Author/s:
Li, H;McCarthy, DJ;Shim, H;Wei, S

Title:
Trade-off between conservation of biological variation and batch effect removal in deep generative modeling for single-cell transcriptomics

Date:
2022-12-01

Citation:
Li, H., McCarthy, D. J., Shim, H. & Wei, S. (2022). Trade-off between conservation of biological variation and batch effect removal in deep generative modeling for single-cell transcriptomics. BMC Bioinformatics, 23 (1), https://doi.org/10.1186/s12859-022-05003-3.

Persistent Link:
https://hdl.handle.net/11343/327147

BMC Bioinformatics

RESEARCH

Open Access

Trade-off between conservation of biological variation and batch effect removal in deep generative modeling for single-cell transcriptomics

Hui Li[1], Davis J. McCarthy[1,2,3], Heejung Shim[1,3] and Susan Wei[1*]

*Correspondence:
susan.wei@unimelb.edu.au

[1] School of Mathematics and Statistics, University of Melbourne, Melbourne, VIC 3010, Australia
[2] Bioinformatics and Cellular Genomics, St Vincent's Institute of Medical Research, Melbourne, VIC 3065, Australia
[3] Melbourne Integrative Genomics, University of Melbourne, Melbourne, VIC 3010, Australia

Abstract

Background: Single-cell RNA sequencing (scRNA-seq) technology has contributed significantly to diverse research areas in biology, from cancer to development. Since scRNA-seq data is high-dimensional, a common strategy is to learn low-dimensional latent representations better to understand overall structure in the data. In this work, we build upon scVI, a powerful deep generative model which can learn biologically meaningful latent representations, but which has limited explicit control of batch effects. Rather than prioritizing batch effect removal over conservation of biological variation, or vice versa, our goal is to provide a bird's eye view of the trade-offs between these two conflicting objectives. Specifically, using the well established concept of Pareto front from economics and engineering, we seek to learn the entire trade-off curve between conservation of biological variation and removal of batch effects.

Results: A multi-objective optimisation technique known as Pareto multi-task learning (Pareto MTL) is used to obtain the Pareto front between conservation of biological variation and batch effect removal. Our results indicate Pareto MTL can obtain a better Pareto front than the naive scalarization approach typically encountered in the literature. In addition, we propose to measure batch effect by applying a neural-network based estimator called Mutual Information Neural Estimation (MINE) and show benefits over the more standard maximum mean discrepancy measure.

Conclusion: The Pareto front between conservation of biological variation and batch effect removal is a valuable tool for researchers in computational biology. Our results demonstrate the efficacy of applying Pareto MTL to estimate the Pareto front in conjunction with applying MINE to measure the batch effect.

Keywords: Conservation of biological variation, Batch effect, MINE, MMD, Pareto front, Pareto MTL, ScRNA-seq, ScVI

Background

Single-cell RNA sequencing (scRNA-seq) measures gene expression at single-cell resolution, allowing for the study of cell types, state, and trajectories to better understand heterogeneous cell states and dynamics in tissues, organs, and organism development. Since scRNA-seq data is high-dimensional and large-scale (e.g., gene expression of tens of thousands of genes for hundreds of thousands of cells or more), common analysis techniques often require discovering low-dimensional latent representations which capture underlying gene expression patterns in the high-dimensional data. Among widely used dimension reduction techniques (e.g., PCA [1], ZIFA [2], t-SNE [3], UMAP [4], PHATE [5]), methods based on deep neural network such as scVI [6] and SAUCIE [7] have emerged as powerful tools as they can be efficiently applied to the large-scale data.

While SAUCIE and scVI attempt to account for batch effects when learning the latent representations to ensure that the learned representations capture biological variations, these methods are not designed to learn the complex trade-off between conserving biological variation and removing batch effects. In this paper, building upon scVI, we aim to learn the trade-offs between these two conflicting objectives rather than prioritizing one over the other. Specifically, we borrow the concept of Pareto front from economics and engineering to construct the trade-off curve. We then use Pareto multi-task learning method [8] to estimate the Pareto front. Along the way, we introduce a new batch effect measure based on deep neural networks.

The rest of this section is as follows. We begin by reviewing the ZINB model for scRNA-seq data and the deep generative model proposed in scVI [6]. Then, we describe the concept of the Pareto front and summarise our contribution.

ZINB model

The data resulting from an scRNA-seq experiment can be represented by an $n \times G$ matrix, $\boldsymbol{x}$, where each entry $\boldsymbol{x}_i^g$ records the expression level measured for cell i and gene g. For each cell i, we observe a batch annotation $\boldsymbol{s}_i$.[1] As the scRNA-seq data exhibits overdispersion and a large proportion of observed zeros, the zero-inflated negative binomial (ZINB) distribution [6] is commonly employed to model it. The ZINB model is a combination of negative binomial distribution for the expression count and logit distribution for the excessive zeros relative to the negative binomial distribution, perhaps due to failure in the assay reliably to capture information from genes with low expression. Specifically, conditional on batch $\boldsymbol{s}_i$ and two latent variables $\boldsymbol{z}_i$ and l_i, we model $\boldsymbol{x}_i^g$ using the ZINB distribution. The latent variable $\boldsymbol{z}_i$, a low-dimensional vector, potentially represents biological variation; its prior $p(\boldsymbol{z})$ is $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The latent variable l_i represents log-library size; its prior $p(l)$ is $\mathcal{N}(l_\mu, l_\sigma)$, where l_μ, l_σ are set to be the empirical mean and variance of log-library size per batch. Putting this together leads to the ZINB model:

[1] We represent batch $\boldsymbol{s}_i$ using dummy encoding, e.g., for B batches, $\boldsymbol{s}_i \in \{0, 1\}^B$.

$$
\begin{aligned}
\boldsymbol{z}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
l_i &\sim \mathcal{N}(l_\mu, l_\sigma) \\
\boldsymbol{\epsilon}_i &= f_{\boldsymbol{\theta}_1}(\boldsymbol{z}_i, \boldsymbol{s}_i) \\
\boldsymbol{w}_i^g &\sim \text{Gamma}(\boldsymbol{\epsilon}_i^g, \boldsymbol{c}^g) \\
\boldsymbol{y}_i^g &\sim \text{Poisson}((\exp l_i)\boldsymbol{w}_i^g) \\
\boldsymbol{h}_i^g &\sim \text{Bernoulli}(f_{\boldsymbol{\theta}_2}^g(\boldsymbol{z}_i, \boldsymbol{s}_i)) \\
\boldsymbol{x}_i^g &= \begin{cases} \boldsymbol{y}_i^g & \text{if} \quad \boldsymbol{h}_i^g = 0 \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
\tag{1}
$$

Above $f_{\boldsymbol{\theta}_1}$ and $f_{\boldsymbol{\theta}_2}$ are **decoder neural networks**. We shall denote by $\boldsymbol{\theta}$ the concatenation of θ_1 and θ_2. The gene-specific inverse dispersion is denoted $\boldsymbol{c} \in \mathbb{R}_+^G$.[2] As with $\boldsymbol{x}_i^g$, we use superscript notation to refer to a specific gene g. The notation $\boldsymbol{\epsilon}_i^g$ means the proportion of gene g expression in the whole expression of cell i, $\boldsymbol{w}_i^g$ is the gene g's expression proportion in cell i after gene inverse dispersion adjustment, $\boldsymbol{y}_i^g$ is the expression count of gene g in cell i from negative binomial distribution and $\boldsymbol{h}_i^g$ is the drop-out rate for gene g in cell i (thus $\boldsymbol{x}_i^g$ is zero-inflated negative binomial).

Variational inference

The posterior distribution $p(\boldsymbol{z}, l|\boldsymbol{x})$[3] is unfortunately intractable. While we could employ MCMC to approximate the posterior, we shall instead consider the fast alternative of variational inference (VI). There are two main ingredients to VI: 1) an approximating family $\mathcal{Q}$, and 2) a criterion for determining the best member $q \in \mathcal{Q}$. For the former, we turn to a mean-field variational family whereby each $q \in \mathcal{Q}$ is stipulated to factor across the latent variables:

$$q(\boldsymbol{z}, l|\boldsymbol{x}) = q(\boldsymbol{z}|\boldsymbol{x})q(l|\boldsymbol{x}).$$

The distribution $q(\boldsymbol{z}|\boldsymbol{x})$ is further chosen to be multivariate Gaussian with diagonal covariance matrix and the distribution $q(l|\boldsymbol{x})$ is chosen to be Gaussian with scalar mean and variance. The mean and variances of $q(\boldsymbol{z}|\boldsymbol{x})$ and $q(l|\boldsymbol{x})$ will be learned using an **encoder neural network** applied to $\boldsymbol{x}$. The collective weights of the encoder neural networks will be denoted by $\boldsymbol{\phi}$.

For the second ingredient of VI, we adopt the conventional Kullback-Leibler (KL) divergence, i.e., we seek to minimize the KL divergence between $q_{\boldsymbol{\phi}}(\boldsymbol{z}, l|\boldsymbol{x})$ and the intractable true posterior $p(\boldsymbol{z}, l|\boldsymbol{x})$. It turns out that minimizing the KL divergence is equivalent to minimizing the negative evidence lower bound (ELBO), which we shall denote by $\mathit{U}_n(\boldsymbol{\phi}, \boldsymbol{\theta})$ (details in The loss function U_n for the scVI generative model).

Controlling batch effect

So far, there is nothing that prevents the learned variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ from outputting latent representations $\boldsymbol{z}$ that are strongly correlated with the batch variable $\boldsymbol{s}$. In this work we set out to characterise the trade-off between learning $\boldsymbol{z}$ that is

[2] The estimation of $\boldsymbol{c} \in \mathbb{R}_+^G$ is described in further detail in [6].

[3] We use a preliminary version of scVI (version 0.3.0 committed on Mar 6, 2019 on Github), where only $\boldsymbol{x}$ is the input for encoder.

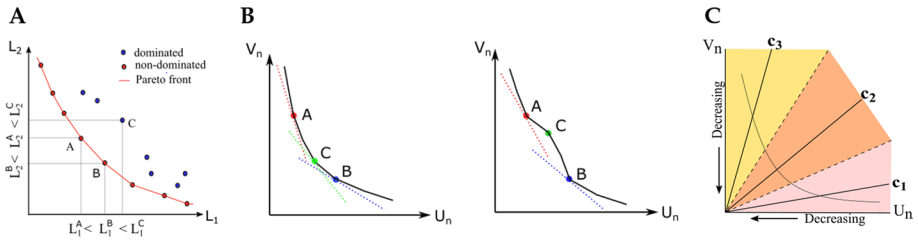

Fig. 1 Panel **A** shows the Pareto front for an example bi-objective minimization problem. Point A and point B are non-dominated points on the Pareto front, while point C is dominated by both A and B. Panel **B** shows example Pareto candidates that can be discovered by the scalarization method for a convex (left) and a non-convex (right) Pareto front. In theory, scalarization cannot recover Pareto candidates in the non-convex part of a non-convex (right) Pareto front, such as Point C [9]. Panel **C** is a schematic of the Pareto MTL method [8], which first decomposes the bi-objective space into subregions according to a set of preference vectors c_k, and then seeks a Pareto candidate within each subregion

biologically meaningful and, simultaneously, disentangled from batch effects. For now, let $V_n(\boldsymbol{\phi})$ be some measure of batch effect in the learned latent variable $\boldsymbol{z}$, where we take the convention that a lower value of V_n is better, i.e., less batch effect. Note that, unlike the generative loss U_n, the batch effect measure V_n does not depend on the decoder parameter $\boldsymbol{\theta}$ because the latent $\boldsymbol{z}$ only depends on the encoder parameter $\boldsymbol{\phi}$.

Consider the bi-objective minimization problem,

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \quad \boldsymbol{L}(\boldsymbol{\phi},\boldsymbol{\theta}) = (U_n(\boldsymbol{\phi},\boldsymbol{\theta}), V_n(\boldsymbol{\phi}))^T. \tag{2}$$

Note that this is a *vector* objective. Associated to this bi-objective problem is the so-called Pareto front:

Definition 1 Suppose we have a general optimisation problem with p objectives:

$$\min_{\boldsymbol{\omega}\in\boldsymbol{\Omega}} \quad \boldsymbol{L}(\boldsymbol{\omega}) = (L_1(\boldsymbol{\omega}), ..., L_p(\boldsymbol{\omega}))^T$$

where $\boldsymbol{\Omega}$ is the parameter space and $L_i : \boldsymbol{\omega} \to \mathbb{R}, i = 1, \ldots, p$. We say $\boldsymbol{\omega} \in \boldsymbol{\Omega}$ is Pareto optimal if and only if it is non-dominated, i.e. there does not exist any $\tilde{\boldsymbol{\omega}} \in \boldsymbol{\Omega}$ such that $\forall i = 1, ..., p, L_i(\tilde{\boldsymbol{\omega}}) \leq L_i(\boldsymbol{\omega})$ with at least one strict inequality. The **Pareto front** is the set of all Pareto optimal points (Fig. 1A).

A naive strategy to obtain the Pareto front is via regularization,

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \quad U_n(\boldsymbol{\phi},\boldsymbol{\theta}) + \lambda V_n(\boldsymbol{\phi}), \tag{3}$$

where $\lambda \in \mathbb{R}$ can be viewed as a penalty factor. Under this strategy, Pareto candidates are generated by sweeping a list of penalty factors. Examples of this approach can be seen in [10] and [7]. In the former, to control batch effect, the generative loss of scVI is penalized by the Hilbert-Schmidt Independence Criterion (HSIC). In the latter, the reconstruction loss of SAUCIE, a sparse autoencoder, is penalized by the Maximum Mean Discrepancy (MMD). We will later refer to these two methods as scVI+HSIC and SAUCIE+MMD respectively.

Now, the regularized objective in (3) is equivalent to the objective

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \quad \lambda U_n(\boldsymbol{\phi},\boldsymbol{\theta}) + (1-\lambda)V_n(\boldsymbol{\phi}), \tag{4}$$

where $\lambda \in [0,1]$. This equivalent formulation can be recognized as the **scalarization scheme** in the multiobjective optimization field. In Fig. 1B, the scalarization approach for a given λ corresponds to one tangential point on the convex part of a Pareto front. If the true Pareto front is convex (left in Fig. 1B), the scalarization approach can in theory produce the full Pareto front (though it will remain challenging to find the proper set of λs). However, if the true Pareto front is non-convex as is often the case (right in Fig. 1B), the scalarization approach is not only inefficient but also unable to recover the Pareto optimal points on the non-convex parts, such as Point C [9].

Contribution

In this research, we first improve upon the naive scalarization approach for estimating the Pareto front associated to (2) by applying the sophisticated Pareto multi-task learning (Pareto MTL) method (Fig. 1C) proposed in [8]. We will see that with Pareto MTL we can find a set of "well-distributed" Pareto optimal points, in contrast to the scalarization approach which is typically only capabale of producing Pareto optimal points at the extremes of the Pareto front. Our second contribution is to propose a new batch effect measure based on the Mutual Information Neural Estimator (MINE) proposed in [11]. MINE leverages the expressiveness of deep neural networks to learn the mutual information (MI) between two variables, which in our case is the MI between the latent $\boldsymbol{z}$ and batch $\boldsymbol{s}$.

Overview of methods

To allow readers to appreciate the results in the upcoming section, we briefly describe here the objectives U_n and V_n and how we performed Pareto front estimation.

The loss function U_n for the scVI generative model

The loss function U_n associated to the scVI generative model [6] arises as follows. Minimizing the KL divergence[4] between the variational distribution and the true posterior,

$$D(q_{\boldsymbol{\phi}}(\boldsymbol{z},l|\boldsymbol{x})||p(\boldsymbol{z},l|\boldsymbol{x})) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z},l|\boldsymbol{x})}(\log q_{\boldsymbol{\phi}}(\boldsymbol{z},l|\boldsymbol{x}) - \log p(\boldsymbol{z},l|\boldsymbol{x})), \tag{5}$$

is equivalent to maximizing the so-called Evidence Lower Bound (ELBO),

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z},l|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z},l,\boldsymbol{s}) - D(q_{\boldsymbol{\phi}}(\boldsymbol{z},l|\boldsymbol{x})||p(\boldsymbol{z},l)), \tag{6}$$

where $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z},l,\boldsymbol{s})$ is the ZINB distribution defined in ZINB model. The prior $p(\boldsymbol{z},l)$ in 6 is assumed to factor, i.e., $p(\boldsymbol{z},l) = p(\boldsymbol{z})p(l)$. Then given a training set $\{(\boldsymbol{x}_i,\boldsymbol{s}_i)\}_{i=1}^n$, define

$$\begin{aligned}\text{ELBO}\,(\boldsymbol{\phi},\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\Big\{ &\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_i,l_i|\boldsymbol{x}_i)} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i,l_i,\boldsymbol{s}_i) - D(q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)||p(\boldsymbol{z}_i)) \\ &- D(q_{\boldsymbol{\phi}}(l_i|\boldsymbol{x}_i)||p(l_i))\Big\}.\end{aligned} \tag{7}$$

[4] KL divergence (D) between any two probabilities P and Q are: $D(P||Q) = \mathbb{E}_P \log(P/Q)$.

Throughout this paper we always employ $U_n(\boldsymbol{\phi}, \boldsymbol{\theta}) = -\,\text{ELBO}\,(\boldsymbol{\phi}, \boldsymbol{\theta})$.

Batch effect measure V_n via MINE or MMD

In the Methods section, we will describe the details for each of the batch effect measures, MINE and MMD. It suffices for now to say that MINE can side-step the challenge of choosing a proper kernel bandwidth as is required in deploying MMD. MINE however incurs an additional computational cost because it has parameters of its own that need to be learned using adversarial training.

Standardization of U_n and V_n

Standardization is an important precursor to the success of multi-objective optimization methods. Whether we use MINE or MMD for V_n, we need to address the challenge posed by the highly imbalanced objectives U_n and V_n. To standardize U_n, we minimize $U_n(\boldsymbol{\phi}, \boldsymbol{\theta})$ using minibatch stochastic gradient descent. Let U_{min} and U_{max} denote, respectively, the minimum and maximum value of generative loss observed across the minibatches over all epochs. Similarly, to standardize V_n, we first record V_{min} and V_{max}. The standardized objectives are then given by

$$\overline{U}_n(\boldsymbol{\phi}, \boldsymbol{\theta}) = \frac{U_n(\boldsymbol{\phi}, \boldsymbol{\theta}) - U_{min}}{U_{max} - U_{min}} \quad \text{and} \quad \overline{V}_n(\boldsymbol{\phi}) = \frac{V_n(\boldsymbol{\phi}) - V_{min}}{V_{max} - V_{min}}.$$

It must be noted that because their goal is not the explicit estimation of the Pareto front, neither scVI+HSIC nor SAUCIE+MMD perform standardization of U_n and V_n before performing the optimization in (3). But as our objective is to estimate the Pareto front, we must standardize each of U_n and V_n.

Pareto front estimation

In all experiments in this work, we solve $K = 10$ subproblems for either Pareto MTL or scalarization, which produce 10 Pareto candidates. As we briefly outlined in the Background section, obtaining the Pareto front via the scalarization scheme is straightforward, entailing only a sweep of various $\lambda \in [0, 1]$ in Equation (4). Throughout, we employ $\lambda \in \{1/(K+1), \ldots, K/(K+1)\}$ for the scalarization scheme. We shall compare the naive scalarization approach to the more sophisticated Pareto MTL method (see Section Pareto MTL) for estimating the conservation of biological variation and batch effect removal Pareto front. We shall see that Pareto MTL produces Pareto candidates that are distributed in distinct regions of the trade-off curve rather than lumped in the extremes.

To obtain a complete Pareto front, we must add the two extreme points of the Pareto front corresponding to when only $U_n(\boldsymbol{\phi}, \boldsymbol{\theta})$ is minimized (i.e. scVI) and when only $V_n(\boldsymbol{\phi})$ is minimized. In total, we have 12 Pareto candidates. Note that when the U_n and V_n objectives are optimized on their own, no imbalance issue arises and hence no standardization is required. Finally, it is worth mentioning that in contrast to minimizing $U_n(\boldsymbol{\phi}, \boldsymbol{\theta})$, when $V_n(\boldsymbol{\phi})$ is alone minimized, the output is simply some $\boldsymbol{\phi}_T$. To obtain the corresponding $\boldsymbol{\theta}_T$, we then minimize $U_n(\boldsymbol{\phi}_T, \boldsymbol{\theta})$ over θ.

Results

For the bi-objective minimization problem in (2), we fix U_n to be the loss function associated to the scVI generative model, while allowing for two possible batch effect measures V_n. We will also consider two Pareto front estimation techniques. This makes for a total of four settings: 1) MINE or MMD for V_n and 2) Pareto MTL or scalarization for Pareto front estimation. We begin by presenting the results of the best combination which is Pareto MTL with MINE.

Pareto MTL with MINE

We first demonstrate that Pareto MTL with MINE can estimate a well-distributed Pareto front in the $(\overline{U}_n, \overline{V}_n)$ space. In Fig. 2A, we plot the 12 Pareto candidates of Pareto MTL for the Tabula Muris Marrow (TM-MARROW) dataset [12, 13], a single cell transcriptome dataset from the model organism Mus musculus (see Single cell RNA-seq datasets). Ideally all 12 Pareto candidates should be non-dominated and appear on the Pareto front (see Definition 1). However, since stochastic optimization is not exact, we obtain dominated points. This explains why we display both the Pareto candidates along with the "culled" set of non-dominated points. We use point size to indicate the extent to which the generative loss is minimized. Points of smaller size stand for smaller generative loss minimization during training (i.e. preference vector closer to x-axis). The smallest and largest point size correspond to the extreme points. In particular, the point with the largest marker size corresponds to scVI alone. An analogous figure for the Macaque retina dataset (MACAQUE-RETINA) (see Single cell RNA-seq datasets) can be found in Fig S1 of Additional file 1.

An important caveat is that there is no objective sense in which one Pareto optimal point is "more optimal" than another Pareto optimal point. All points on the Pareto curve are Pareto optimal. In plain speak, this means that no individual objective can be made better without making the other objectives suffer. Thus, the choice of a single Pareto optimal point from the Pareto curve is an entirely subjective matter. Although it may appear that in the traditional regularization/scalarization method an "optimal" trade-off point is determined by choosing the λ that gives the smallest objective loss (3 or equivalently 4) across a list of λs, this so-called "optimal" point is actually just a single trade-off point in the Pareto front.

The trade-off points in the $(\overline{U}_n, \overline{V}_n)$ space have biological meaning. In Fig. 2B–D, we plot the latent $\boldsymbol{z}$ in a two dimensional plane via t-SNE method [3] for the third, sixth and tenth candidate on the TM-MARROW test dataset, respectively. As expected, from the third (Fig. 2B), to the sixth (Fig. 2C) to the tenth candidate (Fig. 2D), more biological variation is conserved at the cost of more batch effect. The result is consistent with the trade-offs in the left plot of Fig. 2A, where batch effect measure $\overline{V}_n$ is increasing (i.e.

(See figure on next page.)

Fig. 2 Pareto front in $(\overline{U}_n, \overline{V}_n)$ space estimated via Pareto MTL with MINE and associated t-SNE plots. Panel **A** shows all 12 Pareto candidates (left) and the culled non-dominated points (right) in the $(\overline{U}_n, \overline{V}_n)$ space on TM-MARROW dataset. Panel **B**, **C** and **D** show the t-SNE plots of the latent $\boldsymbol{z}$ for the third, sixth and tenth candidate on test set, respectively. Each point in the t-SNE plots indicates a cell and the points are colored by batches (left) and pre-annotated cell types (right). As expected, from Panel **B** to Panel **D**, we see increasingly better clustering performance at the cost of more batch effect

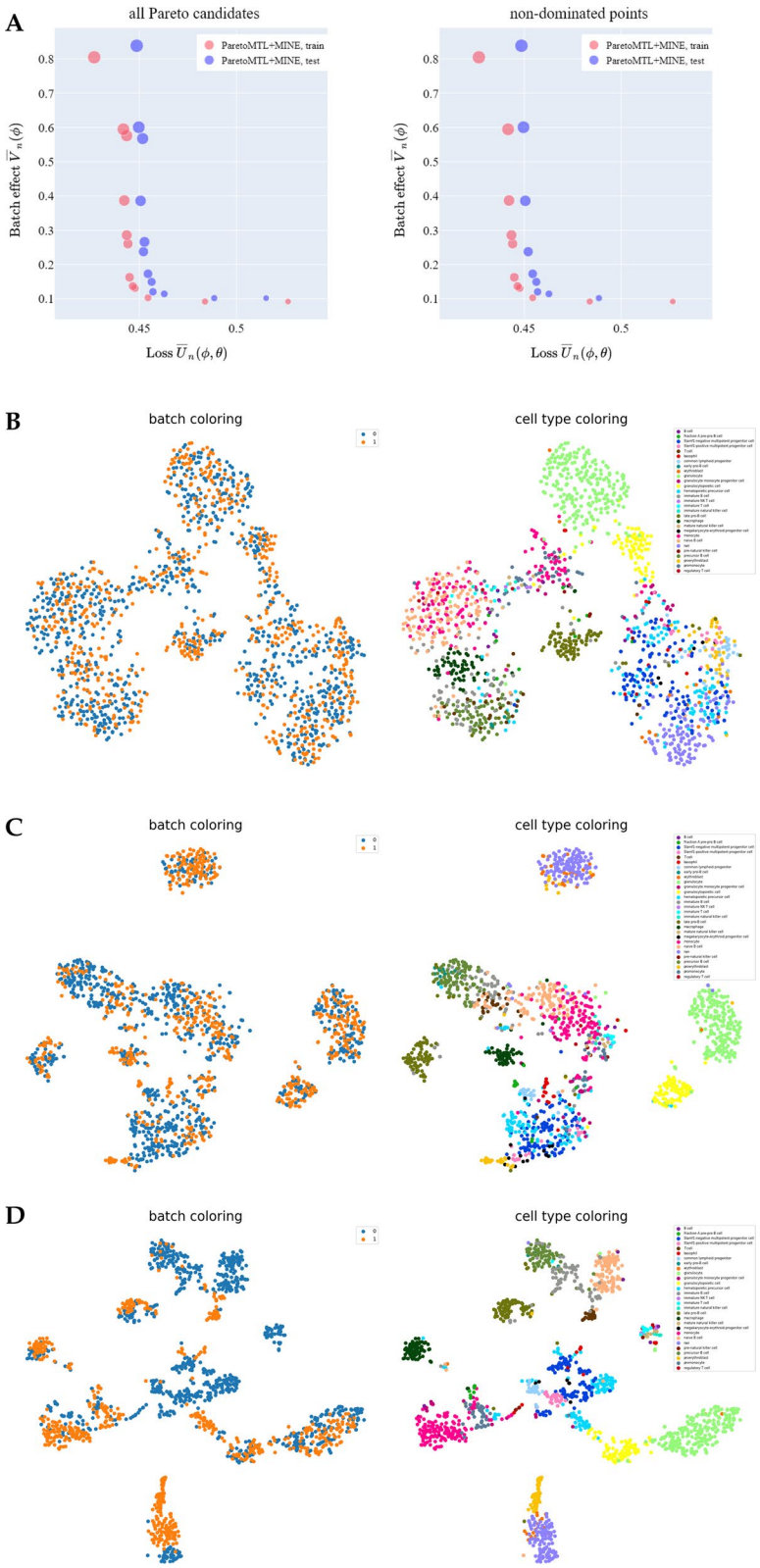

Fig. 2 (See legend on previous page.)

more batch effect) while generative loss $\overline{U}_n$ is decreasing (i.e. more conservation of biological variation) from the third, to the sixth, to the tenth candidate.

While t-SNE plots provide a useful visual aid to understand the various Pareto candidates, we can use surrogate metrics to obtain a quantitative understanding. Following [6], we use batch entropy (BE) to measure (roughly) how well latent $\boldsymbol{z}$ from different batches "mix". We also use the following clustering metrics: Averaged Silhouette Width (ASW), Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and Unsupervised Clustering Accuracy (UCA). For all the above surrogate measures, higher is better. In particular, higher BE means better "mixing" of latent $\boldsymbol{z}$ from different batches. Higher ASW means clusters of latent $\boldsymbol{z}$ are further apart. ARI, NMI and UCA measure how consistent cell clustering of latent $\boldsymbol{z}$ by K-Means is with cell clustering based on the pre-annotated cell types and higher values correspond with better clustering. As expected, the third candidate (Fig. 2B) has largest BE and smallest clustering surrogate metrics with

$$BE = 0.60, ASW = 0.09, ARI = 0.38, NMI = 0.64, UCA = 0.44,$$

, while the sixth candidate (Fig. 2C) has intermediate surrogate metrics with

$$BE = 0.53, ASW = 0.19, ARI = 0.51, NMI = 0.73, UCA = 0.56,$$

, and the tenth candidate (Fig. 2D) has the smallest BE and largest clustering surrogate metrics where

$$BE = 0.33, ASW = 0.21, ARI = 0.54, NMI = 0.75, UCA = 0.57.$$

Pareto MTL versus scalarization

Here we present results comparing Pareto MTL (see Pareto MTL) and the naive scalarization approach for estimating the $(\overline{U}_n, \overline{V}_n)$ Pareto front when V_n is given by either the MINE or MMD measure. The results on the TM-MARROW dataset are shown in Fig. 3; results on the MACAQUE-RETINA dataset can be found in Additional file 1: Fig S2. Points of smaller size stand for smaller generative loss minimization during training (i.e. smaller λ for scalarization). Note that the two extreme points are the same for Pareto MTL and scalarization.

When MINE is the batch effect measure, we observe that as we progress from candidate 1 to 12, the generative loss $\overline{U}_n$ decreases while the batch effect measure $\overline{V}_n$ increases (Fig. 3A, Additional file 1: Fig S2). We describe in this case the candidate ordering is well respected. Furthermore, we see that Pareto MTL is better than the scalarization scheme for estimating a well-distributed Pareto front (Fig. 3B, Additional file 1: Fig S2).

When MMD is the batch effect measure, we first notice that the candidate ordering is not well-respected by neither Pareto MTL nor the scalarization scheme (Fig. 3C). We suspect this issue arises from the difficulty in selecting a proper MMD bandwidth across the candidates. Second, Pareto MTL performs either similarly (Fig. 3D) or better (Additional file 1: Fig S2) than scalarization for estimating an well-distributed Pareto front.

The visual comparison in Fig. 3 and Fig S2 is based on *one* random splitting of the dataset into training and testing set. In order to compare Pareto MTL and the

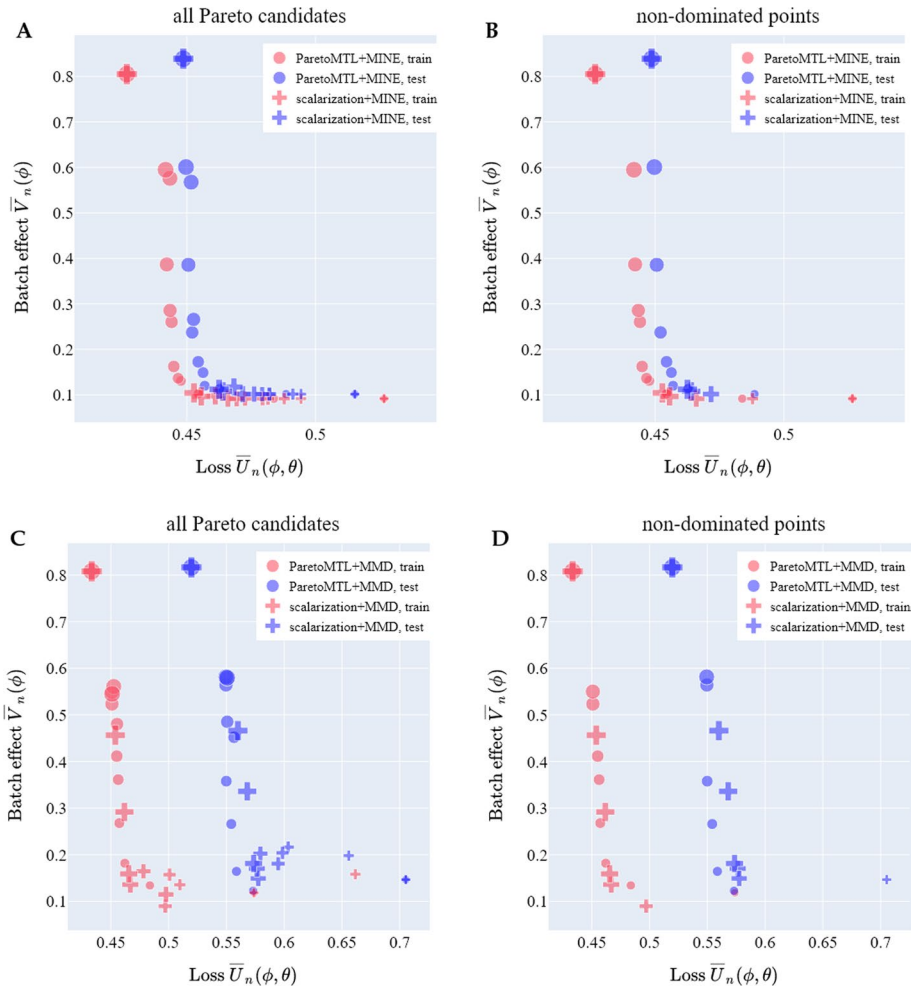

Fig. 3 Pareto MTL versus scalarization. In Panels **A** and **C**, we show the 12 Pareto candidates produced when MINE and MMD, respectively, are used to measure the effect in the TM-MARROW dataset. In the corresponding Panels **B** and **D**, we show the non-dominated points only. When batch effect is measured using MINE, Pareto MTL produces a more diverse set of trade-offs, while scalarization tends to produce trade-offs at the extreme regions. When MMD is the batch effect indication, Pareto MTL seems to perform similarly to scalarization

scalarization scheme more systematically, we focus on three metrics (for which higher is better) that can be measured on Pareto fronts:

- *percentage* of non-dominated points in all Pareto candidates
- *hypervolume* [14] evaluates the coverage area of the estimated Pareto front (see Hypervolume)
- *number of distinct choices* (NDC) [15] measures the number of meaningful Pareto solutions that are sufficiently distinct to one another (see NDC)

These three metrics evaluate three respective properties of the Pareto front:

- *cardinality*, which quantifies the number of non-dominated points,

- *convergence*, which evaluates how close a set of estimated non-dominated points is from the true Pareto front in the objective space,
- *distribution*, which measures how well distributed the points are on the estimated Pareto front [16].

The Pareto MTL and scalarization scheme (either with MINE or MMD as batch effect indication) are each run for 10 training-testing splits. The percentage, hypervolume and NDC of the Pareto fronts in the $(\overline{U}_n, \overline{V}_n)$ space on the two datasets (see Single cell RNA-seq datasets), are shown in Table 1.

We observe higher percentages of non-dominated points for Pareto MTL than scalarization. In terms of hypervolume, Pareto MTL and scalarization perform similarly for both MINE and MMD. Note that hypervolume for Pareto MTL with MMD is not significantly higher than scalarization with MMD when considering the standard deviation. One reason could be that the relative values of the hypervolume metric of two non-dominated point sets (i.e. which set has a larger hypervolume and which set has a smaller hypervolume) depend on the chosen reference point [16]. As for NDC, Pareto MTL with MINE is significantly better than scalarization with MINE; it is consistent with the observation that Pareto MTL with MINE can estimate a more diverse Pareto front than scalarization with MINE (Fig. 3B). Pareto MTL with MMD also produces Pareto fronts with higher NDC than scalarization with MMD, but the superiority is not as dramatic as that between Pareto MTL with MINE and scalarization with MINE.

Pareto MTL with MINE versus Pareto MTL with MMD

Though both MINE and MMD measure batch effect, Pareto MTL with MINE and Pareto MTL with MMD cannot be directly compared in the $(\overline{U}_n, \overline{V}_n)$ space. Specifically, MINE measures the KL divergence between distributions (see MINE for measuring batch effect V_n) while MMD is the maximum mean embedding distance between distributions (see MMD for measuring batch effect V_n). We can however make meaningful comparisons by introducing surrogate measures for V_n.

We use two surrogate measures for V_n: 1) the aforementioned batch entropy (BE) and 2) nearest neighbor (NN). NN here refers to a method proposed in [17] to estimate the mutual information (MI) between the continuous latent $\boldsymbol{z}$ and discrete batch $\boldsymbol{s}$ instead of the traditional k-nearest neighbor algorithm for classification. Larger values of BE correspond to smaller batch effect, while larger values of NN correspond to higher batch effect. In Fig S4 and Fig S5, see Additional file 1, we show simulation results that indicate NN estimates well the mutual information between a continuous variable and a categorical variable. We should note that NN cannot be used during neural network training because it is not amenable to back-propagation.

We could also use the aforementioned clustering metrics (ASW, ARI, NMI and UCA) as the evaluation surrogates for U_n since they are more interpretable than the ELBO. Therefore, in total we can produce ten types of trade-off curves for evaluation purposes, i.e. five measures of $\overline{U}_n$ ($\overline{U}_n$ itself and four clustering surrogates) cross two surrogate measures of $\overline{V}_n$ (BE and NN). For brevity, we will only present a subset of five to compare Pareto MTL with MINE and Pareto MTL with MMD: $\overline{U}_n$ versus *NN*, and negative ASW/NMI/ARI/UCA versus negative BE. The trade-offs of $\overline{U}_n$ versus *NN*, negative

Table 1 Pareto MTL versus scalarization

Dataset	Percentage				Hypervolume				NDC			
	Train		Test		Train		Test		Train		Test	
	Pareto MTL	Scalarization	Pareto MTL	Scalarization	Pareto MTL	Scalarization	Pareto MTL	Scalarization	Pareto MTL	Scalarization	Pareto MTL	Scalarization
$\overline{U}_n(\boldsymbol{\phi},\boldsymbol{\theta})$ **versus** $\overline{V}_n(\boldsymbol{\phi})$ when $V_n(\boldsymbol{\phi}) = MINE(\boldsymbol{\phi})$												
TM-MARROW	**0.88** ± 0.08	0.57 ± 0.17	**0.58** ± 0.16	0.42 ± 0.15	0.19 ± 0.01	0.19 ± 0.01	0.13 ± 0.03	0.13 ± 0.04	**8.1** ± 1.20	3.0 ± 0.47	**5.8** ± 1.55	2.5 ± 0.71
MACAQUE-RETINA	**0.83** ± 0.10	0.46 ± 0.09	**0.47** ± 0.13	0.35 ± 0.13	**0.51** ± 0.01	0.50 ± 0.01	0.35 ± 0.02	0.35 ± 0.02	**7.5** ± 1.27	3.7 ± 0.95	**4.2** ± 1.03	2.3 ± 1.06
$\overline{U}_n(\boldsymbol{\phi},\boldsymbol{\theta})$ **versus** $\overline{V}_n(\boldsymbol{\phi})$ when $V_n(\boldsymbol{\phi}) = MMD(\boldsymbol{\phi})$												
TM-MARROW	**0.84** ± 0.06	0.58 ± 0.06	**0.55** ± 0.08	0.49 ± 0.07	26.68 ± 0.41	**26.91** ± 0.37	25.95 ± 1.32	**26.78** ± 1.03	**8.4** ± 0.70	6.6 ± 0.70	**5.8** ± 1.03	5.3 ± 0.82
MACAQUE-RETINA	**0.78** ± 0.10	0.75 ± 0.10	**0.61** ± 0.09	0.55 ± 0.06	752.35 ± 17.90	**755.46** ± 6.19	623.00 ± 83.73	**700.07** ± 32.06	**8.0** ± 0.47	6.9 ± 1.10	**6.0** ± 0.94	4.7 ± 0.82

Higher percentage/hypervolume/NDC indicates of better Pareto front estimation

The values are mean ± standard deviation over 10 Monte Carlos of training-testing split of the dataset. Bold illustrates higher mean in comparison

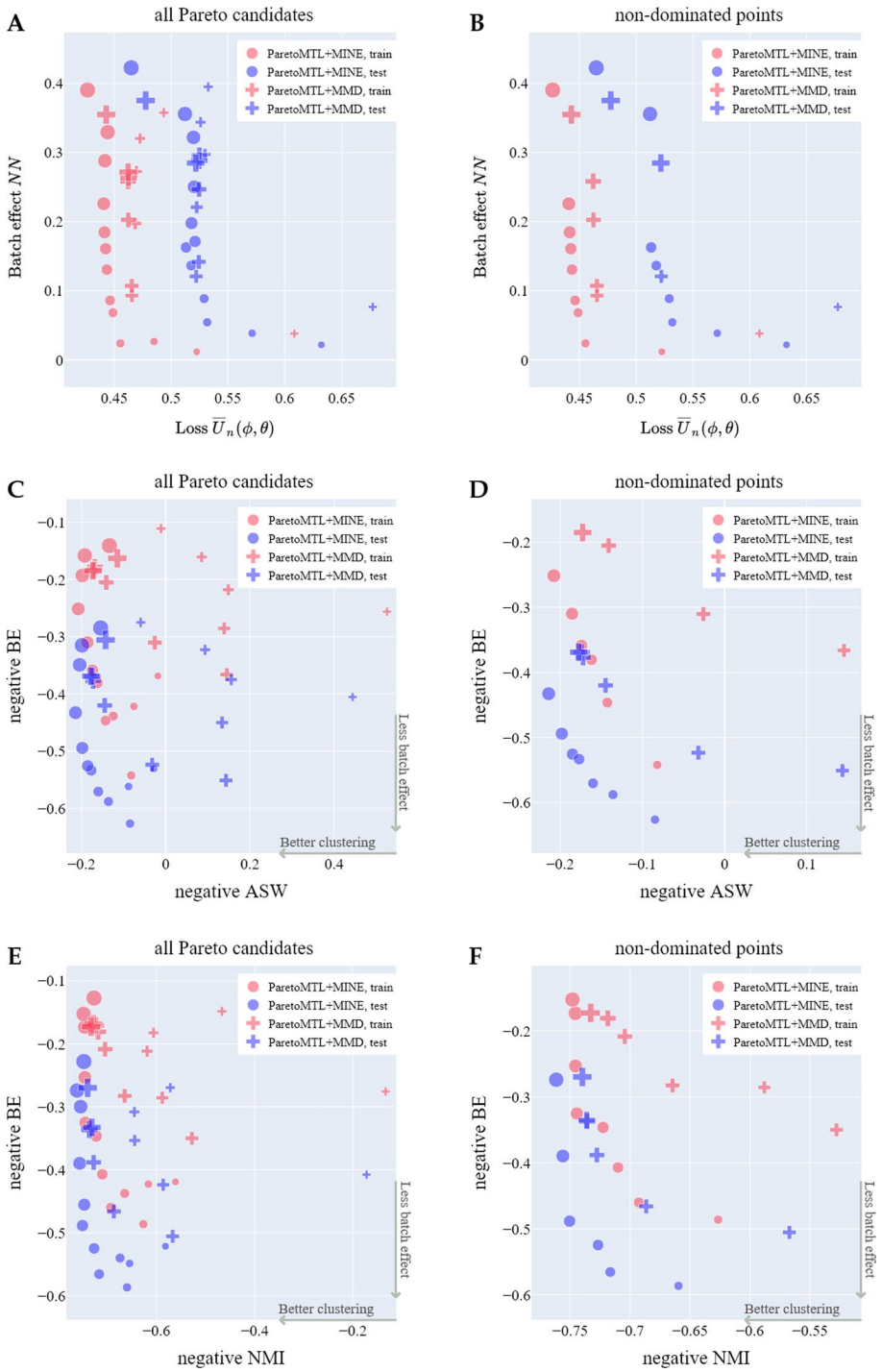

Fig. 4 The trade-off curves of surrogate metrics. Compare Pareto MTL with MINE and Pareto MTL with MMD to estimate three types of trade-offs of surrogate metrics for all Pareto candidates and non-dominated points on the TM-MARROW dataset: **A**, **B** trade-offs between generative loss $\overline{U}_n$ and mutual information estimator *NN*; **C**, **D** the trade-offs between negative ASW and negative BE; **E**, **F** the trade-offs between negative NMI and negative BE. The point with the largest marker size in **C** and **E** corresponds to scVI alone (i.e. minimizing U_n)

ASW/NMI versus negative BE for TM-MARROW dataset are shown in Fig. 4 and the trade-offs of negative ARI/UCA versus negative BE for TM-MARROW dataset in Additional file 1: Fig S6.

The trade-offs in the $(\overline{U}_n, NN)$ space for Pareto MTL with MINE (Fig. 4A) respects the candidate ordering as in the $(\overline{U}_n, \overline{V}_n)$ space in Fig. 3A. In contrast, Pareto MTL with MMD produces trade-offs in the $(\overline{U}_n, NN)$ space in disarray (Fig. 4A). Besides, we observe that Pareto MTL with MINE estimates a better set of non-dominated $(\overline{U}_n, NN)$ trade-off points which dominates that estimated by Pareto MTL with MMD (Fig. 4B).

Similarly, in the negative ASW/NMI and negative BE space, Pareto MTL with MINE produces trade-offs that respect candidate ordering better than Pareto MTL with MMD does (Fig. 4C, E) . With the exception of some points at the extremes, Pareto MTL with MINE has a clearer trend than Pareto MTL with MMD, in the sense that as negative clustering metrics decrease, negative BE increases. Meanwhile, Pareto MTL with MINE produces a set of non-dominated trade-off points which dominates those from Pareto MTL with MMD (Fig. 4D, F). Similar conclusion can be drawn for trade-offs between negative ARI/UCA and negative BE (Additional file 1: Fig S6).

Interestingly, Pareto MTL with MINE can produce points with better clustering *and* better batch removal simultaneously than scVI (Fig. 4C, E). Specifically we find that scVI yields a dominated point in the trade-off of negative ASW/NMI versus negative BE (Fig. 4D, F).

Analogously we could obtain trade-offs in the surrogate space for scalarization with MINE and scalarization with MMD (Additional file 1: Fig S7). The results provide extra support that MINE is better than MMD for measuring batch effect since scalarization with MINE obtains better trade-offs in terms of Pareto candidate ordering and convergence of the non-dominated points. However, as already seen in Fig. 3B, the problem for scalarization with MINE is that it can only recover a small part of the Pareto front (i.e. diversity problem).

Finally, we can quantitatively compare Pareto MTL with MINE and Pareto MTL with MMD by again evaluating percentage, hypervolume and NDC, but this time in the surrogate bi-objective space. The results for $\overline{U}_n$ versus NN and negative ASW/NMI versus negative BE on the two datasets are shown in Table 2; the results for negative ARI/UCA versus negative BE are shown in Additional file 1: Table S1. Compared with Pareto MTL with MMD, Pareto MTL with MINE produces better surrogate Pareto fronts with (1) higher or similar percentage of non-dominated points (i.e. higher or similar cardinality), (2) larger hypervolume (i.e. better convergence) and (3) higher or similar NDC (i.e. better or similar diversity).

Discussion

We briefly discuss limitations and future work in this section. Although we have shown MINE to be superior to MMD in the current context of the work, MINE is computationally more demanding than MMD. Furthermore, using MINE in conjunction with scVI requires adversarial training which can be unstable in the hands of an unpracticed user. Finally, it is possible that designing the proper neural network architecture in MINE is a difficult task, though we did not encounter this in our data analysis. In fact, based on the analysis we have performed so far, the trade-off curves seem quite robust to the MINE

Table 2 Pareto MTL with MINE versus Pareto MTL with MMD

Dataset	Percentage				Hypervolume				NDC			
	Train		Test		Train		Test		Train		Test	
	MINE	MMD	MINE	MMD	MINE	MMD	MINE	MMD	MINE	MMD	MINE	MMD
$\overline{U}_n(\boldsymbol{\phi},\boldsymbol{\theta})$ **versus** ***NN***												
TM-MARROW	**0.85** ± 0.07	0.48 ± 0.07	**0.58** ± 0.18	0.35 ± 0.12	**4.35** ± 0.04	3.91 ± 0.16	**4.29** ± 0.06	3.85 ± 0.21	**7.7** ± 0.82	5.4 ± 0.52	**5.6** ± 1.35	4.1 ± 1.20
MACAQUE-RETINA	**0.72** ± 0.10	0.27 ± 0.08	**0.47** ± 0.13	0.14 ± 0.04	**293.82** ± 1.77	273.48 ± 4.23	**295.15** ± 4.17	257.92 ± 8.90	**4.1** ± 0.74	2.0 ± 0.00	**3.8** ± 1.55	1.2 ± 0.42
-ASW versus -BE												
TM-MARROW	**0.45** ± 0.06	0.43 ± 0.09	**0.48** ± 0.10	0.41 ± 0.05	**0.29** ± 0.02	0.16 ± 0.02	**0.38** ± 0.01	0.30 ± 0.02	**4.6** ± 0.84	4.2 ± 0.79	**4.9** ± 1.37	3.7 ± 0.48
MACAQUE-RETINA	**0.61** ± 0.07	0.33 ± 0.10	**0.64** ± 0.10	0.34 ± 0.11	**0.32** ± 0.01	0.15 ± 0.02	**0.30** ± 0.01	0.19 ± 0.03	**4.4** ± 0.84	3.1 ± 0.57	**5.5** ± 0.85	2.6 ± 0.70
-NMI versus -BE												
TM-MARROW	**0.56** ± 0.08	0.49 ± 0.08	**0.58** ± 0.12	0.40 ± 0.09	**0.28** ± 0.01	0.16 ± 0.02	**0.37** ± 0.01	0.31 ± 0.02	**6.0** ± 0.67	4.5 ± 0.85	**5.0** ± 1.33	4.3 ± 0.95
MACAQUE-RETINA	**0.38** ± 0.11	0.23 ± 0.11	**0.49** ± 0.18	0.33 ± 0.12	**0.26** ± 0.02	0.11 ± 0.02	**0.25** ± 0.01	0.14 ± 0.02	**3.5** ± 0.85	2.5 ± 0.85	**4.5** ± 1.27	3.5 ± 1.08

The Pareto fronts—generative loss $\overline{U}_n$ versus mutual information estimator *NN* and negative ASW/NMI versus negative BE from Pareto MTL with MINE and Pareto MTL with MMD are compared

Better Pareto front has higher percentage/hypervolume/NDC

The values are mean ± standard deviation over 10 Monte Carlos of training-testing split of the dataset. Bold illustrates higher mean in comparison

architecture in contrast to the high sensitivity around MMD kernel and bandwidth choice.

In future work, we aim to incorporate other deep architectures into our framework. Though we have focused in this work on the scVI generative model for scRNA-seq data, our research is broadly applicable to other deep architectures for scRNA-seq data. For instance Pareto MTL with MINE can be wrapped around such deep architectures as SAUCIE and DESC [18]. Future work may also consider generalizing to other nuisance factors such as quality control metrics which assess the errors and corrections of aligning transcripts to some reference genome during scRNA sequencing.

Conclusion

We propose using Pareto MTL for estimation of Pareto front in conjunction with MINE for measurement of batch effect to produce the trade-off curve between conservation of biological variation and removal of batch effect. We first demonstrated that Pareto MTL improves upon the naive scalarization approach in finding the Pareto front. In particularly, Pareto MTL produces well-distributed trade-off points in contrast to the scalarization approach which produces points in the extremes. We next demonstrated that the new batch effect measure based on MINE is preferable to the more standard MMD measure in the sense that the former produces trade-off points that respect candidate ordering and are interpretable in surrogate metric spaces. Our experimental results also show that Pareto MTL with MINE is superior to both Pareto MTL with MMD and scVI alone for clustering of cell types. Finally, in treating batch effect as an objective important in its own right, the multi-objective optimization framework we adopt allows for an understanding of the entire tradeoff curve between conservation of biological information and batch effect removal. This is in contrast to traditional analyses which study at most *one* specific trade-off.

Methods

MINE for measuring batch effect V_n

Mutual information $I(\boldsymbol{z}, \boldsymbol{s})$ measures the dependence between continuous latent variable $\boldsymbol{z}$ and categorical $\boldsymbol{s}$ as follows

$$I(\boldsymbol{z}, \boldsymbol{s}) = D(P_{\boldsymbol{zs}} || P_{\boldsymbol{z}} \bigotimes P_{\boldsymbol{s}})$$

where $P_{\boldsymbol{zs}}$ is the joint distribution and $P_{\boldsymbol{z}} \bigotimes P_{\boldsymbol{s}}$ is the product of the two marginal distributions. It was shown in [11] that we can estimate $I(\boldsymbol{z}, \boldsymbol{s})$ by maximizing a lower bound $I_\Psi(\boldsymbol{z}, \boldsymbol{s})$ defined as

$$I_\Psi(\boldsymbol{z}, \boldsymbol{s}) = \sup_{\psi \in \Psi} \left\{ \mathbb{E}_{P_{\boldsymbol{zs}}} f_\psi - \log(\mathbb{E}_{P_{\boldsymbol{z}} \bigotimes P_{\boldsymbol{s}}} \exp(f_\psi)) \right\}$$

where $f_\psi : \mathbb{R}^{d_z} \times \{0, 1\}^B \to \mathbb{R}$ is a deep neural network with parameters $\psi \in \Psi$.

Let $\boldsymbol{z}_i$ be a realization from the posterior distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_i)$. We should point out that though $(\boldsymbol{z}_i, \boldsymbol{s}_i)$ are independent across i, they are not identically distributed since $\boldsymbol{x}_i$ varies. We will be considering the so-called aggregated variational posterior for latent $\boldsymbol{z}$, see discussion in [10], which in our case is a n-component Gaussian mixture with

equal weights. We shall treat $\{\boldsymbol{z}_i\}_{i=1}^n$ as a sample from this Gaussian mixture and $\boldsymbol{s}_i$ is the known coupled batch for each $\boldsymbol{z}_i$. To obtain a sample from the marginal $P_{\boldsymbol{s}}$ we shuffle $\boldsymbol{s}_i$ to break the coupling with $\boldsymbol{z}_i$; we denote this sample $\{\boldsymbol{s}_i'\}_{i=1}^n$. This leads us to define the following measure for batch effect:

$$\text{MINE}\,(\boldsymbol{\phi}) = \frac{1}{n}\sum_{i=1}^{n} f_{\hat{\boldsymbol{\psi}}}(\boldsymbol{z}_i, \boldsymbol{s}_i) - \log\left(\frac{1}{n}\sum_{i=1}^{n}\exp f_{\hat{\boldsymbol{\psi}}}(\boldsymbol{z}_i, \boldsymbol{s}_i')\right), \tag{8}$$

where $\hat{\psi}$ is the result of using minibatch stochastic gradient ascent, see Alg. 1.

The simulation studies (Additional file 1: Fig S4, Fig S5) demonstrate that MINE approximates well the true MI between a categorical variable and a continuous variable.

Algorithm 1 Train MINE

1: **Inputs:** training data $\{(\boldsymbol{x}_i, \boldsymbol{s}_i)\}_{i=1}^n$, encoder parameter $\boldsymbol{\phi}$, and initial MINE weight $\boldsymbol{\psi}_0$.
2: **for** $t = 1$ to T iterations **do**
3: Sample minibatch $\{(\boldsymbol{x}_i, \boldsymbol{s}_i)\}_{i=1}^m$ of size m;
4: Draw corresponding $\boldsymbol{z}_i$ from $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_i), i = 1, \ldots, m$.
5: Evaluate Eq. (8) on $\{(\boldsymbol{z}_i, \boldsymbol{s}_i)\}_{i=1}^m$ for $\boldsymbol{\psi}_{t-1}$, call the result $\text{MINE}_{\boldsymbol{\psi}_{t-1}}$.
6: $\boldsymbol{\psi}_t = \boldsymbol{\psi}_{t-1} + \delta\nabla_{\boldsymbol{\psi}}\,\text{MINE}_{\boldsymbol{\psi}_{t-1}}(\boldsymbol{\phi})$ where δ is the step size.
7: **end for**
8: **Outputs:** $\hat{\boldsymbol{\psi}} = \boldsymbol{\psi}_T$

MMD for measuring batch effect V_n

The maximum mean discrepancy (MMD) [19] measures the discrepancy between two densities p and q using the unit ball in a reproducing kernel Hilbert space $\mathcal{H}$ with associated kernel $k(\cdot,\cdot)$. The squared MMD is given by

$$MMD^2(p,q) = \mathbb{E}_{x,x'}k(x,x') - 2\,\mathbb{E}_{x,y}k(x,y) + \mathbb{E}_{y,y'}k(y,y').$$

where x and x' are independent random variables with distribution p and y and y' are independent random variables with distribution q.

For simplicity let us assume we have two batches. To measure the batch effect, we will apply MMD to measure the disparity between the distributions $\boldsymbol{z}|\boldsymbol{s}=0$ and $\boldsymbol{z}|\boldsymbol{s}=1$. Let us denote $\{\boldsymbol{z}_{0i}\}_{i=1}^{n_0}$ for the samples from batch 0 and $\{\boldsymbol{z}_{1i}\}_{i=1}^{n_1}$ for the samples from batch 1. We estimate the MMD between $\{\boldsymbol{z}_{0i}\}_{i=1}^{n_0}$ and $\{\boldsymbol{z}_{1i}\}_{i=1}^{n_1}$ using the biased estimator

$$\text{MMD}\,(\boldsymbol{\phi}) = \left[\frac{1}{n_0^2}\sum_{i,j=1}^{n_0}k(\boldsymbol{z}_{0i},\boldsymbol{z}_{0j}) + \frac{1}{n_1^2}\sum_{i,j=1}^{n_1}k(\boldsymbol{z}_{1i},\boldsymbol{z}_{1j}) - \frac{2}{n_0 n_1}\sum_{i=1}^{n_0}\sum_{j=1}^{n_1}k(\boldsymbol{z}_{0i},\boldsymbol{z}_{1j})\right]^{\frac{1}{2}} \tag{9}$$

For development of this estimator see Section 2.2 in [19]. As in the previous section, the latent variable $\boldsymbol{z}_i$ is sampled from $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_i)$ though in the right hand side of Eq. (9) we have suppressed the dependence on $\boldsymbol{\phi}$.

Following [10], we shall take the kernel k in our experiments to be a mixture of 5 Radial Gaussian kernels:

$$k(z_1, z_2) = \frac{1}{5}\sum_{i=1}^{5} \exp\left(-\frac{\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_2^2 \times \sqrt{d_z}}{2b_i^2}\right),$$

where $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are any pair of latent $\boldsymbol{z}$, $(b_1, ..., b_5)$ is the chosen bandwidths and $d_{\boldsymbol{z}}$ is the dimension of $\boldsymbol{z}$. To avoid different scales for each dimension, the latent $\boldsymbol{z}$ is standardized such that each dimension has zero mean and unit standard deviation.

Pareto MTL

Here we describe the application of the Pareto MTL method proposed in [8] to our work. Throughout our experiments, we will use the set of preference vectors

$$\left\{\boldsymbol{c}_k = \left(\cos\left(\frac{(k-1)\pi}{2(K-1)}\right), \sin\left(\frac{(k-1)\pi}{2(K-1)}\right)\right) | k = 1, ..., K\right\}$$

which is sensible since our particular choices of U_n and V_n will always be non-negative. Associated to the k^{th} preference vector $\boldsymbol{c}_k$ is the constrained optimisation subproblem

$$\begin{aligned} &\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \quad \boldsymbol{L}(\boldsymbol{\phi},\boldsymbol{\theta}) = (\overline{U}_n(\boldsymbol{\phi},\boldsymbol{\theta}), \overline{V}_n(\boldsymbol{\phi}))^T \\ &s.t. \quad \boldsymbol{g}_j(\boldsymbol{\phi},\boldsymbol{\theta},k) = (\boldsymbol{c}_j - \boldsymbol{c}_k)^T \boldsymbol{L}(\boldsymbol{\phi},\boldsymbol{\theta}) \leq 0, \forall j = 1, ..., K. \end{aligned} \tag{10}$$

To begin, we seek initialization of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ such that the bi-objective $\boldsymbol{L}(\boldsymbol{\phi},\boldsymbol{\theta})$ starts its minimization from a place near the preference vector $\boldsymbol{c}_k$. The initialization proceeds as follows.

1 Obtain the Lagrange multipliers β_j by solving:

$$\max_{\beta_j} \quad -\frac{1}{2}\|\sum_{j\in I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k)} \beta_j \nabla \boldsymbol{g}_j(\boldsymbol{\phi},\boldsymbol{\theta},k)\|^2 s.t. \quad \sum_{j\in I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k)} \beta_j = 1, \beta_j \geq 0, \forall j \in I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k), \tag{11}$$

where $I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k) = \{j | g_j(\boldsymbol{\phi},\boldsymbol{\theta},k) \geq -\epsilon, j = 1, ..., K\}$ and ϵ is a pre-defined small positive value.

2 Update $(\boldsymbol{\phi},\boldsymbol{\theta})$ by descending the gradient vector $\boldsymbol{d}$ with step size δ:

$$(\boldsymbol{\phi},\boldsymbol{\theta}) \leftarrow (\boldsymbol{\phi},\boldsymbol{\theta}) - \delta \boldsymbol{d}, \text{where } \boldsymbol{d} = \sum_{j\in I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k)} \beta_j \nabla \boldsymbol{g}_j(\boldsymbol{\phi},\boldsymbol{\theta},k). \tag{12}$$

After initialization, the subproblem in Eq. (10) is solved as another dual problem:

1 Obtain the Lagrange multipliers λ_1, λ_2 and β_j by solving:

$$\begin{aligned} &\max_{\lambda_1,\lambda_2,\beta_j} \quad -\frac{1}{2}\|\lambda_1 \nabla \overline{U}_n(\boldsymbol{\phi},\boldsymbol{\theta}) + \lambda_2 \nabla \overline{V}_n(\boldsymbol{\phi}) + \sum_{j\in I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k)} \beta_j \nabla g_j(\boldsymbol{\phi},\boldsymbol{\theta},k)\|^2 \\ &s.t. \quad \lambda_1 + \lambda_2 + \sum_{j\in I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k)} \beta_j = 1, \lambda_1 \geq 0, \lambda_2 \geq 0, \beta_j \geq 0, \forall j \in I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k). \end{aligned} \tag{13}$$

2 Update $(\boldsymbol{\phi},\boldsymbol{\theta})$ by descending the gradient vector $\boldsymbol{d}$ with step size δ:

$$(\boldsymbol{\phi},\boldsymbol{\theta}) \leftarrow (\boldsymbol{\phi},\boldsymbol{\theta}) - \delta \boldsymbol{d}, \text{where } \boldsymbol{d} = \lambda_1 \nabla \overline{U}_n(\boldsymbol{\phi},\boldsymbol{\theta}) + \lambda_2 \nabla \overline{V}_n(\boldsymbol{\phi}) + \sum_{j \in I_\epsilon(\boldsymbol{\phi},\boldsymbol{\theta},k)} \beta_j \nabla g_j(\boldsymbol{\phi},\boldsymbol{\theta},k). \tag{14}$$

Pareto MTL with MINE

When $\overline{V}_n(\boldsymbol{\phi})$ is based on MMD($\boldsymbol{\phi}$), the implementation of Pareto MTL is straightforward. However, when $\overline{V}_n(\boldsymbol{\phi})$ is based on MINE ($\boldsymbol{\phi}$), we employ adversarial training since the MINE network has its own parameters $\boldsymbol{\psi}$ that need to be learned. Specifically, MINE ($\boldsymbol{\phi}$) is **maximized** over the MINE neural network parameters $\boldsymbol{\psi}$ before every **minimization** step over the variational autoencoder parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. Alg. 3 gives an overview of Pareto MTL with MINE via adversarial training for the k-th subproblem. An estimate of the Pareto front can be obtained by running Alg. 3 for $k = 1, ..., K$.

Algorithm 2 Initialization for Pareto MTL

Inputs: training data $\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_{i=1}^n$.
Pre-train scVI to get $(\boldsymbol{\phi}_0, \boldsymbol{\theta}_0)$
Pre-train MINE to get $\boldsymbol{\psi}_0 = \text{Alg. } 1(\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_{i=1}^n, \boldsymbol{\phi}_0, \boldsymbol{\psi})$, where $\boldsymbol{\psi}$ is randomly initialized.
for $t = 1$ to T iterations **do**
    $\boldsymbol{\psi}_t = \text{Alg. } 1(\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_{i=1}^n, \boldsymbol{\phi}_{t-1}, \boldsymbol{\psi}_{t-1})$.
    Sample minibatch $\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_{i=1}^m$ of size m;
    Fix $\boldsymbol{\psi}_t$ and $(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t) = (\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) - \delta \boldsymbol{d}$ where δ is step size and $\boldsymbol{d}$ is the gradient vector in Eq. (12).
end for
Outputs: $(\boldsymbol{\phi}_T, \boldsymbol{\theta}_T, \boldsymbol{\psi}_T)$

Algorithm 3 Pareto MTL with MINE

Inputs: training data $\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_{i=1}^n$.
$(\boldsymbol{\phi}_0, \boldsymbol{\theta}_0, \boldsymbol{\psi}_0) = \text{Alg. } 2(\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_{i=1}^n)$.
for $t = 1$ to T iterations **do**
    $\boldsymbol{\psi}_t = \text{Alg. } 1(\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_{i=1}^n, \boldsymbol{\phi}_{t-1}, \boldsymbol{\psi}_{t-1})$.
    Sample minibatch $\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_{i=1}^m$ of size m;
    Fix $\boldsymbol{\psi}_t$ and $(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t) = (\boldsymbol{\phi}_{t-1}, \boldsymbol{\theta}_{t-1}) - \delta \boldsymbol{d}$ where δ is step size and $\boldsymbol{d}$ is the gradient vector in Eq. (14).
end for
Output: $(\boldsymbol{\phi}_T, \boldsymbol{\theta}_T)$.

Plotting the $(\overline{U}_n, \overline{V}_n)$ Pareto fronts

For each of the candidates in Pareto MTL or scalarization, a resulting encoder-decoder pair is produced $(\boldsymbol{\phi}_T, \boldsymbol{\theta}_T)$. To understand such figures as Figs. 2A and 3, we describe how we evaluated U_n and V_n given a pair $(\boldsymbol{\phi}_T, \boldsymbol{\theta}_T)$. It is straightforward to evaluate U_n given $(\boldsymbol{\phi}_T, \boldsymbol{\theta}_T)$.

When MMD is used to measure batch effect, it is also easy to evaluate V_n given $\boldsymbol{\phi}_T$. However, when MINE is used to measure batch effect, the evaluation of V_n is more subtle since MINE has its own training parameters unlike MMD. We first train a de-novo MINE neural network to its optimal ($\boldsymbol{\psi} = \boldsymbol{\psi}^*$, see MINE for measuring batch effect V_n) with the encoder weight $\boldsymbol{\phi}$ fixed at $\boldsymbol{\phi}_T$. Then with $\boldsymbol{\psi} = \boldsymbol{\psi}^*$, Figs. 2A and 3 proceed to plot $V_n = MINE(\boldsymbol{\phi}_T)$.

Table 3 Hyperparameters for Pareto MTL

	pre-epochs	pre-lr	pre-adv-epochs	pre-adv-lr	epochs	lr	adv-epochs	adv-lr	MMD bandwidths
TM-MARROW									
Pareto MTL with MINE	200	1*e*−3	400	5*e*−5	150	1*e*−3	1	5*e*−5	NA
Pareto MTL with MMD	200	1*e*−3	NA	NA	250	1*e*−3	NA	NA	1,2,5,8,10
MACAQUE-RETINA									
Pareto MTL with MINE	150	1*e*−3	400	5*e*−5	150	1*e*−3	1	5*e*−5	NA
Pareto MTL with MMD	150	1*e*−3	NA	NA	250	1*e*−3	NA	NA	1,2,5,8,10

The abbreviation "pre" means pre-training, "adv" means MINE adversarial training, "lr" means learning rate

Training details

To estimate the expectation in the ELBO, a single $\boldsymbol{z}$ is sampled from the variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$ and likewise for l from $q_{\boldsymbol{\phi}}(l_i|\boldsymbol{x}_i)$. When possible, we use analytic expressions for the KL divergences between two Gaussian distributions.

We use the same encoder and decoder architectures as the scVI in [6]. For MINE neural network architecture, we use 10 fully connected layers with 128 hidden nodes and ELU activation for each layer. The weights $\boldsymbol{\psi}$ of the MINE neural network is initialized by a normal distribution with 0 mean and 0.02 standard deviation.

The hyperparameters for Pareot MTL on the TM-MARROW and MACAQUE-RETINA datasets are shown in Table 3. We employ the same hyperparameters for scalarization as its Pareto MTL counterpart. We use Adam optimiser (a first-order stochastic optimizer) with the parameter $\epsilon = 0.01$ which improves the numerical stability of the optimizer and other parameters at their default values. The batch size for TM-MARROW and MACAQUE-RETINA are 128 and 256 respectively.

Evaluating the estimated Pareto front

Hypervolume

In Additional file 1: Fig S3, we illustrate the hypervolume of a Pareto front. First, a reference point is chosen which has larger value in at least one dimension with no smaller values in all other dimensions than all the estimated points. Then each estimated point forms a rectangle with the reference point. The area of the union of all these rectangles is the hypervolume. Note that the reference point is shared among the methods to allow for proper comparison of Pareto front approximations.

NDC

In Additional file 1: Fig S3, we demonstrate how to obtain NDC of a Pareto front. For simplicity, in a set of Pareto candidates, suppose the range between maximum and minimum values of the loss L_1 and the similar range for the loss L_2 are both divisible by a pre-specified value μ. Then the ranges for the two losses are divided into a grid of squares with width μ. Each square is an indifference region. If there is at least one non-dominated point in the indifference region, the NDC for that region is one,

otherwise 0. The final NDC is the sum of NDC for each indifference region. Note that the grids of squares are shared among the methods to allow for proper comparison of Pareto front approximations.

Single cell RNA-seq datasets

We examined two datasets to evaluate the efficacy of Pareto MTL with MINE for Pareto front estimation. The TM-MARROW dataset is an integration of two bone marrow datasets (MarrowTM-10x, MarrowTM-ss2) from the Tabula Muris project [12, 13]. The read counts in MarrowTM-ss2 are first standardized by mouse gene length and only cells with total count larger than 100 are selected. Then for both MarrowTM-10x and MarrowTM-ss2, the genes are scaled to unit variance and only genes with positive mean are selected. Finally, data selected from MarrowTM-10x and MarrowTM-ss2 are concatenated based on the intersection of their gene names.

The MACAQUE-RETINA dataset, which consists of raw macaque single cell count data, as well as its metadata are downloaded from the Single Cell Portal website [20]. As in [18], we only focus on the 30,302 bipolar cells in the total 165,679 cells from macaque retina. There are different levels of batch effect (sample, region and animal) and we only considered the regional batch effect which includes fovea and periphery of retina. As in TM-MARROW, for both the fovea and periphery part of the raw scRNA-seq data, the genes are scaled to unit variance and only genes with positive mean are selected. Then data selected from the fovea and periphery part are concatenated based on the intersection of their gene names.

Abbreviations

scRNA-seq	Single-cell RNA sequence
scVI	Single-cell variational inference
MINE	Mutual Information Neural Estimator
MMD	Maximum mean-embedding discrepancy
Pareto MTL	Pareto multi-task learning
ELBO	Evidence lower bound
ASW	Averaged silhouette width
ARI	Adjusted rand index
NMI	Normalized mutual information
UC	Unsupervised clustering accuracy
NN	Nearest neighbor
BE	Batch entropy

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-05003-3.

Additional file 1. Supplementary tables and figures.

Acknowledgements

This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200.

Author contributions

SW led the conceptual design of the study. HL implemented the methodology and performed all experiments. SW and HL performed the majority of the writing. HS and DJM contributed to supervision of the study. All authors discussed the results and commented on the manuscript. All authors read and approved the final manuscript.

Funding

This research is supported by the Australian Research Council Discovery Early Career Award received by Dr. Susan Wei (Project Number DE200101253) funded by the Australian Government.

Availability of data and materials
The datasets analysed during the current study are publicly available, large, and obtained as described in the Single cell RNA-seq datasets section. All the developed code is available in the repository: https://github.com/suswei/single-cell-rna-seq.

Declarations

Ethics approval and consent to participate
No ethics approval and consent were required for the study.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 16 July 2022 Accepted: 25 October 2022
Published online: 03 November 2022

References

1. Hotelling H. Analysis of a complex of statistical variables into principal components. J Educ Psychol. 1933;24(6):417.
2. Pierson E, Yau C. Zifa: dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biol. 2015;16(1):1–10.
3. Maaten Lvd, Hinton G. Visualizing data using t-sne. J Mach Learn Res. 2008;9:2579–605.
4. McInnes L, Healy J, Melville J. Umap: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018.
5. Moon KR, van Dijk D, Wang Z, Gigante S, Burkhardt DB, Chen WS, Yim K, Avd E, Hirn MJ, Coifman RR, et al. Visualizing structure and transitions in high-dimensional biological data. Nat Biotechnol. 2019;37(12):1482–92.
6. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods. 2018;15(12):1053–8.
7. Amodio M, Van Dijk D, Srinivasan K, Chen WS, Mohsen H, Moon KR, Campbell A, Zhao Y, Wang X, Venkataswamy M, et al. Exploring single-cell data with deep multitasking neural networks. Nat Methods. 2019;16(11):1139–45.
8. Lin X, Zhen H-L, Li Z, Zhang Q-F, Kwong S. Pareto multi-task learning. In: Advances in neural information processing systems, 2019, pp. 12060–12070.
9. Emmerich MT, Deutz AH. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Nat Comput. 2018;17(3):585–609.
10. Lopez R, Regier J, Jordan MI, Yosef N. Information constraints on auto-encoding variational bayes. In: Advances in neural information processing systems, 2018, pp. 6114–6125.
11. Belghazi MI, Baratin A, Rajeswar S, Ozair S, Bengio Y, Courville A, Hjelm RD. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062, 2018.
12. Xu C, Lopez R, Mehlman E, Regier J, Jordan MI, Yosef N. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. 2020; bioRxiv 532895.
13. Schaum N, Karkanias J, Neff NF, May AP, Quake SR, Wyss-Coray T, Darmanis S, Batson J, Botvinnik O, Chen MB et al. Single-cell transcriptomic characterization of 20 organs and tissues from individual mice creates a tabula muris. 2018; BioRxiv 237446.
14. Zitzler E, Thiele L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: International conference on parallel problem solving from nature, 1998; pp. 292– 301.
15. Wu J, Azarm S. Metrics for quality assessment of a multiobjective design optimization solution set. J Mech Des. 2001;123(1):18–25.
16. Audet C, Bigeon J, Cartier D, Le Digabel S, Salomon L. Performance indicators in multiobjective optimization. Optimization Online, 2018.
17. Ross BC. Mutual information between discrete and continuous data sets. PLoS ONE. 2014;9(2):87357.
18. Li X, Wang K, Lyu Y, Pan H, Zhang J, Stambolian D, Susztak K, Reilly MP, Hu G, Li M. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. Nat Commun. 2020;11(1):1–14.
19. Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A. A kernel two-sample test. J Mach Learn Res. 2012;13(25):723–73.
20. Peng Y-R, Shekhar K, Yan W, Herrmann D, Sappington A, Bryman GS, van Zyl T, Do MTH, Regev A, Sanes JR. Molecular classification and comparative taxonomics of foveal and peripheral cells in primate retina. Cell. 2019;176(5):1222–37.

Publisher's Note