BMC Bioinformatics

RESEARCH Open Access

Scmaskgan: masked multi-scale CNN and attention-enhanced GAN for scRNA-seq dropout imputation

You Wu[1,2], Li Xu[1,2*], Xiaohong Cong[1,2], Hanxiao Li[3] and Yanli Li[1,2]

*Correspondence:
xuli@hrbeu.edu.cn

[1] College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[2] National Engineering Laboratory for Modeling and Emulation in E-Government, Beijing, China
[3] College of Information Technology, University of New South Wales, Sydney, Australia

Abstract

Single-cell RNA sequencing (scRNA-seq) enables high-resolution analysis of cellular heterogeneity, but dropout events, where gene expression is undetected in individual cells, present a significant challenge. We propose scMASKGAN, which transforms matrix imputation into a pixel restoration task to improve the recovery of missing gene expression data. Specifically, we integrate masking, convolutional neural networks (CNNs), attention mechanisms, and residual networks (ResNets) to effectively address dropout events in scRNA-seq data. The masking mechanism ensures the preservation of complete cellular information, while convolution and attention mechanisms are employed to capture both global and local features. Residual networks augment feature representation and effectively mitigate the risk of model overfitting. Additionally, cell-type labels are incorporated as constraints to guide the methods in learning more accurate cellular features. Finally, multiple experiments were conducted to evaluate the methods' performance using seven different data types and scRNA-seq data from ten neuroblastoma samples. The results demonstrate that the data imputed by scMASKGAN not only perform excellently across various evaluation metrics but also significantly enhance the effectiveness of downstream analyses, enabling a more comprehensive exploration of underlying biological information.

Keywords: Single-cell RNA sequencing, Data imputation, Deep learning

Introduction

Single-cell RNA sequencing (scRNA-seq) measures gene expression at the single-cell level, providing valuable insights into cellular heterogeneity and diversity [1, 2]. This technique involves reverse transcription and amplification of full-length mRNA [3], but typically focuses on the 5' and 3' ends, introducing inherent limitations [4]. Variability in reverse transcription efficiency and limited starting RNA contribute to high technical noise, capturing only a fraction of the transcriptome and generating many zero expression values [5]. These zeros may represent either true lack of expression or artificial dropout events caused by technical noise [6], obscuring the true gene expression

landscape. Consequently, advanced methods are needed to accurately identify and mitigate dropout events in scRNA-seq analysis.

Early approaches to address this challenge include methods such as MAGIC [7], SAVER [8], DrImpute [9], VIPER [10], scImpute [11], ENHANCE [12], and SCRABBLE [13]. MAGIC imputes missing values by leveraging similarities between cells, while SAVER uses a Bayesian framework for probabilistic data recovery. DrImpute groups similar cells via clustering for imputation, and VIPER employs a non-negative sparse regression model based on neighborhood information. Additionally, ENHANCE and SCRABBLE utilize principal component analysis and batch normalization techniques, respectively, to reduce dimensionality and mitigate batch effects, thereby enhancing data recovery quality.

Recent years have seen a growing application of deep learning methods to address dropout events in scRNA-seq data [14]. Several approaches-including DeepImpute [15], AutoImpute [16], DCA [17], scVI [18, 19], DISC [20], and sciGANs [21]-have made significant strides in this domain. For example, DeepImpute employs a divide-and-conquer strategy with sub-neural networks to estimate gene expression, outperforming earlier methods such as DrImpute and SAVER. However, its reliance on extensive parameter tuning and using 95% of the data for training can lead to overfitting. AutoImpute learns the inherent data distribution for imputation but has been observed to sometimes produce invalid (negative) values. DCA enhances traditional autoencoders by incorporating negative binomial and zero-inflated noise models [22], thereby improving both denoising and imputation. scVI, a hierarchical Bayesian model, addresses batch correction and imputation by projecting data into latent spaces, although it may struggle when gene counts exceed cell numbers. DISC improves reliability by training on datasets with a high proportion of unexpressed genes (90%), but its efficiency is constrained by the need for high-performance hardware. sciGANs leverages Generative Adversarial Networks (GANs) [23] to frame dropout imputation as a pixel recovery task, thus avoiding overfitting and maintaining robustness in detecting low-expression genes, but converting each row of scRNA-seq data into a square image matrix can introduce significant noise.

Earlier imputation algorithms, such as MAGIC and SAVER, are typically based on assumptions of cell similarity or gene co-expression networks. While these methods can enhance signal clarity, they tend to remove lowly expressed genes and overlook the inherent stochasticity of natural cellular processes, thereby complicating the imputation of rare cell populations. More recent deep learning-based approaches (e.g., scIALM [24], SAE-Impute [25]) often incorporate mathematical assumptions and regularization techniques to constrain the imputation process. Although these methods may improve imputation accuracy or enhance the visual quality of UMAP clustering plots, they frequently produce excessively smoothed results that compromise the representation of true biological variability. In contrast, GAN-based approaches can effectively circumvent these limitations by reframing imputation as a generative process. This strategy allows GANs to learn the true data distribution, generate realistic synthetic data points, and use these generated samples to estimate missing values, thereby mitigating the risk of overfitting while better preserving the intrinsic biological variability.

In this study, we introduce scMASKGAN, a GAN-based framework for scRNA-seq data imputation that combines masking, convolutional neural networks (CNN) [26],

attention mechanisms [27], and ResNets [28] to effectively address dropout events. *Our main contributions are summarized below:*

- We design scMASKGAN with a novel masking mechanism that preserves the intrinsic structure of gene expression data without imposing gene-specific constraints. By integrating *CNNs*, *Self-attention*, and *ResNets*, the model dynamically captures intricate gene-gene and gene-cell interactions, learning hierarchical representations that maintain both local and global biological features.
- Instead of directly modifying the original data, scMASKGAN generates realistic synthetic single-cell data for imputation, thereby avoiding overfitting to dominant cell types while retaining features of rare cells. Additionally, an Isolation Forest algorithm is employed to detect and remove anomalous values during synthesis, ensuring high biological fidelity in the final imputed dataset.
- We validate scMASKGAN on seven diverse datasets and 10 neuroblastoma samples, demonstrating its superior performance over existing imputation methods across multiple quantitative metrics, as well as its ability to restore biologically meaningful gene expression patterns, including improved gene-gene correlations, trajectory inference, batch correction, and differential expression analysis.

Materials and methods

Data preparation

We utilize seven real datasets from Xu et al. [21], encompassing a diverse array of scRNA-seq data types. These datasets include Human brain scRNA-seq data,[1] ERCC spike-in RNAs scRNA-seq data[2], Mouse ESC scRNA-seq data,[3] time-course scRNA-seq data,[4] and three scRNA-seq datasets derived from the sc_Drop-seq,[5] sc_CEL-seq2,[6] and sc_10X platforms.[7] The details is as follows:

- *Human brain scRNA-seq data:* This dataset provides high-quality single-cell RNA sequencing data from human brain tissues. It is used to evaluate the performance of imputation methods in complex biological systems with diverse cell types and intricate gene expression patterns.
- *ERCC spike-in RNAs scRNA-seq data:* This dataset is unique in that it consists of spike-in RNA molecules, where the number of cells exceeds the number of genes. It serves as a benchmark for evaluating the imputation performance on extremely small gene expression matrix datasets.
- *Mouse ESC scRNA-seq data:* This dataset includes embryonic stem cell (ESC) differentiation and large-scale scRNA-seq data. It allows us to test the imputation meth-

[1] https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE67835.
[2] https://www.ebi.ac.uk/biostudies/arrayexpress/studies/E-MTAB-2512.
[3] https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE65525.
[4] https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE75748.
[5] https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE118706.
[6] https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE117617.
[7] https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE111108.

Table 1 Seven scRNA-seq samples

Datasets	Type	Cells	Genes	Accession number	Dropout
Brain	Human	420	22085	GSE67835	81%
ERCC	Spike-in	288	92	E-MTAB-2512	33%
ESC	Mouse	6886	24176	GSE65525	70%
sc_Drop-seq	...	226	15128	GSE118706	62%
sc_CEL-seq2	...	275	28205	GSE117617	74%
sc_10X	...	740	16469	GSE111108	45%
Time-course	...	759	19190	GSE75748	55%

od's effectiveness in developmental biology studies, where gene expression profiles exhibit dynamic changes.

- *Time-course scRNA-seq data:* Temporal sequencing data is crucial for understanding gene expression dynamics over time. This dataset helps evaluate how well the imputation method preserves temporal patterns and recovers missing values in time-dependent scRNA-seq studies.
- *sc_Drop-seq dataset:* Drop-seq is a widely used droplet-based scRNA-seq technology. By including this dataset, we assess the robustness of scMASKGAN across different sequencing platforms, ensuring its compatibility with droplet-based data.
- *sc_CEL-seq2 dataset:* CEL-seq2 is a plate-based sequencing platform known for its improved sensitivity and accuracy in capturing gene expression. This dataset helps evaluate the imputation model's adaptability to high-precision sequencing techniques.
- *sc_10X dataset:* The 10X Genomics platform is one of the most widely used commercial scRNA-seq technologies. Including this dataset ensures that scMASKGAN can generalize to large-scale single-cell datasets generated from commercial platforms, demonstrating its scalability and broad applicability.

By incorporating datasets from different species, sequencing platforms, data sizes, dropout rate and experimental conditions, we ensure that our evaluation of scMASKGAN is comprehensive and reflects its potential utility in diverse real-world applications. Detailed information on these datasets is provided in Table 1.

Additionally, we employ a dataset from Yuan et al. [29] that comprises 10 neuroblastoma samples, including 5 high-risk neuroblastomas (NB) and 5 low-risk neuroblastomas-among which there are 4 ganglioneuroblastomas (GNB) and 1 ganglioneuroma (GN). These data can be downloaded from GEO,[8] as detailed in Table 2. The samples were obtained from clinical surgical resections and belong to the same sequencing batch, which facilitates a comparative assessment of model performance across batches and ensures that the imputation process mitigates batch effects during data integration. *Moreover, as dropout rates exceed 90% in most of these datasets, it facilitates rigorous testing of the model's robustness*

[8] https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE192906.

Table 2 Ten neuroblastoma samples

Datasets	Type	Cells	Genes	Dropout
GSM5768743	NB	960	33514	95%
GSM5768744	NB	768	33514	91%
GSM5768745	NB	445	33514	97%
GSM5768746	NB	357	33514	85%
GSM5768747	NB	639	33514	96%
GSM5768748	GNB	740	33514	97%
GSM5768749	GNB	1052	33514	97%
GSM5768750	GNB	1053	33514	97%
GSM5768751	GNB	360	33514	84%
GSM5768752	GN	551	33514	97%

under extreme conditions. In subsequent figures, we will primarily use the dropout rates of different datasets as legends or titles.

Methods

Inspired by the remarkable performance of GANs in image restoration and the introduction of StackGAN [30], a model designed for text-to-image generation, we propose scMASKGAN, a generative adversarial network tailored for scRNA-seq imputation. Both image inpainting and missing value imputation share the fundamental goal of restoring incomplete data while preserving its structural and contextual integrity. In image inpainting, GANs learn spatial structures and content features to generate realistic pixels that seamlessly blend with surrounding regions. Similarly, in missing value imputation, GANs leverage gene expression patterns and intercellular relationships to generate biologically meaningful values that accurately reflect the underlying data distribution. By converting each cell's normalized gene expression profile into a grayscale image, scMASKGAN extracts essential features and utilizes GANs' robust image synthesis capabilities to generate realistic grayscale representations that impute missing gene data. The detailed framework is shown in Fig. 1, and in this section, we provide a comprehensive description of the scMASKGAN workflow.

Data preprocessing

scMASKGAN reframes the imputation of single-cell RNA-seq data as an image inpainting (pixel restoration) problem by converting the gene expression matrix into an image-based representation. As shown in Fig. 1A, each cell's expression profile (originally a vector of m gene features) is mapped onto a 2D grid of pixels of size $N \times N$ (chosen such that $N \times N \geq m$,). Specifically, the gene expression vector of each cell is zero-padded to a length of $N \times N$, reshaped into a two-dimensional array by sequentially arranging the elements row-wise, thus ensuring consistency with the original matrix. In this image, each pixel corresponds to a specific gene, and its intensity represents the expression level of that gene in the given cell. For a gene j mapped to pixel coordinate (u, v), we have:

$$I_i(u, v) = x_{i,j} \tag{1}$$

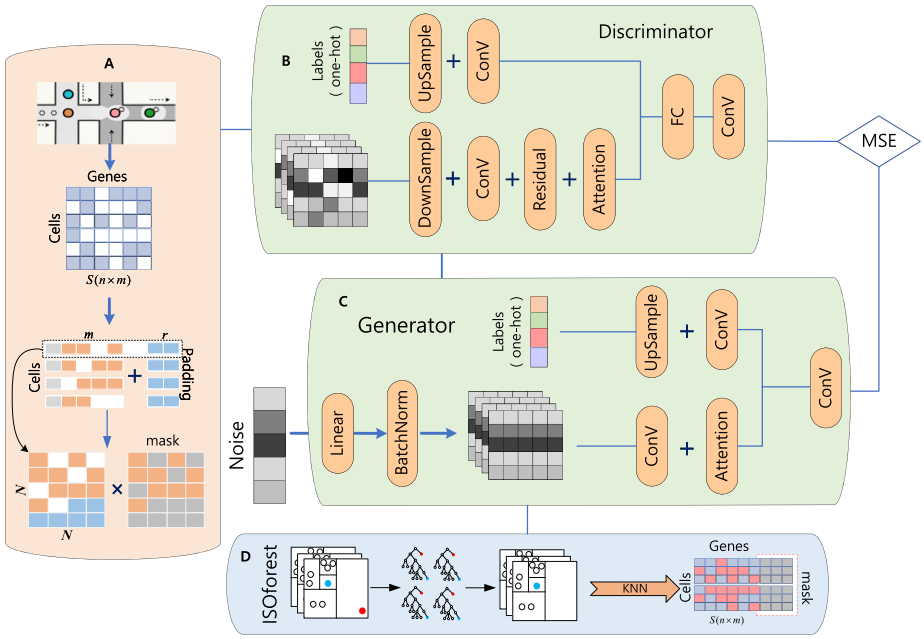


Fig. 1 The overall framework of scMASKGAN is illustrated across four panels. Panel A shows the process of preparing data, beginning with the sequencing-derived gene expression matrix and converting it into a format suitable for model input. Panel B details the architecture of discriminator, designed to distinguish between original and imputed data, thereby driving the generator to produce more biologically accurate imputed values, the upsampling module is designed to increase data dimensionality and capture detailed distribution features. The downsampling module reduces resolution and utilizes convolution to extract higher-order features. Residual blocks are employed to enhance these high-level features, while the attention module is used to capture global characteristics. Panel C presents the architecture of generator, which uses multiple layers and mechanisms to infer and fill missing values based on the observed data patterns. Panel D outlines the complete imputation workflow

where $x_{i,j}$ denotes the expression value of gene j in cell i. Dropout events (genes with undetected expression in a cell) manifest as pixels with zero intensity in the image, and scMASKGAN applies a masking mechanism to flag these missing values. A binary mask of the same $N \times N$ size is generated for each cell, using 0 for dropouts and 1 for observed gene expressions. This masking mechanism serves two critical functions. First, since gene expression values are initially normalized, missing entries would otherwise be assigned nonzero values, potentially introducing noise when extracting features using CNNs and attention mechanisms. The binary mask effectively mitigates this interference by distinguishing observed values from imputed ones. Second, by masking the padded values, the model ensures that they do not participate in the computation process, preserving the integrity of the learned representations. This image-based representation allows missing data imputation to be treated as a pixel-wise image restoration task, enabling convolutional neural networks with attention mechanisms to capture both local gene-gene interaction patterns and global expression structures from the spatial layout, leading to more effective recovery of missing data.

Moreover, cell-type labels ($\mathbf{y}$) is extracted from the dataset and converted into categorical numerical values to represent different cell types. In scMASKGAN training, the label is transformed into a one-hot encoded vector and incorporated into both the generator and discriminator, ensuring class-specific data generation and enhancing

classification performance. In the imputation process, label is utilized to associate each cell with the corresponding generated candidate set, thereby ensuring that missing values are imputed using biologically relevant data from the same category.

Generator

In scMASKGAN, the generator G_{θ_G} transforms a random noise vector $\mathbf{z}$ into a synthetic gene expression tensor, conditioned on a one-hot encoded label matrix $\mathbf{y}$ through a series of linear and non-linear operations, as illustrated in Fig. 1B. Initially, the noise vector $\mathbf{z} \in \mathbb{R}^{d_z}$ is sampled from a prior distribution p_z, ($\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$). The generator's architecture is defined by the following sequence of transformations:

$$G_{\theta_G} : \mathbf{z}, \mathbf{y} \xrightarrow{F_1} \mathbf{H}^{(1)} \xrightarrow{F_2} \mathbf{H}^{(2)} \xrightarrow{F_3,..,F_L} \mathbf{H}^{(L)} = G(\mathbf{z}, \mathbf{y}) \tag{2}$$

where each F_l denotes a parameterized transformation (either linear or convolutional), followed by a ReLU or Sigmoid activation function to ensure the non-negativity of scRNA-seq data and the binary representation of grayscale images. The final output $\mathbf{H}^{(L)}$ constitutes the generated tensor, which represents imputed gene expression data. The objective of the generator is to produce synthetic data that conform to the biological context specified by the label matrix $\mathbf{y}$. The generator is optimized by minimizing the following loss function:

$$\min_G \ \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\left[\log\left(1 - D\big(G(\mathbf{z} \mid \mathbf{y}) \mid \mathbf{y}\big)\right)\right] \tag{3}$$

where D represents the discriminator and $G(\mathbf{z} \mid \mathbf{y})$ is the generator output conditioned on $\mathbf{y}$. This loss encourages the generator to produce outputs that are indistinguishable from real data under the given cell-type conditions. Initially, the generator applies a linear transformation, batch normalization, and ReLU activation to the input noise vector $\mathbf{z}$, yielding an intermediate feature map $\mathbf{H}^{(1)}$. This feature map is subsequently reshaped into a tensor of dimensions $\left(B, \text{cn1}, \frac{\text{img\_size}}{4}, \frac{\text{img\_size}}{4}\right)$, where B is the batch size and cn1 denotes the number of channels in the first convolutional block. Next, a self-attention mechanism is employed to capture long-range dependencies among features. In this mechanism, attention scores are computed by learning queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$, and the attention output $\mathbf{H}^{(\text{att})}$ is calculated as follows:

$$\text{Attention} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right), \quad \mathbf{H}^{(\text{att})} = \mathbf{V} \cdot \text{Attention}. \tag{4}$$

This output is integrated with the preceding feature map via a residual connection, modulated by a learnable scaling parameter γ. Concurrently, the label matrix $\mathbf{y}$ is upsampled to match the spatial dimensions of the feature map $\mathbf{H}^{(3)}$ and processed through a convolutional layer to reduce its channel dimensionality. The resulting label-aligned feature map is then concatenated with $\mathbf{H}^{(3)}$ to form the final feature map $\mathbf{H}^{(5)}$, which is further processed by additional convolutional layers. The output image $G(\mathbf{z}, \mathbf{y})$ is obtained by applying a sigmoid activation to the final convolutional layer's output. The detailed design of the generator is as follows:

Algorithm 1 scMASKGAN Generator

Require: Noise vector $\mathbf{z} \in \mathbb{R}^{d_z}$ $(\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}))$
Require: One-hot label matrix $\mathbf{y}$
Ensure: Generated gene expression tensor $G(\mathbf{z}, \mathbf{y})$
$\mathbf{h}^{(1)} \leftarrow \mathrm{ReLU}\Big(\mathrm{BatchNorm}(\mathrm{Linear}(\mathbf{z}))\Big)$
Reshape $\mathbf{h}^{(1)}$ to tensor with shape $(B, \mathrm{cn1}, \mathrm{img\_size}/4, \mathrm{img\_size}/4)$
$\mathbf{H}^{(2)} \leftarrow \mathrm{ReLU}\Big(\mathrm{BatchNorm}(\mathrm{Conv2d}(\mathbf{h}^{(1)}))\Big)$
$\mathbf{Q} \leftarrow \mathrm{Conv2d}(\mathbf{H}^{(2)})$
$\mathbf{K} \leftarrow \mathrm{Conv2d}(\mathbf{H}^{(2)})$
$\mathbf{V} \leftarrow \mathrm{Conv2d}(\mathbf{H}^{(2)})$
$\mathrm{Attention} \leftarrow \mathrm{Softmax}\Big(\dfrac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d}}\Big)$
$\mathbf{H}^{(att)} \leftarrow \mathbf{V} \times \mathrm{Attention}$
$\mathbf{H}^{(3)} \leftarrow \mathbf{H}^{(2)} + \gamma \cdot \mathbf{H}^{(att)}$
Upsample $\mathbf{y}$ to match the spatial dimensions of $\mathbf{H}^{(3)}$, denoted as $\mathrm{Upsample}(\mathbf{y})$
$\mathbf{H}^{(4)} \leftarrow \mathrm{Conv2d}(\mathrm{Upsample}(\mathbf{y}))$
$\mathbf{H}^{(5)} \leftarrow \mathrm{Concat}(\mathbf{H}^{(3)}, \mathbf{H}^{(4)})$ ▷ Concatenate along the channel dimension
$G(\mathbf{z}, \mathbf{y}) \leftarrow \mathrm{Sigmoid}(\mathrm{Conv2d}(\mathbf{H}^{(5)}))$
return $G(\mathbf{z}, \mathbf{y})$

Discriminator

The primary function of the discriminator D is to distinguish between real and synthetic gene expression data. As illustrated in Fig. 1C, the discriminator receives as input an image x representing gene expression data and a label matrix y encoding cell-type information. Its optimization objective is defined as

$$\max_{D} \ \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x \mid y)] + \mathbb{E}_{z \sim p_z(z)}\Big[\log\Big(1 - D\big(G(z \mid y) \mid y\big)\Big)\Big] \tag{5}$$

where $D(x \mid y)$ denotes the probability that the discriminator assigns to a real sample x under the condition y, and $D(G(z \mid y) \mid y)$ corresponds to the probability assigned to a generated sample.

Specifically, the input image x is first processed by a preprocessing layer that flattens the image and reconstructs it into a feature map suitable for convolutional operations. This feature map is then passed through a downsampling convolutional block to reduce its spatial dimensions, with residual connections incorporated to enhance feature propagation and facilitate model learning. Moreover, a self-attention mechanism is integrated to improve feature extraction by computing attention scores and generating the corresponding output $H^{(\text{att})}$. Simultaneously, the label matrix y is upsampled and processed analogously to the procedure employed in the generator, after which it is concatenated with the image features to form a label-aligned feature map. Finally, a series of upsampling layers is applied to restore the spatial dimensions of the feature map, ultimately yielding a probability map that reflects the authenticity of the input image x under the condition y. The detailed design of the discriminator is as follows:

Algorithm 2 scMASKGAN Discriminator

Require: Input image $\mathbf{x}$ representing gene expression data
Require: One-hot label matrix $\mathbf{y}$
Ensure: Probability map $\mathbf{y}_{\text{final}}$ indicating real/fake classification

$\mathbf{x}_{\text{pre}} \leftarrow \text{Linear}\Big(\text{flatten}(\mathbf{x})\Big)$
$\mathbf{x}_{\text{pre\_reshaped}} \leftarrow \text{Reshape}(\mathbf{x}_{\text{pre}})$
$\mathbf{H}^{(1)} \leftarrow \text{ReLU}\Big(\text{BatchNorm2d}(\text{Conv2d}(\mathbf{x}_{\text{pre\_reshaped}}))\Big)$
Compute self-attention output $\mathbf{H}^{(att)}$ from $\mathbf{H}^{(1)}$
$\mathbf{H}^{(2)} \leftarrow \mathbf{H}^{(1)} + \gamma \cdot \mathbf{H}^{(att)}$
Upsample $\mathbf{y}$ to match the spatial dimensions of $\mathbf{H}^{(2)}$, denoted as $\text{Upsample}(\mathbf{y})$
$\mathbf{y}_{\text{feat}} \leftarrow \text{Conv2d}(\text{Upsample}(\mathbf{y}))$
$\mathbf{H}_{\text{concat}} \leftarrow \text{Concat}(\mathbf{H}^{(2)}, \mathbf{y}_{\text{feat}})$ ▷ Concatenate along the channel dimension
$\mathbf{H}^{(3)} \leftarrow \text{FullyConnected}(\mathbf{H}_{\text{concat}})$
$\mathbf{y}_{\text{final}} \leftarrow \text{Sigmoid}\Big(\text{Conv2d}(\mathbf{H}^{(3)})\Big)$
return $\mathbf{y}_{\text{final}}$

Overall, the generator and discriminator collaborate within an adversarial framework. The generator produces realistic gene expression data conditioned on the label matrix $\mathbf{y}$. The discriminator distinguishes between real and generated data based on the same cell-type label conditioning. This architecture enables the generation of realistic gene expression profiles. The detailed design of the scMASKGAN is as follows:

Algorithm 3 scMASKGAN Training Process

Require: Training dataset $\{(x_i, y_i)\}_{i=1}^{N}$, noise distribution p_z, hyperparameters:

$$\text{num\_epochs},\ \text{batch\_size},\ \eta_G,\ \eta_D,\ \lambda, \ldots$$

Initialize Generator G_{θ_G} and Discriminator D_{θ_D} with random weights.
Initialize optimizers $\mathcal{O}_G$ for G_{θ_G} and $\mathcal{O}_D$ for D_{θ_D}.
Load dataset and construct DataLoader with mini-batch size = `batch_size`.
for *epoch* = 1 to num_epochs **do**
  for each mini-batch (X_{real}, Y) from DataLoader **do**
    // **Preprocess real data**
    $X_{masked} \leftarrow \text{ApplyMask}(X_{real})$
    // **Sample noise vector**
    Sample $z \sim p_z$.
    // **Generate synthetic data**
    $\hat{X} \leftarrow G_{\theta_G}(z, Y)$
    // **Discriminator forward pass**
    $D_{real} \leftarrow D_{\theta_D}(X_{real}, Y)$
    $D_{fake} \leftarrow D_{\theta_D}(\hat{X}, Y)$
    // **Compute Discriminator Loss**

$$\mathcal{L}_D = -\frac{1}{m}\sum_{i=1}^{m}\left[\log D_{real}^{(i)} + \log\left(1 - D_{fake}^{(i)}\right)\right]$$

    // **Update Discriminator**
    $\theta_D \leftarrow \theta_D - \eta_D \nabla_{\theta_D}\mathcal{L}_D$
    // **Compute Generator Loss**

$$\mathcal{L}_G = \frac{1}{m}\sum_{i=1}^{m}\left[\log\left(1 - D_{fake}^{(i)}\right) + \lambda\,\text{ReconstructionLoss}\left(\hat{X}^{(i)}, X_{real}^{(i)}\right)\right]$$

    // **Update Generator**
    $\theta_G \leftarrow \theta_G - \eta_G \nabla_{\theta_G}\mathcal{L}_G$
  end for
  if Convergence criteria met (e.g., loss plateau) **then**
    break ▷ Early stopping
  end if
  Optionally **save** model checkpoints.
end for

Imputed method

For a given cell c_i in subgroup K_{c_i}, we first generate a candidate set $A_{K_{c_i}}$ of n_{can} expression profiles using a well-trained scMASKGAN. Specifically, the one-hot encoded cell-type vectors are combined with randomly sampled noise and fed into the pre-trained generator. Through multiple transformations and mapping processes, the generator produces a candidate set of gene expression profiles that correspond to the given cell type. We then apply the Isolation Forest (ISOforest) algorithm[31] (Fig. 1D) to detect and remove outliers by constructing isolation trees and assigning anomaly scores based on path lengths-where shorter paths indicate potential outliers. Specifically, each tree recursively partitions the data by randomly selecting features and split points. The depth of an isolation tree, which represents the number of splits required to isolate a data point, reflects the difficulty of isolating that point. To quantify the degree of anomaly,

an anomaly score $s(x)$ is computed for each candidate gene expression profile using the formula $s(x) = 2^{-\frac{E(h(x))}{c(n)}}$. where $E(h(x))$ denotes the average path length of data point x across all isolation trees, and $c(n)$ represents the average path length for a dataset of size n. Since anomalous points are generally easier to isolate, they tend to have shorter path lengths and, consequently, higher anomaly scores. Finally, a threshold τ is set to identify outliers, where a data point is classified as an anomaly if $s(x) > \tau$ The threshold τ can be dynamically determined based on the distribution of anomaly scores within the dataset. Moreover, we find scMASKGAN employs the Isolation Forest anomaly detection algorithm to identify and remove outliers during the imputation of batch data in Sect. 4.3. By eliminating anomalies introduced during batch sequencing, this approach reduces the impact of batch effects.

Finally, using Euclidean distance ($d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$), where $\mathbf{x}$ and $\mathbf{y}$ represent the gene expression vectors of two cells, with a dimension of n (i.e., the number of genes). To ensure computational accuracy and comparability, all gene expression data undergo normalization preprocessing before computing the Euclidean distance, thereby eliminating the influence of varying gene expression ranges. Furthermore, to enhance the reliability of nearest neighbor selection, we define a dynamically adjusted similarity threshold θ, which is determined based on the statistical distribution of Euclidean distances between all cell pairs. Typically, it is set as the mean μ plus a certain multiple of the standard deviation σ, i.e.,$\theta = \mu + k\sigma$. To optimize the imputation performance, we further employ Optuna [32] for hyperparameter tuning, primarily optimizing the k -nearest neighbors (KNN) parameter k. The objective is to search for the optimal k value that minimizes the mean squared error (MSE) between the original and imputed data, ensuring that selected neighbors exhibit high similarity while avoiding the influence of outliers. Ultimately, only cells with a Euclidean distance less than θ are retained as candidate neighbors, improving the reliability and biological interpretability of the imputed data. To estimate the expression value of gene j in cell c_i, we use the following formula:

$$\hat{c}_{i,j} = \begin{cases} c_{i,j}, & \text{if } c_{i,j} > 0, \\ c_{i,kNN,j}, & \text{otherwise.} \end{cases} \tag{6}$$

Here, $\hat{c}_{i,j}$ denotes the estimated expression value, $c_{i,j}$ is the original expression value from the raw profile, and $c_{i,kNN,j}$ represents the value estimated from the expression profiles of the nearest neighbor cells. If the expression value of gene j in cell c_i is greater than zero, the original value is retained. Otherwise, the corresponding gene expression value from the nearest neighbors is used for imputation.

Finally, we obtained a matrix consisting of the imputed scRNA-seq data along with padded zeros. Since we applied a masking mechanism that ignored these zeros, they were not processed during model training and imputation. Therefore, by removing the padded zeros, we obtained the final imputed matrix.

Parameter adjustment strategies

The parameter-tuning strategy involves carefully adjusting the learning rate (lr), typically starting from values such as 0.001 or 0.0001, and then employing Adam [33] with weighted iterative refinements based on observed convergence performance. The parameter img_size, defined as dimension N, directly corresponds to gene count,

a higher gene count necessitates larger dimensions. The latent dimension latent_dim is consistently set equal to img_size to preserve spatial coherence. Cell-type labels ($\mathbf{y}$) and the n_{cls} are adjusted simultaneously to ensure biologically relevant category assignments and effective imputation. In addition, we also present the parameter settings and convolutional layer configurations of the generator and discriminator.

Generator parameters and convolutional architecture:

(1) Perform linear transformation and reshape the latent noise vector z through a fully connected layer, generating an initial tensor z_n of dimension $(32, N, N)$.
(2) Conduct convolution on z_n using convolutional layer $GConv1$ with kernel size 3×3, obtaining tensor z_{conv} of dimension $(32, N, N)$.
(3) Apply the self-attention mechanism on z_{conv} to derive an attention-refined tensor.
(4) Perform convolution on one-hot encoded cell-type label tensors using convolutional layer $LConv1$ with kernel size 3×3, producing tensor L_n of dimension $(8, N, N)$.
(5) Concatenate the attention-refined z_{conv} and L_n into tensor $GConcat$ with dimension $(40, N, N)$.
(6) Execute sequential convolutional operations $GConv2_1$ and $GConv2_2$, each with kernel size 3×3, on $GConcat$, yielding the generator output tensor of dimensions $(1, N, N)$.

Discriminator parameters and convolutional architecture:

(1) Perform initial linear transformation and reshape the input tensor into dimensions $(1, N, N)$.
(2) Apply convolutional layer $DConv1$ with kernel size 3×3, followed by max-pooling and a residual block, reducing the tensor dimension to $(32, N/2, N/2)$.
(3) Conduct convolutional layer $DConv2$ with kernel size 3×3, further reducing tensor dimension to $(16, N/2, N/2)$.
(4) Integrate the self-attention mechanism on this reduced tensor to capture inter-feature dependencies.
(5) Perform convolution on one-hot encoded cell-type label tensors using convolutional layer $LConv2$ with kernel size 3×3, generating tensor L_d of dimension $(8, N/2, N/2)$.
(6) Concatenate attention-refined tensor and L_d forming tensor $DConcat$ with dimension $(24, N/2, N/2)$.
(7) Apply fully connected layers and additional convolutional layers with kernel size 3×3 to produce the discriminator output.

The size of the convolutional kernel and the dimensional settings are adjusted based on the number of genes. The dimension typically tuned between 32 and 64, which is sufficient to capture the characteristics of various types of data. The residual parameter γ serves as a balancing factor within residual blocks, moderating the interplay between real and generated features to stabilize model training. Typically, γ is adjusted experimentally, higher values emphasize generated features, whereas lower

values prioritize real features. This parameter is fine-tuned iteratively based on convergence behavior.

Metrics evaluation

The primary objective of data imputation is to construct a "gold standard" dataset by rectifying false zeros introduced by technical noise, thereby accurately recovering the true expression levels of genes. In this process, genes that are genuinely unexpressed remain as true zeros. To mitigate the impact of zero values on computational analyses, minimal non-zero values are assigned to lowly expressed genes, while the expression levels of highly expressed genes remain largely unaffected.

To rigorously assess the performance of scMASKGAN, we conducted comprehensive comparisons with 12 state-of-the-art imputation algorithms-namely, AutoImpute, DCA, DeepImpute, DrImpute, ENHANCE, MAGIC, SAVER, scImpute, SCRABBLE, VIPER, scIGANS, and scGAIN-across seven real scRNA-seq datasets. Evaluation metrics included the following:

- *Uniform Manifold Approximation and Projection (UMAP) distribution plots*: Used to visualize the global structure of the data before and after imputation, ensuring that the overall clustering patterns remain biologically meaningful.
- *Z-score standardized distribution*: Examined to verify that imputed data retains a normalized distribution, facilitating comparability with the original data and minimizing artificial distortions.
- *Coefficient of Variation (CV)*: Assessed to maintain gene expression variability, ensuring that important biological signals are not lost during imputation.
- *Wasserstein distance and Jensen-Shannon distance*: Employed to quantify distributional differences between the imputed and original datasets, providing robust statistical metrics for evaluating imputation performance.
- *Accuracy (ACC)*: Evaluated to measure the ability of imputation methods to correctly classify cell types based on reconstructed expression profiles.
- *Area Under the Receiver Operating Characteristic Curve (AUC)*: Used to assess the sensitivity and specificity of gene expression restoration, providing a robust measure of imputation reliability.
- *F1 scores*: Computed to balance precision and recall, particularly in identifying dropout events versus true biological zeros.
- *Pearson correlation coefficients*: Calculated relative to the original datasets to quantify the fidelity of imputed data, ensuring that gene-gene relationships remain intact.

These metrics were rigorously analyzed to provide a comprehensive assessment of the efficacy of scMASKGAN in accurately imputing gene expression data.

UMAP distribution

UMAP is widely used for dimensionality reduction in scRNA-seq, enabling the visualization of high-dimensional data in two dimensions and facilitating data distribution analysis [34]. As illustrated in Figure 2, UMAP projections of imputed Humanbrain datasets reveal that methods such as MAGIC, AutoImpute, DeepImpute, VIPER, scGAIN, and

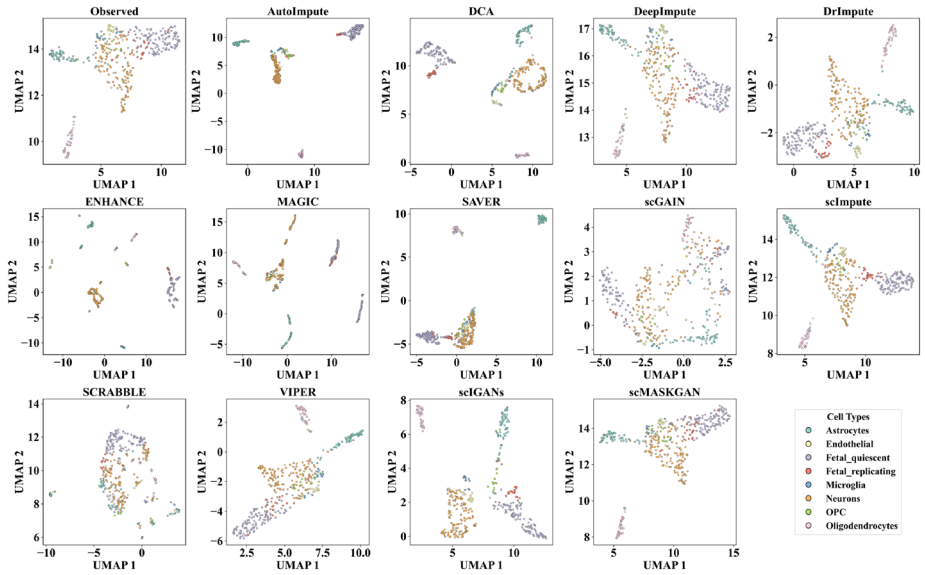


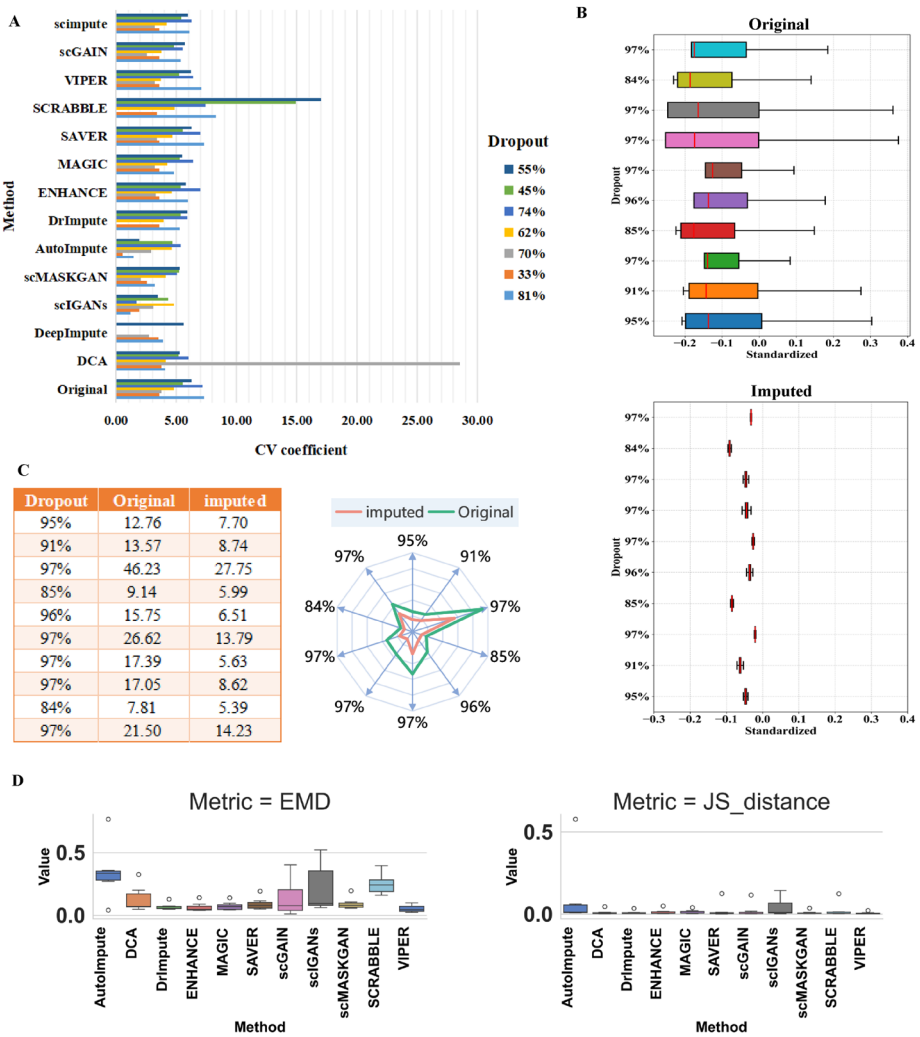
Fig. 2 The UMAP projections of the original scRNA-seq data alongside those of data imputed by 13 different methods, accompanied by cell-type labels

scMASKGAN effectively recover data structure. Notably, AutoImpute exhibits superior cell type separation, clearly distinguishing neurons, astrocytes, and other cell types with clustering patterns that closely resemble the original data. scMASKGAN performs similarly well, maintaining strong alignment with the inherent data structure and suggesting minimal alteration of the underlying biological distribution. While DeepImpute and VIPER also achieve reasonable cell type separation, minor overlaps between clusters are observed. In contrast, DCA, scrImpute, and scGAIN yield substantial cell type overlap and weaker clustering, reducing the interpretability of the imputed data. Overall, AutoImpute and scMASKGAN emerge as the most optimal imputation methods, effectively preserving cell type structure and distribution for robust downstream analysis.

Furthermore, we have presented the distributions of other datasets in Supplementary Figures 1 to 8. Supplementary Figures 1 and 2 display the UMAP distributions of the original neuroblastoma data across 10 groups and the corresponding imputed data generated by scMASKGAN, demonstrating that even under extremely high dropout rates, scMASKGAN maintains excellent imputation performance. Additionally, Supplementary Figures 3 to 8 illustrate the T-SNE and UMAP distributions of the datasets in Table 1, along with the imputed data distributions from scMASKGAN, clearly indicating that favorable imputation results are consistently achieved under various dropout rates.

Coefficient of variation

The Coefficient of Variation (CV) [35] serves as a standardized metric for quantifying data dispersion, higher CV values denote increased variability, whereas lower CV values indicate greater consistency. As illustrated in Fig. 3A, the CV values across datasets with varying dropout rates are presented. Notably, the SCRABBLE method exhibits comparatively elevated CV values, indicative of pronounced variability. In contrast, the DCA method demonstrates exceptionally high variability in the Mouse



Dropout	Original	imputed
95%	12.76	7.70
91%	13.57	8.74
97%	46.23	27.75
85%	9.14	5.99
96%	15.75	6.51
97%	26.62	13.79
97%	17.39	5.63
97%	17.05	8.62
84%	7.81	5.39
97%	21.50	14.23

Fig. 3 Comparison of imputation metrics. Panel A shows the CV coefficients of different imputation methods across various dropout rates. Panel B presents the Z-score standardized distribution comparison between scMASKGAN imputed data and the original data under extreme missingness. Panel C illustrates the CV coefficients comparison between scMASKGAN imputed data and the original data under extreme missingness conditions. Panel D compares multiple imputation methods using the JS test and Wasserstein distance

ESC dataset (70% dropout), which we attribute to potential inaccuracies in parameter estimation due to the large data volume. Conversely, the AutoImpute, scMASKGAN, and scGANs methods yield relatively lower CV values, reflecting superior imputation performance.

Furthermore, to rigorously assess the robustness of scMASKGAN under extreme missing data conditions, Fig. 3C displays the CV values for scMASKGAN's imputed results across 10 neuroblastoma dataset groups. In conjunction with the UMAP distribution analyses, these findings suggest that scMASKGAN effectively integrates low variability with high fidelity to the original data distribution, thereby representing an exemplary imputation approach that is critical for the accuracy of downstream analyses.

Z-score distribution

Z-score [36] distribution refers to the distribution obtained by applying Z-score normalization to the data. Specifically, for each data point x, the Z-score is defined as

$$z = \frac{x - \mu}{\sigma} \tag{7}$$

where μ is the mean of the data and σ is the standard deviation. This normalization procedure transforms the data such that the resulting distribution has a mean of 0 and a standard deviation of 1. We utilize the Z-score distribution to compare the original data with the imputed data and to detect significant deviations. A more concentrated Z-score distribution indicates fewer outliers and superior data quality. As shown in Supplementary Figure 9, the Z-score distributions for datasets with varying dropout rates demonstrate that scMASKGAN performs outstandingly. Moreover, Fig. 3B presents the Z-score distributions of both the original and the scMASKGAN-imputed data under extremely high dropout rates. When combined with the UMAP distribution and CV coefficient analyses from the preceding sections, these results indicate that scMASKGAN maintains excellent performance even under extreme conditions.

Statistical tests

In Fig. 3D, we compare the JS distance [37] and the Wasserstein distance (EMD) [38] between the imputed data generated by various methods and the original data. The JS distance is computed derived from the Jensen-Shannon divergence, which is defined as

$$\mathrm{JSD}(\mathrm{P} \parallel \mathrm{Q}) = \frac{1}{2}\mathrm{D}_{\mathrm{KL}}\left(\mathrm{P} \middle\| \frac{\mathrm{P}+\mathrm{Q}}{2}\right) + \frac{1}{2}\mathrm{D}_{\mathrm{KL}}\left(\mathrm{Q} \middle\| \frac{\mathrm{P}+\mathrm{Q}}{2}\right) \tag{8}$$

where D_{KL} denotes the Kullback–Leibler divergence. The JS distance is then obtained as the square root of the Jensen-Shannon divergence:

$$\mathrm{JSdistance}(\mathrm{P}, \mathrm{Q}) = \sqrt{\mathrm{JSD}(\mathrm{P} \parallel \mathrm{Q})} \tag{9}$$

The Wasserstein distance (also known as the Earth Mover's Distance, EMD) is defined as

$$W_1(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \int_{\mathcal{X}\times\mathcal{X}} \|x - y\| \, d\gamma(x, y), \tag{10}$$

where $\Gamma(P, Q)$ denotes the set of joint distributions with marginals P and Q, and $\|x - y\|$ represents the distance between x and y in the space $\mathcal{X}$.

Our results indicate that scMASKGAN exhibits outstanding performance with respect to both metrics, as evidenced by its low JS distance and low Wasserstein distance, which suggest a high degree of consistency between the imputed data and the original data. In contrast, the AutoImpute, scGAIN, sciGANs, and SCRABBLE methods show only moderate performance in terms of the Wasserstein distance. Furthermore, sciGANs and AutoImpute perform poorly in terms of the JS distance. A closer examination of the imputed data reveals that sciGANs tends to impute all missing values, which does not conform to the requirement of removing dropout values and consequently introduces a large number of spurious biological signals. Additionally,

in order to achieve higher metric scores, AutoImpute introduces negative values into the gene expression matrix, a practice that is biologically implausible.

This observation highlights the limitations in adaptability and stability of both scImpute and Deepimpute across diverse datasets. In particular, while scImpute can effectively impute data from certain platforms, its output for other datasets results in missing values when calculating the JS distance, EMD statistics, and Z-score distributions, rendering these metrics uncomputable. Meanwhile, DeepImpute produced usable results solely in the Human Brain dataset. These findings suggest that some methods may require further optimization or integration with other approaches to enhance estimation performance and ensure the reliability of subsequent metric calculations.

Cluster metrics

ACC, AUC, and F1 score are standard classification evaluation metrics used in clustering, with values ranging from 0 to 1, where higher values indicate better clustering performance[39]. To further assess the effectiveness of various imputation algorithms, we employed the Louvain algorithm to cluster seven datasets and compared the resulting clustering labels with the corresponding cell types. The performance in terms of ACC, AUC, and F1 scores is summarized in Tables 3 and 4. Table 3 details different types of scRNA-seq data, with "..." indicating that certain algorithms were not applicable to specific datasets, while Table 4 records scRNA-seq data from different platforms. Figure 4 visualizes these results.

scMASKGAN outperforms other methods in terms of ACC, followed by DCA, DeepImpute, and SAVER, which also show strong performance from the analysis. In terms of the F1 score, scMASKGAN, DeepImpute, and SAVER exhibit the highest median values. For AUC, scMASKGAN, DeepImpute, and SAVER again show leading performance, with scMASKGAN maintaining a high median and demonstrating

Table 3 Clustering metrics for 13 imputation algorithms across various datasets

Datasets	Human brain			ERCC spike-in			Mouse ESC			Time-course		
Metric	**ACC**	**AUC**	**F1**	**ACC**	**AUC**	**F1**	**ACC**	**AUC**	**F1**	**ACC**	**AUC**	**F1**
AutoImpute	0.876	0.745	0.634	0.608	0.578	0.453	0.518	0.575	0.518	0.743	0.669	0.507
DCA	0.781	0.669	0.472	0.542	0.538	0.434	0.502	0.666	0.506	0.500	0.667	0.504
Deepimpute	0.803	0.678	0.492	0.559	0.557	0.454	0.640	0.730	0.721	0.846	0.865	0.874
DrImpute	0.834	0.732	0.580	0.604	0.61	0.513	...	...	...	...	...	...
ENHANCE	0.862	0.751	0.626	**0.746**	0.718	0.624	...	...	...	...	...	...
MAGIC	0.812	0.689	0.512	0.563	0.542	0.423	0.515	0.670	0.515	0.500	0.667	0.494
SAVER	0.856	0.804	0.671	0.525	0.526	0.425	...	...	...	...	...	...
scGAIN	0.626	0.562	0.332	0.359	0.501	0.491	0.749	0.772	0.808	0.739	0.792	0.789
scImpute	0.857	0.767	0.638	0.711	0.734	0.649	**0.904**	**0.910**	**0.966**	**0.977**	**0.977**	**0.998**
SCRABBLE	0.481	0.575	0.367	0.540	0.522	0.404	...	...	...	...	...	...
VIPER	0.768	0.662	0.460	0.511	0.507	0.404	...	...	...	...	...	...
scIGANs	0.802	0.678	0.492	0.546	0.525	0.405	0.812	0.785	0.940	0.542	0.684	0.549
scMASKGAN	**0.975**	**0.965**	**0.964**	0.54	**0.954**	**0.667**	0.614	0.704	0.684	0.946	0.948	0.949

The bold indicates the best results of the test metrics across different datasets

Table 4 Clustering metrics for 13 imputation algorithms across platform datasets

Datasets	sc_10X			sc_CELseq2			sc_Dropseq		
Metric	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
AutoImpute	0.979	0.977	0.969	0.577	0.565	0.461	0.821	0.800	0.737
DCA	0.997	0.977	0.997	**0.995**	**0.994**	0.993	0.935	0.926	0.905
Deepimpute	0.997	0.997	0.996	0.990	0.988	0.985	0.994	0.993	0.991
DrImpute	0.996	0.995	0.993	0.980	0.977	0.970	0.909	0.897	0.865
ENHANCE	0.791	0.800	0.726	0.958	0.976	**0.996**	0.994	0.993	0.991
MAGIC	0.953	0.948	0.930	0.546	0.507	0.366	0.546	0.535	0.429
SAVER	0.996	0.995	0.993	0.990	0.988	0.985	0.994	**0.993**	**0.991**
scGAIN	0.880	0.872	0.825	0.500	0.508	0.424	0.481	0.500	0.425
scImpute	0.658	0.668	0.503	0.940	0.883	0.787	0.930	0.910	0.890
SCRABBLE	0.990	0.988	0.985	0.501	0.500	0.406	0.519	0.511	0.409
VIPER	0.999	**0.998**	**0.999**	0.564	0.517	0.365	0.923	0.920	0.890
scIGANs	0.658	0.668	0.503	0.430	0.035	0.601	0.412	0.667	0.500
scMASKGAN	**0.999**	0.981	0.966	0.969	0.503	0.668	**0.997**	0.836	0.803

The bold indicates the best results of the test metrics across different datasets

the most stable distribution, as indicated by the boxplot. These results suggest that scMASKGAN not only excels across all metrics but also demonstrates superior stability and generalizability, making it well-suited for diverse scRNA-seq data types, sizes, and platforms.

Correlation analysis

In addition to comparing clustering performance, we evaluated each imputation method by analyzing the Pearson correlation between the imputed and original data [40]. Figure 5 presents boxplots of the Pearson correlation coefficients, where scMASKGAN, SAVER, scImpute, and SCRABBLE show higher median values, indicating a strong alignment with the original data and, thus, more reliable imputation. In contrast, MAGIC, ENHANCE, and VIPER display lower correlation values, implying that their imputed data deviate more from the observed values-likely due to the introduction of bias during imputation.

In conclusion, integrating both clustering performance and correlation analyses, scMASKGAN and SAVER emerge as the most optimal imputation methods, excelling in preserving data consistency and stability. Conversely, ENHANCE and MAGIC show limitations in both aspects, making them less suitable for tasks that require high-fidelity imputation.

Cost analysis

To rigorously assess the performance of scMASKGAN, we conducted benchmarking experiments using a computational platform comprising two NVIDIA 4090 GPUs (24GB each) and one NVIDIA A4000 GPU (16GB). We evaluated both the memory consumption and the runtime efficiency of 13 imputation methods, including scMASKGAN, on gene expression datasets spanning 1,000 to 100,000 cells. As illustrated in Fig. 6, scMASKGAN demonstrates competitive performance, achieves second place in memory consumption-surpassed only by scIGAN-while its execution speed is slightly

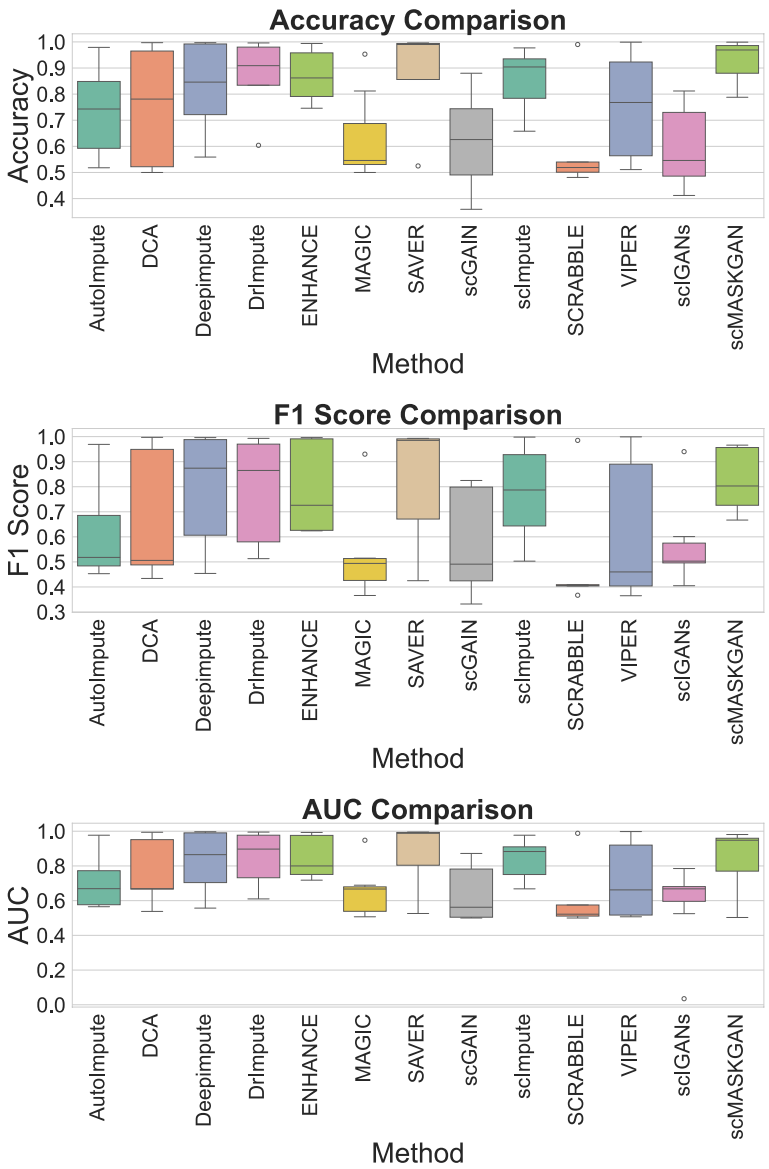


Fig. 4 The clustering metrics for imputation algorithms compared to cell-type labels. The figure presents boxplots comparing ACC, AUC, and F1 scores of 13 imputation methods across different datasets, with each color representing a different method

lower than that of the MAGIC and DeepImpute methods. Overall, these findings underscore the robust computational performance of scMASKGAN, affirming its efficacy in terms of both memory usage and runtime efficiency for large-scale single-cell transcriptomic analyses.

Downstream analysis

We demonstrated the biological relevance of scMASKGAN for data recovery through comprehensive validation using several biological datasets. Specifically, we utilized Mouse ESC scRNA-seq data, temporal data, and 10 neuroblastoma sample datasets.

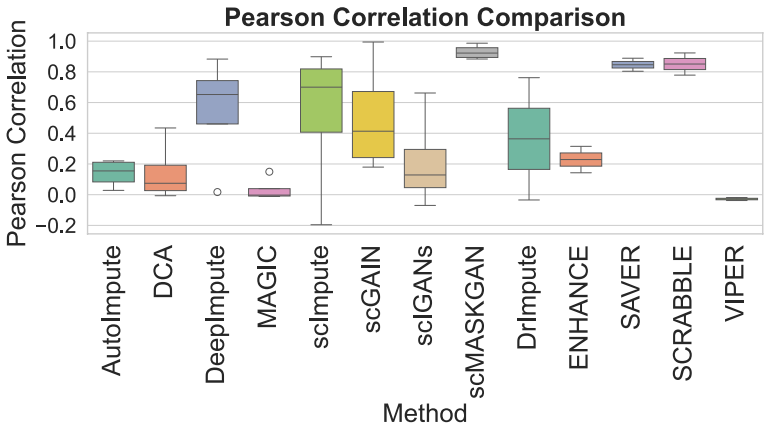


Fig. 5 Boxplot comparison of Pearson correlation coefficients among 13 imputation methods

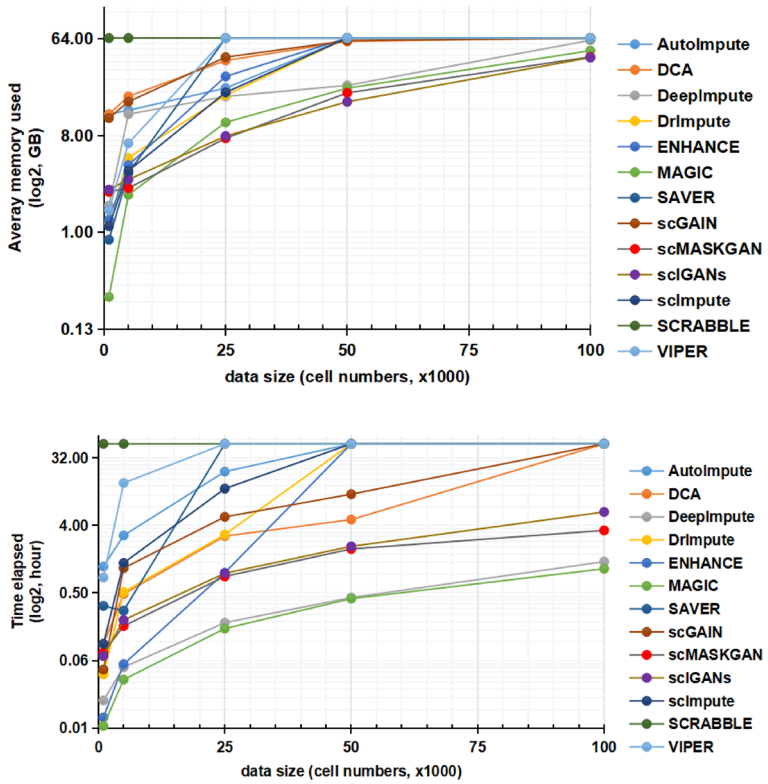


Fig. 6 Comparison of the memory consumption and the runtime efficiency among 13 imputation methods

We conducted a series of analyses, including gene-gene correlation analysis, temporal analysis, gene enrichment pathway analysis, batch data imputation, and differential gene expression analysis, to assess the performance of scMASKGAN across different types of biological data. These analyses not only help clarify the role of scMASKGAN in biological data recovery but also provide strong evidence for its potential use in a variety of biological research applications.

Gene-gene correlation

Gene-gene correlation reveals similarities in expression patterns, aiding in the identification of regulatory networks and gene functions [41]. We analyzed this using a Mouse ESC scRNA-seq dataset, previously applied in scIGAN studies, which comprises 44 cell cycle genes across over 6,800 cells. Figure 7A shows confusion matrices based on Pearson correlation coefficients for both original and imputed data. Notably, significant correlations-such as those among Mcm6, Nasp, Mcm2, and Pcna, as well as between Cdk1, Cenpl, and Top2a-are preserved post-imputation. Additionally, new correlations between cdc20 and Cenpa/PLK1, and between Msh2 and Mcm2/Mcm6, align with known co-expression relationships [42, 43]. Hierarchical clustering further confirms that the imputed data retains the original data's structural characteristics. Overall, these results validate that scMASKGAN effectively restores biologically meaningful gene correlations while preserving the inherent data structure.

Temporal analysis

Temporal analysis is employed in scRNA-seq to infer cellular trajectories during dynamic processes [44]. In this study, we imputed a time-course scRNA-seq dataset-capturing the differentiation of H1 embryonic stem cells (ESCs) into definitive endodermal cells (DEC)-using scMASKGAN, and then reconstructed trajectories with the *Monocle3* R package [45]. As illustrated in Fig. 7B, UMAP plots of the original data reveal

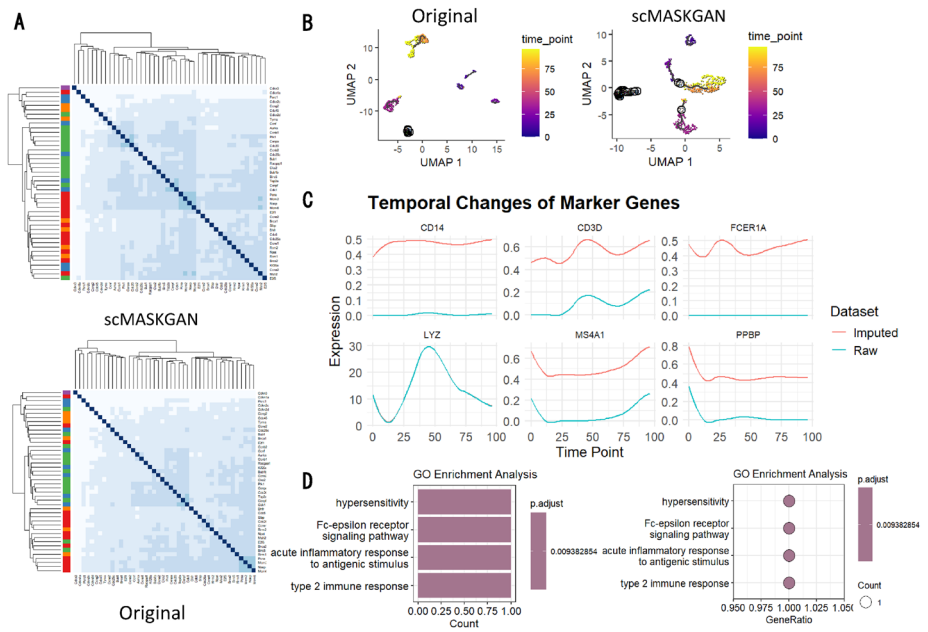


Fig. 7 Gene-gene correlation analysis for *mouse ESC* data, trajectory analysis of *Time-course* scRNA-seq data, and gene pathway enrichment analysis. Panel A shows a heatmap of gene-gene correlations for scMASKGAN-imputed versus original data, with the left color bar representing hierarchical clustering results and the middle section showing a confusion matrix of 44 gene groups where darker blue indicates stronger correlations. Panel B displays trajectory analysis plots for original and imputed data. Panel C shows temporal profiles for six marker genes, with the red line indicating imputed data and the blue line showing original data. Panel D features bar and bubble charts of GO enrichment analysis for *FCER1A*, used to assess the reasons for its temporal upregulation

distinct, dispersed clusters due to technical noise, while the scMASKGAN-imputed data show smoother transitions and enhanced connectivity between cells. Further temporal analysis of marker genes (Fig. 7C) indicates that genes such as CD14, CD3D, MS4A1, and PPBP exhibit increased expression along clearer trajectories, whereas FCER1A, previously affected by dropout events, demonstrates a gradual expression increase over time. GO enrichment analysis (Fig. 7D) confirms that these genes are primarily associated with immune functions, and suggesting that the progressive activation of immune-related pathways underlies the differentiation process from ESC to DEC.

Batch data imputation

We use scRNA-seq data from the same batch to impute and integrate 10 homologous neuroblastoma samples from clinical cases. All data undergo a complete *Scanpy* [46] quality control pipeline: ensuring that each gene is expressed in at least 3 cells, each cell contains at least 200 genes, and cells with mitochondrial gene expression exceeding 5% are removed. Subsequently, the data are log-transformed and subjected to PCA for dimensionality reduction. These preprocessing steps enable us to assess whether the original gene expression patterns are maintained and to evaluate the restoration of key marker genes.

As shown in Fig. 8A, we present UMAP plots of the original data alongside datasets imputed by scMASKGAN, DCA, and DeepImpute. The results indicate that scMASKGAN enhances cell connectivity and effectively reduces the technical noise present in the original data. In contrast, although the DCA-imputed data exhibit a more dispersed

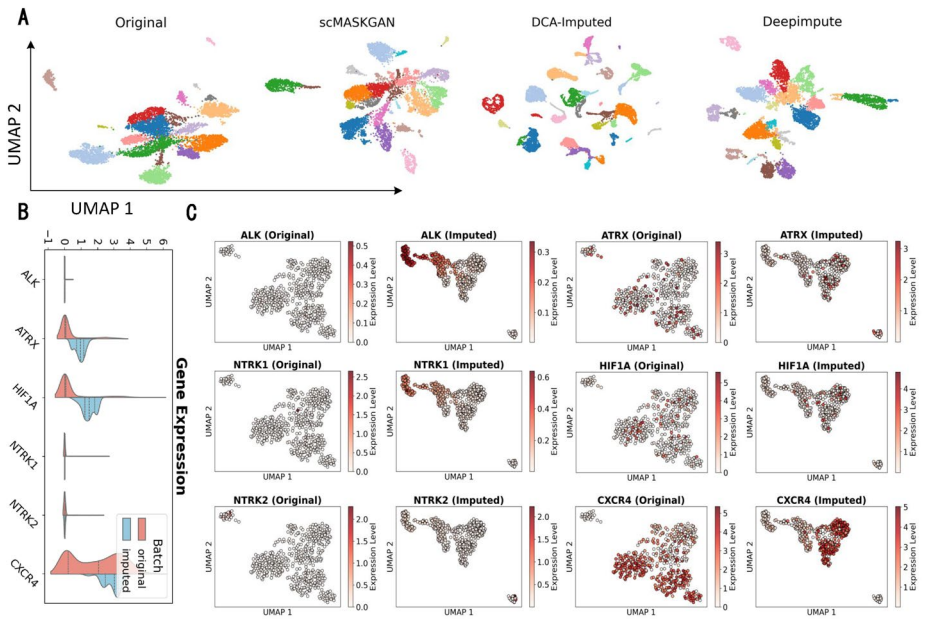


Fig. 8 The results of batch data imputation and differential gene expression analysis are illustrated in three panels. Panel A displays UMAP plots comparing batch-integrated data from three imputation algorithms-scMASKGAN, DCA, and Deepimpute-to the original dataset. Panel B shows the expression levels of five marker genes and the interference gene (NTRK2) in both the original and scMASKGAN-imputed data, with red indicating original data and blue indicating imputed data. Panel C presents scatter plots of the corresponding gene expressions, with the color bar on the right representing gene expression levels

clustering that still delineates cell population differences, the excessive fragmentation may compromise biological relevance. Thus, following batch effect correction and integration, scMASKGAN better reflects the underlying cellular states and connectivity, offering superior imputation performance.

Additionally, we selected one low-risk neuroblastoma (*GSM5768752*) sample to compare the expression of five marker genes and a control gene (NTRK2, which is highly expressed in high-risk neuroblastoma) [47] between the scMASKGAN-imputed and original datasets. As shown in supplementary Fig. 8B, scMASKGAN-imputed data exhibit minimal changes in high-expression genes while notably restoring low-expression genes. Furthermore, supplementary Fig. 8C demonstrates that, except for NTRK2, the expression levels of genes with initially low expression increase, whereas high-expression genes remain largely unchanged. These observations confirm that scMASKGAN achieves high accuracy in data recovery without indiscriminately altering gene expression.

Gene expression analysis

To further validate the effectiveness of the scMASKGAN imputation, we compared the expression profiles of commonly highly variable genes across different cell types between the imputed and original data under various dropout rates, as illustrated in Fig. 9. Specifically, we present the comparison for the Human brain dataset (81.4% dropout) and observe that the data structure remains intact, with no significant alterations in the expression of highly variable genes. In addition, we analyzed datasets with other dropout rates in Supplementary Figure 10, where scMASKGAN consistently preserved the complete biological signal. These results further demonstrate the accuracy and reliability of the scMASKGAN imputation.

Discussion

Robustness across diverse datasets and platforms

To comprehensively evaluate the strengths and weaknesses of scMASKGAN, we compared it with 12 existing imputation methods across various scRNA-seq datasets. In terms of the CV, scMASKGAN's performance is surpassed only by AutoImpute and scIGANs (Fig. 3A). For EMD and JS distance metrics (Fig. 3D), while most methods exhibit robust performance, scImpute and DeepImpute were excluded from some comparisons due to their inability to impute certain datasets, the suboptimal performance of sciGAN and AutoImpute relative to other methods has been discussed in Sect. 3.4. Moreover, clustering assessments (as detailed in Tables 3 and 4, and Fig. 4) indicate that scMASKGAN, alongside SAVER, yields the most effective segregation of cellular heterogeneity. Pearson correlation analysis further confirms that scMASKGAN's imputed data maintains the highest correlation with the original data, and the Z-score standardized distribution (Supplementary Figure 9) demonstrates that its imputation results exhibit minimal outliers and remarkable stability across varying dropout rates. Collectively, these findings underscore the superior efficacy of scMASKGAN as an imputation method.

The results also indicate that scMASKGAN performed exceptionally well on the ERCC spike-in dataset, Human brain dataset, and Time-course dataset, demonstrating

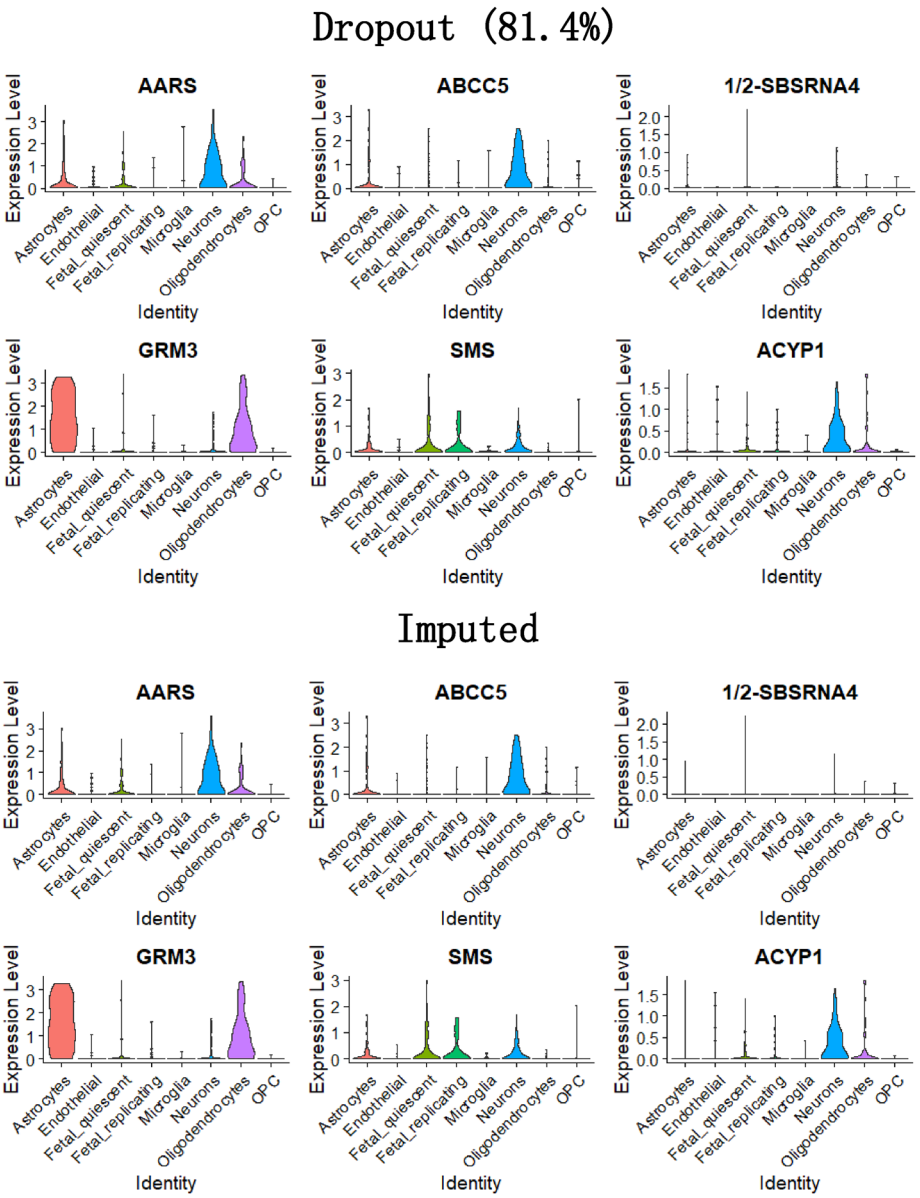


Fig. 9 Comparison of common highly variable genes between imputed data and original data across different cell types

its advantages in handling highly sparse, small-scale, and temporal sequencing data. This superior performance may be attributed to the adversarial training mechanism, which enables scMASKGAN to effectively learn the latent data distribution and generate imputed values that closely resemble the true expression patterns. For the Human brain dataset, scMASKGAN successfully recovered gene expression patterns while preserving cellular heterogeneity. In the Time-course dataset, it maintained the dynamic characteristics inherent in temporal sequencing data, thereby avoiding the loss of time-related information often encountered in traditional imputation methods.

On the large-scale Mouse ESC dataset, scMASKGAN demonstrated moderate performance, without consistently outperforming all comparative methods. As evidenced

by Supplementary Figure 4, although scMASKGAN effectively delineated three distinct clusters, the corresponding clustering metrics remained only moderate. This may be attributable to alterations in the UMAP embedding distribution following cluster separation, potentially introducing inaccuracies in the clustering indices. Nevertheless, with respect to other evaluation criteria, scMASKGAN consistently produced biologically meaningful imputed values and preserved the overall structural integrity of the dataset.

For the sc_10X and sc_Drop-seq datasets, scMASKGAN exhibited robust performance, underscoring its strong generalization capabilities in high-throughput sequencing scenarios. These datasets, characterized by substantial technical noise and sparsity, benefit from scMASKGAN's integrated attention mechanism and adversarial training framework, which collectively enable the effective discrimination of biological signals from technical artifacts, thereby yielding highly accurate imputed values. Moreover, in the neuroblastoma dataset with exceptionally high dropout rates, scMASKGAN demonstrated outstanding imputation performance, further confirming its capacity to enhance data quality under extreme conditions.

The performance on the sc_CEL-seq2 dataset was suboptimal, likely due to its unique characteristics and high levels of sequencing noise. Our imputation results did not fully recover the expression profiles of several key genes, indicating that scMASKGAN could benefit from further refinement to better accommodate various sequencing platforms. Moreover, the pronounced technical noise in sc_CEL-seq2 appears to destabilize the adversarial training process of the GAN-based model, resulting in incomplete gene expression recovery.

Enhanced interpretability and downstream analysis

The downstream analyses presented in our study provide comprehensive evidence of the efficacy and robustness of the scMASKGAN imputation method. The gene-gene correlation analysis demonstrates that scMASKGAN accurately restores biologically meaningful relationships between key cell cycle genes, such as the significant correlations observed among *Mcm6, Nasp, Mcm2,* and *Pcna*, as well as between *Cdk1, Cenpl,* and *Top2a*. Moreover, the emergence of novel correlations that align with known co-expression patterns further substantiates the ability of scMASKGAN to recover latent gene-gene interactions while preserving the intrinsic data structure.

The temporal analysis reinforces these findings by illustrating scMASKGAN's capacity to capture dynamic cellular trajectories. As evidenced by the UMAP plots in Fig. 7B, the imputed data exhibit smoother transitions and enhanced connectivity compared to the original noisy data, thereby facilitating the reconstruction of continuous cellular differentiation pathways. Additionally, the temporal expression profiles of marker genes reveal that dropout-affected genes, such as *FCER1A*, regain a gradual expression increase over time, which is critical for deciphering time-related biological processes. The corresponding GO enrichment analysis further confirms that these genes are predominantly involved in immune functions, underscoring the biological relevance of the imputation.

In the context of batch data imputation (Sect. 4.3 and Fig. 8), scMASKGAN not only attenuates technical noise but also effectively integrates data from multiple samples. Compared to other methods, UMAP visualizations (Fig. 8A) reveal that most algorithms lead to significant fragmentation in gene expression profiles as a result

of imputation. Furthermore, differential expression analyses of key marker genes (Fig. 8B and C) reveal that scMASKGAN not only preserves the expression profiles of highly expressed genes and accurately recovers missing signals, but also avoids introducing erroneous biological artifacts, thereby maintaining the overall integrity of the biological signal.

Finally, the gene expression analysis across datasets with varying dropout rates (Fig. 9 and Supplementary Figure 10) further corroborates the robustness of scMASKGAN. The imputed data retain the expression profiles of highly variable genes, suggesting that the method effectively preserves critical biological signals without introducing significant distortions.

Conclusion

In summary, these results collectively demonstrate that scMASKGAN is a robust and versatile imputation method. It effectively recovers both global and local gene expression patterns across diverse scRNA-seq datasets, ranging from small-scale and highly sparse data to large heterogeneous datasets and time-course studies. While certain platform-specific limitations remain, the overall performance of scMASKGAN underscores its potential as a powerful tool for improving data quality in downstream single-cell analyses. Future research will focus on further optimizing the adversarial training strategy and tailoring the model to the specific characteristics of different sequencing platforms.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06138-9.

Supplementary material 1
Supplementary material 2
Supplementary material 3
Supplementary material 4
Supplementary material 5
Supplementary material 6
Supplementary material 7
Supplementary material 8
Supplementary material 9
Supplementary material 10
Supplementary material 11
Supplementary material 12

Acknowledgements
Not applicable.

Author contributions
You Wu completed the algorithm design, model building and data analysis experiments, Li Xu mainly completed the article writing, Xiaohong Cong completed the drawing of Figures 1 to 3, Hanxiao Li completed the drawing of Figures 4 to 6, and Yanli Li completed the revision of the article.

Funding
This research was funded in part by the National Natural Science Foundation of China under grant No. 62172122, STI 2030—Major Projects 2021ZD0200406, and the Key Research and Development Program of Heilongjiang Province under grant No. 2022ZX01A19.

Availibility of data and materials
Data for this paper is available for download from: https://doi.org/10.5281/zenodo.15067391.

Code availability
Source code for this paper is available for download from the public GitHub repository: https://github.com/A-uo/scMASKGAN.

Declarations

Ethics approval and consent to participate
Not applicable

Competing interests
Not applicable.

Received: 1 March 2025 Accepted: 8 April 2025
Published online: 20 May 2025

References

1. Chang X, Zheng Y, Xu K. Single-cell rna sequencing: technological progress and biomedical application in cancer research. Mol Biotechnol. 2024;66(7):1497–519.
2. Raza K. Introduction to single-cell rna-seq data analysis. In: Machine Learning in Single-Cell RNA-seq Data Analysis, pp. 1–16. Springer, ??? (2024)
3. Jovic D, Liang X, Zeng H, Lin L, Xu F, Luo Y. Single-cell rna sequencing technologies and applications: a brief overview. Clin Transl Med. 2022;12(3):694.
4. Sahin E, Edis G, Keskin E, Akata I. Molecular characterization of the complete genome of a novel ormycovirus infecting the ectomycorrhizal fungus hortiboletus rubellus. Adv Virol. 2024;169(5):110.
5. Pardo-Palacios FJ, Wang D, Reese F, Diekhans M, Carbonell-Sala S, Williams B, Loveland JE, De María M, Adams MS, Balderrama-Gutierrez G, et al. Systematic assessment of long-read rna-seq methods for transcript identification and quantification. Nature methods, 1–15 (2024)
6. Su Y, Yu Z, Yang Y, Wong K-C, Li X. Distribution-agnostic deep learning enables accurate single-cell data recovery and transcriptional regulation interpretation. Adv Sci. 2024;11(16):2307280.
7. Van Dijk D, Sharma R, Nainys J, Yim K, Kathail P, Carr AJ, Burdziak C, Moon KR, Chaffer CL, Pattabiraman D, et al. Recovering gene interactions from single-cell data using data diffusion. Cell. 2018;174(3):716–29.
8. Huang M, Wang J, Torre E, Dueck H, Shaffer S, Bonasio R, Murray JI, Raj A, Li M, Zhang NR. Saver: gene expression recovery for single-cell rna sequencing. Nat Methods. 2018;15(7):539–42.
9. 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.
10. Chen M, Zhou X. Viper: variability-preserving imputation for accurate gene expression recovery in single-cell rna sequencing studies. Genome Biol. 2018;19(1):196.
11. Li WV, Li JJ. An accurate and robust imputation method scimpute for single-cell rna-seq data. Nat Commun. 2018;9(1):997.
12. Wagner F, Barkley D, Yanai I. Accurate denoising of single-cell rna-seq data using unbiased principal component analysis. BioRxiv, 655365 (2019)
13. Peng T, Zhu Q, Yin P, Tan K. Scrabble: single-cell rna-seq imputation constrained by bulk rna-seq data. Genome Biol. 2019;20:1–12.
14. Bao S, Li K, Yan C, Zhang Z, Qu J, Zhou M. Deep learning-based advances and applications for single-cell rna-sequencing data analysis. Brief Bioinform. 2022;23(1):473.
15. 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.
16. Talwar D, Mongia A, Sengupta D, Majumdar A. Autoimpute: autoencoder based imputation of single-cell rna-seq data. Sci Rep. 2018;8(1):16329.
17. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell rna-seq denoising using a deep count autoencoder. Nat Commun. 2019;10(1):390.
18. Gayoso A, Lopez R, Xing G, Boyeau P, Valiollah Pour Amiri V, Hong J, Wu K, Jayasuriya M, Mehlman E, Langevin M, et al. A python library for probabilistic analysis of single-cell omics data. Nat Biotechnol. 2022;40(2):163–6.
19. Virshup I, Bredikhin D, Heumos L, Palla G, Sturm G, Gayoso A, Kats I, Koutrouli M, Berger B, et al. The scverse project provides a computational ecosystem for single-cell omics data analysis. Nat Biotechnol. 2023;41(5):604–6.
20. He Y, Yuan H, Wu C, Xie Z. Disc: a highly scalable and accurate inference of gene expression and structure for single-cell transcriptomes using semi-supervised deep learning. Genome Biol. 2020;21:1–28.
21. Xu Y, Zhang Z, You L, Liu J, Fan Z, Zhou X. scigans: single-cell rna-seq imputation using generative adversarial networks. Nucleic Acids Res. 2020;48(15):85–85.
22. Mi X, Bekerman W, Rustgi AK, Sims PA, Canoll PD, Hu J. Rzimm-scrna: a regularized zero-inflated mixture model framework for single-cell rna-seq data. Ann Appl Stat. 2024;18(1):1–22.
23. Ghosheh GO, Li J, Zhu T. A survey of generative adversarial networks for synthesizing structured electronic health records. ACM Comput Surv. 2024;56(6):1–34.
24. Liu X, Wang H, Gao J. scialm: a method for sparse scrna-seq expression matrix imputation using the inexact augmented Lagrange multiplier with low error. Comput Struct Biotechnol J. 2024;23:549–58.
25. Bai L, Ji B, Wang S. Sae-impute: imputation for single-cell data via subspace regression and auto-encoders. BMC Bioinform. 2024;25(1):317.

26. Zhao X, Wang L, Zhang Y, Han X, Deveci M, Parmar M. A review of convolutional neural networks in computer vision. Artif Intell Rev. 2024;57(4):99.
27. Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning. Neurocomputing. 2021;452:48–62.
28. Sarwinda D, Paradisa RH, Bustamam A, Anggia P. Deep learning in image classification using residual network (resnet) variants for detection of colorectal cancer. Procedia Comput Sci. 2021;179:423–31.
29. Yuan X, Seneviratne JA, Du S, Chen Y, Jin Q, Jin X, Balachandran A, Huang S, Xu Y, Xu Y, et al. Single-cell profiling of peripheral neuroblastic tumors identifies an aggressive transitional state that bridges an adrenergic-mesenchymal trajectory. Cell Rep. 2022;41(1): 111455.
30. Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, Metaxas DN. Stackgan++: realistic image synthesis with stacked generative adversarial networks. IEEE Trans Pattern Anal Mach Intell. 2018;41(8):1947–62.
31. Liu FT, Ting KM, Zhou Z-H. Isolation forest. In: 2008 18th IEEE International Conference on Data Mining, pp. 413–422 (2008). IEEE
32. 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)
33. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
34. Becht E, McInnes L, Healy J, Dutertre C-A, Kwok IW, Ng LG, Ginhoux F, Newell EW. Dimensionality reduction for visualizing single-cell data using umap. Nat Biotechnol. 2019;37(1):38–44.
35. Abdi H. Coefficient of variation. Encyclopedia Res Des. 2010;1(5):169–71.
36. Curtis AE, Smith TA, Ziganshin BA, Elefteriades JA. The mystery of the z-score. Aorta. 2016;4(04):124–30.
37. Connor R, Cardillo FA, Moss R, Rabitti F. Evaluation of Jensen-Shannon distance over sparse data. In: International Conference on Similarity Search and Applications, pp. 163–168 (2013). Springer
38. Panaretos VM, Zemel Y. Statistical aspects of Wasserstein distances. Annu Rev Stat Appl. 2019;6(1):405–31.
39. Xu Z, Tang J, Qi C, Yao D, Liu C, Zhan Y, Lukasiewicz T. Cross-domain attention-guided generative data augmentation for medical image analysis with limited data. Comput Biol Med. 2024;168: 107744.
40. Schober P, Boer C, Schwarte LA. Correlation coefficients: appropriate use and interpretation. Anesthesia & Analgesia. 2018;126(5):1763–8.
41. McKenzie AT, Katsyv I, Song W-M, Wang M, Zhang B. Dgca: a comprehensive r package for differential gene correlation analysis. BMC Syst Biol. 2016;10:1–25.
42. Bieniek J, Childress C, Swatski MD, Yang W. Cox-2 inhibitors arrest prostate cancer cell cycle progression by down-regulation of kinetochore/centromere proteins. Prostate. 2014;74(10):999–1011.
43. Sakai H, Kimura H, Otsubo K, Miyazawa T, Marushima H, Kojima K, Chosokabe M, Furuya N, Koike J, Fujii K, et al. Minichromosome maintenance 2 is an independent predictor of survival in patients with lung adenocarcinoma. Mol Clin Oncol. 2022;16(1):1–7.
44. Ji Z, Ji H. Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. Nucleic Acids Res. 2016;44(13):117–117.
45. Wang XM, Welsh TN. Tat-hum: trajectory analysis toolkit for human movements in python. Behav Res Methods. 2024;56(4):4103–29.
46. Wolf FA, Angerer P, Theis FJ. Scanpy: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19:1–5.
47. Guo L, Lin W, Zhang Y, Wang J. A systematic analysis revealed the potential gene regulatory processes of atra-triggered neuroblastoma differentiation and identified a novel ra response sequence in the ntrk2 gene. Biomed Res Int. 2020;2020(1):6734048.

Publisher's Note