Neural Net Derived Gene Targets For The Nervous System, & Nervous System Derived Hierarchies For The Neural Net

Samuel Budoff [1 2] **Alon Poleg-Polsky** [1]

Abstract

This paper presents a machine learning approach for retinal cell classification, overcoming key constraints in spatial sequencing. We introduce a novel neural network training strategy that effectively classifies cells using just 225 genes, enabling cutting-edge techniques to provide spatial insights into retinal function. Inspired by biological perception, we also develop CuttleNet, a specialized architecture mirroring coarse-to-fine processing. Through hierarchical routing and subclass-specific subnetworks, CuttleNet sets a new benchmark in our limited application space. This is an important advance in spatial biology enabled by machine learning.

1. Introduction

Spatial biology is the latest frontier in quantifying the natural world, building on insights gained through single-cell RNA sequencing (scRNAseq)[1]. Spatial sequencing (spSeq) seeks to leverage the knowledge of defined cell types in order to place them into their context. However, a core challenge constraining real-world applications of this technology is the discrepancy between the number of genes that can be measured in scRNAseq and spSeq. Here we bridge this gap by contributing: 1) A novel neural network (NN) training strategy for efficient dimensionality reduction. As well as, 2) CuttleNet, a hierarchical NN architecture inspired by coarse-to-fine biological perception, achieving state-of-the-art performance scores benchmarked against the best available supervised classifier for this problem (Yin & Chen, 2023); outperforming it on our specific application.

The complex spatial arrangement of retinal cells presents an ideal case study to fill this gap, as mapping these cells is one of the largest gaps in knowledge in retinal biology. The major retinal cell classes include photoreceptors, bipolar cells, amacrine cells (ACs), and ganglion cells (RGCs). As neurons, all of these classes overlap in gene expression, but can still be distinguished from one another. In fact, on the basis of gene expression some classes can be divided into a few subclasses, like the rods and cones of photoreceptors (Masland, 2011), or into many, like the 45 RGC subtypes (Tran et al., 2019). Each subclass serves a distinct computational roles. For example, the alphaRGCs[2] mediate high acuity vision and can be easily distinguished from other subtypes based on genes like Spp1. Importantly, the spatial map of these cells enables this specific function and the associated behaviors (Holmgren et al., 2021; Bleckert et al., 2014).

Retinal maps are an evolved neural computation essential for vision (Hughes, 1975; Szél et al., 1992; Wässle et al., 1994; Ahnelt et al., 2006; Collin, 2008; Bleckert et al., 2014; Heukamp et al., 2020; Goetz et al., 2022). Recent research has highlighted the behavioral relevance of even a single cell subclass for these computations (Holmgren et al., 2021). Critically, scRNAseq has ballooned the number of relevant subclasses (Shekhar et al., 2016; Yan et al., 2020b;a) , thus, spSeq is the only plausible way to fill this gap in knowledge.

Here, we effectively reduced the needed clustering gene set 60-fold from 18k (Shekhar et al., 2016; Tran et al., 2019; Yan et al., 2020a) to 225 genes without sacrificing accuracy. Counter-intuitively, canonical markers were absent from this minimal gene subset. This is surprising, because outside the context of scRNAseq, when a biologist classifies a

[1]Department of Physiology & Biophysics, University of Colorado Anschutz Medical Center, Colorado, USA [2]Department of Mathematics & Statistics, University of Colorado Denver, Colorado, USA.

Manuscript evaluated by Master's Degree committee at the the University of Colorado Denver Mathematics and Statistics Department.
Copyright 2024 by the author(s).

[1]A way of measuring 10,000s of genes in individual cells.

[2]The primate ortholog are the midget RGC. (Hahn et al., 2023).

cell subclass one first uses markers to identify a more general cell class (Bleckert et al., 2014; Tran et al., 2019). Our unbiased gene selection strategy ignored traditional retinal cell class markers, inspiring us to try to impose something like a coarse-to-fine perceptual strategy on it to see if we could improve performance. This, combined with a less than ideal data imputation decision, led us to rigorously explore several approaches which organically resulted in our hierarchical model, CuttleNet. A Cuttlefish's nervous system contains a central ganglion responsible for organizing gross activities, like deciding which fish looks tasty, while a large fraction of the nervous system is actually distributed in disjointed tentacles that perform other complex tasks, like luring the chosen fish. We emulate this with CuttleNet's hierarchical structure performing selective routing. A primary network classifies cell classes, activating specialized subclass networks to refine identification, dynamically accessing only applicable data. CuttleNet advances cell classification in the context of constrained genetic datasets by leveraging biological principles into machine learning. Meanwhile, our innovative gene selection strategy offers to advance spSeq while wet-lab methodologies catch up to spRNAseq.

2. Materials and Methods

2.1. Materials

All experiments were conducted on a custom-built workstation featuring consumer-grade components. This workstation was equipped with an Intel i7-12700KF CPU, an NVIDIA RTX 3060 GPU with 12 GB of vRAM, and 128 GB of RAM (comprising four 32 GB Corsair CMK64GX5M2B5200C40 modules), all mounted on an MSI Z690-A Pro motherboard. The system operated on Ubuntu 22.04. All NNs and experiments were executed locally within a Python 3.9 Conda environment, utilizing Jupyter Notebooks. The primary codebase was developed using the PyTorch library. Auxiliary functions for data organization, visualization, and analysis were implemented using Numpy, Pandas, Matplotlib, and Sci-Kit Learn. Additionally, further data analysis and visualization were carried out in R, predominantly employing tools from the tidyverse library, particularly DPLYR and GGPLOT2. Code is available upon request while versions of this manuscript are undergoing peer review. Please contact samuel.budoff@cuanschutz.edu

2.2. Data

2.2.1. Data Preparation and Preprocessing

Our dataset amalgamated data from three open-source single-cell RNA sequencing studies of retinal cells. The integration process was initiated in R, structuring the data such that individual cells corresponded to rows and genes to columns. The gene set was curated based on inter-subclass variance in order to improve computational performance at the expense of being completely unbiased. For genes not present in a specific scRNAseq study, but still demonstrative of high variance relative to the subclasss present in the given study, cells were assigned a zero value for these genes. We also incorporated metadata related to the original study's clustering classifications and the specific study identifier for each cell. The clustering names were standardized for consistency, with each name receiving a two-character prefix to facilitate uniformity and automated assignment of cell classes based on predefined rules.

Class Encoding: We developed a simple, bespoke function to encode cell subclasss into broader classes, adhering to expert-defined rules. This encoding was important for our hierarchical classification approach, offering a secondary target for the loss function or hierarchical network, depending on the experimental design.

Label Encoding: The categorical cluster labels were numerically transformed utilizing Scikit-learn's LabelEncoder. This conversion is standard practice and essential for the NN's interpretation of the data.

Data Shuffling and Splitting: The dataset underwent shuffling (using a random state of 42 to ensure reproducibility) and was divided into input features (X) and labels (y). The features were normalized, and the dataset was split into training and test sets, allocating 80% for training purposes.

2.3. Biased Dataset Generation and NN Training

2.3.1. Dataset Curriculum Creation

The core of our methodology involved training several NNs, each tailored to learn a specific cell class. To achieve this, we employed a custom curriculum that repeatedly exposed each network to a desired class against various other classes. This curriculum was designed to ensure that each network would encounter the desired class multiple times while also

encountering each non-desired class at least once.

2.3.2. Data Preparation Function

A Python function, head2head, was devised to create a balanced dataset comprising equal parts of the desired and noise (non-desired) classes. In cases where there were discrepancies in class sizes, the function would either replicate instances from the smaller class or randomly exclude instances from the larger class to achieve balance.

2.3.3. Curriculum Function

Another function, DesiredGeneCurriculum, was implemented to generate training sets for each desired class. This function utilized the head2head function to prepare the biased datasets for each class, ensuring a focused learning environment for the network.

2.3.4. NN Training

The NNs were trained using a custom function, QuickNN. This function encapsulated the entire training process, including the instantiation of the NN model, setting up the optimizer and loss function, and executing the training loop. The training was conducted on a GPU-enabled workstation to leverage the computational efficiency of parallel processing.

The NN architecture, Net, was a simple feed-forward network with configurable parameters for the number of classes, hidden layer sizes, and dropout probability. The architecture allowed for experimentation with different configurations, although the final model used in this study was a simplified version focusing on a single hidden layer.

2.3.5. Training Process

The training involved standard procedures such as shuffling and splitting the data, normalizing the input features, and employing a validation set for early stopping to prevent overfitting. The loss function used was cross-entropy, and the Adam optimizer was chosen for its efficiency in handling sparse gradients.

2.3.6. Model Evaluation

Each trained model was evaluated against a test set to assess its performance. The results were stored in a dictionary for subsequent analysis. This step was crucial in determining the effectiveness of each biased training regime.

2.4. Gradient-Based Gene Selection

2.4.1. Feature Importance Computation

To identify the most relevant genes for cell subclass classification, we applied a gradient-based feature importance technique. This approach involved training the network on our biased datasets and then calculating the gradients of the loss with respect to the input data, which in our case, were gene expression levels. The mean absolute value of these gradients across the batch dimension was used as a measure of each gene's importance in classification. This method allowed us to capture the network's sensitivity to changes in gene expression levels, pinpointing the genes most influential for classification.

2.4.2. Gene Selection Process

The feature importance calculation was performed for each subclass-focused network. For each cell class, the corresponding network model was loaded, and the feature importance was computed against a batch of test data, all labeled as the target class. This provided a rank order of gene importance specific to each cell subclass.

2.4.3. Gene Ranking and Selection

The top genes identified by each model were then compiled into a list. This list was further refined by including additional handpicked genes based on biological knowledge. The resultant gene list combined both data-driven insights and expert-driven selections, ensuring a comprehensive coverage of genes relevant for cell classification.

2.4.4. Model Training with Selected Genes

A final NN model was trained using the selected gene set. This model was evaluated on a standard test set to assess its classification performance. The performance of this model served as a benchmark to gauge the effectiveness of our gene selection strategy.

2.4.5. Comparative Analysis

We conducted a comparative analysis between our gene selection results and existing gene lists proposed in the literature for retinal cell classification. Specifically, we compared our results with the gene list proposed by Tran et al. in their single-cell RNA sequencing study. This comparison involved evaluating the true positive rate (TPR) and precision (Prec) of both gene sets and calculating the delta values to quantify the differences in performance.

2.4.6. Result Interpretation

The comparative analysis provided insights into the effectiveness of our gradient-based gene selection approach. By examining the differences in TPR and Prec between our selected gene set and the literature-based set, we were able to assess the relative strengths and limitations of our method.

2.5. Final Gene Selection and Model Training

2.5.1. Training with the Full Gene Set

To finalize our gene list, we trained a NN model using the entire gene set. This comprehensive model was trained without bias towards any specific cell class, providing a holistic view of gene importance across all cell classes.

2.5.2. Feature Importance Computation for Final Gene List

Post-training, we employed the previously described gradient-based feature importance computation method. This time, however, it was applied to the comprehensive model to determine the most relevant genes across all cell classes. For each cell class, we computed the feature importance and extracted the top genes, culminating in a final list that represented the most influential genes for cell classification.

2.5.3. Data Structuring and Ranking

The final gene list was then structured for practical use. We created a dataframe where each row represented a cell class and columns were ordered by gene importance. This format allowed for a hierarchical view of gene significance, essential for designing gene panels in spSeq applications, such as the 10x Xenium platform.

2.5.4. Gene Ranking Algorithm

An algorithm was developed to rank genes based on their order of importance across all cell classes. The algorithm ensured that the final list was devoid of duplicates while maintaining the order of first appearance, thus preserving the hierarchy of gene significance.

2.5.5. Final Gene List Determination

The gene ranking algorithm was iteratively applied to the dataframe, progressively building a rank-ordered list of genes. This process was carefully designed to meet the requirements of spSeq platforms, ensuring a minimum number of genes while prioritizing the most significant ones.

2.5.6. Result Export and Documentation

The finalized rank-ordered gene list was then exported for practical application. The list was saved as a CSV file, providing a readily accessible format for further use in spSeq experiments and panel design.

2.6. Experiment 1: Blended Class and Subclass Cost Functions

In this experiment, we developed a nuanced custom loss function aimed at effectively combining the cost functions of both cell class and subclass classifications. This was achieved through an innovative approach that leverages the hierarchical relationship between classes and subclasses in single-cell RNA sequencing data.

2.6.1. Design of the Custom Loss Function

The cornerstone of our approach is the CustomLoss class, intricately designed to blend two distinct loss functions:

Log Probability Transformation: For each subclass, the model's log probabilities are first exponentiated and then summed up to map onto the corresponding cell classes. This process effectively aggregates subclass predictions into a more general class level. **Dual Loss Calculation:** The function computes two separate loss scores. First, it calculates a subclass loss score using CrossEntropyLoss, which directly deals with the subclass predictions. Concurrently, the aggregated class predictions (converted back into log probabilities) are used to calculate a class loss score with NLLLoss. These two scores are then integrated into a weighted average, where the sum of the class weight and the subclass weight equals 1.

2.6.2. Experimental Setup and Execution

The experiment was meticulously structured to assess the effectiveness of this custom loss function in a hierarchical classification context:

Training Configuration: A NN was trained using the CustomLoss function, configured with a fixed batch size (32) and a set number of epochs (100). **Blending Ratio Variations:** The focal point of the experiment was to vary the blending ratio of class to subclass loss functions. This was operationalized through different blending weights (0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0), exploring the spectrum from purely class-focused to purely subclass-focused loss calculations. **Reproducibility and Robustness:** To ensure reliability, the experiment was repeated with multiple seeds (0, 3, 6, 9, 12, 15, 18), thereby bolstering the robustness of our findings. The primary aim was to observe how varying the emphasis between class and subclass cost functions impacts the network's ability to accurately classify cells into their respective categories. The results from this experiment were pivotal in enhancing our understanding of hierarchical classification dynamics, potentially influencing future strategies in cell class classification using single-cell RNA sequencing data.

2.7. Experiment 2: Dynamic Variation of Class Weight

In the second experiment, we dynamically varied the class weight in the custom loss function from 0 to 1 during the training process. This approach aimed to investigate whether gradually shifting the focus from class to subclass classification would enhance the model's performance. The logic, follows from the way a student will learn to classify cells, first learning to focus on class features, and later to appreciate the subclass features.

2.8. Experiment 3: Hierarchical Network Architecture

This experiment focused on developing and evaluating a novel two-tier NN, named CuttleNet, designed for hierarchical cell classification. This architecture reflects the coarse-to-fine processing strategy observed in biological systems, aiming to improve precision and accuracy in hierarchical cell classification tasks.

2.8.1. Network Design

CuttleNet consists of a primary class classifier and multiple subclass classifiers, each responsible for a specific level of classification granularity. See **Fig. 1**

Class Classifier: The first stage broadly predicts cell classes using a two-layer Fc. The input consists of gene expression data common to all classes, and the output layer size corresponds to the number of classes. Non-linearity is introduced via ReLU activation functions, and a softmax layer generates a probability distribution over classes.

Subclass Classifiers: Each subclass classifier, stored in a ModuleDict for dynamic selection, is tasked with identifying subclasses within a predicted class. These networks take concatenated inputs of class-specific gene expression data and the class classifier's output. They are uniquely configured to output predictions for the number of subclasses they represent, based on a predefined class-to-subclass mapping. Note that by adjusting the input genes to only reflect those measured in

the original scRNAseq dataset we correct the main methodological compromises in the original architecture of assigning zero to these unmeasured parameters.

2.8.2. Forward Propagation

During forward propagation, the input data passes through the class classifier to yield a broad class prediction. This class probability distribution is then concatenated with class-specific gene expression data to feed the relevant subclass classifier. The output from the subclass classifier, scaled by the class probability, contributes to the final comprehensive probability distribution across all subclasses.

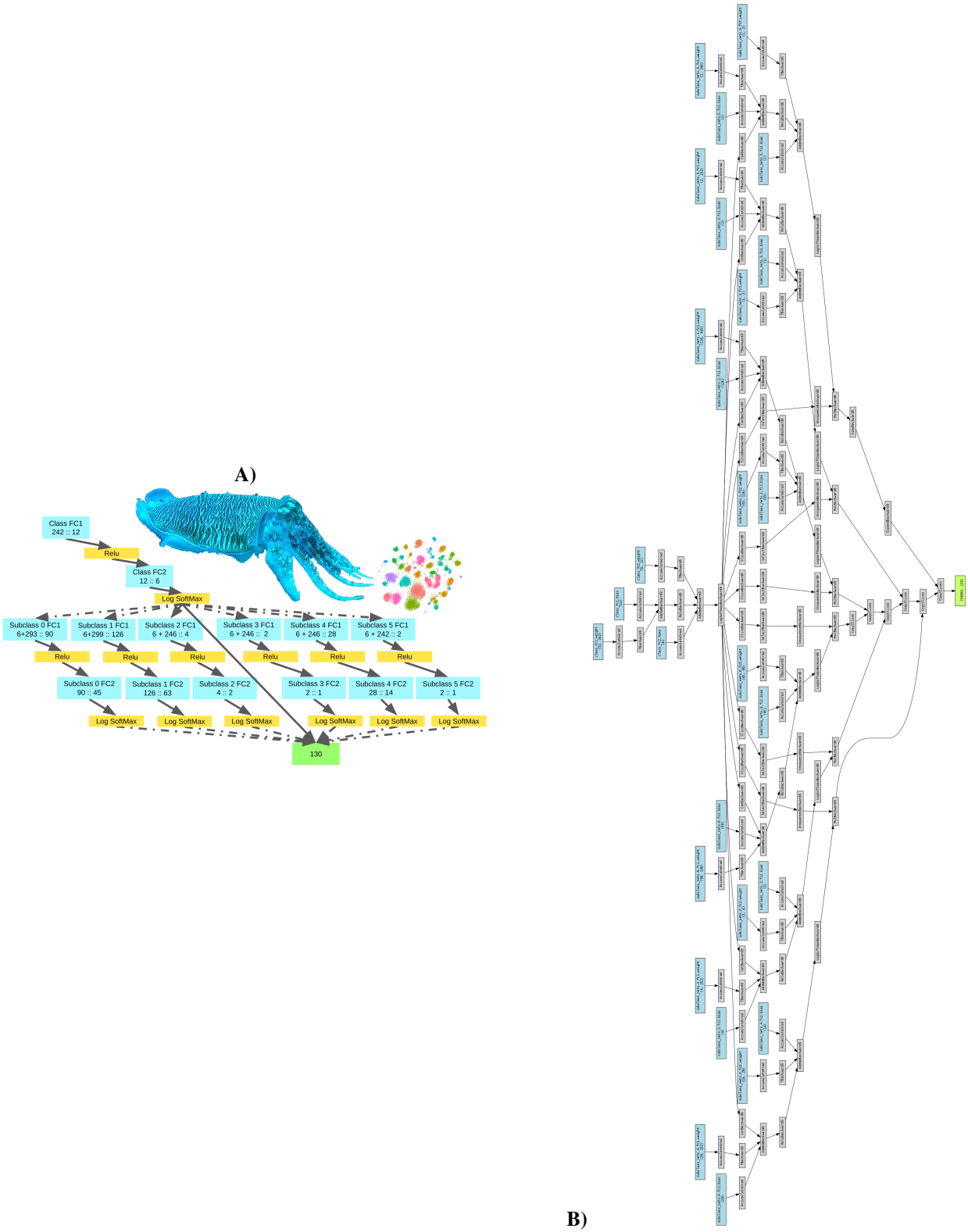


Figure 1. Diagram of CuttleNet. **A)** Blue boxes indicate layers with input genes/nodes :: layer nodes; $6 + g$ in each subclass' tentacle indicates the concatenation of the class output with the g genes available to that layer. Yellow boxes indicate the non-linearity. Dashed arrows indicate conditionally engaged connections. **B)** Full schematic diagram of CuttleNet Architecture, without the specific gene input sizes entering each subnetwork used in our application or our good friend Carl the Cuttlefish featured reaching for the UMAP clustering.

2.9. Hyperparameter Optimization: A Grid Search Approach

To optimize the CuttleNet performance, we conducted a systematic grid search varying epochs, early stopping, and L1 regularization:

Number of Epochs: Tested across 10, 25, 50, 100 epochs to assess the impact on model convergence.

Early Stopping: Implemented with criteria at 0, 5, 10 epochs to prevent overfitting and enhance model generalization.

L1 Regularization: Applied with strengths 0.0, 0.001, 0.005, 0.01, aiding in feature selection and model robustness.

Each hyperparameter combination was evaluated across multiple seeds (18 to 108, in increments of 18) for robustness and reproducibility. The model's performance was assessed on a test set following training with the specified configurations. Finally, 1 layer and 5 layer CellTICS were trained with and without access to imputed data. Each condition was assessed on a test set, replicated five times.

2.9.1. Data Collection and Analysis

For each experiment iteration, detailed results were collected, including the trained model, test set evaluation metrics, hyperparameter settings, and the loss history during training. This comprehensive data collection facilitated a thorough analysis of the impact of each hyperparameter on the model's performance, particularly focusing on metrics such as F1 score and loss reduction across epochs.

The hyperparameter grid search experiment played a pivotal role in fine-tuning the hierarchical network model. By meticulously analyzing the outcomes of varying training epochs, early stopping criteria, and regularization strengths, we were able to identify the optimal settings that maximized the model's predictive accuracy while maintaining generalizability. This systematic approach not only enhanced the model's performance but also provided valuable insights into the dynamics of deep learning model training in the context of hierarchical cell classification.

For all experiments, post-training, each model was evaluated on the test set which was not used at any point in the training. The various metrics calculated for each model created include: True Positives (TP), False Positives (FP), True Negatives (TN), False Negatives (FN), True Positive Rate (TPR), True Negative Rate (TNR), Precision, F1 Score, and Accuracy for each cell cluster. This comprehensive evaluation provided a detailed understanding of the models' performance across different cell classes and subclasses. The data was saved and assessed in R.

2.10. Data Collection and Analysis

Test set evaluations for each model was assessed for precision (Prec), True Positive Rate (TPR) and their harmonic mean (F1) for both cell classes and subclasses. Where appropriate ANOVA and post-hoc Tukey testing with appropriate corrections were employed for comparing fine grained differences. Otherwise, we here defined the "best" respective hyperaprameter set for a given architecture to be those that maximized the sum of class and subclass F1, Prec, and TPR minus their respective standard deviations.

3. Results

3.1. Gene Selection

spSeq promises to comprehensively map retinal cells, but to facilitate this it is necessary to determine a much smaller set of genes capable of effectively and reliably clustering neural cells. To achieve this we explored several strategies but ultimately found that a simple feed-forward NN (FF) could be trained to predict subclasses in a supervised manner; with genes input as a matrix of predictors. We began by focusing on the data of Tran et al (2019), who classified RGCs based on scRNAseq. As part of a wet-lab experiment using the relatively low throughput RNAscope technique, we hand selected 12 genes and used these to subset the expression matrices and trained a FF with a single hidden layer twice as larger as the number of targets. To our surprise, this resulted in successful classification of the cells we predicted with greater than 90% TPR and Prec; hereafter, we will refer to this architecture as the GLM. We further validated this simple model by training it on the set of 63 genes proposed by Tran and colleagues for immunohistochemically classifying each of the 45 RGC subclasses with sets of 1-3 genes each. We expected, that our set of 12 genes targeting 17 subclasss would relatively outperform the

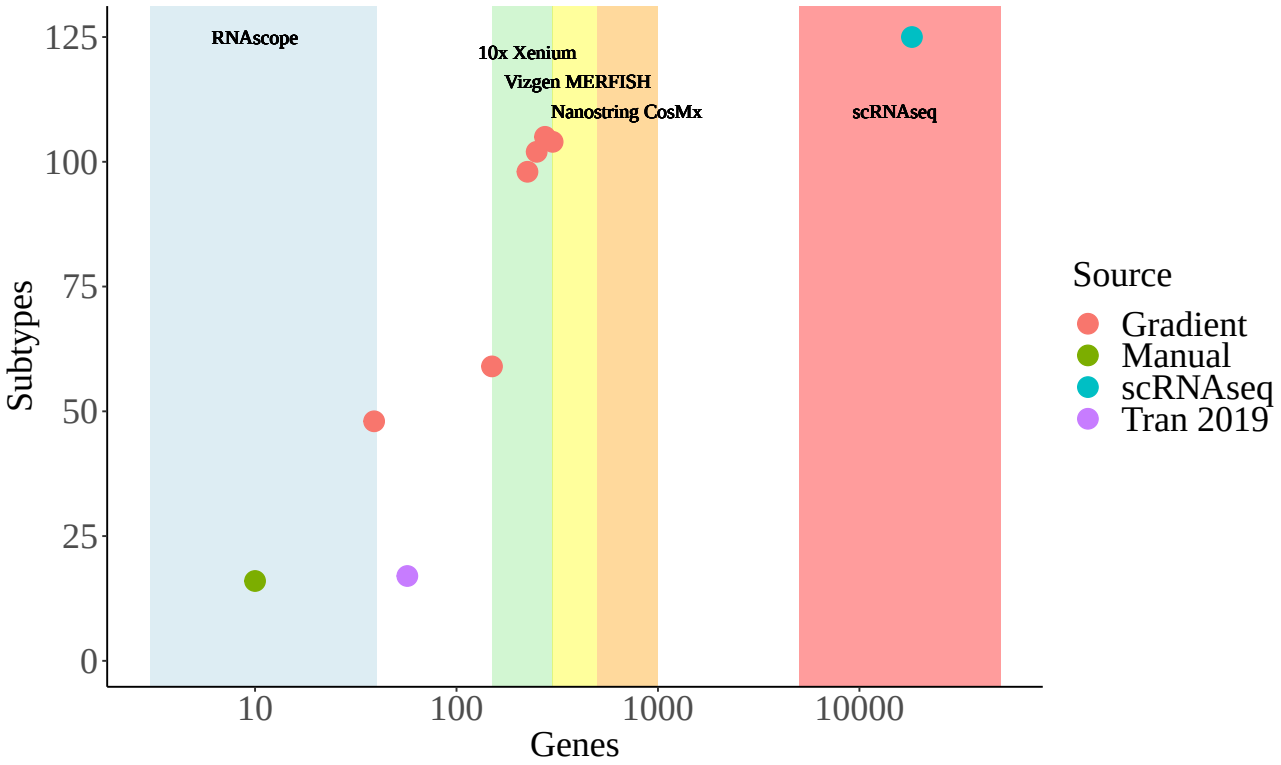


Figure 2. Subclasses classified with more than 90% Prec and TPR relative to the number of genes used to train a GLM to achieve this performance.Gene lists used to train GLMs are: Gradient = Our biased training set method; Manual = A set of 12 genes selected by hand to maximally classify RGC subclasses; scRNAseq = The original classification; Tran 2019 = The gold standard RGC classifying set of genes. X-axis also shows the range of genes that can be measured by respective wet-lab techniques[4]

disjointed sets proposed by Tran et al[5] (2019). We were satisfied with the model recapitulating this expectation (see **Fig. 2**), while successfully classifying subclasses that had particularly obvious gene expressions above our thresholds.

We optimized the NN with a grid search, and found that a hidden layer three times as large as the number of targets was optimal, but did not find the added computational cost beneficial over the GLM (see **Sup. Fig. 3**, and **Sup. Fig. 10**). Moreover, the single layer design meant the GLM could be easily interrogated. This inspired us to try to use it to select genes in an unbiased manner. We had previously tried other methods including dimensionality reduction with inferential statistics, genetic evolutionary algorithms, and other published tools but did not have any success when we re-clustered the proposals. This exploration demonstrated that many genes do not vary between retinal subclasses and could be disregarded. When training the GLMs on this reduced expression matrix we observed that the gradient from any given gene to any given subclass could predict the importance of that gene in classifying the subclass of interest. We combined three different scRNAseq studies that together include all 125 neuronal subclasses. The subclasses in this dataset are highly unbalanced, prompting us to ask what the gradient for a given subclass would be when it was in a balanced dataset. We achieved this by purposefully biasing our training data for each subclass against a background of 50% noise. We trained different GLMs on datasets each biased against one of the subclass in our scRNAseq data and extracted the magnitude of the gradient for each gene toward the subclass of interest. Finally, we combined these lists of genes ranked by importance and subset the top N genes from the list to train new GLMs. The results of this exploration, displayed in **Fig. 2**, show that this strategy yielded a plateau in performance at roughly 225 genes[6] Finally, we clustered these genes using both UMAP and t-SNE projections and matched published clusters with both techniques (see **Fig. 4, 5**). Taking advantage of the surplus of 75 genes available, we were able to add additional targets, constituting our expression matrix.

[5]This expectation was because we were taking advantage of 4 to 12 times as much information per subclass

[6]We expect that beyond 300 genes further performance steps exist, but these do not help our 10X Xenium focused application.

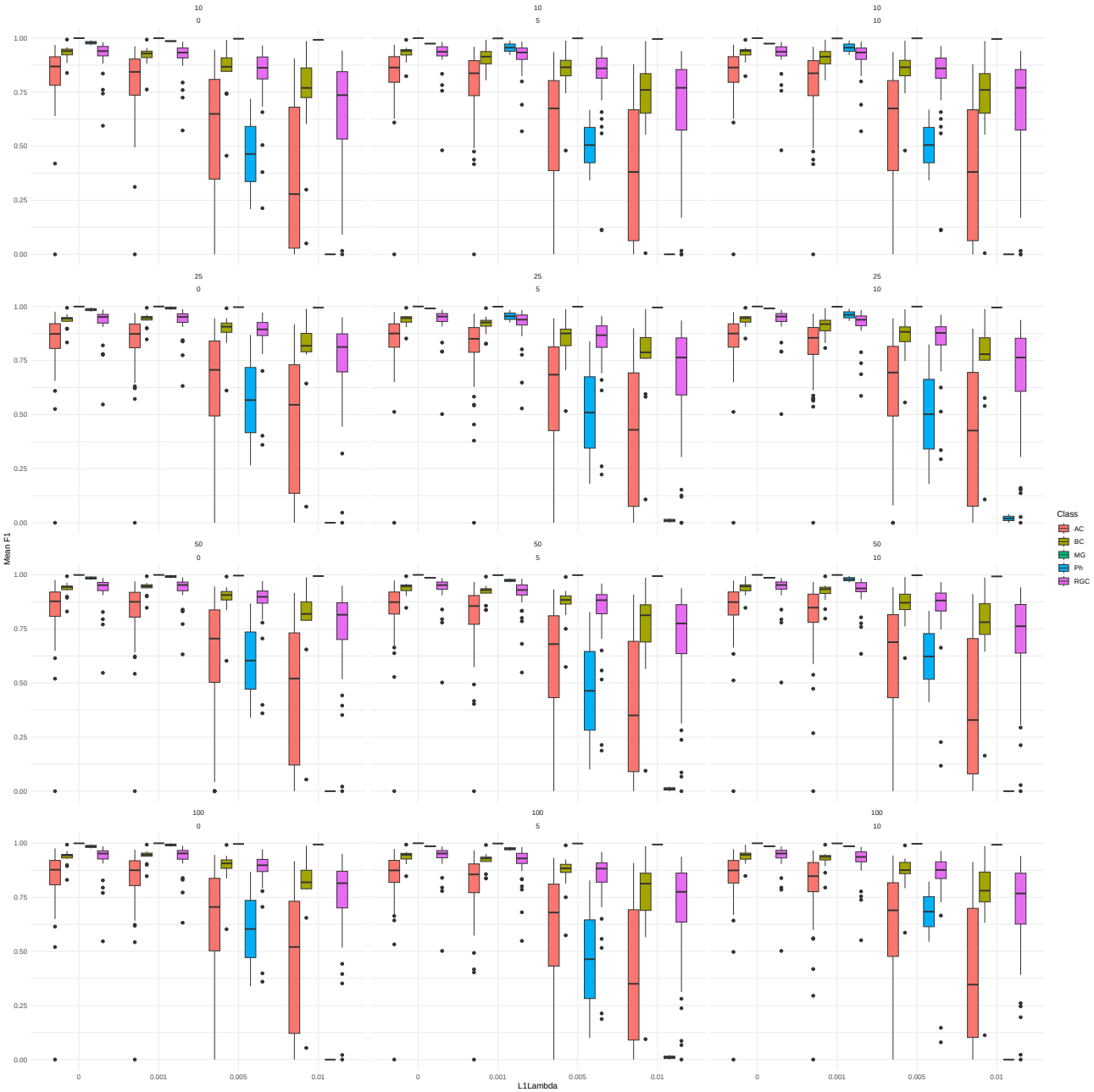
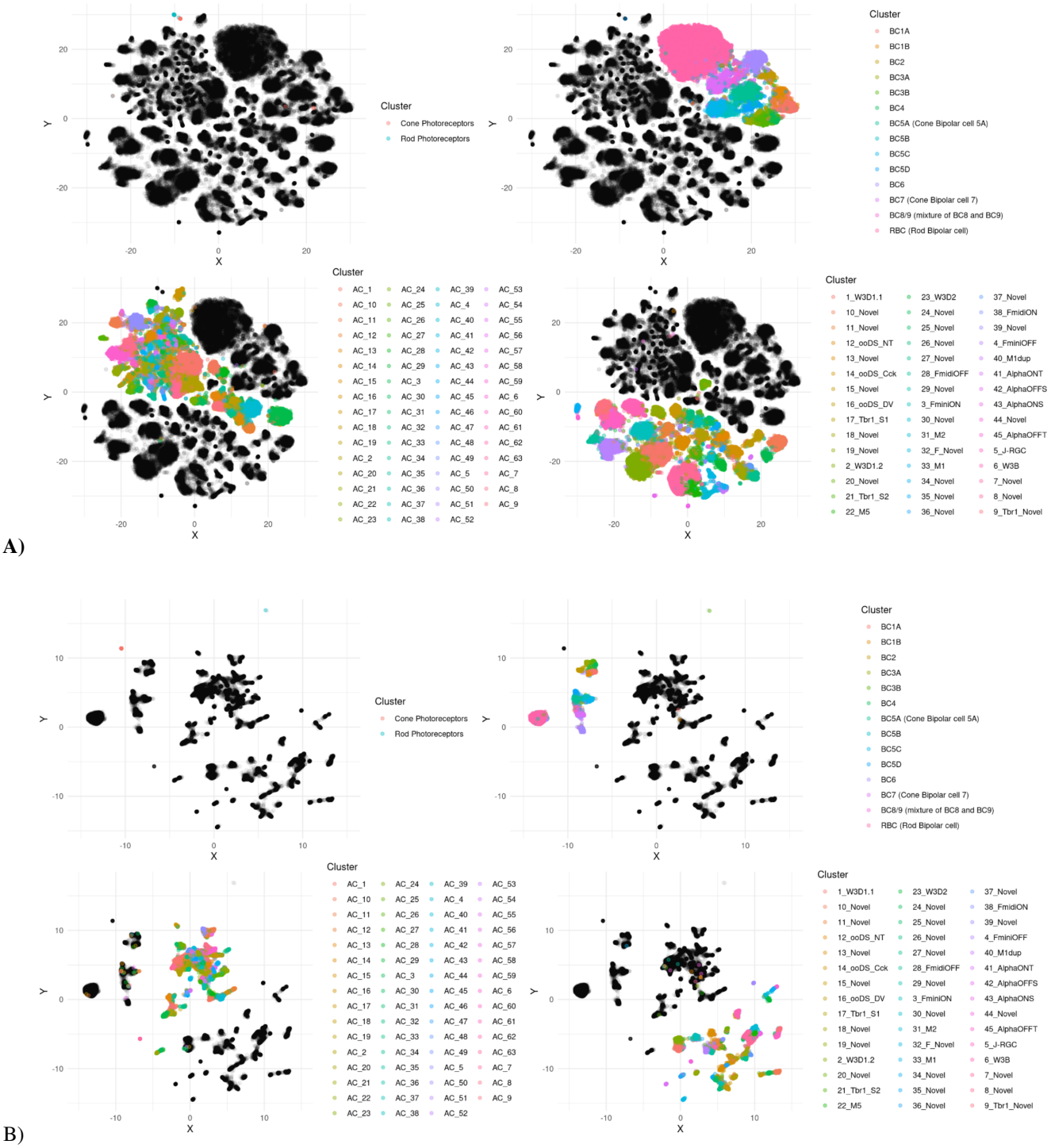


Figure 3. Grid search of GLM with various hyperparameters that were explored. Panels are subset by Number of Epochs from top to bottom, Early Stopping parameter left to right. Colors show classes, box plots show F1 score of the subclasses, and X axis shows L1 Regularization parameter.



A)

B)

Figure 4. **A)** UMAP clustering of all retinal cells, with neuronal classes colored in each subplot to highlight respective subclasses. **B)** t-SNE clustering of all retinal cells, with coloring as in A. An important sanity check for our method visible in these plots is the fact that subclasses of a given class cluster near each other and exclude clusters from other classes. The photoreceptor rods and cones are biologically dissimilar so the exception to this observation they appear to present is in fact its own sanity check that we are not observing a systematic artifact related to differences in datasets.

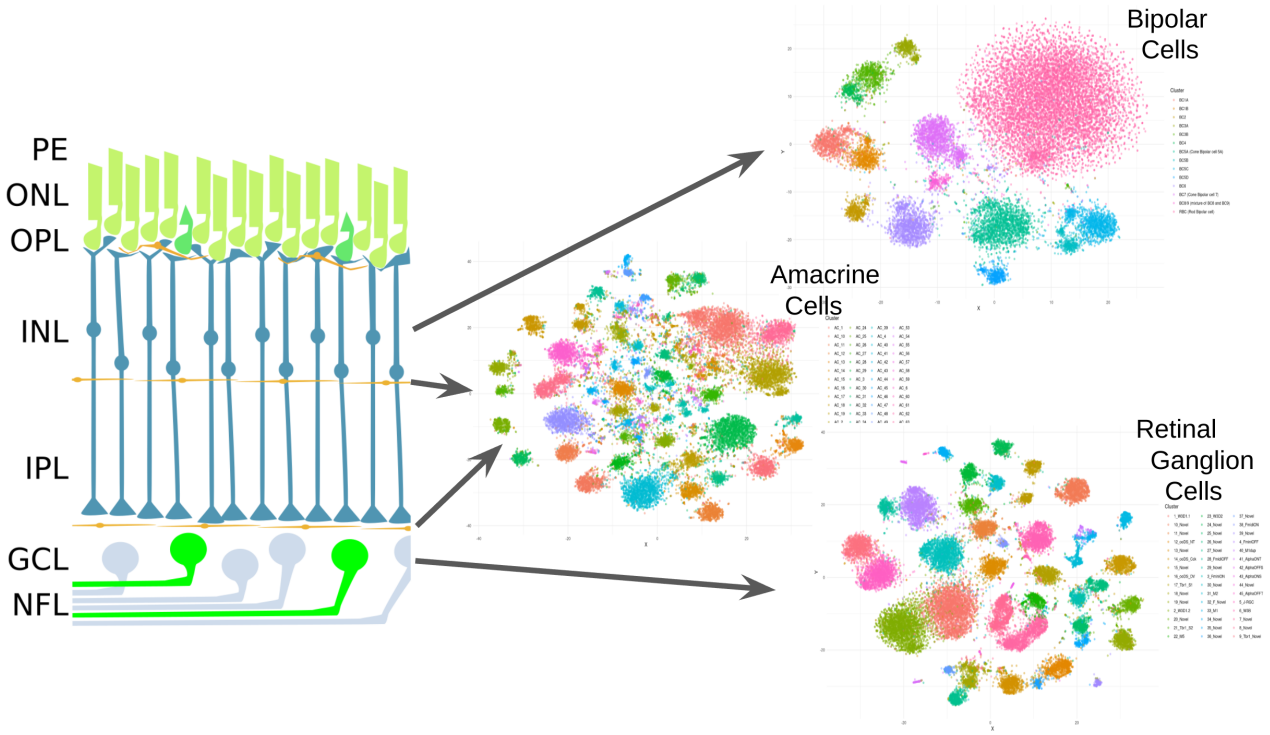


Figure 5. UMAP clustering of retinal cells after pre-filtering for subtypes belonging to respective classes with approximate spatial location of these cells indicated by the diagram of a retinal cross section. Labels are anatomic regions: PE=pigment epithelium, ONL=outer nuclear layer (NL); OPL=outer plexiform layer (PL); INL=inner NL; IPL=inner PL; GCL=ganglion cell layer; NFL=nerve fiber layer.

3.2. Experiment 1 - Simultaneous Learning

A surprising observation we made of the genes selected by our Gradient strategy was that classic cell class markers were absent. At first this highlights the difference in how humans proceed with cell classification; we use a coarse-to-fine strategy of ever more specifically classifying a given cell from the possible alternatives. Whereas the GLM considers all of the genes it is observing in a given cell at once. That said, we observed the ratio of genes per classified subclasses for our handpicked list was superior to the set proposed by the Gradient strategy. Thus, we wondered if we could not embed class information into a model in a manner analogous to the coarse-to-fine processing of natural nervous systems.

Focusing on the cost function, we hypothesized that since the GLM was observing all genes at once, it might be prudent to emphasize genes important for the broad cell class in addition to the subclass. We thus created a custom loss function that would compute a loss value for the model's prediction of both features. We then blended these complimentary loss values using a weighted average of the two[7]. We then proceeded to train GLMs on a variety of ClassWeights using our 300 gene expression matrix, which included canonical class specific genes we purposefully added in.

We specifically focused on key performance metrics: TPR, Prec, and F1 Score, under varying ClassWeights by consolidating results from different replicates and ClassWeights. To statistically assess the impact of different ClassWeights on performance metrics, we conducted an ANOVA test for each cell subclass, which showed significant variations which was further examined using Tukey's HSD test for post-hoc analysis. Specifically, we focused on ClassWeights which resulted in performance metrics significantly differing from the control condition (ClassWeight=0). Our analysis revealed a nuanced response to varying ClassWeights in the custom loss function (**Sup. Fig. 6**). While no statistically significant improvements were observed in F1 scores across any ClassWeight, the story was different for Precision and TPR (**Sup. Fig. 11**). Specifically, 10/125 cell subclasses demonstrated a significant increase in Precision for at least one class weight, with the maximum improvement being 19.4%. Similarly, TPR saw significant improvements in 15/125 subclasses, the highest increase being

[7]The sum of the ClassWeight and SubclassWeight is always 1.

Table 1. Median F1 scores, with mean standard deviations from the replicate runs. 0s indicates if the model had access to imputed zeros. Optimized indicates the version of a given architecture found via grid search. CellTICS models are the gold standard.

EXPERIMENT	SUBCLASS F1	CLASS F1	0S
OPTIMIZED CUTTLENET	93.8±1.5	99.9 ± 10^{-2}	×
OPTIMIZED GLM	93.6±0.6	99.8 ± 10^{-3}	√
CELLTICS 5 LAYER	85.7±3.8	85.5±0.5	√
CELLTICS 5 LAYER	83.5±4.1	85.0±0.3	×
CELLTICS 1 LAYER	68.9±7.2	75.6±1.0	√
CELLTICS 1 LAYER	66.0±7.3	74.7±0.6	×

11.5%. These results indicate the custom loss function can sometimes enhance Prec and TPR for certain subclasses. However, the specific optimal ClassWeight varied unpredictably, and no clear correlation was found with factors like cell count in the dataset. Worse, in aggregate this strategy performed worse than the original subclass only training strategy **(Table 2)**.

3.3. Experiment 2: Dynamic ClassWeight Training

The coarse-to-fine perceptual strategy in biologic nervous systems is in part understood to arise as a developmental and/or learning process. Thus, we sought to use our ClassWeight cost function to try and mimic this process. We did this specifically by systematically varying the ClassWeight during training, with different GLMs being trained under various rates of transition from ClassWeight=1 to ClassWeight=0. This strategy marginally outperformed the GLM (**Fig. 6, Sup. Table 2**), suggesting some value of the ClassWeight strategy. One interpretation of our findings would be similar in principle to transfer learning. Essentially our GLM was pretrained to classify classes, so as the model transitioned to learning subclasses, its weights were primed to do so.

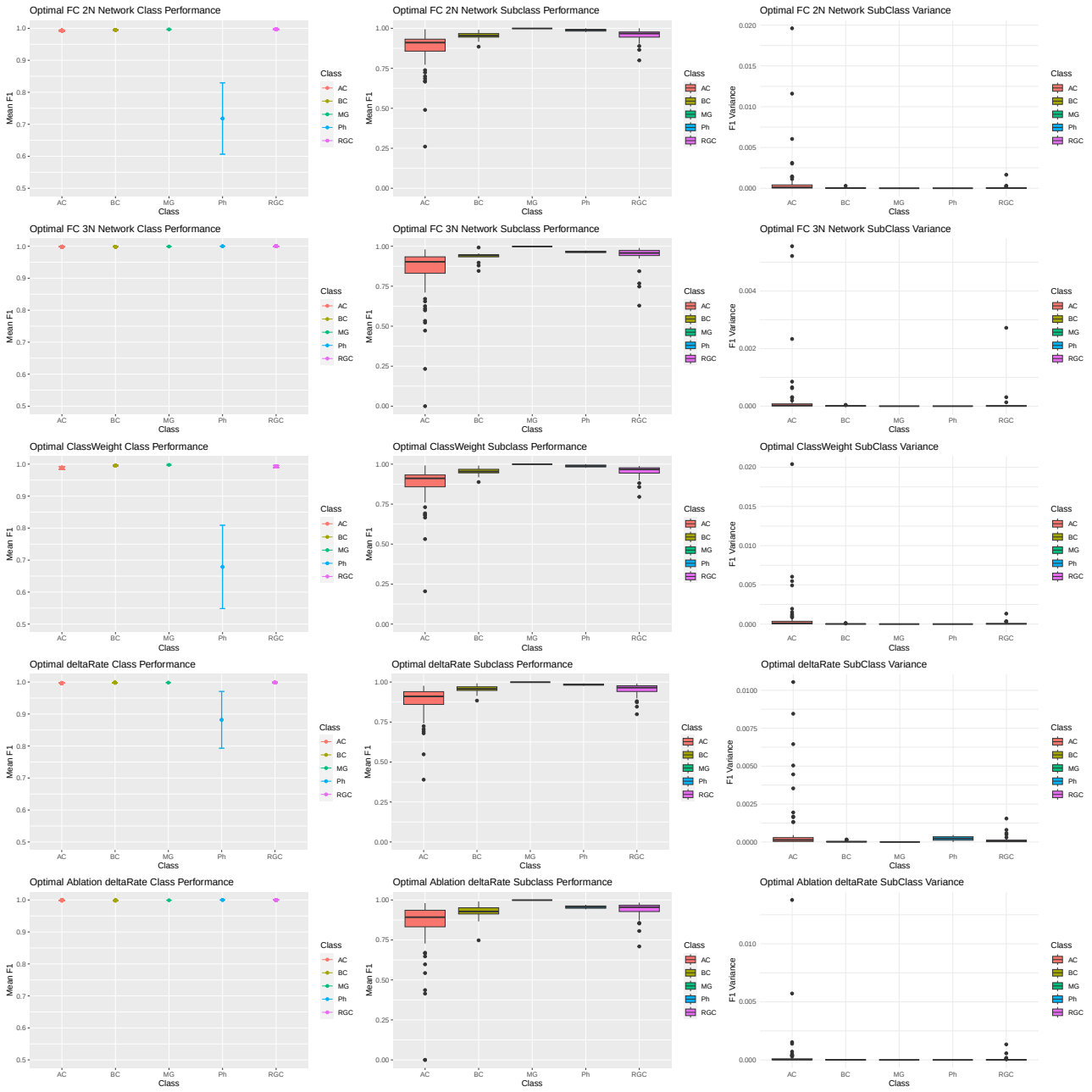


Figure 6. Optimal versions of each Fc network found after perfroming appropriate grid searches for each architecture of experiment. Each experimental condition in the search included five replicates of training and evaluating the respective model; an 80/20 train/test split was done in all cases with the metrics evaluated here coming only from the testing dataset.

3.4. Experiment 3: CuttleNet

A worrisome compromise we introduced in creating a pan-retinal cell expression matrix was the imputation of genes not measured between studies. That is, if a gene was found to be sufficiently variable to pass our initial culling of one expression matrix, it is not guaranteed to have been measured in the other scRNAseq studies. Instead of discarding such candidate genes we imputed values of zero for studies which did not include them. We suspected in doing so that some of the performance of our GLMs may stem from this decision. To test this, we performed an ablation study by removing all genes with imputed 0s from our expression matrix and trained a version of our GLM. This caused a, slight, but significant performance drop using a paired T-test of our subclass classification F1 score $t(153299) = 19.3, p = 2 \times 10^{-16}$. The practical significance in this case is small, unsurprising given how well the GLM performs in general above 225 genes, but it gave additional credence to our qualitative concern about the impact of imputed values. Thus, we devised a hierarchical modification to our GLM, leveraging biologic design principles in which separate GLMs acted as sub-networks responsible for either classes generally, or a specific subclass. Such an architecture, like that found in a cuttlefish's nervous system, allows for quasi independence of networks, where sub regions of the network receive from upstream regions only the information necessary for their task, but all of the sub-networks ultimately act together to perform complex behaviors. The specific information entering a given sub-network was the set of available genes free of imputed zeros, and samples predicted to be the respective class. The ultimate output was an aggregation of the activated subclass focused sub-network's predictions (**Fig. 1**)

Through a comprehensive grid search (**Fig. 7**), we fine-tuned the hyperparameters for our novel hierarchical network, CuttleNet. This search was a deliberate exploration to pinpoint a balance between training duration, regularization, and early stopping criteria that would yield the best performance for both class and subclass predictions. The optimal parameters emerged at 25 epochs of training, without early stopping, and with a regularization strength of 0.001 (**Fig. 8**). This configuration not only maximized the subclass performance—with diminishing returns observed beyond, but in general the grid-search revealed that this architecture rapidly converged on high quality class predictions; achieving near-perfect accuracy before 10 epochs (**Fig. 7**). This aptitude for class identification may be due to the relatively simple task.

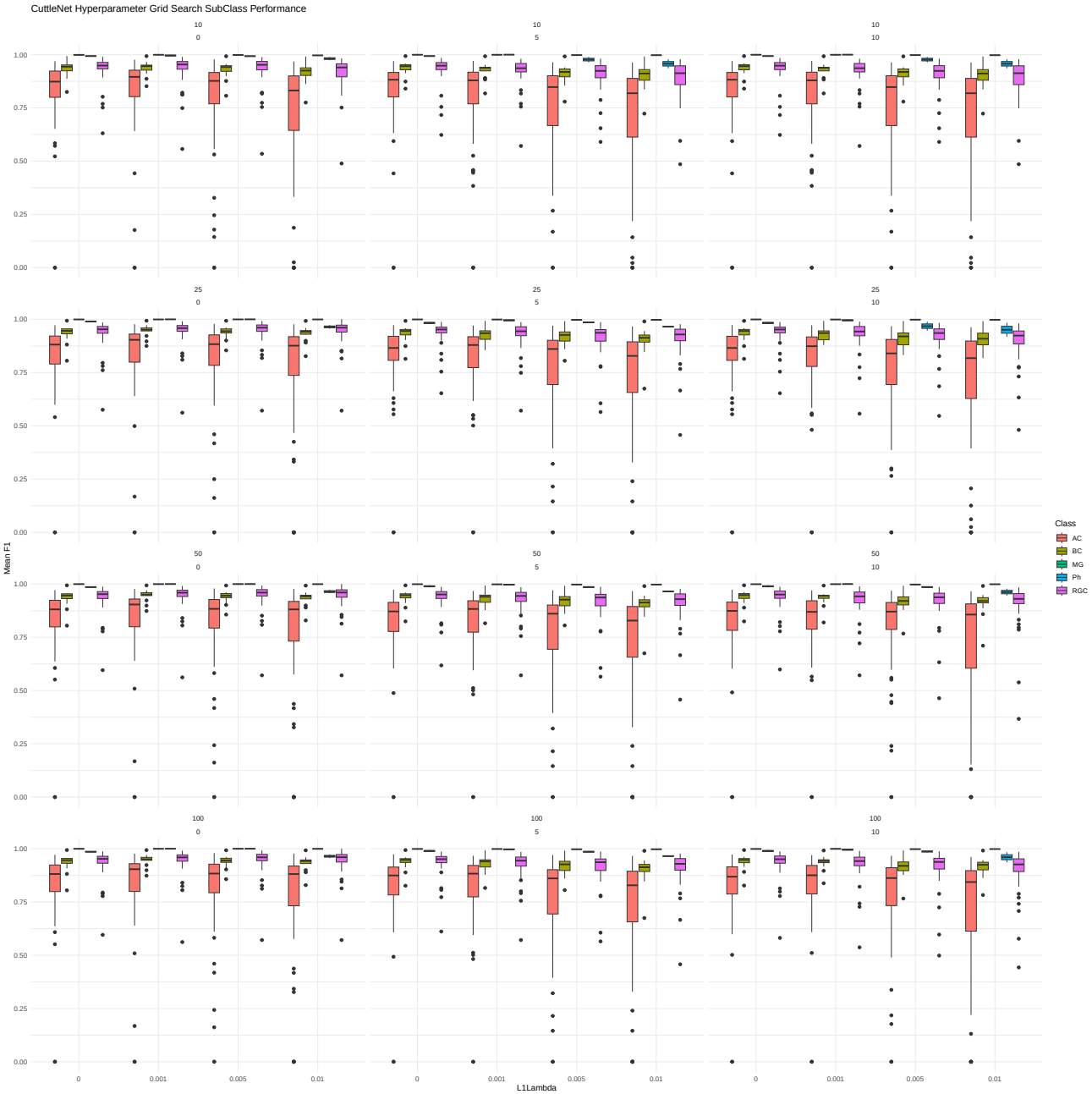


Figure 7. Grid search of CuttleNet with various hyperparameters that were explored. Panels are subset by Number of Epochs from top to bottom, Early Stopping parameter left to right. Colors show classes, box plots show F1 score of the subclasses, and X axis shows L1 Regularization parameter.

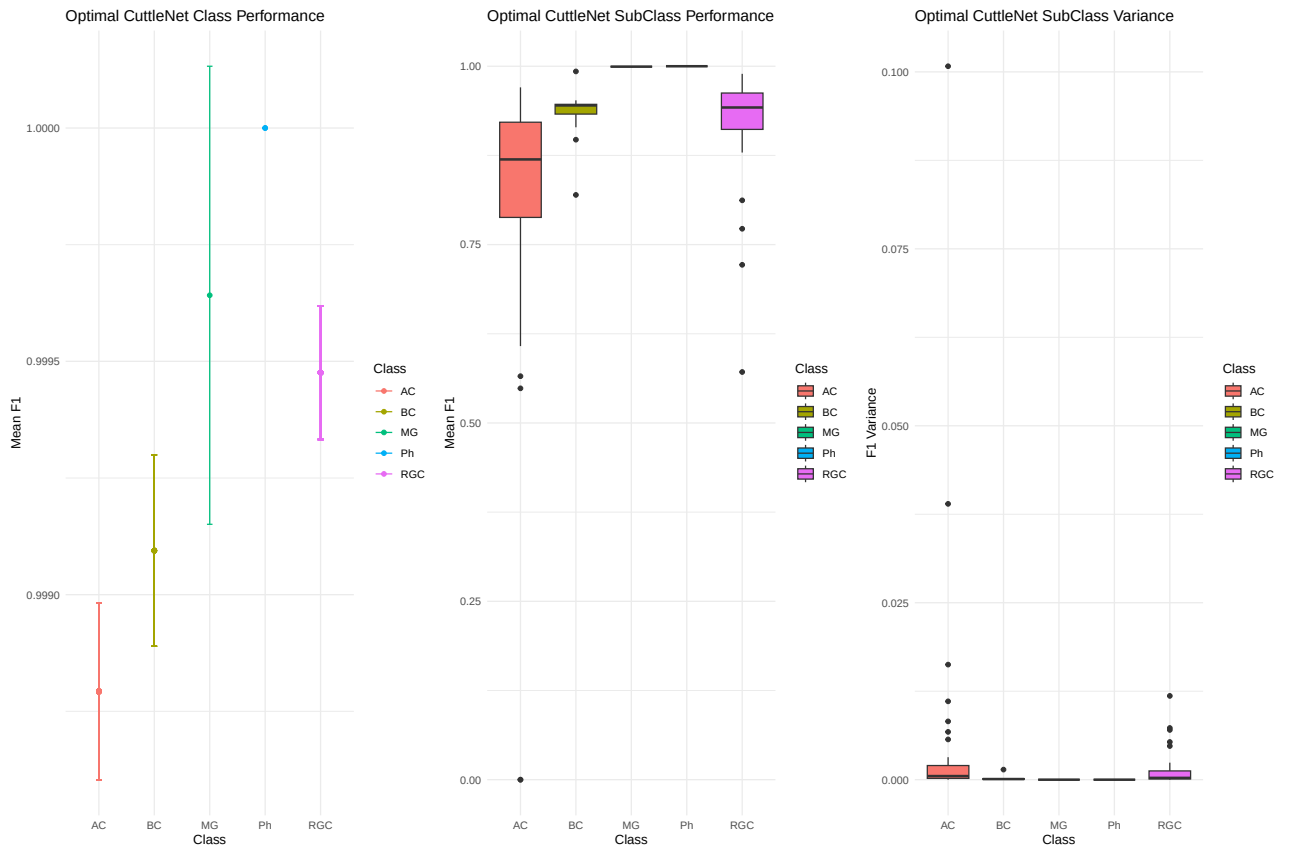


Figure 8. Performance of CuttleNet on each class and subclass for the optimal parameter set, defined as the maximum sum of the median F1 scores for both subclass and class.

The nuanced role of the regularization parameter in our CuttleNet's performance became evident when examining the longevity of class classification accuracy and subclass differentiation over extended training periods. The optimal regularization strength was found to be 0.001, striking a balance that mitigates overfitting while preserving the network's ability to generalize. This was an important finding, as it allowed the network to maintain a high level of class classification accuracy, even with prolonged training epochs. Specifically, we observed that without regularization, the F1 scores not only dropped for all cell classes, but the variability between training replicates dramatically increased proportional to training epochs. Conversely, as regularization increased, the network's capacity to distinguish between the more granular subclass categories diminished. This inverse relationship between regularization strength and subclass accuracy underscores the intricacies of model tuning for our hierarchical architecture (**Fig. 7**).

Unlike the previous attempt at imbuing a network with something akin to a coarse-to-fine processing model, the CuttleNet architecture excelled beyond the baseline GLM network in terms of performance[8]. Moreover, the GLM, even when trained with the dynamic custom-loss function strategy, suffered from a philosophically fatal flaw which CuttleNet circumvents, by dynamically routing imputed data away from any decision making parts of the network. It is worth emphasizing, that by building the hierarchical subnetwork as we did, CuttleNet demonstrated exceptional class predictive abilities, including correctly identify ACs with a staggeringly high F1 score for even the most ambiguous class of cells [9] exceeding 99.8%, underscoring the precision and reliability of the CuttleNet in discerning cellular identities, even in the presence of gene measurement gaps such as occur when merging datasets.

We finally benchmarked our networks against CellTICS, the most cutting-edge published hierarchical NN model for

[8]See statistical comparison related to Figure 3.

[9]It is highly plausible that the original AC scRNAseq data was over segmented in the original clustering.

simultaneous cell class and subclass classification (Yin & Chen, 2023). Appreciating the extensive benchmarking of CellTICS against other leading methods, and its superior performance, we deemed it the best reference for evaluating our models. To fully compare it, we trained it with versions of our retina expression matrix both with and without imputed values. Furthermore, we trained both the 1 Layer architecture, which is most similar to our own, and the developer's recommended 5 Layer network (**Sup. Fig. 12**). To ensure robust comparisons, we re-trained and evaluated CellTICS five different ways by shuffling our dataset and implementing an 80/20 train/test split. This approach facilitated the generation of replicate F1 scores for CellTICS, against which we compared our models' replicated performance metrics.

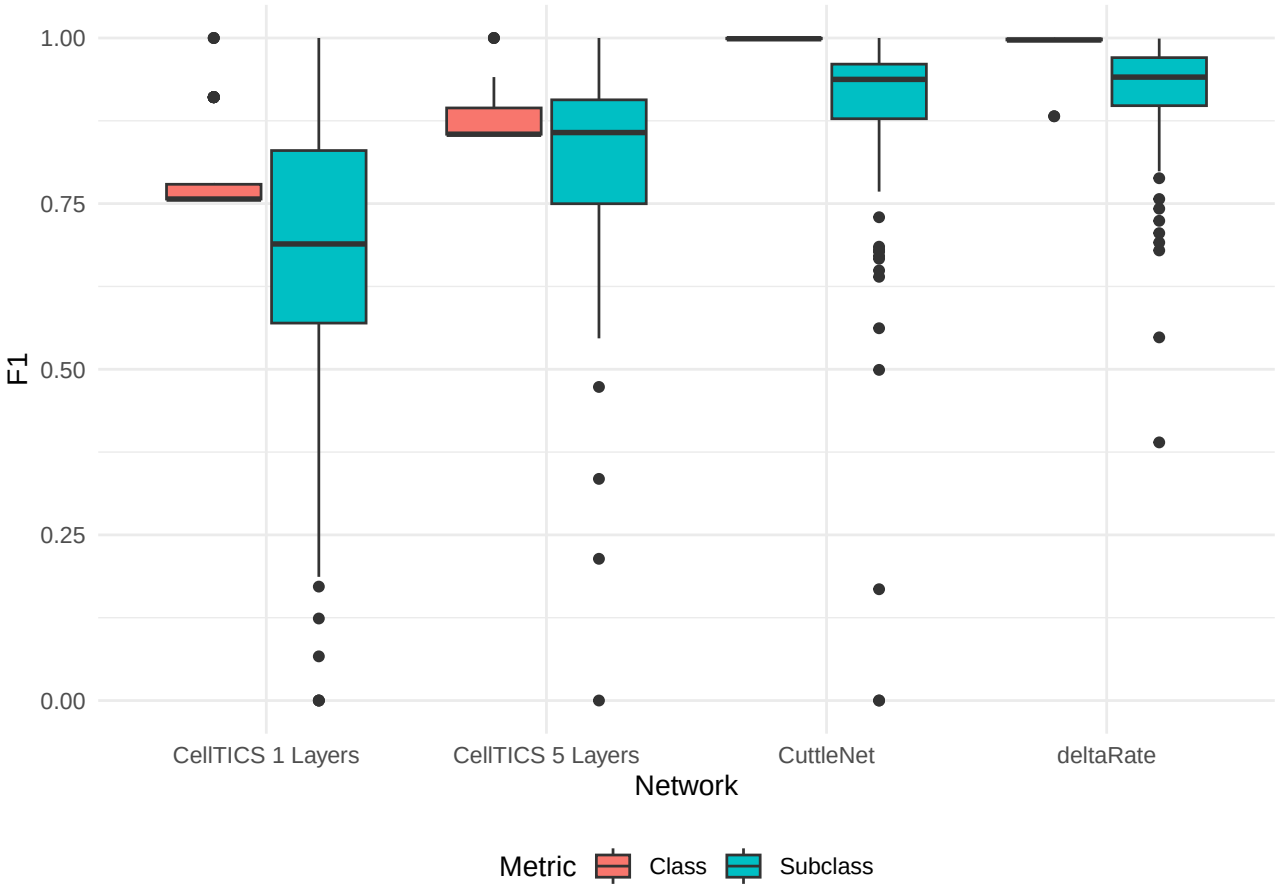


Figure 9. Benchmarking CuttleNet vs. CellTICS

Both CuttleNet and the deltaRate-trained GLM demonstrated significant superiority over CellTICS for our specific application[10]. In an effort to minimize the risk of a type 1 error coming from performing too many statistical tests on uninteresting parts of the combinatoric space we explored, we made the philosophic decision to focus our statistical testing on only the best performing versions of each model. Thus, we performed an ANOVAs and post-hoc Tukey test on the class and subclass F1 scores for CuttleNet[11], the GLM trained with the deltaRate protocol[12], and the gold-standard CellTICS model[13]. In both the class and subclass focused ANOVA we rejected the null hypothesis as expected from **Fig. 9**, with $f(2) = 2783, p = 10^{-16}$

[10]It's important to note the distinct nature of our task compared to typical scRNAseq classification challenges addressed by CellTICS and similar models. These models are typically designed to leverage a much larger gene pool, often in the tens of thousands, and do not typically account for imputed values. Despite this, the comparison with CellTICS underscores the significance of our contribution in addressing the challenges of using scRNAseq data to spSeq

[11]epochs=25, stopping=0, l1=0.001

[12]deltaRate=0.75, epochs=100, stopping=0, l1=0, imputed 0s

[13]5 layers and imputed 0s

and $f(2) = 25.48, p = 10^{-11}$, respectively. It should come as no surprise based on **Fig. 9** that post-hoc testing also rejected the null hypothesis for each of our models against CellTICS with $p = 0$. In comparing CuttleNet with the much simpler GLM, it is important to point out that for class classification performance CuttleNet is significantly better than the deltaRate trained GLM $p = .006$, but for subclass performance there is no significant difference, $p = .012$ with a Bonferoni corrected, $\alpha = 0.008$ which we think is appropriate given the 6 statistical tests performed between the Tukey tests we here assessed. Regardless, even if one wishes to use an uncorrected $\alpha = 0.05$ to reject the null hypothesis, the difference in performance between the models is negligible, with a Cohen's $d = 0.228$. Statistics aside, the practical difference between CuttleNet and the GLM is negligible, which is critically important for us because we learn 1) The use of the simple to interrogate GLM for unbiased gene selection is appropriately powerful, and 2) CuttleNet does not sacrifice this power as a result of its added complexity, which occurs for Fc architectures with added layers on this task. On the contrary, CuttleNet is a superior model because it selectively ignores imputed values, and thus solves a real world problem facing researchers who have had to merge datasets in this very common manner. Our ablation studies highlight (**Table 1**) how models might use the 0s as "obvious" class markers[14] too cheat. In our case the practical difference of this cheating is small, but on other datasets it could have a dramatic impact.

3.5. Discussion

This work presents a significant stride in bridging the pressing methodological gap between spSeq and scRNAseq using machine learning. Our novel approach in CuttleNet, inspired by the biological NNs of cuttlefish & the coarse-to-fine model of perception, demonstrates a hierarchical architecture effectively classifying retinal cell subclasses with a dramatically reduced set of genes, and in the face of inconsistent datasets. It is worth emphasizing, that the universal nature of nucleic acids is such that these tools apply to any biologic system that can be probed with spSeq. Moreover, while it is beyond the scope of this discussion, we are successfully applying these tools to Murine retina[15].

One of the most important applications of this research is the effective determination of a reduced gene set, a necessity born from the constraints of current spSeq tools. This constraint, rather than being a limitation, became an opportunity to refine our machine learning approach and gave rise to a novel architecture. By employing gradient-based gene selection using our biased dataset curricula, we were able to significantly reduce the gene set required for accurate classification, from 18k to just 225. This dramatic reduction not only demonstrates the efficiency of our NN strategy but also lays the groundwork for future research where such constraints are common. In fact, even when spSeq is able to measure the same number of targets as scRNAseq, it is likely few researchers will do so given the greater cost in terms of reagents and time[16]. Thus, minimal classification panels will likely be a persistent feature of spSeq studies.

A data cleaning choice we made in this work was the imputation of zeros. Data imputation is often a necessity born of missing values, as it was in this work. Specifically, we merged three different scRNAseq studies which were created at different times. The result of this history is that the more recently an scRNAseq study was performed, the more genes were measured. This implies the omitted genes simply were available for measurement when the studies were carried out. We could have chosen to be conservative and deleted genes from more recent studies not contained in older studies, but this would have included specific gold standards such as RBPMS. Moreover, we anticipated some unpredictable genes which would be important for subclass separation would be lost if we did this. Thus, we decided to perform data imputation to maintain the largest possible expression matrix. Due to the type of neural networks used for this work, a constraint of the underlying matrix operations is that the matrix must contain real numbers, which made it impossible to use an indicator like `NaN`. Moreover, because we were dealing with log normalized gene expression values negative numbers were similar not possible. Finally, because of the nature of these log normalized values being increasingly important to the network as they grow larger, 0 was the natural choice for imputation. Despite the potential problems this decision could have made, it ultimately inspired CuttleNet and was thus surprisingly valuable.

The exploration of incorporating a coarse-to-fine strategy in our models led to the success we demonstrated with the CuttleNet architecture. This success is a testament to the importance of aligning machine learning models with biological insights. CuttleNet's ability to segregate and process class-specific information, paralleling biological perception, not only enhanced its accuracy but also its applicability to real-world problems in spSeq. Furthermore, the robustness of CuttleNet in

[14] This is a feature of our specific datasets, as each focused on a disjointed subset of retinal cell subtypes.

[15] An example of inference on Xenium data is visualized in the CuttleNet jupyter notebook.

[16] At present, measuring a single retina costs $\sim \$625$ per probe for a single gene and requires 2 days of imaging time for the 300 gene panel, a soon to be available 500 gene option will require 5 days of imaging time. There is no reason to think this trend will significantly change due to the physics involved, and extreme competition already present in this novel market.

the presence of imputed data highlights its practicality. In real-world scenarios, data is often incomplete or merged from different sources, making a model's ability to handle such imperfections crucial. CuttleNet's design inherently accounts for these common data challenges, making it a reliable tool for researchers.

3.5.1. Related NN Bioinformatic Applications

spSeq is a novel challenge and opportunity for wet-lab technicians and bioinformaticians. The challenges in particular prompted our application research to grow independently from a very specific wet-lab bottleneck, however, as we will discuss it is not without precedent or parallel in the deep learning and bioinformatics communities.

Bioinformaticians have made impressive contributions to the analysis of scRNAseq data, including the powerful tools and methods where insights can be gleaned from the tens of thousands of genes sequenced in a typical experiment. One particularly important application from this space is the re-identification of expected cell classes and subclasses discovered by historic low-throughput wet-lab techniques, as well as the identification of novel subclasses not previously discriminable without scRNAseq. The success of these methods is a testament to the power of bridging disciplines as intellectually distinct as histology (Bleckert et al., 2014; Bae et al., 2018) and graph theory (Wang et al., 2021; Zeng et al., 2022).

spSeq brings three new parameters to the bioinformatician interested in the questions histologists have been trying to answer: X, Y and Z. However, before these uniquely important parameters can be used to press multi-omics research forward, the inability for commercially available systems to measure whole-genome scale transciptomic information necessitates novel computational tools. Thankfully, supervised learning is well positioned to fill this need while wet-lab instruments catch up to the most sophisticated classification techniques that already exist. NNs are not the only machine learning based supervised classifier strategy being employed in this space, see for instance the random forest strategy by Tan & Cahanor (2019), or the many other methods reviewed by Qi (2020) or Petegrosso (2020) and colleagues. This discussion would be incomplete without reference to Cui and colleagues' (2023) pioneering work creating a foundation model for scRNAseq data. The potential of this GPT can not be understated. Unfortunately, at this time it has only been trained on human data, and no retinal cells, thus was not applicable to our research.

CuttleNet, and the earlier GLMs we developed rely on a straightforward strategy of treating a matrix of gene expression values as parameters associated with a label, the cell subclass and by extension class. This general scheme of thinking about labeled scRNAseq data has been used successfully by other authors. For instance, Lin and colleagues (2017) compared NNs similar to the GLMs we worked with here, as well as some slightly more involved NNs incorporating prior knowledge in order to cluster scRNAseq data. Their results demonstrate the plausibility of this approach relative to other machine learning methods like k-means clustering. We differ markedly in our application as the dataset they are trying to cluster contained 17k genes, relative to our 300. On the other hand, their target population was much less diverse than ours, attempting to classify 16 cell classes, relative to our 125 subclasses.

This study brings up the other main problem faced by those working with scRNAseq data, dimensionality reduction. Our initial research focus was on exactly this, specifically how does one find the minimal set of genes that can maximally segment cell populations? **Fig. 2** shows the answer we found with the unbiased Gradient strategy. It goes without saying that many important historic methods also exist to this end, like UMAP transformations, which validate our results **Fig. 4**, however, alone they can not answer our question because they do not preserve knowledge of the specific genes of interest. Thus, by solving this challenge by interrogating the gradients of the diverse set of networks we trained, our approach differs significantly from other historic uses of NNs for dimensionality reduction in scRNAseq data (Lin et al., 2017; Wang et al., 2021).

3.5.2. Coarse-to-Fine Cell Classification

Biologic perception relies on the coarse-to-fine processing strategy, a hierarchical approach where broad, general information is processed initially, followed by increasingly specific and detailed analysis. This strategy also has straightforward parallels with historic approaches to cell classification. Initial classification schemes relied heavily on morphological characteristics which allowed the rough division of cell classes. For example, in the retina morphology allowed the differentiation of photoreceptor cells, based on their long photosensitive protrusions, relative to RGCs whose axons clearly conveyed the integrated retinal information to the brain. As technological capabilities advanced, immunohistochemical signatures provided a more nuanced understanding of cell subclasses, such as distinguishing photoreceptive cones on the basis of which wavelength of light they could detect. The advent of scRNAseq offered an unprecedented window into the complexities of cell subclasses by clustering and analyzing the expression of tens of thousands of gene transcripts at the single-cell level. In

the Murine retina, where previous techniques had expected little more than a dozen subclasses of RGCs (Masland, 2011), scRNAseq revealed more than forty(Tran et al., 2019); each responsible for a unique computation important to the Murine visual system. The knowledge gained at each of these methodological levels is thus baked into the thinking and classification schemes dreamt up by human researchers in the coarse-to-fine manner that these techniques emerged and are still employed today.

3.5.3. Related Hierarchical Networks

Hierarchical classification as a NN architectural strategy is in many ways another embodiment of the Coarse-to-Fine perceptual strategy. Specifically, many exciting projects have analogized this idea, by segmenting the task into steps they introspectively perceive their own mind is taking (Wang et al., 2012; Zhu & Bain, 2017; Sarlin et al., 2019), by directly trying to build a system in the style of empirically observed brain structures (Hafner et al., 2017), or even by mimicking behavioral strategies employed by by biologists (Yin & Chen, 2023).

Our work is the least related to the most common hierarchical classification application, segmenting or classifying images based on statistical methodologies. For the sake of completeness, we will point out several notable projects in this space. Wang and colleagues (2012) created the earliest network architecture with several hidden layers that could learn separate classification features we are aware of. The output of this hierarchical strategy was a vote by the separate systems, which differs from CuttleNet's directing of information to specific hidden parts of the network. A more similar hierarchical model is bCNN by Zhu and Bain (2017), which handled hierarchical structures of image data, emphasizing a layered, systematic approach to learning from general to specific features. Of course, the use of convolutional layers and the focus on images is a very different architecture and application from our own. Similarly, in terms of task goals, Sarlin and colleagues devised another analogous network known as HF-Net (2019) that sought to semantically segment both global descriptors and local features from an image. Finally, we want to acknowledge the work of Yan et al (2015) who discovered the possibility of combing a label tree with the hierarchical structure of a CNN in their HD-CNN architecture . **The powerful ideas these authors have investigated, should be considered as a high priority future direction for enhancing CuttleNet-like architectures as the classification of cells can similarly be embedded in a much richer hierarchy than our simple class vs. subclass distinction**. Thus, as spSeq continues to progress to larger and thus more heterogeneous tissue samples it will soon be important to be able to distinguish cells for example first by organ, then tissue type, before class and subclass. This predicted future use case, will be increasingly relevant as technological advances combine spSeq, large volume tissue imaging and relevant image processing tools like Cellpose 3D and NuMorph (Krupa et al., 2020). In fact, these technologies will ultimately necessitate adapting the work of Yu and colleagues (2020) who applied the coarse-to-fine strategy for segmenting and thus classifying information in 3D medical images with their C2FNAS architecture.

Just as we draw inspiration for our architecture from the nervous system's anatomy, so too did Hafner and colleagues for their hierarchical model ThalNet (2017). This recurrent NN model was inspired by communication pathways between the thalamus and cortex, as features were shared with neo-cortical-inspired modules through a routing center likened to a thalamus. This clever design allowed flexible sharing of information over time, and allowed for temporal hierarchies to emerge with impressive performance results. Unlike CuttleNet, ThalNet uses static reading weights to learn a fixed connectivity patterns to create a temporal hierarchy. CuttleNet's dynamic reading scheme is more akin to an anatomical hierarchy.

The most similar project to our own from the perspective of hierarchical NNs is the CellTICS network by Yin & Chen (2023) which is also an explicitly hierarchical NN designed to distinguish cell classes and cell subclasses via supervised training with pre-labeled scRNAseq data. The authors demonstrate through extensive benchmarking that their cutting edge model is both the best architecture for this task published to date, and is by design interpretable. The implementation of this interpretability is where the major distinctions between our architectures reside. CellTICS uses the Reactome gene-function map to identify marker genes from biochemical pathways that in aggregate give rise to cell classes. By weighting these parameters for the purpose of classifying cell subclass, the authors achieve insight into why the network designates a cell as it does. In contrast, CuttleNet has generic subnetworks, lacking explicit biological annotation. Instead mapping cell classes-to-subclasses in a coarse-to-fine manner allows the network to learn the appropriate values without prior knowledge. Additionally, the dynamic routing of genes through our tentacle-like subnetworks affords CuttleNet resistance to imputed values, which can have undo influence with such abridged datasets, as we demonstrate with our ablation studies that both CellTICS and the GLMs' performance suffers when contaminated genes are removed. As spSeq catches up to scRNAseq in the number of genes it can measure, interpretable models like CellTICS will be increasingly valuable. For now, we demonstrate in our benchmarking experiments CuttleNet's coarse-to-fine perception enhances performance, at least on small

expression matrices generated by spSeq.

3.6. Conclusion

In conclusion, our study breaks ground by applying NN based cell classification to the context of spSeq, leveraging the strengths of machine learning. CuttleNet, our hierarchical NN model, efficiently classifies cells with a markedly reduced gene set. This timely solution to the limitations posed by current spSeq tools is also of intellectual interest as an example of using biology to inform model designs. The implications of this work extend beyond the specific application of classifying retinal cells. It offers a versatile framework that can be adapted to any biologic tissue, or machine learning task where hierarchical classification systems are beneficial. Furthermore, the methodologies developed here for gene selection and model training can serve as valuable guides for similar challenges in other domains.

In essence, this study is not just about achieving high accuracy in cell classification. It's about the thoughtful integration of biological insights with machine learning techniques in the face of pressing problems. We hope the path we set forward here will benefit future researchers; both those trying to apply spSeq to their tissue on the bench, as well as those interested in applying machine learning to solve biological problems. After all, the insights we gain from biology may one day recursively improve machine learning.

Acknowledgements

I would like to thank my PhD mentor Dr. Alon Poleg-Polsky for his oversight and support of this project. We completed this project over the course of several years, beginning from a evolutionary learning algorithm, before settling on the final version of our gene selection algorithm presented here. Specifically worth highlighting from this project history are his contribution of the key question "How many subtypes can be classified with some number of genes?" as well as the technical step we used to filtered the raw gene list for computational efficiency.

I would also like to thank my other committee members Drs. Yaning Liu, Joshua French, and Erin Austin. In addition to their valuable insights and evaluations of this master's project, I would like to especially acknowledge their support throughout my studies in the UC Denver MS Statistics program. Despite lacking a formal mathematical background, Yaning allowed me to take his regression course during the first summer of the COVID pandemic when laboratory work was not possible. Joshua and Erin subsequently encouraged me to formerly enroll in the program after I followed this regression course down the rabbit hole. During all of this, Alon supported and encourage this side-quest to my PhD studies. It is an honour to have learned from and now have these four individuals evaluating the fruits of my labor here presented.

Impact Statement

This paper presents an application which will advance the field spatial biology by applying and contributing to the field of Machine Learning. There are many potential societal consequences of spatial sequencing, so this work will likely indirectly or directly contribute to medical discoveries and the associated benefits and consequences there of. None of the more machine learning focused impacts from CuttleNet we feel must be specifically highlighted here.

MS Project Format Statement

This paper was submitted to the ICML conference and is written according to the rules for this audience. Difference from the original manuscript and this version submitted to the UC Denver Statistics program include: 1) Moving supplemental Figures into the body of the text, 2) The verbose methods section (which was originally a supplement) was swapped with the concise methods section, and 3) an additional paragraph explaining the data imputation logic was added to the discussion. In addition to these material changes, the two column format was replaced with a single column format.

References

Ahnelt, P. K., Schubert, C., Kübber-Heiss, A., Schiviz, A., and Anger, E. Independent variation of retinal S and M cone photoreceptor topographies: A survey of four families of mammals. *Visual Neuroscience*, 23(3-4):429–435, August 2006. ISSN 0952-5238. doi: 10.1017/S095252380623342X.

Bae, J. A., Mu, S., Kim, J. S., Turner, N. L., Tartavull, I., Kemnitz, N., Jordan, C. S., Norton, A. D., Silversmith, W. M.,

Prentki, R., Sorek, M., David, C., Jones, D. L., Bland, D., Sterling, A. L. R., Park, J., Briggman, K. L., and Seung, H. S. Digital museum of retinal ganglion cells with dense anatomy and physiology. *Cell*, 173(5):1293–1306.e19, May 2018. ISSN 0092-8674. doi: 10.1016/j.cell.2018.04.040. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6556895/.

Bleckert, A., Schwartz, G. W., Turner, M. H., Rieke, F., and Wong, R. O. L. Visual Space Is Represented by Nonmatching Topographies of Distinct Mouse Retinal Ganglion Cell Types. *Current Biology*, 24(3):310–315, February 2014. ISSN 0960-9822. doi: 10.1016/j.cub.2013.12.020. URL https://www.cell.com/current-biology/abstract/S0960-9822(13)01579-0. Publisher: Elsevier.

Collin, S. P. A web-based archive for topographic maps of retinal cell distribution in vertebrates. *Clinical and Experimental Optometry*, 91(1):85–95, January 2008. ISSN 0816-4622. doi: 10.1111/j.1444-0938.2007.00228.x. URL https://doi.org/10.1111/j.1444-0938.2007.00228.x. Publisher: Taylor & Francis _eprint: https://doi.org/10.1111/j.1444-0938.2007.00228.x.

Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., and Wang, B. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, 2023.

Goetz, J., Jessen, Z. F., Jacobi, A., Mani, A., Cooler, S., Greer, D., Segal, J., Shekhar, K., Sanes, J., and Schwartz, G. W. Unified classification of mouse retinal ganglion cells using function, morphology, and gene expression. pp. 29, 2022.

Hafner, D., Irpan, A., Davidson, J., and Heess, N. Learning Hierarchical Information Flow with Recurrent Neural Modules. 2017.

Hahn, J., Monavarfeshani, A., Qiao, M., Kao, A. H., Kölsch, Y., Kumar, A., Kunze, V. P., Rasys, A. M., Richardson, R., Wekselblatt, J. B., Baier, H., Lucas, R. J., Li, W., Meister, M., Trachtenberg, J. T., Yan, W., Peng, Y.-R., Sanes, J. R., and Shekhar, K. Evolution of neuronal cell classes and types in the vertebrate retina. *Nature*, 624(7991):415–424, December 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06638-9. URL https://www.nature.com/articles/s41586-023-06638-9. Number: 7991 Publisher: Nature Publishing Group.

Heukamp, A. S., Warwick, R. A., and Rivlin-Etzion, M. Topographic Variations in Retinal Encoding of Visual Space. *Annual Review of Vision Science*, 6(1):237–259, 2020. doi: 10.1146/annurev-vision-121219-081831. URL https://doi.org/10.1146/annurev-vision-121219-081831. _eprint: https://doi.org/10.1146/annurev-vision-121219-081831.

Holmgren, C. D., Stahr, P., Wallace, D. J., Voit, K.-M., Matheson, E. J., Sawinski, J., Bassetto, G., and Kerr, J. N. Visual pursuit behavior in mice maintains the pursued prey on the retinal region with least optic flow. *eLife*, 10:e70838, October 2021. ISSN 2050-084X. doi: 10.7554/eLife.70838. URL https://doi.org/10.7554/eLife.70838. Publisher: eLife Sciences Publications, Ltd.

Hughes, A. A comparison of retinal ganglion cell topography in the plains and tree kangaroo. *The Journal of Physiology*, 244(1):61P–63P, January 1975. ISSN 0022-3751.

Krupa, O., Fragola, G., Hadden-Ford, E., Mory, J. T., Liu, T., Humphrey, Z., Rees, B. W., Krishnamurthy, A., Snider, W. D., Zylka, M. J., Wu, G., Xing, L., and Stein, J. L. NuMorph: tools for cellular phenotyping in tissue cleared whole brain images. preprint, Neuroscience, September 2020. URL http://biorxiv.org/lookup/doi/10.1101/2020.09.11.293399.

Lin, C., Jain, S., Kim, H., and Bar-Joseph, Z. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Research*, 45(17):e156–e156, September 2017. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkx681. URL https://academic.oup.com/nar/article/45/17/e156/4056711.

Masland, R. H. Cell Populations of the Retina: The Proctor Lecture. *Investigative Ophthalmology & Visual Science*, 52(7):4581–4591, June 2011. ISSN 0146-0404. doi: 10.1167/iovs.10-7083. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3175966/.

Petegrosso, R., Li, Z., and Kuang, R. Machine learning and statistical methods for clustering single-cell RNA-sequencing data. *Briefings in Bioinformatics*, 21(4):1209–1223, July 2020. ISSN 1477-4054. doi: 10.1093/bib/bbz063. URL https://academic.oup.com/bib/article/21/4/1209/5519426.

Qi, R., Ma, A., Ma, Q., and Zou, Q. Clustering and classification methods for single-cell RNA-sequencing data. *Briefings in Bioinformatics*, 21(4):1196–1208, July 2020. ISSN 1477-4054. doi: 10.1093/bib/bbz062. URL https://academic.oup.com/bib/article/21/4/1196/5528236.

Sarlin, P.-E., Cadena, C., Siegwart, R., and Dymczyk, M. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12708–12717, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.01300. URL https://ieeexplore.ieee.org/document/8953492/.

Shekhar, K., Lapan, S. W., Whitney, I. E., Tran, N. M., Macosko, E. Z., Kowalczyk, M., Adiconis, X., Levin, J. Z., Nemesh, J., Goldman, M., McCarroll, S. A., Cepko, C. L., Regev, A., and Sanes, J. R. COMPREHENSIVE CLASSIFICATION OF RETINAL BIPOLAR NEURONS BY SINGLE-CELL TRANSCRIPTOMICS. *Cell*, 166(5):1308–1323.e30, August 2016. ISSN 0092-8674. doi: 10.1016/j.cell.2016.07.054. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5003425/.

Szél, , Röhlich, P., Gaffé, A. R., Juliusson, B., Aguirre, G., and Van Veen, T. Unique topographic separation of two spectral classes of cones in the mouse retina. *Journal of Comparative Neurology*, 325(3):327–342, 1992. ISSN 1096-9861. doi: 10.1002/cne.903250302. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cne.903250302. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cne.903250302.

Tan, Y. and Cahan, P. SingleCellNet: A Computational Tool to Classify Single Cell RNA-Seq Data Across Platforms and Across Species. *Cell Systems*, 9(2):207–213.e2, August 2019. ISSN 24054712. doi: 10.1016/j.cels.2019.06.004. URL https://linkinghub.elsevier.com/retrieve/pii/S2405471219301991.

Tran, N. M., Shekhar, K., Whitney, I. E., Jacobi, A., Benhar, I., Hong, G., Yan, W., Adiconis, X., Arnold, M. E., Lee, J. M., Levin, J. Z., Lin, D., Wang, C., Lieber, C. M., Regev, A., He, Z., and Sanes, J. R. Single-Cell Profiles of Retinal Ganglion Cells Differing in Resilience to Injury Reveal Neuroprotective Genes. *Neuron*, 104(6):1039–1055.e12, December 2019. ISSN 0896-6273. doi: 10.1016/j.neuron.2019.11.006. URL https://www.sciencedirect.com/science/article/pii/S0896627319309699.

Wang, J., He, H., Cao, Y., Xu, J., and Zhao, D. A Hierarchical Neural Network Architecture for Classification. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Wang, J., Yen, G. G., and Polycarpou, M. M. (eds.), *Advances in Neural Networks – ISNN 2012*, volume 7367, pp. 37–46. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-31345-5 978-3-642-31346-2. doi: 10.1007/978-3-642-31346-2_5. URL http://link.springer.com/10.1007/978-3-642-31346-2_5. Series Title: Lecture Notes in Computer Science.

Wang, J., Ma, A., Chang, Y., Gong, J., Jiang, Y., Qi, R., Wang, C., Fu, H., Ma, Q., and Xu, D. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nature Communications*, 12(1):1882, March 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-22197-x. URL https://www.nature.com/articles/s41467-021-22197-x.

Wässle, H., Grünert, U., Martin, P. R., and Boycott, B. B. Immunocytochemical characterization and spatial distribution of midget bipolar cells in the macaque monkey retina. *Vision Research*, 34(5):561–579, March 1994. ISSN 0042-6989. doi: 10.1016/0042-6989(94)90013-2.

Yan, W., Laboulaye, M. A., Tran, N. M., Whitney, I. E., Benhar, I., and Sanes, J. R. Mouse Retinal Cell Atlas: Molecular Identification of over Sixty Amacrine Cell Types. *The Journal of Neuroscience*, 40(27):5177–5195, July 2020a. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0471-20.2020. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0471-20.2020.

Yan, W., Peng, Y.-R., van Zyl, T., Regev, A., Shekhar, K., Juric, D., and Sanes, J. R. Cell Atlas of The Human Fovea and Peripheral Retina. *Scientific Reports*, 10(1):9802, June 2020b. ISSN 2045-2322. doi: 10.1038/s41598-020-66092-9. URL https://www.nature.com/articles/s41598-020-66092-9. Number: 1 Publisher: Nature Publishing Group.

Yan, Z., Zhang, H., Piramuthu, R., Jagadeesh, V., DeCoste, D., Di, W., and Yu, Y. HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2740–2748, Santiago, Chile, December 2015. IEEE. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.314. URL http://ieeexplore.ieee.org/document/7410671/.

Yin, Q. and Chen, L. CellTICS: an explainable neural network for cell-type identification and interpretation based on single-cell RNA-seq data. *Briefings in Bioinformatics*, 25(1):bbad449, November 2023. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbad449. URL https://academic.oup.com/bib/article/doi/10.1093/bib/bbad449/7461884.

Yu, Q., Yang, D., Roth, H., Bai, Y., Zhang, Y., Yuille, A. L., and Xu, D. C2FNAS: Coarse-to-Fine Neural Architecture Search for 3D Medical Image Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4125–4134, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72817-168-5. doi: 10.1109/CVPR42600.2020.00418. URL https://ieeexplore.ieee.org/document/9156654/.

Zeng, Y., Wei, Z., Pan, Z., Lu, Y., and Yang, Y. A robust and scalable graph neural network for accurate single-cell classification. *Briefings in Bioinformatics*, 23(2):bbab570, March 2022. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbab570. URL https://academic.oup.com/bib/article/doi/10.1093/bib/bbab570/6501353.

Zhu, X. and Bain, M. B-CNN: Branch Convolutional Neural Network for Hierarchical Classification, October 2017. URL http://arxiv.org/abs/1709.09890. arXiv:1709.09890 [cs].

A. Conference Materials and Methods

A.1. Hardware & Data

Experiments were conducted with an Intel i7-12700KF CPU, NVIDIA RTX 3060 GPU, and 128 GB RAM, running Ubuntu 22.04. NNs and analyses were performed in Conda with Python 3.9 using Jupyter Notebooks, primarily utilizing PyTorch. Data analysis was done in R. Code: anonymous.4open.science/r/CuttleNet-F81E

scRNA-seq datasets SCP509, SCP919, and SCP3 of retinal cells available through singlecell.broadinstitute.org/single_cell/ were reorganizes such that columns indicated log normalized gene measurements, and rows to individual cells. The gene set was curated based on inter-subclass variance to improve computational performance at the expense of being completely unbiased. Zeros were imputed for missing data. Labels were the studies' clustering proposals.

A.2. Gradient-Based Gene Selection

Independent NNs for each subclass were trained on their target and other subclasses in a controlled manner to create balanced datasets with equal representation of desired and noise subclasses. Size discrepancies were resolved by either replicating instances in the smaller class or excluding instances from the larger class. The NNs, were feed-forward fully connected networks (Fc) with a single hidden layer twice the size of the labels. More complicated architectures with different sizes/numbers of hidden layers were explored but found unnecessary. Standard training procedures for data shuffling, splitting, and early stopping were utilized, as were fixed seeds for reproducibility. Cross-entropy loss and the Adam optimizer were employed for training. Each networks' gradients were calculated against test data to rank genes. Gradients with respect to the input genes highlight feature relevance. Repeating this collating of gene rankings in a data frame provides a consolidated, hierarchy-preserving ordering of genes by subclass. To aggregate this into a ranked list, our algorithm traversed the rankings and extended a list with each newly encountered gene.

A.3. Experiment 1: Blended Cost Function

A custom loss function weighting subclass and class predictions was tested with varying blending ratios (ClassWeight). This experiment was replicated five times for each ratio.

A.4. Experiment 2: Dynamic Variation of ClassWeight

We dynamically varied ClassWeight during training to assess the impact on model performance. This mimicked the progressive learning approach of focusing on broad class features before moving on to subclass details.

A.5. Experiment 3: Hierarchical Network Architecture

CuttleNet consists of a primary class classifier and multiple subclass classifiers, each responsible for a specific class' members (**Fig. 1, Sup. Fig. ??**). Each subnetwork consists of a two-layer Fc with the input containing only the genes free from imputation for the respective classes or subclasses being predicted. Output layers has as many nodes as the classes or subclasses being predicted. Hidden layers have twice as many nodes. Hidden layers use ReLU while predictions use log softmax. Each subclass subnetwork, dynamically selected via a ModuleDict, can be envisioned as a tentacle that is engaged only when the respective class is predicted.

A.6. Hyperparameter Optimization via Grid Search

A grid search optimized training epochs, early stopping criteria, and L1 regularization for CuttleNet, Fcs (where additional search across number and size of hidden layers was also undertaken). 1 layer and 5 layer CellTICS were trained with and without access to imputed data. Each condition was assessed on a test set, replicated five times.

B. Supplemental Figures

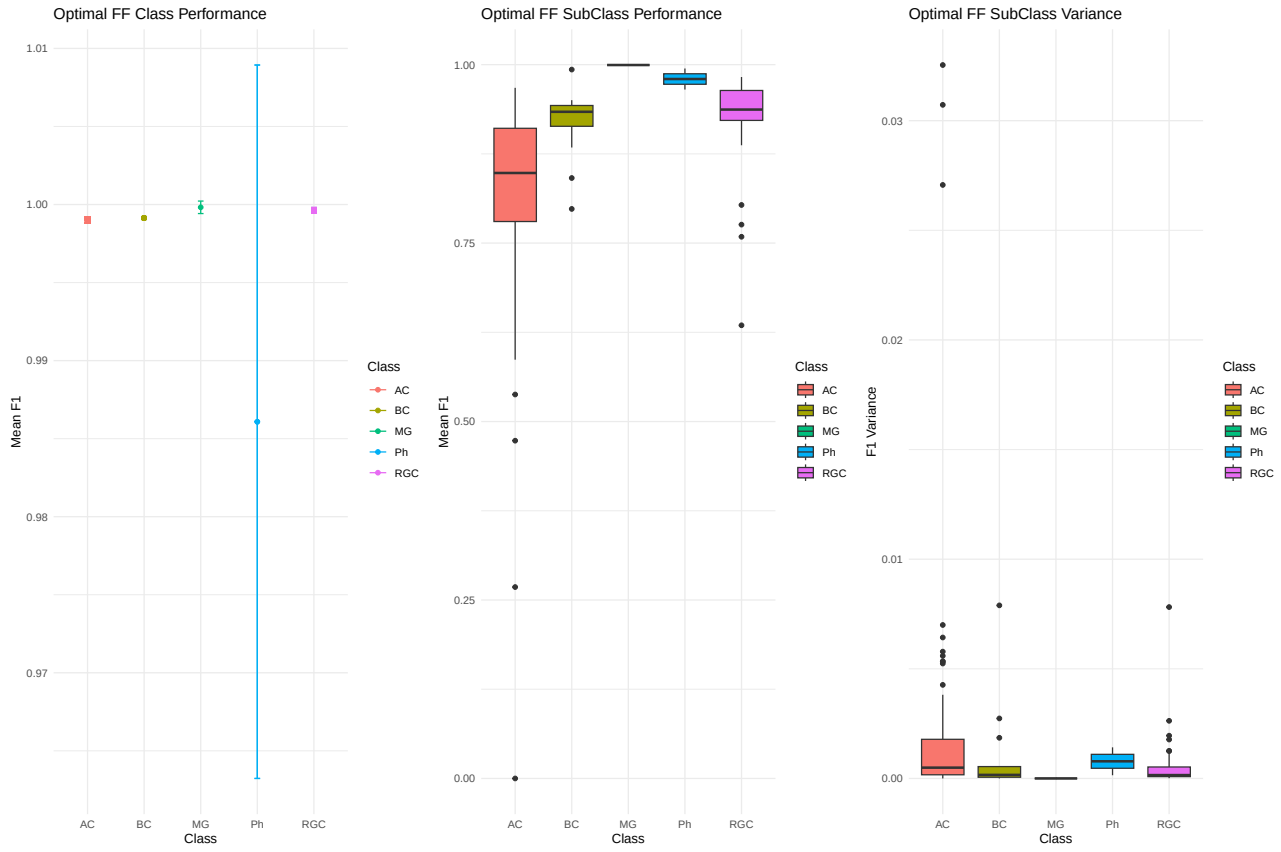


Figure 10. Performance of GLM on each class and subclass for the optimal parameter set, defined as the maximum sum of the median F1 scores for both subclass and class.

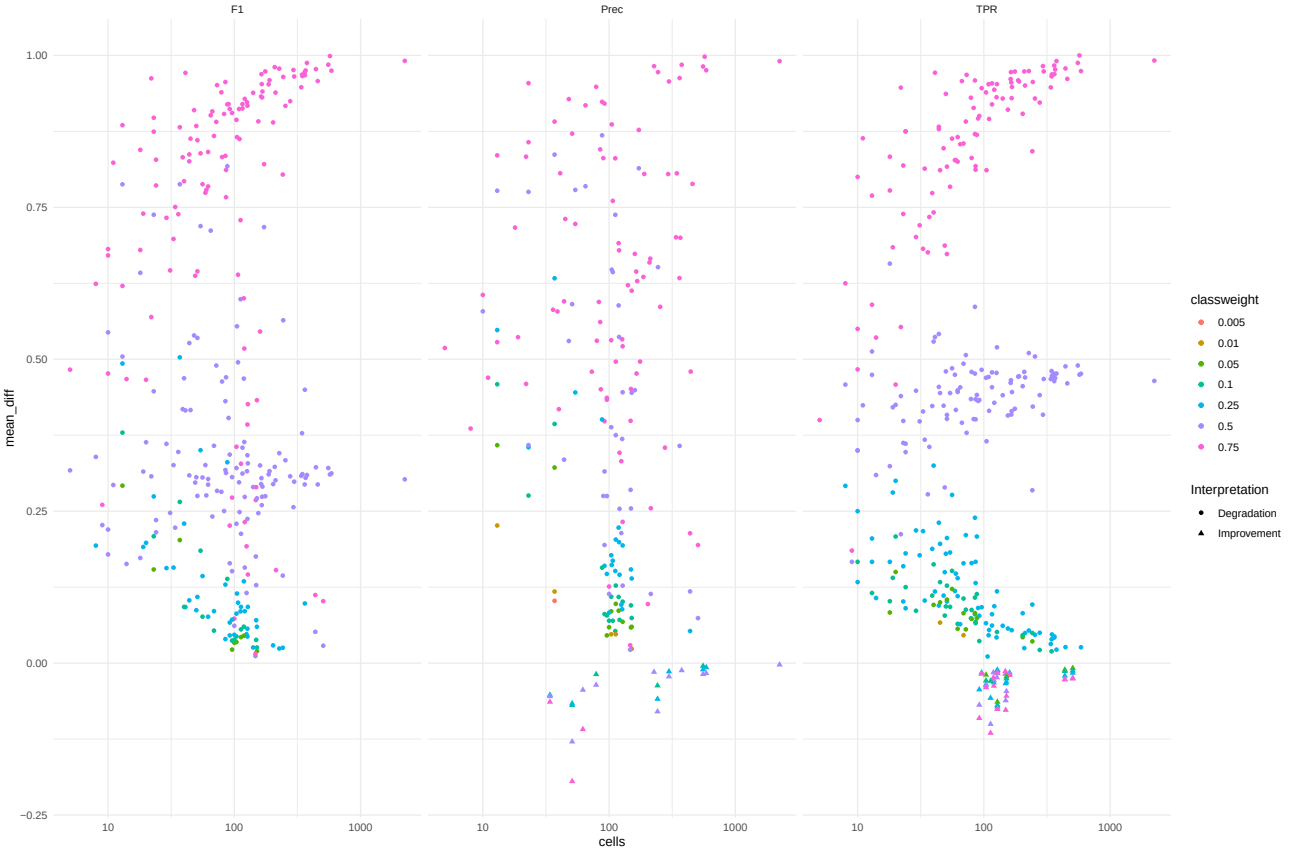


Figure 11. CostWeight experiment results, after an ANOVA significantly different subclasses were identified for differnt CostWeight conditions and plotted in relation to the number of cells in a given subtype and the magnitude of their difference from the unchanged points. Panels indicate post-hoc resultant points for F1, Prec and TPR. A negative value indicates improvement to the GLM trained with ClassWeight=0.

Table 2. Median F1 scores for each model, with mean standard deviations from the replicate runs. 0s indicates if the model had access to imputed zeros. Optimized indicates the version of a given architecture found via grid search. Control GLM indicates the same hyperparameters used to conduct the deltaRate and ClassWeight experiments. CellTICS models are the gold standard.

EXPERIMENT	SUBCLASS F1	CLASS F1	0S
OPTIMIZED CUTTLENET	93.8±1.5	99.9 ± 10^{-2}	×
DELTARATE GLM	94.1±1.3	99.7±0.2	√
CONTROL GLM	94.1±1.2	99.2±0.3	√
CLASSWEIGHT GLM	94.1±1.2	98.9±0.5	√
OPTIMIZED GLM	93.6±0.6	99.8 ± 10^{-3}	√
DELTARATE GLM	92.8±0.7	99.9 ± 10^{-3}	×
CELLTICS 5 LAYER	85.7±3.8	85.5±0.5	√
CELLTICS 5 LAYER	83.5±4.1	85.0±0.3	×
CELLTICS 1 LAYER	68.9±7.2	75.6±1.0	√
CELLTICS 1 LAYER	66.0±7.3	74.7±0.6	×

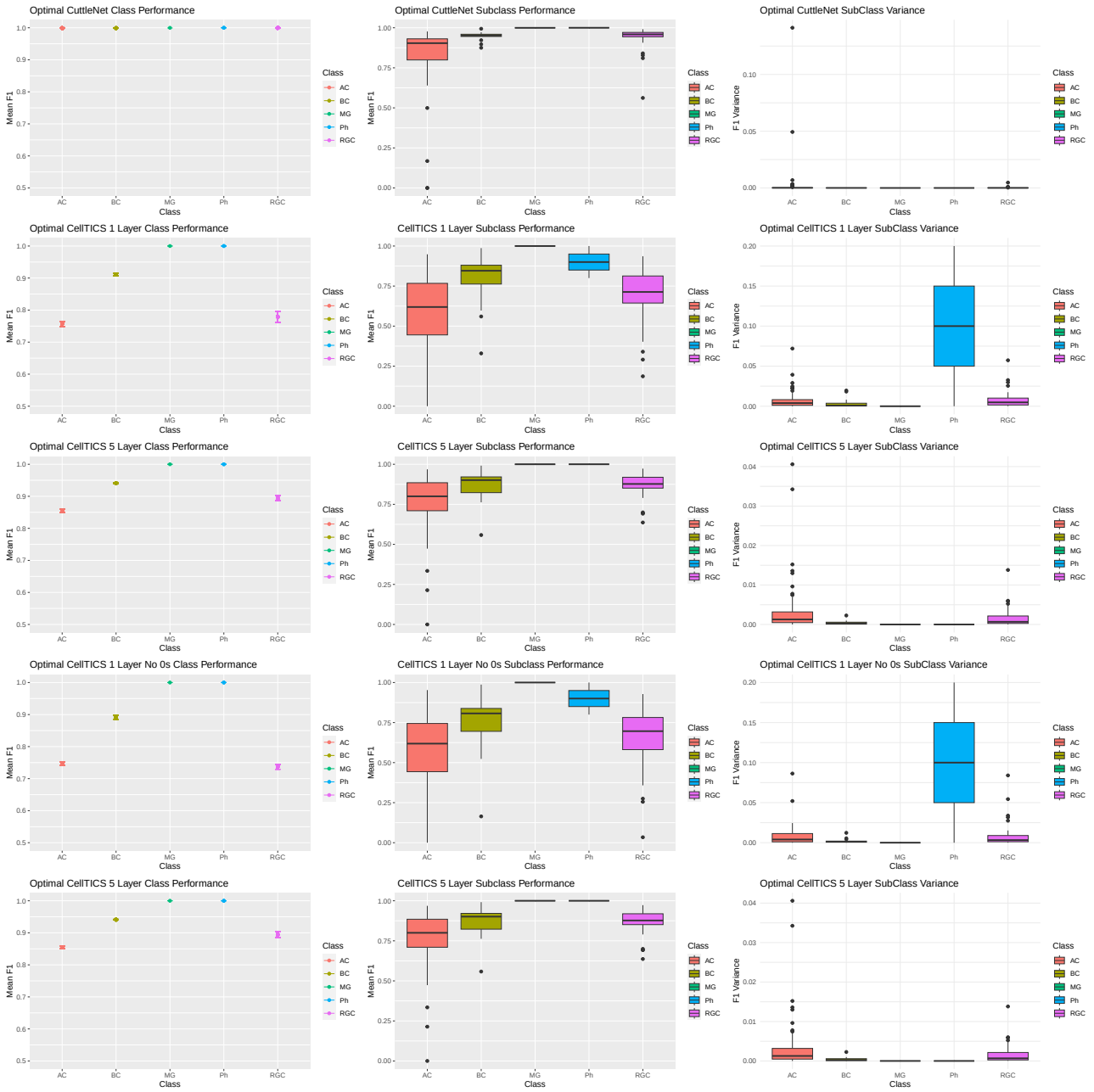


Figure 12. Optimal versions of each CuttleNet network found after perfroming appropriate grid searches for each architecture of experiment. Each experimental condition in the search included five replicates of training and evaluating the respective model; an 80/20 train/test split was done in all cases with the metrics evaluated here coming only from the testing dataset.