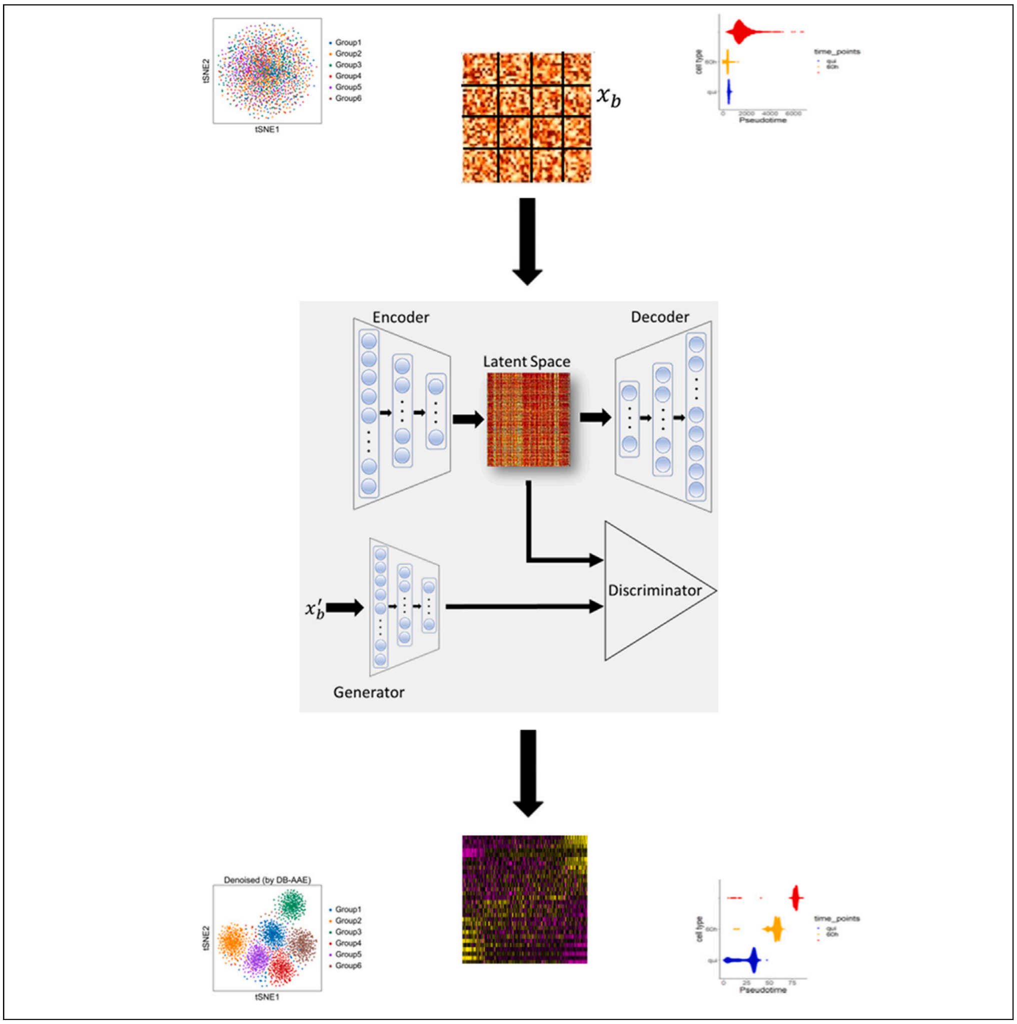
iScience

CellPress
OPEN ACCESS

Article

A deep learning adversarial autoencoder with dynamic batching displays high performance in denoising and ordering scRNA-seq data



Kyung Dae Ko,
Vittorio Sartorelli

kyundae.ko@nih.gov (K.D.K.)
vittorio.sartorelli@nih.gov (V.S.)

Highlights

The dynamic batching adversarial autoencoder (DB-AAE) excels at denoising scRNA-seq datasets

DB-AAE enhances resolution of pseudo-time inference

DB-AAE outperforms other methods in denoising accuracy and biological signal preservation

Ko & Sartorelli, iScience *27*, 109027
March 15, 2024
https://doi.org/10.1016/j.isci.2024.109027

iScience

CellPress
OPEN ACCESS

Article

A deep learning adversarial autoencoder with dynamic batching displays high performance in denoising and ordering scRNA-seq data

Kyung Dae Ko[1,*] and Vittorio Sartorelli[1,2,*]

SUMMARY

By providing high-resolution of cell-to-cell variation in gene expression, single-cell RNA sequencing (scRNA-seq) offers insights into cell heterogeneity, differentiating dynamics, and disease mechanisms. However, challenges such as low capture rates and dropout events can introduce noise in data analysis. Here, we propose a deep neural generative framework, the dynamic batching adversarial autoencoder (DB-AAE), which excels at denoising scRNA-seq datasets. DB-AAE directly captures optimal features from input data and enhances feature preservation, including cell type-specific gene expression patterns. Comprehensive evaluation on simulated and real datasets demonstrates that DB-AAE outperforms other methods in denoising accuracy and biological signal preservation. It also improves the accuracy of other algorithms in establishing pseudo-time inference. This study highlights DB-AAE's effectiveness and potential as a valuable tool for enhancing the quality and reliability of downstream analyses in scRNA-seq research.

INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) has revolutionized gene expression profiling by revealing the transcriptome of individual cells. This method has provided valuable insights into cell heterogeneity, facilitated the discovery of rare cell populations, and enhanced our understanding of the molecular mechanisms underlying cellular function and disease.[1] Despite important analytical advances, scRNA-seq still faces certain technical challenges, including low capture rates and dropout events. These limitations introduce noise that can interfere with data analysis and interpretation.

Dropout is a phenomenon observed when a given gene transcript is expressed at a low or moderate expression in one cell but is not detected in another cell of the same cell-type population.[2] It occurs due to low sequencing depth, amplification bias, or biological factors, and can impact on downstream analysis such as clustering, trajectory analysis, and differential expression analysis. To mitigate the effects of dropout, numerous imputation or denoising methods have been developed that can be categorized into matrix factorization, nearest-neighbor method, probabilistic model, and deep learning-based method.[3]

Matrix factorization decomposes a matrix into lower-rank matrices to approximate the original matrix and estimates missing values based on data patterns.[4] The accuracy of the imputed values depends on the characteristics of the data and the selection of factorization method and hyperparameters. Nearest-neighbor methods, such as K-nearest neighbors (KNN) imputation, estimate missing values by considering values from the nearest neighbors.[5] KNN has a high computational cost to impute or denoise large datasets, and the accuracy decreases if the proportion of missing value is high in the dataset, or the missing values are not related to the key values. Probabilistic models, like the zero-inflated negative binomial (ZINB) model and Gaussian mixture model (GMM), infer missing values based on observed information and distribution assumptions.[3] While useful, these methods can introduce biases in the denoised dataset and struggles to accurately denoise datasets, when the proportion of missing values is high, or the distribution of missing values is non-random.

Deep learning methods, specifically autoencoders, have been developed to capture non-linear relationships in scRNA-seq data.[6–9] Autoencoders use feature extraction and latent space reconstruction to reduce noise and impute missing values. However, they can be sensitive to sparse data and batch effects.[10] To address these issues, the deep count autoencoder (DCA) combines autoencoders with a negative binomial model to capture missing values and mitigate batch effects.[11] Since mean and dispersion of an input matrix are used for the reconstruction of latent space, DCA can suffer overfitting and information loss. Variational autoencoders (VAEs) further improve upon this approach by incorporating a probabilistic generative framework, creating a smoother latent space, and capturing complex non-linear patterns among gene expression values.[12] However, VAEs can suffer from mode collapse or loss of informative features in the latent space if lowly expressed genes in the dataset do not follow a particular statistical distribution such as negative binomial or Gaussian.

[1] Laboratory of Muscle Stem Cells & Gene Regulation, NIAMS, NIH, Bethesda, MD, USA
[2] Lead contact
*Correspondence: kyundae.ko@nih.gov (K.D.K.), vittorio.sartorelli@nih.gov (V.S.)

https://doi.org/10.1016/j.isci.2024.109027

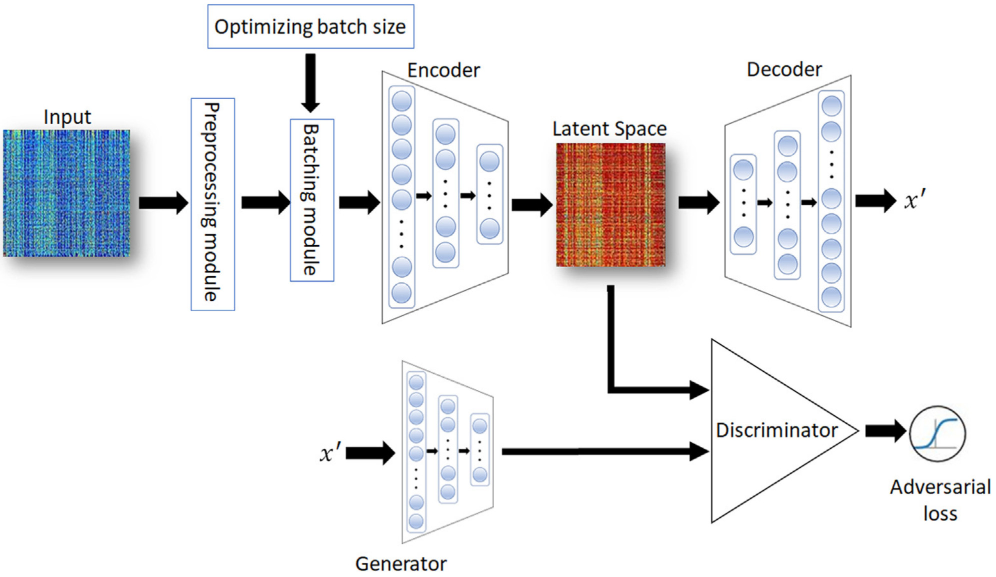


Figure 1. Dynamic batching adversarial autoencoder (DB-AAE)

Schematic illustration of the different components of the DB-AAE. Blue circles represent nodes, x' represent reconstructed output in the DB-AAE structure. After preprocessing data, the encoder generates the authentic latent space using the current input batch, while the generator creates a simulated latent space by emulating the characteristics of the output through the autoencoder with the prior input batch during training. The encoder and decoder components are optimized until the discriminator cannot differentiate between the true and simulated latent spaces across the entire batch.

In this paper, our objective is to tackle the challenges associated with imputation and denoising in scRNA-seq data using a novel generative framework that leverages the power of adversarial autoencoders (AAEs). AAE combines autoencoders and generative adversarial networks (GANs).[13] Traditional AAE frameworks primarily focus on training the generator and the encoder through the adversarial network to generate realistic outputs, making it difficult for the discriminator to distinguish between the generated and real data. Originally designed for synthesizing realistic images, AAEs with statistical models have been applied in scRNA-seq data for tasks such as dimension reduction, clustering, and integration.[14,15] However, the potential of AAEs in denoising and imputing scRNA-seq datasets remains underexplored. In addition, while traditional AAEs exhibit good performance in the analysis of scRNA-seq data,[15] there is a risk of information loss if the variance of gene expression does not follow the statistical models.[10] To enhance denoising performance and mitigate information loss during analysis, we propose the dynamic batching adversarial autoencoder (DB-AAE) employing a competitive model that directly captures optimal features from input data rather than using statistical models. Batching is one of pivotal techniques in deep learning, wherein multiple input samples are processed concurrently as a batch.[16] Numerous studies[17–20] emphasize the crucial impact of batch size on the performance of training in deep neural networks. A large batch size may become stuck at local minima, while a small batch size can lead the loss function to converge to a biased minimum.[16] Adapting the batch size according to the dataset's characteristics has been shown to enhance the efficiency of neural network algorithms.[16] To dynamically adjust the batch size during neural network training, three prominent algorithms are considered. First, random search[21] involves the random selection of combinations of batch sizes. Second, Bayesian optimization[22] utilizes Bayes Theorem to guide the search for the optimal batch size. Lastly, the Hyperband approach,[23] a variant of random search, aims to determine the best resource allocation for adjusting the batching size. In our research, we employ the Hyperband algorithm for dynamic batching in AAEs to enhance the reconstruction performance and converge to an optimal minimum in the loss function. DB-AAE excels at retaining important features, such as cell type-specific gene expression patterns, even in the presence of noise in scRNA-seq data. This enhanced feature preservation significantly improves the reliability and accuracy of downstream analysis tasks such as clustering and pseudo-time inference.

We tested and performed a comprehensive evaluation of our proposed method, comparing it with other commonly used approaches, using both simulated and real datasets. Our analysis indicates that DB-AAE surpasses the performance of other methods in terms of denoising accuracy and preservation of biological signal. Moreover, our findings indicate that this method can significantly enhance the accuracy of other algorithms specifically designed for pseudo-time inference. These results not only validate the effectiveness of our approach but also emphasize its potential as a valuable tool for improving the quality and reliability of downstream analyses in scRNA-seq analysis.

RESULTS

DB-AAE improves denoising of simulated scRNA-seq data

AAE is a deep neural network that combines the advantages of autoencoders and GANs to facilitate unsupervised learning tasks and generate new samples that closely resemble the input data by sampling from a learned latent space, and there are several advantages

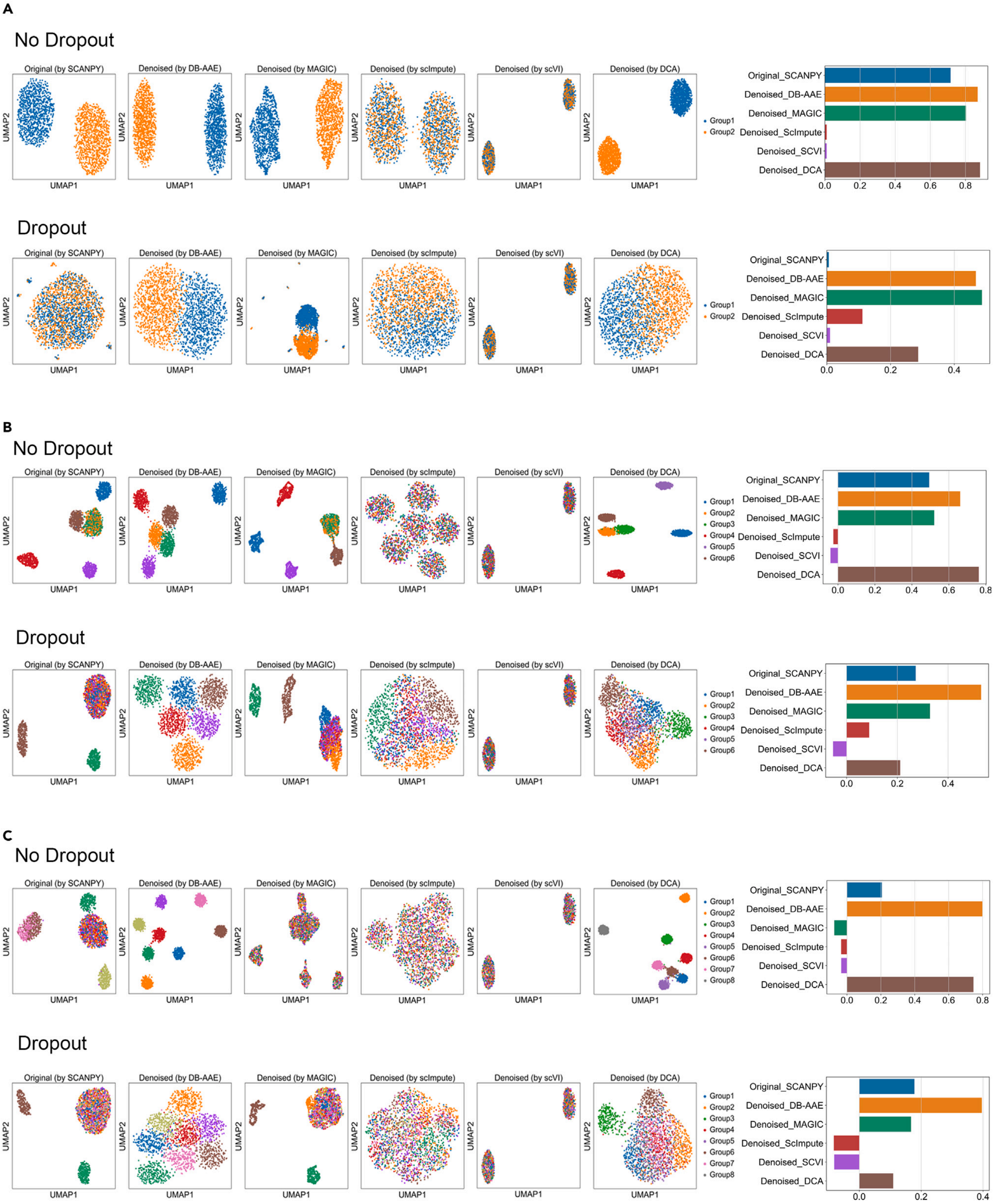

A
No Dropout
Original (by SCANPY)
Denoised (by DB-AAE)
Denoised (by MAGIC)
Denoised (by scImpute)
Denoised (by scVI)
Denoised (by DCA)
UMAP1
UMAP2
Group1
Group2
Original_SCANPY
Denoised_DB-AAE
Denoised_MAGIC
Denoised_ScImpute
Denoised_SCVI
Denoised_DCA
0.0
0.2
0.4
0.6
0.8
Dropout
0.0
0.2
0.4
B
No Dropout
Group1
Group2
Group3
Group4
Group5
Group6
0.0
0.2
0.4
0.6
0.8
Dropout
0.0
0.2
0.4
C
No Dropout
Group1
Group2
Group3
Group4
Group5
Group6
Group7
Group8
0.0
0.2
0.4
0.6
0.8
Dropout
0.0
0.2
0.4

Figure 2. Identification of cell types in simulated data

(A) UMAP (left) and silhouette score bar (right) plots of two distinct cell types without dropout noise (top) and with dropout noise (bottom) using different algorithms.

(B) Six distinct virtual cell types without dropout noise (top) and with dropout noise (bottom).

(C) Eight distinct virtual cell types without dropout noise (top) and with dropout noise (bottom).

of AAEs compared to traditional GANs. First, since AAEs are designed to perform both generative and reconstructive tasks, AAEs can be efficient for tasks such as data denoising and imputation. Second, AAEs can control the generation process of latent space because they explicitly encode input data into a latent space. This makes it easier to decode output data with specific characteristics of the input. However, GANs do not provide a direct mapping from input to a latent space. AAE consists of three key components: encoder, decoder, and adversary modules (Figure 1). The encoder transforms input data into a lower-dimensional latent space, while a decoder reconstructs the input data from the latent space. The adversary modules encompass a generator and discriminator. The generator utilizes the latent space encoded by samples from the input data to produce synthetic data, whereas the discriminator distinguishes between the synthetic data and real data obtained from the original input's latent space. In AAE, the encoder and decoder components are optimized in such a way that the discriminator cannot differentiate between synthetic samples generated by the generator and real data. This adversarial training process empowers the AAE to acquire a more meaningful and structured representation of the latent space.[13] For their optimization, several statistical models have been used but information may be lost if the distribution does not follow statistical models.[10] To remedy this potential loss, we implemented DB-AAE containing a novel adversarial framework with dynamic batching by sampled inputs.

To investigate the characteristics of the DB-AAE model, we conducted performance evaluation using simulated scRNA-seq data generated by the Splatter package[24] after creating a count matrix consisting of 200 genes across 2,000 cells. We modified the simulation to introduce variations in the number of cell types (two, six, or eight virtual cell types) under either dropout or non-dropout conditions. Clustering efficacy and denoising capabilities of the DB-AAE model were evaluated by comparing it to five other methods, SCANPY, MAGIC (Markov affinity-based graph imputation of cells),[25] DCA,[11] scImupte,[26] and SCVI (single cell variational inference).[12] Figure 2 (left panels) illustrates the clustering results obtained by each method in the uniform manifold approximation and projection (UMAP) dimension. The performance of each clustering was assessed using the silhouette score (SC).[7,27] The SC (Figure 2, right panels) quantifies the similarity of gene expression patterns within a cluster and the dissimilarity between different clusters, with values ranging from −1 to +1.[28] An SC approaching 1 indicates that the clustering results are well-defined and that the cells are appropriately assigned to their respective clusters, suggesting a more reliable and meaningful clustering outcome. In the absence of dropout-induced noise in the small number of groups, DB-AAE, MAGIC, and DCA were able to regenerate clusters corresponding to the number of cell types, and their SCs did not differ significantly (no dropout in Figure 2A). However, after denoising datasets containing dropout noise in the large number of groups, DB-AAE exhibited superior performance compared to other algorithms in clustering cells belonging to the same cell types (dropout Figures 2B and 2C). In fact, while DB-AAE showed similar performance without noise in Figure 2 (no dropout), DB-AAE demonstrated superior performance compared to other methods in complex datasets with strong noise, as shown in Figure 2 (dropout), even though the silhouette score of the original simulated dataset in dropout is negative because the dataset is highly diverse and thus difficult to cluster. These simulated results provide evidence that DB-AAE outperforms other methods in terms of denoising and clustering efficiency.

DB-AAE favorably compares to other approaches in denoising real scRNA-seq data

The denoising performance of DB-AAE was compared to five popular methods (SCANPY for clustering,[29] MAGIC using Markov affinity-based graph imputation,[25] DCA using deep count autoencoder,[11] scImupte using a statistical method[26] and SCVI using variational autoencoder[12]) using ten published scRNA-seq datasets reported in Table 1. Our aim was to assess DB-AAE's ability to capture cell-based clusters in datasets with complex cell heterogeneity. In Figure 3A, we first present clustering results of a scRNA-seq mouse pancreas dataset.[30] Compared to the original dataset (SCANPY Figure 3A), DB-AAE more effectively clustered endocrine pancreatic cells (alpha, beta, and delta cells) which were clearly distinct from exocrine ductal pancreatic cells (Figure 3A). Moreover, the similarity among cells belonging to the same cell type obtained by DB-AAE was improved compared to other methods. For instance, insulin-secreting pancreatic beta cells were assigned to one cluster by DB-AAE while other approaches assigned them to two or more clusters (Figure 3A, red clusters). We further analyzed the functional characteristics of the clusters related to pancreatic beta cells using gene ontology (GO) analysis.[31] The three clusters identified as pancreatic beta cells in the original dataset (Figure S1A, left panel, labeled as c0, c1, and c2) shared GO terms not directly related to their endocrine function (Figure S1A, right panel). DB-AAE grouped these three clusters (Figure S1B, left panel) and identified "insulin receptor binding" term, related to pancreatic function,[32] in each of the clusters c0, c1, and c2 (Figure S1B, right panel). Thus, DB-AAE denoising improved accuracy of GO analysis allowing identification of a critical function of pancreatic beta cells which was not evident in the original dataset. To assess the potential impact of biological overfitting on the analysis of a small-sized dataset, we analyzed scRNA-seq dataset comprising blastomeres of mouse embryos (124 cells), spanning zygote to late blastocyst stages.[33] As shown in Figure 3B, DB-AAE successfully identified all clusters associated with cells of five developmental stages. Also in this case, as observed for the pancreas dataset, DB-AAE tightly assigned cells of the same developmental stage to a unique cluster (Figure 3B). Adjacent clusters identify cells with similar genetic-functional characteristics.[34,35] Silhouette scores were calculated for the mouse pancreas dataset[30] analyzed in Figure 3C (left panel) and embryo scRNA-seq dataset[33] (Figure 3C, right panel). DB-AAE consistently achieved higher scores than the other methods across the two datasets. In both cases,

Table 1. References of datasets employed to evaluate performance and pseudo-time inference

Dataset	Tissue	# of cell	# of cell type	Accession ID	Reference
Clustering efficiency					
baron	Mouse pancreas	1886	13	GSE84133	Baron et al.[30]
zeisel	Mouse brain	3005	9	GSE60361	Zeisel et al.[36]
goolam	Mouse embryo	124	5	E-MTAB-3321	Goolam et al.[33]
xin	Human pancreas	1600	8	GSE81608	Xin et al.[37]
lake	Human brain	3042	16	phs000833.v3.p1	Lake et al. 2016
Slyper	Human blood	13316	8	SCP345	Tran et al.[38]
deng	Mouse embryo	268	6	GSE45719	Deng et al.[39]
Wang	Human pancreas	457	7	GSE83139	Wang et al.[40]
Muraro	Human pancreas	2126	10	GSE85241	Muraro et al.[41]
usoskin	Mouse brain	622	4	GSE59739	Usoskin et al.[42]
Pseudo time inference					
sartorelli	Mouse muscle	11046	3	GSE126834	Dell'Orso et al.[43]
ponce	Mouse pancreas	36351	4	GSE132188	Bastidas et al.[44]
treut	Mouse embryo	315	5	GSE67310	Treutlein et al.[45]
qiu	Mouse pancreas	575	7	GSE87375	Qiu et al.[46]
yuzwa	Mouse cortex	6000	4	GSE107122	Yuzwa et al.[47]
vlado	Mouse cerebellum	55000	8	GSE118068	Vladoiu et al.[48]

but especially for the embryo datasets, DB-AAE generated clusters with a closer internal distance between cells compared to the other methods.

To conduct a thorough assessment of DB-AAE performance, we utilized and aggregated ten distinct datasets (Figure 4A), each containing various cell types, ranging from hundreds to thousands, derived from either human or mouse samples. The comparison of silhouette scores across these datasets provides a robust evaluation using six different methods. The average silhouette scores for each method using the ten datasets are presented in Figure 4B. Remarkably, DB-AAE consistently outperformed all other methods across all ten scRNA-seq datasets.

Next, we focused on scRNA-seq datasets derived from skeletal muscle stem cells (MuSCs) to check the biological impact of DB-AAE.[43] In homeostatic condition, MuSCs are quiescent, have a low metabolic rate[49] and a widespread low level of transcription,[50] making it difficult to recover rare transcripts. Isolation procedures lead to transcriptional changes associated with MuSCs activation.[50,51] We employed DB-AAE to assess its capability in detecting expression of rare transcripts expressed in FACS-isolated MuSCs consisting of close-to-quiescence (cQ) and early-activated (eA) MuSCs.[43] Prior to employing DB-AAE, rare transcripts were barely detected and DB-AAE improved their identification (Figure 5A). Next, we wished to evaluate transcripts expressed in cQ MuSCs.[52] Also in this case, DB-AAE greatly improved detection of lowly expressed transcripts (Figure 5B). These results highlight the capability of DB-AAE not only to remove noise but also to recover valuable gene expression patterns that might otherwise be missed.

DB-AAE enhances resolution of pseudo-time inference

Pseudo-time inference is one of important procedures in the analysis of the single cell transcriptome and computationally infers the order of these cells along developmental trajectories.[53] Even though DB-AAE does not have a function of pseudo-time inference, it can support to improve the performance of other algorithms for the inference. Therefore, we performed an evaluation of pseudo-time inference after applying denoising techniques to determine the impact of denoising DB-AAE on pseudo-time inference. We conducted a comprehensive evaluation of three popular autoencoder approaches, aiming to assess their performance on six diverse datasets (Table 1, pseudo-time inference). These datasets encompass various developmental cell states ranging from three to eight, originating from different tissues and organs. After utilizing denoising techniques, we inferred pseudo-time using the widely used Slingshot algorithm.[54] Next, we employed squared R scores to measure the correlation between the inferred pseudo time and annotated developmental stages.[38,55] A higher squared R score indicates a closer alignment between the predicted pseudo-time and the annotated stages. In this analysis, DB-AAE consistently outperformed the other methods across all scRNA-seq datasets (Figure 6A). In Figure 6B, we present the results obtained for MuSCs datasets,[43] comprising three differentiation stages (quiescent MuSCs, activated MuSCs isolated 60 h after muscle injury, and culture myoblasts). Our analysis revealed that, by combining DB-AAE denoised with Slingshot, we achieved higher accurate predictions of the pseudo-time corresponding to the three differentiation cell stages, compared to other methods. These findings highlight the superiority of the DB-AAE denoising method in combination with Slingshot for pseudo-time inference and accurate prediction of developmental stages.

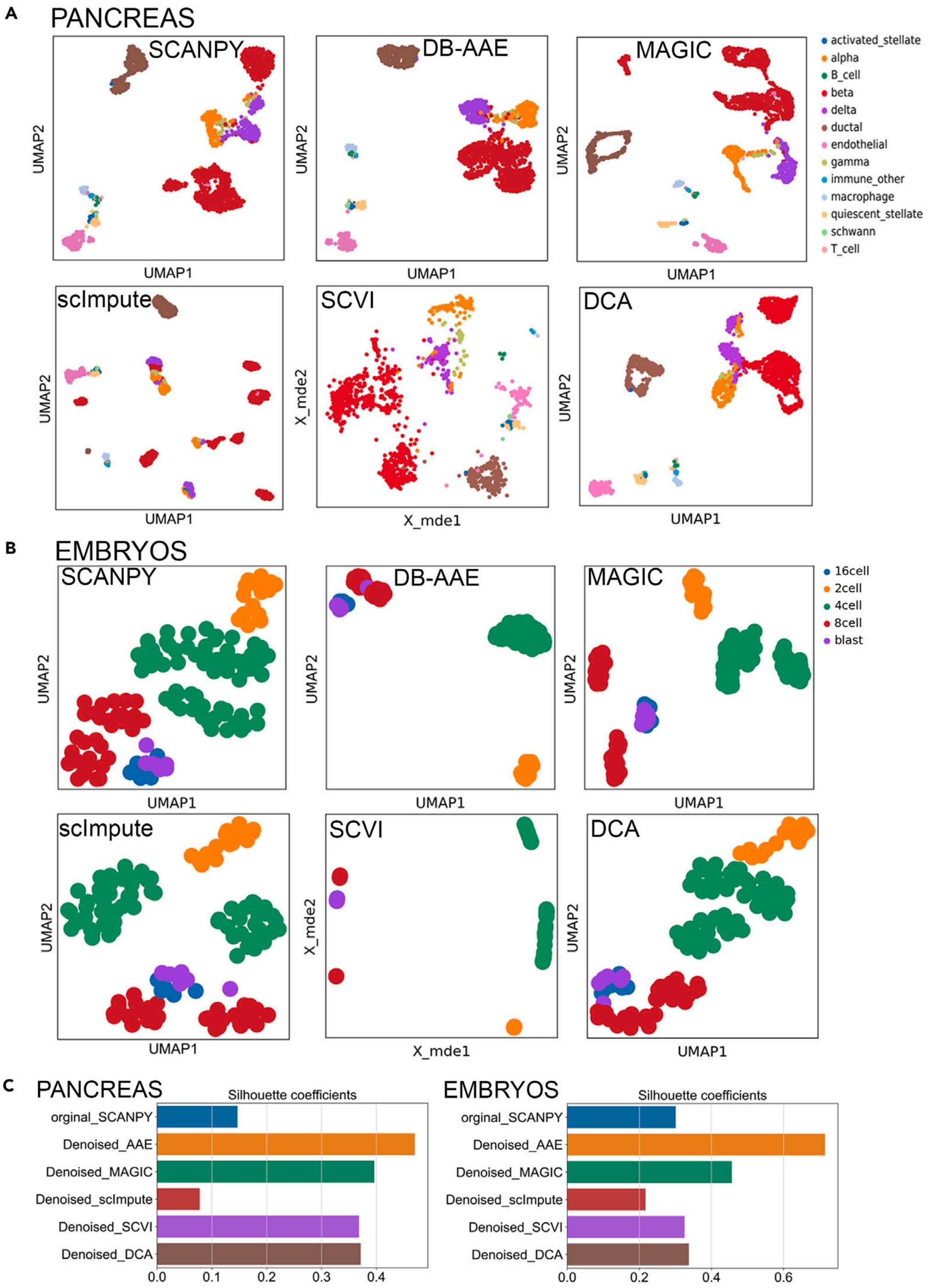


Figure 3. Performance of cell-based clustering with five different methods in biological data

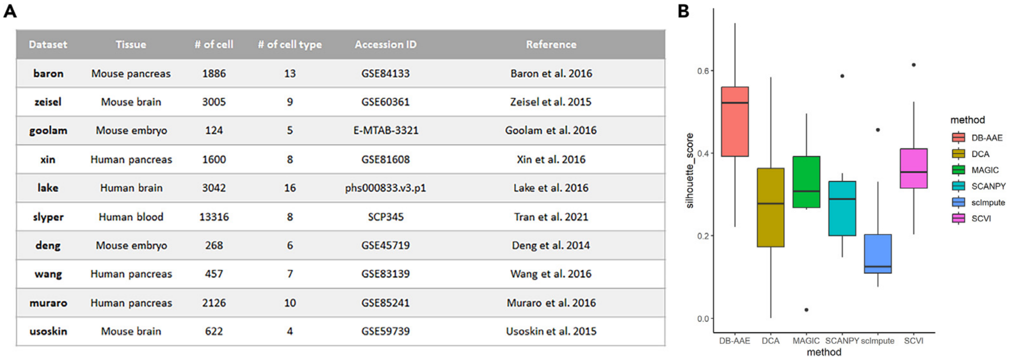
(A) UMAP plot of thirteen cell types from a pancreas dataset.

(B) UMAP plot of five cell types from embryo dataset.

(C) Silhouette score bar plots of thirteen cell types from a pancreas (left) and five cell types from embryo (right) dataset.

DISCUSSION

scRNA-seq provides valuable insights into the diversity of cells and the mechanisms underlying diseases.[1] Nonetheless, this approach comes with challenges, including issues such as limited capture rates and dropout events, which have the potential to introduce undesired variability



Dataset	Tissue	# of cell	# of cell type	Accession ID	Reference
baron	Mouse pancreas	1886	13	GSE84133	Baron et al. 2016
zeisel	Mouse brain	3005	9	GSE60361	Zeisel et al. 2015
goolam	Mouse embryo	124	5	E-MTAB-3321	Goolam et al. 2016
xin	Human pancreas	1600	8	GSE81608	Xin et al. 2016
lake	Human brain	3042	16	phs000833.v3.p1	Lake et al. 2016
slyper	Human blood	13316	8	SCP345	Tran et al. 2021
deng	Mouse embryo	268	6	GSE45719	Deng et al. 2014
wang	Human pancreas	457	7	GSE83139	Wang et al. 2016
muraro	Human pancreas	2126	10	GSE85241	Muraro et al. 2016
usoskin	Mouse brain	622	4	GSE59739	Usoskin et al. 2015

Figure 4. Evaluation of clustering performance using ten datasets with five different methods

(A) Reference for the ten datasets employed for performance evaluation.

(B) Comparison of average silhouette scores for the ten datasets obtained with different algorithms.

in the process of data analysis.[2] Even though numerous imputations or denoising methods have been developed to mitigate the effects of the issues, there are still technical limitations.[3]

In this study, we introduce a novel generative framework called DB-AAE to address the challenges associated with denoising and imputation in scRNA-seq data. This framework leverages the power of AAEs, which combine autoencoders and GANs. While traditional AAEs rely on statistical modeling to generate a latent space that captures expression patterns within scRNA-seq data, the DB-AAE introduces a paradigm shift by employing an adversarial technique. This technique directly samples from the input data to create the latent space, circumventing the limitations of statistical modeling. To evaluate the effectiveness of DB-AAE, we conducted comprehensive testing using both simulated and real datasets. The proposed method was compared to other commonly used approaches such as MAGIC,[25] DCA,[11] scImpute[26] and SCVI,[12] and the analysis demonstrated that DB-AAE outperformed other methods in terms of denoising accuracy and the preservation of biological signal. Additionally, the results showed that DB-AAE significantly improved the accuracy of other algorithms designed for pseudo-time inference, including Slingshot. These findings not only validate the effectiveness of the proposed approach but also highlight its potential as a valuable tool for enhancing the quality and reliability of downstream analyses in scRNA-seq research.

Throughout this study, we show that generative adversarial methods based on deep learning neural networks provide a promising alternative to existing methods. This approach also preserves important features such as cell type-specific gene expression patterns and robustness to noise in scRNA-seq data. DB-AAE can improve the reliability of downstream analyses such as clustering and pseudo-time inference by minimizing information loss during analysis. The DB-AAE framework can be integrated with other existing single-cell sequencing analysis methods to create more comprehensive pipelines. For example, combining DB-AAE with existing clustering algorithms, dimensionality reduction techniques, or trajectory inference methods could lead to more robust and accurate downstream analyses. In addition, optimization techniques can be explored to enhance the training process and convergence of the DB-AAE framework. Techniques like advanced regularization methods, different loss functions, or learning rate can be investigated to improve the stability and efficiency of the model. Additionally, incorporating techniques such as pre-training on related datasets could be explored to leverage prior knowledge and improve performance on specific datasets.

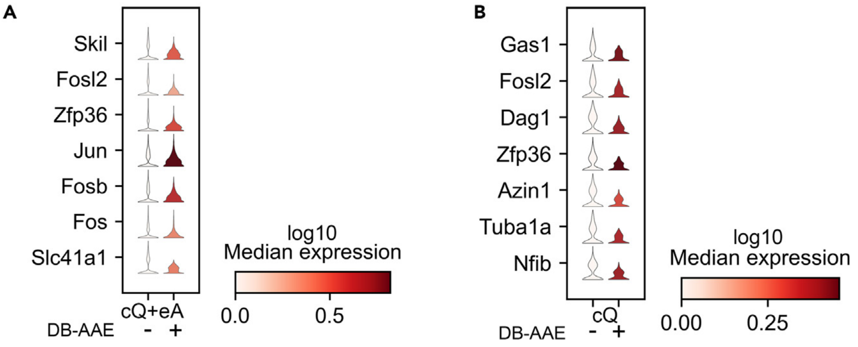


Figure 5. Impact of denoising by DB-AAE in gene-expressing patterns

(A) Recovery of gene transcripts in close-to-quiescent and early activated (cQ + eA) MuSCs before and after denoising with DB-AAE.

(B) Recovery of gene transcripts in cQ MuSCs before and after denoising with DB-AAE.

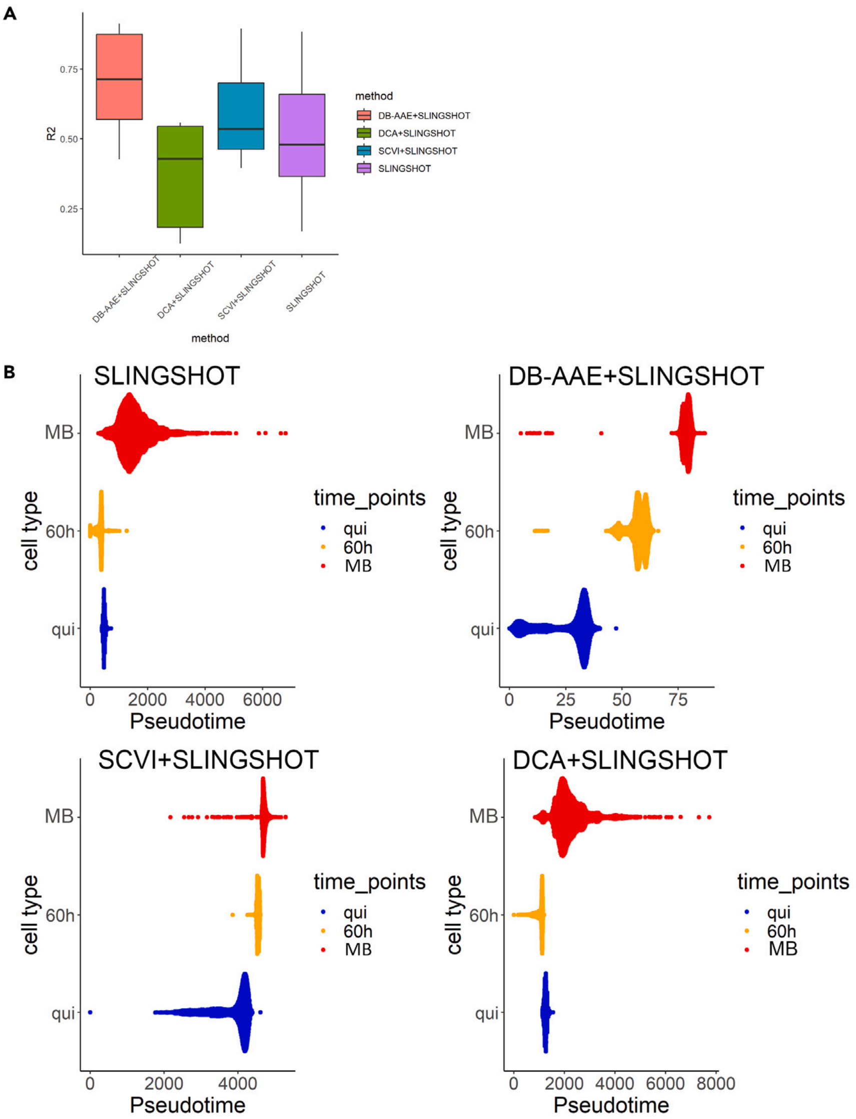


Figure 6. Impact of denoising by DB-AAE in pseudo-time inference

(A) Comparison of average squared R scores for five datasets after processing with four different combined algorithms.

(B) Pseudo-time ordering of homeostatic MuSCs (quiescent, qui), activated MuSCs (60 h after injury) and proliferating myoblasts (MB) after data processing with four different combined algorithms.

Limitations of the study

Since DB-AAE are implemented on a deep-learning model, our study is limited to provide more detailed insights into the precise acquisition and utilization of specific features or gene expression patterns by the model. This is due to the intricate nature of deep learning models, composed of numerous layers with complex interactions between nodes. As a result, internal workings or processes are not easily understandable or interpretable. Although the DB-AAE framework has demonstrated effectiveness on both simulated and real datasets used in the study, further evaluation of additional datasets from different tissues, organisms, or experimental conditions is required. In addition, since the performance of the DB-AAE framework is sensitive to hyperparameters,[56] a comprehensive hyperparameter tuning would be necessary to ensure the stability and robustness of the method.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2024.109027.

ACKNOWLEDGMENTS

This work was supported by the NIAMS IRP through NIH grants AR041126 and AR041164 to V.S. and utilized computational resources of the NIH HPC Biowulf cluster (http://hpc.nih.gov). We thank Drs. Hong-Wei Sun and Seol Kyoung Jung (NIAMS Biodata Mining and Discovery Section) for critical reading of the manuscript and helpful discussions.

AUTHOR CONTRIBUTIONS

K.D.K. conceived the project and performed the analyses. K.D.K. and V.S. interpreted the results. K.D.K. and V.S. wrote the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: September 7, 2023
Revised: November 20, 2023
Accepted: January 22, 2024
Published: January 30, 2024

REFERENCES

1. Hwang, B., Lee, J.H., and Bang, D. (2018). Single-cell RNA sequencing technologies and bioinformatics pipelines. Exp. Mol. Med. *50*, 1–14. https://doi.org/10.1038/s12276-018-0071-8.
2. Kharchenko, P.V., Silberstein, L., and Scadden, D.T. (2014). Bayesian approach to single-cell differential expression analysis. Nat. Methods *11*, 740–742. https://doi.org/10.1038/nmeth.2967.
3. Wang, M., Gan, J., Han, C., Guo, Y., Chen, K., Shi, Y.z., and Zhang, B.G. (2022). Imputation Methods for scRNA Sequencing Data. Appl. Sci. *12*, 10684.
4. Mongia, A., Sengupta, D., and Majumdar, A. (2019). McImpute: Matrix Completion Based Imputation for Single Cell RNA-seq Data. Front. Genet. *10*, 9. https://doi.org/10.3389/fgene.2019.00009.
5. Wagner, F., Yan, Y., and Yanai, I. (2017). K-nearest neighbor smoothing for high-throughput single-cell RNA-Seq data. Preprint at bioRxiv. https://doi.org/10.1101/217737.
6. Geddes, T.A., Kim, T., Nan, L., Burchfield, J.G., Yang, J.Y.H., Tao, D., and Yang, P. (2019). Autoencoder-based cluster ensembles for single-cell RNA-seq data analysis. BMC Bioinf. *20*, 660. https://doi.org/10.1186/s12859-019-3179-5.
7. Franco, E.F., Rana, P., Cruz, A., Calderón, V.V., Azevedo, V., Ramos, R.T.J., and Ghosh, P. (2021). Performance Comparison of Deep Learning Autoencoders for Cancer Subtype Detection Using Multi-Omics Data. Cancers *13*, 2013. https://doi.org/10.3390/cancers13092013.
8. Rao, J., Zhou, X., Lu, Y., Zhao, H., and Yang, Y. (2021). Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. iScience *24*, 102393. https://doi.org/10.1016/j.isci.2021.102393.
9. Tian, T., Min, M.R., and Wei, Z. (2021). Model-based autoencoders for imputing discrete single-cell RNA-seq data. Methods *192*, 112–119. https://doi.org/10.1016/j.ymeth.2020.09.010.
10. Brendel, M., Su, C., Bai, Z., Zhang, H., Elemento, O., and Wang, F. (2022). Application of Deep Learning on Single-cell RNA Sequencing Data Analysis: A Review. Dev. Reprod. Biol. *20*, 814–835. https://doi.org/10.1016/j.gpb.2022.11.011.
11. Eraslan, G., Simon, L.M., Mircea, M., Mueller, N.S., and Theis, F.J. (2019). Single-cell RNA-seq denoising using a deep count autoencoder. Nat. Commun. *10*, 390. https://doi.org/10.1038/s41467-018-07931-2.
12. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. Nat. Methods *15*, 1053–1058. https://doi.org/10.1038/s41592-018-0229-2.
13. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial Autoencoders. Preprint at arXiv. https://doi.org/10.48550/arXiv.1511.05644.
14. Wang, X., Hu, Z., Yu, T., Wang, Y., Wang, R., Wei, Y., Shu, J., Ma, J., and Li, Y. (2023). Con-AAE: contrastive cycle adversarial autoencoders for single-cell multi-omics alignment and integration. Bioinformatics *39*, btad162. https://doi.org/10.1093/bioinformatics/btad162.
15. Wang, H.Y., Zhao, J.P., Zheng, C.H., and Su, Y.S. (2023). scGMAAE: Gaussian mixture adversarial autoencoders for diversification analysis of scRNA-seq data. Brief. Bioinform. *24*, bbac585. https://doi.org/10.1093/bib/bbac585.
16. Takase, T. (2021). Dynamic batch size tuning based on stopping criterion for neural network training. Neurocomputing *429*, 1–11.
17. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P.T.P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. Preprint at arXiv. https://doi.org/10.48550/arXiv.1609.0483.
18. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. Preprint at arXiv. https://doi.org/10.48550/arXiv.1706.02677.
19. Smith, S.L., Kindermans, P.-J., Ying, C., and Le, Q.V. (2017). Don't decay the learning rate, increase the batch size. Preprint at arXiv. https://doi.org/10.48550/arXiv.1711.00489.
20. Takase, T., Oyama, S., and Kurihara, M. (2018). Why Does Large Batch Training Result in Poor Generalization? A Comprehensive Explanation and a Better Strategy from the Viewpoint of Stochastic Optimization. Neural Comput. *30*, 2005–2023. https://doi.org/10.1162/neco_a_01089.

21. Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. J. Mach. Learn. Res. *13*, 281–305.
22. Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. (2015). Scalable bayesian optimization using deep neural networks, pp. 2171–2180.
23. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. J. Mach. Learn. Res. *18*, 1–52.
24. Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. Genome Biol. *18*, 174. https://doi.org/10.1186/s13059-017-1305-0.
25. van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A.J., Burdziak, C., Moon, K.R., Chaffer, C.L., Pattabiraman, D., et al. (2018). Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. Cell *174*, 716–729.e27. https://doi.org/10.1016/j.cell.2018.05.061.
26. Li, W.V., and Li, J.J. (2018). An accurate and robust imputation method scImpute for single-cell RNA-seq data. Nat. Commun. *9*, 997. https://doi.org/10.1038/s41467-018-03405-7.
27. Ren, X., Zheng, L., and Zhang, Z. (2019). SSCC: A Novel Computational Framework for Rapid and Accurate Clustering Large-scale Single Cell RNA-seq Data. Dev. Reprod. Biol. *17*, 201–210. https://doi.org/10.1016/j.gpb.2018.10.003.
28. Shahapure, K.R., and Nicholas, C. (2020). Cluster quality analysis using silhouette score. In International Conference on Data Science and Advanced Analytics (IEEE), pp. 747–748. https://doi.org/10.1109/Dsaa49011.2020.00096.
29. Wolf, F.A., Angerer, P., and Theis, F.J. (2018). SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. *19*, 15. https://doi.org/10.1186/s13059-017-1382-0.
30. Baron, M., Veres, A., Wolock, S.L., Faust, A.L., Gaujoux, R., Vetere, A., Ryu, J.H., Wagner, B.K., Shen-Orr, S.S., Klein, A.M., et al. (2016). A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. Cell Syst. *3*, 346–360.e4. https://doi.org/10.1016/j.cels.2016.08.011.
31. Yu, G. (2020). Gene Ontology Semantic Similarity Analysis Using GOSemSim. Methods Mol. Biol. *2117*, 207–215. https://doi.org/10.1007/978-1-0716-0301-7_11.
32. Bartolome, A. (2023). The Pancreatic Beta Cell: Editorial. Biomolecules *13*, 495. https://doi.org/10.3390/biom13030495.
33. Goolam, M., Scialdone, A., Graham, S.J.L., Macaulay, I.C., Jedrusik, A., Hupalowska, A., Voet, T., Marioni, J.C., and Zernicka-Goetz, M. (2016). Heterogeneity in Oct4 and Sox2 Targets Biases Cell Fate in 4-Cell Mouse Embryos. Cell *165*, 61–74. https://doi.org/10.1016/j.cell.2016.01.047.
34. Diaz-Papkovich, A., Anderson-Trocmé, L., and Gravel, S. (2021). A review of UMAP in population genetics. J. Hum. Genet. *66*, 85–91. https://doi.org/10.1038/s10038-020-00851-4.
35. Choi, S.M., Park, H.J., Choi, E.A., Jung, K.C., and Lee, J.I. (2022). Heterogeneity of circulating CD4(+)CD8(+) double-positive T cells characterized by scRNA-seq analysis and trajectory inference. Sci. Rep. *12*, 14111. https://doi.org/10.1038/s41598-022-18340-3.
36. Zeisel, A., Muñoz-Manchado, A.B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science *347*, 1138–1142. https://doi.org/10.1126/science.aaa1934.
37. Xin, Y., Kim, J., Okamoto, H., Ni, M., Wei, Y., Adler, C., Murphy, A.J., Yancopoulos, G.D., Lin, C., and Gromada, J. (2016). RNA Sequencing of Single Human Islet Cells Reveals Type 2 Diabetes Genes. Cell Metab. *24*, 608–615. https://doi.org/10.1016/j.cmet.2016.08.018.
38. Tran, D., Nguyen, H., Tran, B., La Vecchia, C., Luu, H.N., and Nguyen, T. (2021). Fast and precise single-cell data analysis using a hierarchical autoencoder. Nat. Commun. *12*, 1029. https://doi.org/10.1038/s41467-021-21312-2.
39. Deng, Q., Ramsköld, D., Reinius, B., and Sandberg, R. (2014). Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. Science *343*, 193–196. https://doi.org/10.1126/science.1245316.
40. Wang, Y.J., Schug, J., Won, K.J., Liu, C., Naji, A., Avrahami, D., Golson, M.L., and Kaestner, K.H. (2016). Single-Cell Transcriptomics of the Human Endocrine Pancreas. Diabetes *65*, 3028–3038. https://doi.org/10.2337/db16-0405.
41. Muraro, M.J., Dharmadhikari, G., Grün, D., Groen, N., Dielen, T., Jansen, E., van Gurp, L., Engelse, M.A., Carlotti, F., de Koning, E.J.P., and van Oudenaarden, A. (2016). A Single-Cell Transcriptome Atlas of the Human Pancreas. Cell Syst. *3*, 385–394.e3. https://doi.org/10.1016/j.cels.2016.09.002.
42. Usoskin, D., Furlan, A., Islam, S., Abdo, H., Lönnerberg, P., Lou, D., Hjerling-Leffler, J., Haeggström, J., Kharchenko, O., Kharchenko, P.V., et al. (2015). Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. Nat. Neurosci. *18*, 145–153. https://doi.org/10.1038/nn.3881.
43. Dell'Orso, S., Juan, A.H., Ko, K.D., Naz, F., Perovanovic, J., Gutierrez-Cruz, G., Feng, X., and Sartorelli, V. (2019). Single cell analysis of adult mouse skeletal muscle stem cells in homeostatic and regenerative conditions. Development *146*, dev174177. https://doi.org/10.1242/dev.174177.
44. Bastidas-Ponce, A., Tritschler, S., Dony, L., Scheibner, K., Tarquis-Medina, M., Salinno, C., Schirge, S., Burtscher, I., Bottcher, A., Theis, F.J., et al. (2019). Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. Development *146*, dev173849. https://doi.org/10.1242/dev.173849.
45. Treutlein, B., Lee, Q.Y., Camp, J.G., Mall, M., Koh, W., Shariati, S.A.M., Sim, S., Neff, N.F., Skotheim, J.M., Wernig, M., and Quake, S.R. (2016). Dissecting direct reprogramming from fibroblast to neuron using single-cell RNA-seq. Nature *534*, 391–395. https://doi.org/10.1038/nature18323.
46. Qiu, W.L., Zhang, Y.W., Feng, Y., Li, L.C., Yang, L., and Xu, C.R. (2017). Deciphering Pancreatic Islet beta Cell and alpha Cell Maturation Pathways and Characteristic Features at the Single-Cell Level. Cell Metab. *25*, 1194–1205.e4. https://doi.org/10.1016/j.cmet.2017.04.003.
47. Yuzwa, S.A., Borrett, M.J., Innes, B.T., Voronova, A., Ketela, T., Kaplan, D.R., Bader, G.D., and Miller, F.D. (2017). Developmental Emergence of Adult Neural Stem Cells as Revealed by Single-Cell Transcriptional Profiling. Cell Rep. *21*, 3970–3986. https://doi.org/10.1016/j.celrep.2017.12.017.
48. Vladoiu, M.C., El-Hamamy, I., Donovan, L.K., Farooq, H., Holgado, B.L., Sundaravadanam, Y., Ramaswamy, V., Hendrikse, L.D., Kumar, S., Mack, S.C., et al. (2019). Childhood cerebellar tumours mirror conserved fetal transcriptional programs. Nature *572*, 67–73. https://doi.org/10.1038/s41586-019-1158-7.
49. Rocheteau, P., Gayraud-Morel, B., Siegl-Cachedenier, I., Blasco, M.A., and Tajbakhsh, S. (2012). A subpopulation of adult skeletal muscle stem cells retains all template DNA strands after cell division. Cell *148*, 112–125. https://doi.org/10.1016/j.cell.2011.11.049.
50. van Velthoven, C.T.J., de Morree, A., Egner, I.M., Brett, J.O., and Rando, T.A. (2017). Transcriptional Profiling of Quiescent Muscle Stem Cells In Vivo. Cell Rep. *21*, 1994–2004. https://doi.org/10.1016/j.celrep.2017.10.037.
51. Machado, L., Esteves de Lima, J., Fabre, O., Proux, C., Legendre, R., Szegedi, A., Varet, H., Ingerslev, L.R., Barrès, R., Relaix, F., and Mourikis, P. (2017). In Situ Fixation Redefines Quiescence and Early Activation of Skeletal Muscle Stem Cells. Cell Rep. *21*, 1982–1993. https://doi.org/10.1016/j.celrep.2017.10.080.
52. García-Prat, L., Martínez-Vicente, M., Perdiguero, E., Ortet, L., Rodríguez-Ubreva, J., Rebollo, E., Ruiz-Bonilla, V., Gutarra, S., Ballestar, E., Serrano, A.L., et al. (2016). Autophagy maintains stemness by preventing senescence. Nature *529*, 37–42. https://doi.org/10.1038/nature16187.
53. Saelens, W., Cannoodt, R., Todorov, H., and Saeys, Y. (2019). A comparison of single-cell trajectory inference methods. Nat. Biotechnol. *37*, 547–554. https://doi.org/10.1038/s41587-019-0071-9.
54. Street, K., Risso, D., Fletcher, R.B., Das, D., Ngai, J., Yosef, N., Purdom, E., and Dudoit, S. (2018). Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genom. *19*, 477. https://doi.org/10.1186/s12864-018-4772-0.
55. Harrell, F.E. (2015). General aspects of fitting regression models. In Springer Series in Statistics (Springer), pp. 13–44. https://doi.org/10.1007/978-3-319-19425-7_2.
56. Goodfellow, I., Bengio, Y., and Courville, A. (2016). Regularization for deep learning. In Adaptive Computation and Machine Learning Series (↱MIT Press), pp. 221–265.
57. Li, W.V., and Li, J.J. (2018). An accurate and robust imputation method scImpute for single-cell RNA-seq data. Nat. Commun. *9*, 997.
58. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W.M., 3rd, Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive Integration of Single-Cell Data. Cell *177*, 1888–1902.e21. https://doi.org/10.1016/j.cell.2019.05.031.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Code for development and evaluation	This paper	https://github.com/LMSCGR/DB-AAE
Software and algorithms		
Scanpy	Wolf et al.[29]	https://github.com/scverse/scanpy
MAGIC	van Dijk et al.[25]	https://github.com/pkathail/magic
DCA	Eraslan et al.[11]	https://github.com/theislab/dca
scImpute	Li and Li,[57]	https://github.com/Vivianstats/scImpute
SCVI	Lopez et al.[12]	https://github.com/scverse/scvi-tools
Slingshot	Street et al.[54]	https://github.com/kstreet13/slingshot
R	The R Project for Statistical Computing	https://www.r-project.org/
Python	Python Software Foundation	https://www.python.org/downloads/source/
Custom scripts	This paper	https://doi.org/10.5281/zenodo.10478925

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed and will be fulfilled by the lead contact, Vittorio Sartorelli (vittorio.sartorelli@nih.gov).

Materials availability

This study did not generate new unique reagents.

Data and code availability

This paper analyzes existing publicly available data. The accession numbers of the datasets employed in this study are listed in Table 1.

All original codes have been deposited at GitHub (https://github.com/LMSCGR/DB-AAE) and are publicly available as of the date of publication.

Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Data preprocessing

Table 1 describes 16 single-cell datasets (clustering efficiency:10 and pseudo time inference:6) used in data analysis. Some datasets[30,36,37,39–48] were downloaded from Gene Expression Omnibus (GEO). Other datasets[33,38] were downloaded from Broad Institute Single Cell Portal. After removing cells with ambiguous labels in the datasets, we converted the datasets into the standard h5 anndata format for training DB-AAE and evaluation of the performance.

Dynamic batching adversarial autoencoder

We designed a modified version of AAE (AAE) to mitigate losing information during training procedure. Traditional AAE consists of three key components: encoder, decoder, and adversary modules (Figure 1). The encoder usually transforms input data into a lower-dimensional latent space, and the input of the encoder is normalized gene expression profile using highly variable genes annotated by dispersion-based method.[58] First, after transforming the gene expressions to z-scores using Equation 1, we calculated the normalized variance of each gene and ranked the genes by the normalized variances. Finally, we selected genes as the input of encoder with high variances using pre-processing module in Figure 1.

$$z_{mn} = \frac{G_{mn} - \tilde{G}_m}{\sigma_m} \qquad \text{(Equation 1)}$$

where $\boldsymbol{z_{mn}}$ is Z score of gene m in cell n, $\boldsymbol{G_{mn}}$ is expressing value of gene m in cell n, $\tilde{\boldsymbol{G}}_{\boldsymbol{m}}$ is mean expressing value of gene m, $\boldsymbol{\sigma_m}$ is the expected standard deviation of feature m derived from the global mean-variance.

Subsequently, we implemented dynamic batching procedures utilizing the Hyperband algorithm.[23] As illustrated in Figure 1, we initialize the starting, ending, and increment values for the batching size within the optimizing batch size module. In each cycle, the input data are segmented into batches using predetermined step-ups from the optimizing module. The encoder generates the authentic latent space using the current input batch, while the generator creates a simulated latent space by emulating the characteristics of the output through the autoencoder with the prior input batch. The training process involves continuous iterations, with the autoencoder and discriminator refining their models until the discriminator can no longer differentiate between the true and simulated latent spaces across the entire batch.

Defining a selected gene expressing profile of cell m as input x, the architecture of AAE can be formulated as follows:

$$\begin{aligned} x_b &= S(x) \\ z_b &= l_e(x_b) \\ a_{loss} &= D(z_b, n_r) \\ x'_b &= l_d(z_b) \\ x' &= \bigcup_{i=0}^{n} x'_{bi} \end{aligned} \qquad \text{(Equation 2)}$$

where x_b is an input batched from input data x, z_b is the latent representation from the batched input x_b, $\boldsymbol{n_r}$ is the latent representation from $\boldsymbol{x'_b}$, $\boldsymbol{x'_b}$ is reconstructed input from x_b, $\boldsymbol{a_{loss}}$ is adversarial loss, S is batch sampling function, l_e is encoder layer, D is discriminator layer, l_d is decoder layer, U is union of $\boldsymbol{x'_b}$, $\boldsymbol{x'}$ is reconstructed output from batched inputs, n is the number of batches.

The encoder layer is defined below:

$$l_e = \text{LeakReLU}(XW_e) \qquad \text{(Equation 3)}$$

where X represents input, W_e represents weight values in encoder layer.

The decoder layer is defined below:

$$l_d = \text{LeakReLU}(ZW_d) \qquad \text{(Equation 4)}$$

where Z represents latent matrix, W_d represents weight values in the decoder layer.

To complete adversarial training, DB-AAE uses a discriminator network to distinguish between the true latent space using the current input batch and a synthetic latent space by mimicking the features of the output generated in the preceding input batch to minimize reconstruction (autoencoder) and generator loss, while maximizing the discriminator loss.

For reconstruction loss, we used binary cross-entropy between batched input x and reconstructed output x' below:

$$L_{rec} = -\frac{1}{N}\sum_{i=1}^{N}\left(x_i\, log(x'_i) + (1 - x_i)log(1 - x'_i)\right) \qquad \text{(Equation 5)}$$

The generator loss is defined below:

$$L_{gen} = -\frac{1}{N}\sum_{i=1}^{N} log(1 - D(l_e(x'_i))) \qquad \text{(Equation 6)}$$

The discriminator loss is defined below:

$$L_{disc} = -\frac{1}{N}\sum_{i=1}^{N}(log(D(l_e(x_i)) + log(1 - D(l_e(x'_i))) \qquad \text{(Equation 7)}$$

where N is batch size, D(x) represents the output of the discriminator.

Through these formulas, the autoencoder, generator, and discriminator are updated iteratively until DB-AAE discovers a balance between the reconstruction capability and the ability to generate realistic encoded samples.

After each cycle, the Batching Module stores the current batch size, accuracy, and the minimum loss function values of the autoencoder. A new batch size is then initialized for the subsequent cycle. This process is repeated until the batch size reaches its maximum value in the predetermined step-ups. The batch size associated with high accuracy and low minimum loss is selected from the results of all cycles. Using the chosen batch size, the DB-AAE performs a final training cycle to construct an optimal denoising model.

Hyperparameters

The encoder network dimensions are set to input-1024-512-512, where input stands for the dimension of input data, and the decoder has a symmetric structure with the encoder. In addition, the discriminator network is built with dimensions 512-256-1. the activation function of the last layer of encoder, decoder, and discriminator is relu, while fully connected layers are all activated by LeakyReLU. In the training stage, we utilize the optimizer RMSprop with learning rate 0.00002 for all the datasets.

Measurement of performance with other algorithms

Software and algorithms used for to evaluate the performance of DB-AAE are cited in the appropriate sections in the STAR methods. For the evaluation, we used silhouette score and *r* (unstandardized Pearson's correlation).[2] Silhouette score for clustering performance is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample, and the formula is defined below:

$$s_i = \frac{b_i - a_i}{max(b_i, a_i)} \quad \text{(Equation 8)}$$

,b_i is the inter cluster distance defined as the average distance to closest cluster of data point I except for that it's a part of

$$b_i = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \quad \text{(Equation 9)}$$

, and a_i is the intra cluster distance defined as the average distance to all other points in the cluster to which it's a part of

$$a_i = \frac{1}{|c_i| - 1} \sum_{j \in c_i, i \neq j} d(i, j) \quad \text{(Equation 10)}$$

The value of silhouette score is between −1 and 1, and the value close to 1 means the clusters are well-defined and well-separated from each other. In addition, it shows that the data points within each cluster are more similar to each other than to points in other clusters.

r^2 for the accuracy of pseudo time inference is calculated using Fit-regression model defined below:

$$\hat{y} = a + bx, b = r\left(\frac{s_y}{s_x}\right) and\ a = \overline{y} - b(\overline{x}) \quad \text{(Equation 11)}$$

where s_y, s_x is standard deviations of *y and x*, $\overline{y}$, $\overline{x}$ are means of *y and x*, *r* is unstandardized Pearson's correlation. The high value of r^2 indicates that actual predicted pseudo times are close to target or reference timepoints.