# Single-Cell RNA-Seq Debiased Clustering via Batch Effect Disentanglement

Yunfan Li, Yijie Lin, Peng Hu, Dezhong Peng, Han Luo, and Xi Peng, *Member, IEEE*

***Abstract*— A variety of single-cell RNA-seq (scRNA-seq) clustering methods has achieved great success in discovering cellular phenotypes. However, it remains challenging when the data confounds with batch effects brought by different experimental conditions or technologies. Namely, the data partitions would be biased toward these nonbiological factors. Meanwhile, the batch differences are not always much smaller than true biological variations, hindering the cooperation of batch integration and clustering methods. To overcome this challenge, we propose single-cell RNA-seq debiased clustering (SCDC), an end-to-end clustering method that is debiased toward batch effects by disentangling the biological and nonbiological information from scRNA-seq data during data partitioning. In six analyses, SCDC qualitatively and quantitatively outperforms both the state-of-the-art clustering and batch integration methods in handling scRNA-seq data with batch effects. Furthermore, SCDC clusters data with a linearly increasing running time with respect to cell numbers and a fixed graphics processing unit (GPU) memory consumption, making it scalable to large datasets. The code will be released on Github.**

***Index Terms*— Batch integration, clustering, single-cell RNA analysis.**

## I. Introduction

CLUSTERING analysis plays an important role in discovering and defining cell types based on the transcriptome [1], which aims to group cells according to their similarities without accessing ground truth cell types. To handle high-dimensional single-cell RNA-seq (scRNA-seq) data, a classic pipeline is first using principal component analysis (PCA)-like methods to reduce the dimension of the data, followed by clustering methods such as $k$-means [2], SC3 [3], Mpath [4], and single-cell interpretation via multikernel learning (SIMLR) [5]. Motivated by the recent success of deep learning in biological applications such as cell-type detection [6], [7], signal and state estimation [8], [9], drug repurposing [10], molecular generation [11], and protein complexes detection [12], some deep clustering methods have been proposed recently and shown promising results in scRNA-seq clustering, such as ItClust [13], scDEC [14], deep embedded scRNA-seq clustering (DESC) [15], and scDeepCluster [16] with its semi-supervised variant scDCC [17]. However, despite the true biological difference, variances in single-cell transcriptomic data also come from nonbiological factors such as sequencing techniques and handling laboratories, which are so-called batch effects. Intuitively, data from different technical processing batches would have distinct patterns, which interfere or even overwhelm the true biological variances across different cell types as illustrated in Fig. 1(a). In this case, the standard clustering methods might partition data based on batch effects instead of biological information, thus influencing the subsequent analyses.

To alleviate the influence of batch effects, several task-agnostic batch integration studies have been conducted. Among them, a common solution is utilizing mutual nearest neighbors (MNNs) to capture and correct batch variations, as adopted in MNN correct [18], batch balanced K nearest neighbours (BBKNN) [19], Scanorama [20], scMerge [21], and Seurat [22]. Another solution is modeling scRNA-seq data with some probability distributions to estimate the batch variances, as proposed in ComBat [23] and zero-inflated negative binomial (ZINB)-WaVE [24]. In a coarse-to-fine fashion, linked inference of genomic experimental relationships (LIGER) [25] and Harmony [26] gradually remove batch effects in an iterative fashion. Inspired by the powerful modeling capacity of neural networks, some deep learning methods have been proposed recently, such as maximum mean discrepancy residual neural networks (MMD-ResNet) [27], single-cell variational inference (scVI) [28], scGen [29], and batch effect removal using deep autoencoders (BERMUDA) [30].

Although having achieved promising results, these batch integration methods suffer from two limitations. On the one hand, most of the existing methods model batch effects based on the neighborhood structure, which heavily relies on the assumption that batch differences are much smaller than true biological variations [18]. Once the assumption is violated, it would be daunting to achieve desirable results. In fact, we find that the assumption does not always hold in practice as shown in Fig. 1(a) and (b). On the other hand, almost all existing batch integration methods are clustering-agnostic and solely designed for batch effect removal, which might mistakenly deplete or enrich certain cell types [15]. Iterative methods [25], [26] partly solve this problem, but errors would accumulate during the alternation between batch correction and clustering.

The above limitations arouse the need for a new scRNA-seq data-processing paradigm which could: 1) remove batch effects without the prior assumption on batch differences and

Manuscript received 5 November 2022; revised 22 February 2023; accepted 12 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U21B2040, Grant 62176171, Grant U19A2078, Grant 82103031, and Grant 82272933; and in part by the Sichuan Science and Technology Program under Grant 2023YFS0098. *(Corresponding author: Xi Peng.)*

Yunfan Li, Yijie Lin, Peng Hu, Dezhong Peng, and Xi Peng are with the School of Computer Science, Sichuan University, Chengdu 610000, China (e-mail: pengx.gm@gmail.com).

Han Luo is with West China Hospital, Sichuan University, Chengdu 610000, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3260003.

Digital Object Identifier 10.1109/TNNLS.2023.3260003

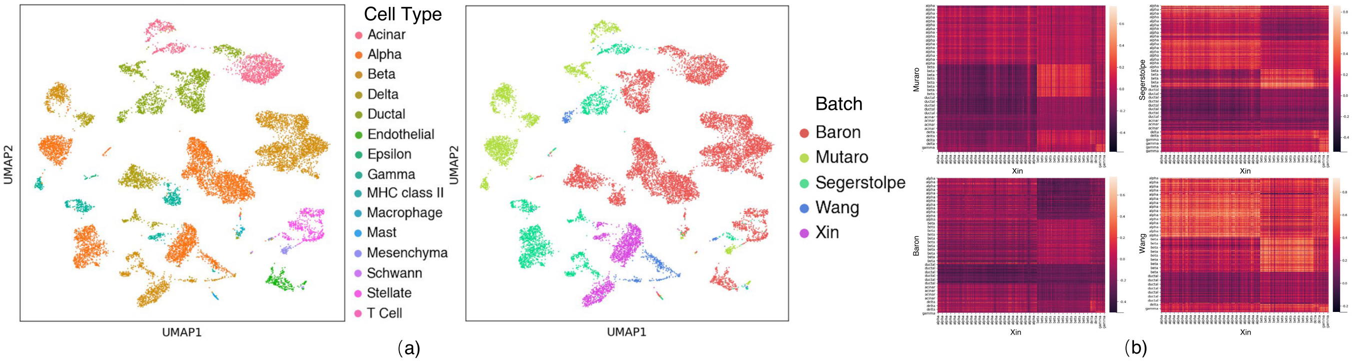


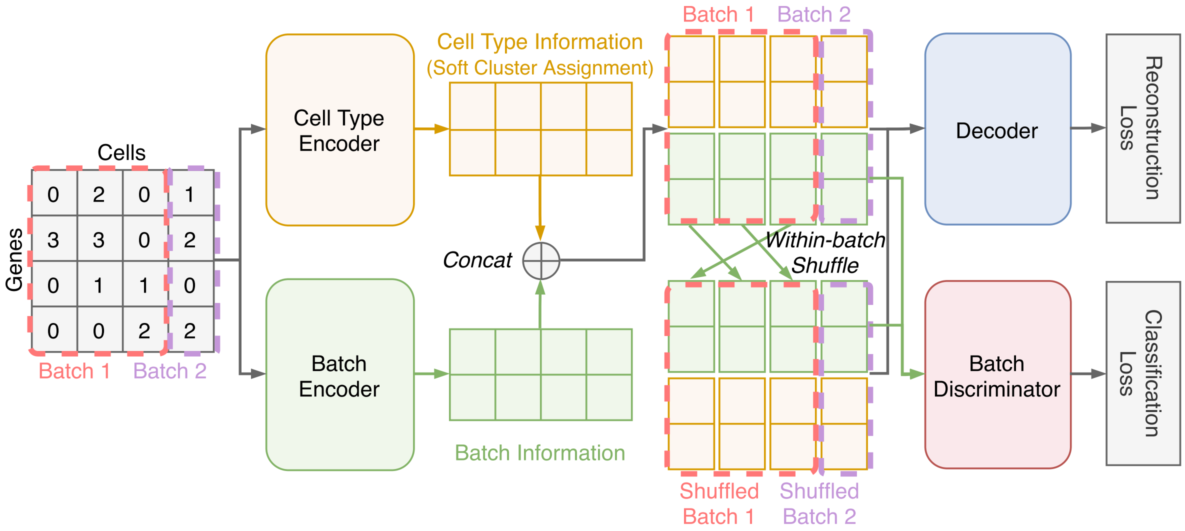
Fig. 1. Illustration of batch effects. (a) UMAP visualization of scRNA-seq data from the human pancreas dataset. Samples are colored according to the intrinsic cell types and sequence batches in the left and right figures, respectively. As can be seen, nonbiological batch effects interfere or even overwhelm the biological variances, e.g., Alpha cells from different batches forms four distinct clusters, and their gaps are even larger than the gap between Alpha and Gamma cells from the same batch. In this case, clustering methods are likely to group samples based on sequence batches instead of cell types, which influences the subsequent analyses. (b) Cosine similarity between cells from batch 5 (GSE81608) and the other four batches (GSE84133, E-MATB-5061, GSE85241, GSE83139) on the human pancreas dataset. Alpha, beta, delta, and gamma Cells are shared across these batches.

biological variations and 2) incorporate batch integration with clustering to bootstrap each other. To this end, we present SCDC, a scRNA-seq debiased clustering method that disentangles biological and nonbiological variations from scRNA-seq data, resulting in a data partition robust to batch effects. The ideas behind SCDC are twofold. First, it is possible to disentangle the biological information for clustering since scRNA-seq data is a joint manifestation of biological variances and batch effects. Notably, SCDC does not make any implicit or explicit assumption on batch differences and biological variations like existing methods, thus enjoying higher applicability. Second, it is feasible to perform disentanglement by utilizing the within-batch invariance, i.e., though varying across batches, batch effects are similar across cells from the same batch (to distinguish the concept of sequence batch here with the data batch in deep learning, we denote the latter as mini-batch in this article). Notably, such an idea holds in most cases since a batch of data is sequenced using the same protocol under similar experimental conditions. Based on the above ideas, we design an end-to-end framework-dubbed SCDC which simultaneously achieves clustering and batch effect removal, overcoming the aforementioned limitations of existing methods. In brief, two encoders are used to extract cell type and batch information from the raw scRNA-seq data. To achieve the disentanglement, SCDC: 1) forces the biological information to be compact by directly predicting cluster assignments and 2) randomly shuffles batch information within each batch and reconstructs each cell with its own cell-type information and the shuffled batch information. Once the batch effects are disentangled, clustering could be correctly achieved based on the biological variations (i.e., cell types). Different from most existing methods that conduct batch effect removal and clustering separately, our SCDC directly outputs the cluster assignments given the batch-effected scRNA-seq data. Such an end-to-end paradigm overcomes the clustering-agnostic and error accumulation problem encountered by existing methods, leading to superior clustering performance for batch-effected data. Another advantage of SCDC is that it takes linearly increasing running time and a constant graphics processing unit (GPU) memory consumption, enjoying high scalability to large-scale data. The main contributions of this work are as follows.

1) This is one of the few studies on clustering scRNA-seq data with batch effects. Different from existing methods that take the assumption of batch differences and biological variations, our SCDC builds upon a more practical observation that batch effects are similar across cells from the same batch, thus enjoying higher applicability.
2) We provide a novel learning paradigm to handle batch effects in scRNA-seq by disentangling biological and nonbiological variations. To achieve the disentanglement, we design a simple but effective autoencoder-like framework which could simultaneously perform batch effect removal and clustering in an end-to-end manner.
3) The proposed SCDC is evaluated on six benchmarks compared with both stage-of-the-art clustering and batch integration methods. Extensive quantitative and qualitative experiments demonstrate the superiority, robustness, and scalability of SCDC.

## II. Related Work

Unsupervised clustering aims to group samples into several clusters without accessing the ground-truth labels [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], which has been widely used in a variety of tasks including scRNA-seq analysis for cellular phenotypes discovery. To handle high-dimensional scRNA-seq data, early works first reduce the dimension of the data, and then apply classic clustering methods. For example, SC3 [3] conducts spectral clustering at different resolutions and computes a consensus matrix to integrate clustering results. SIMLR [5] uses multiple kernels to measure the cell similarities and applies $k$-means [2] to achieve clustering. Motivated by the success of deep clustering [37], [45], [46], [47], [48], [49], some deep models have been developed recently and shown promising results. For example, DESC [15] optimizes a deep-embedded clustering objective in a self-training manner. scDeepCluster [16] explicitly models the scRNA-seq data with the ZINB distribution



| | | | |
|---|---|---|---|
| 0 | 2 | 0 | 1 |
| 3 | 3 | 0 | 2 |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 2 | 2 |

Fig. 2. SCDC framework. SCDC is an autoencoder-like network composed of: 1) a cell-type encoder which extracts biological (i.e., cell type) information; 2) a batch encoder which captures batch information; 3) a batch discriminator which improves the discrimination of the batch information; and 4) a shared decoder that reconstructs the raw input data using both the cell-type and batch information.

and simultaneously achieves feature representation and clustering. scTAG [50] learns cell–cell topology representations and identifies cell clusters based on a deep graph convolutional network. However, with the growth of scRNA-seq techniques, single-cell transcriptomic data are often confounded with nonbiological batch effects. In this case, the standard clustering methods might cluster data based on batch effects instead of biological information, thus influencing subsequent analyses.

To alleviate the influence of batch effects, several task-agnostic batch integration methods have been proposed. A common solution is utilizing MNNs to capture and correct batch variations. For example, MNN correct [18] selects cells with similar neighbors to model batch effects for subsequent correction. Scanorama [20] merges batches according to the percentage of matching cells in the batch. Seurat [22] searches MNN in the subspace learned by cross-batch CCA. In addition to the neighborhood information, scMerge [21] further identifies stably expressed genes to estimate batch effects. Another solution is modeling scRNA-seq data with Gaussian [23] or ZINB [24] probability distributions to estimate the batch variances. In a coarse-to-fine fashion, LIGER [25] and Harmony [26] gradually remove batch effects in an iterative manner. Considering the powerful modeling capacity of neural networks, some deep learning methods have also been proposed recently. For example, MMD-ResNet [27] adopts a residual network to learn the mapping from one batch to another. scGen [29] models the data with a variational autoencoder and transfers the data distribution across batches. scVI [28] aggregates information across similar cells and genes to approximate the distributions for batch integration. BERMUDA [30] utilizes the similarities between cell clusters to align cells among different batches.

Though achieving promising results, existing integration methods heavily rely on the assumption that the batch differences are much smaller than the true biological variations. However, as shown in Fig. 1(b), such an assumption not always holds in practice, and inferior results would be achieved once the assumption is violated. Besides, few efforts have been made on clustering scRNA-seq data with batch effects. As a representative, DESC [15] shows that batch effects could be gradually removed during the clustering process, but it still relies on the above assumption. Different from these existing works, the proposed SCDC deals with batch effects from a new perspective, namely, through disentangling the nonbiological variations from the scRNA-seq data during data partitioning. In other words, the clustering results are debiased toward batch effects and correctly based on the biological variations (i.e., cell types). Note that the proposed disentanglement strategy does not rely on the aforementioned assumption, but builds on a more practical observation that batch effects are similar across cells from the same batch, which provides a more general solution to cluster scRNA-seq data with batch effects.

## III. Method

In this section, we introduce SCDC, an end-to-end clustering model which aims at capturing the biological information and predicting the cluster assignments of the input scRNA-seq data. As illustrated in Fig. 2, SCDC extracts the cell type and batch information through $f_C$ and $f_B$, respectively. To disentangle cell type and batch information, SCDC randomly shuffles batch information within each batch. Then, each cell is reconstructed with its own cell-type information and the shuffled batch information through $g$. In addition, SCDC utilizes a batch discriminator $h_B$ to further enhance the batch information extraction by predicting the batch index. After training, clustering could be directly achieved by the argmax operation on soft cluster assignments predicted by $f_C$. Below, we describe each component of SCDC in detail.

### A. ZINB Reconstruction for Feature Extraction

To extract features from discrete and sparse scRNA-seq data whose variance is larger than the mean, SCDC fits them with the ZINB distribution. Following [51], one could derive that the zero inflation handles the sparseness, and the negative binomial (NB) distribution handles the discreteness and large variances.

*Theorem 1:* Discrete scRNA-seq data with a variance larger than the mean could be approximated by the NB distribution.

*Proof:* The probability mass function of the NB distribution is

$$\mathrm{NB}(x \mid r, p) \equiv \Pr(X = x) = \frac{\Gamma(x+r)}{x!\Gamma(r)} p^{r}(1-p)^{x} \tag{1}$$

where $r \in (0, \infty)$ and $p \in (0, 1)$ correspond to the number and probability of successes in a sequence of i.i.d. Bernoulli processes, respectively. The mean $E(X)$ of the NB distribution could be derived as follows:

$$\begin{aligned} E(X) &= \sum_{x=0}^{\infty} x \frac{\Gamma(x+r)}{x!\Gamma(r)} p^{r}(1-p)^{x} \\ &= \frac{r(1-p)}{p} \sum_{x=1}^{\infty} \frac{\Gamma(x-1+r+1)}{(x-1)!\Gamma(r+1)} p^{r+1}(1-p)^{x-1} \\ &= r\frac{(1-p)}{p}. \end{aligned} \tag{2}$$

Similarly, one could derive the variance $D(X)$ of the NB distribution as follows:

$$\begin{aligned} D(X) &= E\left(X^{2}\right) - E(X)^{2} \\ &= r\frac{(1-p)+(1-p)^{2}}{p^{2}} - r\frac{(1-p)^{2}}{p^{2}} \\ &= r\frac{(1-p)}{p^{2}} > E(X). \end{aligned} \tag{3}$$

Hence, the NB distribution could approximate the scRNA-seq data that has a larger variance than the mean. □

*Theorem 2:* The sparseness in scRNA-seq data could be modeled by the ZINB distribution.

*Proof:* The probability mass function of the ZINB distribution is

$$\mathrm{ZINB}(x \mid \pi, r, p) = \pi\delta_{0}(x) + (1-\pi)\mathrm{NB}(x \mid r, p) \tag{4}$$

where $\delta_0(\cdot)$ is the Dirac delta function and $\pi$ is the inflation parameter, which explicitly strengthens the probability of $x$ being zero as follows:

$$\begin{aligned} \Delta \Pr(X = 0) &= \mathrm{ZINB}(x = 0) - \mathrm{NB}(x = 0) \\ &= \pi + (1-\pi)p^{r} - p^{r} \\ &= \pi\left(1 - p^{r}\right) > 0. \end{aligned} \tag{5}$$

Hence, the inflation parameter $\pi$ could help model the sparseness in scRNA-seq data. □

To fit the raw scRNA-seq count data $X = [x_1, \ldots, x_N]$ with ZINB distribution, we propose to minimize its negative log-likelihood with the following reconstruction loss, namely:

$$\begin{aligned} L_{\mathrm{ZINB}} &= \frac{1}{N}\sum_{i=1}^{N} -\log(\mathrm{ZINB}(x_i \mid \pi_i, r_i, \mu_i)) \\ &= \pi_i\delta_0(x_i) + \frac{\Gamma(x_i + r_i)}{x!\Gamma(r_i)}\left(\frac{r_i}{r_i+\mu_i}\right)^{r_i}\left(\frac{\mu_i}{r_i+\mu_i}\right)^{x_i} \end{aligned} \tag{6}$$

where $\pi_i$, $r_i$, and $\mu_i = r(1-p)/p$ correspond to the inflation, dispersion, and mean parameters of the ZINB distribution, respectively. The three parameters $\pi_i$, $r_i$, and $\mu_i$ are estimated by the ZINB autoencoder. Specifically, three independent fully connected layers $W_\pi$, $W_r$, and $W_\mu$ are stacked on the feature map $d_i$ output by the last hidden layer of the decoder to estimate

$$\pi_i = \mathrm{sigmoid}(W_\pi d_i) \tag{7}$$

$$r_i = \exp(W_r d_i) \tag{8}$$

$$\mu_i = \mathrm{diag}(s_i) \times \exp\left(W_\mu d_i\right) \tag{9}$$

where $s_i$ represents the size factor which is computed during the data preprocessing and is not a part of network parameters.

### *B. Disentanglement of Biological Variances and Batch Effects*

To disentangle biological variances and nonbiological batch effects, two modifications are made to the autoencoder besides the ZINB reconstruction. On the one hand, SCDC learns compact biological information by directly predicting cluster assignments. On the other hand, SCDC randomly shuffles batch information within each batch before the reconstruction and adopts a batch discriminator to capture batch information. More details are provided below.

*1) Gumbel-Softmax for Disentangling Biological Variances:* Let $K$ be the target cluster number which is known in prior or manually set, the cell-type encoder $f_C$ projects each data point $x_i$ into a $K$-dimensional vector $c_i = f_C(x_i)$. To represent the cell type of a datum, $c_i$ is expected to be a one-hot vector. However, directly applying the argmax operation would lead to discrete variables which will break the back-propagation of neural networks. As a solution, we use the Gumbel-Softmax estimator [52] to compute the cell-type representation $y_i = [y_{i1}, \ldots, y_{iK}]$ through

$$y_{ik} = \frac{\exp((\log(c_{ik}) + g_{ik})/\tau)}{\sum_{k=1}^{K}\exp((\log(c_{ik}) + g_{ik})/\tau)}, \quad k = 1, \ldots, K \tag{10}$$

where $g_{i1}, \ldots, g_{iK} \sim \mathrm{Gumbel}(0, 1)$ and $\tau$ is a temperature parameter to control the softness. To produce harder cluster representations as the training progresses, we empirically set $\tau = (1 - \hat{k}/K) * 0.67 + 0.33$ to gradually decay $\tau$ from 1.0 to 0.33, where $\hat{k}$ is the number of nonempty clusters currently. In addition, to stabilize the training, we apply the straight-through trick [52], [53] on large and unbalanced datasets (i.e., mouse retina and mouse brain) where some types of cells are absent in a series of successive minibatches.

*2) Within-Batch Shuffling and Batch Discrimination for Disentangling Batch Effects:* Independent of $f_C$, another encoder $f_B$ is used to encapsulate the batch information through $b_i = f_B(x_i)$. By feeding the concatenation of $y_i$ and $b_i$ into the decoder $g$, SCDC obtains $d_i = g(\mathrm{concat}(y_i, b_i))$ which is then used to compute $\pi_i, r_i, \mu_i$ in $L_{\mathrm{ZINB}}$ with (7)–(9). As simply using two independent encoders would fail to disentangle the cell type and batch information, we propose the following within-batch shuffling and batch discrimination strategy.

Since cells from the same batch are sequenced under similar conditions, it is reasonable to assume that the batch information is similar across within-batch cells. Therefore, we propose utilizing such an invariance to disentangle batch effects by shuffling the batch representation within each batch. Specifically, let $B_i$ be the batch index of cell $i$

$$\tilde{d}_i = g\big(\text{concat}\big(y_i, \tilde{b}_i\big)\big), \quad \tilde{b}_i = b_{\text{rand}(i)} \tag{11}$$

where rand() randomly maps the given index $i$ to another index $j$ that satisfies $B_i = B_j$. After the within-batch shuffling, it is expected that $\tilde{d}_i$ could still approximate the ZINB distribution of cell $i$, which leads to the shuffled ZINB loss, i.e.,

$$\tilde{L}_{\text{ZINB}} = \frac{1}{N}\sum_{i=1}^{N} -\log\big(\text{ZINB}\big(x_i \mid \tilde{\pi}_i, \tilde{r}_i, \tilde{\mu}_i\big)\big) \tag{12}$$

where $\tilde{\pi}_i, \tilde{r}_i, \tilde{\mu}_i$ are estimated by $\tilde{b}_i$ via (7)–(9).

*Theorem 3:* The Gumbel-Softmax operation in $f_C$ and within-batch shuffling strategy in $f_B$ could disentangle batch effects from cluster assignments.

*Proof:* Without loss of generality, we take cell $x_i$ as an example. Define $H(d_i^{\text{ms}})$ be the minimal sufficient information entropy to ensure the log-likelihood of ZINB distribution is above a certain threshold $\epsilon$. In other words, $\log(\text{ZINB}(x_i)) \leq \epsilon, \forall H(d_i)$ s.t. $H(d_i^{\text{ms}}|d_i) \geq 0$. The Gumbel-softmax operation with decaying temperature $\tau$ intrinsically encourage $f_C$ to capture information $H(y_i)$ as compact as possible, that could cover $H(d_i^{\text{ms}})$ together with the shuffled information $H(b_j)(j \neq i)$ captured by $f_B$. From the view of information theory, the optimization objective of SCDC could be formulated as follows:

$$\min H(y_i) \quad \text{s.t. } I\big(y_i, b_j; d_i^{\text{ms}}\big) = H\big(d_i^{\text{ms}}\big). \tag{13}$$

Following the chain rule, we have:

$$I\big(y_i, b_j; d_i^{\text{ms}}\big) = I\big(y_i; d_i^{\text{ms}}\big) + I\big(b_j; d_i^{\text{ms}} \mid y_i\big). \tag{14}$$

Further, since $H(b_j) = H(f_B(x_j)) \leq H(x_j)$ based on the Markov chain and the data processing inequality [54], combining (13) and (14), one could derive that

$$\begin{aligned} H(y_i) &\geq I\big(y_i; d_i^{\text{ms}}\big) \\ &= I\big(y_i, b_j; d_i^{\text{ms}}\big) - I\big(b_j; d_i^{\text{ms}} \mid y_i\big) \\ &\geq H\big(d_i^{\text{ms}}\big) - I\big(x_j; d_i^{\text{ms}}\big) \\ &= H\big(d_i^{\text{ms}} \mid x_j\big). \end{aligned} \tag{15}$$

Optimizing (13) leads to compact clustering information that satisfies $H(y_i) = H(d_i^{\text{ms}} \mid x_j)$. In other words, the cluster assignments $y_i$ for cell $x_i$ would not contain batch information shared by other within-batch cells $x_j$. Hence, SCDC could disentangle batch effects from cluster assignments. □

To further improve the ability of $f_B$ in disentangling batch effects, SCDC stacks a batch discriminator $h_B$ on $b_i$ to predict the batch index of cells. Notably, such a prior is usually available in practice. The following cross-entropy loss is adopted to optimize $f_B$ and $h_B$:

$$L_{\text{CE}} = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{\exp(p_i[B_i])}{\sum_{b=1}^{B}\exp(p_i[b])}\right), \quad p_i = h_B(f_B(x_i)). \tag{16}$$

Ablation studies in Table III show the effectiveness of the within-batch shuffling strategy and batch discriminator in achieving the batch effect disentanglement.

TABLE I
DATASETS WITH BATCH EFFECTS USED FOR EVALUATION

| Dataset | Cells | Genes | Groups | Batches |
|---|---|---|---|---|
| Mouse Atlas | 6,954 | 15,006 | 11 | 2 |
| Human Pancreas | 14,767 | 15,558 | 15 | 5 |
| Cell line | 9,531 | 32,738 | 2 | 3 |
| Hematopoietic Stem | 4,649 | 3,467 | 7 | 2 |
| Mouse Retina | 71,638 | 12,333 | 12 | 2 |
| Mouse Brain | 83,323 | 17,745 | 14 | 2 |

### C. End-to-End Training and Clustering

With (6), (12), and (16), the overall loss function of SCDC is defined as follows:

$$L = L_{\text{ZINB}} + \alpha\tilde{L}_{\text{ZINB}} + \beta L_{\text{CE}} \tag{17}$$

where $\alpha, \beta$ are two hyperparameters to weigh the losses. In practice, we find that promising results could be achieved by simply fixing $\alpha = 1,\ \beta = 0.01$ in all experiments and no exhaustive parameter selection is needed. The overall loss in (17) is used to optimize the entire network including $f_C$, $f_B$, $g$, and $h_B$ by stochastic gradient descent [55] in an end-to-end manner.

After the network converges, SCDC could achieve end-to-end clustering by directly predicting the soft clustering assignment $C_i$ for batch effected data with the cell-type encoder $f_C$ through

$$C_{ik} = \frac{\exp(c_{ik})}{\sum_{j=1}^{K}\exp\big(c_{ij}\big)} \tag{18}$$

where $C_{ik}$ denotes the probability of cell $i$ belonging to cluster $k$. To obtain hard cluster assignments, one could simply apply the argmax operation on $C_i$.

## IV. EXPERIMENTS

In this section, the clustering performance of the proposed SCDC is evaluated on six scRNA-seq datasets with various batch effects. Extensive quantitative and qualitative results demonstrate its superiority, robustness, and scalability.

### A. Datasets

Six batch effect datasets are used for evaluation, with a brief description summarized in Table I. Specifically, the mouse cell atlas and hematopoietic stem are sequenced with two different protocols, which are used for examining the robustness against different scRNA-seq technologies. The human pancreas and cell line dataset are a combination of five and three different sources, respectively, which is used for examining the robustness against multiple batches. The mouse retina dataset is sequenced by two unassociated laboratories, which is used to verify the robustness against nonidentical cell types across batches. Finally, the mouse brain dataset is used to evaluate the scalability of the methods.

### B. Implementation Details

The scRNA-seq count data is preprocessed by the scanpy [56] package as follows. First, we normalize each cell

TABLE II

CLUSTERING PERFORMANCE ON SIX BATCH EFFECT BENCHMARKS (MEAN ± STD). THE BEST RESULT IS DENOTED IN **BOLD**

| Dataset | Mouse Atlas | | | Human Pancreas | | | Cell line | | |
|---|---|---|---|---|---|---|---|---|---|
| Metrics | ARI | ACC | BER | ARI | ACC | BER | ARI | ACC | BER |
| ZINB-WaVE | 51.03±2.80 | 62.58±3.06 | 64.72±25.12 | 16.94±0.57 | 34.30±1.58 | 318.65±15.29 | **98.66±0.03** | **99.66±0.01** | 0.09±0.00 |
| DESC | 59.94±4.78 | 68.94±5.02 | 39.30±6.60 | 41.44±1.58 | 52.35±1.34 | 198.86±17.44 | 16.08±0.18 | 70.06±0.11 | 3082.10±12.40 |
| fastMNN | 63.16±2.08 | 70.73±0.78 | 48.20±12.68 | 46.58±2.51 | 54.83±2.41 | 222.99±29.15 | 12.16±0.01 | 67.44±0.00 | 2437.80±0.50 |
| scMerge | 60.01±0.48 | 69.38±0.43 | 47.53±10.07 | 49.46±2.25 | 59.38±1.69 | 197.39±24.24 | 98.08±0.00 | 99.52±0.00 | 0.04±0.00 |
| LIGER | 50.02±3.54 | 62.96±1.98 | 56.77±12.39 | 58.72±4.34 | 67.09±3.54 | 119.95±35.01 | 20.17±0.00 | 72.48±0.00 | 2934.62±0.00 |
| Harmony | 68.40±1.09 | 71.49±1.24 | 28.31±13.09 | 73.91±7.90 | 72.00±5.07 | 65.75±23.69 | 98.58±0.00 | 99.64±0.00 | 0.07±0.00 |
| Seurat3 | 52.55±7.02 | 62.23±3.97 | 56.86±15.27 | 75.51±7.35 | 75.88±5.14 | 78.00±21.97 | 69.75±0.14 | 91.76±0.04 | 735.46±4.72 |
| SCDC (ours) | **72.39±6.88** | **77.91±3.11** | **19.54±9.57** | **90.04±5.22** | **86.98±3.11** | **19.88±4.46** | **98.66±0.21** | **99.66±0.05** | **0.02±0.03** |
| Dataset | Hematopoietic Stem | | | Mouse Retina | | | Mouse Brain | | |
| Metrics | ARI | ACC | BER | ARI | ACC | BER | ARI | ACC | BER |
| ZINB-WaVE | 30.16±0.37 | 43.81±0.31 | 344.21±13.36 | 18.14±0.18 | 40.08±0.27 | 4340.69±110.45 | 27.18±5.23 | 54.66±5.90 | 3343.01±434.12 |
| DESC | 35.34±2.06 | 46.91±2.11 | 352.13±29.32 | 38.92±2.36 | 49.19±3.35 | 4820.44±325.27 | 16.36±1.79 | 40.42±2.44 | 5021.12±704.63 |
| fastMNN | 34.71±0.35 | 47.10±0.23 | 408.44±28.52 | 32.47±0.30 | 44.62±1.30 | 6558.41±292.32 | 17.06±0.01 | 42.29±0.01 | 5377.27±222.02 |
| scMerge | 25.83±3.52 | 45.97±6.25 | **73.92±52.30** | 23.47±0.22 | 33.27±0.66 | 7487.28±441.99 | 12.33±0.71 | 33.94±1.05 | 4636.61±510.55 |
| LIGER | 41.98±3.48 | 61.95±5.97 | 93.42±70.44 | 34.95±7.99 | 57.31±7.94 | 3894.61±249.42 | 28.97±0.19 | 59.44±0.40 | 2209.18±51.90 |
| Harmony | 46.93±1.68 | 57.04±1.36 | 344.18±67.64 | 36.48±0.82 | 50.91±0.81 | 3521.04±631.78 | 36.12±3.54 | 62.26±4.15 | 1856.50±191.96 |
| Seurat3 | 41.43±1.34 | 58.34±1.85 | 193.36±11.80 | 65.70±2.06 | 70.16±3.41 | 3253.23±790.53 | 26.06±2.57 | 52.16±3.95 | 2743.93±307.14 |
| SCDC (ours) | **62.51±2.73** | **69.07±1.45** | 103.37±31.02 | **67.57±6.82** | **75.99±7.14** | **996.55±582.68** | **83.30±15.02** | **90.76±7.07** | **283.98±352.38** |

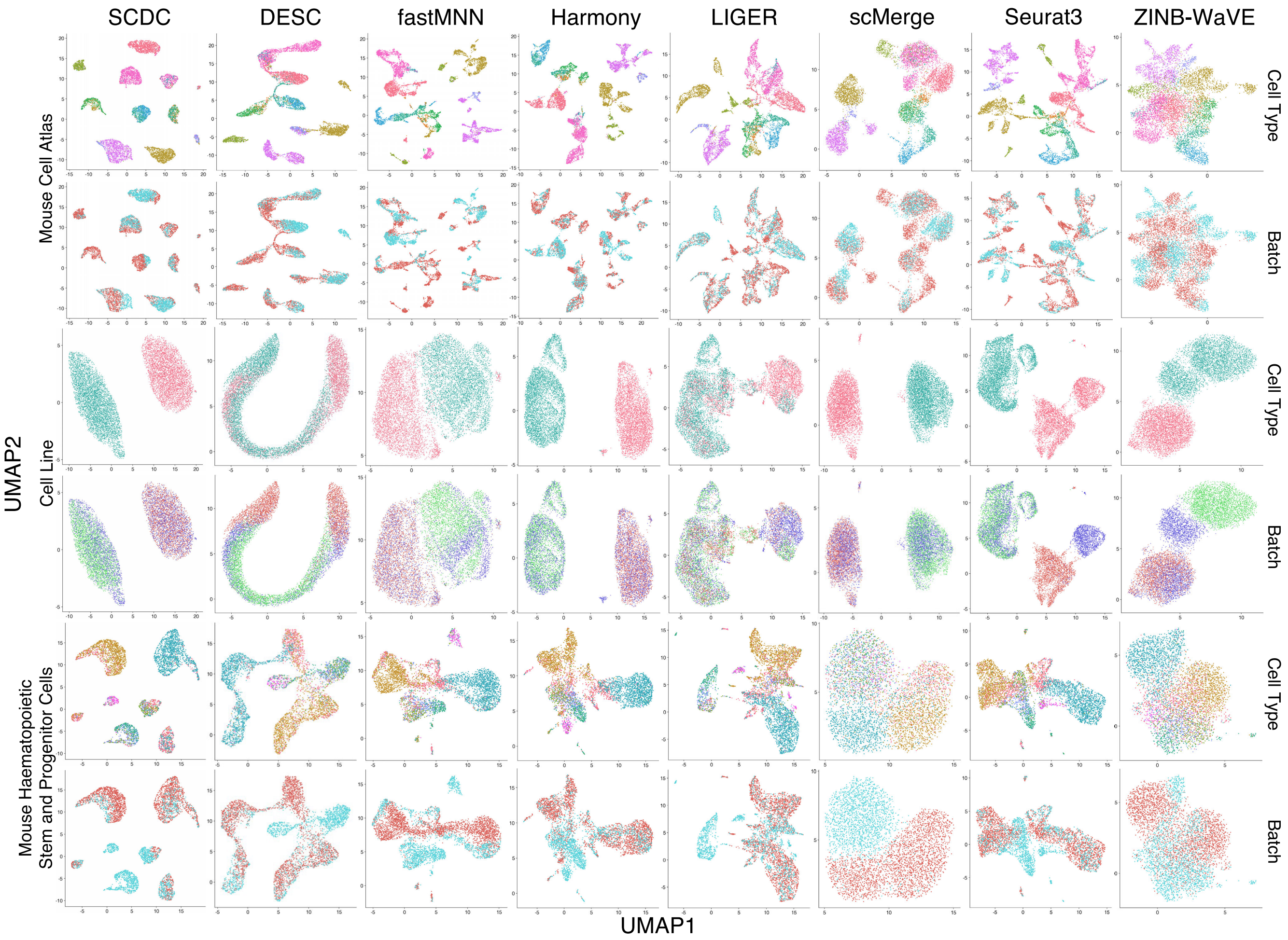


Fig. 3. UMAP plots of SCDC and baselines on mouse cell atlas, cell line, and hematopoietic stem, where cells are colored by cell type and batch in the odd and even rows, respectively.

by dividing its total number of read counts on all genes, and then multiply them by 10 000 to ensure that total counts are the same across cells. After that, we natural log normalize the read counts and then selected highly variable genes (HVGs). Finally, we scale the data to have unit variance and zero mean.

The cell-type encoder $f_C$ is a fully connected network (FCN) with the dimension of M-256-64-32-32-K, where $M$ is the number of HVGs selected in the preprocessing stage and $K$ is the target cluster number. The batch encoder $f_B$ and the batch discriminator $h_B$ are two FCNs with the dimension of M-256-64-32 and 32-32-B, respectively, where $B$ is the batch number. The decoder $g$ is also an FCN with the dimension of (K + 32)-32-64-256. And three parameter estimators $W_\mu$, $W_r$, and $W_\pi$ are of dimension 256-M. When computing $L_{\text{ZINB}}$ and $\tilde{L}_{\text{ZINB}}$, Gaussian noises are added into the input to improve the representability of the extracted features [57]. With a batch size of 256, we train the network for 300 epochs using the Adam optimizer [55] with the default parameters. All experiments are conducted on an Nvidia RTX 2080Ti GPU with CUDA 11.0, on the Ubuntu 20.04 OS.

All baselines are implemented based on their officially released codes or packages. Specifically, for DESC, we use the official desc Python package, v.2.1.1, with $k$-means initialization. For fastMNN, the Seurat [58] preprocessing workflow is adopted to first identify 5000 HVGs, followed by the multiBatchPCA operation from the Scran R package [59], v.1.21.1, resulting in 50 principal components. After that, we use the fastMNN function to integrate the preprocessed data. For Harmony, following the default setting, we select top 20 principal components from the Seurat [58] preprocessed data, and then fed them into the RunHarmony function from the Harmony R package [26], v.0.99.9, with parameters $\theta = 2$, nclust $= 50$, max.iter.cluster $= 100$. For LIGER, we use its preprocessing and integration functions from the liger package [25], v.1.0.0, with recommended parameters $k = 20$, $\lambda = 5$. For scMerge, we run the scMerge function from R package [21], v.1.9.0, on the $\log_2$ normalized data, and project the output into a 20-dimensional PCA space for clustering. For Seurat 3, we follow the default workflow by selecting 2000 HVGs to compute anchors and then integrating the data with the Seurat R package [22], v.3.0.1. For efficient clustering, PCA is performed on the integrated data to reduce the dimension to 20. Note that we also test the recently released Seurat 4 [60] and found it gives similar or slightly worse performance compared with Seurat 3, so here we choose Seurat 3 for comparison. As suggested in the original ZINB-WaVE paper [24], we feed 1000 HVGs to the zinbwave function provided in the zinbwave R package, v.1.15.1.

### C. Evaluation Metrics

Two widely used metrics adjusted rand index (ARI) and accuracy (ACC) are utilized to evaluate the clustering performance. Specifically, let $a_i$ be the number of cells from cluster $i$ according to the ground-truth labels, $b_j$ be the number of cells assigned to cluster $j$ by the algorithm, and $n_{ij}$ be the number of cells that simultaneously belongs to cluster $i$ based on ground-truth labels and cluster $j$ in cluster assignments predicted by the algorithm, ARI measures the similarity between the cluster assignments and ground-truth labels, which is defined as follows:

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]\Big/\binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2}+\sum_j \binom{b_j}{2}\right]-\left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]\Big/\binom{N}{2}}. \tag{19}$$

Let $v_i$ and $u_i$ denote the ground-truth label and cluster assignment of data point $i$, ACC is computed by the best matching between the ground truth and cluster assignment, i.e.,

$$\text{ACC} = \frac{\sum_{i=1}^{N}\delta(v_i, \text{map}(u_i))}{N}, \quad \delta(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

where map() corresponds to the best mapping found by the Hungarian algorithm [61].

In addition, to reflect how robust is the clustering results toward batch effects, we propose using KL divergence between the observed and expected cell-batch co-occurrence matrix to measure the robustness against batch effects. Specifically, let $O^{K\times B}$ and $E^{K\times B}$ be the observed and expected co-occurrence matrices which are based on predictions and ground truth labels, respectively, the robustness metric is defined as follows:

$$\text{BER} = \sum_{b=1}^{B}\sum_{k=1}^{K} O_{bk}\log\left(\frac{O_{kb}+1}{E_{kb}+1}\right) \in [0, \infty) \tag{21}$$

where $O_{kb}/E_{kb}$ is the number of cells belonging to the predicted/ground-truth cluster $k$ and batch $b$. Like the ACC metric, the Hungarian algorithm [61] is first applied to align cluster assignments with ground-truth labels before computing BER. Ideally, BER would become zero, and a smaller BER value indicates better robustness.

### D. Comparisons With State of the Arts

To the best of our knowledge, only a few clustering methods have been proposed to handle the data with batch effects, among which DESC [15] is a representative baseline. To further stress the effectiveness of SCDC, we compare it with the state-of-the-art batch integration methods, including fastMNN [18], Harmony [26], scMerge [21], Seurat 3 [22], and ZINB-WaVE [24]. As these batch integration methods are not specifically designed for clustering, we conduct $k$-means on the integrated representation to achieve clustering. All methods are run seven times with different random seeds and the average performance with standard deviation was reported in Table II. As can be seen, our SCDC outperforms existing methods by a large margin, which proves the effectiveness of our disentanglement idea.

The performance of existing batch integration methods might be limited by their heavy assumption that the biological variances are much smaller than the batch differences. However, we find that the assumption does not always hold in practice. As an example, we illustrate the cosine similarity between cells from batch 5 (GSE81608) and the other four batches of the Human Pancreas dataset in Fig. 1(b). If the assumption is held, the similarity matrix should be diagonal at the rows and columns of four types of cells shared across

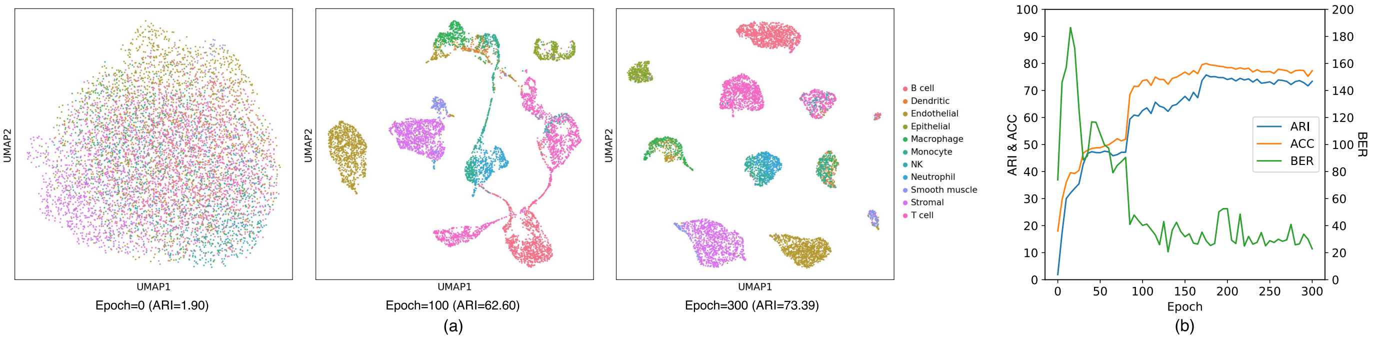


Fig. 4. Visualization of the training process of SCDC on the Mouse Atlas dataset. (a) UMAP plots of the cell embedding at different epochs. (b) Quantitative metrics across the training process.

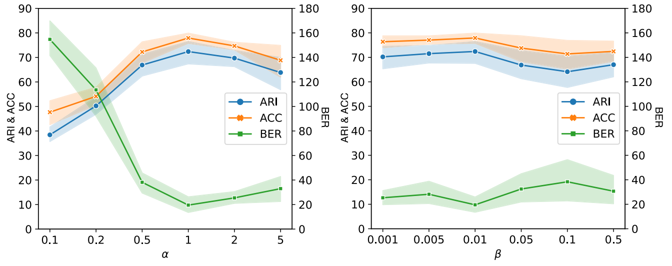


Fig. 5. Performance of SCDC on mouse atlas with different choices of hyperparameters $\alpha$ and $\beta$.

these batches, including alpha, beta, delta, and gamma. Nevertheless, some cases violate the assumption. For example, delta cells from batch 1 (GSE85241) are similar to beta, delta, and gamma cells from batch 5; alpha cells from batch 5 have a similar cosine distance to alpha, beta, acinar, and delta cells from batch 3 (GSE84133); delta and gamma cells from batch 4 (GSE83138) shares high similarities with all cells from batch 5. In other words, when the batch effects become severer, the assumption of biological variances and batch differences could be violated. As a result, most existing methods that build upon the assumption would achieve inferior performance. On the contrary, the proposed SCDC does not rely on such an assumption, which provides a more general solution.

### E. Visualization

To provide an intuitive understanding of the clustering results, we adopt uniform manifold approximation and projection (UMAP) [62] to visualize the features extracted by the cell-type encoder in Fig. 3, compared with features learned by other baseline methods. Ideally, cells from different batches, but of the same type should be clustered together. In other words, clusters are expected to be pure in odd rows, but have more diverse colors in even rows. As shown, fastMNN, Harmony, and Seurat3 improperly enrich the cell types, while LIGER, scMerge, and ZINB-WaVE fail to distinguish some types of cells on the mouse atlas. On cell line, DESC and LIGER are severely influenced by the batch effects, leading to wrong data partition. On the contrary, our SCDC shows superior robustness against batch effects, and it clusters scRNA-seq data based on cell types as expected.

To reveal the clustering procedure of SCDC, we visualize the embeddings and quantitative metrics across the training process. As shown in Fig. 4(a), SCDC learns more compact clusters and clearer cluster boundaries as the training proceeds, which is also reflected in the steadily growing clustering metrics ARI and ACC in Fig. 4(b). Notably, the BER metric is relatively low at the start. Such a phenomenon is reasonable since the initial random embedding contains no batch effects nor cell-type information. Thanks to the within-batch shuffling and batch discrimination strategies, SCDC gradually disentangles batch effects from the cell-type information, and finally converges to a low BER metric.

### F. Ablation Study and Parameter Analysis

To investigate the effectiveness of the proposed within-batch shuffling (i.e., $\tilde{L}_{\mathrm{ZINB}}$) and batch discrimination (i.e., $L_{\mathrm{CE}}$), we conduct ablation studies on mouse atlas and human pancreas by removing either $\tilde{L}_{\mathrm{ZINB}}$ or $L_{\mathrm{CE}}$ from (17). As shown in Table III, the within-batch shuffling strategy is essential to disentangle the biological and nonbiological information from data. Without the strategy, cell-type information would leak into the batch representation. Moreover, the batch discrimination strategy also improves the performance as it encourages the batch encoder to capture nonbiological information for better disentanglement.

We further evaluate SCDC with different hyperparameters $\alpha$ and $\beta$ that weigh the strength of two strategies in Fig. 5. Specifically, for the within-batch shuffling strategy, SCDC cannot fully disentangle biological and nonbiological variances when $\alpha$ is too small, leading to poor batch effect robustness. Meanwhile, an over-large $\alpha$ would hinder SCDC from capturing biological information through the standard ZINB reconstruction. For the batch discrimination strategy, the performance of SCDC is stable for $\beta \in [0.001, 0.01]$ but decreases slightly when $\beta$ is too large. The reason behind such a phenomenon is as follows. Though being similar across within-batch cells, batch effects still slightly vary in different cells, especially for those of diverse types. However, an over-large $\beta$ tends to make the batch information homogeneous across all cells from the same batch, thus leading to inferior performance.

TABLE III

ABLATION STUDY ON WITHIN-BATCH SHUFFLING AND BATCH DISCRIMINATION

| | | Mouse Atlas | | | Human Pancreas | | |
|---|---|---|---|---|---|---|---|
| $\tilde{L}_{\text{ZINB}}$ | $L_{\text{CE}}$ | ARI | ACC | BER | ARI | ACC | BER |
| ✓ | ✓ | **72.39** | **77.91** | **19.54** | **90.04** | **86.98** | **19.88** |
| ✓ | | 65.95 | 70.57 | 36.98 | 88.13 | 84.87 | 25.18 |
| | ✓ | 2.80 | 22.45 | 454.49 | 3.96 | 32.28 | 188.04 |

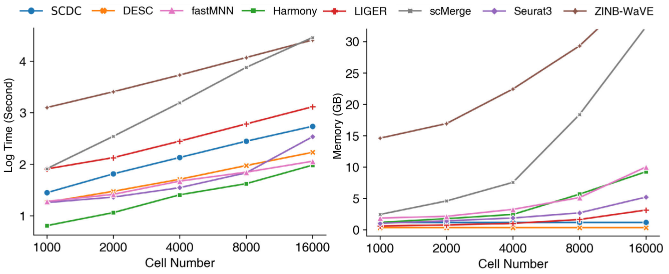


Fig. 6. Running time and memory consumption of different methods for the various number of cells.

## G. Scalability

With the development of sequencing techniques, the number of cells profiled in scRNA-seq experiments grows continually, arousing the demand for efficiently handling large scRNA-seq data. To access how SCDC scales to large data, we evaluate its time and memory consumption when applied to 2000–16 000 cells. As shown in Fig. 6, the running time of SCDC increases linearly with cell numbers. Although SCDC consumes more time than Harmony, the speed of SCDC could be further improved by parallel training with multiple GPUs thanks to its mini-batch optimization. Another benefit of SCDC is that its memory consumption stays constant, whereas other methods such as Harmony require at least linearly increasing memory with cell numbers. Moreover, a more well-chosen optimizer and a dedicated learning rate scheduler could speed up the convergence of SCDC and save training epochs. In short, the linear time and constant memory consumptions make SCDC favorable to handle large scRNA-seq data. Despite the computational efficiency, SCDC also achieves better clustering performance on two large datasets (i.e., mouse retina and mouse brain) as shown in Table II, which further strengthens its scalability.

# V. CONCLUSION

In this work, we present SCDC, an end-to-end clustering method that is debiased toward nonbiological factors and partitions data solely based on biological information. With the designed disentanglement learning framework, SCDC could successfully extract cell-type information from scRNA-seq data confounded with a variety of batch effects, without the assumption of biological variances and batch effects. Different from iterative methods like LIGER [25] and Harmony [26], SCDC simultaneously and directly performs batch effect removal and clustering, which is more simple yet effective, avoiding the potential error accumulation during the alternation. Evaluations on six benchmarks demonstrate the superiority of SCDC compared with both batch integration and clustering methods. Furthermore, SCDC scales to large data with linearly increasing running time and a constant GPU memory consumption. Considering its simplicity, effectiveness, and scalability, SCDC would be a promising tool for clustering scRNA-seq data with a larger number of cells and severer batch effects, catering to the general trend brought by the rapid development of scRNA-seq technologies.

# ACKNOWLEDGMENT

The authors would like to thank the anonymous associate editors and reviewers for their valuable comments and constructive suggestions to improve the quality of this article.

# REFERENCES

[1] V. Y. Kiselev, T. S. Andrews, and M. Hemberg, "Challenges in unsupervised clustering of single-cell RNA-seq data," *Nature Rev. Genet.*, vol. 20, no. 5, pp. 273–282, May 2019.

[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1. Oakland, CA, USA, 1967, pp. 281–297.

[3] V. Y. Kiselev et al., "SC3: Consensus clustering of single-cell RNA-seq data," *Nature Methods*, vol. 14, no. 5, pp. 483–486, 2017.

[4] J. Chen, A. Schlitzer, S. Chakarov, F. Ginhoux, and M. Poidinger, "Mpath maps multi-branching single-cell trajectories revealing progenitor cell progression during development," *Nature Commun.*, vol. 7, no. 1, pp. 1–15, Jun. 2016.

[5] B. Wang, J. Zhu, E. Pierson, D. Ramazzotti, and S. Batzoglou, "Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning," *Nature Methods*, vol. 14, no. 4, pp. 414–416, Apr. 2017.

[6] L. A. Bugnon, C. Yones, D. H. Milone, and G. Stegmayer, "Deep neural architectures for highly imbalanced data in bioinformatics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2857–2867, Aug. 2020.

[7] D.-J. Zhang, Y.-L. Gao, J.-X. Zhao, C.-H. Zheng, and J.-X. Liu, "A new graph autoencoder-based consensus-guided model for scRNA-seq cell type detection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 20, 2022, doi: 10.1109/TNNLS.2022.3190289.

[8] X. Zhang, Y. Han, L. Wu, and Y. Wang, "State estimation for delayed genetic regulatory networks with reaction–diffusion terms," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 299–309, Feb. 2018.

[9] Y. Guo, D. Zhou, P. Li, C. Li, and J. Cao, "Context-aware Poly(A) signal prediction model via deep spatial–temporal neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 12, 2022, doi: 10.1109/TNNLS.2022.3226301.

[10] C. Y. Lee and Y.-P. P. Chen, "New insights into drug repurposing for COVID-19 using deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4770–4780, Nov. 2021.

[11] C. Li et al., "Geometry-based molecular generation with deep constrained variational autoencoder," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 16, 2022, doi: 10.1109/TNNLS.2022.3147790.

[12] L. Hu, J. Zhang, X. Pan, X. Luo, and H. Yuan, "An effective link-based clustering algorithm for detecting overlapping protein complexes in protein-protein interaction networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 3275–3289, Dec. 2021.

[13] J. Hu, X. Li, G. Hu, Y. Lyu, K. Susztak, and M. Li, "Iterative transfer learning with neural network for clustering and cell type classification in single-cell RNA-seq analysis," *Nature Mach. Intell.*, vol. 2, no. 10, pp. 607–618, Oct. 2020.

[14] Q. Liu, S. Chen, R. Jiang, and W. H. Wong, "Simultaneous deep generative modelling and clustering of single-cell genomic data," *Nature Mach. Intell.*, vol. 3, no. 6, pp. 536–544, May 2021.

[15] X. Li et al., "Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis," *Nature Commun.*, vol. 11, no. 1, p. 2338, May 2020.

[16] T. Tian, J. Wan, Q. Song, and Z. Wei, "Clustering single-cell RNA-seq data with a model-based deep learning approach," *Nature Mach. Intell.*, vol. 1, no. 4, pp. 191–198, Apr. 2019.

[17] T. Tian, J. Zhang, X. Lin, Z. Wei, and H. Hakonarson, "Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data," *Nature Commun.*, vol. 12, no. 1, pp. 1–12, Mar. 2021.

[18] L. Haghverdi, A. T. L. Lun, M. D. Morgan, and J. C. Marioni, "Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors," *Nature Biotechnol.*, vol. 36, no. 5, pp. 421–427, May 2018.

[19] K. Polański, M. D. Young, Z. Miao, K. B. Meyer, S. A. Teichmann, and J.-E. Park, "BBKNN: Fast batch alignment of single cell transcriptomes," *Bioinformatics*, vol. 36, no. 3, pp. 964–965, Feb. 2020.

[20] B. Hie, B. Bryson, and B. Berger, "Efficient integration of heterogeneous single-cell transcriptomes using scanorama," *Nature Biotechnol.*, vol. 37, no. 6, pp. 685–691, Jun. 2019.

[21] Y. Lin et al., "ScMerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell RNA-seq datasets," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 20, pp. 9775–9784, May 2019.

[22] T. Stuart et al., "Comprehensive integration of single-cell data," *Cell*, vol. 177, no. 7, pp. 1888–1902, 2019.

[23] W. E. Johnson, C. Li, and A. Rabinovic, "Adjusting batch effects in microarray expression data using empirical Bayes methods," *Biostatistics*, vol. 8, no. 1, pp. 118–127, Jan. 2007.

[24] D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert, "A general and flexible method for signal extraction from single-cell RNA-seq data," *Nature Commun.*, vol. 9, no. 1, pp. 1–17, Jan. 2018.

[25] J. Welch, V. Kozareva, A. Ferreira, C. Vanderburg, C. Martin, and E. Macosko, "Integrative inference of brain cell similarities and differences from single-cell genomics," *BioRxiv*, vol. 2018, Nov. 2018, Art. no. 459891.

[26] I. Korsunsky et al., "Fast, sensitive and accurate integration of single-cell data with harmony," *Nature Methods*, vol. 16, no. 12, pp. 1289–1296, Dec. 2019.

[27] U. Shaham et al., "Removal of batch effects using distribution-matching residual networks," *Bioinformatics*, vol. 33, no. 16, pp. 2539–2546, Aug. 2017.

[28] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," *Nature Methods*, vol. 15, no. 12, pp. 1053–1058, Dec. 2018.

[29] M. Lotfollahi, F. A. Wolf, and F. J. Theis, "ScGen predicts single-cell perturbation responses," *Nature Methods*, vol. 16, no. 8, pp. 715–721, Aug. 2019.

[30] T. Wang et al., "BERMUDA: A novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes," *Genome Biol.*, vol. 20, no. 1, pp. 1–15, Dec. 2019.

[31] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2843–2849.

[32] Q. Wang, M. Chen, F. Nie, and X. Li, "Detecting coherent groups in crowd scenes by multiview clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 46–58, Oct. 2020.

[33] Y. Yang, J. Feng, N. Jojic, J. Yang, and T. S. Huang, "$\ell^0$-sparse subspace clustering," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 731–747.

[34] X. Shen, W. Liu, I. Tsang, F. Shen, and Q.-S. Sun, "Compressed $k$-means for large-scale clustering," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[35] J. Yi, L. Zhang, R. Jin, Q. Qian, and A. Jain, "Semi-supervised clustering by input pattern assisted pairwise similarity matrix completion," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1400–1408.

[36] Z. Kang, C. Peng, Q. Cheng, and Z. Xu, "Unified spectral clustering with optimal graph," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 3366–3373.

[37] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[38] S. Li, K. Li, and Y. Fu, "Temporal subspace clustering for human motion segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4453–4461.

[39] L. Hu, X. Pan, Z. Tang, and X. Luo, "A fast fuzzy clustering algorithm for complex networks via a generalized momentum method," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 9, pp. 3473–3485, Sep. 2022.

[40] D. Cheng, J. Huang, S. Zhang, X. Zhang, and X. Luo, "A novel approximate spectral clustering algorithm with dense cores and density peaks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 4, pp. 2348–2360, Apr. 2022.

[41] D. Wu, X. Dong, J. Cao, R. Wang, F. Nie, and X. Li, "Bidirectional probabilistic subspaces approximation for multiview clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 16, 2022, doi: 10.1109/TNNLS.2022.3217032.

[42] N. Lu, H. Xiao, Z. Ma, T. Yan, and M. Han, "Domain adaptation with self-supervised learning and feature clustering for intelligent fault diagnosis," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 15, 2022, doi: 10.1109/TNNLS.2022.3219896.

[43] E. Osipov et al., "Hyperseed: Unsupervised learning with vector symbolic architectures," *IEEE Trans. Neural Netw. Learn. Syst.*, early accss, Nov. 16, 2022, doi: 10.1109/TNNLS.2022.3211274.

[44] L. Yang, Q. Zhou, and B. Lu, "Marginal subspace learning with group low-rank for unsupervised domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 14, 2022, doi: 10.1109/TNNLS.2022.3218554.

[45] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2017, pp. 373–382.

[46] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep spectral clustering using dual autoencoder network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2019, pp. 4066–4075.

[47] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5736–5745.

[48] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 8547–8555.

[49] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "COMPLETER: Incomplete multi-view clustering via contrastive prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11174–11183.

[50] Z. Yu, Y. Lu, Y. Wang, F. Tang, K.-C. Wong, and X. Li, "ZINB-based graph embedding autoencoder for single-cell RNA-seq interpretations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 4, pp. 4671–4679, Jun. 2022.

[51] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell RNA-seq denoising using a deep count autoencoder," *Nature Commun.*, vol. 10, no. 1, pp. 1–14, Jan. 2019.

[52] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," 2016, *arXiv:1611.01144*.

[53] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.

[54] T. M. Cover, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1999.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[56] F. A. Wolf, P. Angerer, and F. J. Theis, "SCANPY: Large-scale single-cell gene expression data analysis," *Genome Biol.*, vol. 19, no. 1, pp. 1–5, Dec. 2018.

[57] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[58] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature Biotechnol.*, vol. 36, no. 5, pp. 411–420, May 2018.

[59] A. T. L. Lun, D. J. McCarthy, and J. C. Marioni, "A step-by-step workflow for low-level analysis of single-cell RNA-seq data with bioconductor," *FResearch*, vol. 5, p. 2122, Oct. 2016.

[60] Y. Hao et al., "Integrated analysis of multimodal single-cell data," *Cell*, vol. 184, no. 13, pp. 3573–3587, 2021.

[61] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1995.

[62] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.

**Yunfan Li** received the bachelor's degree in computer science and technology from Sichuan University, Chengdu, China, in 2020, where he is currently pursuing the Ph.D. degree in computer science with the College of Computer Science.

His research interests include unsupervised learning and bioinformatics.

**Yijie Lin** received the bachelor's degree in computer science and technology from Sichuan University, Chengdu, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Computer Science.

His research interests include multimodal learning.

**Peng Hu** received the Ph.D. degree in computer science and technology from Sichuan University, Chengdu, China, in 2019.

He is currently an Associate Research Professor at the College of Computer Science, Sichuan University. He has authored more than 30 articles in the top-tier conferences and journals. His research interests mainly focus on multiview learning, cross-modal retrieval, and network compression.

**Dezhong Peng** received the Ph.D. degree in computer software and theory from the University of Electronic Science and Technology of China, Chengdu, China, in 2006.

From 2001 to 2007, he was with the University of Electronic Science and Technology of China, as an Assistant Lecturer and a Lecturer. He was a Post-Doctoral Research Fellow with the School of Engineering, Deakin University, Geelong, VIC, Australia, from 2007 to 2009. He is currently a Professor with the College of Computer Science, Sichuan University, Chengdu. His research interests include blind signal processing and neural networks.

**Han Luo** is currently a Full Faculty (Associate Professor and Board Certificated Surgeon) in West China hospital, Sichuan University, Chengdu, China. He has authored more than 30 articles including *Science Advances*, *Nature Communications*, *The Journal of Clinical Endocrinology and Metabolism*, *International Journal of Surgery*, and six national patents. He focuses on cancer cell plasticity at single cell resolution in multiple cancers, also does effort into translational research.

Dr. Luo serves as an Associate Present for Provincial Thyroid Association and a Guest Editor for *Journal of Visualized Experiments* and *Measles-Mumps-Rubella* journals.

**Xi Peng** (Member, IEEE) is currently a Full Professor at the College of Computer Science, Sichuan University, Chengdu, China. He has authored more than 80 articles published in *Journal of Machine Learning Research*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, International Conference on Machine Learning, and Conference on Neural Information Processing Systems. His current interests mainly focus on machine learning and multimedia analysis.

Dr. Peng has served as an Associate Editor for four journals such as IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and a Guest Editor for four journals such as IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.