h : \mathbb{R}^{B\times N} \rightarrow \mathbb{R}^{B\times d}$ (B is the number of samples and N is the number of genes) done sequentially and independent of one-another (*e.g.* in a `for` loop). However, as noted by (Vaswani et al., 2017), these projections can be done more efficiently through creating a tensor $L \in \mathbb{R}^{B\times h\times d}$, which acts the same way as the collection of the individual linear operators. In this formulation, $0 \equiv N (\text{mod } h)$, and the last projection component, the

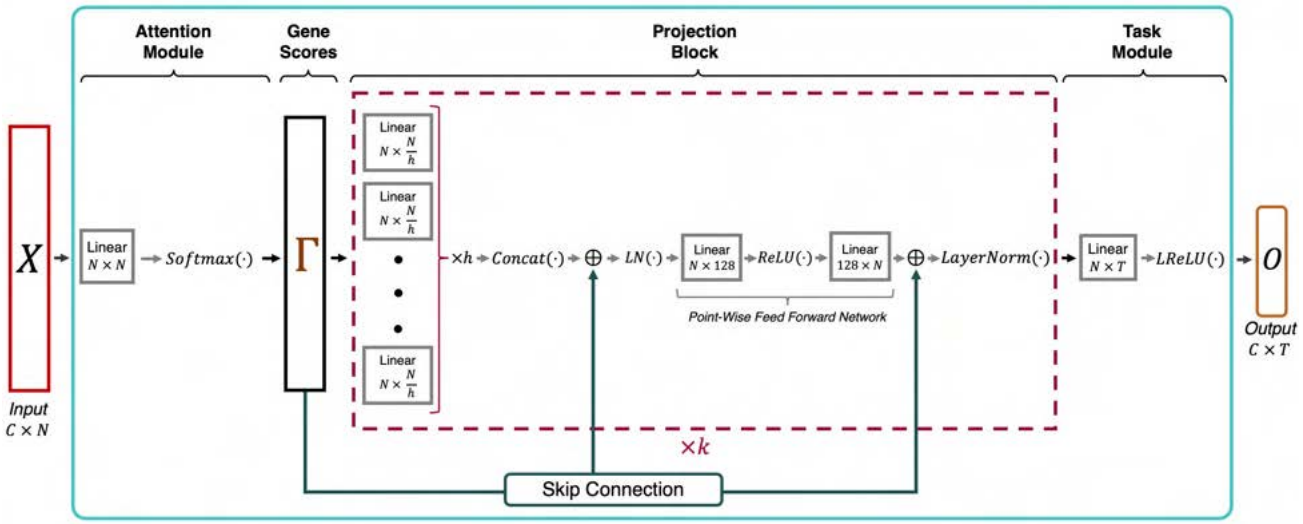

Figure A.7: **An Implementation View of the N-ACT DL Core**. In this illustration, C corresponds to the number of cells inputted, N denotes the number of features ($N = 5000$ in our results), h is the number of projection heads, k is the number of projection blocks(in our model, $h = 10$ and $k = 2$) and T denotes the number of classes (distinct labels) present in the data.

"concatenation" layer (depicted in Fig. A.6) allows us to reshape the model output which can be passed along to remaining layers.

To increase model capacity and allow N-ACT to learn complex mappings, outputs are non-linearly "activated" through a Point-Wise Feed Forward Neural Network (which can be thought of as $1\times$ 1 convolution). However, there is the possibility of projection blocks learning a non-linear mapping unrelated to interpretability, resulting in loss of interpretability. Therefore, we hypothesized that adding residual connection between output of the attention module and input of each subsequent layer would improve performance and interpretability. Our ablation studies show that adding a residual connection improves model performance (Table A.3). Skip connections are added to projection block outputs and normalized using Layer-Norm (Ba et al., 2016)) before being used as input for the next layer.

F. 2. Multi-Head Ablation Study

We investigated the effect of head number in the multi-head projection block and found that 10 heads provided the best results (Table A.2). We also tested our model accuracy when using one, two and three projection blocks and found that two projection blocks provided the best balance of accuracy and efficiency. To test our hypothesis regarding residual connections, we tested N-ACT (10 heads, 2 projection blocks) with and without skip connections. As shown in Table A.3, model accuracy drops without the residual connections. Residual connections are identity mappings added to the output of each layer, with this quantity normalized and used as inputs for subsequent layers.

Table A.2: **Ablation Study on the Number of Projection Heads.** We fixed all other hyperparameters and studied the effect of head number on accuracy and interpretability. Training times are the average of 5 training runs on an A100 GPU. Accuracy of each model remained the same across different training settings, since all random parameters were initialized with the same random seed.

Number of Heads	W-F1	NW-F1	Hit@5	Avg. Training Time
1	0.9278	0.8968	0.85	9.11 ± 0.14 (Min)
5	0.9317	0.9077	0.85	9.38 ± 0.31 (Min)
8	0.9307	0.9156	**1.00**	9.31 ± 0.22 (Min)
10	0.9322	**0.9173**	**1.00**	9.56 ± 0.27 (Min)
20	**0.9324**	0.9156	**1.00**	9.87 ± 0.24 (Min)

Table A.3: **Effectiveness of Residual Connections on N-ACT.**

Residual Connection?	W-F1	NW-F1
Yes	**0.9322**	**0.9173**
No	0.9243	0.8725

F. 3. Training Scheme

As mentioned in the main manuscript, we train N-ACT for 50 epochs using Adam (Kingma & Ba, 2014) optimizer, with a fixed learning rate. We tested training N-ACT for more epochs and found that the accuracy of predictions increases; however, a learning rate scheduling is beneficial in avoiding overfitting (when training over 100 epochs). When training for longer epochs, we employed an exponential learning rate decay with $\gamma = 0.95$ and a decay schedule of every 10 epochs. We chose 50 epochs to balance accuracy and training time efficiency.

F. 4. Supervised ACTI Comparison

Table A.4: **Benchmarking N-ACT on Supervised ACTI**. Although the main goal of N-ACT is interpretable ACTI in an unsupervised manner, our model can be used in a supervised setting as well, while still providing biological interpretability. The reported training times are the median of 5 runs for each model. The same random seed was used to initialize all random parameters to ensure reproducibility of results across different runs. *Training/Testing Support*: the number of samples used in training/evaluation, *W-F1*: Weighted F1 score, *NW-F1*: Non-Weighted F1 score

Model	W-F1	NW-F1	Training Support	Testing Support	Med. Training Time
Mouse HDF					
ACTINN	**0.9703**	0.9677	21,003	2,998	**1.5 Minutes**
N-ACT (Ours)	0.9681	**0.9712**	21,003	2,998	5.4 Minutes
Immune CSF					
ACTINN	**0.9357**	0.8898	66,549	8,991	**3.8 Minutes**
N-ACT (Ours)	0.9285	**0.8963**	66,549	8,991	11.2 Minutes
COVID PBMC					
ACTINN	**0.9323**	0.9148	55,005	9,864	**3.1 Minutes**
N-ACT (Ours)	0.9322	**0.9173**	55,005	9,864	9.4 Minutes
Immune cSCC					
ACTINN	0.9646	0.9315	40,027	6,994	**2.4 Minutes**
N-ACT (Ours)	**0.9684**	**0.9455**	40,027	6,994	7.7 Minutes

Appendix G. Utility of N-ACT for Disambiguation of Broad Annotations

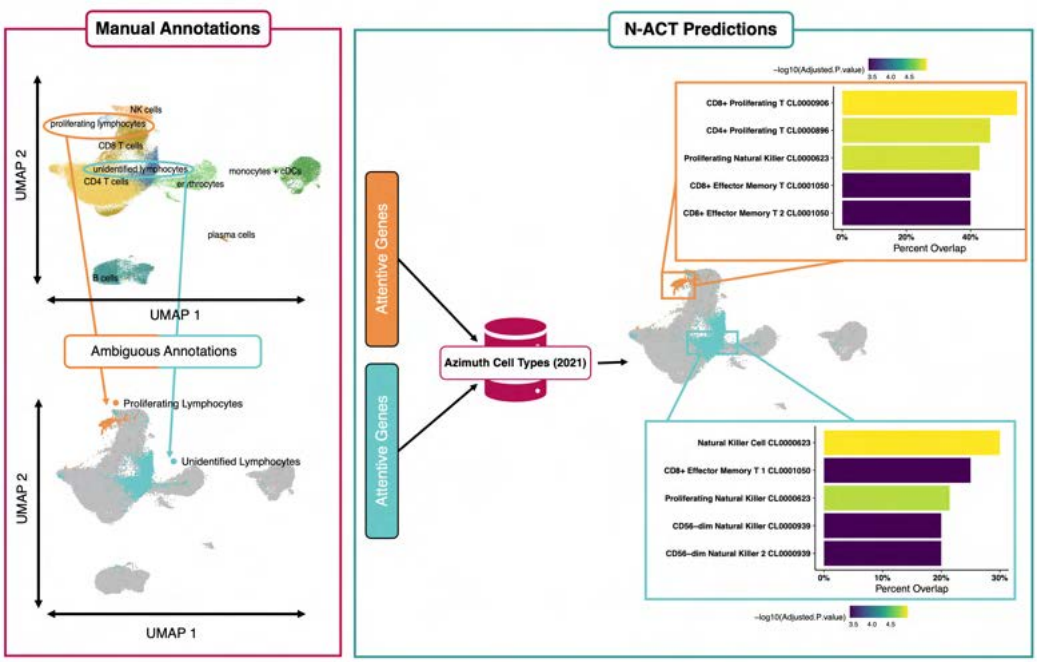

Figure A.8: **N-ACT Used to Disambiguate Broad Annotations**. Here, we use N-ACT-identified salient genes to query a specialized database (Azimuth) to disambiguate original broad manual annotations.

Given that many manual annotations are typically performed with only a few genes (often two or three (Luecken & Theis, 2019)), it is possible to have populations that are broad and ambiguous. This was the case with two populations of COVID PBMC data (Yao et al., 2021), namely the original annotations "Proliferating Lymphocytes" and " Unidentified Lymphocytes" (Fig. A.8). Here we utilize N-ACT to disambiguate these broad annotations without re-clustering or further complex analyses.

In the results shown in the main manuscript, we utilized CellMesh (Mao et al., 2021). However, using specialized databases could provide refined predictions. To demonstrate this, we investigated the two broad annotations in the COVID PBMC data and queried attentive genes from the Azimuth Cell Type 2021 database tailored for immune cells (Hao et al., 2021) [using Enrichr `R` Package (v 3.0) (Chen et al., 2013; Kuleshov et al., 2016; Xie et al., 2021)], to perform an enrichment analysis. We query our top 25 genes against the Azimuth Cell Types 2021 reference for the enrichment analysis and select the top 25 attentive genes (as described in 4.2 and Appendix C) for "proliferating lymphocytes" and "unidentified lymphocytes."

Our enrichment analysis for "proliferating lymphocytes" resulted in multiple significant (significant adjusted $p-$values) populations, with the most probable types being CD8+ proliferating T cells (shown in A.8). Intuitively, these results are expected given the quantitative and qualitative similarity of this population with the CD8+ T cell population. It is important to note that the difference in gene overlap percentages of top three predictions, namely CD8 proliferating T, CD4 proliferating T, and proliferating natural killer populations is very small, which may explain why the original annotations were left broadly as proliferating lymphocytes. Enrichment analysis for "unidentified lymphocytes" yielded natural killer cells as the most probable cell type, with other viable populations also being statistically significant. Similar to the previous population, our results are possibly intuitive given the closeness of the proliferating lymphocytes to CD8+ T cells and natural killer cells. Additionally, there is known overlap in the gene sets between of CD8+ T cells and natural killer cells. Lastly, we note that the second most probable population based on attentive genes are CD8+ effector memory T, suggesting that the "unidentified lymphocyte" population could likely be refined into two or more populations. These results further signify the utility of N-ACT for unsupervised annotation, and the applicability of our framework in tandem with other annotation forms to provide interpretability and validation.

References (Appendix)

[A1] Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

[A2] Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36:411–420, 2018. doi: 10.1038/nbt.4096. URL https://doi.org/10.1038/nbt.4096.

[A3] Chen, E. Y., Tan, C. M., Kou, Y., Duan, Q., Wang, Z., Meirelles, G. V., Clark, N. R., and Ma'ayan, A. Enrichr: interactive and collaborative html5 gene list enrichment analysis tool. *BMC Bioinformatics*, 14:128, Apr 2013. ISSN 1471-2105 (Electronic); 1471-2105 (Linking). doi: 10.1186/1471-2105-14-128.

[A4] Goodwin, S., McPherson, J. D., and McCombie, W. R. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016. doi: 10.1038/nrg.2016.49. URL https://doi.org/10.1038/nrg.2016.49.

[A5] Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, William M., I., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zager, M., Hoffman, P., Stoeckius, M., Papalexi, E., Mimitou, E. P., Jain, J., Srivastava, A., Stuart, T., Fleming, L. M., Yeung, B., Rogers, A. J., McElrath, J. M., Blish, C. A., Gottardo, R., Smibert, P., and Satija, R. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, 2022/04/30 2021. doi: 10.1016/j.cell.2021.04.048. URL https://doi.org/10.1016/j.cell.2021.04.048.

[A6] Heming, M., Li, X., Räuber, S., Mausberg, A. K., Börsch, A.-L., Hartlehnert, M., Singhal, A., Lu, I.-N., Fleischer, M., Szepanowski, F., Witzke, O., Brenner, T., Dittmer, U., Yosef, N., Kleinschnitz, C., Wiendl, H., Stettner, M., and Meyer Zu Hörste, G. Neurological manifestations of covid-19 feature t cell exhaustion and dedifferentiated monocytes in cerebrospinal fluid. *Immunity*, 54(1):164–175, Jan 2021.

[A7] Heydari, A. A. and Sindi, S. S. Deep learning in spatial transcriptomics: Learning from the next next-generation sequencing. *bioRxiv*, 2022. doi: 10.1101/2022.02.28.482392. URL https://www.biorxiv.org/content/early/2022/03/02/2022.02.28.482392.

[A8] Ji, A. L., Rubin, A. J., Thrane, K., Jiang, S., Reynolds, D. L., Meyers, R. M., Guo, M. G., George, B. M., Mollbrink, A., Bergenstråhle, J., Larsson, L., Bai, Y., Zhu, B., Bhaduri, A., Meyers, J. M., Rovira-Clavé, X., Hollmig, S. T., Aasi, S. Z., Nolan, G. P., Lundeberg, J., and Khavari, P. A. Multimodal analysis of composition and spatial architecture in human squamous cell carcinoma. *Cell*, 182(2):497–514, Jul 2020.

[A9] Kan, H., Zhang, K., Mao, A., Geng, L., Gao, M., Feng, L., You, Q., and Ma, X. Single-cell transcriptome analysis reveals cellular heterogeneity in the ascending aortas of normal and high-fat diet-fed mice. *Exp Mol Med*, 53(9):1379–1389, Sep 2021.

[A10] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

[A11] Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.-r., and Raychaudhuri, S. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature Methods*, 16(12):1289–1296, 2019. doi: 10.1038/s41592-019-0619-0. URL https://doi.org/10.1038/s41592-019-0619-0.

[A12] Kuleshov, M. V., Jones, M. R., Rouillard, A. D., Fernandez, N. F., Duan, Q., Wang, Z., Koplev, S., Jenkins, S. L., Jagodnik, K. M., Lachmann, A., McDermott, M. G., Monteiro, C. D., Gundersen, G. W., and Ma'ayan, A. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res*, 44(W1):W90–7, Jul 2016. ISSN 1362-4962 (Electronic); 0305-1048 (Print); 0305-1048 (Linking). doi: 10.1093/nar/gkw377.

[A13] Luecken, M. D. and Theis, F. J. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, 2019. doi: https://doi.org/10.15252/msb.20188746. URL https://www.embopress.org/doi/abs/10.15252/msb.20188746.

[A14] McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL https://arxiv.org/abs/1802.03426.

[A15] Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W. M., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.e21, 2019. doi: https://doi.org/10.1016/j.cell.2019.05.031. URL https://www.sciencedirect.com/science/article/pii/S0092867419305598.

[A16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[A17] Wolf, F. A., Angerer, P., and Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018. doi: 10.1186/s13059-017-1382-0. URL https://doi.org/10.1186/s13059-017-1382-0.

[A18] Xie, Z., Bailey, A., Kuleshov, M. V., Clarke, D. J. B., Evangelista, J. E., Jenkins, S. L., Lachmann, A., Wojciechowicz, M. L., Kropiwnicki, E., Jagodnik, K. M., Jeon, M., and Ma'ayan, A. Gene set knowledge discovery with enrichr. *Current Protocols*, 1(3):e90, 2021. doi: https://doi.org/10.1002/cpz1.90. URL https://currentprotocols.onlinelibrary.wiley.com/doi/abs/10.1002/cpz1.90.

[A19] Yao, C., Bora, S. A., Parimon, T., Zaman, T., Friedman, O. A., Palatinus, J. A., Surapaneni, N. S., Matusov, Y. P., Cerro Chiang, G., Kassar, A. G., Patel, N., Green, C. E. R., Aziz, A. W., Suri, H., Suda, J., Lopez, A. A., Martins, G. A., Stripp, B. R., Gharib, S. A., Goodridge, H. S., and Chen, P. Cell-type-specific immune dysregulation in severely ill covid-19 patients. *Cell Rep*, 34(1):108590, Jan 2021.