A Deep Learning Model for Batch Effect Correction and Clustering in Single-Cell

JIANG Sheng[1*] *20431767@life.hkbu.edu.hk*

***Abstract*—Large-scale single-cell RNA sequencing datasets generated at different times, different operators and different laboratories including batch effects that may affect Data interpretation and integration. We propose an autoencoder method, which select the Maximal Mean Discrepancy (MMD) as the loss function to remove batch-effects. We apply the model on the simulation dataset and show its ability to eliminate the batch effects in the embedding space accurately.**

Keywords—Single cell, Autoencoder, Batch correct, MMD, Simulation data

BACKGROUND

Batch effects are inevitable because data are usually produced at different instruments, and batches can confuse biological differences. Under the assumption that among different cell types, the change of batch-effect is far less than that of biological-effect. In this paper, we select an autoencoder, a neural network with regularization, which learns to reconstruct the input by learning the low-dimensional hidden layer of data representation. For example, LINGER [9] , The low-dimensional representation data is obtained by using integrative non-negative matrix factorization to identify the similar neighborhoods to accomplish batch correction. However, how to find batch correction vectors. A method of batch correction by identifying MNN(Mutual Nearest Neighbor) [1] pairs has been proposed for batch correction. Firstly, The K-Nearest Neighbor algorithm is used to find out the K-cell distance of each batch based on Euclidean distance, and the intersection of the paired K-cells is regarded as the Mutual Nearest Neighbors.

Then, the weighted average of these MNN pairs is calculated as the MNN Correction Vector and the target batch is corrected by the reference batch.

Randomly select a batch from the batches as the reference batch. By default, the first batch has been loaded as the reference batch. SAUCIE [4]

The maximal mean discrepancy (MMD) correction is proposed. SAUCIE minimizes the MMD to estimate the differences in the distribution of similar cell clusters in different batches. Finally, We apply batch correction to the representation data and MMD as the loss function to remove the batch and cluster the dataset to evaluate the performance of the model.

RESULTS

Imputation and MMD. The common methods to eliminate batch effects include MMD algorithm, CCA algorithm, MNN algorithm and the Seurat 3.0. In our Model, we select MMD as a loss function to eliminate batch-effect. Test the algorithms on the simulation datasets, covering different batch size and dropout. To evaluate the performance of removing batch effects, t-Distributed Stochastic Neighbor Embedding (t-SNE) is used to show the different cell types and different batch effects. Compared with the Raw Data, we can clearly conclude the conclusion that the MMD function has a better performance in removing the batch-effect.

Visualization with 2D t-SNE can draw two batches of simulated scRNA-seq data of two cell types are included. B1 is the sample size of Batch 1, B2 is the sample size of Batch 2. Dropout is the probability of the additional discards to increase the number of zeros in the simulation dataset. We test our model under two parameter:

L2-Normalization is used to normalize the latent

*Correspondence: 20431767@life.hkbu.edu.hk

[1]Department of Computer Science, Hong Kong Baptist University, HK

space to avoid overfitting and the parameter of inherit centroid is used to pre cluster the centroid of kmeans to obtain better predictions results.

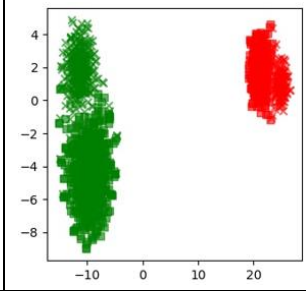
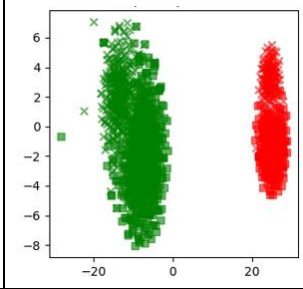
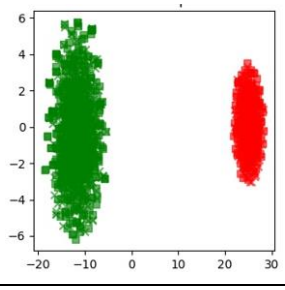
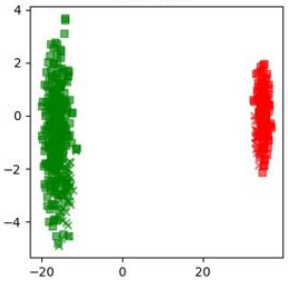
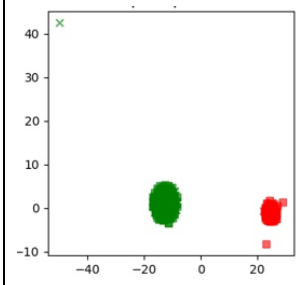
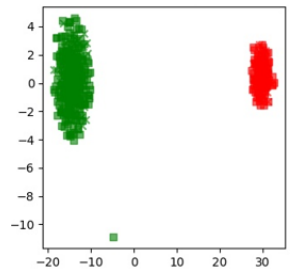
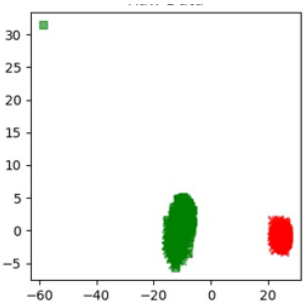
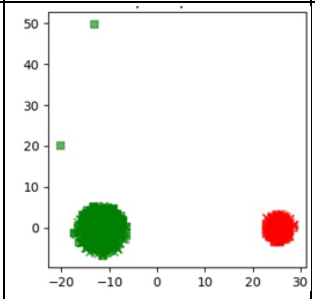
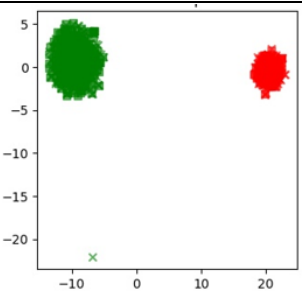
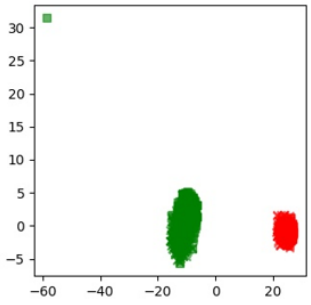
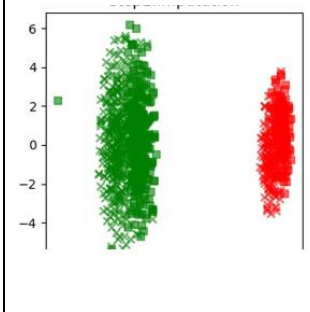
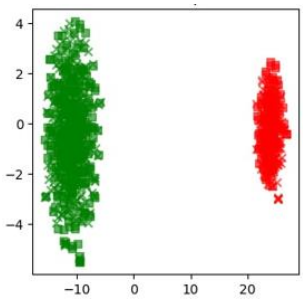
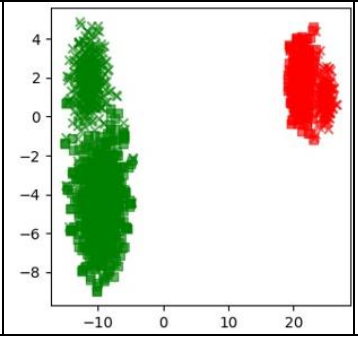
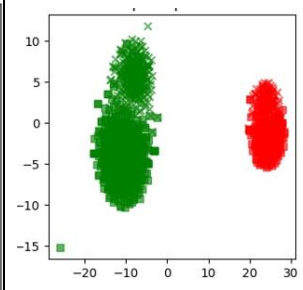
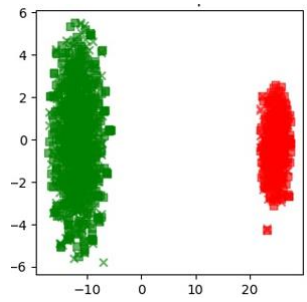
Dropout		Raw_Data	Step1:Imputation	Step2:MMD
B1:800 B2:400	025			
	005			
B1:500 B2:450	025			
	005			
B1:500 B2:900	025			

Fig.1 without L2-Normalization and Inherit centroids.

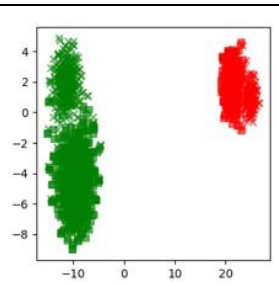
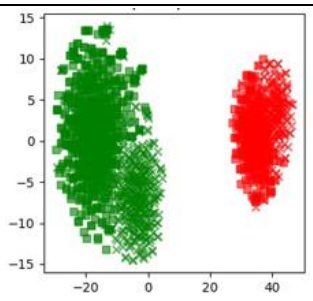
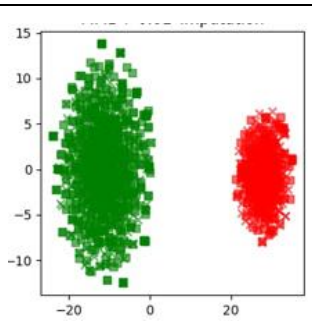
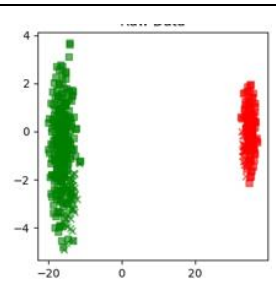
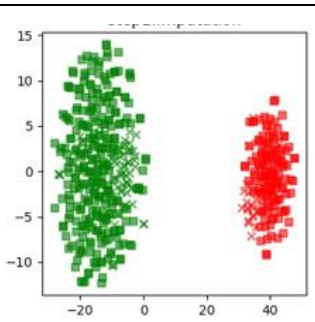
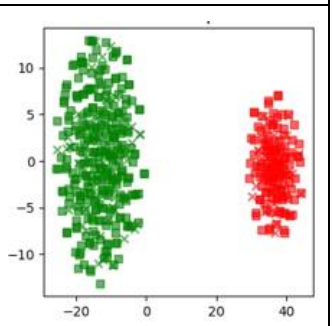
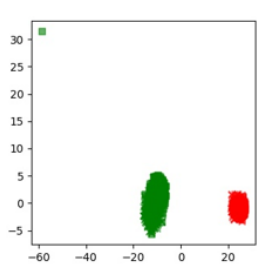
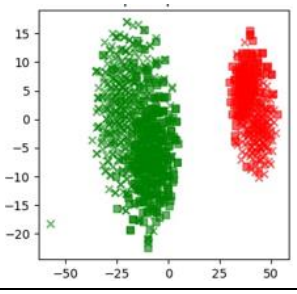
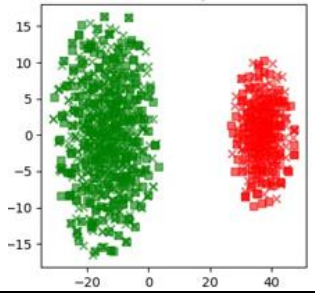
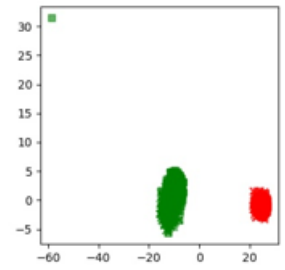
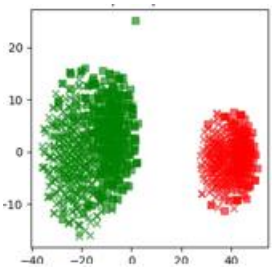
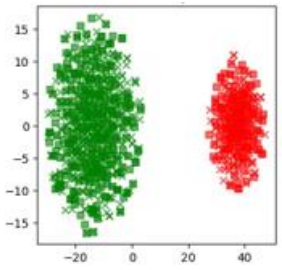
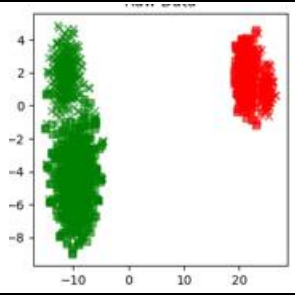
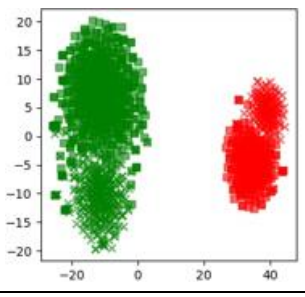
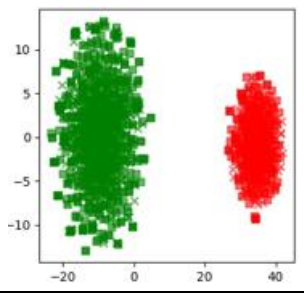
Dropout		Raw_Data	Step1:Imputation	Step2:MMD
B1: 800 B2: 400	025			
	005			
B1: 500 B2: 450	025			
	005			
B1: 500 B2: 900	025			

Fig.2 with L2-Normalization

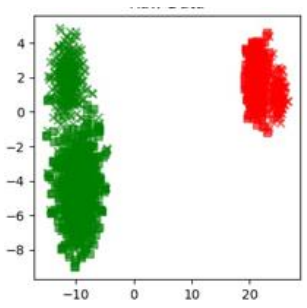
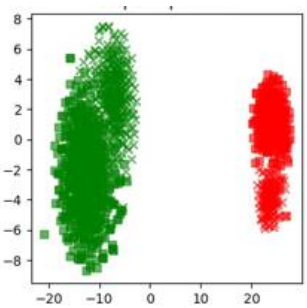
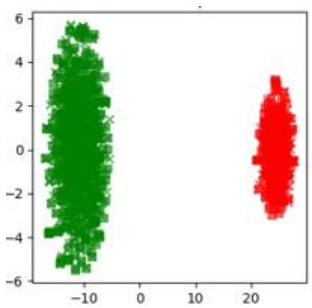
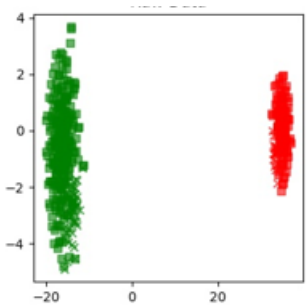
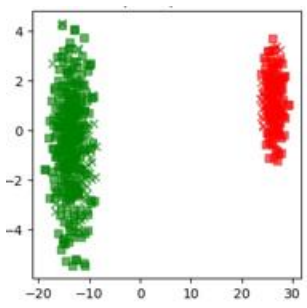
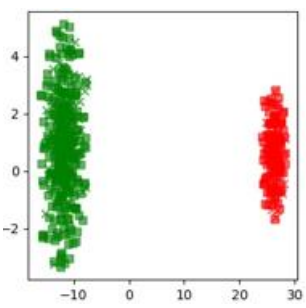
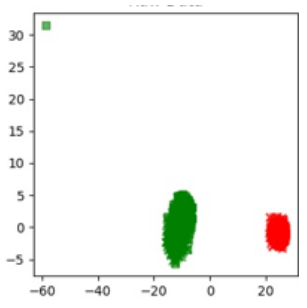
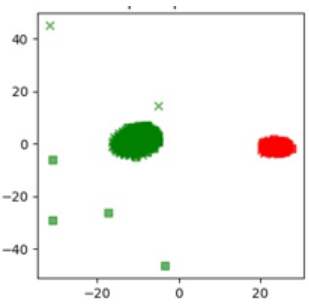
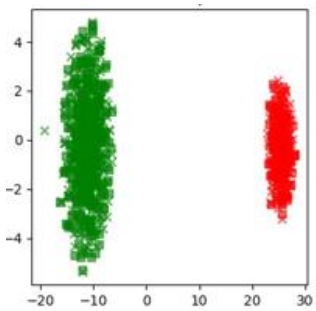
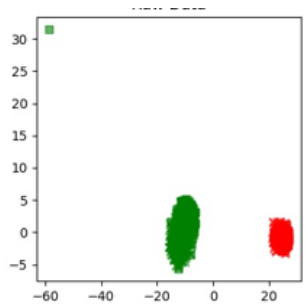
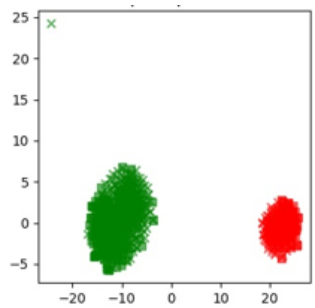
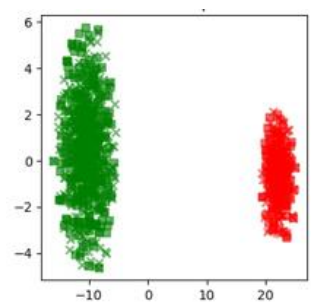
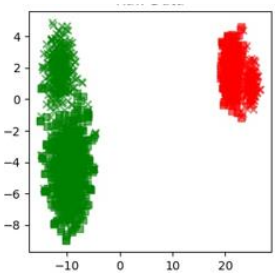
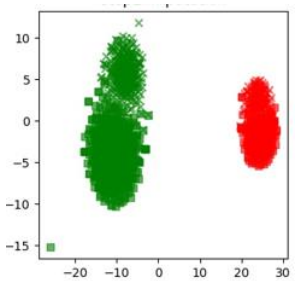
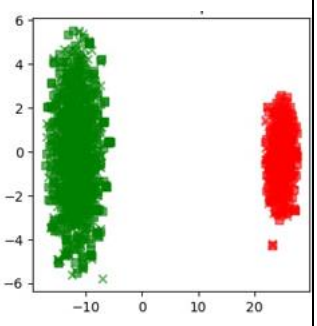
Dropout:		Raw_Data	Step1:Imputation	Step2:MMD
B1:800 B2:400	025			
	005			
B1:500 B2:450	025			
	005			
B1:500 B2:900	025			

Fig.3 with Inherit centroids

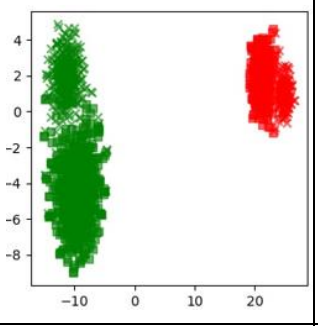
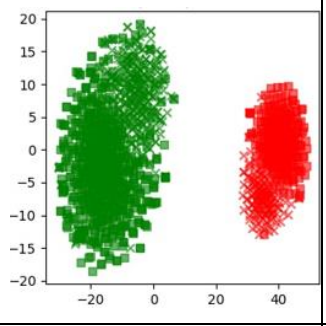
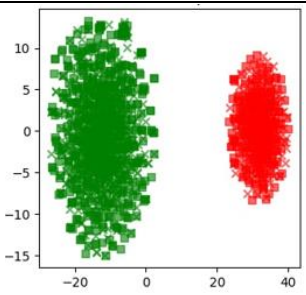
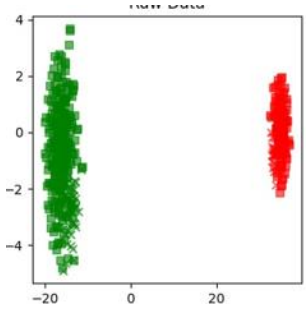
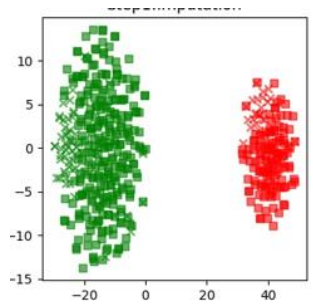
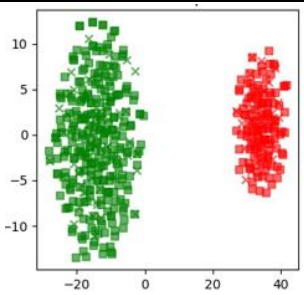
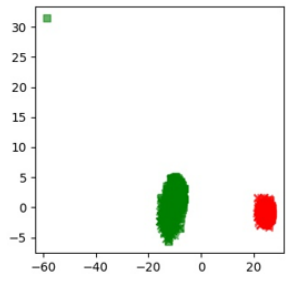
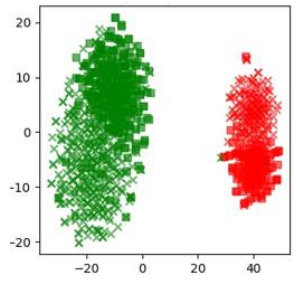
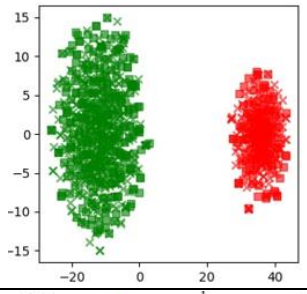
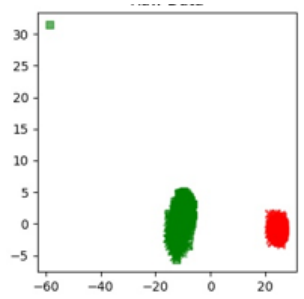
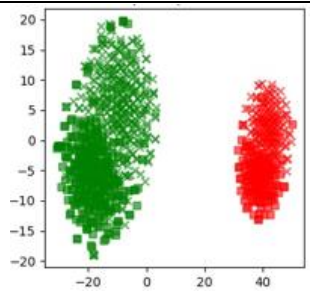
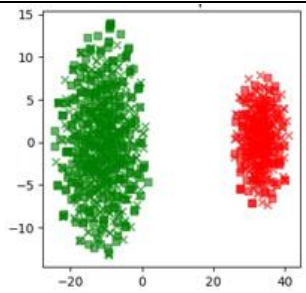
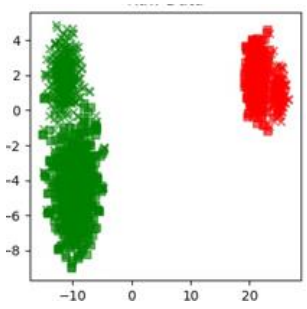
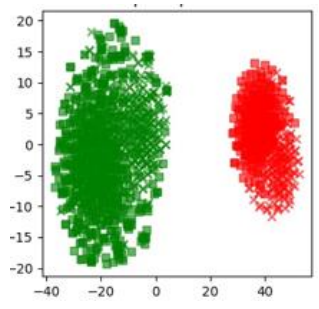
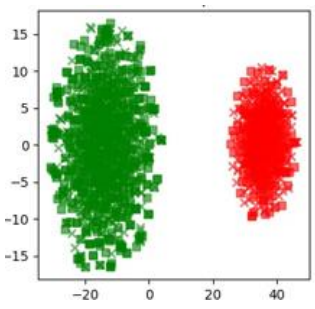
Dropout:		Raw_Data	Step1:Imputation	Step2:MMD
B1:800 B2:400	025			
	005			
B1:500 B2:450	025			
	005			
B1:500 B2:900	025			

Fig.4 L2-Normalization and Inherit centroids

Clustering: After eliminating the batch effect, kmeans cluster the latent space and output the predicted results. The Adjusted Rand Index(ARI) is used to evaluate the performance of DRjCC、Monocle2、RaceID3、SC3、scvi、Seurat3 and SIMLR clustering algorithms.

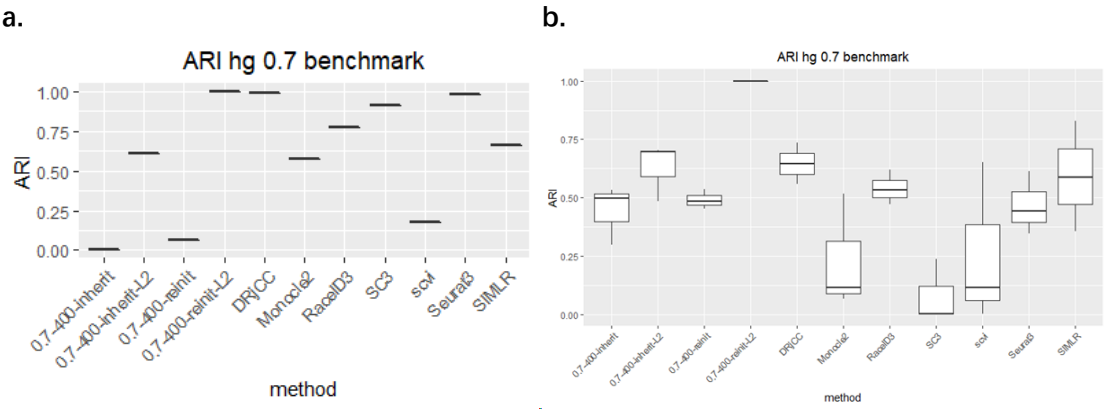


Fig.5 The performance of the model in different clustering algorithms is evaluated by ARI. a, the result of the model has no MMD as the loss function and the simulation dataset have no batch effect. b, the result of the model with MMD as the loss function and the simulation dataset have the batch effect and the batch effect meet the assumptions.

DISCUSSION

The model also has limitations. We have not taken intra-batch variation into consideration. For example, Haghverdi [2] proposed Orithogonalization to the batch vector to avoid the "kissing" problem of dense subpopulation, and to eliminate the change along the average batch in target batch and reference batch. Korsunsky[3] also uses soft clustering to assign each cell to multiple clusters, and Harmony in each cluster calculates the correction factor for each dataset-specific centroids and global centroid for each cluster.

Another limitation is that MMD applies the loss function to the entire batch instead of only considering the loss between similar cell cluster pairs between different batches.

Taking soft clustering into consideration is also our future work. Because clustering and batch effect elimination are interrelated and kmeans is very sensitive to clustering noise data and nonlinear data set [10]. We can use fuzzy clustering to classify each cell to multiple categories in the form of probability with a penalty term to ensure the maximized diversity of datasets in each cluster and then calculate the correction factor for each dataset between the global centroid of each cluster and dataset-specific centroids. After removing the batch effect. We should also use the predicted results of the soft clustering to annotate the cluster results and even discover unannotated experiments in the dataset.

METHODS

Simulation data generation and preprocessing. We use Splatter library [8] to simulate two batches of two different cell types according to differential expression parameters (See Data availability). We filtered out genes expressed in less than ten cells. We use scanpy to normalize each cell, and then log transform the data. Identify highly variable genes, remove conservative genes, and leave differentially expressed genes for subsequent analysis. Finally, we scale the data to the normal distribution. After data preprocessing, only 493 genes and 1400 cells were preserved.

Imputation and MMD correction. The model is an autoencoder which projects the original data into a low-dimensional space H to remove the experiment artifacts of different batches (Fig 5). The mini-batch gradient descent algorithm is used to train the model. For each iteration in each epoch, the mini-batch with a maximum 256 sample size. The loss is calculated on the entire mini-batch between the target batches and the reference batches, and the reference batches are randomly selected, the default batch is the first batch.

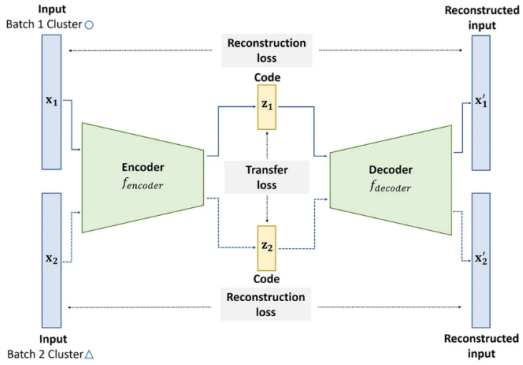


Fig 6 The autoencoder is trained to eliminate batch effects among the target batch and the reference batch. X_1 and X_2 represent the training with cells in Batch1 and Batch2 respectively. The calculation of the loss is presented by the black dashed line.

The optimized loss function consists of two parts: one part for the reconstruction loss, which takes into account the reconstruction loss among the input and the output of the decoder layer, and the other part for the loss among the input and the imputation for the output of the decoder layer.

$$L_{imp} = \sum_{i=1}^{B} \left\| X^i - \widehat{X^i} \right\|_\sigma^2 + \left\| X^i - \widehat{X^i}_{imp} \right\|_\sigma^2 \quad (1)$$

Where B is N batch number, and X^i is the input data, $\widehat{X^i}$ is the auto-encoder reconstruction output X, $\widehat{X^i}_{imp}$ is the imputation of the $\widehat{X^i}$ to recover missing values.

The second part is the loss of calculating the maximum mean discrepancy (MMD), which estimates the differences in distributions between different batches. In its own batch, the MMD distance compares the average distance from each point to the other point and compares the distance from the point in one batch to points in the other batch. The penalty term Lb minimize the distance metric between distributions is defined as

$$L_B = \sum_{i\,!=\mathrm{ref}} \mathrm{MMD}(V_{ref}, V_i) \quad (2)$$

where MMD [7] is defined as

$$\mathrm{MMD}[\mathcal{F}, X, Y] = \left[\frac{1}{m^2} \sum_{i,j=1}^{m} k(x_i, x_j) - \frac{2}{mn} \sum_{i,j=1}^{m,n} k(x_i, y_j) + \frac{1}{n^2} \sum_{i,j=1}^{n} k(y_i, y_j) \right]^{\frac{1}{2}}$$

The kernel trick was used to estimate the distribution of similar cell types within batches. Training the autoencoder is to eliminate batch effects in the hidden layer by minimizing the MMD loss between the similar cell types, To sum up, the total loss function can be written as

$$L_{loss} = L_{imp} + \lambda L_B \quad (3)$$

and the hyper-parameter λ is range from 0.0 to 1.0.

Clustering. Firstly, L2 normalizes the latent space H to avoid overfitting, then we select kmeans to cluster the above latent space H and get the predicted results. The Adjusted rand index (ARI) can be used to evaluate the performance of the predicted result and ground-truth. We visualize the latent space using 2D t-SNE plots the latent space of different simulation dataset which containing two batches of two cell types

DATA AVAILABILITY

R script to generate the simulation dataset and the generated Simulation Dataset are downloadable at https://github.com/20431767/single-cell.

CODE AVAILABILITY

The Model is written in python with tensorflow. The Source code can be downloaded from https://github.com/20431767/single-cell.

REFERENCES:

[1] Haghverdi, L., Lun, A., Morgan, M. et al. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat Biotechnol 36, 421–427 (2018). https://doi.org/10.1038/nbt.4091

[2] Lun A. Further MNN algorithm development. https://MarioniLab.github.io/FurtherMNN2018/theory/description.html. 2019

[3] Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, Baglaenko Y, Brenner M, Loh P-r, Raychaudhuri S. Fast, sensitive and accurate integration of single-cell data with Harmony. Nature Methods; 2019. https://doi.org/10.1038/s41592-019-0619-0. Accessed 1 Mar 2019.

[4] Amodio, M., van Dijk, D., Srinivasan, K. et al. Exploring single-cell data with deep multitasking neural networks. Nat Methods 16, 1139–1145 (2019). https://doi.org/10.1038/s41592-019-0576-7

[5] Brbić, M., Zitnik, M., Wang, S. et al. MARS: discovering novel cell types across heterogeneous single-cell experiments. Nat Methods 17, 1200–1206 (2020).

https://doi.org/10.1038/s41592-020-00979-3

[6] Wang, T., Johnson, T.S., Shao, W. et al. BERMUDA: a novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes. Genome Biol 20, 165 (2019). https://doi.org/10.1186/s13059-019-1764-6

[7] Borgwardt, Karsten & Gretton, Arthur & Rasch, Malte & Kriegel, Hans-Peter & Schölkopf, Bernhard & Smola, Alexander. (2006). Integrating structured biological data by Kernel Maximum Mean Discrepancy. Bioinformatics (Oxford, England). 22. e49-57. 10.1093/bioinformatics/btl242.

[8] Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data.Genome Biol 18, 174 (2017). https://doi.org/10.1186/s13059-017-1305-0

[9] Welch J, Kozareva V, Ferreira A, Vanderburg C, Martin C, Macosko E. Integrative inference of brain cell similarities and differences from single-cell genomics. bioRxiv. 2018:459891 Available from: http://biorxiv.org/content/early/2018/11/02/459891.abstract. Accessed 4 Mar 2019

[10] Cebeci Z , Yldz F . Comparison of K-Means and Fuzzy C-Means Algorithms on Different Cluster Structures. Journal of Agricultural Informatics, 2015.