This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2022.3173587, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Semi-supervised Deep Learning for Cell Type Identification from Single-Cell Transcriptomic Data

Xishuang Dong, Shanta Chowdhury, Uboho Victor, Xiangfang Li, and Lijun Qian

***Abstract*—Cell type identification from single-cell transcriptomic data is a common goal of single-cell RNA sequencing (scRNAseq) data analysis. Deep neural networks have been employed to identify cell types from scRNAseq data with high performance. However, it requires a large mount of individual cells with accurate and unbiased annotated types to train the identification models. Unfortunately, labeling the scRNAseq data is cumbersome and time-consuming as it involves manual inspection of marker genes. To overcome this challenge, we propose a *semi-supervised learning model* "SemiRNet" to use unlabeled scRNAseq cells and a limited amount of labeled scRNAseq cells to implement cell identification. The proposed model is based on recurrent convolutional neural networks (RCNN), which includes a shared network, a supervised network and an unsupervised network. The proposed model is evaluated on two large scale single-cell transcriptomic datasets. It is observed that the proposed model is able to achieve encouraging performance by learning on the very limited amount of labeled scRNAseq cells together with a large number of unlabeled scRNAseq cells.**

***Index Terms*—Single-Cell Sequencing, Semi-supervised Learning, Recurrent Convolutional Neural Networks, Joint Optimization**

I. Introduction

Single-cell RNA sequencing (scRNAseq) enables the profiling of the transcriptomes of individual cells, thus characterizing the heterogeneity of biological samples since scRNAseq experiments are able to yield high volumes of data. For example, in a single experiment, the expression profile is up to 10^5 cells, at the level of the single cell [1]. It is not possible for traditional bulk RNAseq [2] to examine biological samples in such high-resolution.

Cell type identification is a common goal of scRNAseq data analytics to identify the cell type of each individual cell. It can be implemented by unsupervised methods with manual input [3]. To accomplish this, cells are first grouped into different clusters in an unsupervised manner, and the number of these clusters allows us to approximately determine how many distinct cell types are present. To attempt to interpret the identity of each cluster, marker genes are identified as those that are uniquely highly expressed in a cluster, compared to all other clusters. These canonical markers are then used to assign the cell types for the clusters by cross referencing the markers with lists of previously characterized cell type specific markers. To speed up this process, a set of annotation tools have been developed. For example, Kiselev *et al.* proposed scmap-cell [4] to project cells from an scRNA-seq data set onto cell types or individual cells from other experiments. Alquicira-Hernandez *et al.* built scPred [5] that was a new generalizable method for prediction of cell types through combining unbiased feature selection from a reduced-dimension space, and machine-learning classification. It can capture subtle effects of many genes and enhance the prediction accuracy. Aran *et al.* proposed SingleR [6] that is to implement the annotation of scRNA-seq by considering bulk transcriptomes. It enabled the subclustering of macrophages and revealed a disease-associated subgroup with a transitional gene expression profile intermediate. Recently, novel computational methods based

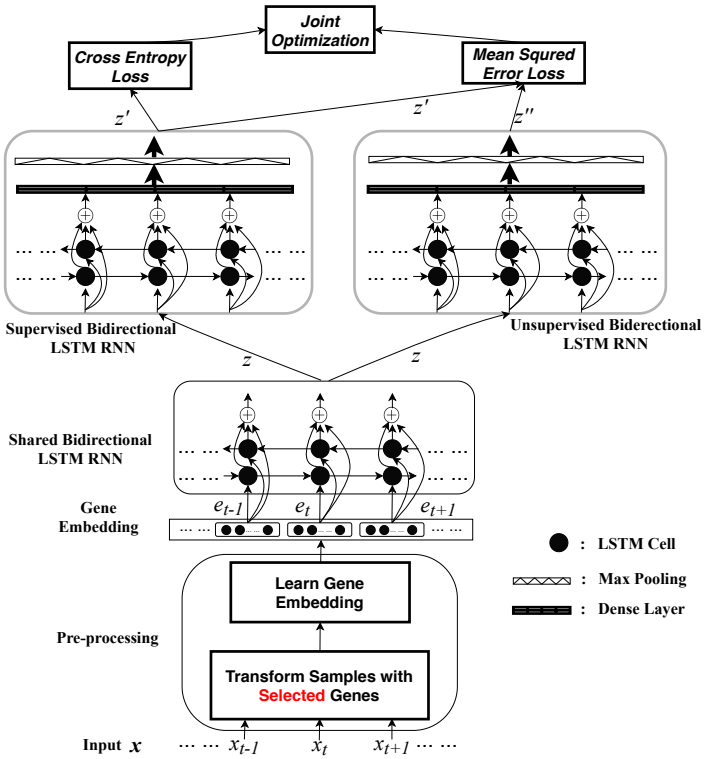


Fig. 1. Framework of the proposed semi-supervised learning. Input x is the cell. Cell types are available only for the labeled inputs and the associated cross-entropy loss component is evaluated only for those. z' and z'' are outputs from the supervised bidirectional LSTM RNN and the unsupervised bidirectional LSTM RNN, respectively. We jointly optimize cross entropy loss and mean squared error loss for supervised learning and unsupervised learning with these outputs. $\oplus$ is the concatenation operation.

X. Dong, S. Chowdhury, U. Victor, X. Li and L. Qian are with the Center of Excellence in Research and Education for Big Military Data Intelligence (CREDIT Center) and Center for Computational Systems Biology (CCSB), Department of Electrical and Computer Engineering, Prairie View A&M University, Texas A&M University System, Prairie View, TX 77446, USA. Email: xidong@pvamu.edu, schowdhury1@pvamu.edu, uboho.dpc@outlook.com, xili@pvamu.edu, liqian@pvamu.edu

on neural networks have been proposed to further improve the performance [3], [7], since cell type classification based on a large number of genes is more robust to noise. For example, Ma *et al.* proposed ACTINN (Automated Cell Type Identification using Neural Networks) [7] with simple neural networks of three neuron layers, which trains on datasets with predefined cell types and predicts cell types for other datasets based on the trained model. It uses all the genes to capture the features for each cell type instead of relying on a limited number of canonical markers.

However, it still faces two main challenge: 1). Annotating a large amount of individual cells costs intensive labor efforts and a time-consuming task, especially since single-cell sequencing technique becomes more and more popular and generate larger and larger datasets. For example, the number of cells involved the single-cell data analysis has been much larger like over $100,000$ [8], which will require more efforts for labeling the data. In addition, this may not be feasible if the task on hand is time-sensitive. For instance, if the labeling has to be done for COVID-19 patients that require urgent care; 2). It is observed that existing neural network based models seems not consistently perform well across different datasets. For example, ACTINN [7] and scmapcell [4] show strong performance on certain datasets such as Baron Mouse and Baron Human [9] while they have weak performance on 68K dataset [10]. However, performing consistently well across different datasets is imperative to build real applications. This paper aims to implement a semi-supervised method that only uses few labeled samples together with huge amounts of unlabeled samples to reduce a mass efforts of data annotation. In addition, the proposed method is able to consistently produce promising performance across different datasets.

In this paper, *we propose a novel deep semi-supervised learning model when only very limited number of cells are labeled, and a large number of cells are unlabeled*. The proposed framework is shown in Figure 1. It is trained on cells with predefined cell types and then can be used to predict cell types on new datasets. The cells in scRNAseq data are transformed to "gene sentences" by taking advantage of similarities between natural language system and gene system. Furthermore, to overcome data sparsity, we employ word embedding techniques [11] to represent the genes in these sentences as gene vectors. Then, these vectors are inputs into the proposed semi-supervised neural networks built on recurrent convolutional neural networks (RCNN) [12]. It consists of three components, namely, a shared bidirectional Long Short-Term Memory Recurrent Neural Network (LSTM RNN), a supervised bidirectional LSTM RNN, and an unsupervised bidirectional LSTM RNN. One path is composed of the shared bidirectional LSTM RNN and supervised bidirectional LSTM RNN while the other path consists of the shared bidirectional LSTM RNN and unsupervised bidirectional LSTM RNN. All data (labeled and unlabeled data) will be evaluated to generate the mean squared error loss, while only labeled data will be evaluated to calculate the cross entropy loss. Experimental results of intra-dataset validation on macosko2015 [13] and 68K [14], and inter-dataset validation on MTG (Human) data [31] for training and ALM (Mouse) data [32] for testing demonstrate the effectiveness of the proposed model even when training it with a very limited amount of labeled cells.

The contributions in this study are as follows.

- We represent the cells in scRNAseq data via embedding techniques to reduce the sparsity of gene expression values, which is able to enhance the performance of cell type identification completed by neural networks based methods.
- We propose semi-supervised deep learning models with RCNN through jointly training supervised bidirectional LSTM RNN and unsupervised bidirectional LSTM RNN. It is shown that the proposed model can learn on unlabeled cells and labeled cells jointly to identify cell types with high performance. It can reduce the efforts of labeling data to build high-performance models for cell type identification.
- The proposed model is validated on two large-scale scRNAseq datasets: macosko2015 [13] and 68K [14]. Experimental results indicate that the new representations of cells enable cell type identification to accomplish with promising performance. Moreover, the proposed semi-supervised learning model is able to effectively identify the cell types by learning on a very limited number of labeled cells together with a large amount of unlabeled cells on various datasets for cell type identification. In addition, the experimental results on inter-dataset validation also demonstrate the effectiveness of the proposed methods.

II. Problem Formulation

Cell type identification on single-cell transcriptomic data is to classify the individual cells into predefined cell types, which is a supervised learning task from machine learning point of view. Specifically, it is a multi-class classification problem with N cell types in the set $C = \{c_1, c_2, c_3, ..., c_N\}$, where $N > 2$. Each cell belongs to one of the N different types. The goal is to construct a function which, given a new individual cell, will correctly predict the cell type where the new individual cell belongs. It is defined by

$$f(x; \theta) \rightarrow c \,, \tag{1}$$

where x is an individual cell, θ denotes the parameters in $f(\cdot)$, and $c \in C$. For the scRNA-seq data, x is composed of a sequence of gene expression values of the cell. Generally we will have more than $10,000$ gene expression values if we employ scRNAseq techniques to generate data [3], [7]. These gene expression values will be input as features to build machine learning models to complete cell type identification. Due to high dimensions and data sparsity of the scRNAseq data [15], it is challenging to solve this problem. Moreover, cell identification is similar to text classification regarding the similarities: (1) the inputs to these two tasks are sequencing samples; (2) these sequencing samples are classified into predefined classes.

III. Proposed Methodology

We propose a **Semi**-supervised **R**ecurrent convolutional neural **Net**work (SemiRNet) to address the challenge of lack-

ing labeled individual cells for cell type identification from scRNAseq data. The proposed model is based on RCNN [12] and the detailed architecture is shown in Figure 1. The first step is to preprocess the scRNAseq data to reduce the data sparsity [15], [16] by building "gene sentences" and representing the gene with word embedding techniques [11], [17]. Specifically, each cell in the scRNAseq data is composed of thousands of gene expression values. Unfortunately, most of these values are zeros because of the limitation of current single-cell sequencing techniques [16], which would reduce the performance of machine learning models significantly [18], [19]. Therefore, it is important to solve the data sparsity problem for cell type identification.

To overcome the data sparsity, we represented the gene sequences with "gene embedding" [20]. Although genomics data is not identical to text data, they share certain similarities: (1) They are all sequencing data. Genomics data is composed of gene sequences while text data consists of word sequences; (2) Gene sequences contain a subset of genes with high gene expression values from the gene database. Similarly, word sequences have a subset of words from the whole word dictionary; (3) The context of a gene can be defined by other genes that co-expressed with it [21]. Analogously, the word context can be determined by its concurring words.

With respect to these similarities, we build gene sentences by selecting k genes and employ word2vec [22] to represent these genes, where word2vec is a powerful technique to overcome data sparsity for natural language processing and understanding [22], [23], [24]. Word2vec [22] is to construct distributed representations of words that can represent the semantics of a word by mapping them to vectors in a high-dimension space, which is implemented by maximizing the probability of word co-occurrences in context, i.e., only a few words apart in a same sentence. Analogously, Gene2Vec [25], [26], [21], [27] is implemented by defining the context of a gene by the other genes that co-expressed with it, which is able to capture gene correlations in the more effective manners. In detail, it aims to derive an embedding such that the probability of the context of a gene is maximized. It has been successfully applied to many tasks such as biomarker discovery [26] and gene-gene interaction prediction [21]. In the proposed method, we select the top k genes in terms of their expression values to build the gene sentence defined by equation (2).

$$S(g) =< g_1, g_2, g_3, ..., g_t, ..., g_k > , \tag{2}$$

where g is the original gene sequence with n genes generated by the single cell sequencer for the cell and $t < k < n$. g_t is the gene selected with respect to the expression value. Then the genes in the gene sentence $S(g)$ are represented as gene embeddings. For instance, the gene sentence $<g_1, g_2, g_3, ..., g_t, ..., g_k>$ will be represented as a sequence of gene embedding $<e_1, e_2, e_3, ..., e_t, ..., e_k>$, where e_t is the embedding representation of the gene g_t. The gene embedding provides a way to use an efficient, dense representation in which similar genes have a similar encoding. An embedding is a dense vector of floating point values, where the length of the vector is a parameter set manually. Instead of specifying the values for the embedding manually, the gene embedding is trainable parameters learned by the model during training. A higher dimensional embedding can capture fine-grained relationships between genes.

After the preprocessing procedure, these gene sentences with gene embeddings will be input to the shared bidirectional LSTM RNN to extract common features for cell identification. The forward layer and backward layer generate two directional correlation features, respectively. Next, we combine these two groups of features with the gene embedding and obtain the output z of the shared RNN, where z is a sequence $<z_1, z_2, z_3, ..., z_t, ..., z_k>$ and z_t is given by

$$z_t = h_t^f \oplus e_t \oplus h_t^b , \tag{3}$$

where

$$h_t^f = a(w_h^f h_{t-1}^f + w_e^f e_t + b_h^f) , \tag{4}$$

$$h_t^b = a(w_h^b h_{t+1}^b + w_e^b e_t + b_h^b) , \tag{5}$$

z_t is the output of g_t of the gene sentence $<g_1, g_2, g_3, ..., g_t, ..., g_k>$. $\oplus$ is the concatenation operation. $a(\cdot)$ is the activation function for hidden layers. w_h^f and w_e^f are forward weights for two layers, namely, forward hidden layer and embedding hidden layer. w_h^b and w_e^b are backward weights for these two layers, respectively. b_h^f and b_h^b are bias for these two layers.

The idea to introduce this shared RNN to the proposed model is motivated by deep multi-task learning [28], [29], since different tasks share a common feature representation based on the original features. In addition, the reason for learning common feature representations instead of directly using the original ones is that the original representation may not have enough expressive power for multiple tasks. With the training data in all tasks, a more powerful representation can be learned for all the tasks and this representation will improve the performance. Therefore, the output z from the shared RNN are evaluated by two bidirectional RNNs, namely, supervised bidirectional LSTM RNN and unsupervised bidirectional LSTM RNN. As shown in Figure 1, the structures of these two RNNs are the same to that of shared RNN. For the supervised RNN, it is to learn the deep features of cells when the sample has the label. The output z' of supervised RNN is the sequence $<z'_1, z'_2, z'_3, ..., z'_t, ..., z'_k>$, where z'_t is given by

$$z'_t = max(tanh(w_{sup} z^{tmp'} + b_{sup})) , \tag{6}$$

where

$$z^{tmp'} = h_{t'}^f \oplus z_t \oplus h_{t'}^b , \tag{7}$$

$$h_{t'}^f = a(w_{h'}^f h_{t'-1}^f + w_{sup}^f z_t + b_{h'}^f) , \tag{8}$$

$$h_{t'}^b = a(w_{h'}^b h_{t'+1}^b + w_{sup}^b z_t + b_{h'}^b) , \tag{9}$$

We employ the same activation function $a(\cdot)$ for the hidden layers of the supervised bidirectional RNN. $tanh(\cdot)$ is the activation function for the dense layer. w_{sup} and b_{sup} are the weights and a bias between the max-pooling layer and the dense layer in the supervised RNN. $w_{h'}^f$ and w_{sup}^f are forward weights for the forward layer and embedding layer in

the supervised bidirectional RNN. $w^b_{h'}$ and w^b_{sup} are backward weights for these two layers, respectively. $b^f_{h'}$ and $b^b_{h'}$ are bias for these two layers, respectively.

Moreover, we build the unsupervised bidirectional RNN to generate another representation of the input and the output z'' is a vector $<z''_1, z''_2, z''_3, ..., z''_t, ..., z''_k>$, where z''_t is given by

$$z''_t = max(tanh(w_{unsup} z^{tmp''} + b_{unsup})) \ , \tag{10}$$

where

$$z^{tmp''} = h^f_{t''} \oplus z_t \oplus h^b_{t''} \ , \tag{11}$$

$$h^f_{t''} = a(w^f_{h''} h^f_{t''-1} + w^f_{unsup} z_t + b^f_{h''}) \ , \tag{12}$$

$$h^b_{t''} = a(w^b_{h''} h^b_{t''+1} + w^b_{unsup} z_t + b^b_{h''}) \ , \tag{13}$$

w_{unsup} and b_{unsup} are the weights and a bias between the max-pooling layer and the dense layer in the unsupervised RNN. $w^f_{h''}$ and w^f_{unsup} are forward weights for two layers, namely, forward layer and embedding layer in the unsupervised bidirectional RNN. $w^b_{h''}$ and w^b_{unsup} are backward weights for these two layers, respectively. $b^f_{h''}$ and $b^b_{h''}$ are bias for these two layers, respectively.

We utilize those two vectors z' and z'' to calculate the cross entropy loss (CEL) and mean squared error loss (MSEL) for supervised and unsupervised paths, respectively. They are given by

$$l^{CEL} = -\sum y \times log\phi(z') \ , \tag{14}$$

$$l^{MSEL} = ||z' - z''||^2 \ , \tag{15}$$

where y is the label for the input and $\phi(\cdot)$ is the softmax activation function. l^{CEL} is the standard cross entropy loss to account for the loss of labeled inputs. Because training RNNs with dropout regularization and gradient-based optimization is a stochastic process, the two RNNs will have different link weights after training. In other words, there will be differences between the two prediction vectors z' and z'' that are from these two RNNs (supervised RNN and unsupervised RNN). These differences can be treated as an error and thus minimizing its mean square error (MSE) is another objective l^{MSEL}, in the proposed model. Furthermore, to combine the supervised loss l^{CEL} and unsupervised loss l^{MSEL}, we scale the latter by time-dependent weighting function $w(t)$ [30] that ramps up, starting from zero, along a Gaussian curve. The total loss is defined by

$$Loss = l^{CEL} + w(t) \times l^{MSEL} \ , \tag{16}$$

At the beginning of training, the total loss and the learning gradients are dominated by the supervised loss component, i.e., the labeled data only. At later stage of training, unlabeled data will contribute more than the labeled data. The detailed learning of the proposed model is shown in Algorithm 1, which demonstrated the training of the proposed model. $f_r(\cdot)$ is to represent cells as gene sentences, $f_e(\cdot)$ is to learn gene embeddings on the gene sentences, and $f_{\theta_{shared}}(\cdot)$ is to learn the common features from the gene embeddings. Parameters of the shared neural network θ_{shared} include w^f_h, w^f_b, w^f_e, w^b_e, b^f_h, and b^b_h.

Algorithm 1 Learning of SemiRNet

Require: training sample x_i, the set of training samples S, labeled samples y_i for x_i ($i \in S$)

for t in [1, num epochs] **do**
  for each minibatch B **do**
    $x'_{i\in B} \leftarrow f_r(x_{i\in B})$ ▷ preprocessing
    $x''_{i\in B} \leftarrow f_e(x'_{i\in B})$ ▷ gene embedding
    $z_{i\in B} \leftarrow f_{\theta_{shared}}(x''_{i\in B})$ ▷ common feature extraction
    $z'_{i\in B} \leftarrow f_{\theta_{sup}}(z_{i\in B})$ ▷ supervised representation
    $z''_{i\in B} \leftarrow f_{\theta_{unsup}}(z_{i\in B})$ ▷ unsupervised representation
    $l^{CEL}_{i\in B} \leftarrow -\frac{1}{|B|}\sum_{i\in B\cap S} log\phi(z'_i)[y_i]$ ▷ supervised loss component
    $l^{MSEL}_{i\in B} \leftarrow \frac{1}{C|B|}\sum_{i\in B} ||z'_i - z''_i||^2$ ▷ unsupervised loss component
    $Loss \leftarrow l^{CEL}_{i\in B} + w(t) \times l^{MSEL}_{i\in B}$ ▷ total loss
    update θ_{shared}, θ_{sup}, θ_{unsup} using the optimizer, e.g., ADAM
return θ_{shared}, θ_{sup}, θ_{unsup}

After extracting common features from gene samples, we use $f_{\theta_{sup}}(\cdot)$ and $f_{\theta_{unsup}}(\cdot)$ to obtain higher level representations of cells to complete cell type identification and enhance the cell representations through optimizing supervised loss and unsupervised loss jointly. Parameters of the supervised RNN θ_{sup} include $w^f_{h'}$, $w^b_{h'}$, w^f_{sup}, w^b_{sup}, $b^f_{h'}$, $b^b_{h'}$, w^{sup}, and b^{sup}. Parameters of the unsupervised RNN θ_{unsup} consist of $w^f_{h''}$, $w^b_{h''}$, w^f_{unsup}, w^b_{unsup}, $b^f_{h''}$, $b^b_{h''}$, w^{unsup}, and b^{unsup}.

During the testing stage, only the supervised path including shared bidirectional RNN and supervised bidirectional RNN is used for cell type identification, which involves parameters of the supervised part θ_{sup} including $w^f_{h'}$, $w^b_{h'}$, w^f_{sup}, w^b_{sup}, $b^f_{h'}$, $b^b_{h'}$, w^{sup}, and b^{sup}, and those of shared part including w^f_h, w^f_e, w^b_h, w^b_e, b^f_h, and b^b_h. In addition, the gene sequences will be represented as gene sentences with gene embeddings for inputs to the proposed model.

The proposed model combines the advantages of deep multi-task learning [28] and Π model [30]. However, there exist significant differences. Compared to deep multi-task learning, the subtasks in the proposed model have two categories of learning, namely, supervised learning and unsupervised learning while there is only supervised learning in the deep multi-task learning. On the other hand, instead of using one path neural networks, we apply two independent RNNs to generate supervised and unsupervised outputs. Furthermore, the proposed model is more flexible as the two independent RNNs can be tuned in terms of specific goals.

IV. Experiment

A. Dataset

We employed two large single-cell sequencing datasets: macosko2015 [13] and 68K [14] to validate the proposed methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2022.3173587, IEEE/ACM Transactions on Computational Biology and Bioinformatics

1) Macosko2015: Macosko2015 [13] is a retina scRNAseq dataset including 44,825 mouse retinal cells with 39 transcriptionally distinct cell populations[1]. The dataset with 24,760 genes contains 12 cell types, namely, rods, cones, muller glia, astrocytes, fibroblasts, vascular endothelium, pericytes, microglia, retinal ganglion, bipolar, horizontal, and amacrine. The cell type distribution is shown in Table I. It can be observed that the cell distribution is imbalanced across different cell types. Therefore, machine learning models built on this data will have bias to majority classes. In other words, the models will tend to obtain high performance for identification of majority cell types, but low performance for identification of minority cell types. It will be a challenge to implement cell type classification with high performance for all cell types. In addition, we present the number of cells for different ratios of labeled data to train the proposed semi-supervised approach as Table II, where the total number of training samples is $31,386$.

TABLE I
CELL DISTRIBUTION IN TWELVE TYPES, NAMELY, RODS, CONES, MULLER GLIA (MG), ASTROCYTES, FIBROBLASTS, VASCULAR ENDOTHELIUM (VE), PERICYTES, MICROGLIA, RETINAL GANGLION (RG), BIPOLAR, HORIZONTAL, AND AMACRINE.

Cell Type	**Rods**	**Cones**	**MG**	**Astrocytes**
Cell Number	29,397	1,871	1,622	54
Cell Type	**Fibroblasts**	**VE**	**Pericytes**	**Microglia**
Cell Number	85	253	63	67
Cell Type	**RG**	**Bipolar**	**Horizontal**	**Amacrine**
Cell Number	434	6,297	252	4,430

TABLE II
NUMBER OF CELLS FOR DIFFERENT RATIOS OF LABELED DATA IN MACOSKO2015 TRAINING DATASETS.

Labeled Ratio	**Labeled Cells**	**Unlabeled Cells**
1%	314	31,072
3%	942	30,444
5%	1,570	29,816
10%	3,139	28,247
30%	9,416	21,970

2) 68K: To our knowledge, 68K [14] is the largest scRNA-seq datasets generated by profiling $68,000$ fresh peripheral blood mononuclear cells (PBMCs) that are related to immune populations. It contains 11 sub-types of immune cells including CD8+Cytotoxic T, CD8+/CD45RA+Naive, CD56+NK, CD4+/CD25T Reg, CD19+B, CD4/CD45RO+Memory, Dendritic, CD14+Monocyte, CD4+CD5RA, CD34+, and CD4+T Helper 2. There are $65,943$ individual cells with $20,387$ genes through data preprocessing. Detailed cell distribution shown in Table III below presents class imbalance that is the same challenge to Macosko2015, which might lead to model bias to the majority classes such as CD8+Cytotoxic T and CD8+/CD45RA+Naive. In addition, this dataset contains 11 immune cell populations which are harder to differentiate, particularly the T cell compartment (6 out of 11 cell populations) [10]. Moreover, the number of cells for different ratios of labeled data is shown in Table IV, where the total number of training samples is $42,204$.

TABLE III
CELL DISTRIBUTION IN ELEVEN TYPES, NAMELY, CD8+CYTOTOXIC T, CD8+/CD45RA+NAIVE, CD56+NK, CD4+/CD25T REG, CD19+B, CD4/CD45RO+MEMORY, DENDRITIC, CD14+MONOCYTE, CD4+CD5RA, CD34+, AND CD4+T HELPER 2.

Cell Type	**CD8+Cytotoxic T**	**CD8+/CD45RA+Naive**
Cell Number	20,307	16,361
Cell Type	**CD56+NK**	**CD4+/CD25T Reg**
Cell Number	8,522	6,116
Cell Type	**CD19+B**	**CD45RO+Memory**
Cell Number	5,579	3,031
Cell Type	**Dendritic**	**CD14+Monocyte**
Cell Number	1,946	1,944
Cell Type	**CD4+CD5RA**	**CD34+**
Cell Number	1,857	188
Cell Type	**CD4+T Helper 2**	**Total**
Cell Number	92	65,943

TABLE IV
NUMBER OF CELLS FOR DIFFERENT RATIOS OF LABELED DATA IN 68K TRAINING DATASETS.

Labeled Ratio	**Labeled Cells**	**Unlabeled Cells**
1%	423	41,781
3%	1,267	40,937
5%	2,111	40,093
10%	4,221	37,983
30%	12,662	29,542

3) ALM and MTG: The inter-dataset validation employed MTG (Human) data [31] for training and ALM (Mouse) data [32] for testing, which is a case from the comprehensive comparison of automatic cell identification methods for single-cell RNA sequencing data [10]. MTG (Human) data and ALM (Mouse) data are from different anatomy structures of brain, namely, middle temporal gyrus, and anterior lateral motor area. MTG (Human) data consists of 14,636 cells while ALM (Mouse) data contains 8,758 cells.

B. Experimental settings

In this experiment, our proposed model is employed to implement cell type identification. The key hyper parameters of the proposed model are: Embedding size: 256, Minibatch size: 128, Number of epoch: 300, Optimizer: Adam optimizer, Learning rate: 0.001, Learning rate decay: 0.9. They are determined by trial and error. To obtain the optimal value of k for building gene sentences, we tried a set of values such as 50, 100, 150, and 200 for Macosko2015 dataset, and the experimental results with 50 genes demonstrated optimal performance. For 68K dataset, we employ Chi-Square Test[2] to select $10,000$ genes and then use the genes whose gene expression values are not zeros to build gene sentences. Therefore, the k values will be various to different samples regarding different numbers of genes with zero values of gene expression. Moreover, the details of the model architecture is

[1]https://github.com/olgabot/macosko2015

[2]https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html

illustrated in Table V. Specifically, the output of the proposed model contains two parts: cell type $\phi(z')$ and a new representation z''.

TABLE V
THE PROPOSED NETWORK ARCHITECTURE.

Name	Description
Input	Gene Sentence
Gene Embedding	Mikolov model [22], [33]
Shared RNN	256 LSTM cells for each hidden layer, one forward hidden layer, one backward hidden layer
Supervised RNN	256 LSTM cells for each hidden layer, one forward hidden layer, one backward hidden layer, one dense layer with 256 neurons, one 2×2 max-pooling layer
Unsupervised RNN	256 LSTM cells for each hidden layer, one forward hidden layer, one backward hidden layer, one dense layer with 256 neurons, one 2×2 max-pooling layer
Output	cell type $\phi(z')$ and a new representation z''

C. Evaluation metric

We applied different evaluation metrics to evaluate the performance of our proposed model, which includes accuracy, macro-average Precision (MacroP), macro-average Recall (MacroR), and macro-average Fscore (MacroF) [34]. Accuracy is calculated by dividing the number of cells identified correctly over the total number of testing cells.

$$Accuracy = \frac{N_{correct}}{N_{total}}. \tag{17}$$

Macro-average [35] is to calculate the metrics such as Precision, Recall and F-scores independently for each cell type and then utilize the average of these metrics. It is to evaluate the whole performance of classifying cell types.

$$MacroF = \frac{1}{C}\sum_{c=1}^{C} Fscore_c. \tag{18}$$

$$MacroP = \frac{1}{C}\sum_{c=1}^{C} Precision_c. \tag{19}$$

$$MacroR = \frac{1}{C}\sum_{c=1}^{C} Recall_c. \tag{20}$$

where C denotes the total number of cell types and $Fscore_c$, $Precision_c$, $Recall_c$ are $Fscore$, $Precision$, $Recall$ values in the c^{th} cell type which are defined by

$$Fscore = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{21}$$

where $Precision$ indicates precision measurement that defines the capability of a model to represent only correct cell types and $Recall$ computes the aptness to refer all corresponding correct cell types:

$$Precision = \frac{TP}{TP + FP}. \tag{22}$$

$$Recall = \frac{TP}{TP + FN}. \tag{23}$$

whereas TP (True Positive) counts total number of cells matched with the cells in the types. FP (False Positive) measures the number of recognized type does not match the annotated cells. FN (False Negative) counts the number of cells that does not match the predicted cells. The main goal for learning from imbalanced datasets such as macosko2015 [13] is to improve the recall without hurting the precision. However, recall and precision goals are often conflicting, since when increasing the true positive (TP) for the minority class (True), the number of false positives (FP) can also be increased; this will reduce the precision [36].

In addition, we employ three deep supervised learning models as baselines including 1) Word-level CNN (Word CNN) [37], 2) Attention-Based Bidirectional RNN (Att RNN) [38], and 3) Recurrent CNN (RCNN) [12], where these models perform well on similar problems such as text classification. For example, Word CNN performs well on sentence classification, which is more suitable to process sequencing data as the length of the content of the data is short like that of the gene sentence. In addition, we build 4) word-level bidirectional RNN (Word RNN) to compare the implemented model, where Word RNN contains one embedding layer and one bidirectional RNN layer, and concatenate all the outputs from the RNN layer to feed to the final layer that is a fully-connected layer. Moreover, we employ 6 traditional machine learning models as the baselines, namely, Naive Bayes, Decision Tree, Random Forest, Adaboost, Neural Networks (NN), and Support Vector Machine (SVM). *Specifically, NN is a shallow neural network that is similar to ACTINN [7].* Thus, there are total 10 baseline models. *Note that baseline models are built on all labeled cells from the original training datasets*.

D. Experimental results

1) Macosko2015: We evaluated the proposed model from two perspectives. One is to verify if the data preprocessing of the cell is able to be employed to identify cell types effectively. The other is to validate performance of the proposed model on cell type identification with limited amount of labeled cells.

a. Data preprocessing: Table VI presents the comparison of identification performance between traditional machine learning (ML) models and deep learning (DL) models, where the ML models are built on the original gene values without data preprocessing while the DL models are built on preprocessed data that includes gene sentences with gene embeddings.

We can observe that most of ML models perform not well on the cell identification due to the data sparsity. For example, Naive Bayes's accuracy and MacroF are not high since it is sensitive to data sparsity and cell imbalance. Other four ML including Decision Tree, Random Forest, Adaboost and NN identify cell type with high accuracy but low MacroF since they cannot overcome the challenge of cell imbalance even if data sparsity will not affect their performance significantly. Only SVM can perform well on accuracy and MacroF. However, it will cost almost one and a half hours to obtain

TABLE VI
COMPARING PERFORMANCE BETWEEN TRADITIONAL MACHINE LEARNING (ML) AND DEEP LEARNING (DL) ON MACOSKO2015 DATASET.

	Machine Learning (ML)	Accuracy	MacroP	MacroR	MacroF	Training Time (s)
Original Gene Expression	Naive Bayes	35.06%	36.96%	30.40%	35.48%	11
	Random Forest	85.09%	55.44%	27.45%	31.03%	22
	Neural Networks	86.72%	19.47%	23.77%	21.23%	187
	Decision Tree	93.78%	86.60%	80.34%	82.69%	1,172
	Adaboost	74.07%	30.38%	26.88%	25.67%	1,767
	Support Vector Machine (SVM)	97.28%	98.24%	93.32%	95.50%	5,554
	Supervised Deep Learning (SDL)	**Accuracy**	**MacroP**	**MacroR**	**MacroF**	**Training Time (s)**
Gene Embedding	Word CNN [37]	96.30%	90.79%	77.22%	81.90%	295
	Word RNN	96.11%	86.69%	82.82%	84.17%	8,368
	Attenion RNN [38]	95.79%	88.18%	84.85%	85.85%	4,661
	RCNN [12]	**96.56%**	**96.55%**	**92.70%**	**94.45%**	**2,383**

a converged model with respect to training on such a big scRNAseq data.

On the contrary, different DL models built on preprocessed cell data can identify cell types with promising and consistent performance. For instance, compared to ML models, all DL models are able to gain high accuracy above 95%, which means they are not struggling to the data sparsity. Moreover, considering MacroF values, DL models can obtain encouraging performance since these models can overcome cell imbalance to some extent. Specifically, the performance difference between RCNN and SVM is not significant regarding accuracy and MacroF. Moreover, compared to SVM, building RCNN only uses about a half of hour to become converged. Based on the observations, we believe that deep learning methods can outperform traditional machine learning models by learning on the preprocessed data generated by gene embedding.

b. Cell type identification: In this section, we will examine if the proposed model is able to effectively identify the cell types by training on very limited amount of annotated cells. Table VII presents the comparison of identification performance between supervised deep learning (SDL) and the proposed model, where the proposed model is built based on RCNN with different ratios of training labeled cells. Firstly, we observe that the performance of proposed model is enhanced through increasing the ratios of annotated cells. In other words, the proposed model is able to obtain stronger identification ability when learning on more labeled data. It is because the unsupervised path is able to enhance the data representation for improving cell identification that is implemented with supervised path.

TABLE VII
COMPARING PERFORMANCE BETWEEN SUPERVISED DEEP LEARNING (SDL), AND OUR MODEL (SEMI-SUPERVISED RECURRENT CONVOLUTIONAL NEURAL NETWORKS, SEMIRNET) ON MACOSKO2015 DATASET.

SDL	Accuracy	MacroP	MacroR	MacroF
RCNN [12]	96.56%	96.55%	92.70%	94.45%
Our model	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
SemiRNet (1%)	95.47%	91.73%	93.90%	92.64%
SemiRNet (3%)	95.76%	92.62%	94.21%	93.28%
SemiRNet (5%)	95.76%	93.12%	93.39%	93.18%
SemiRNet (10%)	95.70%	94.92%	93.18%	93.87%
SemiRNet (30%)	96.44%	96.53%	92.66%	94.46%

Compared to supervised deep learning (SDL), the proposed model can identify the cell types even with extremely small amount of annotated cells. For example, we can obtain encouraging performance with 1% annotated cells. Furthermore, the proposed model is robust since we can gain similar performance with different ratios of annotated cells. For instance, the differences of accuracy and MacroF between the case of 1%, 5%, and 30% are about 1%. Specifically, the MacroP is improved significantly when increasing the ratios of labeled cells for training while the MacroR is stable. The reason for this observation is that enhancing representation with unsupervised learning in the proposed model seems to be more useful to identify cell type precisely.

To further investigate the detailed performance of Table VII, we show the performance with confusion matrix. Fig. 2 presents the confusion matrix on performance generated with different ratios of annotated cells. It is observed that for different cell types, the accuracy is increased when involving more labeled cells to build the model. Specifically, when we use different ratios of labeled cells to build the model, the error distributions are not changed significantly. For instance, for the cell type c_2, the majority errors are from incorrectly classifying the cells into the cell type c_7.

Furthermore, considering the unbalanced feature of cell distribution (See Table I), the results in Fig. 2 presents the model bias for the majority cell types. It means that the model will obtain higher performance for the majority type, but lower performance for the minority types. For the cell type c_{12}, compared to the case of 1% labeled cells, the accuracy is decreased because of the model bias when using 10% labeled cells for training.

On the other hand, although the overall prediction accuracy (See Table VII) is increased when increasing the ratios of labeled cells, it is not always true that the accuracy for each cell type will be enhanced. This can be observed in Fig. 2. Take the cell type c_{12} as an example, the prediction accuracy is not always increased when increasing the ratios of labeled cells. On the contrary, for the cell type c_{11}, the accuracy is improved whenever more labeled cells are involved for building the identification model.

In addition to examining the performance comparisons between the proposed models and baselines, we have to figure out whether the proposed model is sensitive to the hyper-parameters. There are various hyper-parameters involved in the

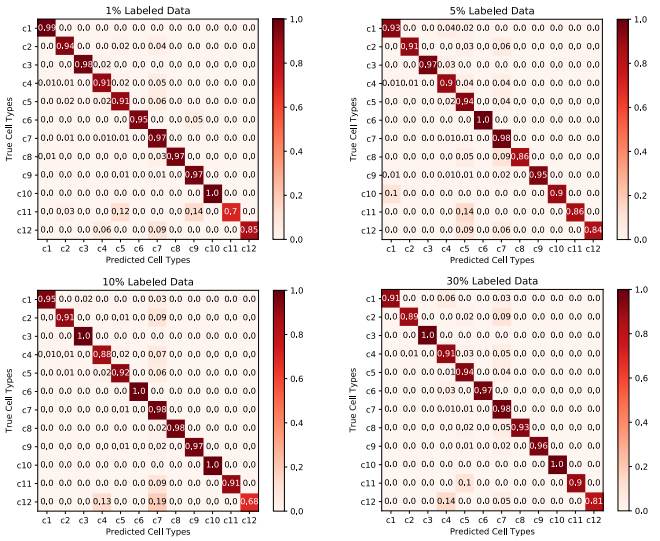


Fig. 2. Confusion matrix on different cell types generated with batch size 128 on Macosko2015 dataset. There are 12 cell types including c_1 (Bipolar), c_2 (Pericytes), c_3 (Vascular endothelium), c_4 (Retinal ganglion), c_5 (Horizontal), c_6 (Rods), c_7 (Cones), c_8 (Amacrine), c_9 (Fibroblasts), c_{10} (Microglia), c_{11} (Astrocytes), c_{12} (Muller glia)

learning procedure of the proposed model. Here, we choose batch size to check since different batch sizes will involve different numbers of labeled cells for building the proposed model when using the same ratio of labeled cells. Table VIII shows the comparison results for two different batch sizes. We observe that there is no significant differences of the performance. It means that the proposed model is not sensitive to the batch size since the supervised and unsupervised RNN in the proposed model could collaborate with each other to overcome the effects from the difference of batch size.

TABLE VIII
COMPARING PERFORMANCE WITH DIFFERENT BATCH SIZES ON DIFFERENT RATIOS OF LABELED CELLS ON MACOSKO2015 DATASET.

	1% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	95.47%	91.73%	93.90%	92.64%
256	95.11%	89.99%	94.40%	91.88%
	3% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	95.76%	92.62%	94.21%	93.28%
256	95.44%	91.76%	94.21%	92.79%
	5% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	95.76%	93.12%	93.39%	93.18%
256	95.49%	91.34%	93.74%	92.31%
	10% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	95.70%	94.92%	93.18%	93.87%
256	95.93%	95.13%	93.11%	94.00%
	30% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	96.44%	96.53%	92.66%	94.46%
256	96.45%	96.58%	92.02%	94.10%

Moreover, compare to Table VIII, Fig. 3 presented the accuracy of each cell to check the effects with different hyper-parameters in detail. To sum up, for the majority cell type c_6, the performance is enhanced for the case of larger batch size. For the minority cell types, when employing larger batch size to build the model, the performances for some cell types such as c_1 and c_2 are decreased whereas for the cell types like c_9 and c_{11}, the accuracy is increased. It means that we have to choose the optimal batch size for improving the performance of certain minority cell types.

On the other hand, compared to the case with more labeled data, the case with low ratios of labeled cells needs larger batch size to improve the performance for the majority cell type such as c_6. For instance, when we compare the confusion matrix for the case of 1% labeled cells, the confusion matrix with batch size 256 has better performance compare to that of batch size 128. It is consistent to the intuition that with larger batch size, we will obtain larger size of labeled samples to enhance the performance of supervised path when using extremely low ratio of labeled cells. In other words, to improve the performance for the proposed model in the case of extremely low ratios of labeled data, we should apply larger batch size for the case of majority cell type.

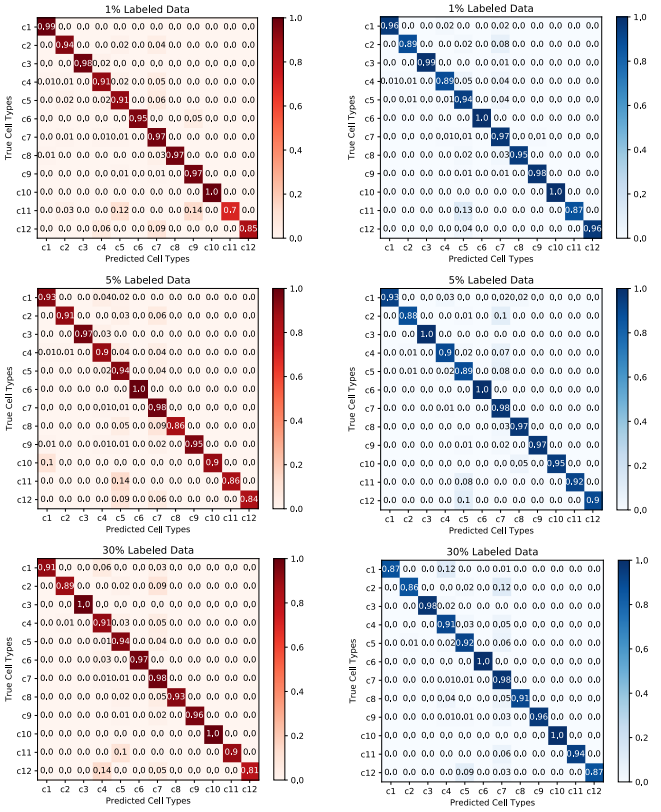


Fig. 3. Comparison of confusion matrix on different cell types generated with batch size 128 and 256. The left column is for the case of 128 while the right column is for the case of 256.

2) 68K: To further validate the proposed methods, we validate its effectiveness on 68K dataset, a larger dataset.

a. Data preprocessing: Compared to the case of Macosko2015, in addition to the traditional machine learning methods, we employ state-of-the-art methods including scmap-cell [4], SingleR [6], singleCellNet [39], and ACTINN [40] to compare the proposed methods, where the Median F1-score of these models are from Abdelaal *et al.*'s work [10].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2022.3173587, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE IX
PERFORMANCE COMPARISON BETWEEN STATE-OF-THE-ART METHODS, TRADITIONAL MACHINE LEARNING MODELS, AND DEEP LEARNING MODELS ON 68K DATASET.

State-of-the-art	**scmapcell [4]**	**SingleR [6]**	**singleCellNet [39]**	**ACTINN [40]**
Median F1-score	64%	32%	**74%**	**74%**
Machine Learning	**SVM**	**Naive Bayes**	**Random Forest**	**Adaboost**
Median F1-score	**69.75%**	18.18%	56.27%	48.97%
Supervised Deep Learning (SDL)	**Word CNN [37]**	**Word RNN**	**ATT RNN [38]**	**RCNN [12]**
Median F1-score	71.62%	73.15%	69.53%	**78.84%**

TABLE X
COMPARING PERFORMANCE BETWEEN TRADITIONAL MACHINE LEARNING (ML) AND DEEP LEARNING (DL) ON 68K DATASET.

	Machine Learning (ML)	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
Original Gene Expression	Naive Bayes	24.36%	31.54%	33.70%	25.92%
	Random Forest	61.84%	49.98%	37.38%	38.42%
	Adaboost	54.35%	46.00%	41.31%	38.26%
	Support Vector Machine (10%)	61.88%	55.58%	50.16%	55.06%
	Support Vector Machine (30%)	63.31%	55.18%	53.42%	54.15%
	Support Vector Machine (100%)	70.84%	56.09%	60.62%	56.83%
	Supervised Deep Learning (SDL)	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
Gene Embedding	Word CNN [37]	76.06%	69.55%	57.75%	60.76%
	Word RNN	70.24%	60.82%	55.08%	57.15%
	Attenion RNN [38]	75.56%	66.18%	58.26%	60.72%
	RCNN [12] (10%)	**68.16%**	**59.60%**	**52.78%**	**54.89%**
	RCNN [12] (30%)	**69.84%**	**61.29%**	**57.05%**	**58.58%**
	RCNN [12] (100%)	**76.62%**	**68.31%**	**62.94%**	**64.67%**

TABLE XI
COMPARING PERFORMANCE BETWEEN SUPERVISED DEEP LEARNING (SDL), AND OUR MODEL (SEMI-SUPERVISED RECURRENT CONVOLUTIONAL NEURAL NETWORKS, SEMIRNET) ON 68K DATASET.

SDL	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
RCNN [12] (10%)	68.16%	59.60%	52.78%	54.89%
RCNN [12] (30%)	69.84%	61.29%	57.05%	58.58%
RCNN [12] (100%)	76.62%	68.31%	62.94%	64.67%
Our model	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
SemiRNet (1%)	61.26 $\pm$ 5.38%	52.38 $\pm$ 2.36%	57.14 $\pm$ 1.74%	52.67 $\pm$ 1.98%
SemiRNet (3%)	64.93 $\pm$ 3.24%	55.65 $\pm$ 2.79%	58.48 $\pm$ 0.87%	55.48 $\pm$ 0.27%
SemiRNet (5%)	62.16 $\pm$ 3.70%	53.77 $\pm$ 3.59%	57.74 $\pm$ 1.87%	53.96 $\pm$ 2.11%
SemiRNet (10%)	68.18 $\pm$ 3.13%	59.14 $\pm$ 3.55%	57.73 $\pm$ 1.38%	57.20 $\pm$ 1.48%
SemiRNet (30%)	71.49 $\pm$ 4.39%	63.61 $\pm$ 2.50%	58.44 $\pm$ 1.93%	59.93 $\pm$ 1.76%

Table IX showed the performance comparison in detailed. It can be observed that the proposed method via deep learning outperforms other methods. Specifically, RCNN presented the optimal performance, which is consistent to the case of Macosko2015. In other words, combining data preprocessing like building gene sentences and learning with bidirectional RNN is able to enhance the performance significantly. For traditional machine learning methods, SVM performed better than other machine learning models.

On the other side, Table X presents the comparison of identification performance between traditional machine learning (ML) models and deep learning (DL) models with comprehensive evaluation metrics. We can obtain the similar observation that generally deep learning based methods perform better than machine learning based methods, especially regarding the MacroF scores. Only SVM showed competitive performance on accuracy. On the contrary, different DL models present higher and consistent performance. For example, all DL models are able to gain higher accuracies above 70%. Furthermore, DL models can obtain MacroF values. Based on the observations, we believe that the preprocessing is an effective step to prepare the data for deep learning based cell type identification. In addition, we present the performance gained by training SVM and RCNN on fewer labeled training samples with two cases: 10% labeled samples and 30% labeled samples. In terms of performance comparison, fewer labeled samples led to lower performance regarding accuracy and MacroF scores. In other words, it needs more labeled data for building models to obtain higher identification performance.

b. Cell Type Identification: in this section, we will examine if the proposed model is able to effectively identify the cell types by training on very limited amount of annotated cells. Table XI demonstrates the comparison of identification performance between supervised deep learning (SDL) and the proposed model, where the proposed model is built based on RCNN with different ratios of training labeled cells. Moreover, we applied 5-fold cross validation to validate the performance and present the performance with 95% confidence intervals. Obviously, the performance of proposed model is enhanced through increasing the ratios of annotated cells, which is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2022.3173587, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE XII
COMPARING PERFORMANCE WITH DIFFERENT BATCH SIZES ON DIFFERENT RATIOS OF LABELED CELLS ON 68K DATASET.

	1% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	60.18%	52.07%	57.79%	52.86%
256	59.98%	49.48%	54.64%	49.98%
512	50.93%	44.08%	52.61%	44.80%
	3% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	64.20%	54.26%	60.91%	55.79%
256	58.77%	58.82%	56.34%	51.96%
512	66.92%	56.32%	53.30%	53.53%
	5% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	53.52%	54.68%	56.73%	54.66%
256	66.10%	55.33%	52.83%	52.05%
512	70.13%	59.40%	57.50%	56.34%
	10% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	63.73%	58.39%	56.29%	54.28%
256	69.37%	57.89%	55.96%	56.36%
512	70.22%	64.44%	57.04%	58.85%
	30% Labeled Data			
Batch size	**Accuracy**	**MacroP**	**MacroR**	**MacroF**
128	71.57%	64.54%	60.63%	62.10%
256	72.99%	65.73%	56.98%	59.95%
512	70.79%	64.44%	57.04%	58.85%

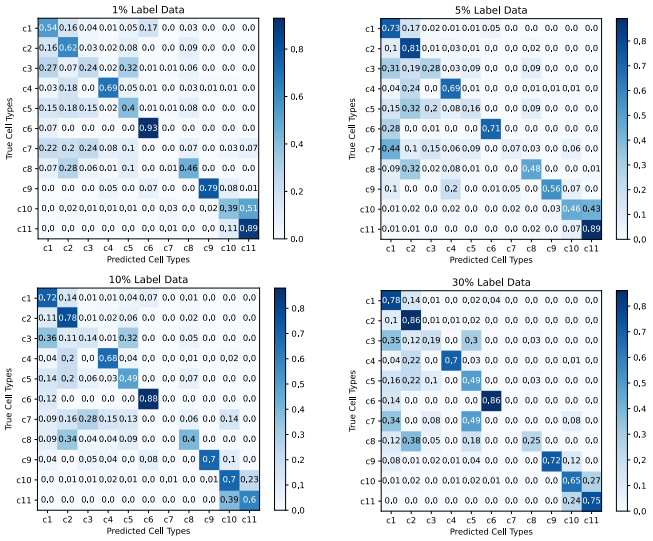


Fig. 4. Confusion matrix on different cell types generated with batch size 256 on 68K dataset. There are 11 cell types including c_1 (CD8+Cytotoxic T), c_2 (CD8+/CD45RA+Naive), c_3 (CD4/CD45RO+Memory), c_4 (CD19+B), c_5 (CD4+/CD25T Reg), c_6 (CD56+NK), c_7 (CD4+T Helper 2), c_8 (CD4+CD5RA), c_9 (CD34+), c_{10} (Dendritic), and c_{11} (CD14+Monocyte). The majority types include c_1, c_2, and c_6 while the minority types contains c_7, c_8, and c_9.

consistent to the case of Macosko2015. In other words, the proposed model is able to obtain stronger identification ability when learning on more labeled data together with unlabeled data. Specifically, regarding the confidence intervals, compared to Accuracy and MacroP, the confidence intervals of MacroF and MacroR are smaller, which means that they would be more suitable to evaluate the performance for this case since smaller confidence intervals indicates less uncertainties. In addition, compared to the performance shown in Table X, the performance is improved for the case of 10% and 30% with the proposed method. It is proved that unsupervised path is able to enhance the performance significantly.

In addition, compared to supervised deep learning (SDL) training on 100% labeled data, the proposed model can identify the cell types even with extremely small amount of annotated cells. For example, we can obtain encouraging performance with 10% annotated cells. Specifically, Accuracy and MacroF are improved significantly when increasing the ratios of labeled cells for training. It is because enhancing representation with unsupervised learning in the proposed model seems to be more helpful to recognize cell types.

Moreover, regarding the sample unbalance (See Table III), Fig. 4 showed the model bias to the majority cell types as the model obtained higher performance for the majority types like c_1, c_2, and c_6, but lower performance for the minority types such as c_7, c_8 and c_9, which is consistent to the observations on results of Macosko2015 dataset. Furthermore, regarding the cell type c_9, compared to the case of 1% labeled data, the model bias led to lower performance for the case of 10% labeled data. On the other hand, although the accuracy (See Table XI) is improved when increasing the ratios of labeled data for training, it doesn't mean that the accuracy for each cell type has been enhanced. When examining the accuracy in Fig 4, the accuracy of the cell c_8 and c_9 is not always increased when using more labeled data for training.

In summary, since the datasets employed for validation including 68K and Macosko2015 are not balanced, more labeled samples for training might result in model bias during prediction, which decreases the performance for supervised learning. On the contrary, semi-supervised method will use less labeled samples for training, which would reduce model bias to some extent. Therefore, the proposed semi-supervised method performance a little bit better comparing to the supervised methods for some cases.

We also examined if the proposed model is sensitive to the hyper-parameters through comparing the performance on three different batch sizes. Table XII shows the comparison results for three different batch sizes. We observe that smaller batch sizes like 128 will lead to higher MacroF scores for lower ratios of labeled data such as 1% and 3%. For larger ratios such as 5% and 10%, we should take larger batch sizes like 256 for better performance like MacroF scores. Furthermore, the confusion matrix in Fig. 5 allows us to check the detailed performance for Table XII. In summary, the proposed model performed better on the majority cell types such as c_2 and c_6 for cases of larger ratios of labeled data (30%) for different batch sizes. For minority cell type such as c_7, we should employ larger batch size and larger ratio of labeled data to obtain higher performance in general.

3) Inter-dataset validation: We selected top k genes to build gene sentences and applied word2vec to generate gene embeddings. In terms of results shown in Table XIII, RCNN model's performance is not very good, which is consistent with observations in the literature such as ACTINN [40]. The reason to this observation is that supervised deep learning models can learn not only general features, but also specific features on the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2022.3173587, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE XIII
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON CELL TYPE IDENTIFICATION FOR INTER-DATASET VALIDATION.

State-of-the-art	SingleR [6]	singleCellNet [39]	**ACTINN [40]**
Median F1-score	44%	22%	**11%**
Supervised Learning	SVM	Random Forest	**RCNN**
Median F1-score	18%	12%	**19**
Semi-supervised Deep Learning	SemiRNet (1%)	SemiRNet (5%)	SemiRNet (10%)
Median F1-score	**30%**	**29%**	**32%**

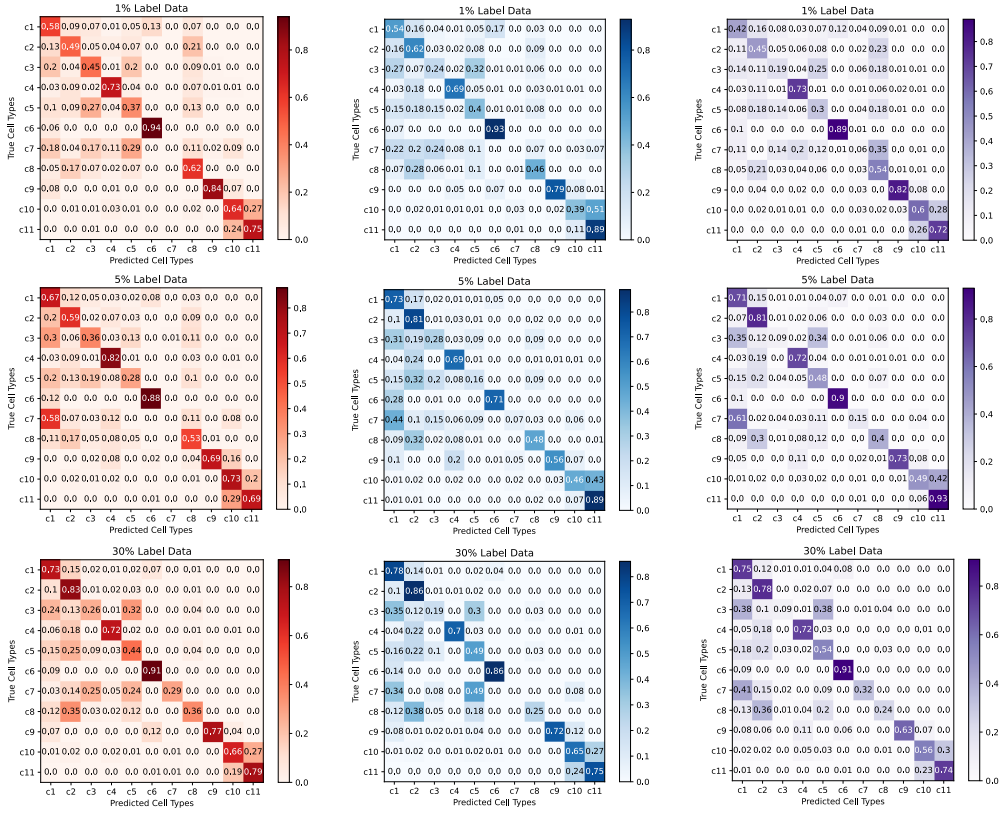


Fig. 5. Comparison of confusion matrix on different cell types generated with batch size 128, 256 and 512 on 68K dataset. The left column, the middle column, and the right column is for the case of 128, 256, and 512, respectively.

training data for specific tasks. It will lead to low performance on testing data whose distribution is significantly different from that of training data. Thus, it is not suitable to directly apply supervised deep learning techniques to inter-dataset applications. However, compared to supervised learning, the proposed semi-supervised method improved the performance significantly since it reduces the influence of distribution differences between training data and testing data by using less labeled data. Furthermore, unlabeled data can enhance the performance through improving data representations.

V. RELATED WORK

Single-cell RNA-seq (scRNAseq) data is able to profile the gene expression levels of cells and to link the dynamics at the molecular level and the cellular level. Analyzing scRNAseq data will be beneficial for obtaining knowledge on cancer drug resistance, gene regulation in embryonic development, and mechanisms of stem cell differentiation and reprogramming [41], [42]. In recent years, a lot of progresses have been made on applying bioinformatics techniques and machine learning tools to scRNAseq data [43]. However, there still exist many challenges due to dropout events, batch effect, noise, high dimensionality, and scalability [15].

To overcome these challenges, deep learning techniques have been employed to build effective and efficient computational methods for scRNAseq data. For example, Shaham *et al.* proposed MMD-ResNet to remove batch effect on both mass cytometry and scRNAseq data by combining residual neural networks (ResNets) with the maximum mean discrepancy (MMD) [44]. To reduce the computational cost, Li *et al.* implemented batch effect removal and clustering in one step [45]. Specifically, they built a stacked autoencoder [46] to enhance clustering performance. On the other hand, to remove fake zeros, autoencoder based methods such as "AutoImpute" [47] and "DCA" [48] have been proposed to implement imputation and denoising to address the issue of dropout. Moreover, autoencoder techniques such as denoising autoencoder (DAE) [49] and variational autoencoder

(VAE) [50] have also been applied to reduce dimensions of scRNAseq data [44], [49], [51]. In addition, Lopez *et al.* developed an integrative pipeline called "scVI" (single-cell variational inference) to implement multiple tasks including correcting batch effect, removing dropout, imputation, dimension reduction, clustering, and visualization [52]. Wang *et al.* proposed an interpretable deep-learning architecture using capsule networks (called scCapsNet) to perform feature selection to identify groups of genes encoding different subcellular types [53]. Shao *et al.* built a pre-trained cell-type annotation tool *scDeepSort* through combining a deep learning model with a weighted graph neural network (GNN), which is the first attempt to annotate cell types of scRNA-seq data with a pre-trained GNN model [54]. Similarly, Wang *et al.* proposed a multimodal end-to-end deep learning model through integrating a graph convolutional network (GCN) and a neural network [55]. Cheng *et al.* combined deep learning with graphic cluster (DGCyTOF) visualization to identify cell types [57]. Chen *et al.* proposed a probabilistic generative model integrated with a Bayesian neural network to annotate scCAS data in a supervised manner [58]. O'Connor *et al.* classified cells based on their time-varying behavior by a recurrent bi-directional long short-term memory (Bi-LSTM) network [59]. Recently, Ma *et al.* performed extensive data analyses to systematically evaluate supervised methods for cell identification and suggested combining all individuals from available datasets to construct the reference dataset and use multi-layer perceptron (MLP) as the classifier [56]. In addition to supervised learning based methods, Lieberman *et al.* employed transfer learning [60] to reuse a classification scheme that was learned from previous similar experiments for cell type classification [3]. Hu *et al.* developed a transfer learning algorithm that borrows ideas from supervised cell type classification algorithms, but also leverages information in target data to ensure sensitivity in classifying cells that are only present in the target data [61].

In summary, most of existing work focuses on supervised learning based methods to implement cell identification, which requires large amounts of fully annotated samples to train the model. In addition, even if transfer learning can be employed to reduce the requirement of big annotated training data, how transfer learning improves the identification performance is not transparent. This paper proposed a semi-supervised method that can leverage a small part of labeled data together with large amounts of unlabeled data for cell identification, which will not require fully annotated training sets. Moreover, how the unlabeled data improves the performance is interpretable.

VI. Conclusion and Future Work

In this paper, a novel framework of deep semi-supervised learning is proposed for cell type identification on scRNAseq data. As an emerging research area, implementing cell type identification automatically is extremely important for the downstream analysis on the scRNAseq data. However, current methods using neural networks rely on the availability of large amount of labeled cells, which costs a huge amount of efforts to label these cells with high quality. Hence, we propose a deep semi-supervised learning model based on recurrent convolutional neural networks (RCNN) that can utilize unlabeled cells to enhance identification performance. There are two paths in the model for obtaining supervised cross entropy loss and unsupervised mean squared error loss, respectively. Then training is performed by jointly optimizing these two losses, and this allows the proposed scheme to take advantage of both information from the labeled cells and information from the unlabeled cells. Furthermore, we introduce a preprocessing procedure to overcome the problem of data sparsity. Experimental results indicate that the proposed model could identify cell type effectively using very limited labeled cells and a large amount of unlabeled cells. In our future work, we plan to extend the proposed model for other tasks such as pathway network construction.

Acknowledgment

This research work is supported in part by the Texas A&M Chancellor's Research Initiative (CRI), the U.S. National Science Foundation (NSF) award 1464387 and 1736196, and by the U.S. Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) under agreement number FA8750-15-2-0119. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. National Science Foundation (NSF) or the U.S. Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) or the U.S. Government.

References

[1] C. Trapnell, "Defining cell types and states with single-cell genomics," *Genome research*, vol. 25, no. 10, pp. 1491–1498, 2015.

[2] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature biotechnology*, vol. 36, no. 5, p. 411, 2018.

[3] Y. Lieberman, L. Rokach, and T. Shay, "Castle–classification of single cells by transfer learning: Harnessing the power of publicly available single cell rna sequencing experiments to annotate new experiments," *PloS one*, vol. 13, no. 10, p. e0205499, 2018.

[4] V. Y. Kiselev, A. Yiu, and M. Hemberg, "scmap: projection of single-cell rna-seq data across data sets," *Nature methods*, vol. 15, no. 5, pp. 359–362, 2018.

[5] J. Alquicira-Hernandez, A. Sathe, H. P. Ji, Q. Nguyen, and J. E. Powell, "scpred: Cell type prediction at single-cell resolution," *bioRxiv*, p. 369538, 2018.

[6] D. Aran, A. P. Looney, L. Liu, E. Wu, V. Fong, A. Hsu, S. Chak, R. P. Naikawadi, P. J. Wolters, A. R. Abate *et al.*, "Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage," *Nature immunology*, vol. 20, no. 2, pp. 163–172, 2019.

[7] F. Ma and M. Pellegrini, "Actinn: automated identification of cell types in single cell rna sequencing," *Bioinformatics*, 2019.

[8] Y. Xu, Z. Zhang, L. You, J. Liu, Z. Fan, and X. Zhou, "scigans: single-cell rna-seq imputation using generative adversarial networks," *Nucleic acids research*, vol. 48, no. 15, pp. e85–e85, 2020.

[9] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein *et al.*, "A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure," *Cell systems*, vol. 3, no. 4, pp. 346–360, 2016.

[10] T. Abdelaal, L. Michielsen, D. Cats, D. Hoogduin, H. Mei, M. J. Reinders, and A. Mahfouz, "A comparison of automatic cell identification methods for single-cell rna sequencing data," *Genome biology*, vol. 20, no. 1, pp. 1–19, 2019.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[12] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[13] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck *et al.*, "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets," *Cell*, vol. 161, no. 5, pp. 1202–1214, 2015.

[14] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu *et al.*, "Massively parallel digital transcriptional profiling of single cells," *Nature communications*, vol. 8, no. 1, pp. 1–12, 2017.

[15] J. Zheng and K. Wang, "Emerging deep learning methods for single-cell rna-seq data analysis," *Quantitative Biology*, vol. 7, no. 4, pp. 247–254, 2019.

[16] D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz *et al.*, "Eleven grand challenges in single-cell data science," *Genome Biology*, vol. 21, no. 1, pp. 1–35, 2020.

[17] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

[18] L. Tran, X. Liu, J. Zhou, and R. Jin, "Missing modalities imputation via cascaded residual autoencoder," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1405–1414.

[19] F. Zhou, Q. Gao, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Trajectory-user linking via variational autoencoder." in *IJCAI*, 2018, pp. 3212–3218.

[20] S. Chowdhury, X. Dong, O. A. Solis, L. Qian, and X. Li, "Cell type identification from single-cell transcriptomic data via gene embedding," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 258–263.

[21] J. Du, P. Jia, Y. Dai, C. Tao, Z. Zhao, and D. Zhi, "Gene2vec: distributed representation of genes based on co-expression," *BMC genomics*, vol. 20, no. 1, pp. 7–15, 2019.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[23] N. Yang, S. Liu, M. Li, M. Zhou, and N. Yu, "Word alignment modeling with context dependent deep neural network," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 166–175.

[24] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims, "Evaluation methods for unsupervised word embeddings," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 298–307.

[25] Q. Zou, P. Xing, L. Wei, and B. Liu, "Gene2vec: gene subsequence embedding for prediction of mammalian n6-methyladenosine sites from mrna," *Rna*, vol. 25, no. 2, pp. 205–218, 2019.

[26] C. T. Choy, C. H. Wong, and S. L. Chan, "Embedding of genes using cancer gene expression data: biological relevance and potential application on biomarker discovery," *Frontiers in genetics*, vol. 9, p. 682, 2019.

[27] K. Ovens, F. Maleki, B. F. Eames, and I. McQuillan, "Juxtapose: a gene-embedding approach for comparing co-expression networks," *BMC bioinformatics*, vol. 22, no. 1, pp. 1–26, 2021.

[28] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European conference on computer vision*. Springer, 2014, pp. 94–108.

[29] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[30] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[31] R. D. Hodge, T. E. Bakken, J. A. Miller, K. A. Smith, E. R. Barkan, L. T. Graybuck, J. L. Close, B. Long, O. Penn, Z. Yao *et al.*, "Conserved cell types with divergent features between human and mouse cortex," *BioRxiv*, p. 384826, 2018.

[32] B. Tasic, Z. Yao, L. T. Graybuck, K. A. Smith, T. N. Nguyen, D. Bertagnolli, J. Goldy, E. Garren, M. N. Economo, S. Viswanathan *et al.*, "Shared and distinct transcriptomic cell types across neocortical areas," *Nature*, vol. 563, no. 7729, pp. 72–78, 2018.

[33] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *arXiv preprint arXiv:1309.4168*, 2013.

[34] V. Van Asch, "Macro-and micro-averaged evaluation measures [[basic draft]]," *Belgium: CLiPS*, vol. 49, 2013.

[35] Y. Yang, "A study of thresholding strategies for text categorization," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 137–145.

[36] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.

[37] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[38] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016, pp. 207–212.

[39] Y. Tan and P. Cahan, "Singlecellnet: a computational tool to classify single cell rna-seq data across platforms and across species," *Cell systems*, vol. 9, no. 2, pp. 207–213, 2019.

[40] F. Ma and M. Pellegrini, "Actinn: automated identification of cell types in single cell rna sequencing," *Bioinformatics*, vol. 36, no. 2, pp. 533–538, 2020.

[41] M. B. Pouyan and M. Nourani, "Clustering single-cell expression data using random forest graphs," *IEEE journal of biomedical and health informatics*, vol. 21, no. 4, pp. 1172–1181, 2016.

[42] F. Tang, K. Lao, and M. A. Surani, "Development and applications of single-cell transcriptome analysis," *Nature methods*, vol. 8, no. 4s, p. S6, 2011.

[43] H. Wang, P. Sham, T. Tong, and H. Pang, "Pathway-based single-cell rna-seq classification, clustering, and construction of gene-gene interactions networks using random forests," *IEEE Journal of Biomedical and Health Informatics*, 2019.

[44] U. Shaham, K. P. Stanton, J. Zhao, H. Li, K. Raddassi, R. Montgomery, and Y. Kluger, "Removal of batch effects using distribution-matching residual networks," *Bioinformatics*, vol. 33, no. 16, pp. 2539–2546, 2017.

[45] X. Li, Y. Lyu, J. Park, J. Zhang, D. Stambolian, K. Susztak, G. Hu, and M. Li, "Deep learning enables accurate clustering and batch effect removal in single-cell rna-seq analysis," *bioRxiv*, p. 530378, 2019.

[46] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[47] D. Talwar, A. Mongia, D. Sengupta, and A. Majumdar, "Autoimpute: Autoencoder based imputation of single-cell rna-seq data," *Scientific reports*, vol. 8, no. 1, pp. 1–11, 2018.

[48] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell rna-seq denoising using a deep count autoencoder," *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.

[49] C. Lin, S. Jain, H. Kim, and Z. Bar-Joseph, "Using neural networks for reducing the dimensions of single-cell rna-seq data," *Nucleic acids research*, vol. 45, no. 17, pp. e156–e156, 2017.

[50] J. Ding, A. Condon, and S. P. Shah, "Interpretable dimensionality reduction of single cell transcriptome data with deep generative models," *Nature communications*, vol. 9, no. 1, pp. 1–13, 2018.

[51] H. Cho, B. Berger, and J. Peng, "Generalizable and scalable visualization of single-cell data using neural networks," *Cell systems*, vol. 7, no. 2, pp. 185–191, 2018.

[52] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," *Nature methods*, vol. 15, no. 12, p. 1053, 2018.

[53] L. Wang, R. Nie, Z. Yu, R. Xin, C. Zheng, Z. Zhang, J. Zhang, and J. Cai, "An interpretable deep-learning architecture of capsule networks for identifying cell-type gene expression programs from single-cell rna-sequencing data," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 693–703, 2020.

[54] X. Shao, H. Yang, X. Zhuang, J. Liao, P. Yang, J. Cheng, X. Lu, H. Chen, and X. Fan, "scdeepsort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network," *Nucleic acids research*, vol. 49, no. 21, pp. e122–e122, 2021.

[55] T. Wang, J. Bai, and S. Nabavi, "Single-cell classification using graph convolutional networks," *BMC bioinformatics*, vol. 22, no. 1, pp. 1–23, 2021.

[56] W. Ma, K. Su, and H. Wu, "Evaluation of some aspects in supervised cell type identification for single-cell rna-seq: classifier, feature selection, and reference construction," *Genome biology*, vol. 22, no. 1, pp. 1–23, 2021.

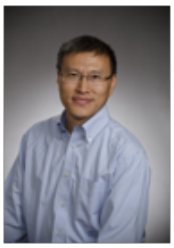
[57] L. Cheng, P. Karkhanis, B. Gokbag, and L. Li, "Dgcytof: deep learning with graphic cluster visualization to predict cell types of single cell mass cytometry data," *bioRxiv*, 2021.
[58] X. Chen, S. Chen, S. Song, Z. Gao, L. Hou, X. Zhang, H. Lv, and R. Jiang, "Cell type annotation of single-cell chromatin accessibility data via supervised bayesian embedding," *Nature Machine Intelligence*, pp. 1–11, 2022.
[59] T. O'Connor, A. Anand, B. Andemariam, and B. Javidi, "Deep learning-based cell identification and disease diagnosis using spatio-temporal cellular dynamics in compact digital holographic microscopy," *Biomedical Optics Express*, vol. 11, no. 8, pp. 4491–4508, 2020.
[60] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
[61] J. Hu, X. Li, G. Hu, Y. Lyu, K. Susztak, and M. Li, "Iterative transfer learning with neural network for clustering and cell type classification in single-cell rna-seq analysis," *Nature machine intelligence*, vol. 2, no. 10, pp. 607–618, 2020.

Xishuang Dong received B.S. degree in Computer Science and Technique at Harbin University of Science and Technology, Harbin, China, M.S. degree in Computer Software and Theory at Harbin Engineering University, Harbin, China, and Ph.D. degree in Computer Application at Harbin Institute of Technology, Harbin, China. He is an Assistant Professor in Electrical and Computer Engineering Department at Prairie View A&M University (PVAMU). His research interests are in the area of Deep Learning, Computational Systems Biology, Biomedical Information Processing, Big Data Analysis, and Natural Language Processing.

Shanta Chowdhury received B. Sc. degree in Electrical and Electronics Engineering, Bangladesh, in 2016 and M.S. degree in Electrical Engineering from Prairie View A&M University (PVAMU), USA, in 2018. She is currently working as a Research Assistant in the Center of Excellence in Research and Education for Big Military Data Intelligence (CREDIT Center) and pursuing Ph.D. degree in the Department of Electrical and Computer Engineering, Prairie View A&M University. Her research interests include Biomedical Big Data Analysis and Machine Learning.

Uboho Victor received B.Tech degree in Information Technology from the Federal University of Technology Owerri, Imo State, Nigeria, in 2014 and M.S. in Computer Science from Prairie View A&M University, Texas, USA in 2020. He is currently working as a Senior Software Engineer with Dell Technologies, Austin, Texas, USA. His research interests are in the area of Social Media Data Analysis, Natural Language Processing, and Deep Learning.

Xiangfang Li is an Associate Professor in the Department of Electrical and Computer Engineering at Prairie View A&M University (PVAMU). Before she joined PVAMU, she was a TEES Associate Research Scientist at Texas A&M University in College Station, TX, and actively participated in the Bioinformatics Training Program sponsored by NIH. She received her M.S. and Ph.D. in Computer Engineering from Rutgers University in 2003 and 2007, respectively. Her research interests are in computational and systems biology, systems pharmacology, computer networking and communication, and artificial intelligence. She is the recipient of the Outstanding Researcher of the Year award from the Roy G. Perry College of Engineering of PVAMU in 2017.



Lijun Qian received B.E. degree from Tsinghua University, Beijing, China, M.S. degree from the Technion-Israel Institute of Technology, Haifa, Israel, and Ph.D. degree from Rutgers University, NJ, USA. He was a Technical Staff Member of Bell-Labs Research, Murray Hill, NJ, USA. He is currently Regents Professor and holds the AT&T Endowment with the Department of Electrical and Computer Engineering, Prairie View A&M University, Prairie View, TX, USA, where he is also the Director of the Center of Excellence in Research and Education for Big Military Data Intelligence (CREDIT Center). He was a Visiting Professor to Aalto University, Espoo, Finland. His research interests are in the area of big data processing, artificial intelligence, wireless communications and mobile networks, network security and intrusion detection, and computational systems biology.