Conditional out-of-sample generation for unpaired data using transfer VAE

Mohammad Lotfollahi[1,2], Mohsen Naghipourfar [1,4], Fabian J. Theis[1,2,3†] & F. Alexander Wolf[1‡]
1 Institute of Computational Biology, Helmholtz Center Munich, Neuherberg, Germany.
2 School of Life Sciences Weihenstephan, Technical University of Munich, Munich, Germany
3 Department of Mathematics, Technische Universität München, Munich, Germany.
4 Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.
† fabian.theis@helmholtz-muenchen.de ‡ alex.wolf@helmholtz-muenchen.de

A METHODS

A.1 STATISTICS

All the differential tests to extract DEGs were performed using Scanpy's rank_genes_groups function with Wilcoxon as the method parameter. Error bars were computed by randomly resampling 90% of the data with replacement 100 times and recomputing Pearson R^2 for each resampled data. The interval represents the mean of R^2 values plus/minus the standard deviation of those 100 R^2 values. We used the mean of 100 R^2 values for the magnitude of each bar. All the R^2 values were calculated by squaring the rvalue output of the scipy.stats.linregress function and denote squared Pearson correlation.

A.2 GENE SPECIFICITY SCORE

In order to bin genes based on differential pattern specificity, we use the following metric for each to score each gene:

$$\text{score}_i = max_{\text{celltype}_j} \mid \overline{\text{lfc}}_{ij} - \text{median}(\overline{\text{lfc}}_i) \mid \tag{1}$$

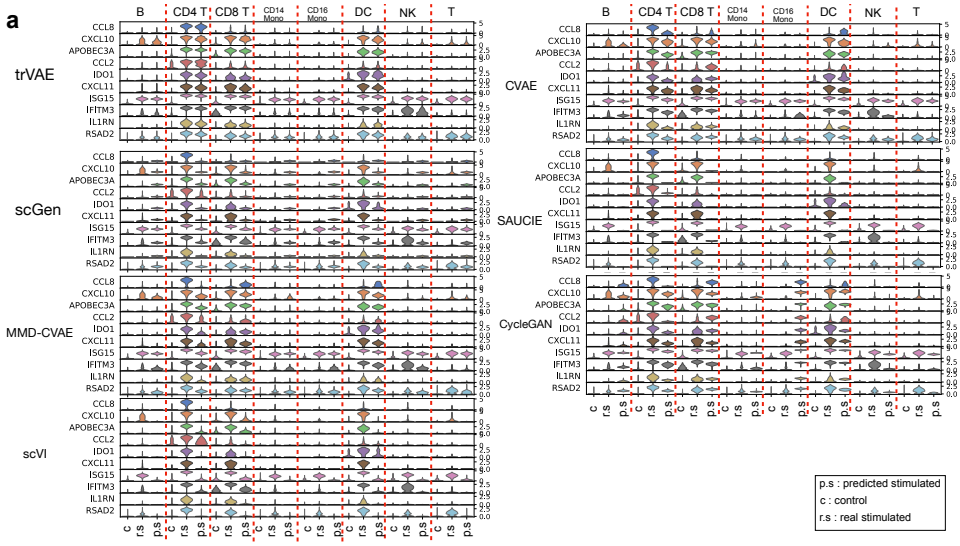
where $\overline{\text{lfc}}_{ij} = \mid \overline{\text{stim}}_{ij} - \overline{\text{ctrl}}_{ij} \mid$ for cell type j and gene i. We also define $\overline{\text{stim}}_{ij}$ and $\overline{\text{ctrl}}_{ij}$ as the average of all cells in cell type j in gene i for stimulated and control cells, respectively and $\text{median}(\overline{\text{lfc}}_i)$ is median of all $\overline{\text{lfc}}_{ij}$ for gene i.

A.3 RUN-TIME COMPARISON

Supplementary Table 1: Run-time of trVAE compared to other models in Kang et. al and Haber et. al datasets. Reported numbers are average run times of three runs ± std of these runs.

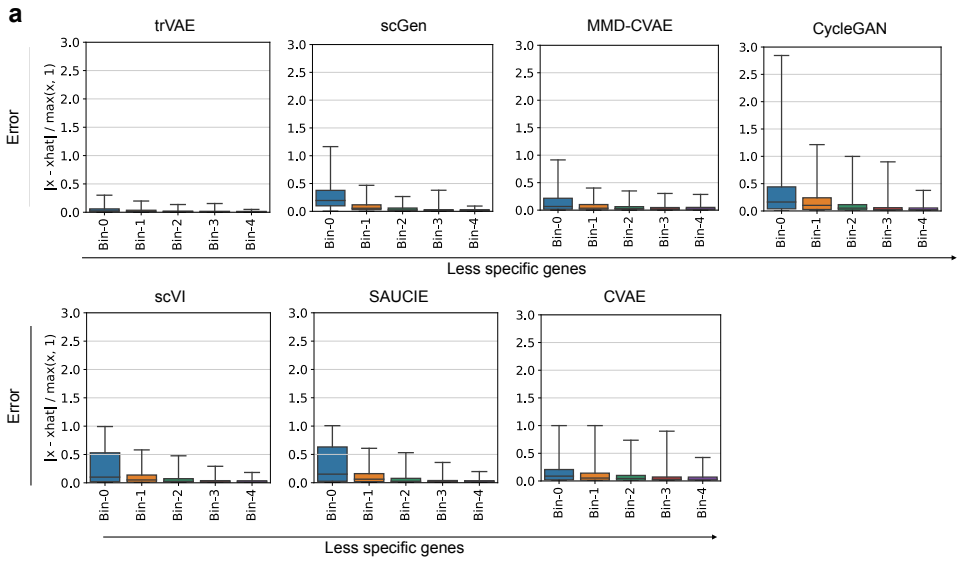
Model \ Dataset	Kang et. al	Haber et. al
trVAE	448.5s ±8s	103.3s ±5s
scGen	327.7s ±7s	206.4s ±5s
MMD-CVAE	378.7s ±10s	176.9s ±4s
scVI	1178.6s ±14s	486.3s ±8s
CVAE	1475.5s ±11s	670.2s ±6s
SAUCIE	39.1s ±3s	28.9s ±2s
CycleGAN	6823.1s ±34s	2095.7s ±28s

A.4 SIMULATION RESPONSE



Supplementary Figure 1: **(a)** Violin plot for top 10 specific response genes after out of 250 DEGs according to the gene specificity score across control (c), real stimulated (r.s), and predicted stimulated (p.s) for different cell types. Vertical axis: expression distribution for top specific genes. Horizontal axis: control, real and predicted distribution by trVAE and other methods for different cell types.

A.5 HYPER-PARAMETERS



Supplementary Figure 2: **(a)** Box plots of top 500 DEGs ordered by the gene specificity score. Each bin is composed of 50 genes and each point in the bin shows the error between average expression of that gene within a cell type and average prediction by trVAE and other methods for that cell type. In total, each boxplot has been derived from 50 (number of genes) x 8 (number of cell types) points (n=400). Box plots indicate the median (center lines), interquantile range (hinges), and whiskers represents min and max values.

Supplementary Table 2: Convolutional trVAE detailed architecture used for Morpho-MNIST dataset.

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	(28, 28, 1)	×	×	-	-
conditions	-	2	×	×	-	-
FC-1	FC	128	×	×	Leaky ReLU	conditions
FC-2	FC	784	0.2	√	Leaky ReLU	FC-1
FC-2_resh	Reshape	(28, 28, 1)	×	×	×	FC-2
Conv2D_1	Conv2D	(4, 4, 64, 2)	×	×	Leaky ReLU	[FC-2_resh, input]
Conv2D_2	Conv2D	(4, 4, 64, 64)	×	×	Leaky ReLU	Conv2D_1
FC-3	FC	128	×	√	Leaky ReLU	Flatten(Conv2D_2)
mean	FC	50	×	×	Linear	FC-3
var	FC	50	×	×	Linear	FC-3
z	FC	50	×	×	Linear	[mean, var]
FC-4	FC	128	×	×	Leaky ReLU	conditions
FC-5	FC	784	0.2	√	Leaky ReLU	FC-4
FC-5_resh	Reshape	(28, 28, 1)	×	×	×	FC-5
MMD	FC	128	×	√	Leaky ReLU	[z, FC-5_resh]
FC-6	FC	256	×	×	Leaky ReLU	MMD
FC-7_resh	Reshape	(2, 2, 64)	×	×	×	FC-6
Conv_transp_1	Conv2D Transpose	(4, 4, 128, 64)	×	×	Leaky ReLU	FC-7_resh
Conv_transp_2	Conv2D Transpose	(4, 4, 64, 64)	×	×	Leaky ReLU	UpSampling2D(Conv_tra
Conv_transp_3	Conv2D Transpose	(4, 4, 64, 64)	×	×	Leaky ReLU	Conv_transp_2
Conv_transp_4	Conv2D Transpose	(4, 4, 2, 64)	×	×	Leaky ReLU	UpSampling2D(Conv_tra
output	Conv2D Transpose	(4, 4, 1, 2)	×	×	ReLU	UpSampling2D(Conv_tra
Optimizer	Adam					
Learning Rate	0.001					
Leaky ReLU slope	0.2					
Batch Size	1024					
# of Epochs	5000					
α	0.001					
β	1000					

Supplementary Table 3: U-Net trVAE detailed architecture used for CelebA dataset.

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	(64, 64, 3)	×	×	-	-
conditions	-	2	×	×	-	-
FC-1	FC	128	×	×	ReLU	conditions
FC-2	FC	1024	0.2	√	ReLU	FC-1
FC-2_reshaped	Reshape	(64, 64, 1)	×	×	×	FC-2
Conv_1	Conv2D	(3, 3, 64, 4)	×	×	ReLU	[FC-2_reshaped, input]
Conv_2	Conv2D	(3, 3, 64, 64)	×	×	ReLU	Conv_1
Pool_1	Pooling2D	×	×	×	×	Conv_2
Conv_3	Conv2D	(3, 3, 128, 64)	×	×	ReLU	Pool_1
Conv_4	Conv2D	(3, 3, 128, 128)	×	×	ReLU	Conv_3
Pool_2	Pooling2D	×	×	×	×	Conv_4
Conv_5	Conv2D	(3, 3, 256, 128)	×	×	ReLU	Pool_2
Conv_6	Conv2D	(3, 3, 256, 256)	×	×	ReLU	Conv_5
Conv_7	Conv2D	(3, 3, 256, 256)	×	×	ReLU	Conv_6
Pool_3	Pooling2D	×	×	×	×	Conv_7
Conv_8	Conv2D	(3, 3, 512, 256)	×	×	ReLU	Pool_3
Conv_9	Conv2D	(3, 3, 512, 512)	×	×	ReLU	Conv_8
Conv_10	Conv2D	(3, 3, 512, 512)	×	×	ReLU	Conv_9
Pool_4	Pooling2D	×	×	×	×	Conv_10
Conv_11	Conv2D	(3, 3, 512, 256)	×	×	ReLU	Pool_4
Conv_12	Conv2D	(3, 3, 512, 512)	×	×	ReLU	Conv_11
Conv_13	Conv2D	(3, 3, 512, 512)	×	×	ReLU	Conv_12
Pool_4	Pooling2D	×	×	×	×	Conv_13
flat	Flatten	×	×	×	×	Pool_4
FC-3	FC	1024	×	×	ReLU	flat
FC-4	FC	256	0.2	×	ReLU	FC-3
mean	FC	60	×	×	Linear	FC-4
var	FC	60	×	×	Linear	FC-4
z-sample	FC	60	×	×	Linear	[mean, var]
FC-5	FC	128	×	×	ReLU	conditions
MMD	FC	256	×	√	ReLU	[z-sample, FC-5]
FC-6	FC	1024	×	×	ReLU	MMD
FC-7	FC	4096	×	×	ReLU	FC-6
FC-7_reshaped	Reshape	×	×	×	FC-7	
Conv_transp_1	Conv2D Transpose	(3, 3, 512, 512)	×	×	ReLU	FC-7_reshaped
Conv_transp_2	Conv2D Transpose	(3, 3, 512, 512)	×	×	ReLU	Conv_transp_1
Conv_transp_3	Conv2D Transpose	(3, 3, 512, 512)	×	×	ReLU	Conv_transp_2
up_sample_1	UpSampling2D	×	×	×	×	Conv_transp_3
Conv_transp_4	Conv2D Transpose	(3, 3, 512, 512)	×	×	ReLU	up_sample_1
Conv_transp_5	Conv2D Transpose	(3, 3, 512, 512)	×	×	ReLU	Conv_transp_4
Conv_transp_6	Conv2D Transpose	(3, 3, 512, 512)	×	×	ReLU	Conv_transp_5
up_sample_2	UpSampling2D	×	×	×	×	Conv_transp_6
Conv_transp_7	Conv2D Transpose	(3, 3, 128, 256)	×	×	ReLU	up_sample_2
Conv_transp_8	Conv2D Transpose	(3, 3, 128, 128)	×	×	ReLU	Conv_transp_7
up_sample_3	UpSampling2D	×	×	×	×	Conv_transp_8
Conv_transp_9	Conv2D Transpose	(3, 3, 64, 128)	×	×	ReLU	up_sample_3
Conv_transp_10	Conv2D Transpose	(3, 3, 64, 64)	×	×	ReLU	Conv_transp_9
output	Conv2D Transpose	(1, 1, 3, 64)	×	×	ReLU	Conv_transp_10
Optimizer	Adam					
Learning Rate	0.001					
Leaky ReLU slope	0.2					
Batch Size	1024					
# of Epochs	5000					
α	0.001					
β	1000					

Supplementary Table 4: trVAE detailed architecture. We used the same architecture for all the examples in the paper. The input_dim parameter for each dataset is: IFN-β (2,000), H.poly (1,000).

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	input_dim	×	×	-	-
conditions	-	n_conditions	×	×	-	-
FC-1	FC	800	0.2	√	Leaky ReLU	[input, conditions]
FC-2	FC	800	0.2	√	Leaky ReLU	FC-1
FC-3	FC	128	0.2	√	Leaky ReLU	FC-2
mean	FC	50	×	×	Linear	FC-3
var	FC	50	×	×	Linear	FC-3
z-sample	FC	50	×	×	Linear	[mean, var]
MMD	FC	128	0.2	√	Leaky ReLU	[z-sample, conditions]
FC-4	FC	800	0.2	√	Leaky ReLU	MMD
FC-5	FC	800	0.2	√	Leaky ReLU	FC-3
output	FC	input_dim	×	×	ReLU	FC-4
Optimizer	Adam					
Learning Rate	0.001					
Leaky ReLU slope	0.2					
Batch Size	512					
# of Epochs	5000					
α	0.00001					
β	100					
η	100					

Supplementary Table 5: scGen detailed architecture.

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	input_dim	×	×	-	-
FC-1	FC	800	0.2	√	Leaky ReLU	input
FC-2	FC	800	0.2	√	Leaky ReLU	FC-1
FC-3	FC	128	0.2	√	Leaky ReLU	FC-2
mean	FC	100	×	×	Linear	FC-3
var	FC	100	×	×	Linear	FC-3
z	FC	100	×	×	Linear	[mean, var]
MMD	FC	128	0.2	√	Leaky ReLU	z
FC-4	FC	800	0.2	√	Leaky ReLU	MMD
FC-5	FC	800	0.2	√	Leaky ReLU	FC-3
output	FC	input_dim	×	×	ReLU	FC-4
Optimizer	Adam					
Learning Rate	0.001					
Leaky ReLU slope	0.2					
Batch Size	32					
# of Epochs	300					
α	0.00050					
β	100					
η	100					

Supplementary Table 6: CVAE detailed architecture.

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	input_dim	×	×	-	-
conditions	-	1	×	×	-	-
FC-1	FC	800	0.2	√	Leaky ReLU	[input, conditions]
FC-2	FC	800	0.2	√	Leaky ReLU	FC-1
FC-3	FC	128	0.2	√	Leaky ReLU	FC-2
mean	FC	50	×	×	Linear	FC-3
var	FC	50	×	×	Linear	FC-3
z-sample	FC	50	×	×	Linear	[mean, var]
MMD	FC	128	0.2	√	Leaky ReLU	[z-sample, conditions]
FC-4	FC	800	0.2	√	Leaky ReLU	MMD
FC-5	FC	800	0.2	√	Leaky ReLU	FC-3
output	FC	input_dim	×	×	ReLU	FC-4
Optimizer	Adam					
Learning Rate	0.001					
Leaky ReLU slope	0.2					
Batch Size	512					
# of Epochs	300					
α	0.001					

Supplementary Table 7: MMD-CVAE detailed architecture.

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	input_dim	×	×	-	-
conditions	-	1	×	×	-	-
FC-1	FC	800	0.2	√	Leaky ReLU	[input, conditions]
FC-2	FC	800	0.2	√	Leaky ReLU	FC-1
FC-3	FC	128	0.2	√	Leaky ReLU	FC-2
mean	FC	50	×	×	Linear	FC-3
var	FC	50	×	×	Linear	FC-3
z-sample	FC	50	×	×	Linear	[mean, var]
MMD	FC	128	0.2	√	Leaky ReLU	[z-sample, conditions]
FC-4	FC	800	0.2	√	Leaky ReLU	MMD
FC-5	FC	800	0.2	√	Leaky ReLU	FC-3
output	FC	input_dim	×	×	ReLU	FC-4
Optimizer	Adam					
Learning Rate	0.001					
Leaky ReLU slope	0.2					
Batch Size	512					
# of Epochs	500					
α	0.001					
β	1					

Supplementary Table 8: Style transfer GAN detailed architecture.

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	input_dim	×	×	-	-
FC-1	FC	700	0.5	√	Leaky ReLU	input
FC-2	FC	100	0.5	√	Leaky ReLU	FC-1
FC-3	FC	50	0.5	√	Leaky ReLU	FC-2
FC-4	FC	100	0.5	√	Leaky ReLU	FC-3
FC-5	FC	700	0.5	√	Leaky ReLU	FC-4
generator_out	FC	6,998	×	√	ReLU	FC-5
FC-6	FC	700	0.5	√	Leaky ReLU	generator_out
FC-7	FC	100	0.5	√	Leaky ReLU	FC-6
discriminator_out	FC	1	×	×	Sigmoid	FC-7
Generator Optimizer	Adam					
Discriminator Optimizer	Adam					
Optimizer	Adam					
Learning Rate	0.001					
Leaky ReLU slope	0.2					
# of Epochs	1000					

Supplementary Table 9: scVI detailed architecture.

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	input_dim	×	×	-	-
conditions	-	1	×	×	-	-
FC-1	FC	128	0.2	√	ReLU	input
mean	FC	10	×	×	Linear	FC-1
var	FC	10	×	×	Linear	FC-1
z	FC	10	×	×	Linear	[mean, var]
FC-2	FC	128	0.2	√	ReLU	[z, conditions]
output	FC	input_dim	×	×	ReLU	FC-2
Optimizer	Adam					
Learning Rate	0.001					
Batch Size	128					
# of Epochs	1000					
α	0.001					

Supplementary Table 10: SAUCIE detailed architecture.

Name	Operation	NoF/Kernel Dim.	Dropout	BN	Activation	Input
input	-	input_dim	×	×	-	-
conditions	-	1	×	×	-	-
FC-1	FC	512	×	√	Leaky ReLU	[input, conditions]
FC-2	FC	256	×	×	Leaky ReLU	FC-1
FC-3	FC	128	×	×	Leaky ReLU	FC-2
FC-4	FC	20	×	×	Leaky ReLU	FC-3
FC-5	FC	128	×	×	Leaky ReLU	FC-4
FC-6	FC	256	×	×	Leaky ReLU	FC-5
FC-7	FC	512	×	×	Leaky ReLU	FC-6
output	FC	input_dim	×	×	ReLU	FC-4
Optimizer	Adam					
Learning Rate	0.001					
Leaky ReLU slope	0.2					
Batch Size	256					
# of Epochs	1000					