Genome Biology

RESEARCH Open Access

MichiGAN: sampling from disentangled representations of single-cell data using generative adversarial networks

Hengshi Yu[1] and Joshua D. Welch[2,3]*

*Correspondence:
welchjd@umich.edu
[2]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, USA
[3]Department of Computer Science and Engineering, University of Michigan, Ann Arbor, USA
Full list of author information is available at the end of the article

Abstract

Deep generative models such as variational autoencoders (VAEs) and generative adversarial networks (GANs) generate and manipulate high-dimensional images. We systematically assess the complementary strengths and weaknesses of these models on single-cell gene expression data. We also develop MichiGAN, a novel neural network that combines the strengths of VAEs and GANs to sample from disentangled representations without sacrificing data generation quality. We learn disentangled representations of three large single-cell RNA-seq datasets and use MichiGAN to sample from these representations. MichiGAN allows us to manipulate semantically distinct aspects of cellular identity and predict single-cell gene expression response to drug treatment.

Keywords: Cellular identity, Disentangled representations, Generative adversarial networks, Representation learning, Single-cell genomics

Introduction

Deep learning techniques have recently achieved remarkable successes, especially in vision and language applications [1, 2]. In particular, state-of-the-art deep generative models can generate realistic images or sentences from low-dimensional latent variables [3]. The generated images and text data are often nearly indistinguishable from real data, and data generating performance is rapidly improving [4, 5]. The two most widely types of deep generative models are variational autoencoders (VAEs) and generative adversarial networks (GANs). VAEs use a Bayesian approach to estimate the posterior distribution of a probabilistic encoder network, based on a combination of reconstruction error and the prior probability of the encoded distribution [6]. In contrast, the GAN framework consists of a two-player game between a generator network and a discriminator network [7]. GANs and VAEs possess complementary strengths and weaknesses: GANs generate much better samples than VAEs [8], but VAE training is much more stable and learns more useful "disentangled" latent representations [9]. GANs outperform VAEs in generating sharp image samples [7], while VAEs tend to generate blurry images [10]. GAN training is generally less stable than VAE training, but some recent derivations of GAN like Wasserstein GAN [11–13] significantly improve the stability of GAN training, which is particularly helpful for non-image data.

Achieving a property called "disentanglement", in which each dimension of the latent representation controls a semantically distinct factor of variation, is a key focus of recent research on deep generative models [14–20]. Disentanglement is important for controlling data generation and generalizing to unseen latent variable combinations. For example, disentangled representations of image data allow prediction of intermediate images [21] and mixing images' styles [22]. For reasons that are not fully understood, VAEs generally learn representations that are more disentangled than other approaches [23–28]. The state-of-the-art methods for learning disentangled representations capitalize on this advantage by employing modified VAE architectures that further improve disentanglement, including β-VAE, FactorVAE, and β-TCVAE [9, 29–31]. In contrast, the latent space of the traditional GAN is highly entangled. Some modified GAN architectures, such as InfoGAN [32], encourage disentanglement using purely unsupervised techniques, but these approaches still do not match the disentanglement performance of VAEs [33–40].

Disentanglement performance is usually quantitatively evaluated on standard image datasets with known ground truth factors of variation [41–44]. In addition, disentangled representations can be qualitatively assessed by performing traversals or linear arithmetic in the latent space and visually inspecting the resulting images [45–49].

Recently, molecular biology has seen the rapid growth of single-cell RNA-seq technologies that can measure the expression levels of all genes across thousands to millions of cells [50]. Like image data, for which deep generative models have proven so successful, single-cell RNA-seq datasets are large and high-dimensional. Thus, it seems likely that deep learning will be helpful for single-cell data. In particular, deep generative models hold great promise for distilling semantically distinct facets of cellular identity and predicting unseen cell states.

Several papers have already applied VAEs [51–61] and GANs [62] to single-cell data. A representative VAE method is scGen, which uses the same objective function as β-VAE [9]. The learned latent values in scGen are utilized for out-of-sample predictions by latent space arithmetic. The cscGAN paper adapts the Wasserstein GAN approach for single-cell data and shows that it can generate realistic gene expression profiles, proposing to use it for data augmentation.

Assessing disentanglement performance of models on single-cell data is more challenging than image data, because humans cannot intuitively understand the data by looking at it as with images. Previous approaches such as scGen have implicitly used the properties of disentangled representations [51], but disentanglement performance has not been rigorously assessed on single-cell data.

Here, we systematically assess the disentanglement and generation performance of deep generative models on single-cell RNA-seq data. We show that the complementary strengths and weaknesses of VAEs and GANs apply to single-cell data in a similar way as image data. We develop MichiGAN, a neural network that combines the strengths of

VAEs and GANs to sample from disentangled representations without sacrificing data generation quality. We employ MichiGAN and other methods on simulated single-cell RNA-seq data [63, 64] and provide quantitative comparisons through several disentanglement metrics [29, 30]. We also learn disentangled representations of three real single-cell RNA-seq datasets [65–67] and show that the disentangled representations can control semantically distinct aspects of cellular identity and predict unseen combinations of cell states.

Our work builds upon that of Lotfollahi et al. [51], who showed that a simple VAE (which they called scGen) can predict single-cell perturbation responses. They also showed several specific biological contexts in which this type of approach is useful. First, they predicted the cell-type-specific gene expression changes induced by treating immune cells with lipopolysaccharide. Second, they predicted the cell-type-specific changes that occur when intestinal epithelial cells are infected by *Salmonella* or *Heligmosomoides polygyrus*. Finally, they showed that scGen can use mouse data to predict perturbation responses in human cells or across other species. For such tasks, one can gain significant biological insights from the generated scRNA-seq profiles.

Our method, MichiGAN, can make the same kinds of predictions and yield the same kinds of biological insights as scGen, but we show that MichiGAN has significant benefits compared to scGen (including disentanglement and data generation performance). In addition, we show that MichiGAN can predict single-cell response to drug treatment, a biological application that was not demonstrated in the scGen paper.

Results

Variational autoencoders learn disentangled representations of single-cell data

Real single-cell datasets usually have unknown, unbalanced, and complex ground-truth variables, and humans cannot readily distinguish single-cell expression profiles by eye, making it difficult to assess disentanglement performance by either qualitative or quantitative evaluations. We thus first performed simulation experiments to generate balanced single-cell data with several data generating variables using the Splatter R package [63]. All the datasets were processed using the SCANPY software [68]. We measured the disentanglement performances of different methods on the simulated single-cell data using several disentanglement metrics and also provided qualitative evaluations on the learned representations using the real datasets.

We first estimated simulation parameters to match the Tabula Muris dataset [65]. Then, we set the differential expression probability, factor location, factor scale, and common biological coefficient of variation to be $(0.5, 0.01, 0.5, 0.1)$. We then used Splatter [63] to simulate gene expression data of 10,000 cells with four underlying ground-truth variables: batch, path, step, and library size. Batch is a categorical variable that simulates linear differences among biological or technical replicates. Step represents the degree of progression through a simulated differentiation process, and path represents different branches of the differentiation process. We simulated two batches, two paths, and 20 steps. The batch and path variables have linear effects on the simulated expression data, while the step variable can be related either linearly or non-linearly to the simulated gene expression values. We tested the effects of this variable by separately simulating a purely linear and a non-linear differentiation process. We also included library size, the total number of expressed mRNAs per cell, as a ground truth variable. A UMAP plot of the simulated data shows that the four ground truth variables each have complementary and distinct effects on the resulting gene expression state (Fig. 1a and Additional file 1: Figure S1a).

We compared the disentanglement performance of three methods: probabilistic principal component analysis (PCA) [69], β-VAE, and β-TCVAE. The probabilistic PCA method assumes a linear relationship between data and representations, while VAE and β-TCVAE can learn non-linear representations. Note that we use probabilistic PCA to allow calculation of mutual information (see below). The β-TCVAE approach penalizes the total correlation of the latent representation, directly minimizing the mutual information between latent dimensions, which has been shown to significantly improve disentanglement performance on image data.

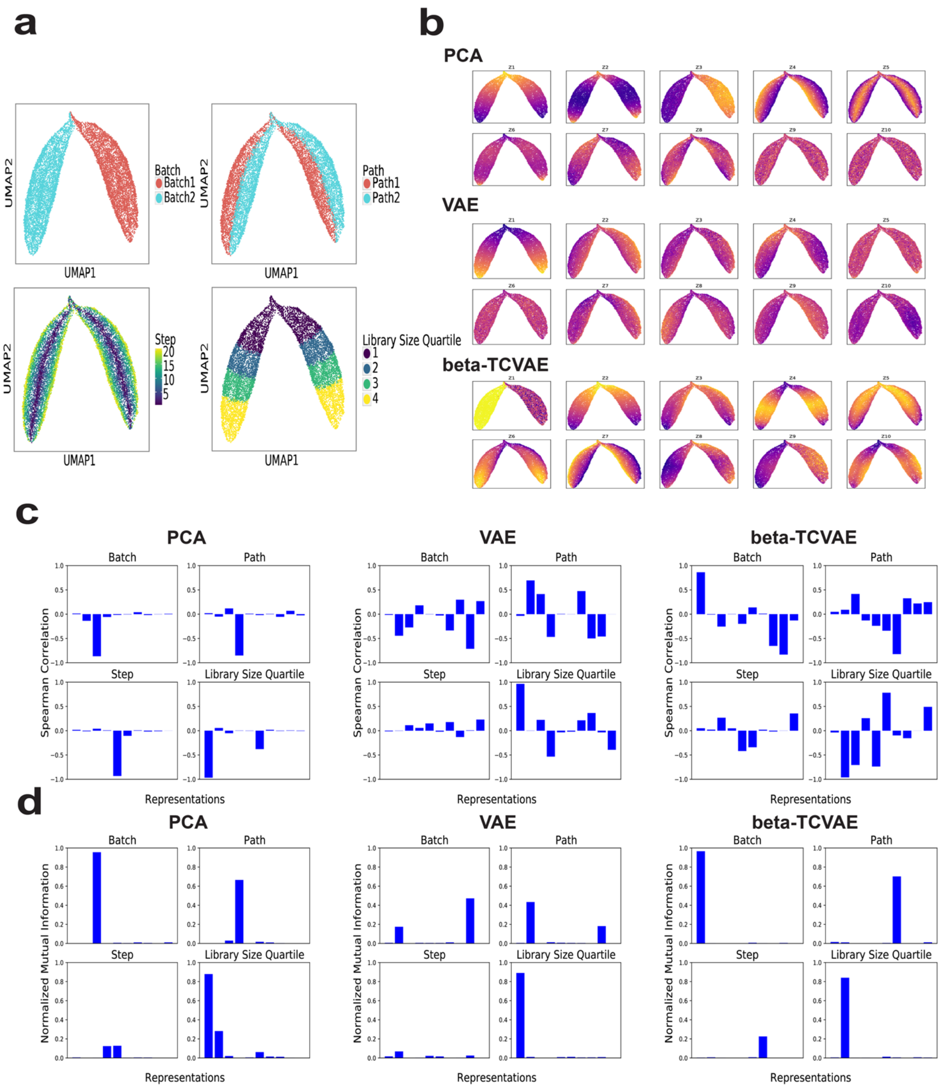

Fig. 1 Evaluating disentanglement performance on simulated data with non-linear step. **a** UMAP plots of simulated data colored by batch, path, step, and library size quartile. **b** UMAP plots of data colored by the ten latent variables learned by PCA, VAE, and β-TCVAE. **c** Bar plots of Spearman correlations between ten latent variables and each of the four ground-truth variables for PCA, VAE, and β-TCVAE. **d** Bar plots of normalized mutual information between ten representations and each of the four ground-truth variables for PCA, VAE, and β-TCVAE

We used the three methods to learn a 10-dimensional latent representation of the simulated data (Fig. 1b and Additional file 1: Figure S1b). Some latent variables learned by each method showed clear relationships with the ground-truth variables. For example, the first latent variable Z1 from PCA seemed related to library size, and Z3, Z4, and Z5 were related to batch, path, and step, respectively. The VAE representations similarly showed some relationships with the ground-truth variables. Based on the UMAP plots, the latent variables from β-TCVAE appeared to show the strongest and most clear relationships with the ground-truth variables.

To quantify the disentanglement performance of the three methods, we calculated Spearman correlation and normalized mutual information between each representation and a ground-truth variable (Fig. 1c, d). Spearman correlation measures the strength of monotonic relatedness between two random variables. The normalized mutual information, on the other hand, is a more general and robust metric of statistical dependence. A disentangled representation should have a bar plot with only four distinct bars in this case, indicating that each ground-truth variable was captured by exactly one latent variable. PCA showed the best performance as measured by Spearman correlation (Fig. 1c), likely because the metric does not fully characterize the complex statistical dependency between true and inferred latent variables for the VAE methods, which learn more complex non-linear relationships. Based on the normalized mutual information metric, both the PCA and VAE representations achieved some degree of disentanglement, but neither approach fully disentangled all ground-truth variables. Multiple PCA representations had measurable mutual information with step and library size quartile, while multiple VAE representations identified batch and path and none of the VAE representations identified step. In contrast, exactly one β-TCVAE representation had significant mutual information for each ground-truth variable. Also, β-TCVAE was the only method with a unique representation for the non-linear step variable.

We also computed the Spearman correlation and normalized mutual information for the simulated data with linear step (Additional file 1: Figure S1c-d). The results for the simulated data with linear step were similar and β-TCVAE did the best at identifying only one representation for each ground-truth variable.

We further calculated the mutual information gap (MIG) metric used in [30] and FactorVAE disentanglement metric [29] to measure disentanglement. The MIG metric is defined as the average gap between the mutual information of the two latent variables that are most related to each ground-truth variable. If there is a single latent variable that has high mutual information with each ground-truth variable, the MIG will be high. The FactorVAE metric is based on the error rate of a linear classifier that identifies which ground truth variable differs based data points using latent dimensions. In addition, we calculated a Spearman correlation gap similar to MIG. Table 1 summarizes the correlation gap, FactorVAE metric, and MIG of the three models over 5 runs for the two simulated datasets. As expected from the bar charts, the PCA representations have the largest Spearman correlation gap and β-TCVAE has the largest MIG, showing the best disentanglement performance for both simulated datasets. The FactorVAE metric also shows that β-TCVAE has the best disentanglement performance. We also evaluated InfoWGAN-GP on the simulated data in Additional file 1: Figure S4 and found that the representations are entangled with the ground-truth variables for simulated datasets with linear and non-linear step.

Table 1 Disentanglement metrics for two splatter-simulated single-cell RNA-seq datasets with four ground truth variables

		Spearman correlation gap ↑	**FactorVAE metric ↑**	**MIG ↑**
Linear step	PCA	**0.68** ±0.00	0.35 ±0.01	0.54 ±0.00
	VAE	0.3 ±0.04	0.4 ±0.02	0.48 ±0.13
	β-TCVAE	0.18 ±0.05	**0.48** ±0.03	**0.72** ±0.02
Non-linear step	PCA	**0.72** ±0.00	0.35 ±0.01	0.55 ±0.00
	VAE	0.27 ±0.07	0.41 ±0.02	0.43 ±0.08
	β-TCVAE	0.16 ±0.06	**0.51** ±0.04	**0.66** ±0.16

The mean and standard deviation over 5 runs are presented for each method. The dimensionality of the latent space was 10 for all three approaches

We also evaluated the disentanglement performance of the three methods with four latent dimensions (the same as the number of ground-truth variables), for the simulated datasets in Additional file 1: Figures S8 and S9. The β-TCVAE representations still most effectively disentangle the ground-truth variables. Table 2 summarizes the disentanglement metrics of the three methods with four latent dimensions. Although FactorVAE metric shows similar values for the three methods, β-TCVAE consistently has much higher MIG than PCA and VAE.

In addition, we utilized the PROSSTT package [64] to simulate three single-cell datasets. PROSSTT simulates cells undergoing a continuous process such as differentiation. As shown in Additional file 1: Figures S10a, S11a and S12a, the three PROSSTT-simulated datasets have 3-, 4-, or 5-way branching trajectories, respectively. The three PROSSTT-simulated datasets also have a continuous time variable. We use three ground-truth variables (branch, time, and library size) to calculate mutual information with the learned latent variables (Additional file 1: Figures S10b, S11b, and S12b). PCA and VAE have multiple latent dimensions with moderate mutual information with branch and time quartile, while β-TCVAE captures each of these quantities mostly in a single variable. We also summarized the disentanglement metrics of the three methods on the PROSSTT-simulated datasets in Table 3. β-TCVAE has the highest FactorVAE metric and MIG for each of the three datasets.

In summary, our assessment indicates that β-TCVAE most accurately disentangles the latent variables underlying single-cell data, consistent with its previously reported superior disentanglement performance on image data [30].

GANs generate more realistic single-cell expression profiles than VAEs

We next evaluated the data generating performance of several deep generative models including VAE, β-TCVAE, and Wasserstein GAN with gradient penalty (WGAN-GP), as

Table 2 Disentanglement metrics for two splatter-simulated single-cell RNA-seq datasets with four ground truth variables

		Spearman correlation gap ↑	**FactorVAE metric ↑**	**MIG ↑**
Linear step	PCA	0.57	0.36	0.56
	VAE	0.37	**0.44**	0.39
	β-TCVAE	**0.65**	0.33	**0.72**
Non-linear step	PCA	**0.60**	0.36	0.58
	VAE	0.4	**0.38**	0.38
	β-TCVAE	0.55	0.34	**0.73**

The dimensionality of the latent space was 4 for all three approaches

Table 3 Disentanglement metrics for three PROSSTT-simulated single-cell RNA-seq datasets with three ground truth variables

		FactorVAE metric ↑	MIG ↑
3 trajectories	PCA	0.54	0.10
	VAE	0.58	0.08
	β-TCVAE	**0.64**	**0.27**
4 trajectories	PCA	0.59	0.12
	VAE	0.61	0.12
	β-TCVAE	**0.72**	**0.15**
5 trajectories	PCA	0.59	0.06
	VAE	0.53	0.06
	β-TCVAE	**0.62**	**0.26**

well as traditional methods of PCA and Gaussian mixture models (GMM) on the Tabula Muris dataset [65]. This dataset contains a comprehensive collection of single-cell gene expression profiles from nearly all mouse tissues and thus represents an appropriate dataset for evaluating data generation, analogous to the ImageNet dataset in computer vision. We also measured data generation performance on a subset of the Tabula Muris containing only cells from the mouse heart. We used two metrics to assess data generation performance: random forest error and inception score. Random forest error was introduced in the cscGAN paper [62] and quantifies how difficult it is for a random forest classifier to distinguish generated cells from real cells. A higher random forest error indicates that the generated samples are more realistic. We also computed inception score [70], a metric commonly used for quantifying generation performance on image data. Intuitively, to achieve a high inception score, a generative model must generate every class in the training dataset (analogous to recall) and every generated example must be recognizable as belonging to a particular class (analogous to precision).

We show the random forest errors over 5 runs of VAE, β-TCVAE, and WGAN-GP during training for the Tabula Muris heart subset and the whole Tabula Muris in Fig. 2a and b. We also evaluate simpler generative models, including PCA and GMM. WGAN-GP achieves the best generation performance, as measured by both metrics, on both the subset and full dataset. The deep generative models significantly outperform PCA and GMM. VAE achieves second-best generating performance and, as expected with an endeavor to pursue more disentangled representation, the quality of β-TCVAE generation is the worst of the three approaches. Figure 2c, d shows the inception scores over 5 runs for the two datasets; this metric reveals the same trend as with random forest errors, indicating that WGAN-GP has the best generation performance and β-TCVAE generates the least realistic data. Additionally, the generation performance of the GAN is still significantly higher than that of the VAE even for the smaller Tabula Muris heart dataset. These results accord well with previous results from the image literature, indicating that GANs generate better samples than VAEs, and VAE modifications to encourage disentanglement come at the cost of sample quality.

MichiGAN samples from disentangled representations without sacrificing generation performance

Having confirmed that VAEs achieve better disentanglement performance, but GANs achieve better generation performance, we sought to develop an approach that combines

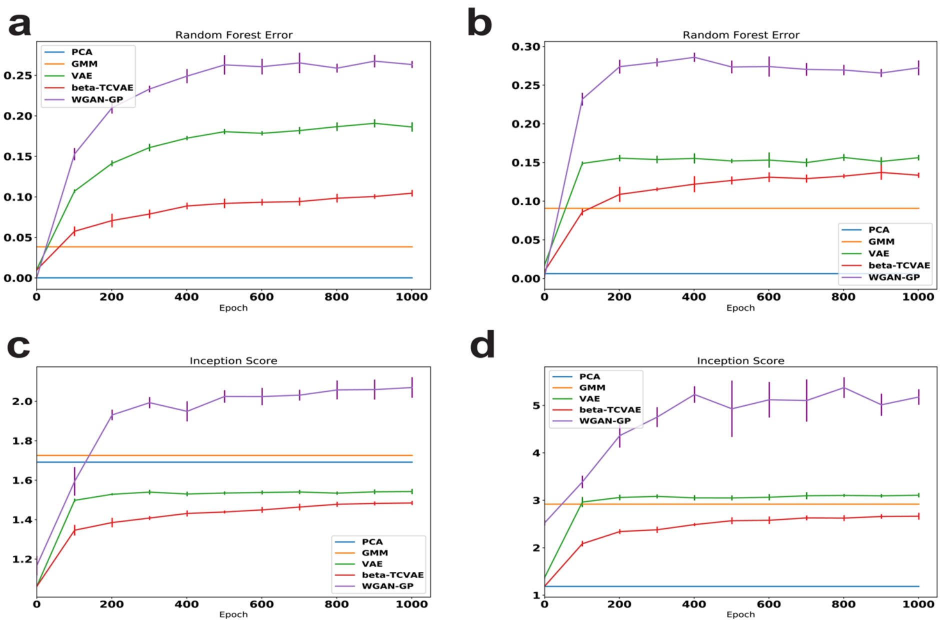

Fig. 2 Generation performance of VAE, β-TCVAE, WGAN-GP, PCA, and GMM on the Tabula Muris heart data and the whole Tabula Muris data. **a** Random forest error for the five methods on the Tabula Muris heart data during training. **b** Random forest error for the five methods on the whole Tabula Muris data during training. **c** Inception score for the five methods on the Tabula Muris heart data during training. **d** Inception score for the five methods on the whole Tabula Muris data during training. Error bars indicate standard deviation across five runs. For clarity, the error bars for PCA and GMM are omitted because of their small and large variability

the strengths of both techniques. Several previous approaches have combined variational and adversarial techniques [10, 71, 72]. However, when we tested these approaches on single-cell data, we found that attempts to jointly perform variational and adversarial training compromised both training stability and generation performance. We also investigated the InfoGAN and semi-supervised InfoGAN, but found that the disentanglement performance was still significantly worse than that of the VAE approaches as shown in Additional file 1: Figure S4.

We thus developed a different approach: we first train a VAE to learn a disentangled representation. Then, we use the VAE encoder's latent representation z for each cell x as a given code and train a conditional GAN using the (z, x) pairs. After training, we can generate high-quality samples from the VAE's disentangled representation. Importantly, the training is no less stable than training VAE and GAN separately, and the GAN generation quality is not compromised by a regularization term encouraging disentanglement. In addition, any kind of representation—from nonlinear methods like VAEs or linear methods like PCA—can be incorporated in our approach. Wanting to follow the convention that the names of many generative adversarial networks end with "GAN", but unable to devise a compelling acronym, we named our approach MichiGAN after our institution.

The MichiGAN architecture is shown in Fig. 3 and also summarized in Algorithm 1. We find that MichiGAN effectively achieves our goal of sampling from a disentangled representation without compromising generation quality (see results below). Several previous approaches have combined variational and adversarial techniques, including VAEGAN [10], adversarial symmetric variational autoencoder [71], and adversarial variational Bayes [72]. InfoGAN and semi-supervised InfoGAN are also conceptually related to

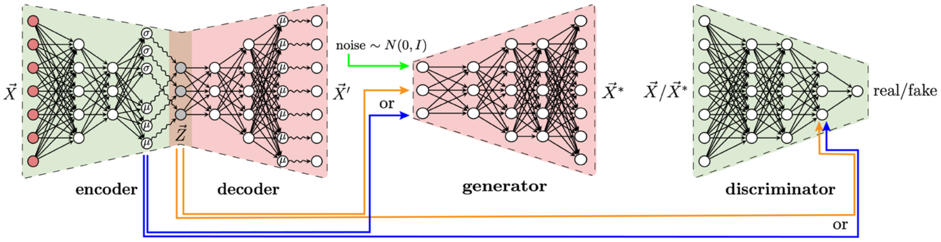

Fig. 3 Overview of the MichiGAN architecture. We first train a model, such as β-TCVAE, to learn a disentangled representation of the real data. We then use the resulting latent codes to train a conditional GAN with projection discriminator, so that the GAN generator becomes like a more accurate decoder. Because the VAE and GAN are trained separately, training is just as stable as training each one individually, but the combined approach inherits the strengths of each individual technique. After training, we can generate high-quality samples from the disentangled representation using the GAN generator

MichiGAN, but we found that none of these previous approaches produced good results on single-cell data. While we were writing this paper, another group released a preprint with an approach called ID-GAN, which also uses a pre-trained VAE to learn a disentangled representation [40]. However, they use the reverse KL divergence framework to enforce mutual information between the VAE representation and the generated data, which we previously tested and found does work as well as a conditional GAN with projection discriminator [73]. Furthermore, ID-GAN uses a convolutional architecture and classic GAN loss for image data, whereas we use a multilayer perceptron architecture and Wasserstein loss for single-cell expression data.

Although our approach is conceptually simple, there are several underlying reasons why it performs so well, and recognizing these led us to pursue this approach. First, training a conditional GAN maximizes mutual information between the condition variable and the generated data. This is a similar intuition as the InfoGAN, but unlike InfoGAN, MichiGAN does not need to learn its own codes, and thus the discriminator can focus exclusively on enforcing the relationship between code and data. A nearly optimal discriminator is crucial for maximizing this mutual information, but the Wasserstein loss also has this requirement, and we meet it by training the discriminator 5 times for every generator update. Second, the adversarial loss allows the GAN generator to capture complex, multi-modal distributional structure that cannot be modeled by the factorized Gaussian distribution of the VAE decoder. This is particularly helpful if multiple distinct types of cells map to a similar latent code, in which case the unimodal Gaussian distribution of the VAE decoder will generate the average of these cell types. In contrast, even though the GAN generates from the same latent representation as the VAE, the GAN can fit complex, multimodal distributions by minimizing the Wasserstein distance between generated and true data distributions. Additionally, a data-dependent code (the posterior of the VAE encoder) allows the GAN to generate from a flexible latent space that reflects the data distribution, rather than an arbitrary distribution such as the commonly used standard normal. We believe this inflexibility contributes significantly to the relatively poor disentanglement performance of InfoGAN. For example, InfoGAN is highly sensitive to the number and distribution chosen for the latent codes; if classes are imbalanced in the real data but the prior has balanced classes, it cannot learn a categorical variable that reflects the true proportions.

Based on the results from our disentanglement comparison (see below), we chose to use the β-TCVAE to learn the latent representation for MichiGAN. We then use either the posterior means or the random samples from the posterior as the condition for the GANs; both choices have been utilized to evaluate disentanglement performance in previous studies [9, 29, 30].

The last step of MichiGAN involves training a conditional GAN. We found that a conditional Wasserstein GAN with projection discriminator [73] and gradient penalty [12] is most effective at enforcing the condition. We also assessed semi-supervised InfoGAN [74] and a conditional GAN based on simple concatenation, but found that these were less effective at enforcing the relationship between code and generated data (Additional file 1: Figure S7) and less stable during training.

We evaluated the MichiGAN algorithm on the simulated single-cell data with the trained β-TCVAE models. Figure 4a shows the UMAP plots of real data colored by β-TCVAE latent representations and generated data colored by code using WGAN-GP and MichiGAN on the simulated data with non-linear step. The WGAN-GP representations are very entangled and none of the representations shows an identifiable coloring pattern. In contrast, the UMAP plots have consistent coloring patterns between the β-TCVAE and MichiGAN representations. Thus, the generator of MichiGAN preserves the relationship between latent code and data, effectively sampling from the disentangled representation learned by the β-TCVAE. Because there is no inference network for the generated data of either WGAN-GP or MichiGAN, we are unable to measure the mutual information for the generators. Therefore, we used Spearman correlation as an indicator of whether MichiGAN retains the relationship between disentangled latent representation and data. Figure 4b also shows the bar plots of Spearman correlations between representations and variables for the three methods. We used the correlations between each representation and ground truth variables for β-TCVAE, WGAN-GP, and MichiGAN. For GAN models, we trained a k-nearest neighbor regressor ($k = 3$) for each variable based on the real data and predicted the variables for the generated data. The WGAN-GP representations do not show large correlation with any inferred ground-truth variable. In contrast, the representations for β-TCVAE and MichiGAN show nearly identical correlations to the true variables in the real data and predicted variables in the generated data, respectively.

We also trained MichiGAN using PCA to obtain the latent code, instead of β-TCVAE. Additional file 1: Figure S3a-b show the UMAP plots of real data colored by the PCA representations and generated data colored by the MichiGAN-PCA representations on the two simulated datasets. In addition, Additional file 1: Figure S3c-d show nearly identical Spearman correlation bar plots between PCA and MichiGAN. MichiGAN trained with principal components preserves the relationship between the latent representations and real data, underscoring the generalizability of our approach.

We present the UMAP plots colored by the representations as well as bar plots of correlations for the simulated data with linear step in Additional file 1: Figure S2a-b. The results for the simulated data with linear step also indicate that MichiGAN restores the disentanglement performance of β-TCVAE, while the WGAN-GP representations are entangled. We further summarize the correlation gaps for the three methods on two simulated datasets in Table 4. For each simulated dataset, the MichiGAN and β-TCVAE have very similar correlation gaps and WGAN-GP has a very small correlation gap, as expected.

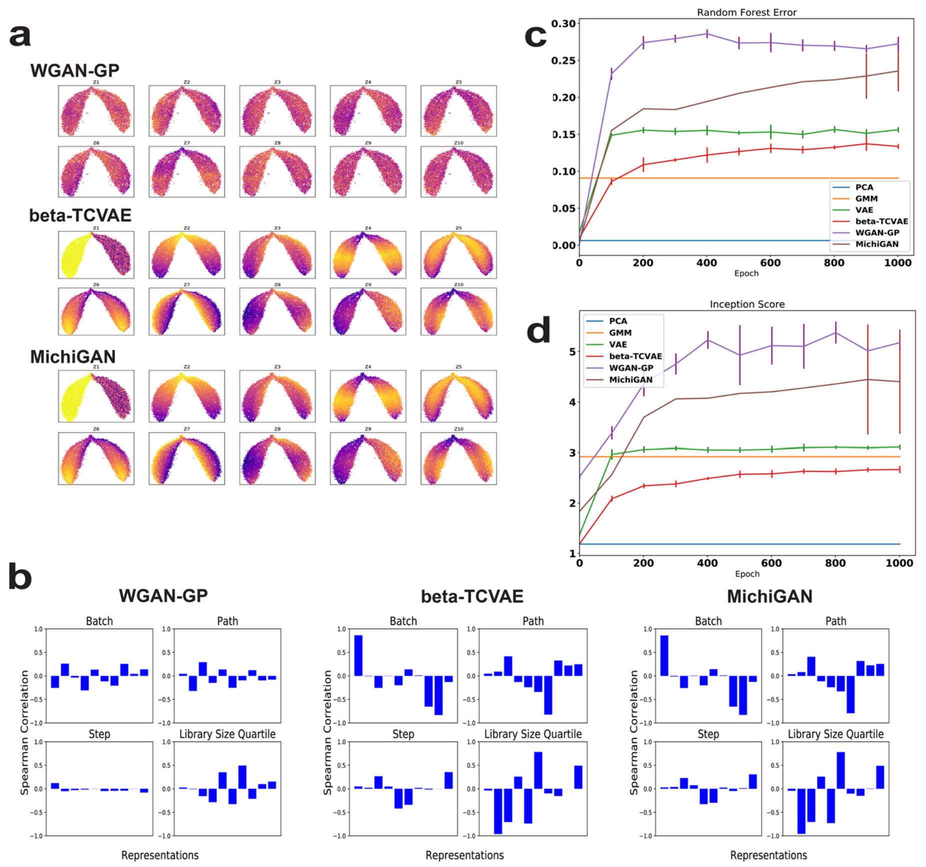

Fig. 4 Disentanglement and generation performance of WGAN-GP, β-TCVAE, and MichiGAN. **a** UMAP plots of real data colored by the ten representations of β-TCVAE and generated data colored by the ten representations of WGAN-GP and MichiGAN on the simulated data with non-linear step. The β-TCVAE panel is reproduced from Fig. 1b for clarity. **b** Bar plots of Spearman correlations between ten representations and each of the four ground-truth or inferred variables for WGAN, β-TCVAE and MichiGAN on the simulated data with non-linear step. The β-TCVAE panel is reproduced from Fig. 1c for clarity. **c** Random forest error of PCA, GMM, VAE, β-TCVAE, WGAN-GP, and MichiGAN on the whole Tabula Muris data during training. **d** Inception score of PCA, GMM, VAE, β-TCVAE, WGAN-GP, and MichiGAN on the whole Tabula Muris data during training. Error bars indicate standard deviation across 5 runs. For clarity, the error bars for MichiGAN are shown only for the last 100 epochs because the convergence speed in earlier epochs is variable, and the error bars for PCA and GMM are omitted because of their small and large variability

We evaluated MichiGAN on the whole Tabula Muris dataset (Fig. 4c, d). MichiGAN greatly improved the data generation performance based using the disentangled representations of β-TCVAE. The random forest error of MichiGAN was larger than VAE and nearly as good as the WGAN-GP, while still generating samples from a disentangled latent space.

Additionally, we applied PCA, GMM, VAE, β-TCVAE, WGAN-GP, and MichiGAN on the pancreas endocrinogenesis dataset [66]. We obtained the cells' latent time and cell cycle scores for G2M and S phases from [75]. Additional file 1: Figure S13a shows the UMAP plots of data colored by latent time and the difference between G2M and S scores. The β-TCVAE method gives qualitatively more disentangled representations (Additional file 1: Figure S13b), and gives much better disentanglement metrics (Additional file 1: Figure S13c). In addition, Additional file 1: Figure S13c also shows that MichiGAN significantly improves the data generation performance of β-TCVAE.

Table 4 Spearman correlation gap for WGAN-GP, InfoWGAN-GP, PCA, MichiGAN-PCA, VAE, β-TCVAE, and MichiGAN on the two splatter-simulated single-cell RNA-seq datasets

Model	Linear step	Non-linear step
WGAN-GP	0.07 ±0.02	0.10 ±0.06
InfoWGAN-GP	0.05 ±0.05	0.04 ±0.02
PCA	0.68 ±0.00	0.72 ±0.00
MichiGAN-PCA	0.65 ±0.01	0.68 ±0.00
VAE	0.3 ±0.04	0.27 ±0.07
β-TCVAE	0.18 ±0.05	0.16 ±0.06
MichiGAN	0.18 ±0.04	0.15 ±0.05

The mean and standard deviation are presented for each method over 5 runs

MichiGAN enables semantically meaningful latent traversals

Disentangled representations of images are often evaluated qualitatively by performing latent traversals, in which a single latent variable is changed holding the others fixed. Looking at the resulting changes in the generated images to see whether only a single semantic attribute changes provides a way of visually judging the quality of disentanglement. We wanted to perform a similar assessment of MichiGAN, but single-cell gene expression values are not individually and visually interpretable in the same way that images are. We thus devised a way of using UMAP plots to visualize latent traversals on single-cell data.

We performed latent traversals using both the Tabula Muris dataset and data from the recently published sci-Plex protocol [67]. After training on the Tabula Muris dataset (Additional file 1: Figure S5a), we chose a starting cell type, cardiac fibroblasts (Additional file 1: Figure S5b). We then varied the value of each latent variable from low to high, keeping the values of the other variables fixed to the latent embedding of a particular cell. For the sci-Plex dataset, which contains single-cell RNA-seq data from cells of three types (A549, K562, MCF7; Additional file 1: Figure S5c) treated with one of 188 drugs, we subsampled the data to include one drug treatment from each of 18 pathways by selecting the drug with the largest number of cells (Additional file 1: Figure S5b). This gives one treatment for each pathway; the numbers of cells for each combination are shown in Additional file 1: Table S1. We then performed latent traversals on cells with cell type MCF7 and treatment S7259 (Additional file 1: Figure S5e).

To visualize the traversals, we plotted each of the generated cells on a UMAP plot containing all of the real cells and colored each generated cell by the value of the latent variable used to generate it. Figure 5a and b show how traversing the latent variables concentrates the generated values on each part of the UMAP plots for Tabula Muris data using the first 10 dimensions of 128-dimensional WGAN-GP and MichiGAN, respectively. Figure 5c and d are the latent-traversal plots for the sci-Plex data using WGAN-GP and MichiGAN. As shown in Fig. 5b, all but three of the latent variables learned by the β-TCVAE behave like noise when we traverse them starting from the fibroblast cells, a property previously noted in assessments of disentangled latent variables learned by VAEs [29]. The remaining dimensions, Z3, Z6, and Z10, show semantically meaningful latent traversals. Latent variable Z3 shows high values for mesenchymal stem cells and fibroblasts, with a gradual transition to differentiated epithelial cell types from bladder, intestine, and pancreas at lower values of Z3. This is intriguing, because the mesenchymal-epithelial transition is a key biological process in normal development,

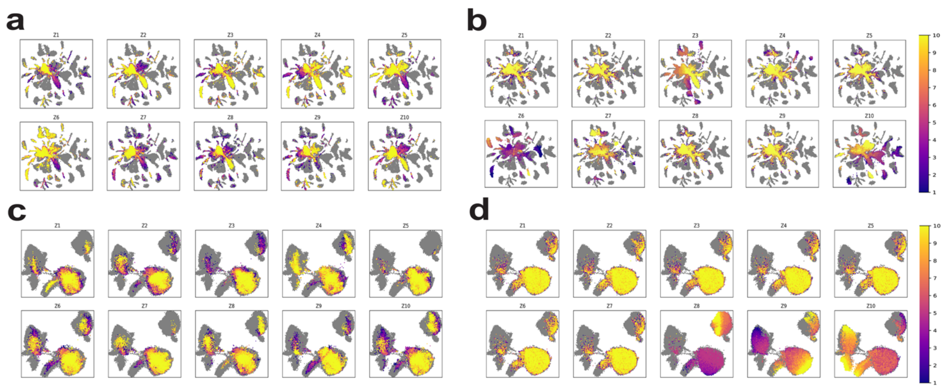

Fig. 5 Latent traversals of WGAN-GP and MichiGAN on Tabula Muris and sci-Plex datasets. **a** UMAP plot of latent traversals of the 10 representations of latent values that generate data closest to fibroblast cells in heart within the Tabula Muris data using WGAN-GP with 128 dimensions. **b** UMAP plot of latent traversals of the 10 representations of latent values of fibroblast cells in heart within the Tabula Muris data using MichiGAN. **c** UMAP plot of latent traversals of the 10 representations of latent values that generate data closest to MCF7-S7259 cells within the sci-Plex data using WGAN-GP with 128 dimensions. **d** UMAP plot of latent traversals of the 10 representations of latent values of MCF7-S7259 cells within the sci-Plex data using MichiGAN

wound healing, and cell reprogramming [76]. Latent variable Z6 generates mesenchymal and endothelial cells at low values, and mammary epithelial and cardiac muscle cells at high values. Latent variable Z10 is clearly related to immune function, generating immune cells at low and medium values and traversing from hematopoietic stem and progenitor cells to monocytes, T cells, and B cells. In contrast, latent traversals in the latent space of 128-dimensional WGAN-GP (Fig. 5a) do not show semantically meaningful changes along each dimension.

Figure 5d also shows that MichiGAN's latent traversals gives meaningful changes on the sci-Plex data. Latent variable Z8 has lower values on MCF7 cells and gradually transitions to higher values on K562 cells. In addition, latent variable Z9 also shows an A549-MCF7 transition with lower values on the A549 cells. The latent traversals of the 128-dimensional WGAN-GP, however, do not provide interpretable changes across the UMAP plot along each dimension. We also provide the latent traversals using 10-dimensional WGAN-GP for the two datasets in Additional file 1: Figure S6a-b and find that the latent traversals are still not semantically meaningful.

MichiGAN predicts single-cell gene expression changes under unseen drug treatments

One of the most exciting applications of disentangled representations is predicting high-dimensional data from unseen combinations of latent variables. We next investigated whether MichiGAN can predict single-cell gene expression response to drug treatment for unseen combinations of cell type and drug.

We trained MichiGAN on data from the recently published sci-Plex protocol. The dataset contains single-cell RNA-seq data from cells of three types (A549, K562, MCF7), each treated with one of 188 drugs. The drug is known for each scRNA-seq profile. We subsampled the data to include one drug treatment from each of 18 pathways by selecting the drug with the largest number of cells (Fig. 6a). We then have one treatment for each pathway; the numbers of cells for each combination are shown in Additional file 1:

Table S1. We also held out three drug/cell type combinations (A549-S1628, K562-S1096 and MCF7-S7259) to test MichiGAN's out-of-sample prediction ability.

We predict single-cell gene expression for each drug/cell type combination in a two-step process. First, we estimate the mean latent difference between the target cell type and another control cell type for other treatments using either posterior means or posterior samples from the β-TCVAE encoder. We then add the average latent difference to the latent values with the same treatment and the control cell type. This latent space vector arithmetic assumes the mean cell type latent differences are homogeneous across different treatments. Note that this assumption may not hold if there is a strong interaction effect between cell type and drug treatment.

Because there are a total of three cell types, we have a total of six predictions for the three held-out drug/cell type combinations. Figure 6b shows UMAP plots for these six

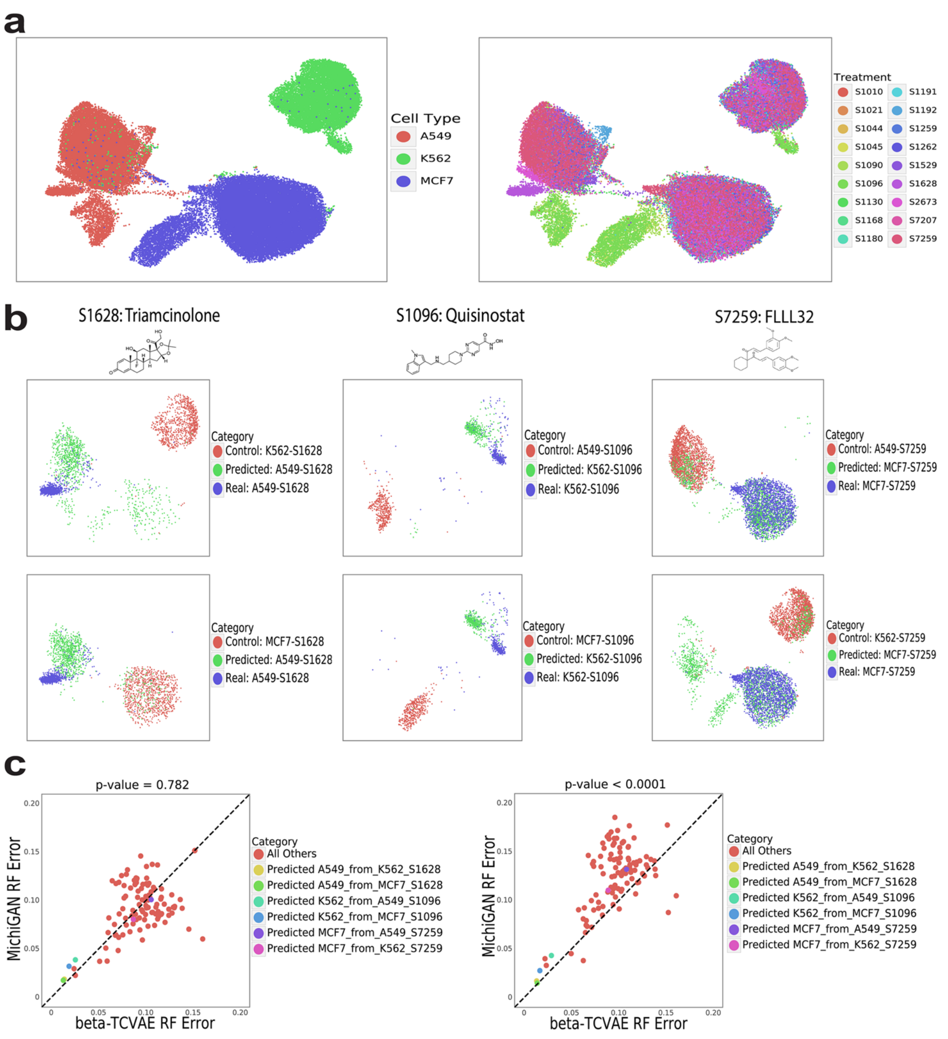

Fig. 6 Predicting single-cell gene expression effects of unseen drugs using MichiGAN. **a** UMAP plots of sci-Plex dataset colored by cell type (left) and treatment (right). **b** UMAP plots of the predicted (green), real (blue) and control (red) cells for 6 predictions of three missing cell type/drug combinations (A549-S1628, K562-S1096, and MCF7-S7259). **c** Random forest errors between MichiGAN and β-TCVAE for all combinations. MichiGAN was trained using mean representations (left) or representations sampled from the posterior distribution (right)

predictions. For all six predictions, the predicted values are closer to the true drug-treated cells on the UMAP plot than the control cells used to calculate the latent vector. However, the predicted cells do not overlap with the treated cells for the combinations A549-S1628 and K562-S1096, while the two predictions for MCF7-S7259 appear to be more accurate. For both β-TCVAE and MichiGAN, we measure their random forest errors between the real and predicted cells for each combination. The random forest scatter plots for sampled representations are shown in Fig. 6c. MichiGAN with sampled representations has significantly better random forest error than β-TCVAE ($p < 10^{-4}$, one-sided Wilcoxon test) and most of the points are above the diagonal line. We also show the random forest scatter plots for mean representations in Fig. 6c, which does not show significantly larger random forest errors compared to β-TCVAE ($p > 0.05$, one-sided Wilcoxon test) and might be due to the remaining correlations among mean representations of β-TCVAE [19]. Thus, MichiGAN with sampled representations is able to more accurately make predictions from latent space arithmetic than β-TCVAE. However, some of the six predictions for the missing combinations show low random forest errors from both methods, and some of the predictions from MichiGAN are only marginally better than those of β-TCVAE.

Accuracy of latent space arithmetic influences MichiGAN prediction accuracy

We next examined factors influencing the accuracy of MichiGAN predictions from latent space arithmetic. We suspected that the prediction accuracy might depend on the accuracy of the latent coordinates calculated by latent space arithmetic, which could vary depending, for example, on whether the drug exerts a consistent effect across cell types.

To investigate the reason for the difference in prediction accuracy, we developed a novel metric for assessing the accuracy of latent space arithmetic for a particular held-out cell type/perturbation combination. For a subset of the data $g(X)$ and the latent space $\tau(Z)$, we define the latent space entropy as:

$$H\left\{\tau(Z), g(X)\right\} = -E_{\tau(Z)}\left[\log E_{g(X)}\left\{q_{\phi}(Z \mid X) \mid Z\right\}\right].$$

Intuitively, H quantifies the concentration of Z with respect to X. We can then compare the entropy of the latent embeddings for the held-out data and the latent values predicted by latent space arithmetic by calculating $\Delta H = H\left\{\tau_{Fake}(Z), g(X)\right\} - H\left\{\tau_{Real}(Z), g(X)\right\}$, where τ_{Fake} is calculated by latent space arithmetic and τ_{Real} is calculated using the encoder. The quantity ΔH then gives a measure of how accurately latent space arithmetic predicts the latent values for the held-out data. If ΔH is positive, then the latent space prediction is less concentrated (and thus more uncertain) than the encoding of the real data.

The quantity ΔH measures how accurately latent space arithmetic predicts the latent values for the held-out data. Thus, we expect that MichiGAN should be able to more accurately predict drug/cell type combinations with a small ΔH.

As Fig. 7a shows, ΔH is significantly correlated with the difference in random forest error between MichiGAN and β-TCVAE, when sampling from either the posterior distribution of the latent representations or the posterior means. This supports our hypothesis that accuracy of the latent space arithmetic influences MichiGAN performance. To further test this, we selected the three drug/cell type combinations with the lowest overall ΔH values, and re-trained the network using all combinations except these three. Figure 7b shows the predicted, real and control cells for the six predictions of the

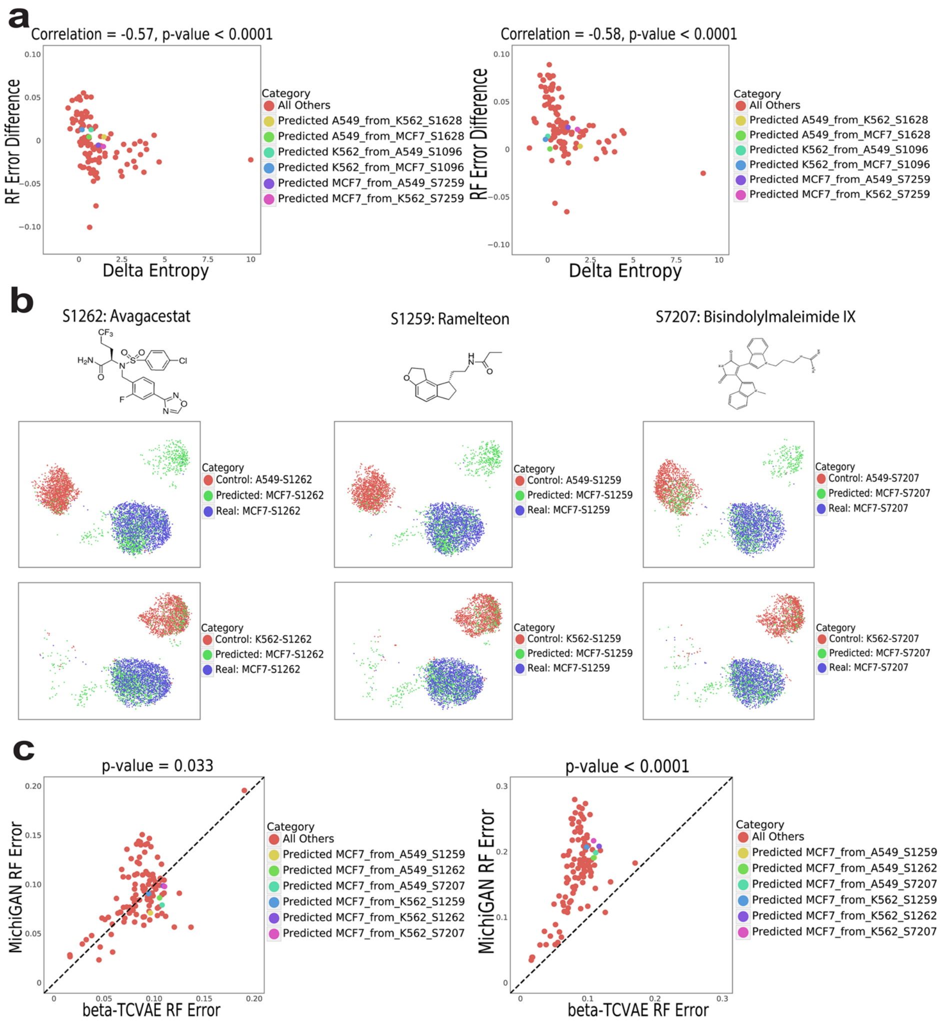

Fig. 7 MichiGAN predicts unseen or observed combinations in the large screen sci-Plex data. **a** Scatter plots of random forest errors' difference between MichiGAN and β-TCVAE versus delta entropy for MichiGAN with mean representations (left) and sampled representations (right) on the large screen sci-Plex data without three combinations of A549-S1628, K562-S1096 and MCF7-S7259. **b** UMAP plots of the predicted (green), real (blue) and control (red) cells for 6 predictions of the three missing combinations of MCF7-S1262, MCF7-S1259, and MCF7-S7207. **c** Random forest errors between MichiGAN and β-TCVAE for MichiGAN with mean representations (left) and sampled representations (right) after selecting held-out combinations with low ΔH

three new missing combinations based on MichiGAN using sampled representations. The predicted cells (green) overlap most parts of the real cells (blue) for all six predictions. As expected, MichiGAN predicted each of these low ΔH held-out combinations significantly more accurately than β-TCVAE (Fig. 7c).

We also compared the performance of VAE and MichiGAN trained with VAE on the sci-Plex data after holding out the selected drug/cell type combinations with lowest overall ΔH values in Additional file 1: Figure S15. MichiGAN trained with VAE gives accurate prediction of the unseen combinations (Additional file 1: Figure S15a), and also has significantly higher random forest error than that of VAE to predict different drug/cell type combinations using the latent space vector arithmetic algorithm (Additional file 1: Figure S15b).

Discussion

Our work provides fundamental evaluations of disentanglement performances of deep generative models on single-cell RNA-seq data. We show that combining GANs and VAEs can provide strong performance in terms of both data generation and disentanglement. MichiGAN provides an alternative to the current disentanglement learning literature, which focuses on learning disentangled representations through improved VAE-based or GAN-based methods, but rarely by combining them. Additionally, as the state of the art in disentangled representation advances, we can immediately incorporate new approaches in the MichiGAN framework, since the training of representation and GAN are completely separate.

We envision several exciting future directions. First, it would be interesting to investigate the representations learned by β-VAE or β-TCVAE across a range of biological contexts. Second, incorporating additional state-of-the-art GAN training techniques may further improve data generation quality. Additionally, there are many other biological settings in which predicting unseen combinations of latent variables may be helpful, such as cross-species analysis or disease state prediction.

Methods

Real scRNA-seq datasets

The Tabula Muris dataset is a compendium of single-cell transcriptomic data from the model organism Mus musculus [65]. We processed the Tabula Muris data using SCANPY [68] and the dataset contains 41,965 cells and 4062 genes from 64 cell types. The sci-Plex dataset has three cell types treated with 188 molecules targeting 22 pathways [67]. We selected the 18 common pathways among the three cell types and chose the drug treatment from each pathway with largest number of cells. We also use SCANPY to process the data and then have 64,050 cells and 4295 genes. The pancreatic endocrinogenesis contains 3696 cells and 27,998 genes [66]. We filtered and normalized the pancreas data to 2,000 genes using the scVelo package [75]. We also and obtained the latent time and G2M and S cell cycle scores for each cell.

Simulated scRNA-seq datasets

To simulate data with the Splatter package, we first estimated simulation parameters to match the Tabula Muris dataset [65]. Then, we set the differential expression probability, factor location, factor scale, and common biological coefficient of variation to be $(0.5, 0.01, 0.5, 0.1)$. We then used Splatter [63] to simulate gene expression data of 10,000 cells with four underlying ground-truth variables: batch, path, step, and library size.

Using the PROSSTT package, we simulated 2000 genes across 10,500 (3 trajectories), 10,800 (4 trajectories), and 11,000 cells (5 trajectories). We followed the steps and parameter settings exactly as described in the PROSSTT tutorial (https://github.com/soedinglab/prosstt/blob/master/examples/many_branches_cells.ipynb), varying only the number of branches, cells, and genes.

Variational autoencoders VAEs have an encoder network with parameters (ϕ), which maps the input data (X) to a latent space Z, and a decoder network parameterized by (θ), which reconstructs the high-dimensional data from the latent space.

Rather than learning a deterministic function for the encoder as in a conventional autoencoder, a VAE learns the mean and variance parameters of the posterior distribution over the latent variables. However, even using a factorized Gaussian prior, the posterior is intractable. Thus, VAEs perform parameter inference using variational Bayes. Following a standard mean-field approximation, one can derive an evidence lower bound (ELBO). The objective function of VAE is to maximize the ELBO or minimize its opposite with respect to ϕ and θ:

$$L_{\mathrm{VAE}} = -\mathrm{ELBO} = E_{q(X)}\left[-E_{q_\phi(Z|X)} p_\theta(X \mid Z) + \mathrm{KL}\left\{q_\phi(Z \mid X) || p(Z)\right\}\right],$$

The ELBO has a nice interpretation: the first term is reconstruction error and the second term is the Kullback-Leibler divergence between the posterior and prior distributions of the latent variables (Z). If the prior distribution $p(Z)$ is factorized Gaussian or uniform distribution, the KL divergence encourages the latent factors to be statistically independent, which may contribute to the good disentanglement performance of VAEs. This effect can be further enhanced by introducing a weight β to place more emphasis on the KL divergence at the cost of reconstruction error, an approach called β-VAE [9].

β-TCVAE The total correlation variational autoencoder (β-TCVAE) is a VAE extension that further promotes disentanglement. The KL divergence of VAE can be further decomposed into several parts:

$$E_{q(X)}\left\{\mathrm{KL}\left[q_\phi(Z \mid X) || p(Z)\right]\right\} = \mathrm{KL}\left[q_\phi(Z, X) || q_\phi(Z) q(X)\right] + \mathrm{KL}\left[q_\phi(Z) || \prod_j q_\phi(Z_j)\right] + \sum_j \mathrm{KL}\left[q_\phi(Z_j) || p(Z_j)\right],$$

The first part is referred to as the index-code mutual information (MI), the second part is the total correlation (TC), and the third part is the dimension-wise KL divergence [30]. The total correlation is the most important term for learning disentangled representations, while penalizing the two other parts does not directly improve the disentanglement performance, but increases the reconstruction error.

The β-TCVAE specifically penalizes the TC in the loss function [29, 30]:

$$L_{\beta\text{-TCVAE}} = L_{\mathrm{VAE}} + \beta \mathrm{KL}\left[q_\phi(Z) || \prod_j q_\phi(Z_j)\right],$$

where $\beta = 0$ gives the VAE loss function. There is no closed form for the total correlation of the latent representation, so β-TCVAE approximates it as follows:

$$E_{q_\phi(Z)}\left\{\log q_\phi(Z)\right\} \approx E_{q_\phi(Z)}\left[\log E\left[\{q_\phi(Z \mid X) \mid Z\right\}\right],$$

and

$$E_{q_\phi(Z_j)}\left\{\log q_\phi(Z_j)\right\} \approx E_{q_\phi(Z_j)}\left[\log E\left\{q_\phi(Z_j \mid X) \mid Z_j\right\}\right].$$

Estimating TC is difficult from a small minibatch, so we utilize the minibatch stratified sampling mentioned in [30] to estimate $E\left\{q_\phi(Z \mid X) \mid Z\right\}$ during training.

Generative adversarial networks (GAN) A generative adversarial network consists of a generator network G and a discriminator network D. There are many types of GANs, but we specifically focus on Wasserstein GAN with gradient penalty (WGAN-GP) [12], which significantly stabilizes GAN training. The discriminator loss function for WGAN-GP is

$$L_{\text{Discriminator}} = E_{p(Z),q(X)}\left[D(X) - D\left\{G(Z)\right\} + \lambda\left\{||\nabla_{\widetilde{X}} D(\widetilde{X})||_2 - 1\right\}^2\right],$$

where $\nabla_X D$ is the gradient of the discriminator on input X and $\widetilde{X} = \epsilon X + (1-\epsilon)G(Z)$ with ϵ sampled from a uniform distribution on $[0, 1]$. The generator loss function for WGAN-GP is

$$L_{\text{Generator}} = E_{p(Z)}\left[D\{G(Z)\}\right].$$

Upon convergence, WGAN-GP gives the generated data distribution $G(Z)$ that matches the real data distribution $P(X)$.

InfoGAN and ssInfoGAN The Information Maximizing Generative Adversarial Networks (InfoGAN) framework extends the regular GAN to encourage disentanglement [32]. The InfoGAN decomposes the latent variables into latent code C and noise Z. To encourage disentanglement, InfoGAN maximizes the mutual information between the latent code and the generated data. To estimate mutual information, InfoGAN relies on an additional network Q that takes generated data as input and predicts the code $Q(C \mid X)$ that generated the data. $Q(C \mid X)$ is very similar to an encoder in a VAE and estimates a posterior distribution in the same as the prior distribution of the code $p(C)$. InfoGAN then maximizes mutual information between the code and generated data with the following loss functions for the discriminator and generator:

$$\min_{G,Q}\max_{D} L(D, G, Q) = \min_{G,Q}\max_{D}\left\{L_{\text{GAN}}(G, D) - \lambda_{\text{MI}} L_{\text{MI}}(G, Q)\right\},$$

where $L_{\text{MI}}(G, Q) = E_{C\sim P(C), X\sim G(C,Z)}[\log Q(C \mid X)] + H(C)$ is a lower bound for the mutual information between C and X and $H(C)$ is the entropy of the codes. We implemented InfoGAN with the Wasserstein distance, which we refer to as InfoWGAN-GP. We choose a factorized normal distribution with unit variance for $Q(C \mid X)$ (the unit variance stabilizes InfoGAN training [32, 36]).

InfoGAN architecture can also be extended to semi-supervised InfoGAN (ssInfoGAN), if labels are available for some or all of the data points [74]. The ssInfoGAN maximizes mutual information not only between the generated data and the codes, but also between the real data and corresponding labels. This guides the learned codes to reflect the label information.

Conditional GAN and PCGAN The conditional GAN extends GANs to respect the relationship between generated data and known labels [77]. There are many different network architectures for conditional GAN [77–79], but found the conditional GAN with projection discriminator (PCGAN) [73] works best. A recent paper similarly found that PCGAN worked well for single-cell RNA-seq data [62]. The original PCGAN paper mentions that the projection discriminator works most effectively when the conditional distribution $p(C|X)$ is unimodal. One theoretical reason why PCGAN may be well-suited for MichiGAN is that the posterior multivariate Gaussian distributions of latent variables from VAEs are, in fact, unimodal.

In implementing the PCGAN, we do not use the conditional batch normalization or spectral normalization mentioned in [73], but instead use standard batch normalization and Wasserstein GAN with gradient penalty. Thus, we refer to this approach as PCWGAN-GP.

MichiGAN Algorithm 1 describes how to train MichiGAN:

```
Algorithm 1: MichiGAN
Input: single-cell RNA-seq data $X$
1. Obtain disentangled representations $Z_X$ from an approach such as PCA, VAE or $\beta$-TCVAE.
2. Utilize the representations $Z_X$ as codes.
3. Train a conditional GAN [73] using the codes.
Result: a generator network that produces high-quality samples from a disentangled latent representation
```

Latent space vector arithmetic

MichiGAN's ability to sample from a disentangled representation allows predicting unseen combinations of latent variables using latent space arithmetic. We perform latent space arithmetic as in [51] to predict the single-cell gene expression of unseen cell states. Specifically, suppose we have m cell types $C_1, \ldots, C_m$ and n perturbation $D_1, \ldots, D_n$. Denote $Z\left(C_i, D_j\right)$ as the latent value corresponding to the expression data with combination $\left(C_i, D_j\right)$ for $1 \leq i \leq m$ and $1 \leq j \leq n$. If we want to predict the unobserved expression profile for the combination $\left(C_{i'}, D_{j'}\right)$, we can calculate the average latent difference between cell type $C_{i'}$ and another cell type C_k in the set of observable treatments Ω that $\Delta_{C_{i'},C_k} = \int_\Omega \{Z\left(C_{i'}, D_s\right) - Z\left(C_k, D_s\right)\} \, dP(s)$ and then use the latent space $Z\left(C_k, D_{j'}\right)$ of observed combination $\left(C_k, D_{j'}\right)$ to predict

$$\widehat{Z}\left(C_{i'}, D_{j'}\right) = Z\left(C_k, D_{j'}\right) + \Delta_{C_{i'},C_k}.$$

The predicted $\widehat{Z}\left(C_{i'}, D_{j'}\right)$ is further used to generate predicted data of the unseen combination. The predicted latent space assumes the average latent difference across observed treatments is equal to the latent difference of the unobserved treatment, which may not hold if there is a strong cell type effect for the perturbation.

Disentanglement metrics

Mutual information

Following [30], we measure the disentanglement performance of the representations using mutual information gap (MIG). Denote $p\left(V_k\right)$ and $p\left(X \mid V_k\right)$ as the probability of a ground-truth variable V_k and the conditional probability of the data X under V_k. Given $q_\phi\left(Z_j, V_k\right) = \int_X p\left(V_k\right) p\left(X \mid V_k\right) q_\phi\left(Z_j \mid X\right) dx$, the mutual information between a latent variable Z_j and a ground-truth variable V_k is defined as

$$I\left(Z_j, V_k\right) = E_{q_\phi\left(Z_j, V_k\right)} \left\{ \log \int_{X \in \mathcal{X}_{V_k}} q_\phi\left(Z_j \mid X\right) p\left(X \mid V_k\right) dX \right\} + H\left(Z_j\right),$$

where $\mathcal{X}_{V_k}$ is the support of $p\left(X \mid V_k\right)$ and $H\left(Z_j\right)$ is the entropy of Z_j. Due to the different variabilities of the ground-truth variables, the normalized mutual information is better to be used with a normalization term of $H\left(V_k\right)$, the entropy of V_k. The posterior distribution $q_\phi\left(Z_j \mid X\right)$ is obtained from the encoder (for VAEs) or the derived posterior distribution for probabilistic PCA [80]. With K ground truth variables $\{V_1, \ldots, V_k\}$, the mutual information gap (MIG) is further defined as

$$\text{MIG} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{H(V_k)} \left\{ I\left(Z_{j^{(k)}}, V_k\right) - \max_{j \neq j^{(k)}} I(Z_j, V_k) \right\},$$

where $j^{(k)} = \arg\max_j I\left(Z_j, V_k\right)$.

The MIG metric is the average difference between largest and the second largest normalized mutual information value across all ground-truth variables. Intuitively, this indicates how much each ground truth variable is captured by a single latent variable. As described in [30], the MIG metric has the axis-alignment property and is unbiased for all hyperparameter settings.

FactorVAE metric

For completeness, we also calculated the disentanglement metric introduced in the FactorVAE paper [29]. In each of multiple repetitions, we first randomly choose a ground-truth variable and then generate data, keeping this variable fixed and other variables at random. We normalize each dimension by the empirical standard deviation over the whole data and choose the dimension with the lowest empirical variance. The dimension with the lowest empirical variance and the fixed ground-truth variable are then used as (x, y) pairs to train a majority vote classifier. The FactorVAE disentanglement metric is defined as the accuracy of the resulting classifier.

Spearman correlation

Inspired by the MIG metric, we also utilized Spearman correlation to quantify disentanglement performance. Although Spearman correlation is a more restricted metric of statistical dependence than mutual information, it has the advantage that it can be computed without a distributional estimate of a latent representation, which is not available for GAN models. Given the Spearman correlation $S = \text{cor}\left(Z_j, V_k\right)$ between inferred representation Z_j and ground truth variable V_k, we define the corresponding correlation gap as $|\text{cor}\left(Z_{j^{(k)}}, V_k\right)| - \max_{j \neq j^{(k)}} |\text{cor}\left(Z_j, V_k\right)|$, where $j^{(k)} = \arg\max_j |\text{cor}\left(Z_j, V_k\right)|$.

Generation metrics

Random forest error

We follow the random forest error metric introduced in the cscGAN paper [62] to quantify how difficult it is for a random forest classifier to distinguish generated cells from real cells. A higher random forest error indicates that the generated samples are more realistic. We randomly sample 3000 cells and generate 3000 additional cells. Then we train a random forest classifier on the 50 principal components of the 6000 cells to predict each cell to be a real or fake cell. We train with 5-fold cross validation and report the average error across the 5 folds.

Inception score

We also define an inception score metric similar to the one widely used in evaluating performance on image data [70]. Intuitively, to achieve a high inception score, a generative model must generate every class in the training dataset (analogous to recall) and every generated example must be recognizable as belonging to a particular class (analogous to precision). We train a random forest classifier on 3000 randomly sampled real cells to predict their cell types. Based on the trained cell-type classifier, we are able to predict the probabilities of being different cell types for each generated cell. We then input the predicted probabilities to the calculations of the inception score.

Tuning β values in β-TCVAE

The β value is a hyperparameter in the β-TCVAE model that controls the relative importance of penalizing the total correlation of the learned representation. Because β is a hyperparameter in an unsupervised learning approach (no ground truth is available in general), there is no direct way to pick a single best value for β. This is not a problem unique to the β-TCVAE, but is a general challenge with any unsupervised learning approach. Our best recommendation is to choose a value in the range of 10–50 and use whatever biological prior knowledge is available, such as annotations of cell time point, condition, or cell type, to qualitatively assess the disentanglement of representations for different values. One of the best things one can hope for with unsupervised learning algorithms is that the results are robust to different hyperparameter settings. To show that this is true in this case, we measured disentanglement performance of VAE and β-TCVAE for $\beta = 10$ and 50 on the simulated datasets as shown in Additional file 1: Figure S14. We found that β-TCVAE with $\beta = 10$ or 50 consistently gives a higher mutual information gap (MIG) than VAE. In short, even if you do not choose the perfect value of β, it is still better to use β-TCVAE than VAE.

Implementation

The VAE-based methods use multilayer perceptron (MLP) units and have two fully connected (FC) hidden layers with 512 and 256 neurons, followed by separate parameters for mean and variance of the latent representation. The first two hidden layers in the decoder have 256 and 512 neurons, while the last layer gives mean gene expression and has the same number of neurons as the number of genes. Each hidden layer utilizes batch normalization, activated by Rectified Linear Unit (ReLU) or Leaky ReLU. Each hidden layer employs dropout regularization, with a dropout probability of 0.2. We also experimented with three hidden layers for the VAE encoders, but found that the training became unstable. This is consistent with a previous report [60] that found most VAEs for biological data have only two hidden layers. The GAN-based methods also use MLP for both generator and discriminator. There are three FC hidden layers with 256, 512, and 1024 neurons as well as three hidden layers with 1024, 512, and 10 neurons from data to output. The hidden layers of GANs also have Batch Normalization and ReLU or Leaky ReLU activation. The generator uses dropout regularization with dropout probability of 0.2 for each hidden layer. The VAE-based methods are trained with Adam optimization, while the GAN-based methods are trained with Adam and the gradient prediction method [81]. All the hyperparameters of each method on different datasets are tuned for the optimal results.

We trained all models for 1000 epochs and used 10 latent variables. We used $\beta = 10$ for β-TCVAE on all of the splatter-simulated single-cell RNA-seq datasets, except that we used $\beta = 5$ for β-TCVAE with 4 latent dimensions on the simulated data with linear step. We used $\beta = 50$ for β-TCVAE on the PROSSTT-simulated datasets and pancreas dataset. For the two real scRNA-seq datasets, we used $\beta = 100$. We used 118-dimensional Gaussian noise for MichiGAN. All models were implemented in TensorFlow.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-021-02373-4.

Additional file 1: Tables S1 and Figures S1–S15

Additional file 2: Review history

Acknowledgements

We thank Hojae Lee for helpful discussions about this project.

Peer review information

Barbara Cheifet was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Review history

The review history is available as Additional file 2.

Authors' information

Twitter handles: @HengshiY (Hengshi Yu); @LabWelch (Joshua D. Welch).

Authors' contributions

HY and JDW conceived the idea of MichiGAN. HY implemented the approach and performed the data analyses. HY and JDW wrote the paper. The authors read and approved the final manuscript.

Funding

JDW gratefully acknowledges support from NIH grant R01-HG010883.

Availability of data and materials

MichiGAN code is available at a DOI-assigning repository Zenodo (https://doi.org/10.5281/zenodo.4728278) [82] and at GitHub (https://github.com/welch-lab/MichiGAN) [83] under GNU General Public License v3.0. Detailed documentation and a Jupyter notebook demonstrating how to use the package are available on the GitHub page.

The tabula muris data [65] supporting the conclusions of this article are available in the tabula-muris Python GitHub repository, https://github.com/czbiohub/tabula-muris. The large sci-Plex data [67] are available on GEO (GSM4150378, https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM4150378). The pancreas endocrinogenesis dataset [66] as well as its cells' latent time and cell cycle scores for G2M and S phases are available from the examples of the scVelo package [75] at https://scvelo.readthedocs.io/Pancreas.html.

The R package splatter is implemented on R version 3.6.1. The PROSSTT package is available at https://github.com/soedinglab/prosstt. The deep generative models and performance metrics are based on TensorFlow version 1.14.0 and Python 3.6.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Biostatistics, University of Michigan, Ann Arbor, USA. [2]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, USA. [3]Department of Computer Science and Engineering, University of Michigan, Ann Arbor, USA.

Received: 17 November 2020 Accepted: 5 May 2021
Published online: 20 May 2021

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44.
2. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell. 2013;35(8):1798–828.
3. Theis L, Oord A, Bethge M. A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844. 2015.
4. Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096. 2018.
5. Wu Y, Donahue J, Balduzzi D, Simonyan K, Lillicrap T. Logan: Latent optimisation for generative adversarial networks. arXiv preprint arXiv:1912.00953. 2019.

6. Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114. 2013.
7. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Advances in Neural Information Processing Systems; 2014. p. 2672–80.
8. Goodfellow I, Bengio Y, Courville A. Deep Learning: MIT Press; 2016. http://www.deeplearningbook.org.
9. Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A. beta-vae: Learning basic visual concepts with a constrained variational framework. Iclr. 2017;2(5):6.
10. Larsen ABL, Sønderby SK, Larochelle H, Winther O. Autoencoding beyond pixels using a learned similarity metric. In: International Conference on Machine Learning. PMLR; 2016. p. 1558–66.
11. Arjovsky M, Chintala S, Bottou L. Wasserstein gan. arXiv preprint arXiv:1701.07875. 2017.
12. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC. Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems; 2017. p. 5767–77.
13. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems; 2017. p. 6626–37.
14. Desjardins G, Courville A, Bengio Y. Disentangling factors of variation via generative entangling. arXiv preprint arXiv:1210.5474. 2012.
15. Ridgeway K. A survey of inductive biases for factorial representation-learning. arXiv preprint arXiv:1612.05299. 2016.
16. Denton EL, et al. Unsupervised learning of disentangled representations from video. In: Advances in Neural Information Processing Systems; 2017. p. 4414–23.
17. Achille A, Soatto S. Emergence of invariance and disentanglement in deep representations. J Mach Learn Res. 2018;19(1):1947–80.
18. Eastwood C, Williams CK. A framework for the quantitative evaluation of disentangled representations. In: International Conference on Learning Representations; 2018.
19. Locatello F, Bauer S, Lucic M, Raetsch G, Gelly S, Schölkopf B, Bachem O. Challenging common assumptions in the unsupervised learning of disentangled representations. In: International Conference on Machine Learning; 2019. p. 4114–24.
20. Higgins I, Amos D, Pfau D, Racaniere S, Matthey L, Rezende D, Lerchner A. Towards a definition of disentangled representations. arXiv preprint arXiv:1812.02230. 2018.
21. Berthelot D, Raffel C, Roy A, Goodfellow I. Understanding and improving interpolation in autoencoders via an adversarial regularizer. arXiv preprint arXiv:1807.07543. 2018.
22. Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 4401–10.
23. Hsu W-N, Zhang Y, Glass J. Unsupervised learning of disentangled and interpretable representations from sequential data. In: Advances in Neural Information Processing Systems; 2017. p. 1878–89.
24. Dupont E. Learning disentangled joint continuous and discrete representations. In: Advances in Neural Information Processing Systems; 2018. p. 710–20.
25. Bai Y, Duan LL. Tuning-free disentanglement via projection. arXiv preprint arXiv:1906.11732. 2019.
26. Rolinek M, Zietlow D, Martius G. Variational autoencoders pursue pca directions (by accident). In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 12406–15.
27. Esmaeili B, Wu H, Jain S, Bozkurt A, Siddharth N, Paige B, Brooks DH, Dy J, Meent J-W. Structured disentangled representations. In: The 22nd International Conference on Artificial Intelligence and Statistics. PMLR; 2019. p. 2525–34.
28. Khemakhem I, Kingma D, Monti R, Hyvarinen A. Variational autoencoders and nonlinear ica: A unifying framework. In: International Conference on Artificial Intelligence and Statistics; 2020. p. 2207–17.
29. Kim H, Mnih A. Disentangling by factorising. In: International Conference on Machine Learning; 2018. p. 2649–58.
30. Chen TQ, Li X, Grosse RB, Duvenaud DK. Isolating sources of disentanglement in variational autoencoders. In: Advances in Neural Information Processing Systems; 2018. p. 2610–20.
31. Gao S, Brekelmans R, Ver Steeg G, Galstyan A. Auto-encoding total correlation explanation. In: The 22nd International Conference on Artificial Intelligence and Statistics; 2019. p. 1157–66.
32. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems; 2016. p. 2172–80.
33. Ramesh A, Choi Y, LeCun Y. A spectral regularizer for unsupervised disentanglement. arXiv preprint arXiv:1812.01161. 2018.
34. Kaneko T, Hiramatsu K, Kashino K. Generative adversarial image synthesis with decision tree latent controller. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 6606–15.
35. Jeon I, Lee W, Kim G. IB-GAN: Disentangled representation learning with information bottleneck GAN. 2018.
36. Lin Z, Thekumparampil KK, Fanti G, Oh S. Infogan-cr: Disentangling generative adversarial networks with contrastive regularizers. arXiv preprint arXiv:1906.06034. 2019.
37. Kazemi H, Iranmanesh SM, Nasrabadi N. Style and content disentanglement in generative adversarial networks. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE; 2019. p. 848–56.
38. Shen Y, Gu J, Tang X, Zhou B. Interpreting the latent space of gans for semantic face editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 9243–52.
39. Liu B, Zhu Y, Fu Z, de Melo G, Elgammal A. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. In: AAAI; 2020. p. 4836–43.
40. Lee W, Kim D, Hong S, Lee H. High-fidelity synthesis with disentangled representation. In: European Conference on Computer Vision. Springer; 2020. p. 157–74.
41. Matthey L, Higgins I, Hassabis D, Lerchner A. dsprites: Disentanglement testing sprites dataset. 2017. https://github.com/deepmind/dsprites-dataset/. Accessed on: 08 May 2018.
42. Paysan P, Knothe R, Amberg B, Romdhani S, Vetter T. A 3D face model for pose and illumination invariant face recognition. In: 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance. IEEE; 2009. p. 296–301.

43. Aubry M, Maturana D, Efros AA, Russell BC, Sivic J. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2014. p. 3762–9.
44. Liu Z, Luo P, Wang X, Tang X. Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision; 2015. p. 3730–8.
45. Burgess CP, Higgins I, Pal A, Matthey L, Watters N, Desjardins G, Lerchner A. Understanding disentangling in $\backslash\beta$-vae. arXiv preprint arXiv:1804.03599. 2018.
46. White T. Sampling generative networks. arXiv preprint arXiv:1609.04468. 2016.
47. Laine S. Feature-based metrics for exploring the latent space of generative models. 2018.
48. Dosovitskiy A, Tobias Springenberg J, Brox T. Learning to generate chairs with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 1538–46.
49. Sainburg T, Thielk M, Theilman B, Migliori B, Gentner T. Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions. arXiv preprint arXiv:1807.06650. 2018.
50. Efremova M, Teichmann SA. Computational methods for single-cell omics across modalities. Nat Methods. 2020;17(1):14–7.
51. Lotfollahi M, Wolf FA, Theis FJ. scGen predicts single-cell perturbation responses. Nat Methods. 2019;16(8):715–21.
52. Tan J, Ung M, Cheng C, Greene CS. Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. In: Pacific Symposium on Biocomputing Co-Chairs. World Scientific; 2014. p. 132–43.
53. Gupta A, Wang H, Ganapathiraju M. Learning structure in gene expression data using deep architectures, with an application to gene clustering. In: 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). sIEEE; 2015. p. 1328–35.
54. Way GP, Greene CS. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. BioRxiv. 2017174474.
55. Rampasek L, Hidru D, Smirnov P, Haibe-Kains B, Goldenberg A. Dr. vae: Drug response variational autoencoder. arXiv preprint arXiv:1706.08203. 2017.
56. Deng Y, Bao F, Dai Q, Wu LF, Altschuler SJ. Massive single-cell rna-seq analysis and imputation via deep learning. bioRxiv. 2018315556.
57. Grønbech CH, Vording MF, Timshel PN, Sønderby CK, Pers TH, Winther O. scvae: Variational auto-encoders for single-cell gene expression datas. bioRxiv. 2018318295.
58. Wang D, Gu J. VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. Genomics Proteomics Bioinforma. 2018;16(5):320–31.
59. Ding J, Condon A, Shah SP. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nat Commun. 2018;9(1):1–13.
60. Hu Q, Greene CS. Parameter tuning is a key part of dimensionality reduction via deep variational autoencoders for single cell RNA transcriptomics. In: PSB. World Scientific; 2019. p. 362–73.
61. Cui H, Zhou C, Dai X, Liang Y, Paffenroth R, Korkin D. Boosting gene expression clustering with system-wide biological information: a robust autoencoder approach. Int J Comput Biol Drug Des. 2020;13(1):98–123.
62. Marouf M, Machart P, Bansal V, Kilian C, Magruder DS, Krebs CF, Bonn S. Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. Nat Commun. 2020;11(1):1–12.
63. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. Genome Biol. 2017;18(1):174.
64. Papadopoulos N, Gonzalo PR, Söding J. PROSSTT: probabilistic simulation of single-cell RNA-seq data for complex differentiation processes. Bioinformatics. 2019;35(18):3517–9.
65. Tabula Muris Consortium, et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature. 2018;562(7727):367–72.
66. Bastidas-Ponce A, Tritschler S, Dony L, Scheibner K, Tarquis-Medina M, Salinno C, Schirge S, Burtscher I, Böttcher A, Theis FJ, Lickert H, Bakhti M, Klein A, Treutlein B. Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. Development. 2019;146(12):dev173849. https://doi.org/10.1242/dev.173849.
67. Srivatsan SR, McFaline-Figueroa JL, Ramani V, Saunders L, Cao J, Packer J, Pliner HA, Jackson DL, Daza RM, Christiansen L, et al. Massively multiplex chemical transcriptomics at single-cell resolution. Science. 2020;367(6473): 45–51.
68. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19(1): 15.
69. Tipping ME, Bishop CM. Probabilistic principal component analysis. J R Stat Soc Ser B Stat Methodol. 1999;61(3): 611–22.
70. Barratt S, Sharma R. A note on the inception score. arXiv preprint arXiv:1801.01973. 2018.
71. Pu Y, Wang W, Henao R, Chen L, Gan Z, Li C, Carin L. Adversarial symmetric variational autoencoder. In: Advances in Neural Information Processing Systems; 2017. p. 4330–9.
72. Mescheder L, Nowozin S, Geiger A. In: International Conference on Machine Learning. PMLR; 2017. p. 2391–400.
73. Miyato T, Koyama M. cGANs with projection discriminator. arXiv preprint arXiv:1802.05637. 2018.
74. Spurr A, Aksan E, Hilliges O. Guiding infogan with semi-supervision. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2017. p. 119–34.
75. Bergen V, Lange M, Peidli S, Wolf FA, Theis FJ. Generalizing RNA velocity to transient cell states through dynamical modeling. Nat Biotechnol. 2020;38(12):1408–14.
76. Pei D, Shu X, Gassama-Diagne A, Thiery JP. Mesenchymal–epithelial transition in development and reprogramming. Nat Cell Biol. 2019;21(1):44–53.
77. Mirza M, Osindero S. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784. 2014.
78. Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H. Generative adversarial text to image synthesis. In: International Conference on Machine Learning. PMLR; 2016. p. 1060–9.
79. Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org; 2017. p. 2642–51.

80. Bishop CM. Pattern Recognition and Machine Learning (Information Science and Statistics). Berlin: Springer-Verlag; 2006.
81. Yadav A, Shah S, Xu Z, Jacobs D, Goldstein T. Stabilizing adversarial nets with prediction methods. arXiv preprint arXiv:1705.07364. 2017.
82. Yu H, Welch J. MichiGAN: sampling from disentangled representations of single-cell data using generative adversarial networks. Zenodo. 2021. https://doi.org/10.5281/zenodo.4728278.
83. Yu H, Welch J. MichiGAN: sampling from disentangled representations of single-cell data using generative adversarial networks. Github. 2021. https://github.com/welch-lab/MichiGAN.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.