

# Supervised Adversarial Alignment of Single-Cell RNA-seq Data

Songwei Ge<sup>1</sup>✉, Haohan Wang<sup>2</sup>, Amir Alavi<sup>1</sup>, Eric Xing<sup>2,3</sup>, and Ziv Bar-Joseph<sup>1,3</sup>✉

<sup>1</sup> Computational Biology Department

<sup>2</sup> Language Technology Institute

<sup>3</sup> Machine Learning Department

Carnegie Mellon University, Pittsburgh, 15312, US

{songweig, haohanw, aalavi, epxing, zivbj}@cs.cmu.edu

**Abstract.** Dimensionality reduction is an important first step in the analysis of single cell RNA-seq (scRNA-seq) data. In addition to enabling the visualization of the profiled cells, such representations are used by many downstream analyses methods ranging from pseudo-time reconstruction to clustering to alignment of scRNA-seq data from different experiments, platforms, and labs. Both supervised and unsupervised methods have been proposed to reduce the dimension of scRNA-seq. However, all methods to date are sensitive to batch effects. When batches correlate with cell types, as is often the case, their impact can lead to representations that are batch rather than cell type specific. To overcome this we developed a domain adversarial neural network model for learning a reduced dimension representation of scRNA-seq data. The adversarial model tries to simultaneously optimize two objectives. The first is the accuracy of cell type assignment and the second is the inability to distinguish the batch (domain). We tested the method by using the resulting representation to align several different datasets. As we show, by overcoming batch effects our method was able to correctly separate cell types, improving on several prior methods suggested for this task. Analysis of the top features used by the network indicates that by taking the batch impact into account, the reduced representation is much better able to focus on key genes for each cell type.

**Keywords:** Dimensionality Reduction · Single-Cell RNA-seq · Batch Effect Removal · Domain Adversarial Training · Data Integration.

## 1 Introduction

Single-cell RNA sequencing (scRNA-seq) has revolutionized the study of gene expression programs [14, 26]. The ability to profile genes at the single-cell level has revealed novel specific interactions and pathways within cells [41], differences in the proportions of cell types between samples [16, 42], and the identity and characterization of new cell types [38]. Several biological tissues, systems, and processes have recently been studied using this technology [16, 42, 41].

While studies using scRNA-seq provide many insights, they also raise new computational challenges. One of the major challenges involves the ability to integrate and compare results from multiple scRNA-seq studies. There are several different commercial platforms for performing such experiments, each with their own biases. Furthermore, similar to other high throughput genomic assays, scRNA-seq suffers from batch effects which can make cells profiled in one lab look very different from the same cells profiled at another lab [37, 36]. Moreover, other types of high throughput transcriptomics profiling, including microscopy-based techniques, are also generating single cell expression datasets [39, 7]. The goal of fully utilizing these spatial datasets motivates the development of methods that can combine them with scRNA-seq when studying specific biological tissues and processes.

A number of recent methods have attempted to address this challenge by developing methods for aligning scRNA-seq data from multiple studies of the same biological system. Many of these methods rely on identifying nearest neighbors between the different datasets and using them as anchors. Methods that use this approach include Mutual Nearest Neighbors (MNN) [11] and Seurat [35]. Others including scVI and scAlign first embed all datasets into a common lower dimensional space. scVI encodes the scRNA-seq data with a deep generative model conditioning on the batch identifiers [23] while scAlign regularizes the representation between two datasets by minimizing the random walk probability differences between the original and embedding spaces. While these methods were successful for some datasets, here we show that they are not always able to correctly match all cell types. A key problem with these methods is the fact that they are unsupervised and rely on the assumption that cell types profiled by the different studies overlap. While this works for some datasets, it may fail for studies in which cells do not fully overlap or for those containing rare cell types. Unsupervised methods tend to group rare types with the larger types making it hard to identify them in a joint space.

Recent machine learning work has focused on a related problem termed “domain generalization” or “domain adaptation”. Methods developed for these problems attempt to learn representations of diverse data that are invariant to technical confounders [24, 4]. These methods have been used for multiple applications such as machine translation for domain specific corpus [3] and face detection [27]. Several of the methods proposed for domain adaptation rely on the use of adversarial methods [4, 9, 20, 40], which has been proved effective to align latent distributions. In addition to the original task such as classification, these methods apply a domain classifier upon the learned representations. The

encoder network is used for both improving accurate classification while at the same time reducing the impact of the domain (by “fooling” a domain classifier). This is achieved by learning encoder weights that simultaneously perform gradient *descent* on the label classification task and gradient *ascent* on the domain classification task.

Here we extend these approaches, coupling them with Siamese network learning [19] for overcoming batch effects in scRNA-seq analysis. We define a “domain” in this paper as a standalone dataset profiled at a single lab using a single platform. We define “label” as the cell type for each cell in the dataset. Considering the specificity of the cell types in the scRNA-seq datasets, we propose a conditional pair sampling strategy that constrains input pair selection when training the adversarial network. We discuss how to formulate a domain adaptation network for scRNA-seq data, how to learn the parameters for the network, and how to train it using available data.

We tested our method on several datasets ranging in size from 10 to 39 cell types and from 4 to 155 batches. As we show, for all of the datasets our domain adversarial method improves on previous methods, in some cases significantly. Visualization of the learned representation from several different methods helps highlight the advantages of the domain adversarial framework. As we show, the framework is able to accurately mitigate the batch effects while maintaining the grouping of cells from the same type across different batches. Biological analysis of the resulting model identifies key genes that can correctly distinguish between cell types across different experiments. Such batch invariant genes are promising candidates for a cell type specific signature that can be used across different studies to annotate cells.

## 2 Methods

### 2.1 Problem Formulation

To formulate the problem we start with a few notation definitions. We assume that the single cell RNA-seq data are drawn from the input space  $\mathbf{X} \in \mathbb{R}^p$  where each sample (a cell)  $\mathbf{x}$  has  $p$  features corresponding to the gene expression values. Cells are also associated with the label  $y \in \mathbf{Y} = \{1, 2, \dots, K\}$  which represents their cell types. We associate each sample with a specific domain/batch  $d \in \mathcal{D}$  that represents any standalone dataset profiled at a single lab using a single platform. Note that we will use domain and batch interchangeably in this paper for convenience. The data are divided into a training set and a test set that are drawn from multiple studies. The domains used to collect training data are not used for the test set and so batch effects can vary between the training and test data. In practice, each of the domains only contains a small subset of the cell types. This means that the distribution of cell types is correlated with the distribution of domains. Thus, the methods that naively learn cell types based on expression profile [2, 17, 21] may instead fit domain information and not generalize well to the unobserved studies.

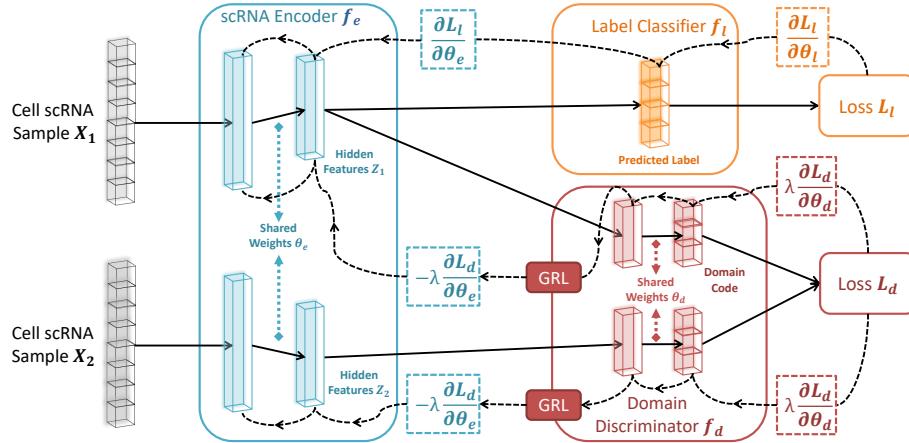


Fig. 1: Architecture of scDGN. The network includes three modules: scRNA encoder  $f_e$  (blue), label classifier  $f_l$  (orange) and domain discriminator  $f_d$  (red). Note that the red and orange networks use the same encoding as input. Solid lines represent the forward direction of the neural network while the dashed lines represent the backpropagation direction with the corresponding gradient it passes. Gradient Reversal Layers (GRL) have no effect in forward propagation, but flip the sign of the gradients that flow through them during backpropagation. This allows the combined network to simultaneously optimize label classification and attempt to “fool” the domain discriminator. Thus, the encoder leads to representations that are invariant to the different domains while still distinguishing cell types.

## 2.2 Domain Adversarial Training with Siamese Network

To overcome this problem and remove the domain impact when learning a cell type representation we propose a neural network framework which includes three modules as shown in Figure 1: scRNA encoder, label classifier, and domain discriminator. The encoder module  $f_e(\mathbf{x}; \theta_e)$  is used to reduce the dimensions of the data and contains fully connected layers which produce the hidden features, where  $\theta_e$  represents the parameters in these layers. The label classifier  $f_l(f_e; \theta_l)$  attempts to predict the label of input  $\mathbf{x}_1$  whereas the goal of the domain discriminator  $f_d(f_e; \theta_d)$  is to determine whether a pair of inputs  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are from the same domain or not. Past work for classifying scRNA-seq data only attempted to minimize the loss function for the label classifier  $\mathcal{L}_l(f_l(f_e; \theta_l))$  [22, 2]. Here, we extend these methods by adding a regularization term based on the adversarial loss of the domain discriminator  $\mathcal{L}_d(f_d(f_e; \theta_d))$  which we will elaborate later. The overall loss  $E$  on a pair of samples  $\mathbf{x}_1$  and  $\mathbf{x}_2$  is denoted by:

$$E(\theta_e, \theta_l, \theta_d) = \mathcal{L}_l(f_l(f_e(\mathbf{x}_1; \theta_e); \theta_l)) - \lambda \mathcal{L}_d(f_d(f_e(\mathbf{x}_1; \theta_e); \theta_d), f_d(f_e(\mathbf{x}_2; \theta_e); \theta_d)),$$

where  $\lambda$  can control the trade-off between the goals of domain invariance and higher classification accuracy. For convenience, we use  $\mathbf{z}_1$  and  $\mathbf{z}_2$  to denote

the hidden representations of  $\mathbf{x}_1$  and  $\mathbf{x}_2$  calculated from  $f_e(\mathbf{x}; \theta_e)$ . Inspired by Siamese networks [19], we implement our domain discriminator by adopting a contrastive loss [10]:

$$\begin{aligned}\mathcal{L}_d(f_d(\mathbf{z}_1; \theta_d), f_d(\mathbf{z}_2; \theta_d)) &= U \frac{1}{2} D(f_d(\mathbf{z}_1), f_d(\mathbf{z}_2))^2 \\ &\quad + (1 - U) \frac{1}{2} (\max\{0, m - D(f_d(\mathbf{z}_1), f_d(\mathbf{z}_2))\})^2,\end{aligned}$$

where  $U = 1$  indicates that two samples are from the same domain  $d$  and  $U = 0$  indicates that they are not,  $D(\cdot)$  is the euclidean distance, and  $m$  is the margin that indicates the prediction boundary. The domain discriminator parameters,  $\theta_d$ , are updated using back propagation to *maximize* the total loss  $E$  as mentioned in the Introduction while the encoder and classifier parameters,  $\theta_e$  and  $\theta_l$ , are updated to *minimize*  $E$ . To allow all three modules to be updated together end-to-end, we use a Gradient Reversal Layer (Figure 1) [9, 28]. Specifically, Gradient Reversal Layers (GRL) have no effect in forward propagation, but flip the sign of the gradients that flow through them during backpropagation. The following provides the overall optimization problems solved for the network parameters:

$$\begin{aligned}(\widehat{\theta}_e, \widehat{\theta}_l) &= \arg \min_{\theta_e, \theta_l} E(\theta_e, \theta_l, \widehat{\theta}_d) \\ (\widehat{\theta}_d) &= \arg \max_{\theta_d} E(\widehat{\theta}_e, \widehat{\theta}_l, \theta_d)\end{aligned}$$

In other words, the goal of the domain discriminator is to tell if two samples are drawn from the same or different batches. By optimizing the scRNA encoder adversarially against the domain discriminator, we attempt to make sure that the network representation cannot be used to classify based on domain knowledge. During the training, the maximization and minimization tasks compete with each other, which is achieved by adjusting the representations to improve the accuracy of the label classifier and simultaneously fool the domain discriminator.

### 2.3 Conditional Domain Generalization Strategy

Most prior domain adaption or generalization methods focused on the cases where the distribution of labels is independent of the domains [24, 4]. In contrast, as we show in Results, for scRNA-seq experiments different studies tend to focus on certain cell types [16, 41, 42]. Consequently, it is not reasonable to completely merge the scRNA-seq data from different batches. To be specific, aligning the scRNA-seq data from two batches with different sets of cell types would sacrifice its biological significance and prevent the cell classifier from predicting effectively. To overcome this issue, instead of arbitrarily choosing positive pairs (samples from the same domain) and negative pairs (samples from different domains), we constrain the selection as follows: 1) for positive pairs, only the

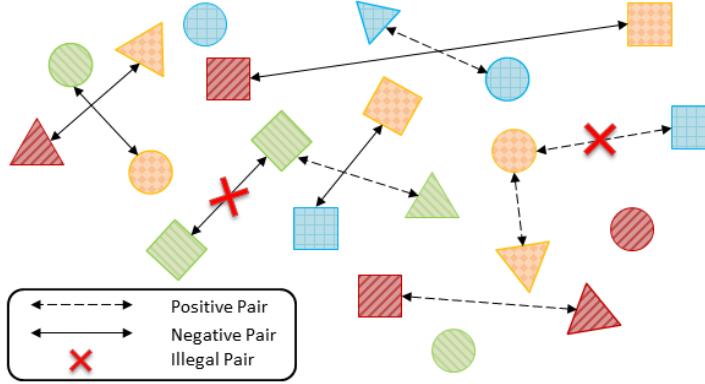


Fig. 2: Conditional domain generalization strategy: Shapes represent different labels and colors (or patterns) represent different domains. For negative pairs from different domains, we only select those samples with the same label. For positive pairs from the same domain, we only select the samples with different labels.

samples with different labels from the same domain are selected. 2) for negative pairs, only the samples with the same label from different domains are selected. Figure 2 provides a visual interpretation of this strategy. Formally, letting  $y_i$  and  $z_i$  represent the  $i$ -th sample's cell-type label and domain label respectively, we have the following equations to define the value of  $U$  for sample pairs:

$$U = \begin{cases} 0, & z_1 \neq z_2 \text{ and } y_1 = y_2 \\ 1, & z_1 = z_2 \text{ and } y_1 \neq y_2 \end{cases}$$

This strategy prevents the domain adversarial training from aligning samples with different labels or separating samples with same labels. For example, in order to fool the discriminator with a positive pair, the encoder must implicitly increase the distance of two samples with different cell types. Therefore, combining this strategy with domain adversarial training allows the network to learn cell type specific, focused representations. We term our model Single Cell Domain Generalization Network (scDGN).

### 3 Results

#### 3.1 Experiment Setups

**Datasets** To test our method and to compare it to previous methods for aligning and classifying scRNA-seq data, we used several recent datasets. These datasets contain between 6,000 and 45,000 cells, and all include cells profiled in multiple experiments by different labs and on different platforms.

Table 1: Basic statistics for scQuery, Suerat pancreas, and PBMC datasets

	scQuery			Seurat pancreas			Seurat pbmc		
	data	cell type	domain	data	cell type	domain	data	cell type	domain
training	37697	39	99	6321	13	3	25977	10	8
validation	3023	19	26	-	-	-	-	-	-
test	3770	23	30	638	13	1	2992	10	1

*scQuery*: We use a subset of the dataset provided by scQuery, which includes uniformly processed data from hundreds of studies [2]<sup>1</sup>. The dataset we use contains 44,490 samples from 155 different experiments. scQuery assigns cells to 39 types spanning diverse categories ranging from immune cells to neurons to organ specific cells. We use 99 of the 155 batches for training, 26 for validation, and 30 for testing. We provide a list of the studies used for each set in the Appendix A.1. Statistics for the different datasets are shown in Table 1. RPKM normalization is applied to the 20,499 genes in each sample. Note that while there are 39 cell-types in the training set, only 19 and 23 of them are included in the validation and test set. This mimics the application of the methods to future studies that may not profile all types of cells.

*PBMC* : The Peripheral Blood Mononuclear Cells (PBMC) dataset contains 28,969 cells assigned to 10 blood cell types. The data are profiled in 9 batches (from 8 different sequencing technologies) [5]. We use the data from the *10xChromiumv2A* platform as test data and the rest as training data. Following the provided tutorial [1], we use the top 3000 variable genes for the analysis.

*Seurat pancreas*: The Seurat pancreas dataset is designed for evaluating single cell alignment algorithms and contains 6321 scRNA-seq samples of human pancreatic islet cell produced by four studies. We use the smallest study for the test data and the other three for training as shown in Table 1. Thirteen canonical labels of the pancreatic islet cell are assigned to cells in each study. Similar to the Seurat PBMC dataset, we only used the 3000 most variable genes. To further simulate the correlation between cell types and domains for this dataset we randomly remove the data for 6 of the 13 cell types for each of the training domains. As a result, we construct 6 synthetic datasets based this strategy to evaluate the alignment performance of different methods under a high label-domain correlation setting. The specific cell type information of each dataset is listed Appendix A.3.

**Model Configurations** We used the network of Lin et al [22] as the components for the encoder and the label classifier in our model. The encoder contains two hidden layers with 1136 and 100 units. The label classifier is directly connected to the 100 unit layer and makes predictions based on these values. The

<sup>1</sup> [https://scquery.cs.cmu.edu/processed\\_data/](https://scquery.cs.cmu.edu/processed_data/)

domain discriminator contains an additional hidden layer with 64 units and is also connected to the 100 unit layer of the encoder (Figure 1). For each layer,  $tanh()$  is used as the non-linear activation function. We test several other possible configurations but did not observe improvement in performance. As is commonly done, we use a validation set to tune the hyperparameters for learning including learning rates, decay, momentum, and the adversarial weight and margin parameters  $\lambda$  and  $m$ . Generally, our analysis indicates that for larger datasets a lower weight  $\lambda$  and larger margin  $m$  for the adversarial training is preferred and vice versa. More details about the hyperparameters and training are provided in Appendix A.2.

**Baselines** We compared scDGN to several prior methods for classifying and aligning scRNA-seq data. These included the neural network (NN) model of Lin et al [22] which is developed for classifying scRNA-seq data, CaSTLe [21] which performs cell type classification based on transfer learning, and several state-of-the-art alignment methods. For alignment, we compared to MNN [11] which utilizes mutual nearest neighbors to align data from different batches, scVI [23] which trains a deep generative model on the scRNA-seq data and uses an explicit batch identifier to retain conditional independence property of the representation, and Seurat [35] which first identifies the anchors among different batches and then projects different datasets using a correction vector based on the order defined by hierarchical clustering with pairwise distances. Our comparisons include both visual projection of the learned alignment (Figure 4 and 5) and quantitative analysis of the accuracy of the predicted test cell types (Table 2). For the latter, to enable comparisons of the supervised and unsupervised methods, we used the resulting aligned data from the unsupervised methods to train a neural network that has the same configuration as Lin et al [22]. For scVI, which results in a much lower dimensional representation, we used a smaller input vector and a smaller hidden layer. Note that these alignment methods actually use the scRNA-seq test data to determine the final dimensionality reduction function while our method does not utilize the test data for any model decision or parameter learning. To effectively apply Seurat to scQuery, we remove the batches which have  $< 100$  samples. Also, for those datasets that the assumption of overlapped cell types is not guaranteed such as scQuery, we find that the performance of MNN highly depends on the order of alignment. Therefore, for MNN on the scQuery dataset, we use 10 random permutations of batch orders and report the average accuracy.

### 3.2 Overall Performance

As mentioned above, we use the validation set to select the best model when using the scQuery dataset. For the smaller datasets, we use the model obtained after 250 epochs (all models converged after this number of epochs). Test accuracy for the different methods is presented in Table 2. We show both mean and standard deviation of the accuracy for 10 randomly initialized experiments. We

Table 2: Overall performances of different methods.  $MI$  represents the mutual information between batch and cell type in the corresponding dataset. The highest test accuracy for each dataset is bolded.

Experiments	MI	NN[22]	CaSTLe [21]	MNN [11]	scVI [23]	Seurat [35]	scDGN
scQuery	3.025	0.255	0.156	0.200	0.257	0.144	<b>0.286</b>
PBMC	0.112	0.861	0.865	0.859	0.808	0.830	<b>0.868</b>
Pancreas 1	0.902	0.720	0.705	0.591	0.855	0.812	<b>0.856</b>
Pancreas 2	<b>0.733</b>	0.891	0.764	0.764	0.852	0.825	<b>0.918</b>
Pancreas 3	0.931	0.545	0.722	0.722	0.651	<b>0.751</b>	0.663
Pancreas 4	0.458	0.927	0.914	0.914	<b>0.925</b>	0.881	<b>0.925</b>
Pancreas 5	0.849	0.928	0.882	<b>0.932</b>	0.895	0.865	0.923
Pancreas 6	0.670	0.944	0.917	0.946	0.893	0.907	<b>0.950</b>
Average	-	0.826	0.817	0.842	0.845	0.840	<b>0.872</b>

also report the performances on different cell types in Appendix B. In addition, Table 2 presents the Mutual Information (MI) between labels and domains which corresponds to the difficulty of the dataset. A larger MI indicates that models that do not account for the domain are likely to fit the domain information rather than the cell type. For the scQuery dataset, we find the accuracy is low for all methods indicating that this dataset is relatively difficult. This is corroborated by the large MI value. For such data we see a clear advantage for the scDGN: scDGN improves by over 10% over all other methods ( $p = 5.069 \times 10^{-5}$  based on Student's t-test when compared to the NN baseline which is tied for second best). The improvements over other single cell alignment methods are even more significant. scDGN also achieves the best performance on the second largest dataset, the PBMC dataset. However, given the very low MI for this dataset the performance of the other methods, including the baseline NN, is almost as good as the performance of scDGN. The third dataset we test on is the Seurat pancreas dataset. This is the smallest dataset and so it has the least number of training samples. Still, of the 6 settings we tested (which differed in the subset of cells that were excluded from training), we find that scDGN is the top performer in 4 of them, comparable to the top performer for another 1 and in only one setting (Pancreas 3, with the highest MI) is significantly outperformed by Seurat. Note that even for the Pancreas 3 data the domain adversarial training helps: using this the scDGN is able to improve by more than 20% over the baseline NN used for the label classifier.

### 3.3 Visualization of the Representation Learned by Alignment and Classification Methods

To further explore the effectiveness of the batch removal provided by our proposed domain adversarial training with conditional domain generalization strategy, we visualize the 100-dimensional hidden representations learned by NN and scDGN: Figure 3 presents both PCA and t-SNE plots for several different cell

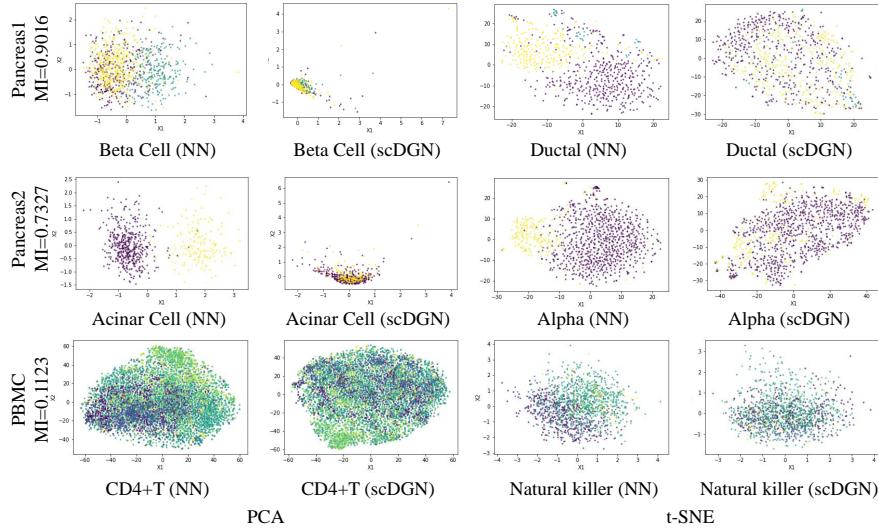


Fig. 3: Visualization of learned representations for NN and scDGN: using PCA and t-SNE Rows: The three datasets we tested the method on. Columns: Methods and cell types. For each row, data from different batches are distinguished using different colors.

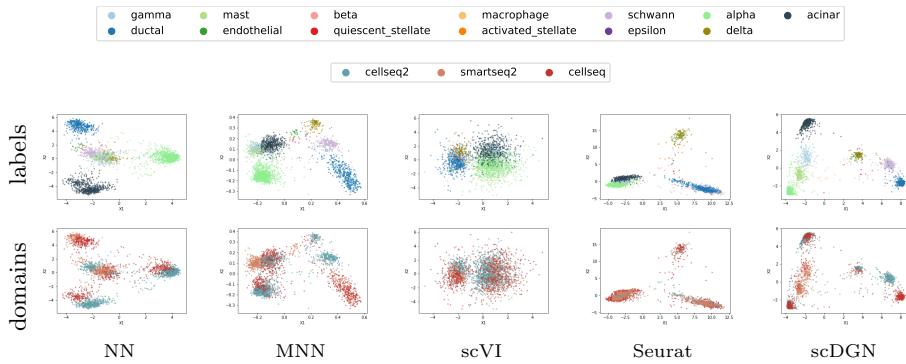


Fig. 4: PCA visualizations of the representations learned by different models on the full Pancreas2 dataset. Colors for different cell types and domains are shown in the legend at the top.

types across the three datasets. Points are colored using their batch IDs in order to evaluate batch effects. As can be seen, using scDGN we obtain results that are much better at mixing cells from the different batches when compared to the baseline NN model. The impact is larger for the pancreas datasets which have larger MI compared to the PBMC dataset, which helps explain the large increase in performance for these two datasets.

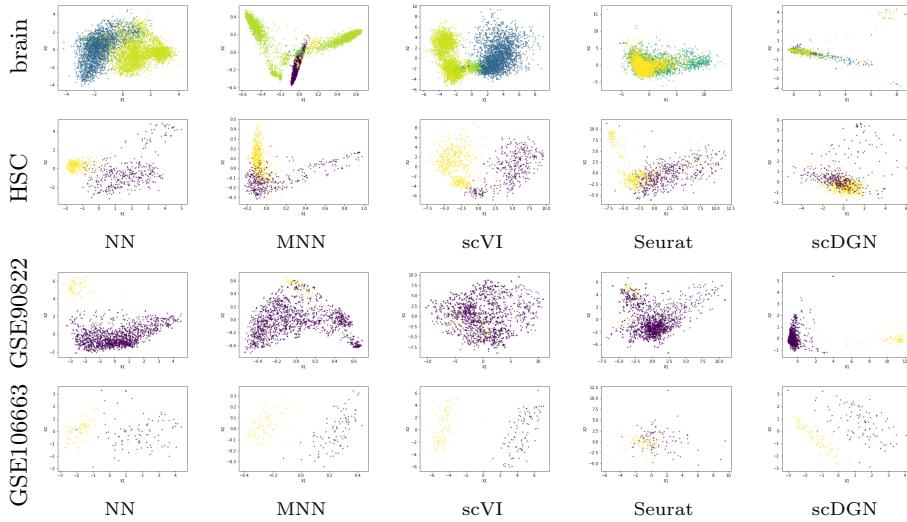


Fig. 5: PCA visualizations of the representations of certain cell types and batches by different models for the scQuery dataset. Top two rows: Cell types. Colors represent different batches. HSC = hematopoietic stem cell. Bottom two rows: Batches. Colors represent different cell types.

We next extended this comparison and visualized the learned (aligned) representations for all methods using data from both the Pancreas2 and scQuery datasets (Figures 4 and 5). For the Pancreas2 dataset, we visualize the entire dataset. For scQuery, given the large number of cell types and domains, we present PCA visualization of a subset of cell types and domains. As can be seen, in addition to scDGN, Seurat is also able to successfully mix the data from different batches. However, as the results in Table 2 indicate this may come at the expense of not correctly separating cell types. MNN and scVI are not always effective at removing batch effects for the cell types. In contrast, scDGN is able to do both domain mixing and cell type assignment, leading to its better performance overall. For example, for the acinar and alpha cell types in the pancreas dataset (Figure 4), only scDGN, MNN, and Seurat are able to align the data from different domains. However, MNN and Seurat over-correct the representation by aligning different cell types from different domains, mixing acinar and gamma cells. Additional visualizations for other cell types and domains can be found in Appendix C, where the same advantages of scDGN over other methods can be consistently observed.

### 3.4 Analysis of Key Genes

While NNs are often treated as black boxes, recent methods provide useful directions for making them more interpretable [30]. Here we use activation maximization, which relies on the gradient of the correct category logit with respect

Table 3: GO analysis results for top 100 scQuery liver genes in the NN method.

term_name	term_id	$p_{adj}$	$-\log_{10} p_{adj}$
symbiotic process	GO:0044403	1.16E-08	7.935246875
interspecies interaction between organisms	GO:0044419	3.14E-08	7.503093471
viral process	GO:0016032	3.69E-08	7.433145019
immune response	GO:0006955	2.5491E-06	5.593613105
multi-organism process	GO:0051704	1.40837E-05	4.851282542
immune effector process	GO:0002252	4.53533E-05	4.34339136
response to stress	GO:0006950	5.56335E-05	4.254663785
defense response	GO:0006952	6.18759E-05	4.208478308

Table 4: GO analysis results for top 100 scQuery liver genes in the scDGN method.

term_name	term_id	$p_{adj}$	$-\log_{10} p_{adj}$
chylomicron remodeling	GO:0034371	3.04042E-05	4.517066786
positive reg. of cholesterol esterification	GO:0010873	3.04042E-05	4.517066786
negative reg. of cellular component organization	GO:0051129	3.94437E-05	4.404022507
protein-lipid complex remodeling	GO:0034368	7.34551E-05	4.133978335
plasma lipoprotein particle remodeling	GO:0034369	7.34551E-05	4.133978335
protein-containing complex remodeling	GO:0034367	8.8522E-05	4.052948555

to the input vector to select the key inputs for each of the models [8, 32, 33]. Formally, given a particular cell type  $i$  and a trained neural network  $\phi$ , activation maximization looks for important input genes  $x'$  by solving the following optimization problem:

$$x' = \max_x(\phi(x) \cdot e_i),$$

where  $e_i$  is the natural basis vector associated with the  $i$ -th category. This can be solved through backpropagation, where the gradient of  $\phi(x)$  with respect to  $x$ , which can be viewed as the weight of the first-order Taylor expansion of the neural network, are calculated to iteratively update the input. We follow a previous method [32] and initialize the optimization with a zero vector. Given this setting, we ran the optimization for 100 iterations with learning rate set to 1. The important genes are selected as those inputs leading to the largest changes compared with the initialization values. To compare scDGN and NN for certain cell types, we select the top  $k$  genes with the largest changes and perform GO analysis on these selected genes.

As an example, consider the genes identified for the liver cell type using the scQuery dataset. We select the top 100 genes for this cell type from NN and scDGN and present the enriched GO categories on Biological Process with adjusted p-value  $< 1.0 \times 10^{-4}$  in Tables 3 and 4. We also list these genes by order in Appendix D.1. As can be seen, while a number of significant GO categories

are identified for the top 100 NN genes, these are generic and not liver specific. They include general terms related to interactions between organs and immune response categories that are active in multiple organs and cell types. In sharp contrast, the categories identified for scDGN are much more specific and highlight key pathways that are mainly utilized in the liver. For example, the top category for the scDGN genes, “chylomicron remodeling”, refers to the main physiological purpose of chylomicron remnants: to facilitate the return of bile lipoproteins and cholesterol to the liver [29]. Specifically, in this pathway chylomicrons (lipoproteins) are broken down (remodeled via hydrolysis) and converted to a form called “chylomicron remnant” that is taken up by specific receptors that exist primarily on the surface of liver cells [12]. The second term, “pos. regulation of cholesterol esterification” refers to cholesterol esterification, a critical step in reverse cholesterol transport, the process in which excess cholesterol is sent to the liver to be removed from the body [15, 25]. Furthermore, Cholesteryl Ester Transfer Protein (CETP) is a key enzyme involved in this process and is highly expressed in liver cells, and variants of CETP are associated with increased risk of atherosclerosis [31, 15]. The fifth most significant term, “lipoprotein remodeling” is part of the two aforementioned processes. The top 100 genes identified by the scDGN include *apoa1* (main protein component of High-Density Lipoprotein cholesterol), *apoa2*, and *apoc1*, all of which encode lipoproteins that are primarily expressed in the liver [6, 18]. These genes were not included in the top 100 genes by the NN. We present the GO analysis results comparison for several additional cell types in Appendix D.

## 4 Discussion

Single cell computational methods that do not account for batch effects are likely to fit the noise introduced by the batches. Several recent methods have been proposed for aligning scRNA-seq from multiple studies of the same tissues or processes. Most of these methods are unsupervised and assume that the cell types among different batches overlap. However, we show that these methods would fail on the studies in which cell types do not fully overlap, which is often the case when dealing with multiple datasets. To overcome this problem we extend a supervised scRNA-seq cell type assignment method based on NN and regularize its prediction to be invariant to batch effects.

Our method is based on the ideas of domain adversarial training. In such training, two competing tasks are used to optimize the representation of scRNA-seq data. The first focuses on the traditional goal of cell type identification while the second attempts to construct representations that are not affected by specific batch or experimental artifacts. This is accomplished by jointly minimizing a loss function that takes into account both goals, accounting for the weight of each of the goals using a gradient reversal layer. We also proposed a conditional strategy to avoid over-correction. We presented efficient learning methods for this setting and tested it on three large scale scRNA-seq datasets containing experiments from several different platforms for partially overlapping cell types.

As we show, our scDGN method is able to correctly identify cell types in the test (unobserved) datasets. For the largest dataset we tested on which contained close to 40 different cell types, scDGN significantly outperformed all prior methods. It also ranked first for the 2nd largest dataset and for all but 1 of the 6 tests on the third dataset. Importantly, it always outperformed the supervised learning based method indicating that batch effects should be addressed when designing such methods for cell type assignments. In addition to accurately assigning cell types, further analysis of significant genes indicates that by overcoming batch effects scDGN is better able to focus on relevant sets of genes for various cell types when compared to prior supervised methods, explaining its improvement in accuracy.

While scDGN performed best on the data we analyzed, there are a number of possible issues with this approach. First, it learns a large number of parameters which require large input datasets. However, as we showed, scDGN is able to perform well even for datasets with a few thousand cells which matches current sizes of scRNA-seq datasets. Second, scDGN is based on NNs which are often seen as a black box, making it hard to interpret the resulting model and its biological relevance. Recent work provides a number of directions that can be used to overcome this issue. As we showed, using activation maximization we were able to identify several relevant cell type specific genes in the learned network. Future work would include using additional NN interpretation methods, including LIME [30] or ROAR and KAR [13], to further identify the set of genes that play the largest role in the decisions the network makes. Third, as shown in Figure C.13, scDGN sometimes does not mix up the representations from different batches for all cell types. Considering the visualization results for NN in Figure C.18 and its competitive performance in Table 2 together, it may indicate that it is not always necessary to remove batch effects for the model to achieve high test accuracy. Therefore, it is worthwhile to further study when the alignment is imperative. Finally, unlike prior scRNA-seq alignment methods scDGN is supervised. While this is an advantage when it comes to accuracy, as we have shown, it may be a problem for the new data. We believe that as more scRNA-seq and other high throughput single cell data accumulate, we would have labeled data for most cell types which would enable training an scDGN for even more cell types. As we have shown with the scQuery dataset, for which scDGN significantly outperformed all other methods, when such data exists scDGN is able to correctly align experiments and platforms not seen in the training set.

scDGN is implemented in Python with the PyTorch API [34] and users can obtain the code and sampled data from <https://github.com/SongweiGe/scDGN>.

## A Experiment Details

### A.1 scQuery Dataset

The studies from which the data are collected for our scQuery dataset are shown as below:

**Training Set:** ERP017366, ERP021445, ERP022096, ERP022251, ERP022289, ERP022298, ERP022654, GSE101487, GSE101984, GSE102159, GSE102332, GSE102346, GSE102455, GSE102456, GSE103268, GSE103892, GSE104156, GSE104396, GSE105054, GSE106447, GSE106472, GSE106663, GSE107115, GSE107122, GSE108097, GSE109774, GSE109796, GSE112033, GSE115070, GSE22182, GSE29087, GSE33979, GSE38198, GSE42268, GSE42704, GSE42706, GSE45719, GSE52564, GSE57403, GSE57609, GSE59114, GSE60066, GSE61346, GSE61844, GSE62952, GSE64959, GSE64960, GSE65160, GSE66343, GSE66390, GSE66582, GSE68981, GSE69761, GSE69970, GSE70713, GSE71982, GSE72852, GSE72854, GSE72855, GSE72856, GSE75454, GSE75659, GSE75804, GSE76381, GSE77113, GSE78140, GSE78471, GSE78521, GSE79306, GSE79374, GSE79380, GSE79578, GSE79812, GSE80155, GSE80168, GSE80280, GSE80483, GSE81275, GSE84498, GSE87375, GSE89468, GSE90697, GSE90822, GSE90824, GSE90860, GSE92707, GSE93524, GSE94333, GSE94389, GSE94579, GSE98048, GSE98664, GSE98816, GSE98969, GSE98971, GSE99235, GSE99701, GSE99786, GSE99866.

**Validation Set:** ERP013319, ERP022703, GSE100120, GSE102163, GSE103267, GSE107053, GSE107527, GSE112642, GSE113043, GSE44183, GSE57393, GSE59127, GSE65924, GSE68770, GSE71585, GSE71794, GSE71802, GSE74534, GSE78401, GSE78510, GSE79108, GSE85234, GSE85627, GSE93421, GSE94388, GSE96981.

**Test Set:** ERP022293, GSE102827, GSE106471, GSE107740, GSE107909, GSE108291, GSE108478, GSE32190, GSE39522, GSE39523, GSE57249, GSE57391, GSE59129, GSE60749, GSE65525, GSE65970, GSE67120, GSE68769, GSE75790, GSE75901, GSE77705, GSE78045, GSE82174, GSE84324, GSE86479, GSE87631, GSE89900, GSE90047, GSE96986, GSE99058.

### A.2 Training Details

For the scQuery, we find that the optimal value of  $\lambda$  is 0.02 and  $m = 17$ . For other datasets we use  $m = 1$  except for Pancreas 3 to which we apply  $m = 3$ . For Pancreas 3 we use  $\lambda = 0.1$  and Pancreas 1 and 2 we use  $\lambda = 0.2$ . For all other datasets we use  $\lambda = 1$ . The models are implemented in PyTorch and trained on a machine with GeForce GTX 980 Ti and 32 GB memory. As for the ScQuery dataset, it takes around 9 and 19 seconds to train NN and scDGN for each epoch respectively.

### A.3 Seurat Pancreas Datasets

The cell type compositions in our pancreas datasets are shown as below:

	CelSeq	CelSeq2	SMART-Seq2	Fluidigm
Pancreas 1	gamma, ductal, mast, endothelial, beta, quiescent_stellate, macrophage	gamma, ductal, mast, epsilon, alpha, delta, acinar	macrophage, activated_stellate, schwann, epsilon, alpha, delta, acinar	All
Pancreas 2	macrophage, activated_stellate, schwann, epsilon, alpha, delta, acinar	gamma, ductal, mast, endothelial, beta, quiescent_stellate, macrophage	gamma, ductal, mast, epsilon, alpha, delta, acinar	All
Pancreas 3	gamma, ductal, mast, epsilon, alpha, delta, acinar	macrophage, activated_stellate, schwann, epsilon, alpha, delta, acinar	gamma, ductal, mast, endothelial, beta, quiescent_stellate, macrophage	All
Pancreas 4	gamma, ductal, mast, epsilon, alpha, delta, acinar	gamma, ductal, mast, endothelial, beta, quiescent_stellate, macrophage	macrophage, activated_stellate, schwann, epsilon, alpha, delta, acinar	All
Pancreas 5	macrophage, activated_stellate, schwann, epsilon, alpha, delta, acinar	gamma, ductal, mast, epsilon, alpha, delta, acinar	gamma, ductal, mast, endothelial, beta, quiescent_stellate, macrophage	All
Pancreas 6	gamma, ductal, mast, endothelial, beta, quiescent_stellate, macrophage	macrophage, activated_stellate, schwann, epsilon, alpha, delta, acinar	gamma, ductal, mast, epsilon, alpha, delta, acinar	All

## B Per Cell-type Performance

Fig. B.1: Test Accuracy of each model on different cell types from scquery dataset. The darker color represents the better performance. Note that the cell types that are not contained in the test dataset are not shown in this Table. The value is calculated by 10 experiments with different initializations for NN-based model to alleviate the randomness. The average is weighted by the number of the test samples. Note that scDGN performs the best on average.

# train	# test	Cell Type	NN	CaSTLe	MNN	scVI	Seurat	scDGN
946	91	hematopoietic stem cell	0.2516	0.736264	0.042879	0.3275	0	0.3593
14	401	type B pancreatic cell	0.08	0.014963	0	0	/	0.0688
128	11	B cell	0	1	0	0.5	0	0
106	40	blastoderm cell	0.89	0.45	0	0.2275	0	0.715
956	50	neuron	0.514	0	0.076078	0.059	0	0.534
525	45	hematopoietic cell	0.0289	0	0.016993	0.0056	0	0
3	516	embryonic cell	0	0	0	0	/	0
3202	1105	embryonic stem cell	0.3707	0.245249	0.143479	0.1867	0.3705	0.4396
1	49	mesenchymal cell	0	0	0	0	/	0
3598	11	kidney cell	0.1364	0	0	0.5	0	0.2182
13	423	testis	0	0	0	0	/	0
6	6	tissue	0	0	0	0	/	0
152	6	epithelium	0	0	0	0	0	0
542	40	embryo	0.19	0	0.139216	0.3125	0	0.305
6531	251	brain	0.8705	0.729084	0.906588	0.9442	0.1608	0.7988
287	13	pancreas	0.0231	0	0	0	0	0.0308
1825	14	cortex	0.1643	0	0	0.1	0	0.0714
882	8	midbrain	0	0	0	0	0	0
1719	145	lung	0.9779	0.089655	0.953888	0.1145	0.0295	0.9766
256	2	heart left ventricle	0	0	0	0.05	0	0
2876	8	spleen	0.0875	0.25	0	0.0812	0	0.025
1676	529	liver	0.1414	0.035917	0.368917	0.1167	0.0877	0.1989
2888	6	thymus	0.8667	0	0	0.1333	0	0.4333
29132	3770	Average	0.255	0.156	0.2	0.257	0.144	0.286

Fig. B.2: Test Accuracy of each model on different cell types from PBMC dataset.

# train	# test	Cell Type	NN	CaSTLe	MNN	scVI	Seurat	scDGN
4562	288	B	0.9757	0.954861	0.9781	0.975	0.9851	0.9812
4347	602	CD14+monocyte	0.9816	0.988372	0.9822	0.9498	0.9701	0.9867
692	102	CD16+monocyte	0.798	0.745098	0.7961	0.7127	0.7353	0.8294
6749	550	CD4+T	0.8911	0.883636	0.8975	0.7882	0.8833	0.9036
7518	1174	CytotoxicT	0.7886	0.818569	0.7797	0.7485	0.7381	0.794
356	55	Dendritic	0.7509	0.618182	0.7727	0.5909	0.6727	0.7455
186	29	Megakaryocyte	1	1	1	1	1	1
1399	166	Naturalkiller	0.6886	0.674699	0.6771	0.609	0.5982	0.6723
134	26	Plasmacytoiddendritic	0.9615	0.807692	0.9615	0.8462	0.9154	0.9615
25943	2992	Average	0.861	0.865	0.859	0.808	0.83	0.868

Supervised Adversarial Alignment of Single-Cell RNA-seq Data 17

Fig. B.3: Test Accuracy of each model on different cell types from pancreas1 dataset.

# train	# test	Cell Type	NN	CaSTLe	MNN	scVI	Seurat	scDGN
7	1	gamma	0	1	0.2	0.9	0	0
771	36	ductal	0.925	0.833333	0.9472	0.95	0.9917	0.9917
8	3	mast	0	0	0.1333	0.5	0.2667	0
18	18	endothelial	0.2778	0.444444	0.4	0.9056	0.9389	0.2778
191	239	beta	0.7054	0.330544	0.8	0.9079	0.8586	0.9042
161	258	quiescent_stellate	0.714	0.794574	0.776	0.8484	0.8279	0.8678
5	1	macrophage	0.9	1	0.7	1	0	0.3
274	21	activated_stellate	0.9952	1	0.9905	0.8524	0.8381	0.9952
4	5	schwann	0	0	0.02	0.36	0	0
145	16	epsilon	1	0.5625	1	0.6562	1	1
330	25	alpha	0.924	0.64	0.896	0.384	0.16	0.864
22	1	delta	1	1	1	1	0.9	0.3
42	14	acinar	0.4929	0.428571	0.4929	0.8643	0.65	0.4571
1978	638	Average	0.72	0.591	0.785	0.855	0.812	0.856

Fig. B.4: Test Accuracy of each model on different cell types from pancreas2 dataset.

# train	# test	Cell Type	NN	CaSTLe	MNN	scVI	Seurat	scDGN
330	25	gamma	0.832	0.56	0.82	0.716	0.724	0.794
462	21	ductal	1	1	1	1	1	1
323	18	mast	0.95	0.833333	0.9556	0.9833	0.8833	0.8833
21	14	endothelial	0.4786	0.428571	0.4786	0.75	0.0286	0.5
6	3	beta	0	0	0.0333	0	0.3333	0.0333
15	1	quiescent_stellate	1	1	1	1	0	0.9
5	1	macrophage	0.6	1	0.6	0.7	0.1	0.4
1	1	activated_stellate	0.5	1	0.4	1	0	0
327	36	schwann	0.7806	0.805556	0.7639	0.8556	0.1667	0.866
3	5	epsilon	0.16	0	0.18	0.34	1	0
1199	239	alpha	0.8695	0.648536	0.8833	0.777	0.9063	0.9381
74	16	delta	0.975	0.625	0.9687	0.9	1	0.9812
469	258	acinar	0.9647	0.910853	0.9725	0.8	0.8725	0.9667
3235	638	Average	0.891	0.764	0.899	0.852	0.825	0.918

Fig. B.5: Test Accuracy of each model on different cell types from pancreas3 dataset.

# train	# test	Cell Type	NN	CaSTLe	MNN	scVI	Seurat	scDGN
7	1	gamma	0	1	0	0	0	0
771	36	ductal	1	0.888889	1	0.9806	0.8639	1
8	3	mast	0.1333	0	0.0667	0.3333	0	0.0667
18	18	endothelial	0	0.777778	0	0.2722	0.2389	0
191	239	beta	0.4895	0.987448	0.5515	0.8025	0.6837	0.7042
161	258	quiescent_stellate	0.4814	0.965116	0.5574	0.3938	0.8605	0.5806
5	1	macrophage	0	0	0	1	0	0
274	21	activated_stellate	0.9952	0.952381	1	0.8429	0.7095	1
4	5	schwann	0.18	0	0.18	0.46	0	0
145	16	epsilon	0.9937	0.3125	1	0.9437	1	1
330	25	alpha	0.94	0.04	0.944	0.824	0.556	0.928
22	1	delta	1	1	1	1	0	1
42	14	acinar	0.5857	0.285714	0.5429	0.7643	0.4429	0.5429
1978	638	Average	0.545	0.722	0.564	0.651	0.751	0.663

Fig. B.6: Test Accuracy of each model on different cell types from pancreas4 dataset.

# train	# test	Cell Type	NN	CaSTLe	MNN	scVI	Seurat	scDGN
253	25	gamma	0.64	0.48	0.636	0.616	0.692	0.664
502	21	ductal	0.9619	1	0.9571	0.9524	1	0.9524
128	18	mast	0.6333	0.666667	0.6333	0.9444	0.9444	0.7111
5	14	endothelial	0.0643	0.214286	0.0643	0.3429	0	0
1	3	beta	0	0	0	0	0	0
1	1	quiescent_stellate	0	0	0	0	0	0
9	1	macrophage	0.5	1	0.4	0.8	0	0
6	1	activated_stellate	0	1	0	0	0	0
444	36	schwann	0.8972	0.916667	0.9028	0.9694	0.4361	0.9139
6	5	epsilon	0.08	0	0.1	0.56	0.4	0.02
1851	239	alpha	0.9954	0.983264	0.9958	0.9837	0.9326	0.9879
145	16	delta	1	0.6875	1	1	1	1
753	258	acinar	0.9938	0.984496	0.9934	0.943	0.9624	0.993
4104	638	Average	0.927	0.914	0.928	0.925	0.881	0.925

Fig. B.7: Test Accuracy of each model on different cell types from pancreas5 dataset.

# train	# test	Cell Type	NN	CaSTLe	MNN	scVI	Seurat	scDGN
18	1	gamma	1	1	1	1	0	0.2
702	36	ductal	0.9222	0.888889	0.9278	0.9583	0.9083	0.9194
13	3	mast	0.2333	0	0.2	0.6667	0.6667	0.0333
213	18	endothelial	0.9389	0.777778	0.9444	0.95	0.95	0.9167
1008	239	beta	0.9854	0.987448	0.9828	0.9414	0.9222	0.9619
308	258	quiescent_stellate	0.9922	0.965116	0.9926	0.7888	0.9349	0.9907
9	1	macrophage	0.3	0	0.1	0.9	0.1	0
228	21	activated_stellate	0.9524	0.952381	0.9524	0.8952	0.281	0.9714
1	5	schwann	0	0	0	0.02	0	0
109	16	epsilon	0.9187	0.3125	0.9062	0.8375	1	0.975
253	25	alpha	0.352	0.04	0.46	0.604	0.368	0.468
16	1	delta	0.5	1	0.7	1	0	0.7
26	14	acinar	0.3429	0.285714	0.3571	0.5429	0.3643	0.3857
2904	638	Average	0.928	0.882	0.932	0.895	0.865	0.923

Fig. B.8: Test Accuracy of each model on different cell types from pancreas6 dataset.

# train	# test	Cell Type	NN	CaSTLe	MNN	scVI	Seurat	scDGN
330	25	gamma	0.768	0.68	0.772	0.74	0.84	0.824
462	21	ductal	1	1	1	0.9714	1	0.9952
323	18	mast	0.9556	0.833333	0.95	0.9889	1	0.9722
21	14	endothelial	0.3643	0.142857	0.3857	0.5286	0.1214	0.3857
6	3	beta	0	0.333333	0	0	0	0.0333
15	1	quiescent_stellate	1	1	1	1	0.2	1
5	1	macrophage	0	0	0.1	0	0	0
1	1	activated_stellate	0	0	0	0	0	0
327	36	schwann	0.8944	0.75	0.9028	0.8889	0.5083	0.9583
3	5	epsilon	0.06	0	0.06	0.36	0.38	0
1199	239	alpha	0.9824	0.979079	0.9812	0.9774	0.9682	0.9791
74	16	delta	1	0.75	1	0.825	0.975	1
469	258	acinar	0.9922	0.988372	0.9942	0.8678	0.9628	0.9922
3235	638	Average	0.944	0.917	0.946	0.893	0.907	0.95

## C Representation Visualization

### C.1 scQuery

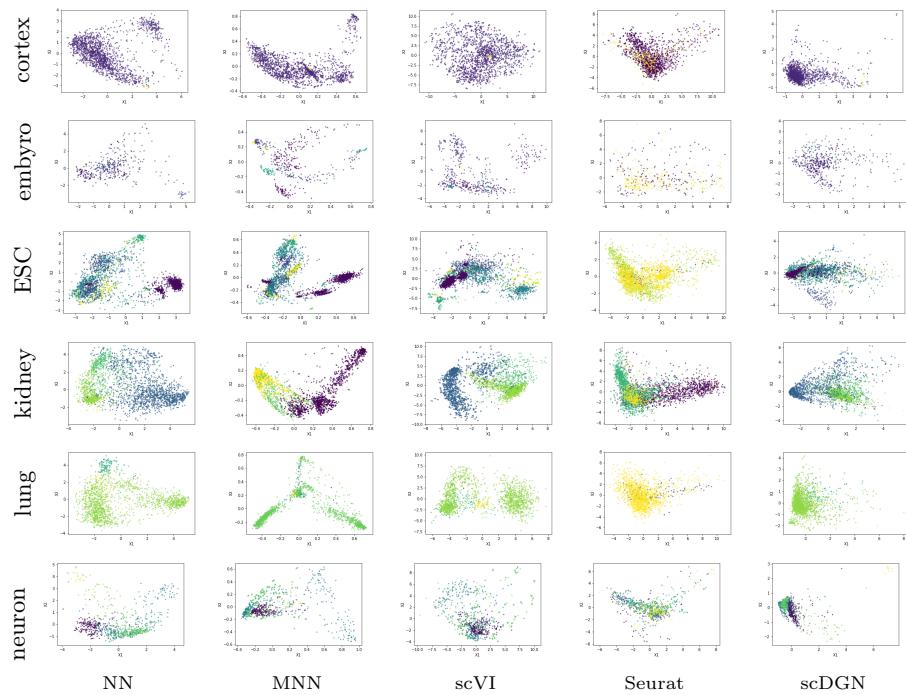


Fig. C.9: PCA visualization of the representations learned by different models on scQuery dataset for certain cell types. The colors are used to distinguish the batches.

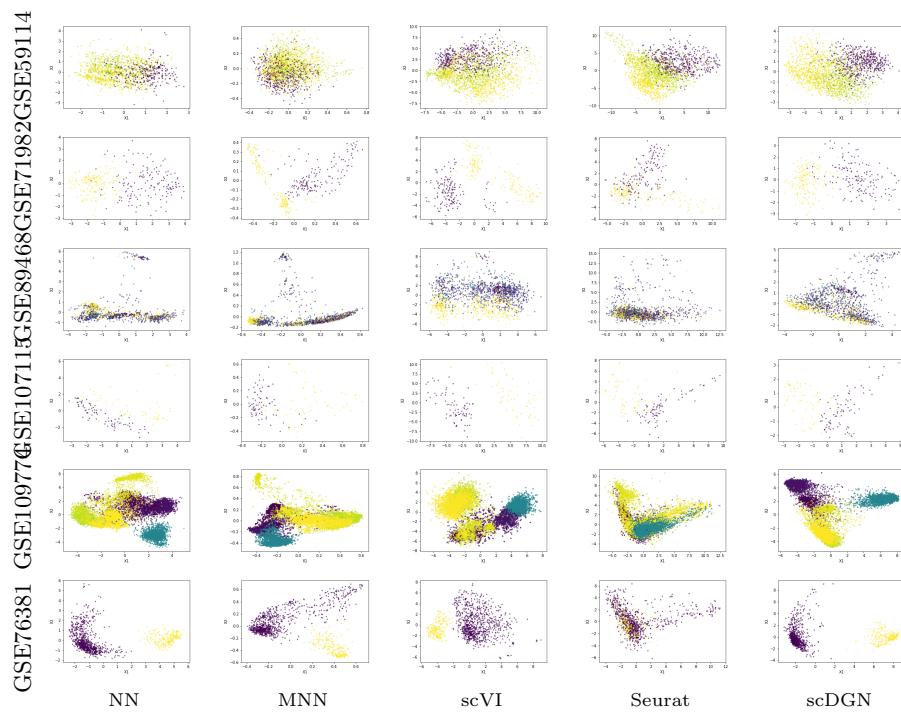


Fig. C.10: PCA visualization of the representations learned by different models on sc-Query dataset for certain batches. The colors are used to distinguish the cell types.

## C.2 Full Pancreas Datasets

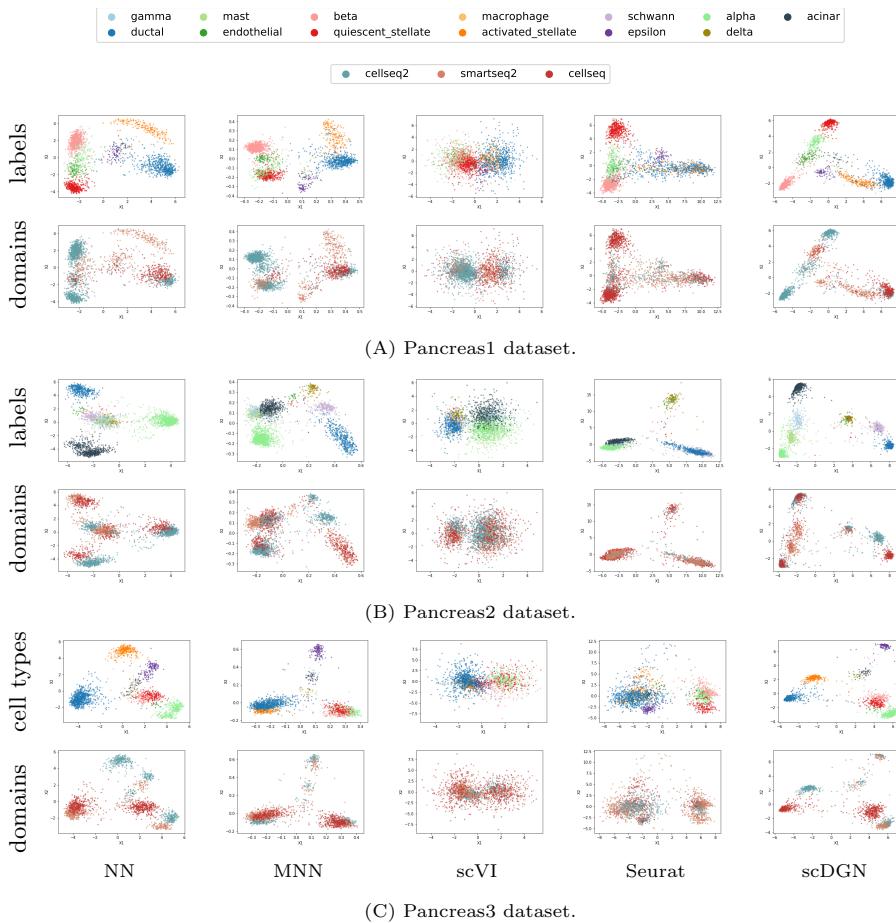


Fig. C.11: PCA visualization of the representations learned by different models on the whole Pancreas datasets.

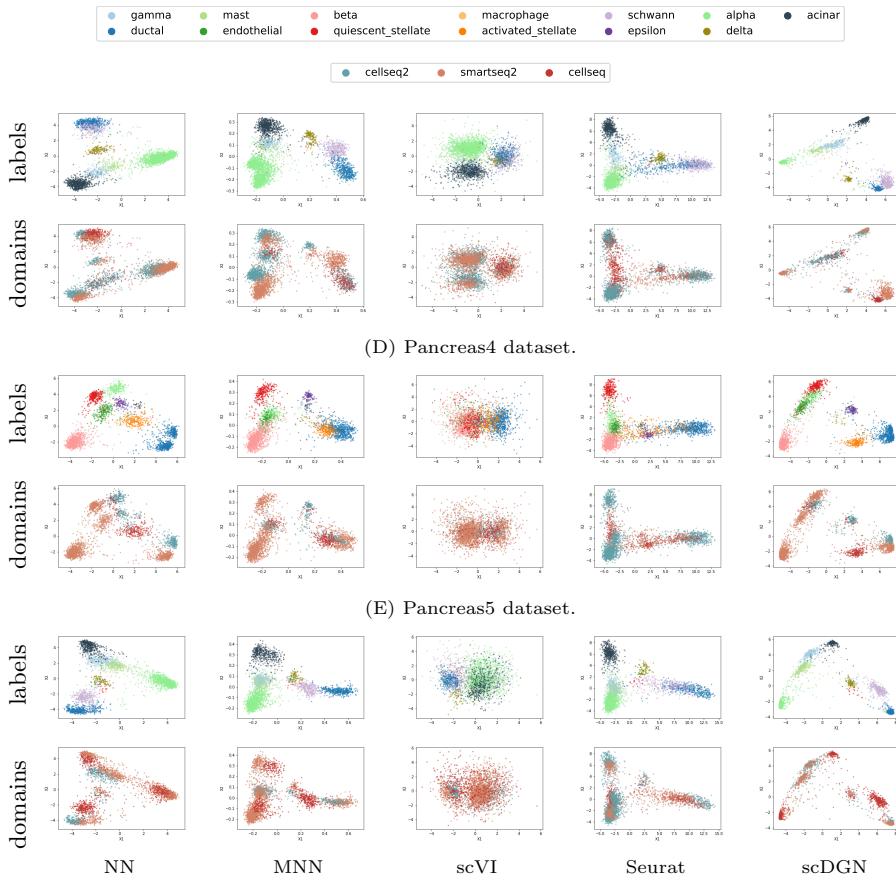
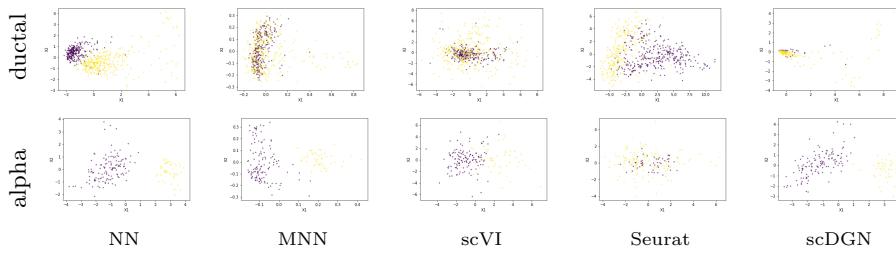
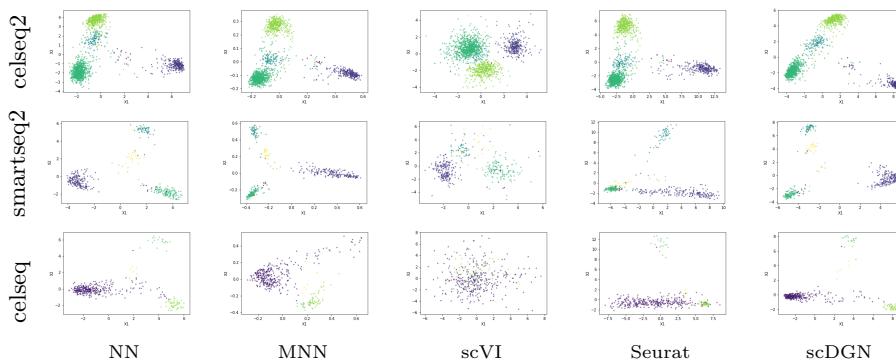


Fig. C.12: PCA visualization of the representations learned by different models on the Pancreas datasets.

### C.3 Pancreas1



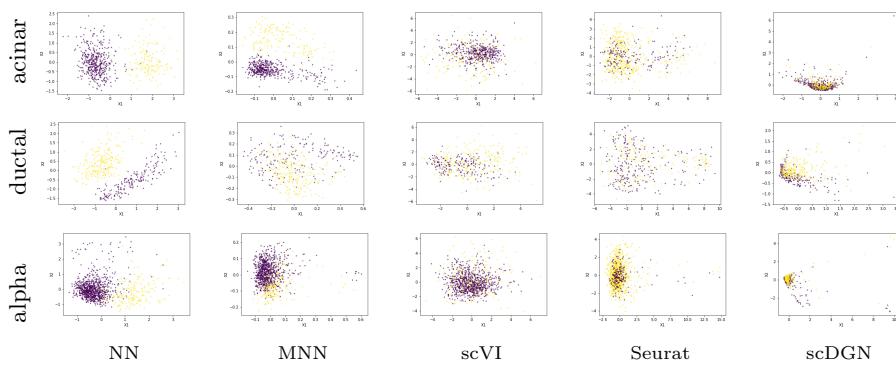
(A) Visualization of representation for certain cell types. The colors are used to distinguish the batches.



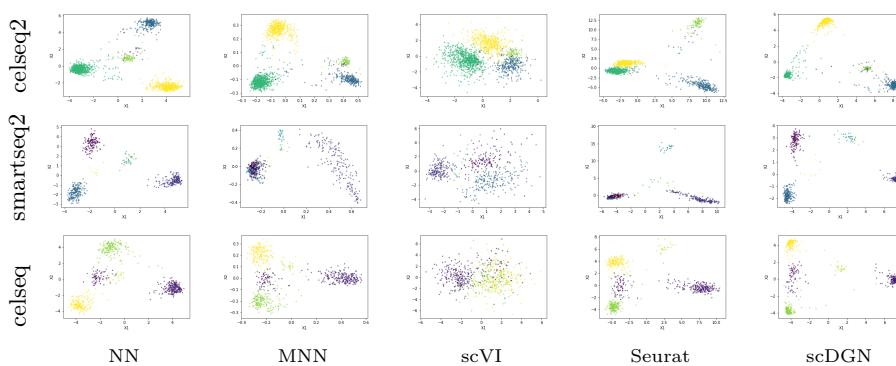
(B) Visualization of representation for certain batches. The colors are used to distinguish the cell types.

Fig. C.13: PCA visualization of the representations for certain cell types and domains on Pancreas1 dataset.

#### C.4 Pancreas2



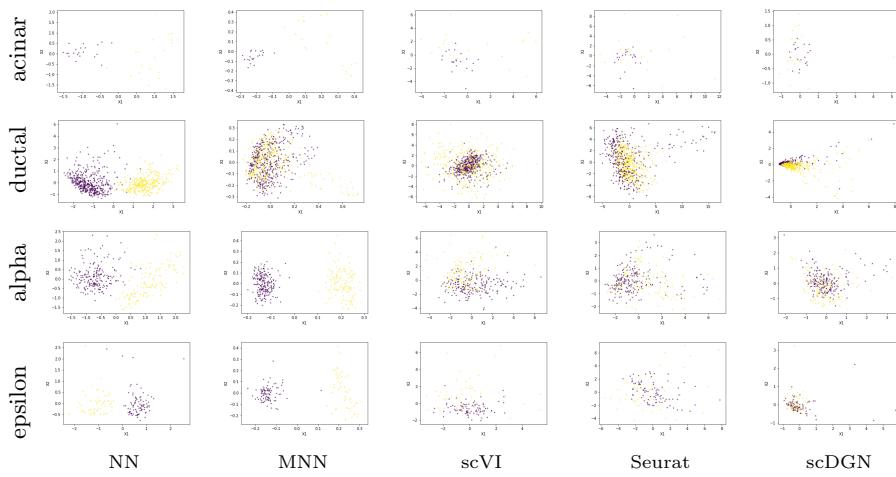
(A) Visualization of representation for certain cell types. The colors are used to distinguish the batches.



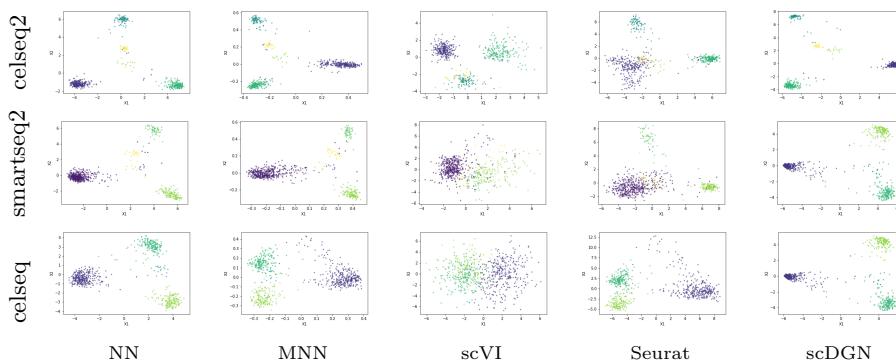
(B) Visualization of representation for certain batches. The colors are used to distinguish the cell types.

Fig. C.14: PCA visualization of the representations for certain cell types and domains on Pancreas2 dataset.

### C.5 Pancreas3



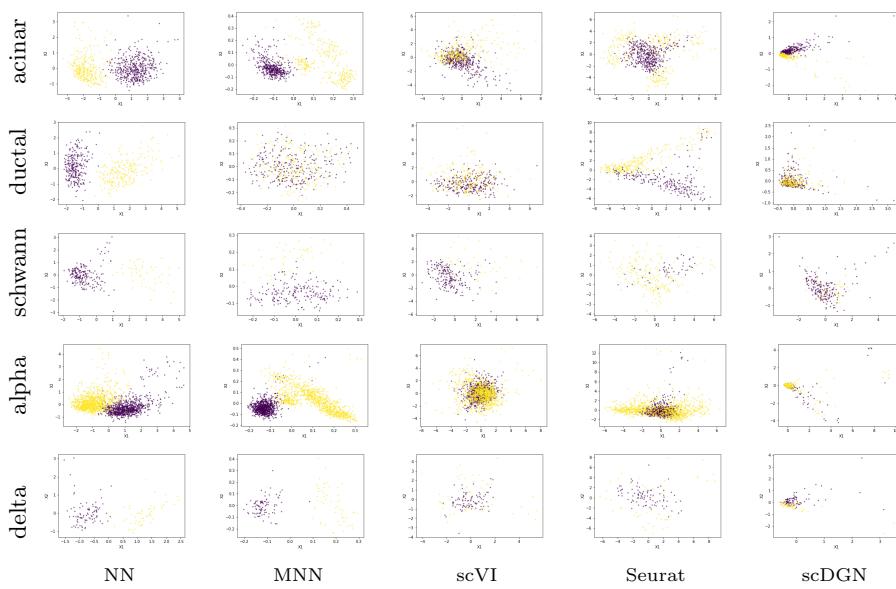
(A) Visualization of representation for certain cell types. The colors are used to distinguish the batches.



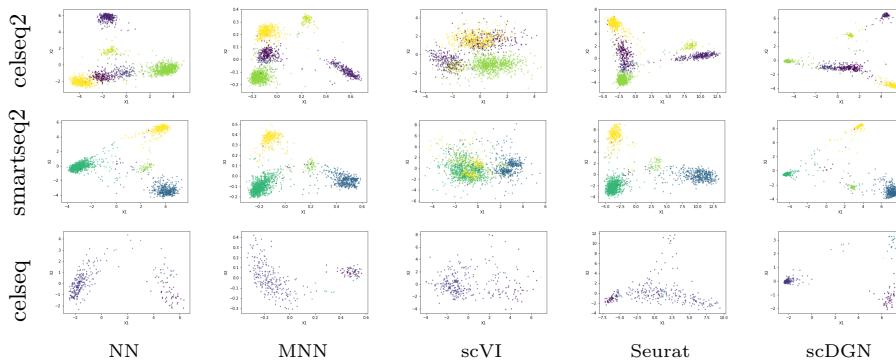
(B) Visualization of representation for certain batches. The colors are used to distinguish the cell types.

Fig. C.15: PCA visualization of the representations for certain cell types and domains on Pancreas3 dataset.

### C.6 Pancreas4



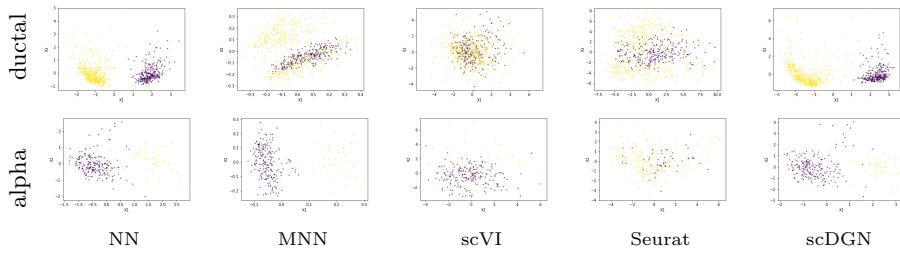
(A) Visualization of representation for certain cell types. The colors are used to distinguish the batches.



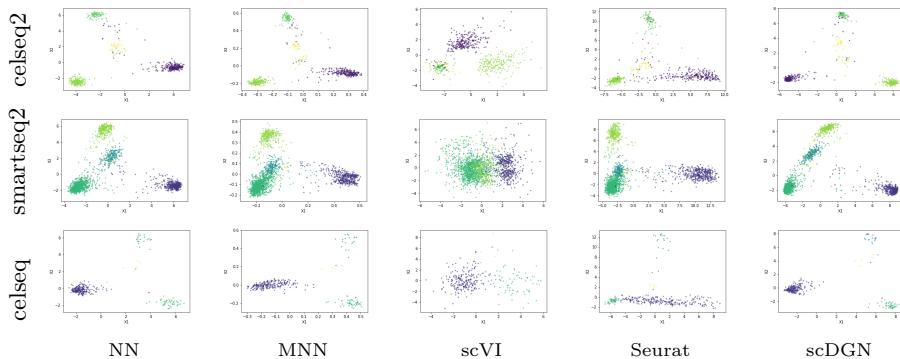
(B) Visualization of representation for certain batches. The colors are used to distinguish the cell types.

Fig. C.16: PCA visualization of the representations for certain cell types and domains on Pancreas4 dataset.

### C.7 Pancreas5



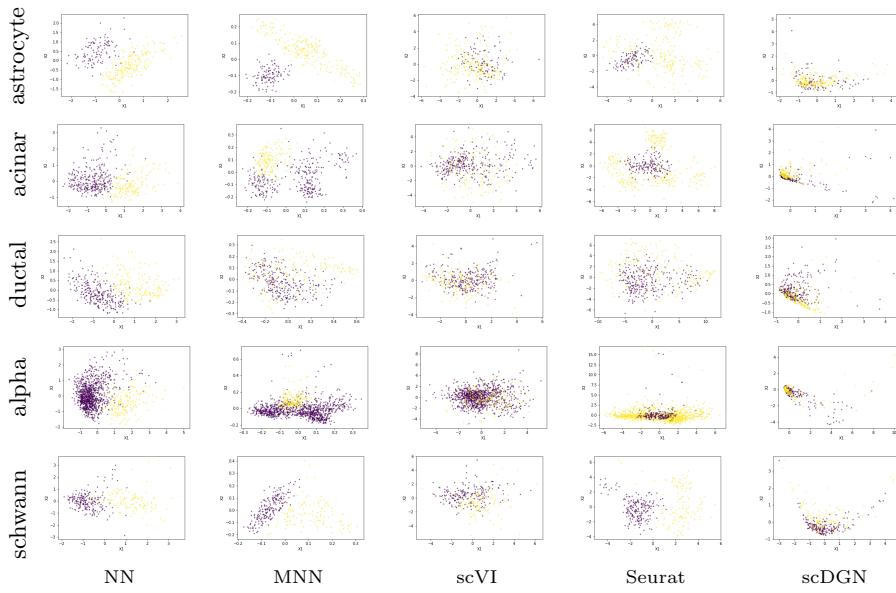
(A) Visualization of representation for certain cell types. The colors are used to distinguish the batches.



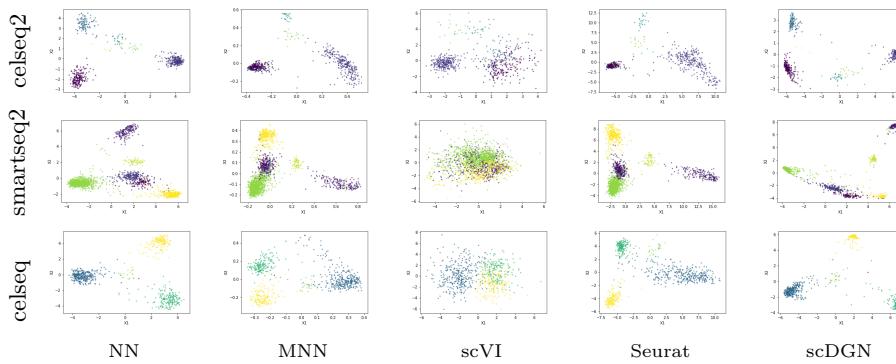
(B) Visualization of representation for certain batches. The colors are used to distinguish the cell types.

Fig. C.17: PCA visualization of the representations for certain cell types and domains on Pancreas5 dataset.

### C.8 Pancreas6



(A) Visualization of representation for certain cell types. The colors are used to distinguish the batches.



(B) Visualization of representation for certain batches. The colors are used to distinguish the cell types.

Fig. C.18: PCA visualization of the representations for certain cell types and domains on Pancreas6 dataset.

## D Key gene analysis

### D.1 Top 100 Genes for Liver Cell Type<sup>2</sup>

- **NN:** *ubb, ins2, apoe, hba-a1, actb, b2m, hba-a2, mup11, cd74, dnase1l3, h2-d1, mup22, tuba1a, tmsb4x, mup14, selenop, aw112010, ins1, cst3, mup7, iapp, ccdc152, ftl1, mup13, mup10, chl1, hbb-bt, clec4g, igfbp7, nrep, mup19, sh3bgrl3, hbb-bs, mup1, mup18, h2-ab1, mup16, mup12, rpl18a, c1qb, map1lc3b, mup15, ccl5, hbb-y, rps29, ctsd, ly86, ifitm2, mup9, h3f3b, mup8, nkg7, tmsb10, mt1, tagln2, ttr, gstm1, hba-x, atpif1, lyz2, apoc1, lgals1, mup2, stmn2, apoa2, plp1, calm1, gpx3, prdx1, igfbp4, prl3d1, c1qa, apod, tma7, h2-aa, cmss1, ptma, cd79a, gapdh, plpp1, sst, abi1, dennd1b, cft1, rgs1, gclm, pitpnc1, dppa3, fcgr2b, serp1, rpl8, tyrobp, sumo1, zfp976, rplp1, gm10591, gm21541, rpsa, s100a8, pcnp.*
- **scDGN:** *rps29, apoe, mup11, ins2, mup22, dnase1l3, selenop, mup10, b2m, gcg, mup7, h2-d1, mup14, tmsb10, ccdc152, clec4g, apoa2, hba-a1, ttr, mup13, myl7, mgp, mup18, mup16, mt1, mup15, npm1, mup19, hba-a2, aw112010, igfbp7, chl1, mup1, actg1, apoc1, ins1, ly6e, mup9, mup8, ptma, mup12, mup2, cst3, fabp3, btg1, iapp, rpl35, apoa1, h2-k1, cmss1, mup17, lgals1, tmsb4x, stmn1, gm13304, ptp4a2, prdx1, gm21541, hmga1, snap25, set, plp1, ccl4, fabp4, trim30a, gstp1, gm10591, ubc, scgb1a1, resp18, sumo1, fabp1, nrep, h2-q7, npy, itm2b, hspe1, car2, sub1, slc25a5, h2afz, ywhah, ccl21b, pomic, rpl41, cbx3, ctsd, rps27rt, laptm5, chchd2, s100a8, actc1, hba-x, hbb-bt, myl4, eif1, gpihbp1, sod1, gabarapl2, calm1.*

<sup>2</sup> The results in text format are also available at: [https://github.com/SongweiGe/scDGN/blob/master/supplementary\\_materials/top\\_genes.txt](https://github.com/SongweiGe/scDGN/blob/master/supplementary_materials/top_genes.txt)

## D.2 GO Analysis Results for the Other Cell Types

Table 5: GO analysis results of the genes with respect to embryonic stem cell that are only recognized by scDGN.

term_name	term_id	$p_{adj}$	$-\log_{10} p_{adj}$
SRP-dependent cotranslational protein targeting to membrane	GO:0006614	1.8109E-06	5.74210612
cotranslational protein targeting to membrane	GO:0006613	2.59394E-06	5.58603936
protein targeting to ER	GO:0045047	4.44101E-06	5.35251780
establishment of protein localization to endoplasmic reticulum	GO:0072599	5.72221E-06	5.24243629
nuclear-transcribed mRNA catabolic process, nonsense-mediated decay	GO:0000184	9.79366E-06	5.00905515
protein localization to endoplasmic reticulum	GO:0070972	2.19947E-05	4.65768238

Table 6: GO analysis results of the genes with respect to hematopoietic stem cell that are only recognized by scDGN.

term_name	term_id	$p_{adj}$	$-\log_{10} p_{adj}$
protein targeting to ER	GO:0045047	0.041444862	1.382529302
cotranslational protein targeting to membrane	GO:0006613	0.026817903	1.571575182
SRP-dependent cotranslational protein targeting to membrane	GO:0006614	0.020030591	1.698306246
protein localization to endoplasmic reticulum	GO:0070972	0.013637519	1.865264616
translational initiation	GO:0006413	0.001555052	2.808254977

## References

1. Integration and label transfer - standard workflow (Oct 2019), [#standard-workflow](https://satijalab.org/seurat/v3.1/integration.html)
2. Alavi, A., Ruffalo, M., Parvangada, A., Huang, Z., Bar-Joseph, Z.: A web server for comparative analysis of single-cell rna-seq data. *Nature communications* **9**(1), 4768 (2018)
3. Chu, C., Wang, R.: A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258* (2018)
4. Csurka, G.: Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374* (2017)
5. Ding, J., Adiconis, X., Simmons, S.K., Kowalczyk, M.S., Hession, C.C., Marjanovic, N.D., Hughes, T.K., Wadsworth, M.H., Burks, T., Nguyen, L.T., et al.: Systematic comparative analysis of single cell rna-sequencing methods. *BioRxiv* p. 632216 (2019)
6. Domingo-Espín, J., Nilsson, O., Bernfur, K., Giudice, R.D., Lagerstedt, J.O.: Site-specific glycanations of apolipoprotein a-i lead to differentiated functional effects on lipid-binding and on glucose metabolism. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease* **1864**(9, Part B), 2822 – 2834 (2018). [https://doi.org/https://doi.org/10.1016/j.bbadi.2018.05.014](https://doi.org/10.1016/j.bbadi.2018.05.014), <http://www.sciencedirect.com/science/article/pii/S0925443918301844>
7. Eng, C.H.L., Lawson, M., Zhu, Q., Dries, R., Koulena, N., Takei, Y., Yun, J., Cronin, C., Karp, C., Yuan, G.C., et al.: Transcriptome-scale super-resolved imaging in tissues by rna seqfish+. *Nature* **568**(7751), 235 (2019)
8. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. *University of Montreal* **1341**(3), 1 (2009)
9. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* **17**(1), 2096–2030 (2016)
10. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. vol. 2, pp. 1735–1742. IEEE (2006)
11. Haghverdi, L., Lun, A.T., Morgan, M.D., Marioni, J.C.: Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology* **36**(5), 421 (2018)
12. Hara, T., Tan, Y., Huang, L.: In vivo gene delivery to the liver using reconstituted chylomicron remnants as a novel nonviral vector. *Proceedings of the National Academy of Sciences* **94**(26), 14547–14552 (1997)
13. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: Evaluating feature importance estimates. *arXiv preprint arXiv:1806.10758* (2018)
14. Hwang, B., Lee, J.H., Bang, D.: Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine* **50**(8), 1–14 (2018)
15. Inazu, A.: Chapter 3 - plasma cholestryl ester transfer protein (cetp) in relation to human pathophysiology. In: Komoda, T. (ed.) *The HDL Handbook*, pp. 35 – 59. Academic Press, Boston (2010). <https://doi.org/https://doi.org/10.1016/B978-0-12-382171-3.10003-8>, <http://www.sciencedirect.com/science/article/pii/B9780123821713100038>
16. Jaitin, D.A., Kenigsberg, E., Keren-Shaul, H., Elefant, N., Paul, F., Zaretsky, I., Mildner, A., Cohen, N., Jung, S., Tanay, A., et al.: Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. *Science* **343**(6172), 776–779 (2014)

17. Kiselev, V.Y., Yiu, A., Hemberg, M.: scmap: projection of single-cell rna-seq data across data sets. *Nature methods* **15**(5), 359 (2018)
18. Ko, H.L., Wang, Y.S., Fong, W.L., Chi, M.S., Chi, K.H., Kao, S.J.: Apolipoprotein c1 (apoc 1) as a novel diagnostic and prognostic biomarker for lung cancer: A marker phase i trial. *Thoracic cancer* **5**(6), 500–508 (2014)
19. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop. vol. 2 (2015)
20. Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR) (2018)
21. Lieberman, Y., Rokach, L., Shay, T.: 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* **13**(10), e0205499 (2018)
22. Lin, C., Jain, S., Kim, H., Bar-Joseph, Z.: Using neural networks for reducing the dimensions of single-cell rna-seq data. *Nucleic acids research* **45**(17), e156–e156 (2017)
23. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., Yosef, N.: Deep generative modeling for single-cell transcriptomics. *Nature methods* **15**(12), 1053 (2018)
24. Motiian, S., Piccirilli, M., Adjero, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: The IEEE International Conference on Computer Vision (ICCV). vol. 2, p. 3 (2017)
25. Murakami, T., Michelagnoli, S., Longhi, R., Gianfranceschi, G., Pazzucconi, F., Calabresi, L., Sirtori, C.R., Franceschini, G.: Triglycerides are major determinants of cholesterol esterification/transfer and hdl remodeling in human plasma. Arteriosclerosis, thrombosis, and vascular biology **15**(11), 1819–1828 (1995)
26. Papalexis, E., Satija, R.: Single-cell rna sequencing to explore immune cell heterogeneity. *Nature Reviews Immunology* **18**(1), 35 (2018)
27. Patel, V.M., Gopalan, R., Li, R., Chellappa, R.: Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine* **32**(3), 53–69 (2015)
28. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: AAAI Conference on Artificial Intelligence (2018)
29. Redgrave, T.: Chylomicron metabolism. *Biochemical Society Transactions* **32**(1), 79–82 (02 2004). <https://doi.org/10.1042/bst0320079>, <https://doi.org/10.1042/bst0320079>
30. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you?: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144. ACM (2016)
31. Seidman, M.A., Mitchell, R.N., Stone, J.R.: Chapter 12 - pathophysiology of atherosclerosis. In: Willis, M.S., Homeister, J.W., Stone, J.R. (eds.) *Cellular and Molecular Pathobiology of Cardiovascular Disease*, pp. 221 – 237. Academic Press, San Diego (2014). <https://doi.org/https://doi.org/10.1016/B978-0-12-405206-2.00012-0>, <http://www.sciencedirect.com/science/article/pii/B9780124052062000120>
32. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013)
33. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014)

34. Steiner, B., DeVito, Z., Chintala, S., Gross, S., Paszke, A., Massa, F., Lerer, A., Chanan, G., Lin, Z., Yang, E., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* **32** (2019)
35. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W.M., Hao, Y., Stoeckius, M., Smibert, P., Satija, R.: Comprehensive integration of single-cell data. *Cell* (2019)
36. Stuart, T., Satija, R.: Integrative single-cell analysis. *Nature Reviews Genetics* p. 1 (2019)
37. Tung, P.Y., Blischak, J.D., Hsiao, C.J., Knowles, D.A., Burnett, J.E., Pritchard, J.K., Gilad, Y.: Batch effects and the effective design of single-cell gene expression studies. *Scientific reports* **7**, 39921 (2017)
38. Villani, A.C., Satija, R., Reynolds, G., Sarkizova, S., Shekhar, K., Fletcher, J., Griesbeck, M., Butler, A., Zheng, S., Lazo, S., et al.: Single-cell rna-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science* **356**(6335), eaah4573 (2017)
39. Wang, G., Moffitt, J.R., Zhuang, X.: Multiplexed imaging of high-density libraries of rnas with merfish and expansion microscopy. *Scientific reports* **8**(1), 4847 (2018)
40. Wang, H., Ge, S., Xing, E.P., Lipton, Z.C.: Learning robust global representations by penalizing local predictive power. *arXiv preprint arXiv:1905.13549* (2019)
41. Yu, Y., Tsang, J.C., Wang, C., Clare, S., Wang, J., Chen, X., Brandt, C., Kane, L., Campos, L.S., Lu, L., et al.: Single-cell rna-seq identifies a pd-1 hi ilc progenitor and defines its development pathway. *Nature* **539**(7627), 102 (2016)
42. Zeisel, A., Muñoz-Manchado, A.B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al.: Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science* **347**(6226), 1138–1142 (2015)