BMC Bioinformatics

RESEARCH

Open Access

SAE-Impute: imputation for single-cell data via subspace regression and auto-encoders

Liang Bai[1], Boya Ji[1*] and Shulin Wang[1*]

*Correspondence: byj@hnu.edu.cn; books@hnu.edu.cn

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

Abstract

Background: Single-cell RNA sequencing (scRNA-seq) technology has emerged as a crucial tool for studying cellular heterogeneity. However, dropouts are inherent to the sequencing process, known as dropout events, posing challenges in downstream analysis and interpretation. Imputing dropout data becomes a critical concern in scRNA-seq data analysis. Present imputation methods predominantly rely on statistical or machine learning approaches, often overlooking inter-sample correlations.

Results: To address this limitation, We introduced SAE-Impute, a new computational method for imputing single-cell data by combining subspace regression and autoencoders for enhancing the accuracy and reliability of the imputation process. Specifically, SAE-Impute assesses sample correlations via subspace regression, predicts potential dropout values, and then leverages these predictions within an autoencoder framework for interpolation. To validate the performance of SAE-Impute, we systematically conducted experiments on both simulated and real scRNA-seq datasets. These results highlight that SAE-Impute effectively reduces false negative signals in single-cell data and enhances the retrieval of dropout values, gene-gene and cell-cell correlations. Finally, We also conducted several downstream analyses on the imputed single-cell RNA sequencing (scRNA-seq) data, including the identification of differential gene expression, cell clustering and visualization, and cell trajectory construction.

Conclusions: These results once again demonstrate that SAE-Impute is able to effectively reduce the droupouts in single-cell dataset, thereby improving the functional interpretability of the data.

Keywords: Single-cell RNA-seq, Dropout events, Imputation, Subspace regression, Autoencoders.

Background

The rapid advancement of single-cell RNA sequencing (scRNA-seq) technology has revolutionized the study of intercellular heterogeneity and dynamics in complex tissues [1]. However, scRNA-seq data presents unique challenges compared to bulk RNA sequencing (RNA-seq), such as increased sparsity leading to disproportionate distortions in relative transcript abundance and gene expression [2]. Additionally, the dropout phenomenon is a significant obstacle, caused by the low number of RNA molecules in a single cell and randomness in gene expression, particularly during sequencing [3]. This

results in low RNA capture efficiency, amplification failure, and many expressed genes being measured as zero values. These dropout events primarily occur due to sequencing technology limitations, which only capture a small amount of initial mRNA in a single cell, leading to low sequencing depth and amplification failure. It is crucial to estimate zero expression values introduced by dropouts and sequencing errors since downstream analysis of scRNA-seq heavily relies on expression measurement accuracy [4].

In recent years, there has been a surge of proposed Single-cell RNA-sequencing data imputation method to address the challenge of redundant zeros in scRNA-seq data. These methods can be classified into four types.

The first type is model-based imputation, which directly employs a probabilistic model to interpolate sparsity modeling. For example, SAVER, proposed by Huang [5], is a method for scRNA-seq data expression imputation based on UMI counts. They assume that each expression measure is recovered by estimating prior parameters following a Poisson-Gamma mixture distribution, also known as the negative binomial model. This is followed by an empirical Bayes-like method of Poisson-LASSO regression for gene counts. Another model-based method ScRecover [6], is an expression recovery method for scRNA-seq data based on a zero-inflated negative binomial model. This method predicts the number of genes that are expressed in a cell by estimating the probability of "dropout-zeros" for each gene in each cell. The ScImpute method learns the dropout probability of each gene in each cell based on a mixture model and then automatically identifies values in gene expression affected by dropout events with the help of the same genes in other similar cells [7]. Computational methods based on model imputation generally yield fewer false positives, but this depends largely on the diversity of cell types in the sample. The more cell types there are, the more cell type-specific labeling can be reduced by model-based methods. VIPER [8] employs statistical models to estimate missing values, emphasizing the preservation of variability in the data. This approach enhances the consistency of the imputed data with actual biological conditions. SDImpute [9] employs statistical models to fully utilize cell-level and gene-level information for imputation. This approach captures the structural features of the data by analyzing the relationships between cells and genes.

The second type of method is the smoothing-based imputation method. For instance, DrImpute [10] is a clustering algorithm that identifies similar cells and performs data imputation through the average expression level of the expression values of similar cells. On the other hand, MAGIC [11] first constructs a distance matrix by calculating the distance between every two cells, then converts the distance matrix into an affinity matrix using the Gaussian kernel, and constructs an affinity map based on the Markov chain after normalization to restore the dropout value. However, these methods require the identification of structures in the data that can be used to predict dropout expression levels, such as similar cell expression, affinity relationship, or neighbor asymptotic relationship, and then use smoothing algorithms for imputation. As a result, these methods can introduce a significant number of false positive signals. It is crucial to note that while these methods can be effective, they rely on assumptions that may not always hold in practice.

The third type of imputation method is based on deep learning [12]. DCA is an interpolation algorithm that utilizes a deep counting autoencoder, which designs a unique

loss function for scRNA-seq data [13]. It utilizes the noise model and the mean value of the distribution parameters to gene expression in an unsupervised way, according to the input gene expression data. The expression matrix is then reconstructed and output. scVI is a probabilistic representation analysis tool for single-cell gene expression that utilizes a hierarchical Bayes model design algorithm of deep neural network [14]. By compressing each cell and its gene expression, and using the decoder to map the latent space to the posterior of the gene expression distribution parameters, estimates are interpolated. AutoImpute [15] is an imputation algorithm based on autoencoders and sparse representation matrices, which solves dropout events by learning the inherent distribution of data. DeepImpute [16] is an interpolation algorithm based on deep neural networks that utilizes the divide-and-conquer approach to reduce the complexity of the algorithm by learning small-scale problems and fine-tuning the sub-neural network, resulting in effective interpolation. scIGANs [17] is an algorithm based on an adversarial neural network that converts the expression profile of each cell into an image. The process of interpolating matrix data is the image inpainting process, and it is also a method of interpolating cells of the same type using nearest neighbors. However, most of these algorithms rely on deep learning models for image restoration and are limited by the lack of real interpolation labels for single-cell data, making it challenging to verify and train for accuracy like images.

The fourth type of methods is low-rank matrix imputation methods. ALRA is an imputation method that uses an adaptive threshold low-rank approximation via singular value decomposition (SVD) and exploits the non-negativity and correlation of the matrix representation for imputation [18]. mcImpute [19] is an imputation algorithm based on low-rank matrix completion, which uses the kernel norm minimization algorithm to solve the non-convex optimization problem of the observation matrix to restore gene dropouts. These methods utilize matrix decomposition to distinguish "true zeros" from "dropout zeros". However, "dropout zeros" may not fully conform to some matrix characteristics, so the interpolation effect needs to be verified.

A systematic evaluation conducted by Hou et al. [20] on single-cell RNA sequencing (scRNA-seq) imputation methods reveals that the majority of these techniques outperform non-imputation methods in recovering gene expression, as observed in bulk RNA sequencing. However, it is notable that while these methods enhance gene expression recovery, they generally do not enhance the performance of downstream analyses, such as clustering and trajectory analysis, when compared to the absence of imputation. Thus, caution is advised in their application. Furthermore, the evaluation demonstrates significant variability in the performance of these methods across different evaluation aspects. Cheng et al. [21] demonstrated that various imputation methods exhibit differing effects across distinct datasets, indicating that imputation may be dataset-specific and that challenges in imputation persist.

Here, we present a novel approach, SAE-Impute, designed to accurately fill in dropout values within single-cell data. SAE-Impute accurately estimates missing values by exploiting the low-dimensional subspace structure of the data. This method effectively captures the intrinsic relationships within the data through a linear combination of observations, thereby enhancing the accuracy of interpolation. The autoencoder learns feature representations from the data, while subspace regression

adeptly addresses missing patterns. This combination not only improves the model's robustness but also enhances its adaptability. The method combines a subspace regression model with an autoencoder model. A subspace regression method was employed to address missing values within the dataset. This method leverages the low-dimensional subspace structure of the data to estimate missing values through a linear combination of observed values, thus enhancing the precision of missing value imputation. Additionally, our innovative use of feature engineering and data preprocessing techniques sets our method apart from existing algorithms, enabling it to effectively handle the complexities and challenges inherent in single-cell RNA sequencing data. Just like data heterogeneity and technical noise, single-cell RNA sequencing data is derived from individual cells, leading to significant variations in gene expression among different cells. Factors such as the type, state, and environment of each cell can influence its RNA expression profile, complicating the analysis. Additionally, the single-cell sequencing technology itself can introduce noise, including biases in library construction and errors during sequencing. These sources of noise may hinder the extraction of true signals and compromise the reliability of the data analysis results. Subsequently, an autoencoder is integrated to capture the nonlinear characteristics of the data, consequently enhancing the accuracy and resilience of the interpolation process, and the data processed using the subspace regression model can decrease the computational complexity of the autoencoder. The fusion of these methodologies enables more effective management of the prevalent missing value issue encountered in single-cell RNA sequencing data, thereby enhancing data integrity and reliability. Firstly, we use the subspace regression model to identify possible dropout values, ensuring that only highly correlated information is used to impute true dropout values by grouping genes with similar patterns into smaller groups. We then incorporate the predicted values obtained by the subspace regression model into the autoencoder, leveraging its inherent advantages to train and find the actual dropout values alongside the original data. The subspace regression model effectively preserves the correlation between cells and minimizes the introduction of false positive signals and noise interference [22], while the autoencoder is capable of handling diverse cell distributions and expression patterns, resulting in faster and more scalable imputation.

Results

In order to assess the effectiveness of imputation techniques, we conducted a comparison of the SAE-impute method against six other methods across four popular models. These included SAVER [5], ScRecover [6], ScImpute [7], MAGIC [11], AutoImpute [15], ALRA [18] and scGCL [12]. SAVER, ScRecover, and ScImpute are model-based interpolation methods, while MAGIC is a smooth-based interpolation method. AutoImpute is a machine-learning-based interpolation method, and ALRA is a low-rank matrix interpolation method.

SAE-Impute clustering performance evaluation

To evaluate the performance of SAE-impute in identifying dropout values, we use simulated data with known ground truth. In our experimental analysis, we first calculate the clustering performance evaluation using the Adjusted Rand Index (ARI) by clustering the cells with the Louvain method [23]. ARI is calculated by comparing the Louvain clustering results with known cell labels, with a value range of −1 to 1, where 1 represents perfect consistency and 0 represents random partition. The ARI formula is defined as follows:

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}, RI = \frac{a+b}{\binom{n}{2}} \tag{1}$$

where C represents the actual category division, a is the number of instance pairs that are classified into the same class in C and the same cluster in the Louvain result, b is the number of instance pairs that are divided into different categories in C and into different clusters in the Louvain result, and n is the total number of instances.

The SAE-Impute model leverages imputation weight labels consisting of data filled with the predicted value I_P of the subspace regression model to improve the effectiveness of missing value imputation. This approach not only preserves the correlation between cells but also helps to mitigate the issue of overfitting. Additionally, the weight label associated with the prediction matrix enhances the clustering effect of the model, as evidenced in Fig 1. The results demonstrate that the SAE-Impute model outperforms SAVER, Screcover, ScImpute, MAGIC, AutoImpute, ALRA and scGCL methods in terms of clustering performance across four different datasets.

SAE-Impute effectively identifies dropout values

The F1 score [24] is a metric used to evaluate the performance of a classification model, which takes into account both the precision and recall of the model. And the F1 score is a measure of a model's accuracy and recall, weighted by their harmonic mean. It ranges

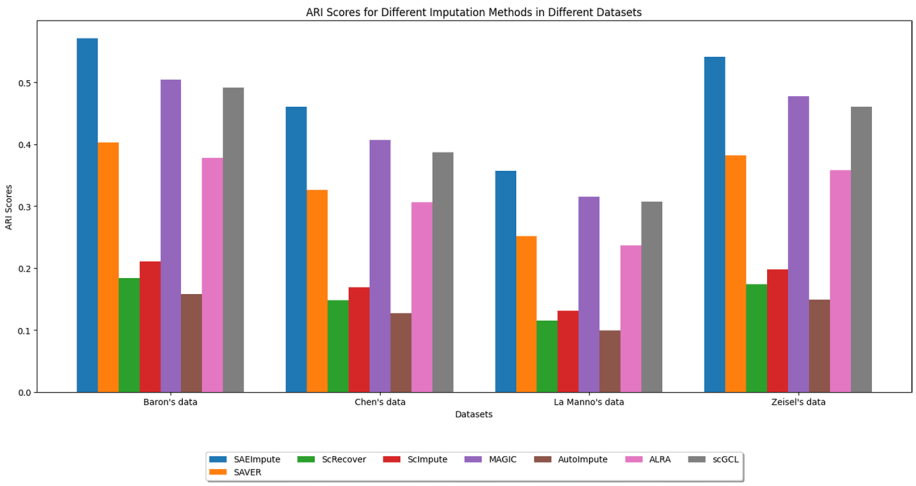

Fig. 1 Performance comparison of SAE-Impute and other methods for cell clustering on four different single-cell data. Adjusted Rand Index: ARI

Table 1 The F1 scores of imputation methods on different dropout datasets

Methods	Dropout rate: 30%	Dropout rate: 40%	Dropout rate: 50%	Dropout rate: 60%	Dropout rate: 70%
SAEImpute	0.34	0.51	0.58	0.61	0.65
	0.32	0.48	0.53	0.57	0.63
	0.30	0.45	0.52	0.56	0.61
	0.36	0.52	0.61	0.61	0.64
SAVER	0.39	0.47	0.54	0.55	0.57
	0.36	0.45	0.50	0.52	0.54
	0.34	0.42	0.48	0.49	0.53
	0.41	0.50	0.56	0.53	0.55
AutoImpute	0.36	0.31	0.27	0.22	0.19
	0.35	0.29	0.24	0.19	0.17
	0.32	0.26	0.23	0.18	0.15
	0.38	0.30	0.26	0.21	0.18
MAGIC	0.42	0.54	0.55	0.59	0.64
	0.38	0.50	0.53	0.55	0.60
	0.36	0.49	0.51	0.54	0.59
	0.40	0.55	0.57	0.59	0.63
ScRecover	0.58	0.49	0.25	0.17	0.11
	0.58	0.48	0.24	0.17	0.09
	0.55	0.46	0.23	0.15	0.08
	0.57	0.49	0.24	0.13	0.10
ScImpute	0.31	0.22	0.13	0.11	0.10
	0.29	0.20	0.12	0.11	0.10
	0.27	0.19	0.11	0.10	0.09
	0.29	0.23	0.12	0.11	0.10
scGCL	0.41	0.53	0.55	0.58	0.62
	0.37	0.51	0.52	0.53	0.60
	0.35	0.48	0.51	0.52	0.57
	0.39	0.53	0.56	0.58	0.61

from 0 (worst) to 1 (best) and indicates how well the model performs. A higher F1 score indicates better model performance. The F1 score formula is defined as follows:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{precision + recall} \tag{2}$$

Among these metrics, *precision* measures the proportion of true positives among all positive predictions made by the model, and *recall* measures the proportion of true positives among all actual positive cases.

Drawing an analogy to single-cell RNA-sequencing data interpolation, the F1 score can be utilized to assess the accuracy of the interpolation and the reconstruction effect of the original data. Specifically, in a single-cell dataset with numerous missing values, the F1 score of the interpolation method can reflect the reliability and quality of the interpolated dataset. In single-cell imputation, data with a missing rate exceeding 70% is typically considered to be of no practical value.

Table 1 and Fig. 2 present the average F1 score across four datasets from Barron et al [25]. Chen et al [26]. Ramanno et al [27]. and Zeisel et al [28]. While our method may not

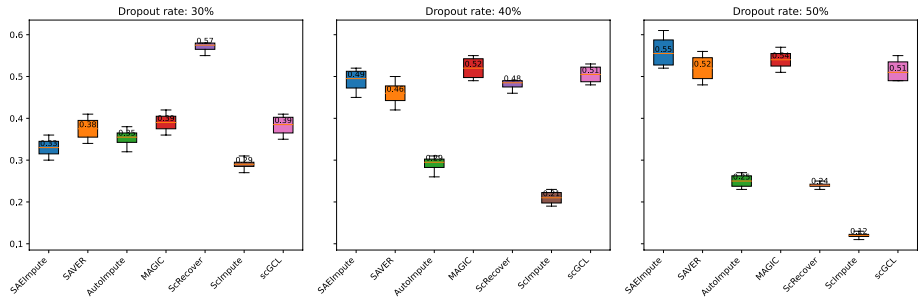

Fig. 2 Boxplots showing F1 scores of imputation methods on different dropout level datasets

achieve the highest accuracy when the missing rate is low, its effectiveness significantly improves as the missing rate increases. This improvement is particularly significant when considering real-world situations where the missing rate reaches approximately 60%.

And we assess the performance of multiple methods across four datasets by computing the mean absolute error (MAE) [24] of genes influenced by dropout events. To enhance the clarity of comparative visualization, we conducted experiments using data from La Manno et al. [27] And Fig. 3 illustrates that our method exhibits a lower absolute error, suggesting its strong interpolation capabilities on these datasets.

SAE-impute improves gene-to-gene and cell-to-cell correlations

The average correlation quantifies the similarity between the imputed data and the actual data, serving as a critical criterion for assessing the effectiveness of the imputation method. By comparing the correlation between the imputed results and the true expression matrix, researchers can evaluate the model's performance.

To quantify the similarity between the imputed and original landscapes, we calculate the distance correlation index ($dCor$) [29] for each imputed landscape generated by $t-SNE$. Given X and Y as the 2D representations of the raw and imputed data, $dCor$ is calculated as $dCor = \frac{dCov(X,Y)}{\sqrt{dVar(X)dVar(Y)}}$, where $dCov(X, Y)$ is the distance covariance between X and Y, Var is the variances. Specifically, this method calculates the pairwise distance of X by computing the distance between each element of X, generating a square matrix for each pair of cells. Next, it calculates the pairwise distance of Y. Finally, it compares the two matrices and obtains the distance correlation using the formula above.

We calculated the distance correlation between the raw and imputed data using the first two components obtained from $t-SNE$. A higher correlation value indicates a greater similarity between the estimated and original landscapes. Our method demonstrates a high average correlation (Fig. 4). A higher average correlation typically indicates that the imputation method is more effective in recovering missing values.

SAE-Impute enhances the differential expression analysis

Identifying differentially expressed genes is a crucial step in analyzing single-cell RNA-sequencing data, as it enables the discovery of driver genes within cells and facilitates the diagnosis and analysis of diseased cells. In this study, we compared the performance

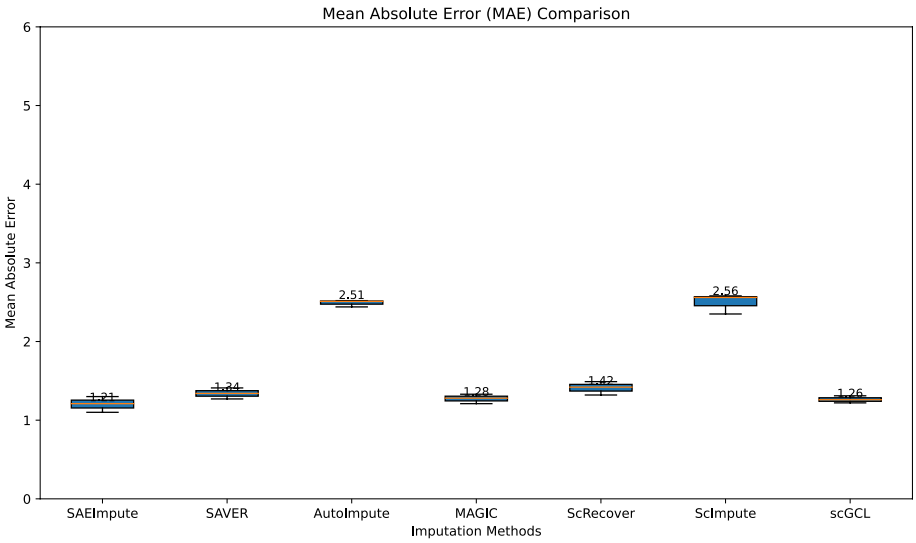

Fig. 3 Assessment of SAEImpute, SAVER, AutoImpute, MAGIC, ScRecover, and ScImpute through simulation studies. The mean absolute error (MAE) was computed by comparing the estimated data with the complete data. Across all analyses, SAEImpute consistently exhibited lower MAE values compared to the other methods

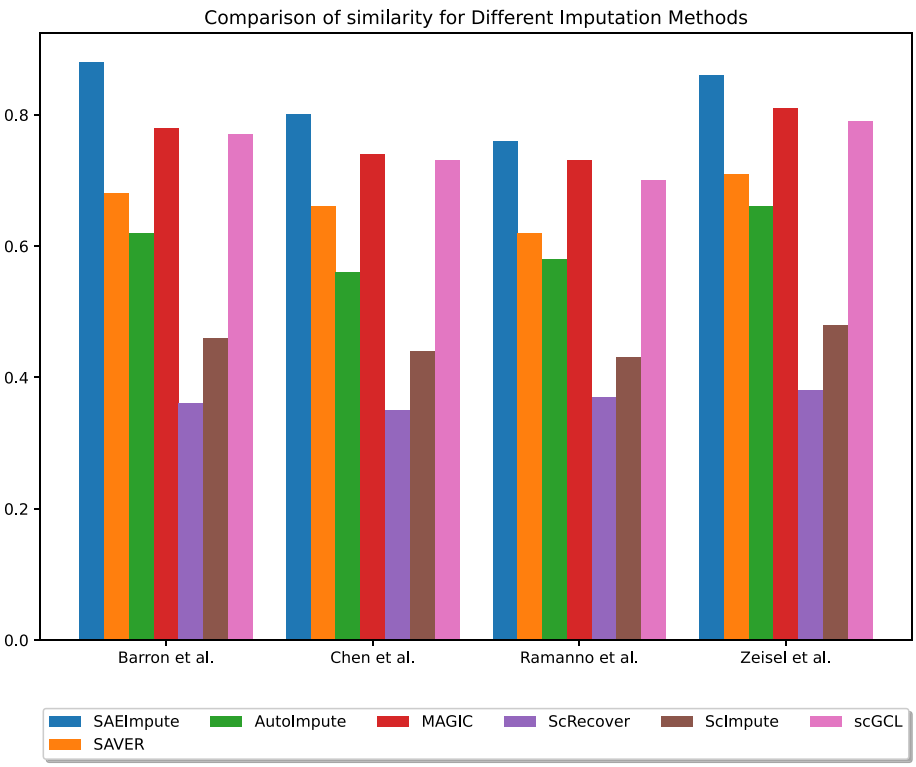

Fig. 4 Transcriptome landscape similarity

of various estimation methods using simulated data. To detect differentially expressed genes in both real and estimated single-cell sequencing data, we utilized the Wilcoxon

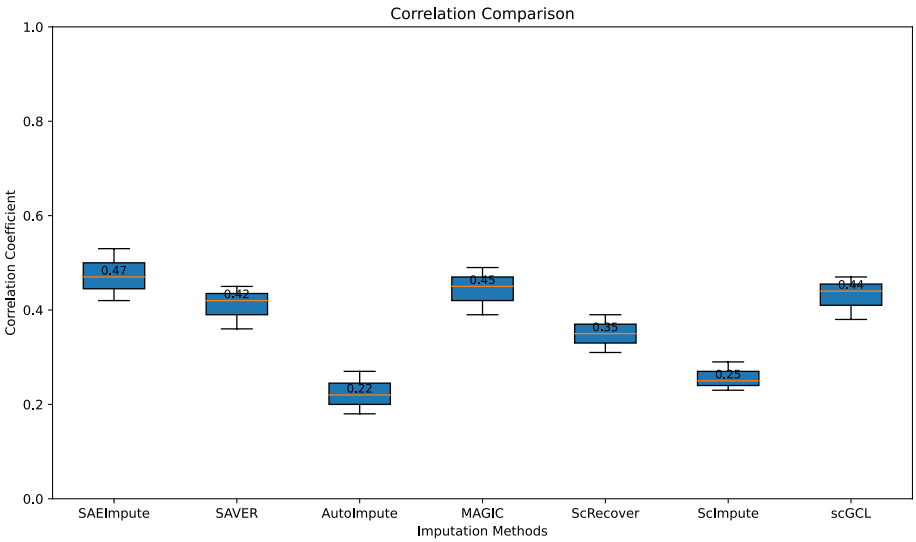

Fig. 5 Assessment of SAEImpute, SAVER, AutoImpute, MAGIC, ScRecover, and ScImpute through simulation studies. The correlation coefficient is computed by comparing the imputed data with the complete data. Our method consistently exhibits a higher correlation coefficient compared to the other methods in all analyses

rank sum test [30]. To enhance the clarity of comparative visualization, we conducted experiments using data from La Manno et al. [27].

The assessment of SAEImpute, SAVER, AutoImpute, MAGIC, ScRecover, and ScImpute through simulation studies is depicted in Fig. 5. The correlation coefficient is calculated by comparing the imputed data with the complete data. Our method consistently demonstrates a higher correlation coefficient compared to the other methods across all analyses.

SAE-Impute facilitates the estimation of cell trajectories

Trajectory inference methods uncover the dynamic development of cells, thereby enhancing the identification of new dynamic cell subpopulations [31]. However, missing events might impede the construction of cellular pseudotime trajectories. Hence, evaluating the performance of interpolation methods on cell trajectory inference can better illustrate the potential capabilities of such methods for downstream data derivation analysis. Initially, we applied the imputation method to three cell mixture datasets that offer clear pseudotemporal developmental trajectories from one cell line to another, rendering them ideal for evaluating imputation method performance. Subsequently, utilizing both the original and imputed datasets, we employed the trajectory analysis method monocle2 to establish a cellular pseudotime path, selecting the H2228 cell line as the trajectory's root state, consistent with a previous study [32]. The evaluation metric comprises the maximum overlap ratio of cells between the inferred branches and those within the true trajectory. The improvement in overlap rate reflects the beneficial role of single-cell imputation in enhancing data completeness, reducing noise, and improving both statistical power and algorithm performance, ultimately increasing the accuracy and reliability of cell trajectory inference.

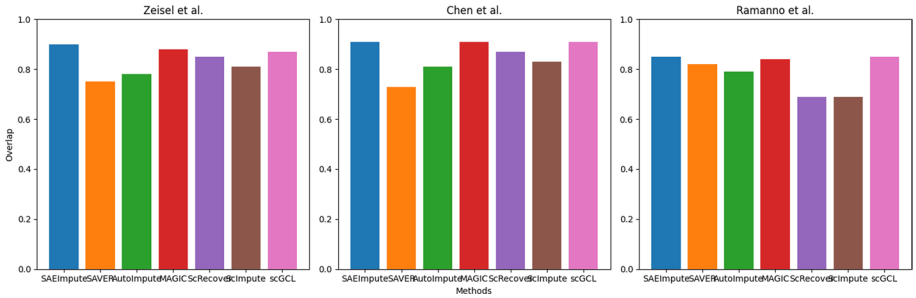

Fig. 6 The overlap of different imputation methods on three cell mixture datasets

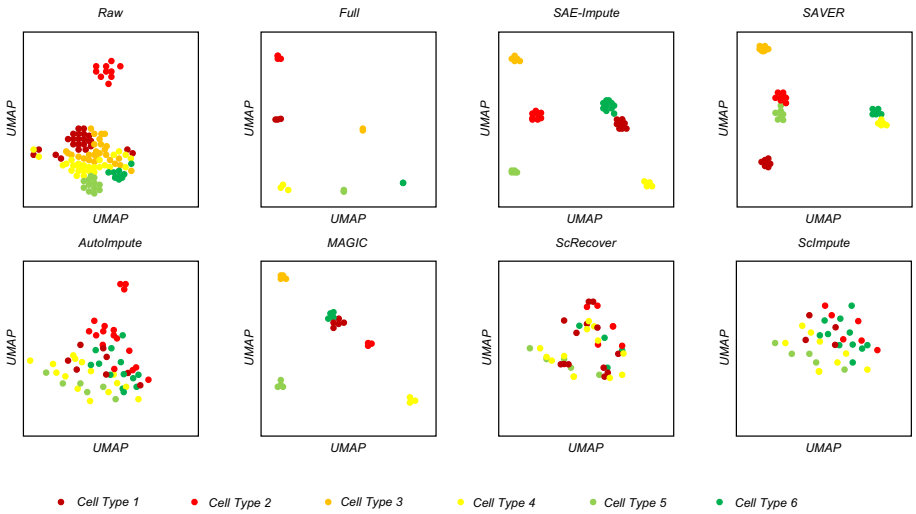

Fig. 7 The results of visualizations through UMAP

As illustrated in Fig. 6, SAE-Impute achieved the highest overlap rate, indicating its enhanced performance in cell trajectory inference on the evaluated dataset.

Figure 7 illustrates the UMAP visualization of simulated single-cell RNA sequencing (scRNA-seq) data, depicting across 6 distinct cell types, before and after imputation, utilizing the human islet dataset from Barron et al [25]. Subpopulation stratification was assessed by comparing the original data with imputed data. "Full" denotes simulated scRNA-seq data without any loss, whereas "raw" indicates data with a 50% loss. The remaining subfigures depict visualizations based on the post-imputation dataset using six imputation methods.

Conclusion

As single-cell RNA-sequencing (scRNA-seq) data often contains missing events that can impede downstream analysis, we propose a novel imputation method called SAE-Impute. This method combines a subspace regression model [33] and an autoencoder to effectively denoise scRNA-seq data and enable data recovery while preserving the heterogeneity of gene expression across cells. One of the key advantages of SAE-Impute is its ability to seamlessly integrate with various downstream analysis tools for scRNA-seq data. We conducted analytical experiments on both simulated and real datasets, and the

results demonstrate that our method not only improves the original data, but also outperforms other imputation methods under certain conditions. This highlights the potential of SAE-Impute to significantly enhance the quality of scRNA-seq data and facilitate downstream analysis.

As the extent of data loss in real single-cell RNA-sequencing data is often unknown, it is common for a large number of true zeros to be present, which can pose a challenge for algorithms to distinguish between missing and true zeros. By examining Table 1 and Fig 6, we can observe that our method performs better with an increasing number of real deletions, which is consistent with the situation in real single-cell sequencing data. This highlights the effectiveness of our method in accurately interpolating missing values and distinguishing between true and dropout zeros.

After conducting a comprehensive analysis of simulated and real datasets, we have drawn the following conclusions. Existing methods such as scImpute and scRecover assume that single-cell data follows specific distribution models, but due to the complexity of single-cell data, relying on a single distribution model may not be sufficient to accurately impute dropout values. SAVER is a UMI-based imputation method that reduces false positives, but its effectiveness decreases as more data is lost. MAGIC relies on finding structures in the data to predict dropout expression levels, which can introduce false positive signals. ALRA uses singular value decomposition, but may not accurately capture the characteristics of real zeros and dropout zeros in the data. AutoImpute is limited by the lack of real imputation labels for single-cell data.

To address these limitations, we propose a novel imputation method called SAE-Impute, which offers three key advantages. First, we use a subspace regression model to preserve the correlation between cells and minimize the introduction of noise, while retaining biological information. Second, the subspace regression model classifies possible dropout values as predicted values, providing targeted interpolation. Third, SAE-Impute uses an autoencoder framework to learn the underlying structure of single-cell sequencing data, making it suitable for high-throughput datasets. Our experimental results on both simulated and real datasets demonstrate that SAE-Impute effectively enhances the recovery of missing expression values and improves the accuracy of downstream analyses.

Methods

The overview of the SAE-Impute algorithm

In recent years, the exponential growth of biological information data has led to an increased adoption of deep learning in the fields of biology and biomedicine. The subspace regression algorithm has demonstrated its effectiveness in preserving correlation relationships between data during the clustering process of single-cell data. Additionally, the autoencoder, acknowledged by numerous scholars as a potent tool for comprehending the intricate structure of data for reconstruction [34], plays a pivotal role in this context. Building upon these insights, we introduce SAE-Impute, an innovative model that amalgamates the subspace regression model and autoencoder. Emphasizing the preservation of correlation between cells, the SAE-Impute algorithm initially employs the subspace regression model to predict potential dropout values. Subsequently, an autoencoder model is constrained by these predicted values to impute dropout events

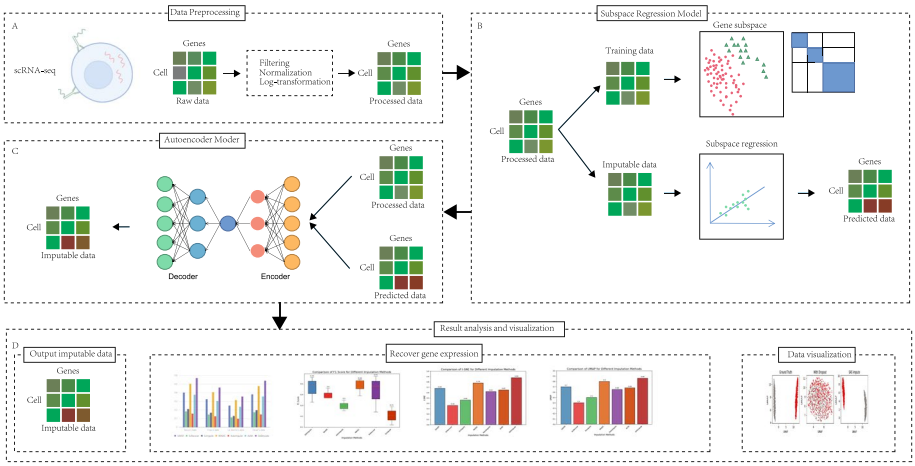

Fig. 8 Overview of SAE-impute. **A** acquiring single-cell sequencing data and its systematic organization, **B** employing subspace regression models for predictive analyses, and **C** identifying dropout values, subsequently addressing them through imputation utilizing autoencoder models. Subsequently, **D** the imputation results are generated, and the experiments are comprehensively compared, evaluated, and analyzed

in scRNA-seq data (Fig. 8). In this study, our methodology involves (A) acquiring single-cell sequencing data and its systematic organization, (B) employing subspace regression models for predictive analyses, and (C) identifying dropout values, subsequently addressing them through imputation utilizing autoencoder models. Subsequently, (D) the imputation results are generated, and the experiments are comprehensively compared, evaluated, and analyzed. A comprehensive description of the proposed method will be provided below.

Data processing

To generate a reference dataset from real scRNA-seq data, we first selected high-quality cells and genes with high expression from the original dataset, treating them as the real expression IR. We then generated downsampling by drawing a Poisson distribution with the mean parameter, resulting in the observation dataset IO. Here are the specific details of the data collection process:

For the human islet data from Barron et al [25]. we filtered out genes with mean expression less than 0.001 and non-zero expression in less than three cells. Genes with non-zero expression in 25% of cells and cells with a library size greater than 5,000 were then selected from the filtered dataset containing 14,729 genes and 1,937 cells. This resulted in 2,284 genes and 1,076 cells. For the mouse hypothalamus data from Chen et al. [26], we filtered out cells with a library size greater than 15,000, as well as genes with mean expression less than 0.0002 and non-zero expression in less than five cells. We then selected genes with non-zero expression in 20% of cells and cells with a library size greater than 2,000, resulting in 2,159 genes and 7,712 cells. For the human ventral midbrain data from Ramanno et al. [27], we filtered out genes with mean expression less than 0.001 and non-zero expression in less than three cells. We then selected genes with non-zero expression in 30% of cells and cells with a library size greater than 5,000, resulting in 2,059 genes and 947 cells. For the Zeisel et al. [28] mouse cortex and hippocampus

data, we selected genes with non-zero expression in 40% of cells and cells with a library size greater than 10,000 UMI, resulting in 3,529 genes and 1,800 cells.

Subspace regression models to make predictions

This section presents the specific calculation method for the subspace regression model as a priori model. Firstly, it is determined whether each zero value is a result of dropouts. Given a zero-valued entry, let $p1$ and $p2$ denote the probability of observing a zero value in the corresponding gene and cell, respectively. Since genes and cells have zero values that are binomially distributed as $X \sim Bin(n, p_1)$ and $Y \sim Bin(m, p_2)$, assuming n is the number of gene measurements and m is the number of cell measurements, in the case of zero values, $p = p1 = p2$. If X and Y are independent, then $X + Y \sim Bin(n + m, p)$ holds true. Therefore, the conditional distribution of X, $P(X = x \mid X + Y = r)$ is a hypergeometric distribution, where x represents the number of zero values observed in a gene, and r is the total number of genes observed and zero values in the selected gene and cell pair. The probability function of the hypergeometric distribution can be expressed as follows:

$$P(X = x - 1 \mid X + Y = r - 1) = \frac{\binom{n-1}{x-1}\binom{m}{r-x}}{\binom{n+m-1}{r-1}} \tag{3}$$

Please note that Equation (1) accounts for overlapping entries in both X and Y for each gene and cell pair. To address this, we adopt the following strategy: (i) We use $(n + m - 1)$ instead of $(n + m)$ as the total number of observations in the selected gene pair, (ii) we use $(n - 1)$ instead of (n) as the number of gene measurements, and (iii) we use $(x - 1)$ instead of (x) as the number of genes with zero values observed in the cells. This is because such genes do not contribute to the hypergeometric probability calculation.

We then calculate a p-value for each zero-value and perform two tests: underrepresentation analysis and overrepresentation analysis, with a significance threshold of 0.01. Entries with significant p-values in the overrepresentation analysis are considered implausible and should be classified. On the other hand, entries with significant p-values in the underrepresentation analysis are considered reliable. Entries that fall in neither category should be disregarded. These values are not extrapolated and should not be used for extrapolation purposes.

Based on this hypothesis testing process, we obtain a set of genes that can be used for training (training data) and a set of genes that need to be attributed (attributable data). A gene is classified as plausible if all its entries are plausible, while a gene is considered attributable if at least one value is attributable.

In order to accurately infer dropout values for genes, it is crucial to utilize related genes with similar expression patterns. This module aims to identify subspaces of genes within the training data that share similar patterns. We will then use a generalized linear regression model on the gene subspace to estimate dropout values in groups. To assign a gene in the imputable set $g \in I_O$ to a subspace, we compute the Euclidean distance from that gene to the centroid of each gene subspace $d(AB) = \sqrt{\sum_{i=1}^{N} \left(x_i - y_i\right)^2}$. Based on

the calculated distances, we assign the gene to the closest subspace (with the smallest Euclidean distance). To estimate dropout values in gene g, we train a generalized linear model using only highly correlated genes within the specified subspace in T. The linear regression process consists of two steps: selecting highly correlated genes from the training set and training a linear model using these genes to estimate dropout values. To ensure that genes with high expression values do not dominate the regression process, we always use a logarithmic transformation (base 2) to rescale the data to an acceptable range ([0,100] by default). To obtain the prediction matrix I_P for matrix I_O.

Autoencoder models

In recent years, the accuracy of collaborative filtering has significantly improved due to the emergence of representation learning. In particular, autoencoder-based models have played a prominent role in this enhancement [35–37]. Unlike matrix factorization or kernel norm minimization techniques, autoencoders require estimating $2 \times m.r$ independent variables. This reduction in the number of parameters is advantageous in data-constrained scenarios like ours, where models are susceptible to overfitting. Fewer parameters decrease the model's propensity for overfitting and enhance its generalization capabilities, resulting in improved overall performance.

As a self-supervised learning approach, autoencoders inherently learn data structures, making them well-suited for the analysis of single-cell sequencing data.

An autoencoder consists of two main components: an encoder E and a decoder D. Initially, the input matrix is transformed into a latent representation (H) where the activation function ϕ is applied, resulting in $H = \phi(EY)$. Subsequently, the decoder (D) maps the latent space (H) back to the input space to yield $X = DH = D\phi(EX)$. During the training phase, the encoder and decoder work collaboratively to minimize the Euclidean cost function $f(x) = \underset{D,E}{\operatorname{argmin}} \parallel X - D\phi(EX) \parallel_F^2$.

In our approach, we leverage the similarities between this problem and collaborative filtering by using the original matrix I_O and the matrix I_P predicted by the subspace regression model as input data Y. Both matrices are mapped to the latent space (H) during training of the encoder and decoder functions. Our ultimate goal is to regenerate the estimated expression matrix I_R by minimizing the cost function for optimal imputation.To enhance the effectiveness of the autoencoder, further details regarding the model architecture and hyperparameter selection are necessary, particularly the choice of the regularization coefficient λ. Additionally, we need to clarify how the output from the subspace regression model is integrated within the autoencoder framework. Ultimately, we aim to regenerate the estimated expression matrix I_R by minimizing the cost function for perfect imputation as fellow:

$$\underset{E,D}{\operatorname{argmin}} \parallel R - D\sigma(E(R))) \parallel_O^2 + \frac{\lambda}{2}\left(\parallel E \parallel_F^2 + \parallel D \parallel_F^2 \right) \tag{4}$$

In this context, R is computed as the Hadamard product of $R = M \circ X$ (where M is a binary mask), E and D is the decoding mask representation, while λ is the regularization coefficient. The loss is computed using the count of non-zero elements in the sparse

expression matrix $M \circ X$, denoted by O. The sigmoid activation function is applied by the encoder layer in the neural network and represented by σ. To avoid overfitting to the non-zero values in the count matrix, we apply regularization to the encoder and decoder matrices during training. After training, the learned matrices are used to estimate the expression matrix, which is represented by Eq. 3. The estimated matrix, denoted as $\tilde{X}$, contains predicted count values for all positions in the matrix.

$$\tilde{X} = D\sigma(E(R)) \tag{5}$$

The input raw gene expression matrix undergoes a series of preprocessing steps including filtering for bad genes, normalization to library size, trimming by gene selection, and log-transformation. The resulting processed matrix and the prediction matrix obtained from the subspace regression model are then inputted into the AutoImpute model.

The SAE-Impute model is comprised of a fully connected multi-layer perceptron consisting of three layers: an input layer, a hidden layer, and an output layer. The model leverages imputation weight labels, which are comprised of data filled with subspace regression model predictors I_P, to improve the filling of missing values. Gradients are computed using backpropagation of errors and the model is trained using gradient descent to minimize the cost function.

Acknowledgements
Not applicable.

Author contributions
All authors were involved in the conceptualization of the SAE-Impute method. SLW conceived and supervised the project. LB and BYJ designed the study and developed the approach. LB and BYJ analyzed the results. LB, BYJ and SLW contributed to the review of the manuscript before submission for publication. All authors read and approved the final manuscript.

Funding
This work was supported by Graduate Research Innovation Project of Hunan Province (QL20230101, CX20230440); NSFC-FDCT Grants 62361166662; National Key R&D Program of China 2023YFC3503400, 2022YFC3400400; Key R&D Program of Hunan Province 2023GK2004, 2023SK2059, 2023SK2060; Top 10 Technical Key Project in Hunan Province 2023GK1010.

Availability of data and materials
All datasets analyzed in this manuscript are publicly available. The ERCC spike-ins data are available at the Gene Expression Omnibus (GEO) under accession code GSE60361. The cell cycle data are available at ArrayExpress under accession code E-MTAB-2512. The mouse embryo data are available at GEO under accession code GSE45719. The fish data from the melanoma cell line can be found at (https://www.dropbox.com/s/ia9x0iom6dwueix/fishSubset.txt?dl=0.) Other public datasets were used in this study: Baron et al. [25] (GSM2230757), Chen et al. [26](GSE87544), La Manno et al. [27] (GSE76381), Zeisel et al. [28] (https://linnarssonlab.org/cortex). All the data and code used in our experiments has been deposited in the GitHub repository: https://github.com/a1035073186/SAE-imputation.git.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no conflict of interest.

Received: 1 February 2024 Accepted: 23 September 2024
Published online: 01 October 2024

References

1. Nawy T. Single-cell sequencing. Nat Methods. 2014;11:18–18.

2. Vallejos CA, Risso D, Scialdone A, Dudoit S, Marioni JC. Normalizing single-cell RNA sequencing data: challenges and opportunities. Nat Methods. 2017;14:565–71.
3. Qiu P. Embracing the dropouts in single-cell RNA-seq analysis. Nat Commun. 2020;11(1):1169.
4. Kim TH, Zhou X, Chen M. Demystifying drop-outs in single-cell UMI data. Genome Biol. 2020;21:196.
5. Huang M, et al. Saver: gene expression recovery for single-cell RNA sequencing. Nat Methods. 2018;15:539–42.
6. Miao Z, Li J, Zhang X. screcover: Discriminating true and false zeros in single-cell RNA-seq data for imputation. BioRxiv 2019;665323.
7. Li WV, Li JJ. An accurate and robust imputation method scimpute for single-cell RNA-seq data. Nat Commun. 2018;9:997.
8. Chen M, Zhou X. Viper: variability-preserving imputation for accurate gene expression recovery in single-cell RNA sequencing studies. Genome Biol. 2018;19:196.
9. 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 Comput Biol. 2021;17:e1009118.
10. Gong W, Kwak I-Y, Pota P, Koyano-Nakagawa N, Garry DJ. Drimpute: imputing dropout events in single cell RNA sequencing data. BMC Bioinform. 2018;19:1–10.
11. Van Dijk D, et al. Recovering gene interactions from single-cell data using data diffusion. Cell. 2018;174:716–29.
12. Xiong Z, et al. Scgcl: an imputation method for scrna-seq data based on graph contrastive learning. Bioinformatics. 2023;39:btad098.
13. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell RNA-seq denoising using a deep count autoencoder. Nat Commun. 2019;10:390.
14. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods. 2018;15:1053–8.
15. Talwar D, Mongia A, Sengupta D, Majumdar A. Autoimpute: Autoencoder based imputation of single-cell RNA-seq data. Sci Rep. 2018;8:16329.
16. Arisdakessian C, Poirion O, Yunits B, Zhu X, Garmire LX. Deepimpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. Genome Biol. 2019;20:1–14.
17. Xu Y, et al. scIGANs: Single-cell RNA-seq imputation using generative adversarial networks. Nucleic Acids Res. 2020;48:e85–e85.
18. Linderman GC, et al. Zero-preserving imputation of single-cell RNA-seq data. Nat Commun. 2022;13:192.
19. Mongia A, Sengupta D, Majumdar A. Mcimpute: matrix completion based imputation for single cell RNA-seq data. Front Genet. 2019;10:9.
20. Hou W, Ji Z, Ji H, Hicks SC. A systematic evaluation of single-cell RNA-sequencing imputation methods. Genome Biol. 2020;21:1–30.
21. Cheng Y, Ma X, Yuan L, Sun Z, Wang P. Evaluating imputation methods for single-cell RNA-seq data. BMC Bioinform. 2023;24:302.
22. Tran D, Tran B, Nguyen H, Nguyen T. A novel method for single-cell data imputation using subspace regression. Sci Rep. 2022;12:2697.
23. Steinley D. Properties of the Hubert-arable adjusted rand index. Psychol Methods. 2004;9:386.
24. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation. BMC Genomics. 2020;21:1–13.
25. Baron M, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. Cell Syst. 2016;3:346–60.
26. Chen R, Wu X, Jiang L, Zhang Y. Single-cell RNA-seq reveals hypothalamic cell diversity. Cell Rep. 2017;18:3227–41.
27. La Manno G, et al. Molecular diversity of midbrain development in mouse, human, and stem cells. Cell. 2016;167:566–80.
28. Zeisel A, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science. 2015;347:1138–42.
29. Ramos-Carreño C, Torrecilla JL. DCOR: Distance correlation and energy statistics in python. SoftwareX. 2023;22:101326.
30. Datta S, Satten GA. Rank-sum tests for clustered data. J Am Stat Assoc. 2005;100:908–15.
31. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nature Biotechnol. 2019;37:547–54.
32. Gentleman RC, et al. Bioconductor: open software development for computational biology and bioinformatics. Genome Biol. 2004;5:1–16.
33. Wichitaksorn N, Kang Y, Zhang F. Random feature selection using random subspace logistic regression. Expert Syst Appl. 2023;217:119535.
34. Zhang G, Liu Y, Jin X. A survey of autoencoder-based recommender systems. Front Comput Sci. 2020;14:430–50.
35. Liang, D., Krishnan, R. G., Hoffman, M. D. & Jebara, T. Variational autoencoders for collaborative filtering. In Proceedings of the 2018 world wide web conference, 689–698 (2018).
36. Zamany S, Li D, Fei H, Li P. Towards deeper understanding of variational auto-encoders for binary collaborative filtering. In: Proceedings of the 2022 ACM SIGIR international conference on theory of information retrieval 2022;254–263.
37. Liu J, Xiao Y, Zhu K, Zheng W, Hsu C-H. Hybrid variational autoencoder for collaborative filtering. In: 2022 IEEE 25th International conference on computer supported cooperative work in design (CSCWD), 2022;251–256 (IEEE).

Publisher's Note