An Efficient and Reliable scRNA-seq Data Imputation Method using Variational Autoencoders

Widad Alyassine[1], Anuradha samkham Raju[3], Ali Braytee[2], Ali Anaissi[1], and Mohamad Naji[1]

[1] School of Computer Science, The University of Sydney
[2] School of Computer Science, The University of Technology Sydney
[3] Victorian Institute of Technology, Sydney Australia
widad.yassien@gmail.com, anuradha.raju@vit.edu.au,
ali.braytee@uts.edu.au, ali.anaissi@sydney.edu.au,
mohamad.naji@uts.edu.au

Abstract. Single-cell RNA sequencing (scRNA-seq) provides the expression profiles of individual cells to study cell-to-cell variation within a cell population and analyses single-cell RNA-seq data to discover population heterogeneity. However, the original scRNA-seq data often contains many zeros due to dropout events, which can cause significant deviation during analysis. Although deep learning-based methods have been developed for imputing missing data in scRNA-seq, they are not explicitly designed to model the data distribution, hindering the generation of multiple plausible imputations. In this study, we propose a novel method to restore incomplete scRNA-seq data by handling dropouts and retaining true zeros with potential applications in Cancer research projects. Our method employs a variational autoencoder (VAE) as a deep generative model to learn the data distribution and reconstruct the imputed cell-gene matrix. VAE can learn a low-dimensional representation of the data that captures the most important features to generate plausible imputations. Several experiments have been conducted to evaluate the accuracy and efficiency of our method compared to state-of-the-art methods using datasets of various sizes. Evaluation metrics such as mean squared error (MSE) and clustering tests were employed, and the results demonstrated that our method outperformed other approaches. Interestingly, our proposed model exhibit strong stability when dealing with different data magnitudes.

Keywords: Variational autoencoders, single-cell data, data imputation

I Introduction

The recent development of high-throughput technologies has provided an efficient way to measure gene expression in single cells. However, unlike traditional bulk RNA sequencing, the traditional RNA-seq method cannot capture the complexity of tissues or systems at the cellular level because it measures thousands

of cells simultaneously [4]. In other words, analyzing single-cell RNA data poses a challenge due to its high sparsity. Consequently, more information can be obtained by comparing transcriptional similarities and differences at the cellular level. Researchers can gain insights into fundamental gene expression characteristics, such as splicing patterns or noise in transcriptional responses. Additionally, clinicians can customize more specific and efficient treatment plans for diseases such as tumors or nervous system disorders. However, a limitation of single-cell RNA sequencing (scRNA-seq) is the generation of sparse data, which contains numerous zero expressions [13]. These zeros can indicate either no reads mapping to a specific gene in a cell or dropout events (false zeros). When a high percentage of zeros is present, unreliable results may arise due to dropouts, leading to biological fluctuations in the measured trait and difficulties in quantifying small numbers of molecules. The high dropout rate in single-cell RNA sequencing can be attributed to several reasons. One reason is the technical limitations that hinder the detection and extraction of low-expression genes or genes that are not expressed in a particular cell, resulting in incomplete data extraction. To address this issue, various imputation methods have been developed to replace the false zeros based on statistical indicators [8]. These methods can be categorized into three groups. The first group involves direct imputation of dropouts using probabilistic models, such as Savers and ScImpute. The second group utilizes clustering algorithms to replace zeros with similar gene expressions, exemplified by methods like MAGIC [5] and DrImpute [7]. The third group employs deep learning neural networks to identify latent space representations of genes in a cell. However, it is important to note that these methods do not possess comprehensive advantages for imputing RNA sequencing data. They may have limitations such as longer running times or lower performance levels.

In this work, we propose a novel method for imputing missing values in gene expression matrices using Variational Autoencoder (VAE). The VAE [9] is trained to capture the underlying distribution of the input single-cell RNA sequencing (scRNA-seq) data and impute the missing values while preserving the true values with minimal modification. We compared the performance of our model with the other three popular imputation models (DeepImpute [2], MAGIC [5], and SCVI [11]), using four different datasets obtained from two public biology repositories, namely 10x Genomics and the Gene Expression Omnibus. We focused on evaluating the accuracy and clustering capabilities of our model compared to state-of-the-art methods. To measure accuracy, we calculated the mean squared error (MSE) of 10% of the covered true values, and for evaluating clustering results, we used the silhouette score.

II Related Work

Although single-cell technology can handle a large amount of cell data simultaneously and analyzing cell heterogeneity, the high dropout rate brings a computational challenge to the experiments [8]. Currently, several imputation methods have been applied to single-cell RNA data and have presented promising results.

In this section, we review the classic and deep learning-based methods to handle the missing scRNA-seq data.

A Classic methods

MAGIC [5] shares information across similar cells considering heat diffusion concept [12]. The key idea behind this approach is to construct a Markov transition matrix by normalizing the similar matrix of the single cells. This approach can denoise the high-dimensional data commonly applied to single-cell RNA-seq data. It has the capability to restore two or three-dimensional gene interactions, impute complex and non-linear shapes of interactions, preserve clustering structure, enhance cluster specificity, and restore trajectories. Another method, DrImpute [7] utilizes a clustering algorithm to aggregate similar genes together and impute the dropouts (zero-values) by assigning them the average value of cells within the corresponding cluster.

B Deep learning methods

AutoImpute [15] uses an overcomplete autoencoder to impute the missing gene expression values. The DCA [6] model denoises the data by constructing an autoencoder algorithm and forming the likelihood of the missing zero instead of the input itself. DeepImpute [2] was introduced as the next generation of imputation after the DCA method, which splits a large dataset into subgroups and constructs sub-neural networks to improve the efficiency of the training process. Due to its utilization of sub-neural networks, DeepImpute has demonstrated superior performance compared to other imputation methods, as it reduces complexity and improves training efficiency for large datasets. SCVI [11] is a scalable framework for probabilistic representation and analysis of gene expression in single cells. It uses stochastic optimization and a deep neural network to aggregate information across similar cells and an approximated posterior distribution by assuming a zero-inflated negative binomial distribution. The model also considers probabilistic representation to dropouts, but both the encoder and decoder parts form a distribution of data instead of directly compressing the data. SCVI models are comprehensive in capabilities and scalable to very large datasets. Furthermore, an unsupervised clustering algorithm called SAUCIE [1], which applies two novel regularizations: (1) an information dimension regularization to penalize entropy as computed on normalized activation values of the layer and thereby encourage binary-like encodings that are amenable to clustering; and (2) a maximal mean discrepancy penalty to correct batch effects. Another work is proposed, known as GNNimpute [16], which applies the graph attention convolutional layer to aggregate the similar neighbouring nodes. Therefore, its special structure related to the autoencoder could transfer the expression value in the same tissue in low-dimensional vectors. Also, the attention structure can assign weights to different cells according to attention coefficients. A method based generative adversarial network (GAN) is proposed for data imputation [17]. This method employs the generator component of the GAN model to synthesize new plausible data

by summarizing the data distribution, while the discriminator component attempts to differentiate whether the data originates from the training data or the generator.

III Methods

A A brief on VAE

A variational autoencoder (VAE) is a type of generative model that combines elements of both autoencoders and variational inference. It is a powerful neural network architecture used for unsupervised learning and data generation tasks [12]. The input data is sampled from a parametrized distribution. The encoder and decoder are trained jointly to optimize two main objectives: the reconstruction loss and the KL divergence loss. The reconstruction loss measures how well the decoder can reconstruct the original input from the latent space. The KL divergence loss encourages the learned latent space to follow a prior distribution (often a standard Gaussian) by minimizing the divergence between the learned distribution and the prior distribution. By optimizing these two objectives simultaneously, VAEs learn to generate new samples from the learned latent space.

B Our data imputation approach based VAE

The matrix of raw data comprises of m rows of cells and n rows of genes, including potentially bad genes. Preprocessing the raw data involves gene selection, normalization by library size, and log transformation to obtain the processed data matrix, as illustrated in Figure 1. We subsequently apply VAE imputation to the processed data matrix. The preprocessing steps are further discussed in section B.

The processed data matrix serves as the input for our model. The VAE model extracts one cell at a time, and the input layer contains nodes x_1 to x_n, depending on the number of cells. The gene values of each cell pass through two fully connected layers, which use linear transformation as shown in Equation 1, where W_t are the parameters, X is the input values, and b is the bias.

$$Y = W_t X + b \tag{1}$$

The VAE model produces two parallel middle hidden layers, namely, the Z and T layers. Herein lies our modification compared to the vanilla VAE, aiming to enhance the model's feature learning capabilities. In the encoder part, we learn the mean (μ) and variance $log(\sigma^2)$ from the input data sets. An E layer, representing epsilon, stores random errors according to the normal distribution. To compute the Z and T layers, we first calculate the standard deviation (σ) of the output. We then compute Equation 2 for both layers, where ε denotes the random errors, σ and μ denote the standard deviation and mean, respectively, for each node in the Z and T layer. The results are further computed using

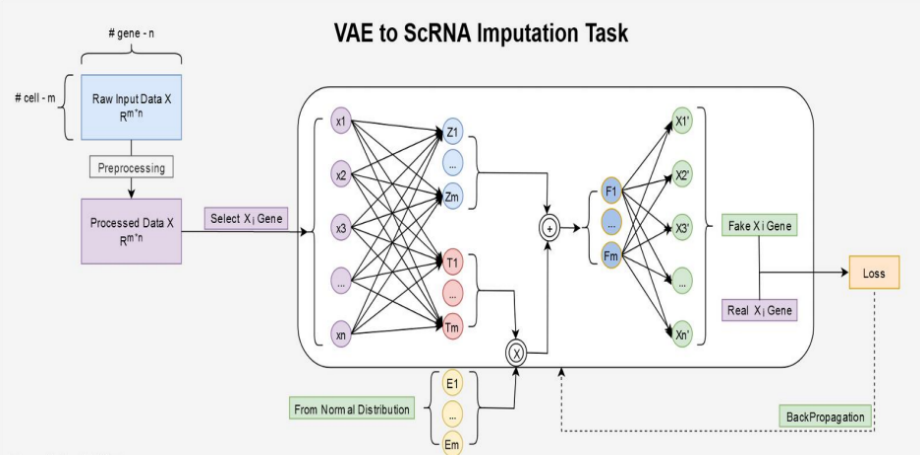


Fig. 1. The components of our proposed approach

Equation 3 and stored in the F layer. This step increases the uncertainty of the model. The F layer then serves as the input for the decoder, which decodes the values and reconstructs the matrix with the same dimension as the input layer of the encoder.

$$Inter = \varepsilon \times \sigma + \mu \tag{2}$$

$$F_i = \frac{((\varepsilon_{zi} \times \sigma_{zi} + \mu_{zi}) + (\varepsilon_{ti} \times \sigma_{ti} + \mu_{ti}))}{2} \tag{3}$$

Our model generates a simulated gene sequence, which is then compared to the actual gene sequence. To fine-tune the model's parameters, we leverage the Kullback-Leibler Divergence (KLD) and Mean Squared Error (MSE) metrics. The total loss function used for parameter tuning is calculated by means of Eq. 4. Finally, the model applies backpropagation to update the parameters of each node.

$$totalloss = KLD + MSE \tag{4}$$

where KLD is the similarity between the generated distribution and the true distribution. The purpose of KLD is to make the generated distribution set μ equals to 0 and σ^2 equals to 1 and it is calculated as follows

$$KLD = y \times log(\frac{y}{\hat{y}}) \tag{5}$$

Mean Squared Error (MSE) measures the average squared distance between the predicted values and the true values. Its objective is to minimize the difference between the predicted and actual values as follows

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_l)^2}{n} \tag{6}$$

In summary, our proposed model has a significant advantage over vanilla VAE in that it employs two parallel hidden layers, namely the Z layer and the T layer. This design allows the model to learn more features, which in turn leads to better performance.

IV Experiments

A Datasets

We evaluated our proposed approach and the state-of-art using the following datasets which can be downloaded from Gene Expression Omnibus [4] and the 10xGenomics website [5]

- Blakeley [3]: Single-cell RNA sequencing was performed on a human embryo to define three cell lineages of the human blastocyst. This data set includes 30 cells x 23784 genes. The total number of genes with a value of 0 in this dataset is 513264, and the percentage is 71.90%. The data set is available at the Gene Expression Omnibus (GEO66507)
- Line_FPKM [10]: Single-cell RNA sequencing was performed on human colorectal tumours to define and elucidate cellular heterogeneity. This data set includes 562 cells x 75241 genes. The total number of genes with a value of 0 in this dataset is 25,426,102, and the percentage is 60.13%. The data set is available at the Gene Expression Omnibus (GSE81861)
- Brain Nuclei_1 and Brain Nuclei_4 [6]: both of are single-cell RNA Sequences of nuclei that were isolated from E18 mouse combined cortex, hippocampus and ventricular zone tissues. Brain Nuclei_1 includes 2566 cells x 123 genes and Brain Nuclei_4 includes 5284 cells x 123 genes. The total number of genes with a value of 0 in these datasets are 250330 and 507671 respectively, and the percentages are 79.31% and 78.11% respectively. Table 1 compares the shapes of the four data sets and the proportion of 0 values.

B Data preprocessing

Gene filtering We define a gene as valid if it is present in at least three cells. Genes that exist in a very limited number of cells (less than three) are considered outliers and are not included in the training process.

[4] https://www.ncbi.nlm.nih.gov/geo/
[5] https://www.10xgenomics.com/
[6] https://www.10xgenomics.com/resources/datasets

Table 1. Datasets

	Blakeley	Line_FPKM	Brain Nuclei_1	Brain Nuclei_4
Shape (cells x genes)	30 x 23794	562 x 75241	2566 x 123	5284 x 123
Count of 0 values	51,3264	25,426,102	250,330	507,671
The proportion of 0 values	71.90%	60.13%	79.31%	78.11%

Median Normalization Expression matrices are normalized by dividing each read count by the total counts in the corresponding cell and subsequently multiplying it by the median of the total read counts across all cells. Median normalization is a common method to normalize single-cell sequence data. This method establishes a geometric mean pseudo-sample, identifies the median expressed gene in it, and adjusts the counts of all other samples to match that gene's expression level. The anchoring of counts based on the gene with median expression is considered reliable and yields consistent results.

Gene Selection For imputation, only the top 1000 high-dispersion genes are retained. The selection of these genes was based on the method namely DropClust [14]. It was employed to determine approximate neighbourhoods for individual transcriptomes and apply an exponential decay function to select a greater number of expression profiles from clusters of relatively smaller sizes.

Log transformation The log-transformed matrix is generated by adding 1 pseudo count to the raw expression data. The resulting log-transformed expression matrix is used as input for our model and other comparable models for single-cell RNA-seq imputation using variational autoencoders. By using the log transform method, we can reduce the skew of the data, especially the numbers which are highly skewed. Furthermore, the log-transformed data is easier to interpret, making it more accessible for further analysis.

Preprocessed data After applying the preprocessing steps described earlier, we obtained the preprocessed datasets as presented in Table 2, which will be used as training and testing data. To simulate the dropout phenomenon that will occur in the cell experiment, we randomly selected 20% of the non-zero data points and replaced them with 0. We also recorded the indexes of these masked values for the evaluation phase.

V Results

A Mean square error (MSE)

Initially, we evaluate the performance of our model and state-of-the-art methods by computing the mean squared error (MSE). Single-cell RNA sequencing

Table 2. Preprocessed datasets

	Blakeley	Line_FPKM	Brain Nuclei_1	Brain Nuclei_4
Shape (cells x genes)	30 x 1000	562 x 1000	2566 x 68	5284 x 80
Count of 0 values	15,444	109,251	110,392	281,454
The proportion of 0 values	51.48%	19.44%	63.23%	66.58%
Improvement of 0 values	20.42%	40.69%	16.08%	11.53%

experiments are often associated with high dropout rates, resulting in missing values in expression datasets. Imputation methods are commonly used to address these missing values, but they often generate an excessive number of zero expression values. This abundance of zeros poses a challenge in distinguishing between a true zero and a dropout event without a confirmatory method. To evaluate the performance of the compared imputation methods, we first set a seed and randomly select 10% of the original values from the dataset. Subsequently, we replace these selected values with zeros, while storing their original value and index in an array. For testing our model, we utilize the dataset with 10% of zero-covered values. After applying each imputation method, we obtain a reconstructed expression matrix. We then locate the reconstructed values based on their index in the previously saved array and calculate the MSE score of each method by comparing the reconstructed values with the original values. In our approach, we disregard the true meaning of zeros, as our primary focus is to assess the difference between the covered zeros and the true values. The MSE scores of all the compared methods across four datasets are visualized in Fig. 2.

In Fig. 2, it is evident that our proposed method achieves the best performance in terms of mean squared error (MSE) scores for the Brain Nuclei_1, Brain Nuclei_4, and Blakeley datasets. However, in the Line_FPKM dataset, our MSE score is slightly higher than that of the MAGIC method. Overall, our model demonstrates accurate imputation performance compared to the other three imputation methods across all four datasets. Specifically, considering the Brain Nuclei_1 dataset depicted in Fig. 2, our model, along with DeepImpute and MAGIC, exhibits remarkably low MSE scores. Conversely, the SCVI method performs poorly in this dataset, with an MSE score greater than 9. Moving to the Brain Nuclei_4 dataset illustrated in Fig. 2, our model achieves substantially lower MSE scores than any other imputation method, with an MSE score of less than 1. The MSE scores of the DeepImpute and MAGIC methods are quite similar, but the SCVI method still has the poorest MSE score among the four methods. For the Blakeley dataset illustrated in Fig. 2, our model continues to exhibit the best MSE score. The MSE scores of the DeepImpute and SCVI methods are quite similar, while the MAGIC method has the poorest MSE score among the four methods. However, it is worth noting that all MSE scores of these four methods for the Blakeley dataset are higher than those of the other three datasets, indicating that the expression values in the Blakeley dataset are more variable. Finally, in the Line_FPKM dataset shown in Fig. 2, the MAGIC method achieves the lowest MSE score among the four methods. However, the

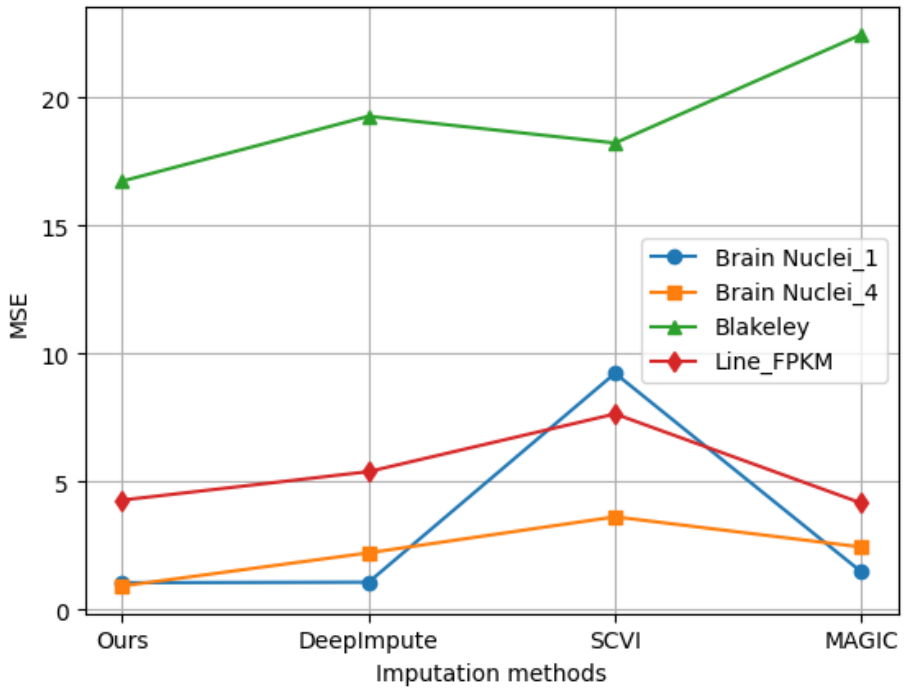


Fig. 2. MSE results of our proposed method and other existing imputation methods

difference is only slight compared to our model's MSE score. Once again, the SCVI method has the poorest MSE score among the four methods.

B Clustering results

Determine the optimal number of clusters To generate clusters that capture similar data types within the single-cell dataset, we employ the K-means algorithm using Line_FPKM dataset. First, the missing values are imputed using our method, resulting in complete expression data for each cell. Subsequently, K-means clustering is applied to the imputed dataset. The initialization parameter "K" which represents the number of clusters, is explored over the range from 1 to the square root of the number of dimensions. To determine the optimal value of "K" we cyclically assess the sum of intra-cluster sum of squared errors (SSE) for various values of "K" In Fig. 3, we observe that the SSE curve exhibits a distinctive elbow at a value of 7. Hence, we select 7 as the optimal value for "K" to achieve meaningful clusters.

Clustering results To compare the clustering results between our imputation method based VAE-based method and state-of-the-art approaches, we employed

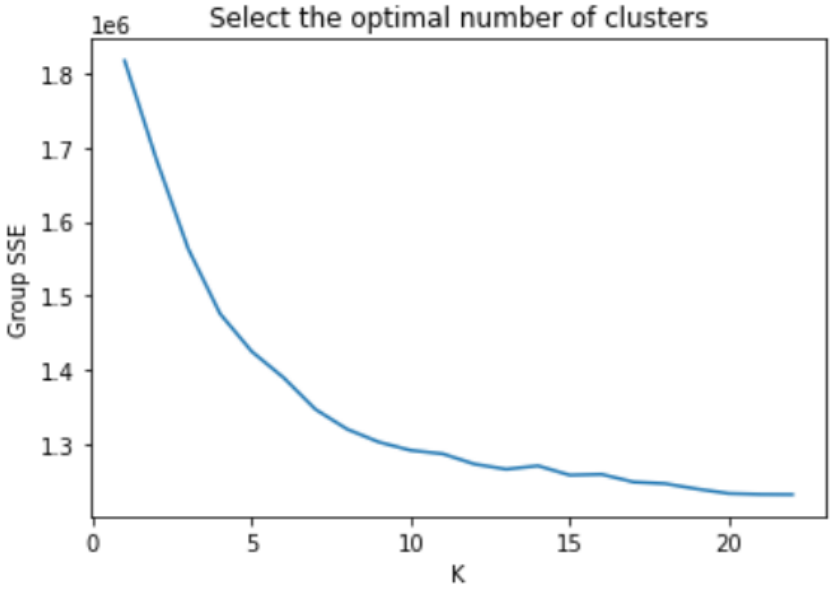


Fig. 3. Select optimal number of clusters

several clustering measures, including the silhouette score and visual inspection. The silhouette score serves as an indicator of clustering quality and ranges from -1 to 1. A score of 1 suggests dense and well-separated clusters, while a score close to 0 indicates overlapping clusters with samples near the decision boundary of neighboring clusters. On the other hand, a negative score suggests that samples may have been assigned to incorrect clusters.

For the evaluation, we generated clusters using four different imputation models: our method, DeepImpute, SCVI, and MAGIC, based on four different gene expressions. Table 3 presents the average silhouette scores across seven clusters for both the original and imputed data. The results demonstrate that the average silhouette scores for imputed data from all compared imputation methods were higher than those for the original data without imputation. This underscores the importance of imputation methods in handling missing single-cell data. In most datasets, our proposed method exhibited the highest silhouette score and significantly outperformed the state-of-the-art methods in the Brain Nuclei_1, Brain Nuclei_4, and Blakeley datasets. Notably, the DeepImpute method yielded an extremely low silhouette score.

In addition to evaluating silhouette scores, we utilized UMAP components (refer to Fig. 4) to visualize cell clusters. Subsequently, we performed cell clustering using K-means on the imputed data of Line_FPKM dataset obtained from various imputation methods. Remarkably, our proposed method demonstrated a distinct separation of cells into well-defined clusters (see Fig.4(a)), resulting

Table 3. Silhouette score results across the compared methods

	Brain Nuclei_1	Brain Nuclei_4	Blakeley	Line_FPKM
No imputation	0.12	0.18	0.05	0.04
Ours	**0.88**	**0.90**	**0.48**	0.15
DeepImpute	0.26	0.17	0.08	0.07
SCVI	0.28	0.19	0.38	0.46
MAGIC	0.18	0.49	0.37	**0.73**

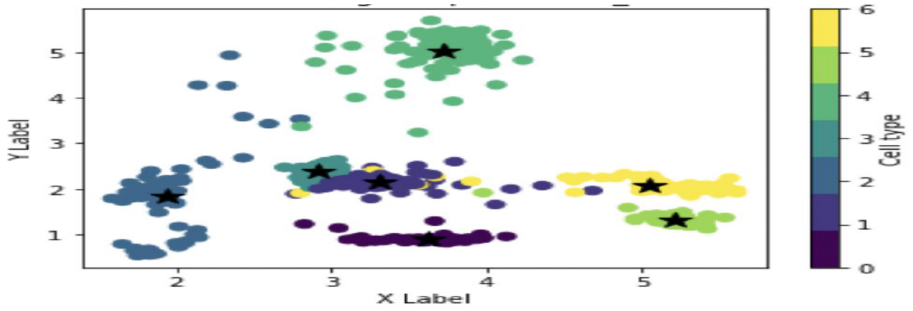
in the most substantial improvement in clustering metrics compared to other imputation methods (see Fig.4(b-d)). While DeepImpute, SCVI, and MAGIC methods also managed to separate some clusters, they occasionally partitioned clusters in a manner that deviated from the original cell type labels and lacked clear boundaries between clusters.

C Running time

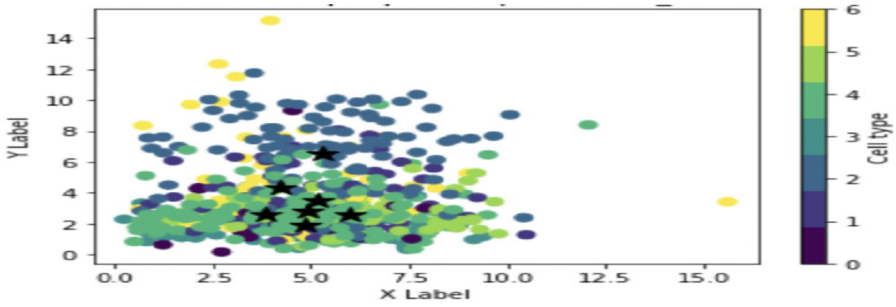
Single-cell sequencing datasets often contain thousands or even millions of data points. Consequently, it is crucial for model designers to consider the runtime performance of their approaches. In this regard, we evaluated the performance of the compared imputation models across four datasets. In Fig. 5, it is evident that the MAGIC model exhibits significantly better runtime performance compared to the other three methods. This is because the MAGIC model relies solely on statistical methods for imputation, without involving the reconstruction of new gene expressions using deep learning models. Given that our proposed model, DeepImpute, and SCVI methods all generate new gene expressions based on similar underlying principles, we specifically compared the runtime performance of our model with these two methods. Our findings indicate that our model surpasses both the DeepImpute and SCVI models in terms of runtime efficiency.

VI Conclusion

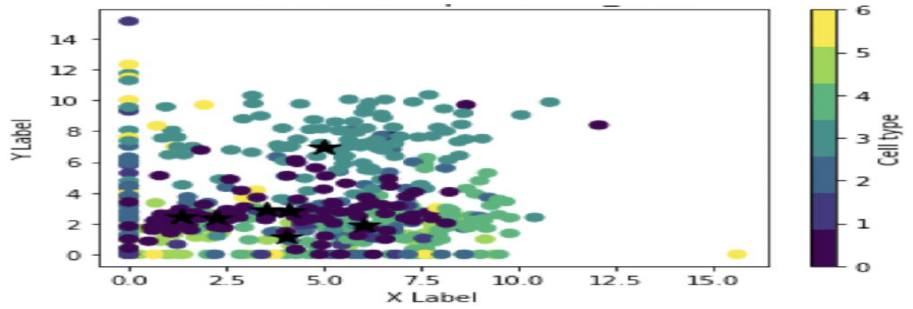
This study compared the performance of our proposed imputation method, based on a single-cell variational autoencoder (VAE), with three other imputation models. Our method consistently achieved superior results in terms of MSE scores across four different datasets. This indicates its effectiveness in imputing missing values while minimizing disruption to biologically inactive genes. In the clustering experiments, our method outperformed the state-of-the-art methods by successfully recovering a high number of dropouts and improving the separability of cell types. Furthermore, it exhibited the best running time performance compared to the other two neural network models. This provides an efficient and reliable approach for single-cell RNA sequence imputation for gene researchers. This interdisciplinary research combines deep learning using neural networks



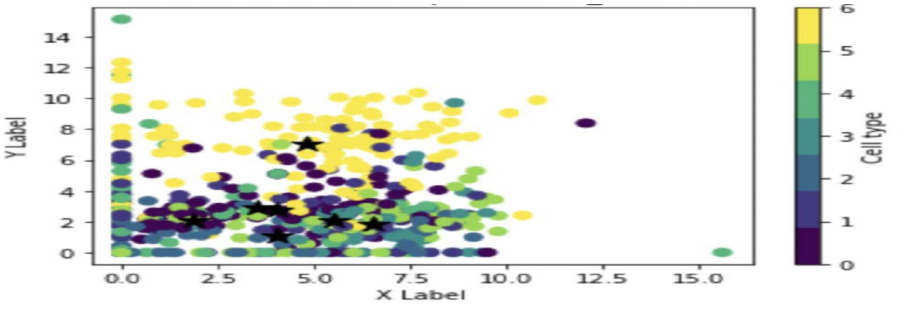
(a) Ours



(b) DeepImpute



(c) SCVI



(d) MAGIC

Fig. 4. UMAP plots for various imputation methods to assess the impact of imputation on downstream functional analysis of the Line_FPKM dataset. Colors represent the annotated original cell types

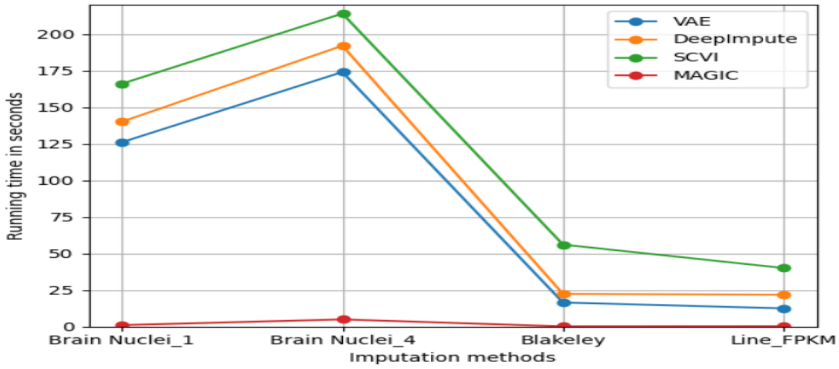


Fig. 5. Running time comparison for the imputation methods on four datasets

with the field of genetic engineering. While the VAE model has its limitations, such as the inability to capture the hierarchical relationship between latent variables, further exploration is necessary to address this constraint and enhance its capabilities.

Acknowledgement

We used the AI tool ChatGPT (https://openai.com) for light editing to improve the paper's readability.

References

1. Amodio, M., Van Dijk, D., Srinivasan, K., Chen, W.S., Mohsen, H., Moon, K.R., Campbell, A., Zhao, Y., Wang, X., Venkataswamy, M., et al.: Exploring single-cell data with deep multitasking neural networks. Nature methods **16**(11), 1139–1145 (2019)
2. Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X., Garmire, L.X.: Deepimpute: an accurate, fast, and scalable deep neural network method to impute single-cell rna-seq data. Genome biology **20**(1), 1–14 (2019)
3. Blakeley, P., Fogarty, N.M., Del Valle, I., Wamaitha, S.E., Hu, T.X., Elder, K., Snell, P., Christie, L., Robson, P., Niakan, K.K.: Defining the three cell lineages of the human blastocyst by single-cell rna-seq. Development **142**(18), 3151–3165 (2015)
4. Deng, Y., Bao, F., Dai, Q., Wu, L.F., Altschuler, S.J.: Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. Nature methods **16**(4), 311–314 (2019)
5. Dijk, D.v., Nainys, J., Sharma, R., Kaithail, P., Carr, A.J., Moon, K.R., Mazutis, L., Wolf, G., Krishnaswamy, S., Pe'er, D.: Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data. BioRxiv p. 111591 (2017)

6. Eraslan, G., Simon, L.M., Mircea, M., Mueller, N.S., Theis, F.J.: Single-cell rna-seq denoising using a deep count autoencoder. Nature communications **10**(1), 390 (2019)
7. Gong, W., Kwak, I.Y., Pota, P., Koyano-Nakagawa, N., Garry, D.J.: Drimpute: imputing dropout events in single cell rna sequencing data. BMC bioinformatics **19**, 1–10 (2018)
8. Hou, W., Ji, Z., Ji, H., Hicks, S.C.: A systematic evaluation of single-cell rna-sequencing imputation methods. Genome biology **21**, 1–30 (2020)
9. Kingma, D.P., Welling, M., et al.: An introduction to variational autoencoders. Foundations and Trends® in Machine Learning **12**(4), 307–392 (2019)
10. Li, H., Courtois, E.T., Sengupta, D., Tan, Y., Chen, K.H., Goh, J.J.L., Kong, S.L., Chua, C., Hon, L.K., Tan, W.S., et al.: Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. Nature genetics **49**(5), 708–718 (2017)
11. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., Yosef, N.: Deep generative modeling for single-cell transcriptomics. Nature methods **15**(12), 1053–1058 (2018)
12. Lopez, R., Regier, J., Jordan, M.I., Yosef, N.: Information constraints on auto-encoding variational bayes. Advances in neural information processing systems **31** (2018)
13. Lukusa, T., Lee, S., Li, C.S.: Review of zero-inflated models with missing data. Current Research in Biostatistics **7**(1), 1–12 (2017)
14. Sinha, D., Kumar, A., Kumar, H., Bandyopadhyay, S., Sengupta, D.: dropclust: efficient clustering of ultra-large scrna-seq data. Nucleic acids research **46**(6), e36–e36 (2018)
15. Talwar, D., Mongia, A., Sengupta, D., Majumdar, A.: Autoimpute: Autoencoder based imputation of single-cell rna-seq data. Scientific reports **8**(1), 1–11 (2018)
16. Xu, C., Cai, L., Gao, J.: An efficient scrna-seq dropout imputation method using graph attention network. BMC bioinformatics **22**(1), 1–18 (2021)
17. Xu, Y., Zhang, Z., You, L., Liu, J., Fan, Z., Zhou, X.: scigans: single-cell rna-seq imputation using generative adversarial networks. Nucleic acids research **48**(15), e85–e85 (2020)