scVGAE: A Novel Approach using ZINB-Based Variational Graph Autoencoder for Single-Cell RNA-Seq Imputation

Yoshitaka Inoue[1]

University of Minnesota, Department of Computer Science and Engineering, Minneapolis, MN, USA

Abstract. Single-cell RNA sequencing (scRNA-seq) has revolutionized our ability to study individual cellular distinctions and uncover unique cell characteristics. However, a significant technical challenge in scRNA-seq analysis is the occurrence of "dropout" events, where certain gene expressions cannot be detected. This issue is particularly pronounced in genes with low or sparse expression levels, impacting the precision and interpretability of the obtained data. To address this challenge, various imputation methods have been implemented to predict such missing values, aiming to enhance the analysis's accuracy and usefulness. A prevailing hypothesis posits that scRNA-seq data conforms to a zero-inflated negative binomial (ZINB) distribution. Consequently, methods have been developed to model the data according to this distribution. Recent trends in scRNA-seq analysis have seen the emergence of deep learning approaches. Some techniques, such as the variational autoencoder, incorporate the ZINB distribution as a model loss function. Graph-based methods like Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) have also gained attention as deep learning methodologies for scRNA-seq analysis. This study introduces scVGAE, an innovative approach integrating GCN into a variational autoencoder framework while utilizing a ZINB loss function. This integration presents a promising avenue for effectively addressing dropout events in scRNA-seq data, thereby enhancing the accuracy and reliability of downstream analyses. scVGAE outperforms other methods in cell clustering, with the best performance in 11 out of 14 datasets. Ablation study shows all components of scVGAE are necessary. scVGAE is implemented in Python and downloadable at https://github.com/inoue0426/scVGAE.

Keywords: scRNA Imputation · Variational Autoencoder · Graph Neural Network

1 Introduction

Single-cell analysis has emerged as a rapidly advancing field in recent years, allowing us to delve into the intricate details of individual cells [20]. This remarkable progress has led to the development of numerous applications that significantly enhance our comprehension of biology and drug discovery [11]. However, a pivotal challenge has arisen within this domain - the dropout phenomenon. Given the imperative to gather data from each cell individually, insufficient expression levels often impede the extraction of meaningful cellular characteristics. Various methodologies have been devised to address this issue to recover missing information effectively.

One prominent method is MAGIC [26], renowned for using K-nearest neighbors (KNN) with a Markov chain. Alternatively, the Deep Count Autoencoder (DCA) [6] approach employs the Zero Inflated Negative Binomial (ZINB) loss in instances where dropout events are prevalent. It is well-established that the single-cell RNA data distribution often conforms to the ZINB distribution.

Furthermore, the field has witnessed significant advancements in machine learning, notably the advent of Graph Neural Networks (GNN) [33]. These innovations have complemented the existing efforts, resulting in novel approaches that seamlessly integrate these techniques to perform cell clustering while preserving the intrinsic cellular characteristics.

Recently, scTAG [32] introduced a clustering method that utilizes a topology-adaptive graph convolutional autoencoder with ZINBLoss. This method employs a graph encoder and a fully connected decoder to predict the distribution and perform cell clustering from the hidden representation, achieving high prediction accuracy. However, it does not retain the imputed matrix. In response to the limitation of scTAG, we aim to generate an imputed matrix using the graph autoencoder and subsequently perform clustering based on this imputed matrix.

We propose a novel method called scVGAE, which integrates ZINBLoss and GCN to enhance imputation while maintaining cellular information integrity. By harnessing the power of Graph Neural Networks in conjunction with the Variational Autoencoder (VAE) [15], both of which leverage ZINBLoss, our approach excels in filling in missing data while faithfully preserving cellular phenotypes.

The main contributions include:

1. We developed a novel approach for scRNA-Seq imputation, utilizing a variational graph convolutional autoencoder with ZINB loss. This approach maintains cell-cell similarity while effectively fitting gene expression data into the distribution.
2. To enhance clustering stability, we utilize normalized matrices to maintain the original topology and prevent overfitting.

2   Related Work

2.1   Statistical Methods

Statistical methods are employed to extract information inherent in gene expression statistically. These techniques excel at distinguishing between biological 0s and mechanical 0s and detecting which genes are expressed correctly, which can be used as a basis for imputation. Notable examples includ ALRA [19], SDImpute [21], and scISR [25].

2.2   Affinity-based Methods

Affinity-based imputation methods rely on cell or gene similarity to predict dataset missing values. These techniques typically employ KNN or kernel functions to calculate a similarity matrix and then use this matrix to impute missing data through matrix multiplication. Examples of affinity-based imputation methods include MAGIC [26], scImpute [18], and DrImpute [8].

2.3   Deep Learning Methods

Deep learning-based approaches utilize feedforward networks, often employing MSE or ZINB loss. In this context, some methods split the data into training and testing sets for imputation, while others use autoencoders to reconstruct the original matrix. Several noteworthy examples of these deep learning-based imputation methods include AdImpute [31], DeepImpute [2] and DCA [6].

2.4   Deep Graph Methods

Graph Neural Networks (GNNs) have become increasingly prevalent in current research within this field. A popular tool is scGNN2.0 [9], which combines three autoencoders: the feature autoencoder, cluster autoencoder, and graph attention autoencoder. Through this process, scGNN2.0 generates cell clusters and an imputed matrix. Another method, GNNImpute [30], is based on an autoencoder constructed with a graph convolution encoder and a linear decoder. It takes the gene expression data as input and generates an imputed matrix. After generating the imputed matrix, MSELoss is calculated as a measure of imputation quality or accuracy.

3   Methods

3.1   Data Preprocessing

As input, we use the scRNA gene expression matrix $X \in \mathbb{R}^{n \times m}$, with n denoting the row count and m corresponding to the column count. The initial step involves filtering out genes expressed as non-zero in over 5% of the cells and genes expressed in more than 5%. Then, the normalization step was conducted, which is defined as below:

$$X = \log{(X + 1)}$$
$$X_{ij} = \sqrt{\frac{X_{ij}}{\Sigma_j X_{ij}}}. \tag{1}$$

According to equation 1, the data is rescaled by log + 1 transformation, library normalization, and square root transformation.
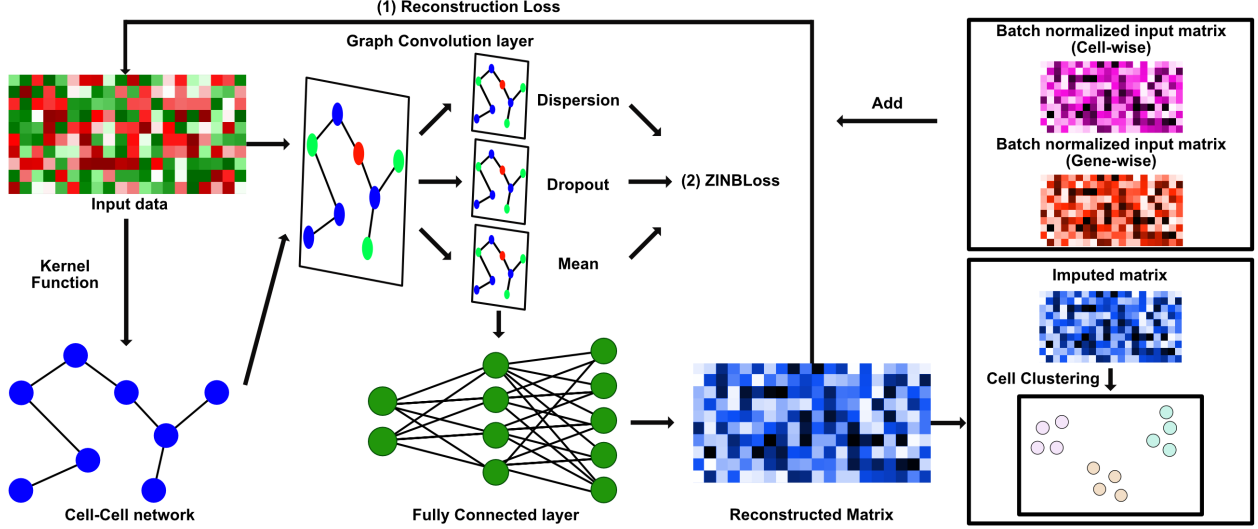
Fig. 1: An overview of scVGAE: A cell-cell network is constructed from input data representing gene expression. The network informs an affinity matrix via a kernel function that measures cell similarity, and also the input data is concurrently transformed into a feature matrix for the Graph Convolution layer. Outputs of the graph convolution, capturing mean, dispersion, and dropout values, contribute to computing the ZINBLoss. The mean output also feeds into a fully connected layer to reconstruct the original matrix. Reconstruction loss is assessed by augmenting the reconstructed matrix with cell-wise and gene-wise normalized matrices, facilitating comparison against the original matrix. Through iterative optimization, the refined reconstructed matrix is an imputed matrix for cell clustering analysis.

3.2 Adjacent matrix

The gene expression data is transformed using the linear kernel function, generating a similarity matrix. Subsequently, we retain only the highest 85% of values to establish the framework for constructing the graph. The process for graph creation is detailed as follows:

$$
\begin{aligned}
S_{ij} &= X_i^T \cdot X_j, \\
A &= S \times M, \\
\text{where } M_{ij} &= \begin{cases} 1 & \text{if } S_{ij} > \text{percent}(S, 85) \\ 0 & \text{otherwise} \end{cases}.
\end{aligned}
\tag{2}
$$

In this context, S represents the similarity matrix, X stands for the gene expression matrix, A corresponds to an adjacency matrix, M functions as a mask applied to establish graph similarities, and $\text{percent}(S, p)$ is a function that returns a threshold value at the specified percentile rank $\left(\frac{p}{100}\right)$ among the elements in S.

Next, in order to derive the symmetrically normalized Laplacian matrix L from A, the following calculations are executed:

$$
\begin{aligned}
D_{ii} &= \sum_j A_{ij}, \\
L &= D^{-\frac{1}{2}} A D^{-\frac{1}{2}}.
\end{aligned}
$$

In this context, D_{ii} represents the summation of edge weights. The term $D^{-\frac{1}{2}}$ signifies a matrix obtained through an element-wise inverse square root transformation of the matrix D.

3.3 Zero-Inflated Negative Binomial Distribution

scVGAE employs the Zero-Inflated Negative Binomial (ZINB) loss function in its methodology. This loss function is well-established and highly suitable for modeling single-cell RNA data distributions [24]. The ZINB distribution combines two essential distributions: the negative binomial and zero inflation.

The negative binomial distribution is commonly used to model count data, especially when the variance exceeds the mean. In the context of biological data, substantial variations in characteristics like cell types, genes, and other biological factors often lead to overdispersion. Therefore, the negative binomial distribution is employed in single-cell RNA analysis to represent the baseline expression accurately.

Zero inflation becomes necessary when dealing with specific challenges in single-cell RNA data, including technological limitations, dropout events, and biological non-expression, which result in numerous zero entries. By merging these two distributions, we achieve a more precise representation of single-cell RNA data expression patterns.

The formula of ZINBLoss is described as below:

$$\text{ZINBLoss} = \log \pi + (1 - \pi) * \text{NegBinomLoss}$$
$$\text{NegBinomLoss} = -\log P(X = x)$$
$$= -\frac{\log (x + \theta - 1)!}{(x! * (\theta - 1)!) + \theta \log \frac{\theta}{\mu + \theta}},$$

In this formula, π represents the zero-inflation probability, x denotes the expression data, θ signifies the dispersion parameter, and μ signifies the mean parameter for the negative binomial distribution. The range of π falls between 0 and 1, while θ and μ are positive real numbers.

Our model predicts values for π, θ, and μ to apply them in the ZINBLoss function.

3.4   ZINB-Based Graph Autoencoder

For a normalized Laplacian matrix L, we utilized one GCN layer for the encoder, three GCN layers for mean, dropout, and dispersion prediction, respectively, and two fully connected layers for the decoder. The encoder is as follows.

$$Z = Dropout\left(ReLU\left(GraphNorm\left(L \cdot X \cdot W\right)\right)\right), \tag{3}$$

where $Dropout$ is a dropout layer, $ReLU$ is an activation function, $GraphNorm$ [5] is a graph normalization, L is the normalized Laplacian matrix, X is a gene expression, and W is a weight matrix. Here, the number of hidden units is 128. After that, we apply a GCN layer to estimate parameters μ, π, θ in the latent embedded representation Z as follows:

$$M = \exp\left(L \cdot Z \cdot W_\mu\right),$$
$$\Pi = \text{sigmoid}\left(L \cdot Z \cdot W_\pi\right), \tag{4}$$
$$\Theta = \exp\left(L \cdot Z \cdot W_\theta\right),$$

where W_μ, W_π, and W_θ represent the learned weights; M represents the mean parameter matrix, Π is the dropout probability, and Θ describes dispersion, respectively. In this context, the number of hidden units is 1024, and the activation function is chosen to align with the definition of each parameter. We choose the sigmoid function because the dropout probability ranges from 0 to 1. Additionally, we apply the exponential function to ensure non-negative values for both the mean and dispersion. The negative log-likelihood of the ZINB distribution is employed in the definition of the ZINBLoss function, as shown below:

$$L_{ZINB} = -\log(X|\pi, \mu, \theta)$$

Here, two fully connected layers are also utilized to reconstruct the matrix $\hat{X}$ from the M as follows.

$$\hat{X} = ReLU(BatchNorm(f_D(M))),$$

where $\hat{X}$ is a reconstruction matrix from cell embedding, $BatchNorm$ is a Batch Normalization, and f_D consists of two fully connected layers with 1024 hidden nodes and a number of input nodes. Since the gene expression data is a non-negative matrix, we utilize the $ReLU$ function.

In addition, We integrated the normalized gene expression matrix into the imputation matrix $\hat{X}$ for each cell and gene. This integration helps retain the unique characteristics of genes and cells that might otherwise be lost during the imputation process. The formula is as follows:

$$\hat{X} = \hat{X} + BatchNorm(X) + BatchNorm(X^T)^T.$$

By employing Batch Normalization, the data becomes trainable, which aids in maintaining specific representations and preventing overfitting.

Then $\hat{X}$ is utilized for the calculation of the reconstruction loss L_r as follows:

$$L_r = \left\| X - \hat{X} \right\|_2^2$$

3.5   Joint optimization

Here, the joint optimization of the reconstruction and ZINB losses is conducted. We minimize the combined loss function as follows:

$$L = \alpha L_r + (1 - \alpha) L_{ZINB},$$

where α represents a weight coefficient used to balance each loss function, where L_r denotes the reconstruction loss and L_{ZINB} represents the ZINB loss. We utilize Adam [14] to optimize jointly.

After optimization, the output matrix can be treated as an imputed matrix, which can then be employed for subsequent downstream analysis.

4   Experiments

4.1   scRNA-seq data

We conducted a comprehensive performance comparison of our model against various baseline methods using 14 real-world scRNA-seq datasets for cell clustering. The 14 scRNA-seq datasets utilized in our experiments were sourced from recently published papers detailing scRNA-seq experiments and are diverse in terms of species (including mice and humans) and originating organs (such as the brain, liver, and pancreas). Specifically, these datasets exhibit varying characteristics with regard to the number of cells (ranging from 1,014 to 22,770), the number of genes (ranging from 10,160 to 27,202), and the number of distinct cell classes (ranging from 8 to 26). All data are publicly available on GitHub https://github.com/inoue0426/scVGAE-paper.

Table 1: Dataset overview. Accession shows GSE's accession number, link to the box.com, and Seurat is from the Seurat package.

Data	Class	Cell	Gene	Density [%]	Accession	Reference
Baron	14	1937	20125	9.56	GSE84133	[3]
Brosens	11	5829	20697	8.04	GSE168405	[22]
Carey	10	9082	27202	6.95	SCP1903	[23]
cbmc	15	8617	20501	4.67	GSE100866	[19]
Chang	16	22770	10160	2.89	GSE125527	[4]
Fujii	9	2484	14679	26.74	GSE119969	[7]
hcabm40k	8	12000	17369	5.80	Seurat [10]	[19]
Hrvatin	8	7390	19155	7.62	GSE102827	[12]
Jakel	8	8000	21581	5.02	GSE118257	[13]
Jiang	8	1014	25742	24.38	GSE156456	[28]
Manno	26	1977	19531	11.76	GSE76381	[16]
Mingyao	8	6840	13007	9.84	box	[17]
pbmc3k	9	2638	13714	6.19	Seurat [10]	10x Genomics
Xu	14	8196	19988	6.94	GSE189539	[28]

4.2   Baseline

The performance of scVGAE was compared with several state-of-the-art scRNA-seq data imputation methods, including affinity-based, statistic-based, deep learning, and Deep Graph models.

– Markov Affinity-based Graph Imputation of Cells (MAGIC): MAGIC [26] is a widely recognized affinity-based imputation method. It leverages the concept of a Markov random walk on a cell-cell similarity graph to estimate missing values. MAGIC effectively imputes missing data, particularly in single-cell RNA sequencing datasets, by considering the local neighborhood of cells with similar expression profiles.
– Adaptive Local Relevance Analysis (ALRA): ALRA [19] is a sophisticated statistical imputation method that adapts to the local data structure. It is well-suited for capturing intricate gene-gene relationships. By considering the neighboring genes and their interplay, ALRA provides robust imputations that enhance the overall data quality.
– Deep Count Autoencoder (DCA): DCA [6] is another prominent deep learning-based imputation method tailored for scRNA-seq data. It employs autoencoder to model the data's underlying structure and dependencies. During training, DCA minimizes a ZINBLoss to guide the learning process. Once trained, DCA can impute missing values in a way that respects the intricate relationships between genes and cells in the dataset.
– DeepImpute: DeepImpute [2] is a powerful deep learning-based imputation method designed explicitly for scRNA-seq data. It utilizes several neural networks to capture the underlying patterns in the data. DeepImpute optimizes an MSELoss to minimize the discrepancy between observed and imputed values.
– GNNImpute: GNNImpute [30] is an imputation model using graph attention networks (GAT) [27]. Using GAT layers as encoder and linear layers as decoder, this architecture is employed to reconstruct a masked scRNA matrix using MSEloss.

4.3   Implementation details

To implement this model, we utilized the following libraries and versions: torch 2.0.1, torch-cluster 1.6.1+pt20cu117, torch-geometric 2.3.1, torch-scatter 2.1.1+pt20cu117, and torch-sparse 0.6.17+pt20cu117. Detailed implementation information can be found on GitHub https://github.com/inoue0426/scVGAE.

We began by configuring the first GCN layer with 128 units and a dropout rate 0.2. Subsequently, we matched their dimensions with the original gene expression matrix for the GCN layers responsible for handling mean, dispersion, and dropout. The GCN layer dedicated to calculating the mean generated an output, which we passed to a subsequent fully connected layer with 1024 units to reduce its dimension. Following this, we applied a dropout layer with a rate of 0.4. Afterward, we employed another fully connected layer to restore the matrix to its original size, completing the imputation process. We set the parameters α at 0.05 and epochs at 100 during this process. Model optimization was performed using the Adam algorithm [14] with a learning rate 5e-4. These parameter values and the number of layers were defined by leveraging Optuna [1] for parameter tuning.

We kept the parameters for the baseline methods the same as those specified in their original papers. Our experiments were conducted on an Ubuntu server with an NVIDIA A40 GPU with 48 GB of memory. We used the default settings for the initial weights and bias in PyTorch.

4.4   Clustering performance

After applying these techniques, we employed k-means and spectral clustering algorithms with various similarity measures, including linear kernel, polynomial kernel, and cosine similarity, to generate clustering results. Subsequently, we utilized two widely accepted clustering evaluation metrics: the Adjusted Rand Index (ARI), and the Adjusted Mutual Information (AMI), to assess and quantify the effectiveness of our approach in comparison to five other baseline methods. Higher values of these metrics indicate superior clustering performance.

Table 2 compares the clustering performance between scVGAE and baseline methods across 14 scRNA-seq datasets. Each clustering approach was executed ten times to calculate the average, and the highlighted values in the table represent the best results. It is evident from the results that our method consistently outperforms the baseline clustering methods in terms of clustering performance. In addition, our analysis employs the t-test to compare scVGAE with other methods. The results indicate that all comparisons yield p-values below 0.05, except for the AMI metric in the case of MAGIC. This consistency in achieving statistically significant results across various metrics further underscores the robustness and superiority of scVGAE.

4.5   Visualization

Here, we visualize the result using UMAP and TSNE using the Scanpy [29] package. Fig 2 shows the representation of UMAP space using Brosens data. When compared to Raw and scVGAE, the findings indicate no substantial variance.

Table 2: Classification performance comparison on 14 scRNA-seq Datasets. The best values among the compared methods are highlighted in bold.

Metric	Dataset	scVGAE	Original	MAGIC	ALRA	DeepImpute	DCA	GNNImpute
ARI	Baron	0.7278	0.5606	0.7657	0.4235	**0.8623**	0.6392	0.5165
	Brosens	**0.5605**	0.5083	0.4685	0.4967	0.1991	0.3775	0.3746
	Carey	**0.7978**	0.6598	0.7437	0.6510	0.3750	0.4799	0.7023
	cbmc	0.5937	0.4358	**0.6622**	0.4699	0.3348	0.4432	0.5632
	Chang	**0.1840**	0.1747	0.1620	0.1351	0.1217	0.1502	0.0956
	Fujii	**0.4792**	0.3565	0.4439	0.3462	0.1403	0.1978	0.3041
	hcabm40k	**0.0940**	0.0371	0.0347	0.0895	0.0246	0.0352	0.0370
	Hrvatin	**0.7914**	0.7119	0.7771	0.6021	0.4492	0.6681	0.6282
	Jakel	**0.6108**	0.5043	0.4117	0.4419	0.3589	0.4634	0.4562
	Jiang	0.2811	0.3421	0.2871	**0.3435**	0.1379	0.2612	0.2284
	Manno	**0.3095**	0.2929	0.2123	0.2858	0.1269	0.1726	0.1883
	Mingyao	**0.2373**	0.2303	0.1778	0.2020	0.1629	0.1855	0.1735
	pbmc3k	**0.6933**	0.6381	0.6501	0.6370	0.2911	0.5122	0.5583
	Xu	**0.7357**	0.6749	0.6995	0.6629	0.3445	0.4711	0.6531
AMI	Baron	0.7896	0.7500	**0.8356**	0.7143	0.7691	0.7567	0.6641
	Brosens	**0.6687**	0.6355	0.6494	0.6296	0.3557	0.5506	0.5434
	Carey	**0.8146**	0.7213	0.7782	0.7088	0.5577	0.6164	0.7266
	cbmc	0.6855	0.6367	**0.7462**	0.6437	0.5244	0.6156	0.6258
	Chang	**0.3244**	0.3135	0.3023	0.2558	0.2266	0.2855	0.1941
	Fujii	**0.5716**	0.4767	0.5658	0.4749	0.2289	0.3216	0.4276
	hcabm40k	**0.1343**	0.0525	0.0517	0.1311	0.0420	0.0532	0.0498
	Hrvatin	0.8590	0.8514	**0.8677**	0.7979	0.6888	0.8155	0.6856
	Jakel	**0.6627**	0.6316	0.5875	0.5986	0.4643	0.5718	0.5742
	Jiang	0.4789	0.5101	0.4964	**0.5360**	0.2456	0.4017	0.3974
	Manno	**0.4735**	0.4466	0.4301	0.4574	0.2995	0.3779	0.3680
	Mingyao	**0.3764**	0.3224	0.2942	0.3105	0.2555	0.3039	0.2646
	pbmc3k	**0.7732**	0.7248	0.7649	0.7389	0.4693	0.6661	0.6615
	Xu	**0.7929**	0.7396	0.7814	0.7414	0.4942	0.5811	0.7011

This can likely be attributed to the efficacy of incorporating a normalized matrix in preserving cell information, thus rendering these two approaches quite analogous. In the case of ALRA and MAGIC, both exhibit difficulties depicting SS1. ALRA employs SVD to identify biological and technological zeroes while addressing the technological ones. Sometimes, this fails to identify these zeroes, disrupting distribution. MAGIC also influences the distribution, which may be attributed to its utilization of K-nearest neighbors (KNN) for constructing a similarity matrix, relying on local information and potentially giving rise to smaller clusters. In the case of DCA and DeepImpute, their results suggest that the imputation techniques deployed may erode cell cluster information. Utilizing a feed-forward network with a ZINB loss function may not be ideal for cell clustering. GNNImpute, on the other hand, appears to induce a comprehensive alteration of the distribution. While the results may seem distinguishable, the ARI does not yield favorable outcomes, and UMAP embedding does not straightforwardly represent cell clustering results.

4.6   Ablation Study

In the context of this experiment, we conducted an ablation study to systematically investigate the impact of individual components within the scVGAE method. Specifically, we performed ablations in five distinct scenarios: 1) Excluding ZINBLoss, 2) Excluding reconstruction loss, 3) Excluding the addition of normalization, 4) Excluding both ZINBLoss and the addition of normalization, and 5) Excluding ZINBLoss and including normalization. Table 3 presents the average values of ARI and AMI obtained across 14 datasets for these five distinct cases when utilizing the scVGAE method. In summary, all components of the scVGAE method are deemed reasonable and practical.
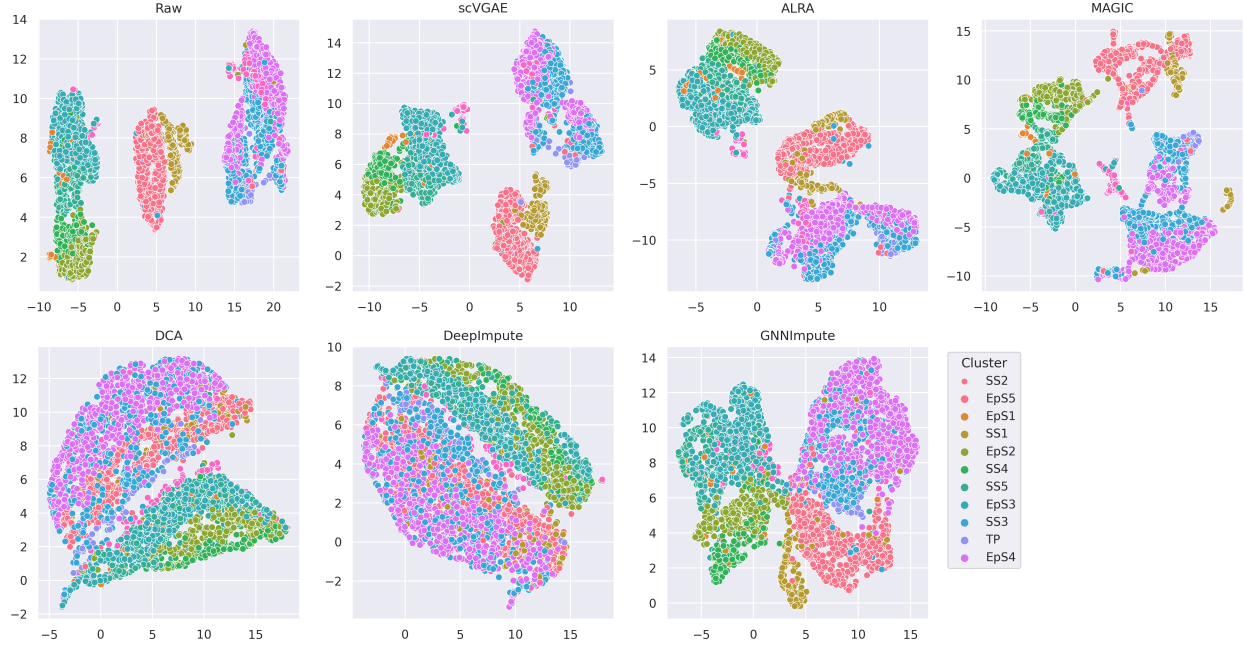
Fig. 2: Visualization of scRNA Imputation Results Using 7 Methods with UMAP

Table 3: Investigating the Impact of Components: Component-wise Evaluation in scVGAE

Components	ARI	NMI
without ZINB	0.4600	0.5788
without Recon	0.4573	0.5592
without adding	0.2398	0.3588
without ZINB&adding	0.2416	0.3529
without Recon&adding	0.1811	0.2815
scVGAE	**0.5069**	**0.6004**

Notes: ZINB: ZINB Loss, Recon: Reconstruction Loss, adding: adding cell-wise and gene-wise normalization.

4.7   Speed Assessment

We also evaluated the speed of imputation using the following references: Baron (1937, 20125), Brosens (5829, 20697), Carey (9082, 27202), hcabm40k (12000, 17369), and Chang (22770, 10160). Figure 3 presents the performance of six models on five datasets for running speed. It is worth noting that MAGIC, known for its simplicity as a multiplication model, proved to be the fastest overall. However, for the Chang dataset, scVGAE demonstrated the highest speed. Our model ranked second fastest among the models, except for the Carey dataset. The Carey dataset, being the most extensive in terms of genes, and the Chang dataset, having the most substantial cell count, stand out as exceptions. These findings indicate that our model performs more efficiently on datasets with a high cell count but lags when dealing with datasets rich in gene information. Nevertheless, scVGAE stands as a reasonable-speed model when compared to existing models.

5   Conclusion

We present scVGAE, a novel method for imputing and clustering single-cell RNA sequencing data. It employs a variational graph convolutional autoencoder (VGAE) with ZINBLoss, effectively preserving the intrinsic topological information derived from cell-cell similarity. A linear kernel is utilized to create this similarity, forming the basis for constructing the affinity matrix for input into the graph convolutional networks. Furthermore, our ongoing research involves exploring additional variations, including incorporating functions like the RBF Kernel and cosine similarity.
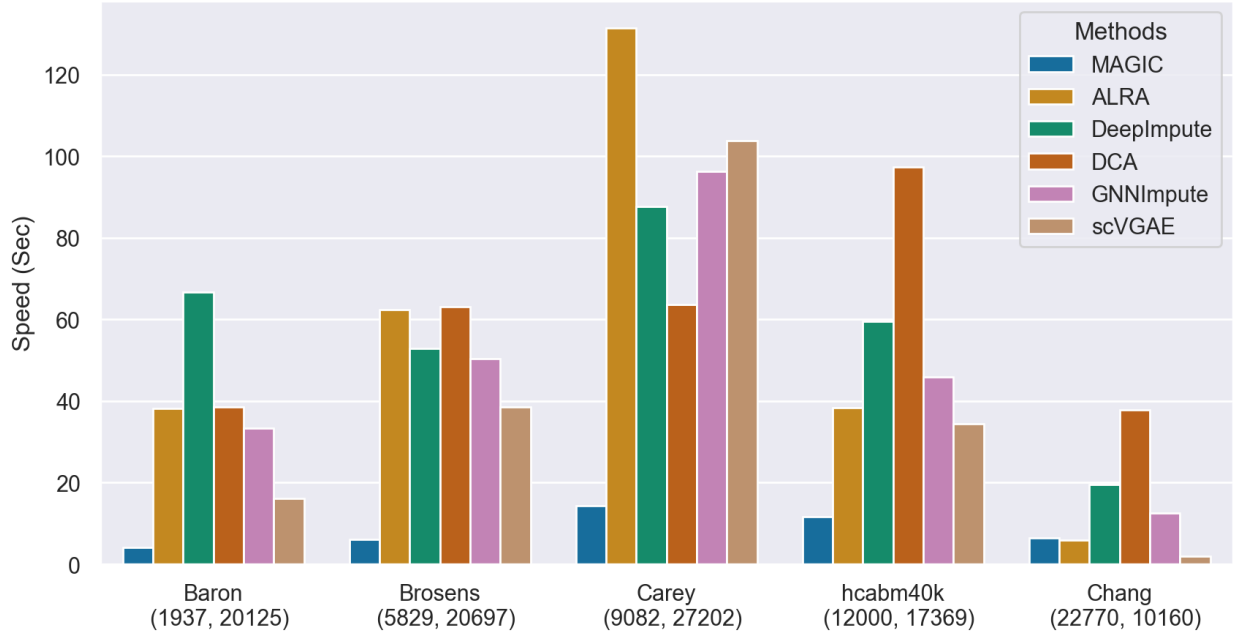
Fig. 3: Effect of Data Size on Processing Time: Comparative Evaluation of Multiple Models. The x-axis displays dataset name and dimensions, while the y-axis represents speed in seconds.

The ZINBLoss function from the encoder is applied to ensure that the matrix conforms to the original distribution, as scRNA data often follows a zero-inflated negative binomial distribution, as observed in other ZINBLoss-based methods that have delivered exceptional results. In addition to ZINBLoss, scVGAE also incorporates MSELoss to maintain the distribution of the original matrix. This dual-loss strategy helps preserve valuable information and enhance imputation results, as evidenced in the ablation study.

Overall, scVGAE demonstrates remarkable performance in clustering scRNA-seq data. These findings underscore the robustness of generative models with ZINBLoss for the imputation task. This result also suggests that exploring new generative models, such as diffusion and graph diffusion, may prove effective, as no such methods currently exist. This opens up exciting avenues for future research in this direction, and we plan to investigate these possibilities as their next research steps.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2623–2631 (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. Baron, M., Veres, A., Wolock, S.L., Faust, A.L., Gaujoux, R., Vetere, A., Ryu, J.H., Wagner, B.K., Shen-Orr, S.S., Klein, A.M., Melton, D.A., Yanai, I.: A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. Cell Systems **3**(4), 346–360.e4 (Oct 2016). `https://doi.org/10.1016/j.cels.2016.08.011`
4. Boland, B.S., He, Z., Tsai, M.S., Olvera, J.G., Omilusik, K.D., Duong, H.G., Kim, E.S., Limary, A.E., Jin, W., Milner, J.J., et al.: Heterogeneity and clonal relationships of adaptive immune cells in ulcerative colitis revealed by single-cell analyses. Science immunology **5**(50), eabb4432 (2020)
5. Cai, T., Luo, S., Xu, K., He, D., Liu, T.y., Wang, L.: Graphnorm: A principled approach to accelerating graph neural network training. In: International Conference on Machine Learning. pp. 1204–1215. PMLR (2021)
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. Fujii, M., Matano, M., Toshimitsu, K., Takano, A., Mikami, Y., Nishikori, S., Sugimoto, S., Sato, T.: Human Intestinal Organoids Maintain Self-Renewal Capacity and Cellular Diversity in Niche-Inspired Culture Condition. Cell Stem Cell 23(6), 787–793.e6 (Dec 2018). https://doi.org/10.1016/j.stem.2018.11.016, https://www.cell.com/cell-stem-cell/abstract/S1934-5909(18)30552-6

8. 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)

9. Gu, H., Cheng, H., Ma, A., Li, Y., Wang, J., Xu, D., Ma, Q.: scgnn 2.0: a graph neural network tool for imputation and clustering of single-cell rna-seq data. Bioinformatics 38(23), 5322–5325 (2022)

10. Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W.M., Zheng, S., Butler, A., Lee, M.J., Wilk, A.J., Darby, C., Zager, M., et al.: Integrated analysis of multimodal single-cell data. Cell 184(13), 3573–3587 (2021)

11. Heath, J.R., Ribas, A., Mischel, P.S.: Single-cell analysis tools for drug discovery and development. Nature reviews Drug discovery 15(3), 204–216 (2016)

12. Hrvatin, S., Hochbaum, D.R., Nagy, M.A., Cicconet, M., Robertson, K., Cheadle, L., Zilionis, R., Ratner, A., Borges-Monroy, R., Klein, A.M., et al.: Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex. Nature neuroscience 21(1), 120–129 (2018)

13. Jäkel, S., Agirre, E., Mendanha Falcão, A., Van Bruggen, D., Lee, K.W., Knuesel, I., Malhotra, D., Ffrench-Constant, C., Williams, A., Castelo-Branco, G.: Altered human oligodendrocyte heterogeneity in multiple sclerosis. Nature 566(7745), 543–547 (2019)

14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

16. La Manno, G., Gyllborg, D., Codeluppi, S., Nishimura, K., Salto, C., Zeisel, A., Borm, L.E., Stott, S.R.W., Toledo, E.M., Villaescusa, J.C., Lönnerberg, P., Ryge, J., Barker, R.A., Arenas, E., Linnarsson, S.: Molecular Diversity of Midbrain Development in Mouse, Human, and Stem Cells. Cell 167(2), 566–580.e19 (Oct 2016). https://doi.org/10.1016/j.cell.2016.09.027

17. Lakkis, J., Schroeder, A., Su, K., Lee, M.Y.Y., Bashore, A.C., Reilly, M.P., Li, M.: A multi-use deep learning method for CITE-seq and single-cell RNA-seq data integration with cell surface protein prediction and imputation. Nature Machine Intelligence 4(11), 940–952 (Nov 2022). https://doi.org/10.1038/s42256-022-00545-w, https://www.nature.com/articles/s42256-022-00545-w

18. Li, W.V., Li, J.J.: An accurate and robust imputation method scimpute for single-cell rna-seq data. Nature communications 9(1), 997 (2018)

19. Linderman, G.C., Zhao, J., Roulis, M., Bielecki, P., Flavell, R.A., Nadler, B., Kluger, Y.: Zero-preserving imputation of single-cell rna-seq data. Nature communications 13(1), 192 (2022)

20. Luecken, M.D., Theis, F.J.: Current best practices in single-cell rna-seq analysis: a tutorial. Molecular systems biology 15(6), e8746 (2019)

21. Qi, J., Zhou, Y., Zhao, Z., Jin, S.: Sdimpute: a statistical block imputation method based on cell-level and gene-level information for dropouts in single-cell rna-seq data. PLoS Computational Biology 17(6), e1009118 (2021)

22. Rawlings, T.M., Makwana, K., Taylor, D.M., Molè, M.A., Fishwick, K.J., Tryfonos, M., Odendaal, J., Hawkes, A., Zernicka-Goetz, M., Hartshorne, G.M., et al.: Modelling the impact of decidual senescence on embryo implantation in human endometrial assembloids. Elife 10, e69603 (2021)

23. Strieder-Barboza, C., Flesher, C.G., Geletka, L.M., Delproposto, J.B., Eichler, T., Akinleye, O., Ky, A., Ehlers, A.P., O'Rourke, R.W., Lumeng, C.N.: Single-nuclei Transcriptome of Human AT Reveals Metabolically Distinct Depot-Specific Adipose Progenitor Subpopulations (Jun 2022). https://doi.org/10.1101/2022.06.29.496888, https://www.biorxiv.org/content/10.1101/2022.06.29.496888v1

24. Tian, T., Min, M.R., Wei, Z.: Model-based autoencoders for imputing discrete single-cell rna-seq data. Methods 192, 112–119 (2021)

25. Tran, D., Tran, B., Nguyen, H., Nguyen, T.: A novel method for single-cell data imputation using subspace regression. Scientific Reports 12(1), 2697 (2022)

26. Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A.J., Burdziak, C., Moon, K.R., Chaffer, C.L., Pattabiraman, D., et al.: Recovering gene interactions from single-cell data using data diffusion. Cell 174(3), 716–729 (2018)

27. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)

28. Wang, Z., Shao, X., Wang, K., Lu, X., Zhuang, L., Yang, X., Zhang, P., Yang, P., Zheng, S., Xu, X., et al.: Single-cell analysis reveals a pathogenic cellular module associated with early allograft dysfunction after liver transplantation. bioRxiv pp. 2022–02 (2022)

29. Wolf, F.A., Angerer, P., Theis, F.J.: Scanpy: large-scale single-cell gene expression data analysis. Genome biology 19, 1–5 (2018)

30. Xu, C., Cai, L., Gao, J.: An efficient scrna-seq dropout imputation method using graph attention network. BMC bioinformatics 22, 1–18 (2021)

31. Xu, L., Xu, Y., Xue, T., Zhang, X., Li, J.: Adimpute: an imputation method for single-cell rna-seq data based on semi-supervised autoencoders. Frontiers in Genetics 12, 739677 (2021)

32. Yu, Z., Lu, Y., Wang, Y., Tang, F., Wong, K.C., Li, X.: Zinb-based graph embedding autoencoder for single-cell rna-seq interpretations. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 4671–4679 (2022)
33. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. AI open **1**, 57–81 (2020)