


# scTCA: a hybrid Transformer-GNN architecture for imputation and denoising of scDNA-seq data

Zhenhua Yu <sup>1,2,\*</sup>, Furui Liu<sup>1</sup>, Yang Li<sup>1</sup>

<sup>1</sup>School of Information Engineering, Ningxia University, 750021 Ningxia, China

<sup>2</sup>Ningxia Key Laboratory of Artificial Intelligence and Information Security for Channeling Computing Resources from the East to the West, Ningxia University, 750021 Ningxia, China

\*Corresponding author. E-mail: zhyu@nxu.edu.cn

## Abstract

Single-cell DNA sequencing (scDNA-seq) has been widely used to unmask tumor copy number alterations (CNAs) at single-cell resolution. Despite that arm-level CNAs can be accurately detected from single-cell read counts, it is difficult to precisely identify focal CNAs as the read counts are featured with high dimensionality, high sparsity and low signal-to-noise ratio. This gives rise to a desperate demand for reconstructing high-quality scDNA-seq data. We develop a new method called scTCA for imputation and denoising of single-cell read counts, thus aiding in downstream analysis of both arm-level and focal CNAs. scTCA employs hybrid Transformer-CNN architectures to identify local and non-local correlations between genes for precise recovery of the read counts. Unlike conventional Transformers, the Transformer block in scTCA is a two-stage attention module containing a stepwise self-attention layer and a window Transformer, and can efficiently deal with the high-dimensional read counts data. We showcase the superior performance of scTCA through comparison with the state-of-the-arts on both synthetic and real datasets. The results indicate it is highly effective in imputation and denoising of scDNA-seq data.

**Keywords:** single-cell sequencing; intra-tumor heterogeneity; imputation; denoising; deep learning; transformer

## Introduction

Single-cell DNA sequencing (scDNA-seq) can provide unprecedented view of tumor copy number alterations (CNAs) at single-cell resolution [1], and thus enable a high-resolution profiling of clonal copy number substructure. Being a major genomic variation of tumor genome, CNA represents a change in the number of copies of a chromosomal section, and promotes cancer initialization and progression [2]. The size of CNAs can scale from a few kilobases to hundreds of megabases spanning an entire chromosome, and a cutoff of  $\leq 3\text{Mb}$  is usually used to define the size of focal CNAs [3]. To comprehensively understand the clonal copy number substructure, it is critical to precisely quantify CNAs with different scales.

A plenty of methods have been developed to detect single-cell CNAs from scDNA-seq data [4, 5]. Due to the low sequencing coverage and amplification bias of scDNA-seq, these methods typically use a large bin size (e.g. 500Kb) to measure read counts along the genome. Under this resolution, the read counts are less fluctuated and can be utilized to accurately call large-scale CNAs with size  $> 3\text{Mb}$ . As focal CNAs play an important role in driving tumor evolution in multiple cancer types [6, 7], precise identification of small-sized CNAs is equally important to get insights into tumor evolution. Unlike arm-level CNAs, it is much difficult to detect focal CNAs from scDNA-seq data. As shown in Fig. 1, single-cell read counts under 20-Kb resolution are heavily

complicated with zero values and technical noise (we divide the genome into non-overlapping bins with size of 20 Kb, and count the reads mapped to each bin. The left subfigure is generated by calculating the frequency of each read count value. As the reads are from paired-end sequencing, the measured read counts tend to be even counts.). To correctly segment the read counts and identify copy number states, an imputation and denoising preprocessing step is sorely required to recover the zero entries, and remove biologically unrelated factors from the read counts.

Unfortunately, there are no currently available methods specifically designed for imputation and denoising of scDNA-seq data. Extensive efforts have been made to address the same issues of single-cell RNA sequencing (scRNA-seq) data in the last few years [8–11]. As technical issues, such as low capture efficiency and sequencing coverage, make it difficult to capture transcripts of all expressed genes during the sequencing process, dropout events are more striking in scRNA-seq data [12]. Imputation methods aim to restore the values of zero entries associated with dropouts, and simultaneously make structural zeros (entries related to the genes that are not expressed) unchanged. In addition, biological variation and technical noise inevitably induce extensive fluctuation of the gene expression levels, and a denoising step is essential for correcting the non-zero entries [9]. To address these issues, a large arsenal of computational methods have been proposed to reconstruct the scRNA-seq data [13–23]. Patruno *et al.* [9] divide these methods into five groups: (1) relying on cell similarity or

Received: June 19, 2024. Revised: October 5, 2024. Accepted: October 29, 2024

© The Author(s) 2024. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited.

For commercial re-use, please contact [journals.permissions@oup.com](mailto:journals.permissions@oup.com)

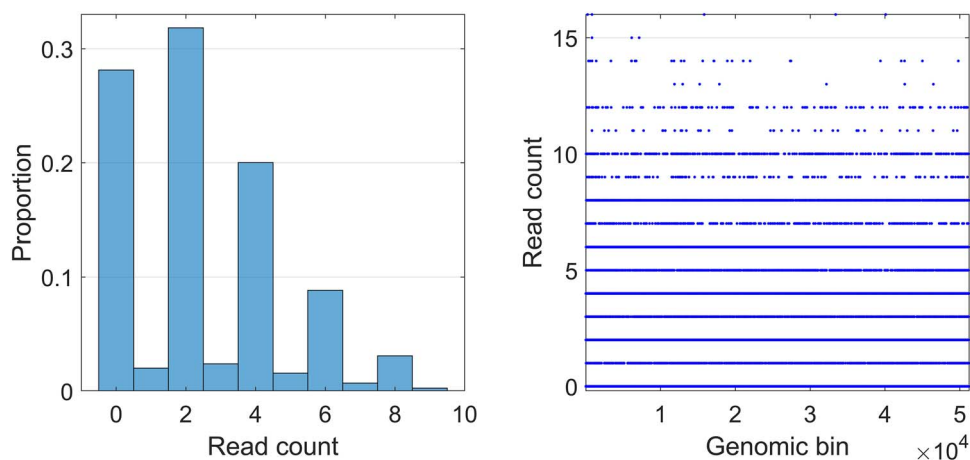


Figure 1. Read counts with 20-Kb resolution. The read counts are measured from a 10X Genomics breast cancer dataset available at [https://support.10xgenomics.com/single-cell-dna/datasets/1.0.0/breast\\_tissue\\_E\\_2k](https://support.10xgenomics.com/single-cell-dna/datasets/1.0.0/breast_tissue_E_2k).

gene similarity to smooth the expression profiles [15, 18, 19, 24, 25]; (2) exploiting external knowledge or data to enhance the quality of scRNA-seq data [20, 26, 27]; (3) machine learning-based correction for technical noise [13, 14, 16, 17, 21, 23]; (4) using matrix factorization to denoise scRNA-seq data [28, 29]; and (5) fitting the observed gene expression data with statistical models to perform imputation and denoising [22, 30, 31]. For instance, GE-Impute [18] builds a cell-cell similarity network and updates the similarity based on graph embeddings, then uses the averaged expression profile of the neighborhood cells to impute the dropouts for each cell; scMultiGAN [21] employs multiple generative adversarial networks (GAN) to impute the scRNA-seq data; scGGAN [20] additionally leverages bulk RNA sequencing data to construct gene network, then employs a graph-based GAN to impute the scRNA-seq data; bayNorm [30] uses negative binomial distribution to model the scRNA-seq data and infers the posterior distributions under a Bayesian framework; and, finally, TsImpute [22] identifies and imputes dropouts by fitting the data with zero-inflated negative binomial distribution, then performs distance weighted imputation of the data.

Despite that scRNA-seq based methods perform well in imputation and denoising of gene expression data, they may not be suitable for processing single-cell read counts data derived from scDNA-seq. The main reasons lie in two aspects: (1) most of the scRNA-seq based methods take an assumption of independency between genes, while this does not hold for scDNA-seq data as copy numbers are characterized by both local (e.g. a CNA may affect multiple adjacent genes) and global (e.g. co-occurred CNAs across the genome in each tumor clone) patterns; (2) statistical distributions of gene expression and read counts data are different due to the inherent technical difference between the two omics, which implies the preconceived model structures or assumptions in scRNA-seq data analysis may not hold for scDNA-seq data. As convolutional neural network (CNN) is powerful in learning local features, and Transformer model [32] is more effective in identifying global patterns, hybrid architectures that combine Transformer with CNN can simultaneously capture the local and global patterns of copy numbers, and thus enable better imputation and denoising of read counts. Hybrid Transformer-CNN architectures have shown high performance in fusing multi-view features [33], especially in computer vision tasks [34, 35]. Considering the complex characteristics of scDNA-seq data, i.e. high dimensionality, high sparsity and low signal-to-noise ratio,

how to implement an efficient Transformer-CNN architecture is still an unresolved problem in scDNA-seq data analysis.

In this study, we develop a novel method called scTCA to improve the quality of scDNA-seq data (Fig. 2). In scTCA, an autoencoder (AE) is used to encode the observed single-cell read counts into latent embeddings, and then reconstruct read counts through decoding. To exploit correlations between genes for precise recovery of the read counts, we employ a two-branch Transformer-CNN mixture architecture to extract and fuse multi-view features. The residual CNN block can identify and utilize the local patterns to recover the data, while the Transformer block is used to build connections between non-adjacent genes, and thus enhance the data reconstruction quality by capturing global patterns. As Transformer models typically suffer high computational complexity when applied to high-dimensional sequential data, we implement the Transformer block as a two-stage attention module including a step-wise self-attention layer and a window Transformer as adopted in Swin-T [36], which reduces the computational burden. Considering copy numbers dominantly encompass local patterns, only the residual CNN blocks are used in the decoder network to reconstruct the read counts with censored regression models, and this enables higher computational efficiency. We showcase the superior performance of scTCA on synthetic as well as real datasets, and the results suggest our method surpasses the state-of-the-art methods in multiple performance metrics.

## Materials and methods

### Overview of scTCA

The backbone of scTCA is an autoencoder that receives single-cell read counts as input, and outputs imputed and denoised read counts (Fig. 2). Each encoder layer of the AE is built based on a novel Transformer-CNN hybrid architecture, and the decoder layer consists of a residual block followed by an upsampling convolutional component. The Transformer-CNN module contains a pair of parallel residual CNN and Transformer blocks, extracting local and global features from the input. To reduce computational complexity, the input feature map is equally split into two parts along the channel dimension, which are fed into the CNN and Transformer, respectively. A concatenation operation and a  $1 \times 1$  convolutional layer are then employed to fuse the two types of features into a semantically meaningful feature map.

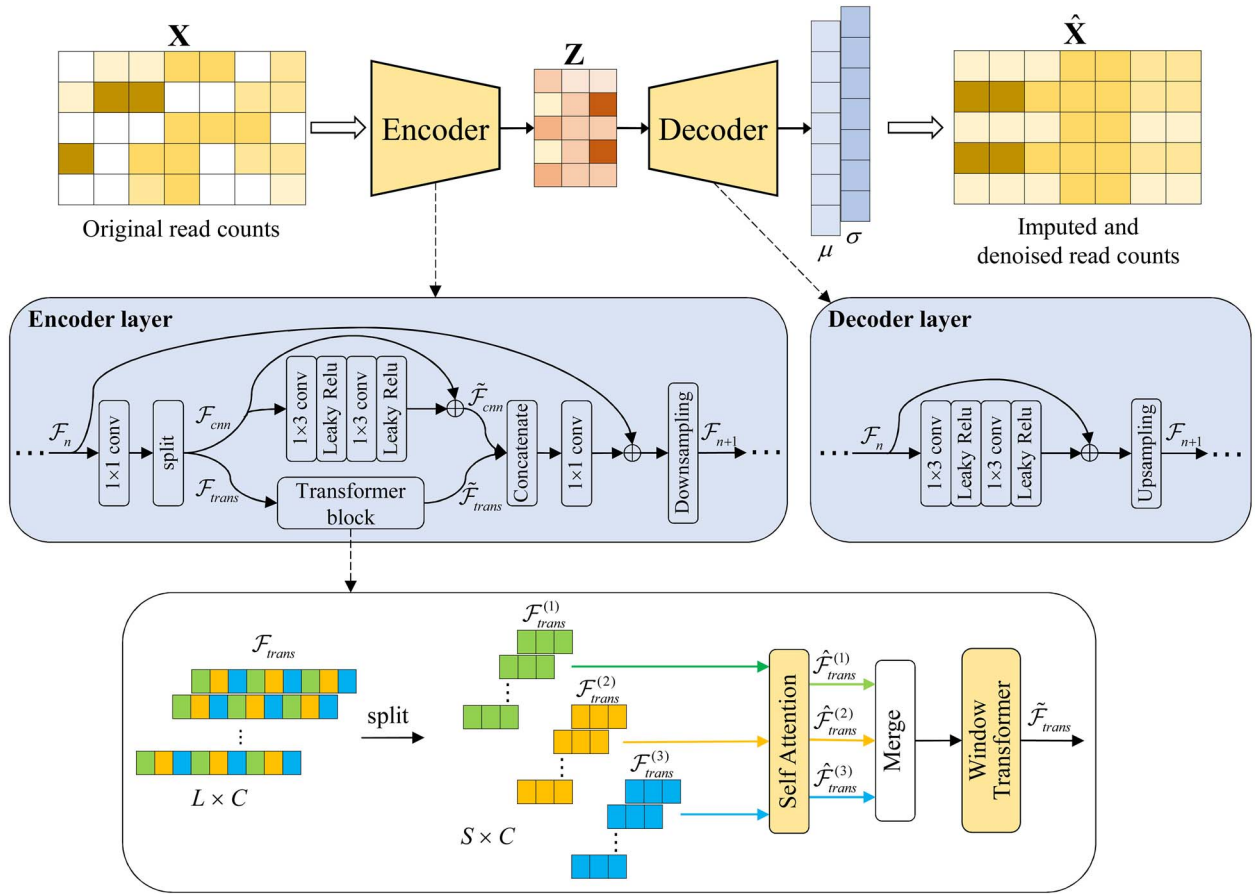


Figure 2. The model architecture of scTCA. The backbone of scTCA is an autoencoder that receives single-cell read counts as input, and outputs imputed and denoised read counts. In each encoder layer, the input feature map is first processed with a  $1 \times 1$  convolutional layer, then equally split into two parts along the channel dimension; the two sub-feature maps are sent to a Transformer-CNN mixture block to learn local and global features; the features are concatenated and fused with a  $1 \times 1$  convolutional layer; a skip connection between the input and fused feature map is built, followed by a down-sampling layer to reduce the feature map size. In each decoder layer, a residual CNN block and an up-sampling layer are employed to gradually recover the data. The decoder network outputs the tobit parameters ( $\mu, \sigma$ ) that are used to measure the likelihood of input read counts. To efficiently deal with the high-dimensional read counts data, the Transformer block is implemented as a two-stage attention module, including a step-wise self-attention layer and a window Transformer.

As the single-cell read counts are high dimensional sequential data (e.g. sequence length is as large as  $\sim 150,000$  under a bin size of 20 Kb for human genome), conventional Transformer models such as ViT [37] and Swin-T [36] are computationally expensive when directly applied to this type of data. To resolve this challenge, we propose a combination of stepwise self-attention mechanism and window Transformer to capture global features with affordable computational costs. Specifically, the sequence is split into several subsequences under a predefined step size, and a self-attention module is applied to each of the subsequences, the resulted feature maps are then merged and fed into a window Transformer with the window size equal to the step size. This light-weight Transformer block is of higher computational efficiency, and capable of building connection between any pair of distant positions. The following sections give a detailed description about the workflow of scTCA.

## Data preprocessing

The whole genome is divided into non-overlapping and fix-sized bins, and read count of each bin for each cell is extracted from the BAM file. We exclude bins with extreme GC-content percent ( $< 10\%$  or  $> 90\%$ ) from downstream analysis, and this results in a  $N \times M$  read count matrix  $\mathbf{D}$  of  $N$  cells and  $M$  bins. As sequencing

coverage may vary between single cells, to make read counts comparable between cells, we normalize the read counts of each cell with the mean read count:

$$\mathbf{X}_{ij} = m \cdot \frac{\mathbf{D}_{ij}}{m_i} \quad (1)$$

where  $\mathbf{X}$  denotes the resulted normalized read count matrix,  $m_i$  represents mean read count of the  $i$ th cell, and  $m$  denotes the median value of  $m_i$  ( $1 \leq i \leq N$ ).

## Encoder of the AE

As read count is a proxy of the copy number, the local and non-local characteristics of copy numbers can be utilized to impute and denoise the read counts. Considering CNN can well capture local patterns while transformers are more effective in extracting non-local information, we propose an efficient hybrid architecture that pairs a residual convolutional block and a Transformer block, to extract multi-view features of read counts. Specifically, the  $(n+1)$ th encoder layer of the AE takes  $\mathcal{F}_n$  with size  $B \times 2C \times L$  as input (here  $B$  and  $L$  denote batch size and sequence length, respectively), and outputs feature map  $\mathcal{F}_{n+1}$  by three steps:

1. use a  $1 \times 1$  convolutional layer to fuse  $\mathcal{F}_n$ , and split the resulted feature map into two equally sized tensors  $\mathcal{F}_{cnn}$  and  $\mathcal{F}_{trans}$  (both with size  $B \times C \times L$ ) along the channel dimension;
2.  $\mathcal{F}_{cnn}$  is sent to the residual block to generate local feature map  $\tilde{\mathcal{F}}_{cnn}$ :

$$\tilde{\mathcal{F}}_{cnn} = \mathcal{F}_{cnn} + \text{Conv}(\mathcal{F}_{cnn}) \quad (2)$$

where  $\text{Conv}$  denotes convolutional layers with  $1 \times 3$  kernel and the LeakyReLU activation function.  $\mathcal{F}_{trans}$  is sent to the Transformer block to get global feature map  $\tilde{\mathcal{F}}_{trans}$ . The Transformer block is implemented as a two-stage attention module, including a stepwise self-attention layer and a window Transformer. Specifically, give a step size  $W$ ,  $\mathcal{F}_{trans}$  is split into several small subsequences  $\{\mathcal{F}_{trans}^{(1)}, \mathcal{F}_{trans}^{(2)}, \dots\}$  along the positional dimension:

$$\mathcal{F}_{trans}^{(i)} = \mathcal{F}_{trans}(\cdot, \cdot, [i, i+W, i+2W, \dots]) \quad (3)$$

The step size  $W$  is determined by  $\lceil \frac{L}{\tau} \rceil$ , here  $\tau$  denotes the pre-defined maximum length of the subsequences. Each  $\mathcal{F}_{trans}^{(i)}$  is transposed to a tensor with size  $B \times L_i \times C$ , and sent to the self-attention layer to get  $\hat{\mathcal{F}}_{trans}^{(i)}$ :

$$\hat{\mathcal{F}}_{trans}^{(i)} = \text{softmax} \left( \frac{\mathcal{F}_{trans}^{(i)} (\mathcal{F}_{trans}^{(i)})^T}{\sqrt{C}} \right) \mathcal{F}_{trans}^{(i)} \quad (4)$$

We then merge  $\{\hat{\mathcal{F}}_{trans}^{(1)}, \hat{\mathcal{F}}_{trans}^{(2)}, \dots\}$  to create a tensor  $\hat{\mathcal{F}}_{trans}$  with size  $B \times L \times C$ :

$$\hat{\mathcal{F}}_{trans}(\cdot, [i, i+W, i+2W, \dots], \cdot) = \hat{\mathcal{F}}_{trans}^{(i)} \quad (5)$$

After that, a window Transformer adopted in Swin-T [36] is introduced to process  $\hat{\mathcal{F}}_{trans}$  with window size equal to  $W$ , and generate global feature map  $\tilde{\mathcal{F}}_{trans}$ .

3. with a concatenation operation,  $\tilde{\mathcal{F}}_{cnn}$  and  $\tilde{\mathcal{F}}_{trans}$  are merged into a tensor with size  $B \times 2C \times L$ . And then, a  $1 \times 1$  convolutional layer is used to fuse features, and the skip connection between  $\mathcal{F}_n$  and the output is built. Finally, a convolutional layer with stride of 2 is used for down-sampling, yielding the tensor  $\mathcal{F}_{n+1}$  with size  $B \times C \times L'$ .

The encoder network outputs a tensor with size  $B \times C_e \times L_e$ , which is flattened and sent to a fully connected layer to generate latent representations. We first reshape the read count matrix  $\mathbf{X}$  to a tensor with size  $N \times 1 \times M$ , and then employ a convolutional layer with 128 channels and stride of 2 to get the input of the encoder. The encoder finally generates the cell embeddings  $\mathbf{Z}$  with size  $N \times d$ , here  $d$  denotes the latent dimension.

## Decoder of the AE

Given the cell embeddings  $\mathbf{Z}$ , the decoder network outputs imputed and denoised read counts. We simply employ a residual CNN block followed by an up-sampling convolution to define the structure of the decoder layer, and do not exploit the complex hybrid Transformer-CNN module due to two reasons: (1) local features play a more important role in reconstructing single-cell read counts as copy numbers dominantly exhibit local patterns; (2) CNN-based modules are more effective in capturing local features and computationally more efficient than the hybrid Transformer-CNN architectures. We employ censored regression models to characterize the distribution of  $\mathbf{X}$ , and define the

likelihood function as follows:

$$f(x|\mu, \sigma) = \left( \frac{1}{\sigma} \varphi \left( \frac{x-\mu}{\sigma} \right) \right)^{I(x)} \left( 1 - \Phi \left( \frac{\mu}{\sigma} \right) \right)^{1-I(x)} \quad (6)$$

where  $I(x)$  is an indicator function that takes value of 1 when  $x > 0$ ,  $\varphi$  denotes the standard normal probability density function, and  $\Phi$  represents the standard normal cumulative distribution function. The normal parameters  $(\mathbf{M}, \mathbf{\Sigma})$  are inferred from the output of the decoder:

$$\begin{aligned} \mathbf{M} &= \exp(\text{Conv}_\mu(f_D(\mathbf{Z}))) \\ \mathbf{\Sigma} &= \exp(\text{Conv}_\sigma(f_D(\mathbf{Z}))) \end{aligned} \quad (7)$$

where  $f_D(\mathbf{Z})$  is the output of the decoder,  $\text{Conv}_\mu$  and  $\text{Conv}_\sigma$  represent two convolutional layers.

## Optimization of the model

Given the parameters of the censored regression models, the reconstruction loss is defined as the negative log-likelihood of read counts:

$$\mathcal{L} = - \sum_{i=1}^N \sum_{j=1}^M \log(f(\mathbf{X}_{ij} | \mathbf{M}_{ij}, \mathbf{\Sigma}_{ij})) \quad (8)$$

We use the ‘Adam’ algorithm to optimize the weights of the AE by minimizing the reconstruction loss. After the model converges, the parameter  $\mathbf{M}$  is treated as the imputed and denoised read counts. For implementation of the model, we set the number of encoder/decoder layers to 2, and set latent dimension  $d$  to 5.

## Performance evaluation

As there are no specifically developed methods for imputation and denoising of scDNA-seq data, we compare scTCA to 9 scRNA-seq based methods, including scImpute [38], DeepImpute [26], bayNorm [30], GE-Impute [18], SCDD [19], scMultiGAN [21], Tslmpute [22], MAGIC [39], and CarDEC [40]. These methods all perform well in imputing dropouts and/or denoising scRNA-seq data, but their effectiveness on scDNA-seq data is underdetermined, which inspires us to make a comprehensive comparison between scTCA and the competitors. Details about parameter settings of each method are given in Supplementary Methods.

## Simulations

Despite that several methods [41, 42] are currently available for simulating tumor scDNA-seq data, they all involve in time-consuming generation of FASTQ and BAM files. We propose a simulation pipeline that directly produces single-cell read counts data without emulating raw sequencing reads. We generate various datasets under different configurations of the simulation-related parameters. Specifically, given the number of clones  $K$ , number of cells  $N$ , number of bins  $M$  and bin size  $s$ , the simulation pipeline contains four sub-modules to generate single-cell data: (1) divide the genome into non-overlapping copy number segments; (2) build a clonal copy number tree containing  $K+1$  nodes to represent the phylogenetic relationship between tumor clones, and emulate copy number changes along the edges of the tree; (3) assign cells to the nodes of the tree to generate different-sized clonal clusters; and (4) produce read counts data for each cell according to the copy number profile. A detailed illustration of the simulation pipeline is given in Algorithms S1-S3 of Supplementary material. The simulation algorithms output: (1) a  $S \times 3$  matrix defining start position, end position and length



of each copy number segment, here  $S$  denotes the number of segments; (2) a  $K \times M$  matrix representing copy number profile (CNP) of each tumor clone; (3) a  $1 \times N$  vector indicating clonal assignment of each cell; and (4) a  $N \times M$  matrix  $\mathbf{D}$  containing read counts data of all cells.

The main simulation parameters are configured as follows:  $K \in \{7, 9, 11\}$ ,  $N \in \{500, 1000\}$ ,  $M \in \{12\ 000, 18\ 000, 24\ 000\}$ , and  $s = 10\text{Kb}$ . Other simulation-related parameters are also specified. For instance, the minimum and maximum sizes of copy number segments are set to 0.2Mb and 100Mb, respectively; the weights of the segments with size of 1Mb~20Mb are set to 1, and a weight of 0.5 is assigned to other segments when emulating copy number segments; the maximum copy number of CNAs is set to 10; the distribution of dropout rate is inferred from a 10X Genomics dataset obtained from [https://support.10xgenomics.com/single-cell-dna/datasets/1.0.0/breast\\_tissue\\_E\\_2k](https://support.10xgenomics.com/single-cell-dna/datasets/1.0.0/breast_tissue_E_2k), and used to mimic dropout events when generating single-cell read counts. More details about the parameter settings are given in Algorithms S1–S3.

### Real datasets

We also obtain five tumor datasets (Breast Tissue nuclei section {A, B, C, D, E} 2000 cells) from the 10X Genomics (<https://www.10xgenomics.com/datasets>). The CNP matrix and clonal assignments of cells for each dataset are obtained from <https://github.com/raphael-group/chisel-data>. We extract single-cell read counts from the BAM files, and exclude bins with  $<0.1$  or  $>0.9$  GC-content percent to get the read count matrix  $\mathbf{D}$  for downstream analysis.

### Performance metrics

Given the ground truth (GT) cell labels, CNP matrix  $\mathbf{E}$  of all cells, original read count matrix  $\mathbf{D}$  and reconstructed read count matrix  $\hat{\mathbf{D}}$  of each method, we employ three performance metrics to assess the methods by following a previous study [9]: 1) the Spearman correlation coefficient between  $\hat{\mathbf{D}}$  and  $\mathbf{E}$  calculated with the zero entries in  $\mathbf{D}$ . This metric is used to verify if each method effectively corrects dropout events; 2) the difference between the two Spearman correlation coefficients (denoted by  $\Delta\text{Spearman}$ ) calculated for  $(\hat{\mathbf{D}}, \mathbf{E})$  and  $(\mathbf{D}, \mathbf{E})$ . This measure is used to indicate how well each method recovers the true read count profiles; and 3) the difference between the average silhouette score (denoted by  $\Delta\text{Silhouette}$ ) calculated from  $\hat{\mathbf{D}}$  and that measured from  $\mathbf{D}$ , using the GT labels to group the cells. Higher silhouette coefficient indicates higher consistency of the clustering with the GT labels. This metric quantifies the effectiveness of each method in increasing intra-cluster similarity and enlarging the inter-cluster distance. We also employ the coefficient of variation (CV) as another metric to measure the quality of data imputation and denoising. Specifically, for each copy number state of each cell, the difference (denoted as  $\Delta\text{CV}$ ) between the CV of original data and that of reconstructed data is calculated, and mean  $\Delta\text{CV}$  across all copy number states of all cells is calculated to assess the methods.  $\Delta\text{CV}$  is used to indicate how well a method reduces the dispersion degree of single-cell read counts, and the higher the  $\Delta\text{CV}$ , the better the performance.

## Results

### Performance evaluation on the synthetic datasets

We first assess the performance of scTCA and other methods on the synthetic datasets, and the  $\Delta\text{Silhouette}$  scores are shown in Fig. 3. scTCA consistently achieves high  $\Delta\text{Silhouette}$  scores

under different data sizes and levels of heterogeneity, surpassing the competitors by a large margin. For instance, the average  $\Delta\text{Silhouette}$  score of scTCA is 0.891 on datasets with 500 cells, 7 clones and 24 000 bins, while the second-best method MAGIC yields the average  $\Delta\text{Silhouette}$  score of 0.625. On datasets with 7 clones, scTCA shows enhanced ability of disentangling cellular heterogeneity when the number of cells or bins increases (e.g. the mean  $\Delta\text{Silhouette}$  score increases from 0.789 at  $M = 12\ 000$  to 0.912 at  $M = 24\ 000$  on datasets with 1000 cells). When the number of clones increases to 9, scTCA still achieves consistently higher  $\Delta\text{Silhouette}$  scores than other methods. scTCA exhibits degraded performance as the number of bins increases when there are 11 clones (e.g. the mean  $\Delta\text{Silhouette}$  score decreases from 0.902 at  $M = 12\ 000$  to 0.825 at  $M = 24\ 000$  on datasets with 1000 cells). This implies our method may suffer insufficient modeling of long sequence data at higher data heterogeneity, and similar performance degradation is also observed for other methods. In addition, the results indicate both the imputation and denoising steps are required to accurately unscramble the cellular heterogeneity from single-cell read counts, as demonstrated by the higher  $\Delta\text{Silhouette}$  scores of MAGIC, CarDEC, and scTCA compared to other methods. To further investigate how well the cell subpopulations are separated with each other, we visualize the imputed and denoised read counts of each method using tSNE as shown in Fig. 4 and Supplementary Fig. 1. Compared to other methods, scTCA provides compact cell clusters and clear separation of different subpopulations.

The Spearman correlation coefficients that measure the performance of each method in correcting dropout events are shown in Supplementary Fig. 2. On datasets with 500 cells, our method shows the highest Spearman coefficients, followed by MAGIC, CarDEC, scImpute, DeepImpute, and scMultiGAN. For instance, for datasets with 11 clones and 12 000 bins, the average Spearman coefficient of scTCA is 0.757, while those of MAGIC, CarDEC, scImpute, DeepImpute and scMultiGAN are 0.752, 0.722, 0.678, 0.532, and 0.428, respectively. GE-Impute and TsImpute are less effective in recovering the zero entries of read counts. With the increased number of cells, all methods especially GE-Impute tend to deliver more accurate imputation results. For instance, the average Spearman coefficient of GE-Impute increases from 0.077 at  $N = 500$  to 0.385 at  $N = 1000$  when the number of clones is 7, and that of scTCA increases from 0.658 to 0.67. In general, scTCA, MAGIC and CarDEC exhibit similar overall average Spearman coefficients. These results imply scTCA performs robustly against the change of data size and heterogeneity.

Next, we assess the performance of each method in recovering the true read count profiles, and the  $\Delta\text{Spearman}$  scores are shown in Supplementary Fig. 3. In general, our method surpasses other methods across all test conditions. The overall average  $\Delta\text{Spearman}$  of scTCA is 0.427, while the second-best method MAGIC yields the overall average  $\Delta\text{Spearman}$  of 0.414. As the  $\Delta\text{Spearman}$  metric measures the overall Spearman coefficient by considering both the zero and non-zero entries of read counts, methods that only conduct imputation of the data are undoubtedly at a disadvantage. For instance, scImpute, DeepImpute, GE-Impute, SCDD and TsImpute provide the average  $\Delta\text{Spearman}$  scores of 0.219, 0.161, 0.092, 0.103, and 0.027, respectively, falling evidently short of MAGIC, CarDEC and scTCA. The differences are also obvious when investigating the imputed and denoised read counts of each method (as shown in Fig. 5). Although MAGIC and CarDEC significantly improve the quality of single-cell read counts, they fall short of scTCA in mapping read counts to the GT copy number states. For instance, they do not well reconstruct the

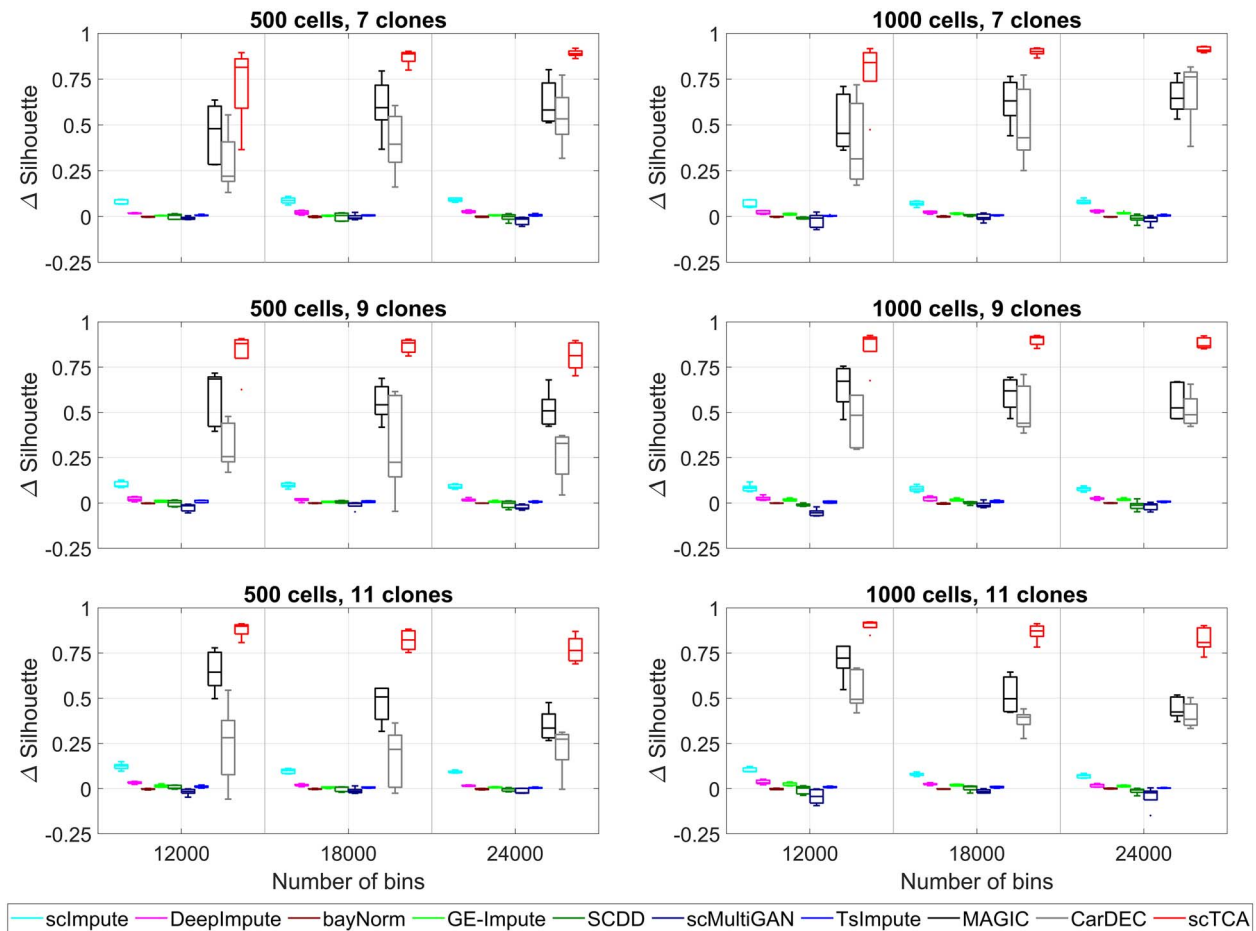


Figure 3.  $\Delta$ Silhouette scores of the methods on the simulated datasets. The number of cells  $N$  in {500, 1000}, number of clones  $K$  in {7, 9, 11}, and number of bins  $M$  in {12 000, 18 000, 24 000} are used to generate the simulated datasets.

quantitative relationship between copy number and read count in small segments harboring focal CNAs (as observed for cluster 4 in Fig. 5), which may lead to biased estimation of the copy number states in downstream analysis. By comparison, our method correctly rebuilds the read counts associated with different copy number states, and largely reduces the variance of the data. These improvements make it easier to get unbiased inference of the single-cell copy number profiles.

We proceed to assess how well each method reduces the variance of single-cell read counts data, and the results are shown in Supplementary Fig. 4. scMultiGAN, MAGIC and CarDEC show good performance with average  $\Delta$ CV scores of 0.698, 0.724, and 0.711, respectively. By comparison, our method surpasses the competitors by achieving average  $\Delta$ CV score of 0.783. We also assess the runtime performance by comparing the peak memory and time required by each method. As shown in Supplementary Table 1, our method tends to be computationally less efficient than conventional statistical methods as the convolutional and Transformer blocks of scTCA involve in extensive computation.

Finally, we conduct several ablation experiments to examine the effectiveness of model architecture of scTCA. To demonstrate the critical role of the hybrid Transformer-CNN module, we generate two model variants, denoted as scTCA-wot and scTCA-woc, by switching off the Transformer and residual blocks, respectively. In addition, we change the model architecture by employing MSE-based loss function (denoted as scTCA-mse), to verify the effectiveness of the censored regression model in data reconstruction.

Performance comparison is conducted between the full model and the three variants under same hyper-parameters. As shown in Supplementary Fig. 5, the full model generally outperforms the variants in the different performance metrics, especially in deciphering the cellular heterogeneity. For instance, the mean  $\Delta$ Silhouette scores of scTCA, scTCA-wot, scTCA-woc and scTCA-mse are 0.823, 0.814, 0.576, and 0.704, respectively. This verifies the advantage of the hybrid Transformer-CNN module and the censored regression model in imputation and denoising of single-cell read counts. Furthermore, when capturing non-local information with a two-stage attention module in scTCA, the maximum length (denoted as  $\tau$ ) of the subsequences may have a significant impact on learning of non-local features, therefore we tune the value of  $\tau$  to investigate how it affects the performance of scTCA. As depicted in Supplementary Fig. 6, our method performs robustly against the change of  $\tau$ , and yields consistently high Spearman coefficient,  $\Delta$ Spearman,  $\Delta$ Silhouette and  $\Delta$ CV scores across the different conditions. As such, we set the default value of  $\tau$  to 500, and conduct all experiments under this setting.

## Performance evaluation on the real datasets

The breast cancer datasets (denoted as A, B, C, D, and E) consist of 2008, 1693, 1254, 1384, and 1446 cells, respectively. It is reported that there are three subpopulations in dataset A that show distinct copy number profiles [43]. Datasets B and C contain six cell subpopulations, while datasets D and E are both comprised of five

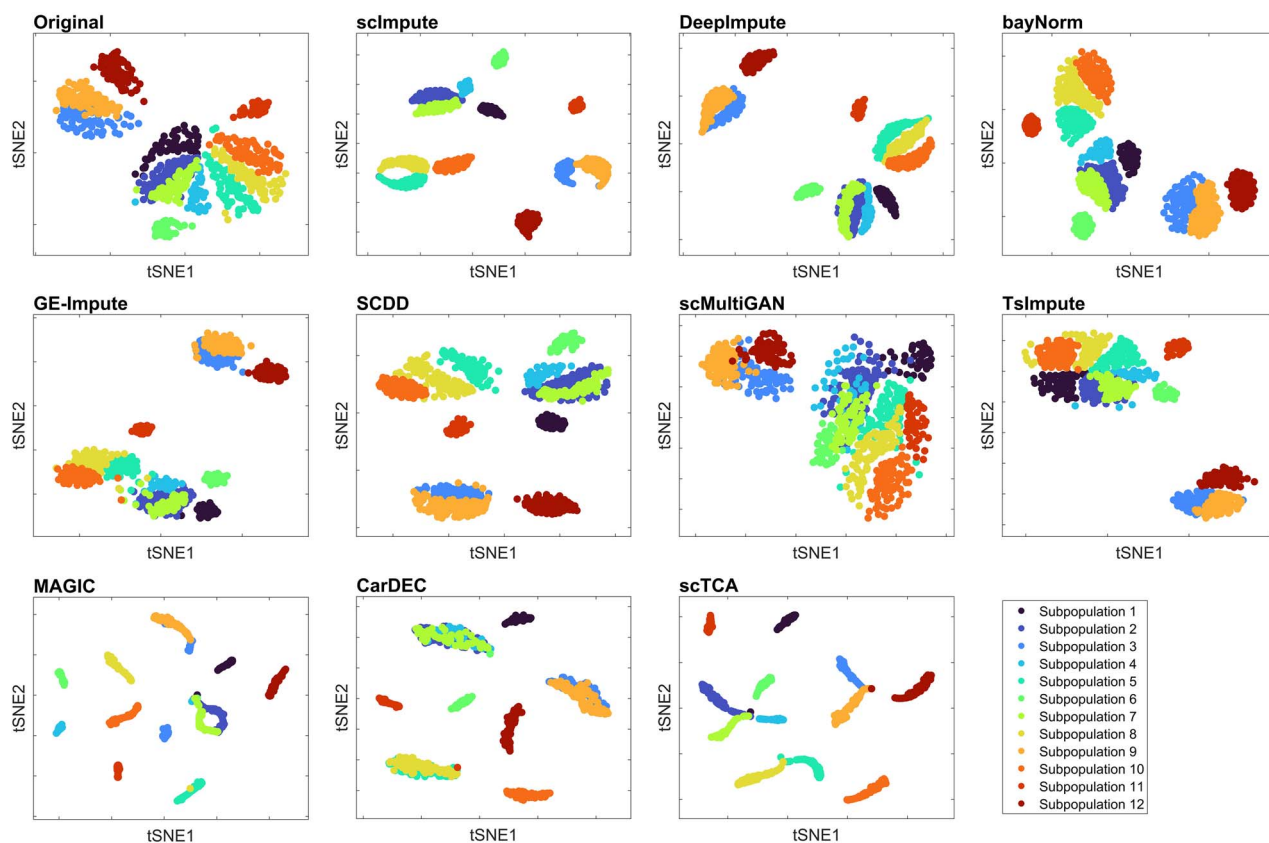


Figure 4. Visualization of the imputed and denoised read counts using tSNE on a simulated dataset with 12 cell subpopulations. The dimension of read counts data is first reduced to 50 with the principal component analysis, and then reduced to 2 with tSNE. The cell subpopulations include a normal subpopulation and 11 tumor clones. The  $\Delta$ Silhouette scores of the methods are 0.076, 0.024,  $-0.002$ , 0.015, 0.013,  $-0.025$ , 0.002, 0.645 (MAGIC), 0.398 (CarDEC), and 0.895 (scTCA), respectively.

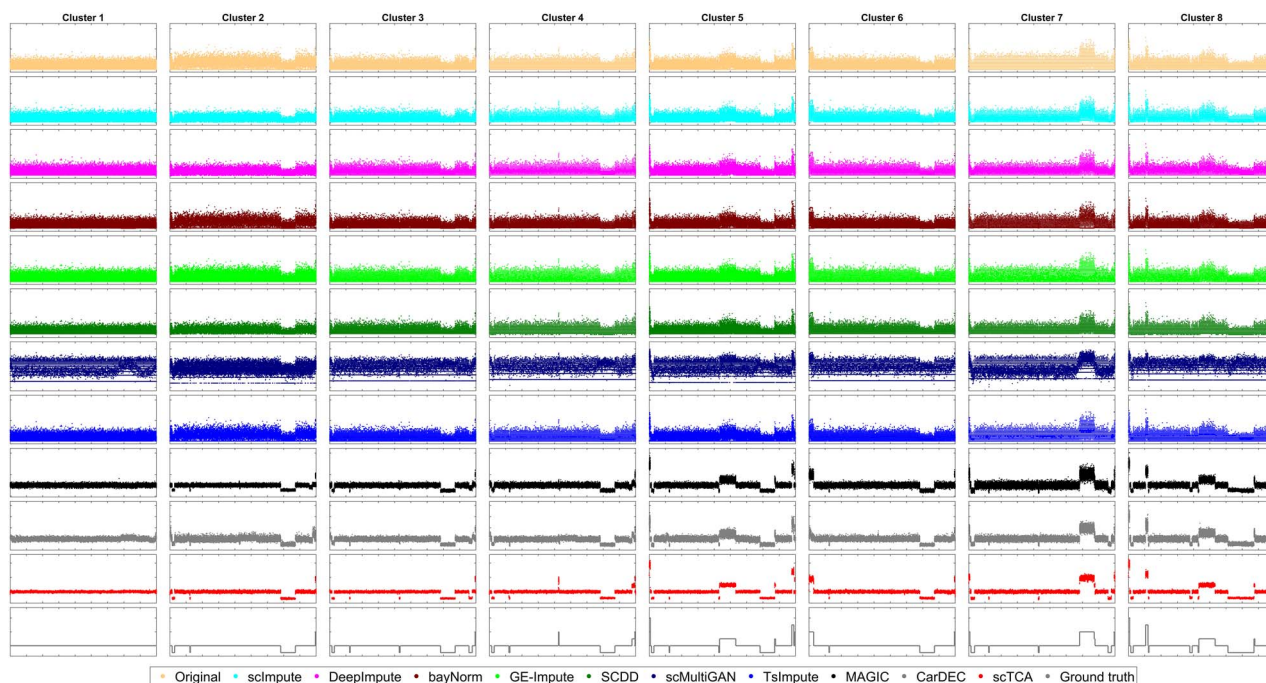


Figure 5. Comparison between the reconstructed read counts and GT copy number profiles on a simulated dataset with eight cell subpopulations.

clonal clusters. These datasets are derived from different tumor sections of a breast cancer patient, and we obtain the BAM files that contain read alignments of the sequenced cells from the 10X Genomics (<https://www.10xgenomics.com/datasets>). As the

differences of copy number profiles between the tumor clones mainly come from the chromosomes 2, 3, 4, 5, 6, 8, and 11, we only extract single-cell read counts from these chromosomes for downstream analysis. The bin size is set to 20 Kb when



counting reads in each bin, and bins with  $<0.1$  or  $>0.9$  GC-content percent are excluded since they are considered as confounders and outliers. The GT copy number profiles and cell labels are obtained from <https://github.com/raphael-group/chisel-data>. Given the single-cell read counts, we use scTCA as well as other methods to impute the dropouts and denoise the data. Due to runtime errors or excessive memory consumption, we fail to get the results of scImpute, scMultiGAN, and Tsimpute on these datasets, therefore exclude them from performance evaluation.

As shown in Fig. 6, our method achieves generally better performance than the competitors, especially in recovering the intra-tumor heterogeneity. For imputation of the dropouts, DeepImpute, SCDD, MAGIC, CarDEC, and scTCA get high Spearman coefficients (mean values are 0.753, 0.746, 0.75, 0.762, and 0.826, respectively) on datasets B-E, while bayNorm and GE-Impute show lower accuracy. For recovering the true read count profiles, scTCA outperforms MAGIC and CarDEC by delivering higher  $\Delta$ Spearman scores (the mean  $\Delta$ Spearman scores are 0.439, 0.417, and 0.418, respectively). As DeepImpute, bayNorm, GE-Impute, and SCDD do not denoise the data, they yield low  $\Delta$ Spearman scores. In addition, all methods exhibit low Spearman coefficients and  $\Delta$ Spearman scores on dataset A, which may result from the fact that dataset A is dominantly composed of normal cells (1948 out of 2008 cells), and thus most of the entries of copy number profiles are a fixed value of 2 when measuring the Spearman coefficients. To check if each method correctly builds the quantitative relationship between copy number and read count, we visualize the reconstructed read counts of all methods as depicted in Fig. 7 and Supplementary Figs 7–10, where the cell that has the nearest average distance to other cells in each clonal cluster is selected to make the comparison. As shown in Fig. 7, scTCA significantly improves the data quality by making the read counts with same copy number state more centralized, and correctly quantifies the relationship between read counts of different copy number states. Despite that MAGIC and CarDEC have superior ability of denoising the single-cell read counts, they show degraded performance in mapping read counts to correct copy number states, as observed for clonal cluster 4 in Fig. 7. scTCA also makes it easier to detect focal CNAs (as shown in Supplementary Fig. 11). For instance, the cell 'AAAGTAGCATGGCACC' from dataset E harbors a focal CNA with copy number of 1 on chromosomal region 5:68860001-70660000, and the original read counts of this region are extensively fluctuated, which impedes accurate estimation of the copy number, while it is much easier to infer the copy number from the reconstructed read counts.

The advantage of scTCA is more evident in recovering the cellular heterogeneity (Fig. 6). For instance, the mean  $\Delta$ Silhouette score of scTCA is 0.779, while the corresponding metric value of the second-best method CarDEC is 0.515. MAGIC achieves the mean  $\Delta$ Silhouette score of 0.285, falling short of CarDEC and scTCA by a large margin. As observed on simulated datasets, methods that only conduct imputation of the data are much less effective in revealing the cellular heterogeneity from single-cell read counts. scTCA shows relatively lower  $\Delta$ Silhouette scores on datasets B-E when compared to the simulated datasets. As our simulation strategy is unable to fully model the complex biological process, the simulated data are less noised than the real data, which makes it easier to recover the cellular heterogeneity from the synthetic datasets. Comparing the  $\Delta$ CV scores of the methods also shows the advantage of scTCA in reducing dispersion of single-cell read counts. For instance, the average  $\Delta$ CV scores of MAGIC, CarDEC and scTCA are 0.538, 0.533, and 0.599, respectively.

We further examine the robustness of scTCA based on subsampling strategy. Specifically, sub datasets are generated from the dataset E under subsampling ratios of 0.2, 0.4, 0.6, and 0.8. For each candidate subsampling ratio, five sub datasets are produced by conducting five trials of random sampling. To make a comparison to other methods, we also get results of MAGIC and CarDEC on these datasets. As shown in Supplementary Fig. 12, our method achieves similar performance as observed on the full dataset when subsampling ratio is larger than 0.2, surpassing MAGIC and CarDEC across different subsampling ratios.

Finally, we visualize the imputed and denoised read count profiles of all method using tSNE, as shown in Fig. 8 and Supplementary Figs 13–16. Compared to the competitors, scTCA shows better ability to distinguish between the tumor clones. For instance, on dataset E (Fig. 8), despite that scTCA divides the subpopulations 1 and 2 into subclusters, it still gives clear separation of different subpopulations; on dataset A (Supplementary Fig. 13), our method correctly identifies two smaller tumor clones and clearly separates them from each other, while other methods tend to confuse them with the normal subpopulation.

## Discussion

Precisely detecting CNAs with different scales, including arm-level and focal CNAs, is important to identify cancer driver genes. scDNA-seq can be used to accurately unmask CNAs of both major and low-prevalence tumor clones from read counts, free of complex deconvolution as required in bulk sequencing. Despite that current CNA calling methods perform well in identifying arm-level CNAs, they may fail to deliver accurate inference of focal CNAs, due to the high dimensionality, high sparsity and low signal-to-noise ratio of the read counts in small-sized bins (e.g.  $\leq 20$  Kb). This requires an imputation and denoising step prior to CNA calling, such that focal CNAs can also be detectable with current off-shelf computational methods.

We introduce a new method scTCA to reconstruct high-quality single-cell read counts using an autoencoder framework. To exploit both local and non-local patterns of copy numbers in learning cell embeddings, we propose a novel Transformer-CNN hybrid architecture, containing a pair of parallel residual CNN and Transformer blocks, to extract and fuse multi-view features of the read counts. Unlike conventional Transformers such as ViT and Swin-T, the Transformer block in scTCA contains a stepwise self-attention layer followed by a window Transformer, and is computationally efficient to process the high-dimensional read counts. When reconstructing high-quality single-cell read counts through decoding, the censored regression models are employed to characterize the data distributions by effectively handling the zero values. On synthetic and real datasets, we showcase the superior performance of scTCA, and demonstrate its stronger ability of imputing and denoising single-cell read counts compared to existing scRNA-seq based methods.

There are two potential improvement directions to further enhance scTCA. First, scTCA does not utilize cell similarities when learning latent embeddings from sparse read counts, while exploiting information from neighborhood cells with a graph convolutional network may provide more semantically meaningful representations, as well as more precise reconstruction of the read counts. Thus, a hybrid architecture combining graph convolutional networks with Transformers is an effective alternative to the Transformer-CNN mixture block of scTCA, although it may suffer higher computational cost. Second, scTCA does not perform well in accurately recovering the true read count profiles



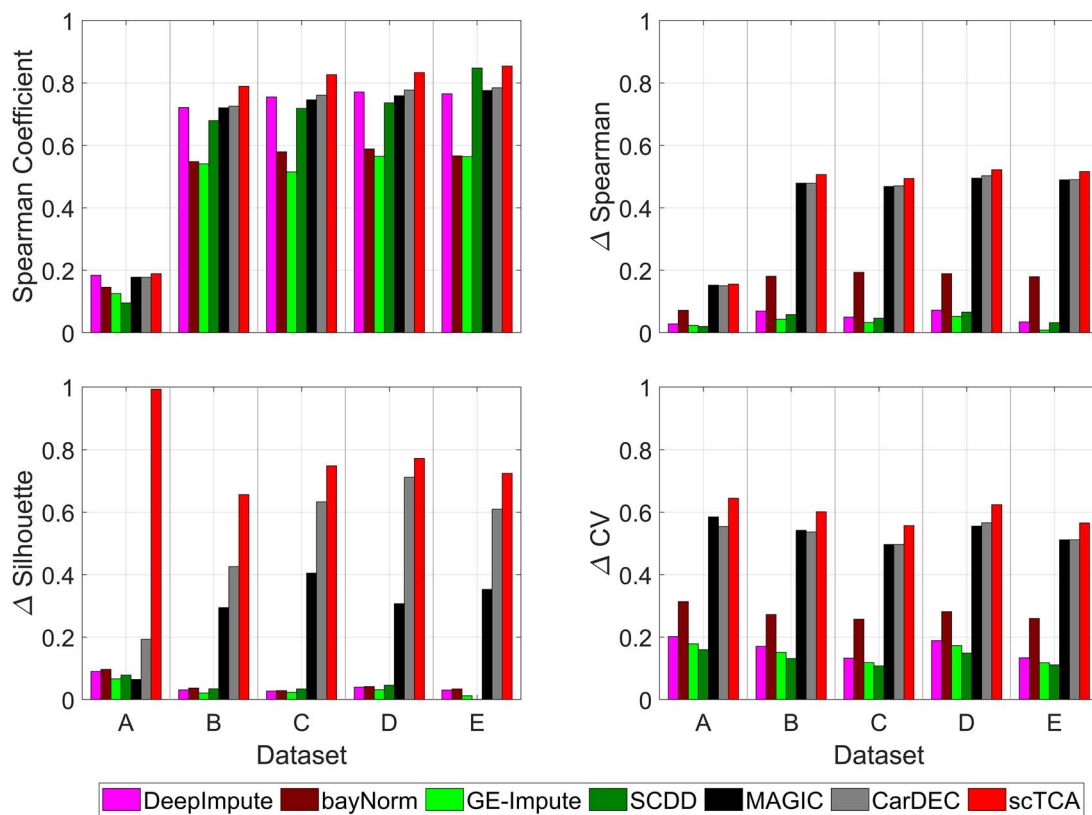


Figure 6. Performance comparison results on five real breast cancer datasets A-E. scImpute, scMultiGAN, and TsImpute are excluded from performance comparison as they encounter runtime errors.

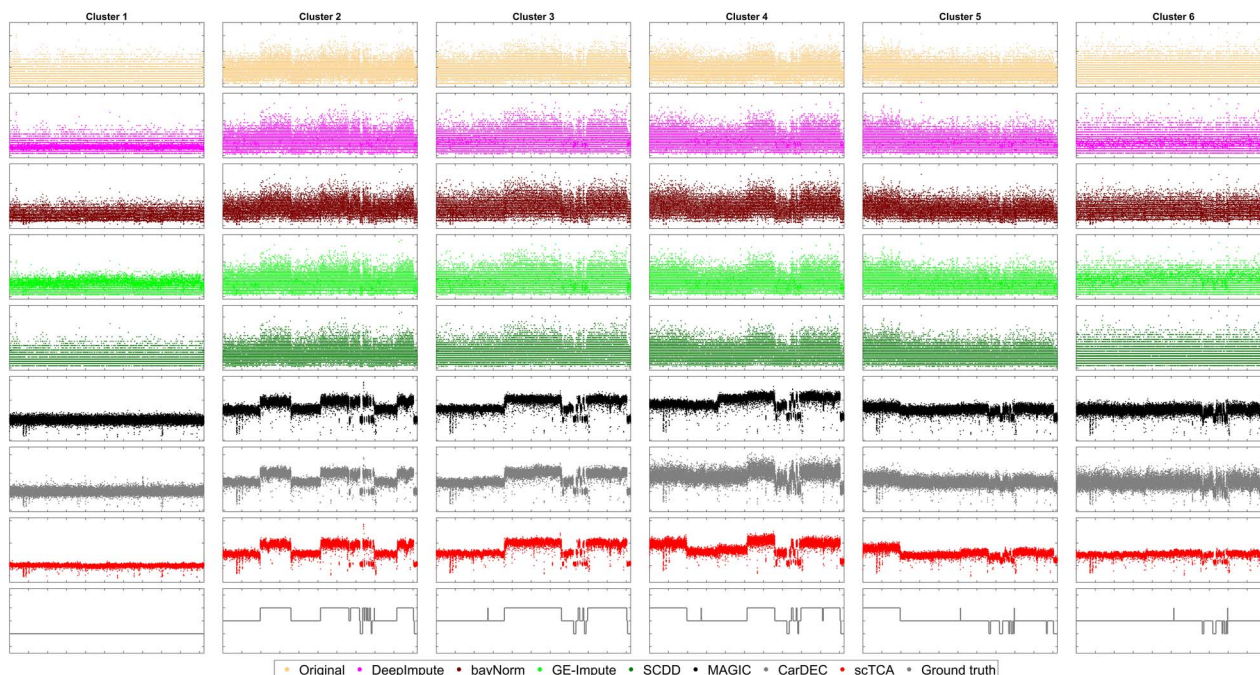


Figure 7. Comparison between the reconstructed read counts and GT copy number profiles on real breast cancer dataset B.

for low-prevalence copy number states (e.g. the first segment of cluster 5 in [Supplementary Fig 8](#)). Data augmentation approaches based on generative models, such as GANs and diffusion autoencoders [44, 45], can be employed to generate synthetic cells for minor clones, thus alleviating the class imbalance and improving

the quality of data reconstruction. We plan to investigate these directions in the future works.

In summary, scTCA can be used to rebuild high-quality single-cell read counts, and simultaneously learn discriminative latent representations for the cells, assisting in downstream analysis

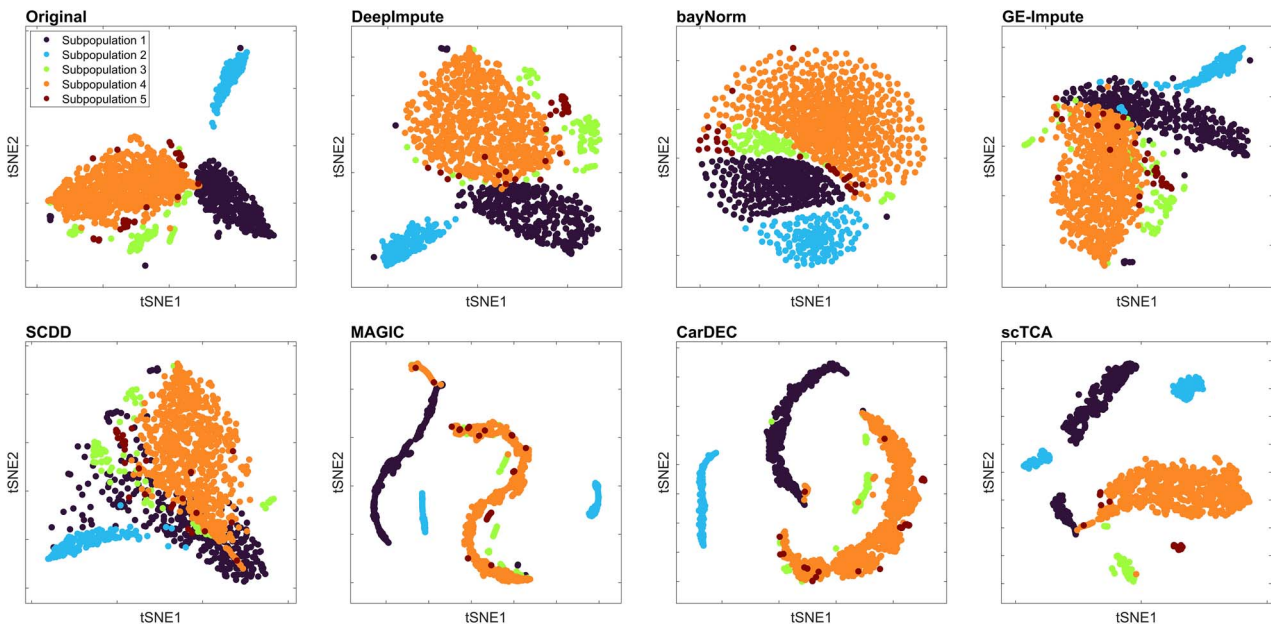


Figure 8. Visualization of the imputed and denoised read counts using tSNE on real breast cancer dataset E. The dimension of read counts data is first reduced to 50 with the principal component analysis, and then reduced to 2 with tSNE. The real dataset E contains five cell subpopulations. The  $\Delta$ Silhouette scores of the methods are 0.031, 0.034, 0.013, 0.001, 0.353 (MAGIC), 0.61 (CarDEC), and 0.724 (scTCA), respectively.

including detection of both arm-level and focal CNAs as well as identification of tumor clones.

#### Key Points

- We develop a novel method scTCA based on a framework with Transformer-CNN hybrid architectures, to impute and denoise single-cell read counts.
- A two-stage attention module, including a stepwise self-attention layer and a window Transformer, is proposed to extract non-local features from the read counts.
- scTCA outperforms the state-of-the-arts on both synthetic and real datasets, assisting in downstream analysis of both arm-level and focal CNAs.

## Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Conflict of interest: None declared.

## Funding

This work has been supported in part by the Natural Science Foundation of Ningxia Province (2023AAC05006) and Key Research and Development Program of Ningxia (2023BEG02009).

## Data availability

BAM files of the 10X Genomics breast cancer datasets are available at <https://www.10xgenomics.com/datasets>, and corresponding copy number profiles and cell labels are obtained from <https://github.com/raphael-group/chisel-data>.

## Software availability

The scTCA method is available at <https://github.com/zhyu-lab/scTCA>.

## Authors' contributions

Z.Y. conceived the study. Z.Y. implemented the method and drafted the manuscript. F.L. and Y.L. analyzed the data. All authors read and approved the final manuscript for publication.

## Competing interests

No competing interest is declared.

## Ethics approval and consent to participate

Not applicable.

## References

1. Evrony GD, Hinch AG, Luo C. Applications of single-cell DNA sequencing. *Annu Rev Genomics Hum Genet* 2021;**22**:171–97. <https://doi.org/10.1146/annurev-genom-111320-090436>.
2. Beroukhir R, Mermel CH, Porter D. et al. The landscape of somatic copy-number alteration across human cancers. *Nature* 2010;**463**:899–905. <https://doi.org/10.1038/nature08822>.
3. Krijgsman O, Carvalho B, Meijer GA. et al. Focal chromosomal copy number aberrations in cancer—needles in a genome haystack. *Biochim Biophys Acta-Mol Cell Res* 2014;**1843**:2698–704. <https://doi.org/10.1016/j.bbamcr.2014.08.001>.
4. Mallory XF, Edrisi M, Navin N. et al. Methods for copy number aberration detection from single-cell DNA-sequencing data. *Genome Biol* 2020;**21**:208. <https://doi.org/10.1186/s13059-020-02119-8>.

5. Zhenhua Y, Liu F, Shi F, Du F. rcCAE: a convolutional autoencoder method for detecting intra-tumor heterogeneity and single-cell copy number alterations. *Brief Bioinform* 2023;**24**:bbad108.
6. Deshpande V, Luebeck J, Nguyen N-PD. et al. Exploring the landscape of focal amplifications in cancer using amplicon-architect. *Nat Commun* 2019;**10**:392. <https://doi.org/10.1038/s41467-018-08200-y>.
7. Steele CD, Ammal Abbasi SM, Islam A. et al. Signatures of copy number alterations in human cancer. *Nature* 2022;**606**:984–91. <https://doi.org/10.1038/s41586-022-04738-6>.
8. Hou W, Ji Z, Ji H. et al. A systematic evaluation of single-cell RNA-sequencing imputation methods. *Genome Biol* 2020;**21**:218–30. <https://doi.org/10.1186/s13059-020-02132-x>.
9. Patruno L, Maspero D, Craighero F. et al. A review of computational strategies for denoising and imputation of single-cell transcriptomic data. *Brief Bioinform* 2021;**22**:bbaa222. <https://doi.org/10.1093/bib/bbaa222>.
10. Dai C, Jiang Y, Yin C. et al. scIMC: a platform for benchmarking comparison and visualization analysis of scRNA-seq data imputation methods. *Nucleic Acids Res* 2022;**50**:4877–99. <https://doi.org/10.1093/nar/gkac317>.
11. Cheng Y, Ma X, Yuan L. et al. Evaluating imputation methods for single-cell RNA-seq data. *BMC Bioinform* 2023;**24**:302. <https://doi.org/10.1186/s12859-023-05417-7>.
12. Haque A, Engel J, Teichmann SA. et al. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Med* 2017;**9**:1–12. <https://doi.org/10.1186/s13073-017-0467-4>.
13. Yungang X, Zhang Z, You L. et al. scIGANs: single-cell RNA-seq imputation using generative adversarial networks. *Nucleic Acids Res* 2020;**48**:e85–5.
14. Wang J, Ma A, Chang Y. et al. scGNN is a novel graph neural network framework for single-cell RNA-seq analyses. *Nat Commun* 2021;**12**:1882. <https://doi.org/10.1038/s41467-021-22197-x>.
15. Tjåmberg A, Mahmood O, Jackson CA. et al. Optimal tuning of weighted KNN-and diffusion-based methods for denoising single cell genomics data. *PLoS Comput Biol* 2021;**17**:e1008569. <https://doi.org/10.1371/journal.pcbi.1008569>.
16. Li H, Brouwer CR, Luo W. A universal deep neural network for in-depth cleaning of single-cell RNA-seq data. *Nature. Communications* 2022;**13**:1901. <https://doi.org/10.1038/s41467-022-29576-y>.
17. Jin K, Li B, Yan H. et al. Imputing dropouts for single-cell RNA sequencing based on multi-objective optimization. *Bioinformatics* 2022;**38**:3222–30. <https://doi.org/10.1093/bioinformatics/btac300>.
18. Xiaobin W, Zhou Y. Ge-impute: Graph embedding-based imputation for single-cell RNA-seq data. *Brief Bioinform* 2022;**23**:bbac313.
19. Liu J, Pan Y, Ruan Z. et al. SCDD: a novel single-cell RNA-seq imputation method with diffusion and denoising. *Brief Bioinform* 2022;**23**:bbac398. <https://doi.org/10.1093/bib/bbac398>.
20. Huang Z, Wang J, Xudong L. et al. scGGAN: single-cell RNA-seq imputation by graph-based generative adversarial network. *Brief Bioinform* 2023;**24**:bbad040. <https://doi.org/10.1093/bib/bbad040>.
21. Wang T, Zhao H, Yungang X. et al. scMultiGAN: cell-specific imputation for single-cell transcriptomes with multiple deep generative adversarial networks. *Brief Bioinform* 2023;**24**:bbad384. <https://doi.org/10.1093/bib/bbad384>.
22. Zheng W, Min W, Wang S. Tsimpute: an accurate two-step imputation method for single-cell RNA-seq data. *Bioinformatics* 2023;**39**:btad731. <https://doi.org/10.1093/bioinformatics/btad731>.
23. Hausmann F, Ergen C, Khatri R. et al. DISCERN: deep single-cell expression reconstruction for improved cell clustering and cell subtype and state detection. *Genome Biol* 2023;**24**:212. <https://doi.org/10.1186/s13059-023-03049-x>.
24. Weimiao W, Liu Y, Dai Q. et al. G2s3: a gene graph-based imputation method for single-cell RNA sequencing data. *PLoS Comput Biol* 2021;**17**:e1009029.
25. Liu Q, Luo X, Li J. et al. scESI: evolutionary sparse imputation for single-cell transcriptomes from nearest neighbor cells. *Brief Bioinform* 2022;**23**:bbac144. <https://doi.org/10.1093/bib/bbac144>.
26. Arisdakessian C, Poirion O, Yunits B. et al. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome Biol* 2019;**20**:1–14.
27. Eraslan G, Simon LM, Mircea M. et al. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat Commun* 2019;**10**:390. <https://doi.org/10.1038/s41467-018-07931-2>.
28. Zhang L, Zhang S. Imputing single-cell RNA-seq data by considering cell heterogeneity and prior expression of dropouts. *J Mol Cell Biol* 2021;**13**:29–40. <https://doi.org/10.1093/jmcb/mjaa052>.
29. Pan X, Li Z, Qin S. et al. scLRTC: imputation for single-cell RNA-seq data via low-rank tensor completion. *BMC Genom* 2021;**22**:1–19.
30. Tang W, Bertaux F, Thomas P. et al. Baynorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics* 2020;**36**:1174–81. <https://doi.org/10.1093/bioinformatics/btz726>.
31. Fleming SJ, Chaffin MD, Arduini A. et al. Unsupervised removal of systematic background noise from droplet-based single-cell experiments using cellbender. *Nat Methods* 2023;**20**:1323–35. <https://doi.org/10.1038/s41592-023-01943-7>.
32. Liu F, Shi F, Fang D. et al. CoT: a transformer-based method for inferring tumor clonal copy number substructure from scDNA-seq data. *Brief Bioinform* 2024;**25**:bbae187. <https://doi.org/10.1093/bib/bbae187>.
33. Xie Y, Zhang J, Shen C. et al. CoTr: efficiently bridging CNN and transformer for 3D medical image segmentation. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III* 24, pp. 171–80. Berlin, German: Springer, 2021.
34. Liu J, Sun H, Katto J. Learned image compression with mixed transformer-CNN architectures. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14388–97. Piscataway, NJ: IEEE, 2023.
35. Zhang N, Nex F, Vosselman G. et al. Lite-mono: a lightweight CNN and transformer architecture for self-supervised monocular depth estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18537–46. Piscataway, NJ: IEEE, 2023.
36. Liu Z, Lin Y, Cao Y. et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–22. Piscataway, NJ: IEEE, 2021.
37. Dosovitskiy A, Beyer L, Kolesnikov A. et al. An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* 2020. <https://doi.org/10.48550/arXiv.2010.11929>.
38. Li WV, Li JJ. An accurate and robust imputation method scimpute for single-cell RNA-seq data. *Nat Commun* 2018;**9**:997. <https://doi.org/10.1038/s41467-018-03405-7>.
39. Van Dijk D, Sharma R, Nainys J. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* 2018;**174**:716–729.e27. <https://doi.org/10.1016/j.cell.2018.05.061>.

40. Lakkis J, Wang D, Zhang Y. et al. A joint deep learning model enables simultaneous batch effect correction, denoising, and clustering in single-cell transcriptomics. *Genome Res* 2021;**31**:1753–66. <https://doi.org/10.1101/gr.271874.120>.
41. Zhenhua Y, Fang D, Sun X. et al. SCSsim: an integrated tool for simulating single-cell genome sequencing data. *Bioinformatics* 2020;**36**:1281–2.
42. Mallory XF, Nakhleh L. SimSCSnTree: a simulator of single-cell DNA sequencing data. *Bioinformatics* 2022;**38**:2912–4. <https://doi.org/10.1093/bioinformatics/btac169>.
43. Zaccaria S, Raphael BJ. Characterizing allele- and haplotype-specific copy numbers in single cells with chisel. *Nat Biotechnol* 2021;**39**:207–14. <https://doi.org/10.1038/s41587-020-0661-6>.
44. Preechakul K, Chatthee N, Wizadwongsa S. et al. Diffusion autoencoders: toward a meaningful and decodable representation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–29. Piscataway, NJ: IEEE, 2022.
45. Wei C, Mangalam K, Huang P-Y. et al. Diffusion models as masked autoencoders. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16284–94. Piscataway, NJ: IEEE, 2023.