

## A Additional background

### A.1 Priors from posteriors

Given the long-standing debates around the role, selection and treatment of the prior within Bayesian statistics, it is natural that the choice of  $p(\mathbf{z})$  in VAEs has come under scrutiny. While many of the traditional arguments revolve around principled points on objectivity, the primary issue for VAEs is the lack of expressiveness of the standard Normal distribution [25]. The shared concern is that the prior is often selected for practical but, ultimately, spurious reasons of technical convenience (e.g. conjugacy, reparametrization trick).

One solution is to simply replace the prior with the posterior. The apparent simplicity of this approach obscures the multiple issues that arise from double-dipping the data [44, 45]. Nevertheless the idea has endured – from the earlier proposal of posterior Bayes Factors as a solution to Lindley’s paradox [45], modern Empirical Bayes methods, to likelihood-free models such as the calibration in approximate Bayesian computation models [46], invoking the posterior ‘before its time’ is increasingly performed to anchor statistical models to a more objective foundation.

For VAEs, a well-known proposal is to replace the prior with a mixture of variational posteriors, formed using pseudo-observations  $\mathbf{u}_1, \dots, \mathbf{u}_K$  [26]. This Variational Mixture of Posteriors (VaMP) prior is given by

$$p^{\text{VaMP}}(\mathbf{z}) = \frac{1}{K} \sum_{k=1}^K q_\phi(\mathbf{z}|\mathbf{u}_k). \quad (8)$$

This results in a multi-modal prior, with the pseudo-observations learned by stochastic backpropagation along with the other parameters  $\theta, \phi$ . As we define in Section 4, the CoMP method adopts a similar non-parametric approach to defining the prior.

## B Proofs

We give the proof of Theorem 1, which is restated for convenience.

**Theorem 1.** *The following are equivalent: 1)  $\mathbf{z} \perp\!\!\!\perp c$  under distribution  $q$ , 2) for every  $c, c' \in \mathcal{C}$ ,  $q(\mathbf{z}|c) = q(\mathbf{z}|c')$ , 3) for every  $c \in \mathcal{C}$ ,  $q(\mathbf{z}|c) = q(\mathbf{z}|\neg c)$ , 4) the mutual information  $I(\mathbf{z}, c) = 0$  under distribution  $q$ , 5)  $\mathbf{z}$  cannot predict  $c$  better than random guessing.*

*Proof.* 1.  $\implies$  2. If  $\mathbf{z} \perp\!\!\!\perp c$ , then for every  $c, c' \in \mathcal{C}$ ,  $q(\mathbf{z}|c) = q(\mathbf{z}) = q(\mathbf{z}|c')$ .

2.  $\implies$  3. For  $c \in \mathcal{C}$ , by the definition of  $q(\mathbf{z}|\neg c)$  we have

$$q(\mathbf{z}|\neg c) = \frac{\sum_{c' \in \mathcal{C}, c' \neq c} p(c')q(\mathbf{z}|c')}{\sum_{c' \in \mathcal{C}, c' \neq c} p(c')} = \frac{\sum_{c' \in \mathcal{C}, c' \neq c} p(c')q(\mathbf{z}|c)}{\sum_{c' \in \mathcal{C}, c' \neq c} p(c')} = q(\mathbf{z}|c) \quad (9)$$

using condition 2.

3.  $\implies$  4. We have by definition of the mutual information under distribution  $q$

$$I(\mathbf{z}, c) = \mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} \left[ \log \frac{p(c)q(\mathbf{z}|c)}{p(c)q(\mathbf{z})} \right] \quad (10)$$

which can be written

$$= \mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} \left[ \log \frac{p(c)q(\mathbf{z}|c)}{p(c)[p(c)q(\mathbf{z}|c) + (1 - p(c))q(\mathbf{z}|\neg c)]} \right] \quad (11)$$

applying condition 3. gives

$$= \mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} \left[ \log \frac{p(c)q(\mathbf{z}|c)}{p(c)[p(c)q(\mathbf{z}|c) + (1 - p(c))q(\mathbf{z}|c)]} \right] \quad (12)$$

$$= \mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} \left[ \log \frac{p(c)q(\mathbf{z}|c)}{p(c)q(\mathbf{z}|c)} \right] \quad (13)$$

$$= 0. \quad (14)$$

532 4.  $\implies$  5.<sup>1</sup> Let  $Q(c|\mathbf{z})$  be some prediction rule for predicting  $c$  using  $\mathbf{z}$ . By Gibbs' Inequality, we  
 533 have

$$I(\mathbf{z}, c) \geq \mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} \left[ \log \frac{Q(c|\mathbf{z})}{p(c)} \right]. \quad (15)$$

534 Since  $I(\mathbf{z}, c) = 0$ , we have

$$\mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} [\log Q(c|\mathbf{z})] \leq \mathbb{E}_{p(\mathbf{x}, c)} [\log p(c)]. \quad (16)$$

535 Observe that the left hand side above is the expected log-likelihood for the prediction rule  $Q$ , whilst the  
 536 right hand side is the the log-likelihood for random guessing of  $c$  using only its marginal distribution  
 537  $p(c)$ . We see that random guessing obtains a log-likelihood which is at least as good as that obtained  
 538 using the rule  $Q$ .

539 5.  $\implies$  1. Consider the prediction rule

$$Q^*(c|\mathbf{z}) := \frac{p(c)q(\mathbf{z}|c)}{q(\mathbf{z})}. \quad (17)$$

540 By condition 5., we have

$$\mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} [\log Q^*(c|\mathbf{z})] \leq \mathbb{E}_{p(\mathbf{x}, c)} [\log p(c)]. \quad (18)$$

541 Hence,

$$\mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} \left[ \log \frac{p(c)q(\mathbf{z}|c)}{p(c)q(\mathbf{z})} \right] \leq 0. \quad (19)$$

542 By Gibbs' Inequality,

$$\mathbb{E}_{p(\mathbf{x}, c)q(\mathbf{z}|\mathbf{x}, c)} \left[ \log \frac{p(c)q(\mathbf{z}|c)}{p(c)q(\mathbf{z})} \right] \geq 0 \quad (20)$$

543 with equality if and only if  $p(c)q(\mathbf{z}|c) = p(c)q(\mathbf{z})$ . By (19), equality does hold, so  $p(c)q(\mathbf{z}|c) =$   
 544  $p(c)q(\mathbf{z})$  meaning  $\mathbf{z} \perp\!\!\!\perp c$  under distribution  $q$ .  $\square$

545 We next restate and prove Theorem 2.

546 **Theorem 2.** *The CoMP misalignment penalty satisfies*

$$\begin{aligned} & \mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i)q(\mathbf{z}_i|\mathbf{x}_i, c_i)} \left[ \frac{1}{B} \sum_{i=1}^B \log \left( \frac{1}{|I_{c_i}|} \sum_{j \in I_{c_i}} q(\mathbf{z}_i|\mathbf{x}_j, c_i) \right) - \log \left( \frac{1}{|I_{\neg c_i}|} \sum_{j \in I_{\neg c_i}} q(\mathbf{z}_i|\mathbf{x}_j, c_j) \right) \right] \\ & \geq \sum_{c \in \mathcal{C}} p(c) \text{KL}[q(\mathbf{z}|c) \| q(\mathbf{z}|\neg c)] \end{aligned}$$

547 and the bound becomes tight as  $B \rightarrow \infty$ .

548 *Proof.* First, by linearity of the expectation we have

$$\begin{aligned} & \mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i)q(\mathbf{z}_i|\mathbf{x}_i, c_i)} \left[ \frac{1}{B} \sum_{i=1}^B \log \left( \frac{1}{|I_{c_i}|} \sum_{j \in I_{c_i}} q(\mathbf{z}_i|\mathbf{x}_j, c_i) \right) - \log \left( \frac{1}{|I_{\neg c_i}|} \sum_{j \in I_{\neg c_i}} q(\mathbf{z}_i|\mathbf{x}_j, c_j) \right) \right] \\ & = \mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i)q(\mathbf{z}_i|\mathbf{x}_i, c_i)} \left[ \log \left( \frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1|\mathbf{x}_j, c_1) \right) - \log \left( \frac{1}{|I_{\neg c_1}|} \sum_{j \in I_{\neg c_1}} q(\mathbf{z}_1|\mathbf{x}_j, c_j) \right) \right]. \end{aligned} \quad (21)$$

549 Focusing on the latter term, Jensen's Inequality gives

$$\mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i)q(\mathbf{z}_i|\mathbf{x}_i, c_i)} \left[ -\log \left( \frac{1}{|I_{\neg c_1}|} \sum_{j \in I_{\neg c_1}} q(\mathbf{z}_1|\mathbf{x}_j, c_j) \right) \right] \quad (22)$$

$$\geq \mathbb{E}_{p(\mathbf{x}_1, c_1)q(\mathbf{z}_1|\mathbf{x}_1, c_1)} \left[ -\log \left( \mathbb{E}_{\prod_{i>1}^B p(\mathbf{x}_i, c_i)q(\mathbf{z}_i|\mathbf{x}_i, c_i)} \left[ \frac{1}{|I_{\neg c_1}|} \sum_{j \in I_{\neg c_1}} q(\mathbf{z}_1|\mathbf{x}_j, c_j) \right] \right) \right] \quad (23)$$

$$= \mathbb{E}_{p(\mathbf{x}_1, c_1)q(\mathbf{z}_1|\mathbf{x}_1, c_1)} [-\log q(\mathbf{z}_1|\neg c_1)]. \quad (24)$$

---

<sup>1</sup>We interpret 'better prediction' in condition 5. as achieving a higher expected log-likelihood.

550 For the other term, we take our inspiration from recent work on experimental design [47]. We have

$$\mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i) q(\mathbf{z}_i | \mathbf{x}_i, c_i)} \left[ \log \left( \frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1 | \mathbf{x}_j, c_i) \right) \right] \quad (25)$$

$$= \mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i) q(\mathbf{z}_i | \mathbf{x}_i, c_i)} \left[ \log q(\mathbf{z}_1 | c_1) + \log \left( \frac{\frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1 | \mathbf{x}_j, c_i)}{q(\mathbf{z}_1 | c_1)} \right) \right] \quad (26)$$

$$= \mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i) q(\mathbf{z}_i | \mathbf{x}_i, c_i)} [\log q(\mathbf{z}_1 | c_1)] + \Delta \quad (27)$$

551 Then applying the tower rule with variable  $c_1, |I_{c_1}|$  we have the difference term equal to

$$\Delta = \mathbb{E}_{c_1, |I_{c_1}|} \left[ \mathbb{E}_{\prod_{i=1}^{|I_{c_1}|} p(\mathbf{x}_i | c_1) q(\mathbf{z}_1 | \mathbf{x}_1, c_1)} \left[ \log \left( \frac{\frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1 | \mathbf{x}_j, c_i)}{q(\mathbf{z}_1 | c_1)} \right) \right] \right] \quad (28)$$

$$= \mathbb{E}_{c_1, |I_{c_1}|} \left[ \mathbb{E}_{\prod_{i=1}^{|I_{c_1}|} p(\mathbf{x}_i | c_1) q(\mathbf{z}_1 | \mathbf{x}_1, c_1)} \left[ \log \left( \frac{\prod_{i=1}^{|I_{c_1}|} p(\mathbf{x}_i | c_1) \frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1 | \mathbf{x}_j, c_i)}{\prod_{i=1}^{|I_{c_1}|} p(\mathbf{x}_i | c_1) q(\mathbf{z}_1 | c_1)} \right) \right] \right] \quad (29)$$

552 Now observe that  $\mathbf{z}_1, \dots, \mathbf{z}_{|I_{c_1}|}$  are equal in distribution, so we can change the sampling distribution  
553 to be over

$$\prod_{i=1}^{|I_{c_1}|} p(\mathbf{x}_i | c_1) \frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1 | \mathbf{x}_j, c_i) \quad (30)$$

554 which amounts to choosing at random which of the  $\mathbf{x}_1, \dots, \mathbf{x}_{|I_{c_1}|}$  to sample  $\mathbf{z}_1$  from. Finally, we  
555 observe that

$$\prod_{i=1}^{|I_{c_1}|} p(\mathbf{x}_i | c_1) q(\mathbf{z}_1 | c_1) \quad (31)$$

556 is a normalised distribution over  $\mathbf{x}_1, \dots, \mathbf{x}_{|I_{c_1}|}, \mathbf{z}_1$ . Thus we can write  $\Delta$  as the following expected  
557 KL divergence

$$\Delta = \mathbb{E}_{c_1, |I_{c_1}|} \left[ \text{KL} \left[ \prod_{i=1}^{|I_{c_1}|} p(\mathbf{x}_i | c_1) \frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1 | \mathbf{x}_j, c_i) \left\| \prod_{i=1}^{|I_{c_1}|} p(\mathbf{x}_i | c_1) q(\mathbf{z}_1 | c_1) \right\| \right] \right]. \quad (32)$$

558 Since the KL divergence is non-negative, we have shown that  $\Delta \geq 0$ . Therefore

$$\mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i) q(\mathbf{z}_i | \mathbf{x}_i, c_i)} \left[ \log \left( \frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1 | \mathbf{x}_j, c_i) \right) \right] \geq \mathbb{E}_{p(\mathbf{x}_1, c_1) q(\mathbf{z}_1 | \mathbf{x}_1, c_1)} [\log q(\mathbf{z}_1 | c_1)]. \quad (33)$$

559 Putting these two results together, we have shown that

$$\mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i) q(\mathbf{z}_i | \mathbf{x}_i, c_i)} \left[ \frac{1}{B} \sum_{i=1}^B \log \left( \frac{1}{|I_{c_i}|} \sum_{j \in I_{c_i}} q(\mathbf{z}_i | \mathbf{x}_j, c_i) \right) - \log \left( \frac{1}{|I_{\neg c_i}|} \sum_{j \in I_{\neg c_i}} q(\mathbf{z}_i | \mathbf{x}_j, c_j) \right) \right] \quad (34)$$

$$\geq \mathbb{E}_{\prod_{i=1}^B p(\mathbf{x}_i, c_i) q(\mathbf{z}_i | \mathbf{x}_i, c_i)} \left[ \log \frac{q(\mathbf{z}_1 | c_1)}{q(\mathbf{z}_1 | \neg c_1)} \right] \quad (35)$$

$$= \mathbb{E}_{p(\mathbf{x}_1, c_1) q(\mathbf{z}_1 | \mathbf{x}_1, c_1)} \left[ \log \frac{q(\mathbf{z}_1 | c_1)}{q(\mathbf{z}_1 | \neg c_1)} \right] \quad (36)$$

$$= \sum_{c \in \mathcal{C}} p(c_1) \mathbb{E}_{p(\mathbf{x}_1 | c_1) q(\mathbf{z}_1 | \mathbf{x}_1, c_1)} \left[ \log \frac{q(\mathbf{z}_1 | c_1)}{q(\mathbf{z}_1 | \neg c_1)} \right] \quad (37)$$

$$= \sum_{c \in \mathcal{C}} p(c_1) \mathbb{E}_{q(\mathbf{z}_1 | c_1)} \left[ \log \frac{q(\mathbf{z}_1 | c_1)}{q(\mathbf{z}_1 | \neg c_1)} \right] \quad (38)$$

$$= \sum_{c \in \mathcal{C}} p(c) \text{KL}[q(\mathbf{z} | c) \| q(\mathbf{z} | \neg c)]. \quad (39)$$

560 Finally, as  $B \rightarrow \infty$ , the Strong Law of Large Numbers implies that

$$\log \left( \frac{1}{|I_{c_1}|} \sum_{j \in I_{c_1}} q(\mathbf{z}_1 | \mathbf{x}_j, c_i) \right) \rightarrow q(\mathbf{z}_i | c_i), \quad (40)$$

$$\log \left( \frac{1}{|I_{\neg c_i}|} \sum_{j \in I_{\neg c_i}} q(\mathbf{z}_i | \mathbf{x}_j, c_j) \right) \rightarrow q(\mathbf{z}_i | \neg c_i) \quad (41)$$

561 so (under mild technical assumptions) we conclude that the bound becomes tight in this limit. This  
562 completes the proof.  $\square$

## 563 C Analysing CoMP gradients

564 We provide additional details and a full derivation for the results discussed in Section 4.1. To analyse  
565 MMD and CoMP gradients, we focus on the two specific cases that highlight the similarities between  
566 these methods, revealing the remaining differences. Specifically, we consider MMD with a simple  
567 unnormalised Radial Basis Kernel [34]

$$k(\mathbf{z}, \mathbf{z}') = e^{-\|\mathbf{z} - \mathbf{z}'\|^2}, \quad (42)$$

568 and a Gaussian variational posterior family with fixed covariance matrix  $\frac{1}{2}I$

$$q(\mathbf{z} | \mathbf{x}, c) \propto e^{-\|\mathbf{z} - \boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x}, c)\|^2}. \quad (43)$$

569 We also assume just two conditions  $|\mathcal{C}| = 2$ . For an MMD penalty, the simplest form of the Kernel  
570 Two-sample Test statistic [17] with batch size  $B$  can be written as follows

$$\mathcal{P}_{\text{MMD}} = \sum_{i=1}^B \mathcal{P}_{\text{MMD}}(\mathbf{z}_i, c_i) \quad (44)$$

$$= \sum_{i=1}^B \left( \frac{1}{|I_{c_i}|^2} \sum_{j \in I_{c_i}} e^{-\|\mathbf{z}_i - \mathbf{z}_j\|^2} - \frac{1}{|I_{\neg c_i}| |I_{c_i}|} \sum_{j \in I_{\neg c_i}} e^{-\|\mathbf{z}_i - \mathbf{z}_j\|^2} \right), \quad (45)$$

571 taking gradients with respect to  $\mathbf{z}_i$  gives us

$$\nabla_{\mathbf{z}_i} \mathcal{P}_{\text{MMD}}(\mathbf{z}_i, c_i) = \frac{2}{|I_{c_i}|^2} \sum_{j \in I_{c_i}} e^{-\|\mathbf{z}_i - \mathbf{z}_j\|^2} (\mathbf{z}_j - \mathbf{z}_i) - \frac{2}{|I_{\neg c_i}| |I_{c_i}|} \sum_{j \in I_{\neg c_i}} e^{-\|\mathbf{z}_i - \mathbf{z}_j\|^2} (\mathbf{z}_j - \mathbf{z}_i), \quad (46)$$

the gradients of the total penalty are

$$\nabla_{\mathbf{z}_i} \mathcal{P}_{\text{MMD}} = \frac{4}{|I_{c_i}|^2} \sum_{j \in I_{c_i}} e^{-\|\mathbf{z}_i - \mathbf{z}_j\|^2} (\mathbf{z}_j - \mathbf{z}_i) - \frac{4}{|I_{\neg c_i}| |I_{c_i}|} \sum_{j \in I_{\neg c_i}} e^{-\|\mathbf{z}_i - \mathbf{z}_j\|^2} (\mathbf{z}_j - \mathbf{z}_i). \quad (47)$$

The CoMP penalty (ignoring normalising constants) is

$$\mathcal{P}_{\text{CoMP}} = \sum_{i=1}^B \mathcal{P}_{\text{CoMP}}(\mathbf{z}_i, c_i) \quad (48)$$

$$= \frac{1}{B} \sum_{i=1}^B \log \left( \frac{1}{|I_{c_i}|} \sum_{j \in I_{c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2} \right) - \log \left( \frac{1}{|I_{\neg c_i}|} \sum_{j \in I_{\neg c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2} \right), \quad (49)$$

if we take the gradient with respect to  $\mathbf{z}_i$  we obtain

$$\nabla_{\mathbf{z}_i} \mathcal{P}_{\text{CoMP}}(\mathbf{z}_i, c_i) = \frac{2 \sum_{j \in I_{c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2} (\boldsymbol{\mu}_{\mathbf{z}_j} - \mathbf{z}_i)}{B \sum_{j \in I_{c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2}} - \frac{2 \sum_{j \in I_{\neg c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2} (\boldsymbol{\mu}_{\mathbf{z}_j} - \mathbf{z}_i)}{B \sum_{j \in I_{\neg c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2}} \quad (50)$$

where  $\boldsymbol{\mu}_{\mathbf{z}_j} = \boldsymbol{\mu}(\mathbf{x}_i, c_i)$  is the variational mean for  $\mathbf{z}_j$ . The gradient of the full penalty with respect to  $\boldsymbol{\mu}_{\mathbf{z}_i}$ , noting  $\mathbf{z}_i = \boldsymbol{\mu}_{\mathbf{z}_i} + \boldsymbol{\epsilon}_i$ , is

$$\begin{aligned} \nabla_{\boldsymbol{\mu}_{\mathbf{z}_i}} \mathcal{P}_{\text{CoMP}} &= \frac{2 \sum_{j \in I_{c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2} (\boldsymbol{\mu}_{\mathbf{z}_j} - \mathbf{z}_i)}{B \sum_{j \in I_{c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2}} - \frac{2 \sum_{j \in I_{\neg c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2} (\boldsymbol{\mu}_{\mathbf{z}_j} - \mathbf{z}_i)}{B \sum_{j \in I_{\neg c_i}} e^{-\|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2}} \\ &\quad + 2 \sum_{j \in I_{c_i}} \frac{e^{-\|\mathbf{z}_j - \boldsymbol{\mu}_{\mathbf{z}_i}\|^2} (\mathbf{z}_j - \boldsymbol{\mu}_{\mathbf{z}_i})}{B \sum_{k \in I_{c_i}} e^{-\|\mathbf{z}_j - \boldsymbol{\mu}_{\mathbf{z}_k}\|^2}} - 2 \sum_{j \in I_{\neg c_i}} \frac{e^{-\|\mathbf{z}_j - \boldsymbol{\mu}_{\mathbf{z}_i}\|^2} (\mathbf{z}_j - \boldsymbol{\mu}_{\mathbf{z}_i})}{B \sum_{k \in I_{\neg c_i}} e^{-\|\mathbf{z}_j - \boldsymbol{\mu}_{\mathbf{z}_k}\|^2}}. \end{aligned} \quad (51)$$

Finally, to see the connection with nearest neighbour methods, we repeat this analysis with Gaussian posterior with fixed variance  $\sigma^2$ . The gradient term is then

$$\begin{aligned} \nabla_{\mathbf{z}_i} \mathcal{P}_{\text{CoMP}}(\mathbf{z}_i, c_i) &= \frac{\frac{1}{\sigma^2} \sum_{j \in I_{c_i}} e^{-\frac{1}{2\sigma^2} \|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2} (\boldsymbol{\mu}_{\mathbf{z}_j} - \mathbf{z}_i)}{B \sum_{j \in I_{c_i}} e^{-\frac{1}{2\sigma^2} \|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2}} \\ &\quad - \frac{\frac{1}{\sigma^2} \sum_{j \in I_{\neg c_i}} e^{-\frac{1}{2\sigma^2} \|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2} (\boldsymbol{\mu}_{\mathbf{z}_j} - \mathbf{z}_i)}{B \sum_{j \in I_{\neg c_i}} e^{-\frac{1}{2\sigma^2} \|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2}}. \end{aligned} \quad (52)$$

As  $\sigma \rightarrow 0$ , we have

$$\frac{e^{-\frac{1}{2\sigma^2} \|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_k}\|^2}}{\sum_{j \in I_{\neg c_i}} e^{-\frac{1}{2\sigma^2} \|\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{z}_j}\|^2}} \rightarrow \delta_{k \text{nn}_i} \quad (53)$$

where  $\text{nn}_i$  is the index of the nearest neighbour to  $\mathbf{z}_i$  among the set  $\{\mathbf{z}_j : j \in I_{\neg c_i}\}$ , i.e.

$$\text{nn}_i = \underset{j \in I_{\neg c_i}}{\text{argmin}} \|\mathbf{z}_i - \mathbf{z}_j\| \quad (54)$$

indicating that the gradient between  $\mathbf{z}_i$  and  $\mathbf{z}_{\text{nn}_i}$  becomes the dominant term in the limit.

## D Experimental details

### D.1 Dataset details and data processing

Details of the datasets and for each one, how we processed the data.

**Tumour / Cell Line** This dataset, as used in the experiments in *Cellinger* [21]<sup>2</sup>, consists of bulk expression profiles for tumours (n=12,236) and cancer cell-lines (n=1,249) across 39 different cancer types. The tumour samples are taken from The Cancer Genome Atlas (TCGA) [48]<sup>3</sup> and Therapeutically Applicable Research To Generate Effective Treatments (TARGET) [49]<sup>4</sup> and was compiled by the Treehouse Childhood Cancer Initiative at the UC Santa Cruz Genomics Institute [50]<sup>5</sup>. The cell lines are from the Cancer Cell Line Encyclopedia (CCLE) [51]<sup>6</sup>. The condition variable is the tumour / cell line label. The expression data is restricted to the intersecting subset of 16,612 protein-coding genes and are TPM  $\log_2$ -transformed values.

In our experiments, as is common practice in omics data analysis (e.g. [21]), we pre-process the data by filtering out low-variance genes. Here we select the 8,000 highest variance genes across cell-lines and tumours separately and take the union to give a final feature set of 9,468 genes.

For our calculation of  $m\text{-kBET}_{k,\alpha}$  and  $\tilde{s}_{k,c}$  metrics we only include cancer types with at least 400 samples (i.e.  $4 \times k$  for our choice of  $k = 100$ ) to ensure that the metric retains the ability to evaluate *local* mixing. 15 cancer types pass this threshold, representing 82% of all samples.

**Single-cell PBMCs** This dataset consists of single-cell expression profiles of 14,053 genes for peripheral blood mononuclear cells (PBMCs), various immune cell types pooled from eight lupus patient samples. 7,217 of the cells were stimulated with interferon (IFN)- $\beta$  while 6,359 were left untreated (control). [41]. This dataset has been used in [9] and [10] previously. We obtained an annotated and pre-filtered dataset from [10]<sup>7</sup> <sup>8</sup>, which includes metadata on immune cell type labels along the condition label; stimulated or control.

The file was read into scanpy [52] and pre-processed using `sc.pp.normalize_total(data, inplace=True)`, which normalises the data such that each cell has a total count equal to the median total count across all cells. The normalised counts were then  $\log(x + 1)$  transformed using the scanpy function, `sc.pp.log1p(data)`. We selected the top 2,000 most variable genes using the scanpy function, `sc.pp.highly_variable_genes(data, flavor="seurat", n_top_genes=2000)`.

We obtained the top 50 differentially expressed (DE) genes between stimulated and control cells for each cell type by subsetting the data for each cell type and using scanpy's function `sc.tl.rank_genes_groups(cell_type_data, groupby='stim', n_genes=50, method='wilcoxon')`, which ranks genes based on a Wilcoxon rank-sum test. For each cell type, we separated the top 50 DE genes into those that were up-regulated and down-regulated by IFN- $\beta$  stimulation.

**UCI Adult Income** This dataset is derived from the 1994 United States census bureau and contains information relating to education, marriage status, ethnicity, self-reported gender of census participants and a binary high / low income label (\$50,000 threshold). Data was downloaded from the UCI Machine Learning Repository [22].

## 620 D.2 Evaluation metrics

621 Let  $\{(\mathbf{x}_i, c_i, d_i)\}_{i=1}^N$  be the dataset  $N$  samples with  $c_i \in \{0, 1\}$  the binary condition variable and  
622  $d_i \in \{d^{(m)}\}_{m=1}^M$  an additional discrete random variable of interest not used in training.

623 We start with some housekeeping definitions of sample index subsets of the full dataset  $[1, N]$ . Let  
624  $N_{i,k}$  be the index set of the  $k$  nearest-neighbors of  $z_i$ . Let  $I_c$  be the index set of samples has  $c_j = c$ ,  
625 and  $J_d$  for samples that has  $d_j = d$ .

<sup>2</sup>[www.nature.com/articles/s41467-020-20294-x#data-availability](https://www.nature.com/articles/s41467-020-20294-x#data-availability)

<sup>3</sup>[www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga](https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga)

<sup>4</sup>[ocg.cancer.gov/programs/target](https://ocg.cancer.gov/programs/target)

<sup>5</sup>[https://treehousegenomics.soe.ucsc.edu/public-data/previous-compendia.html#tumor\\_v10\\_polyA](https://treehousegenomics.soe.ucsc.edu/public-data/previous-compendia.html#tumor_v10_polyA)

<sup>6</sup>[portals.broadinstitute.org/ccle](https://portals.broadinstitute.org/ccle)

<sup>7</sup>[https://github.com/theislabs/trVAE\\_reproducibility](https://github.com/theislabs/trVAE_reproducibility)

<sup>8</sup>[https://drive.google.com/drive/folders/1n1SLbXha40H7j7zZ0zZAXrj\\_-2kczgl8](https://drive.google.com/drive/folders/1n1SLbXha40H7j7zZ0zZAXrj_-2kczgl8), filename: kang\_count.h5ad

From [42],  $\text{kBET}_{k,\alpha}$  is the proportion of rejected null hypotheses from the set of separate  $\chi^2$  independence tests, with significance threshold  $\alpha$ , on the  $k$  nearest-neighbours of every sample. If we let  $\text{kBET}_{k,\alpha}^d$  be the metric calculated on the filtered sub-population with index set  $J_d$ , then we define a mean kBET metric as

$$\text{m-kBET}_{k,\alpha} := \frac{1}{M} \sum_{m=1}^M \text{kBET}_{k,\alpha}^{d^{(m)}} \quad (55)$$

We also consider local Silhouette Coefficients [43]

$$s_{k,c} := \frac{1}{|I_c|} \sum_{i \in I_c} \frac{b_{i,k} - a_{i,k}}{\max(a_{i,k}, b_{i,k})}, \quad s_k := \frac{1}{|I_c \cup I_{\neg c}|} \sum_{i \in I_c \cup I_{\neg c}} \frac{b_{i,k} - a_{i,k}}{\max(a_{i,k}, b_{i,k})}, \quad (56)$$

where  $a_{i,k}$  and  $b_{i,k}$  are the mean Euclidean distances between  $\mathbf{z}_i$  and all other sample points in the  $k$  nearest-neighbor set that are of the same and different condition variable respectively; i.e.

$$a_{i,k} \equiv \frac{1}{|N_{i,k} \cap I_{c_i}|} \sum_{j \in N_{i,k} \cap I_{c_i}} \|\mathbf{z}_i - \mathbf{z}_j\|, \quad b_{i,k} \equiv \frac{1}{|N_{i,k} \cap I_{\neg c_i}|} \sum_{j \in N_{i,k} \cap I_{\neg c_i}} \|\mathbf{z}_i - \mathbf{z}_j\|. \quad (57)$$

Similar to the mean kBET metric, we can also define a mean Silhouette Coefficient  $\tilde{s}_{k,c}$  as follows. We first define

$$s_{k,c}^d := \frac{1}{|I_c \cap J_d|} \sum_{i \in I_c \cap J_d} \frac{b_{i,k,d} - a_{i,k,d}}{\max(a_{i,k,d}, b_{i,k,d})}, \quad (58)$$

with

$$a_{i,k,d} \equiv \frac{1}{|N_{i,k} \cap I_{c_i} \cap J_d|} \sum_{j \in N_{i,k} \cap I_{c_i} \cap J_d} \|\mathbf{z}_i - \mathbf{z}_j\|, \quad (59)$$

$$b_{i,k,d} \equiv \frac{1}{|N_{i,k} \cap I_{\neg c_i} \cap J_d|} \sum_{j \in N_{i,k} \cap I_{\neg c_i} \cap J_d} \|\mathbf{z}_i - \mathbf{z}_j\|.$$

Then the mean local Silhouette Coefficient is

$$\tilde{s}_{k,c} := \frac{1}{M} \sum_{m=1}^M s_{k,c}^{d^{(m)}}, \quad (60)$$

with  $\tilde{s}_k$  defined analogously to  $s_k$  in (56). A well-mixed representation that keeps samples with identical  $d_i$  together will have low values of  $\text{m-kBET}_{k,\alpha}$  and  $\tilde{s}_{k,c}$  close to zero. Higher values near 1 would indicate either an undesirable dependency between  $\mathbf{z}$  and  $c$  in the form of identifiable clusters around values of  $c$ , a censoring process that fails to preserve the clustering with respect to  $d$ , or a combination of both.

### D.3 Tumour / Cell Line representations for individual cancer types

As the cancer type labels are not used in training, there is the possibility that cell lines of one cancer type will cluster around tumours of a different type. Here we illustrate this risk by examining the subset of Prostate Cancer latent representations inferred by CoMP and trVAE, where the majority of tumours and cell lines for this cancer type can be found in a single group. As shown in Figure 7, trVAE has cell-lines from other cancer types erroneously placed within the prostate cancer cluster; CoMP, on the other hand, maintains a relatively high level of specificity with fewer non-prostate cancer cell lines present. On average across all cancer types, this favourable behaviour of CoMP is reflected in the low  $\tilde{s}$  and m-kBET scores.

### D.4 Condition mixing metrics for single-cell PBMC expression data

In this section we present additional results of our experiments on the single-cell PBMC expression data evaluating the condition mixing capabilities of CoMP. We focus on the two mixing metrics –  $s_k$  and  $\text{kBET}_{k,\alpha}$  – and report both the mean values and their standard errors over 10 random model initialisations. We have the following three sets of experiments:

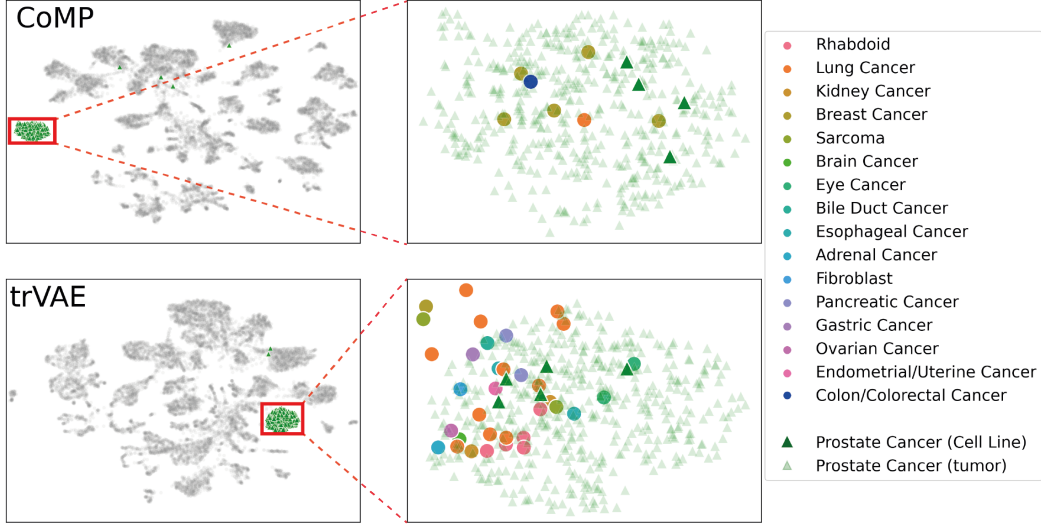


Figure 7: 2D UMAP projection of the CoMP and trVAE posterior means of  $\mathbf{z}_i$  from Tumour / Cell Line data and the detailed Prostate Cancer tumour sample clusters.

Table 3: kBET metrics for single-cell PBMC dataset with  $k = 100$  and  $\alpha = 0.1$ . Here, kBET and m-kBET refer to the mean kBET and mean m-kBET across 10 random seeds for each model, respectively. SEM represents the standard error of the mean.

Model	kBET $_{k,\alpha}$	kBET $_{k,\alpha} \pm \text{SEM}$	m-kBET $_{k,\alpha}$	m-kBET $_{k,\alpha} \pm \text{SEM}$
VAE	0.9788	(0.9754, 0.9821)	0.9443	(0.9351, 0.9535)
CVAE	0.9056	(0.8973, 0.9139)	0.8202	(0.8060, 0.8344)
VFAE	0.4753	(0.4660, 0.4847)	0.4067	(0.3942, 0.4192)
trVAE	0.5082	(0.4946, 0.5218)	0.3819	(0.3683, 0.3955)
CoMP	0.1211	(0.0845, 0.1577)	0.0681	(0.0388, 0.0975)

**Benchmarking** In Tables 3 to 4 we benchmark CoMP against the four other VAE models and show that CoMP outperforms the other models by significant margins on both metrics.

**Cell type level evaluation** In Table 5 we evaluate the mixing at a cell type level, where the strong mixing capabilities of CoMP is seen consistently across cell types. In particular, we highlight the good mixing of the CD14 Mono cell type by CoMP relative to the other penalised models.

**CoMP penalty scale  $\gamma$**  In Tables 6 to 8 we explore the effect of varying the CoMP penalty scale  $\gamma$  at both the population and cell type levels. Here we see that the optimum value is  $\sim 1$ .

Table 4: Silhouette Coefficient metrics for single-cell PBMC dataset with  $k = 100$ . Here,  $s$  and  $\tilde{s}$  refer to the mean  $s$  and mean  $\tilde{s}$  across 10 random seeds for each model, respectively. SEM represents the standard error of the mean.

Model	$s_k$	$s_k \pm \text{SEM}$	$\tilde{s}_k$	$\tilde{s}_k \pm \text{SEM}$
VAE	0.6354	(0.6303, 0.6404)	0.5249	(0.5172, 0.5326)
CVAE	0.4872	(0.4805, 0.4939)	0.3856	(0.3802, 0.3910)
VFAE	0.0501	(0.0457, 0.0544)	0.0793	(0.0731, 0.0855)
trVAE	0.0651	(0.0596, 0.0705)	0.0605	(0.0574, 0.0636)
CoMP	-0.0026	(-0.0032, -0.0020)	-0.0013	(-0.0027, 0.0001)

Table 5: Cell type specific kBET and Silhouette Coefficient metrics for the single-cell PBMC dataset summarised for 10 random seeds for each model. Metrics represent the mean value across the 10 random seeds for each model. Here,  $k = 100$  and  $\alpha = 0.1$ .

Cell type	Model	kBET $_{k,\alpha}$	kBET $_{k,\alpha} \pm \text{SEM}$	$s_k$	$s_k \pm \text{SEM}$
B	VAE	0.9724	(0.9683, 0.9765)	0.5375	(0.5225, 0.5526)
B	CVAE	0.9016	(0.8964, 0.9068)	0.2884	(0.2783, 0.2985)
B	VFAE	0.3892	(0.3357, 0.4426)	0.0263	(0.0205, 0.0320)
B	trVAE	0.2697	(0.2243, 0.3151)	0.0102	(0.0083, 0.0121)
B	CoMP	0.0110	(0.0002, 0.0217)	-0.0040	(-0.0050, -0.0030)
CD14 Mono	VAE	1.0000	(1.0000, 1.0000)	0.9388	(0.9356, 0.9420)
CD14 Mono	CVAE	1.0000	(1.0000, 1.0000)	0.9373	(0.9317, 0.9428)
CD14 Mono	VFAE	0.8192	(0.8084, 0.8300)	0.0548	(0.0486, 0.0610)
CD14 Mono	trVAE	0.9360	(0.9213, 0.9508)	0.1579	(0.1414, 0.1743)
CD14 Mono	CoMP	0.1709	(0.0817, 0.2601)	0.0003	(-0.0015, 0.0022)
CD16 Mono	VAE	1.0000	(1.0000, 1.0000)	0.7462	(0.7168, 0.7756)
CD16 Mono	CVAE	1.0000	(1.0000, 1.0000)	0.7455	(0.7204, 0.7706)
CD16 Mono	VFAE	0.8796	(0.8572, 0.9020)	0.2328	(0.1860, 0.2796)
CD16 Mono	trVAE	0.8059	(0.7856, 0.8263)	0.0535	(0.0493, 0.0576)
CD16 Mono	CoMP	0.0947	(0.0184, 0.1711)	0.0005	(-0.0011, 0.0021)
CD4 T	VAE	0.9964	(0.9960, 0.9968)	0.5069	(0.4932, 0.5206)
CD4 T	CVAE	0.9104	(0.9028, 0.9179)	0.2103	(0.1956, 0.2250)
CD4 T	VFAE	0.1460	(0.1358, 0.1562)	0.0035	(0.0030, 0.0039)
CD4 T	trVAE	0.2236	(0.1985, 0.2487)	0.0042	(0.0036, 0.0047)
CD4 T	CoMP	0.0538	(0.0442, 0.0634)	-0.0022	(-0.0028, -0.0016)
CD8 T	VAE	0.9041	(0.8633, 0.9448)	0.2828	(0.2714, 0.2942)
CD8 T	CVAE	0.4805	(0.4080, 0.5529)	0.0653	(0.0579, 0.0728)
CD8 T	VFAE	0.0397	(0.0307, 0.0486)	0.0094	(0.0083, 0.0106)
CD8 T	trVAE	0.0317	(0.0224, 0.0409)	0.0071	(0.0052, 0.0090)
CD8 T	CoMP	0.0634	(0.0437, 0.0830)	-0.0000	(-0.0008, 0.0008)
DC	VAE	1.0000	(1.0000, 1.0000)	0.6834	(0.6678, 0.6991)
DC	CVAE	1.0000	(1.0000, 1.0000)	0.6723	(0.6598, 0.6847)
DC	VFAE	0.7095	(0.6715, 0.7476)	0.2901	(0.2733, 0.3069)
DC	trVAE	0.6286	(0.5972, 0.6600)	0.2339	(0.2225, 0.2453)
DC	CoMP	0.0784	(0.0329, 0.1239)	0.0034	(-0.0027, 0.0096)
NK	VAE	0.9645	(0.9525, 0.9764)	0.2609	(0.2358, 0.2860)
NK	CVAE	0.8798	(0.8682, 0.8914)	0.1095	(0.0975, 0.1215)
NK	VFAE	0.1548	(0.1258, 0.1838)	0.0093	(0.0072, 0.0114)
NK	trVAE	0.1013	(0.0514, 0.1512)	0.0113	(0.0067, 0.0159)
NK	CoMP	0.0721	(0.0488, 0.0953)	-0.0025	(-0.0035, -0.0015)
T	VAE	0.7172	(0.6377, 0.7967)	0.2423	(0.2135, 0.2711)
T	CVAE	0.3891	(0.3315, 0.4466)	0.0561	(0.0459, 0.0663)
T	VFAE	0.1155	(0.0934, 0.1376)	0.0082	(0.0060, 0.0104)
T	trVAE	0.0585	(0.0354, 0.0815)	0.0061	(0.0047, 0.0075)
T	CoMP	0.0009	(0.0003, 0.0016)	-0.0062	(-0.0074, -0.0050)

Table 6: Effect of varying  $\gamma$  for the CoMP model on the kBET metrics for the single-cell PBMC dataset. Here, kBET and m-kBET refer to the mean kBET and mean m-kBET across 10 random seeds for each value of  $\gamma$ , respectively. SEM represents the standard error of the mean. Here,  $k = 100$  and  $\alpha = 0.1$ .

Model	$\gamma$	kBET $_{k,\alpha}$	kBET $_{k,\alpha} \pm \text{SEM}$	m-kBET $_{k,\alpha}$	m-kBET $_{k,\alpha} \pm \text{SEM}$
CoMP	0.25	0.2703	(0.2482, 0.2924)	0.1276	(0.1085, 0.1467)
CoMP	0.50	0.1741	(0.1438, 0.2045)	0.0763	(0.0543, 0.0983)
CoMP	1.00	0.1211	(0.0845, 0.1577)	0.0681	(0.0388, 0.0975)
CoMP	5.00	0.4426	(0.3889, 0.4963)	0.4419	(0.3827, 0.5011)
CoMP	10.00	0.5311	(0.4456, 0.6167)	0.5118	(0.4135, 0.6100)
CoMP	15.00	0.4288	(0.3511, 0.5065)	0.4614	(0.3738, 0.5489)
CoMP	20.00	0.5880	(0.5101, 0.6660)	0.6383	(0.5547, 0.7219)

Table 7: Effect of varying  $\gamma$  for the CoMP model on the Silhouette Coefficient metrics for the single-cell PBMC dataset. Here,  $s$  and  $\tilde{s}$  refer to the mean  $s$  and mean  $\tilde{s}$  across 10 random seeds for each value of  $\gamma$ , respectively. SEM represents the standard error of the mean. Here,  $k = 100$ .

Model	$\gamma$	$s_k$	$s_k \pm \text{SEM}$	$\tilde{s}_k$	$\tilde{s}_k \pm \text{SEM}$
CoMP	0.25	-0.0024	(-0.0028, -0.0021)	-0.0006	(-0.0018, 0.0006)
CoMP	0.50	-0.0029	(-0.0030, -0.0028)	-0.0023	(-0.0031, -0.0015)
CoMP	1.00	-0.0026	(-0.0032, -0.0020)	-0.0013	(-0.0027, 0.0001)
CoMP	5.00	0.0043	(0.0026, 0.0059)	0.0209	(0.0145, 0.0274)
CoMP	10.00	0.0028	(0.0016, 0.0039)	0.0523	(0.0300, 0.0746)
CoMP	15.00	0.0046	(0.0025, 0.0067)	0.0750	(0.0484, 0.1016)
CoMP	20.00	0.0061	(0.0038, 0.0083)	0.1319	(0.1053, 0.1584)

## 663 D.5 Counterfactual prediction of single-cell PBMC expression data (IFN- $\beta$ stimulation)

664 In this section we present the full results on the counterfactual prediction of single-cell PBMC  
665 expression data under IFN- $\beta$  stimulation. In Tables 9 and 10, we present the mean and standard  
666 error of the Pearson correlation coefficient and MSE metrics respectively for CoMP and the other  
667 four VAE models. We present our results for each cell type separately. As is consistent with the  
668 summary presented in Figures 5 and 8, we see that CoMP produces highly accurate counterfactual  
669 reconstructions. Indeed, this can be seen in the scatter plots showing the mean expression of (actual)  
670 stimulated cells against the mean of counterfactually stimulated control cells (Figure 9). Here, we see  
671 that the other VAE models tend to underestimate the expression of genes that are up-regulated by  
672 IFN- $\beta$  stimulation and overestimate the expression of genes that are down-regulated. However, this  
673 is not as evident with CoMP.

674 Similar to the mixing metrics, we evaluate the effect of varying the penalty scale  $\gamma$ . As we see in  
675 Tables 12 and 11 the optimal value is  $\sim 1$ .

## 676 D.6 Implementation details and hyperparameters

677 The encoder and decoders are parameterised by multi-layer fully-connected networks. Following  
678 the trVAE implementation [10], we implement a multi-scale Gaussian kernel for both trVAE and  
679 VFAE benchmark models, except on the Adult Income dataset where a single scale kernel was used  
680 to match the original implementation. The details of the model architectures and hyperparameters  
681 used in CoMP, VFAE and trVAE across three sets of experiments are given in Tables 13-21.

## 682 D.7 Model training resources

683 Experiments were performed on NVIDIA Tesla V100 GPUs. Each training run of CoMP for a single  
684 hyperparameter configuration on the Tumour / Cell Line dataset (our largest dataset) on a single GPU  
685 takes 2-3 hours. Running times for the other models are broadly similar.

Table 8: Effect of varying  $\gamma$  for cell type specific kBET and  $s$  metrics for the single-cell PBMC dataset. Metrics represent the mean value across the 10 random seeds. Here,  $k = 100$  and  $\alpha = 0.1$ .

Cell type	Model	$\gamma$	kBET $_{k,\alpha}$	kBET $_{k,\alpha} \pm \text{SEM}$	$s_k$	$s_k \pm \text{SEM}$
B	CoMP	0.25	0.0061	(0.0037, 0.0086)	-0.0046	(-0.0053, -0.0038)
B	CoMP	0.50	0.0030	(0.0004, 0.0056)	-0.0033	(-0.0039, -0.0028)
B	CoMP	1.00	0.0110	(0.0002, 0.0217)	-0.0040	(-0.0050, -0.0030)
B	CoMP	5.00	0.4860	(0.3881, 0.5840)	0.0094	(0.0073, 0.0115)
B	CoMP	10.00	0.5321	(0.4226, 0.6415)	0.0195	(0.0118, 0.0273)
B	CoMP	15.00	0.5229	(0.4331, 0.6127)	0.0167	(0.0109, 0.0224)
B	CoMP	20.00	0.6135	(0.5308, 0.6963)	0.0729	(0.0384, 0.1075)
CD14 Mono	CoMP	0.25	0.6868	(0.6336, 0.7400)	0.0060	(-0.0001, 0.0121)
CD14 Mono	CoMP	0.50	0.3840	(0.3001, 0.4680)	-0.0007	(-0.0017, 0.0003)
CD14 Mono	CoMP	1.00	0.1709	(0.0817, 0.2601)	0.0003	(-0.0015, 0.0022)
CD14 Mono	CoMP	5.00	0.4530	(0.3295, 0.5765)	0.0254	(0.0092, 0.0416)
CD14 Mono	CoMP	10.00	0.7073	(0.6114, 0.8031)	0.0745	(0.0419, 0.1072)
CD14 Mono	CoMP	15.00	0.6991	(0.5946, 0.8036)	0.1240	(0.0795, 0.1685)
CD14 Mono	CoMP	20.00	0.7889	(0.6943, 0.8834)	0.1429	(0.1012, 0.1846)
CD16 Mono	CoMP	0.25	0.0963	(0.0382, 0.1543)	0.0031	(0.0018, 0.0045)
CD16 Mono	CoMP	0.50	0.0589	(0.0216, 0.0962)	0.0003	(-0.0008, 0.0013)
CD16 Mono	CoMP	1.00	0.0947	(0.0184, 0.1711)	0.0005	(-0.0011, 0.0021)
CD16 Mono	CoMP	5.00	0.5897	(0.4934, 0.6859)	0.0421	(0.0248, 0.0593)
CD16 Mono	CoMP	10.00	0.6113	(0.4660, 0.7566)	0.1741	(0.0946, 0.2537)
CD16 Mono	CoMP	15.00	0.5950	(0.4552, 0.7348)	0.2778	(0.1705, 0.3850)
CD16 Mono	CoMP	20.00	0.8420	(0.7381, 0.9460)	0.4705	(0.3782, 0.5628)
CD4 T	CoMP	0.25	0.0642	(0.0543, 0.0742)	-0.0027	(-0.0032, -0.0022)
CD4 T	CoMP	0.50	0.0401	(0.0344, 0.0458)	-0.0027	(-0.0031, -0.0023)
CD4 T	CoMP	1.00	0.0538	(0.0442, 0.0634)	-0.0022	(-0.0028, -0.0016)
CD4 T	CoMP	5.00	0.3631	(0.2966, 0.4296)	0.0036	(0.0024, 0.0048)
CD4 T	CoMP	10.00	0.4058	(0.3020, 0.5096)	0.0075	(0.0039, 0.0111)
CD4 T	CoMP	15.00	0.3287	(0.2466, 0.4108)	0.0080	(0.0043, 0.0117)
CD4 T	CoMP	20.00	0.5127	(0.4140, 0.6115)	0.0561	(0.0192, 0.0929)
CD8 T	CoMP	0.25	0.0081	(0.0051, 0.0111)	-0.0015	(-0.0026, -0.0004)
CD8 T	CoMP	0.50	0.0216	(0.0129, 0.0304)	-0.0021	(-0.0032, -0.0010)
CD8 T	CoMP	1.00	0.0634	(0.0437, 0.0830)	-0.0000	(-0.0008, 0.0008)
CD8 T	CoMP	5.00	0.4287	(0.3508, 0.5067)	0.0266	(0.0167, 0.0365)
CD8 T	CoMP	10.00	0.4289	(0.3251, 0.5326)	0.0115	(0.0073, 0.0156)
CD8 T	CoMP	15.00	0.2733	(0.1927, 0.3540)	0.0071	(0.0051, 0.0090)
CD8 T	CoMP	20.00	0.4913	(0.3731, 0.6095)	0.0495	(0.0127, 0.0863)
DC	CoMP	0.25	0.1379	(0.0972, 0.1786)	0.0056	(0.0026, 0.0085)
DC	CoMP	0.50	0.0739	(0.0394, 0.1085)	0.0009	(-0.0029, 0.0047)
DC	CoMP	1.00	0.0784	(0.0329, 0.1239)	0.0034	(-0.0027, 0.0096)
DC	CoMP	5.00	0.2339	(0.1546, 0.3132)	0.0461	(0.0230, 0.0693)
DC	CoMP	10.00	0.4642	(0.3143, 0.6141)	0.1064	(0.0433, 0.1695)
DC	CoMP	15.00	0.6008	(0.4650, 0.7367)	0.1502	(0.0832, 0.2172)
DC	CoMP	20.00	0.7962	(0.6907, 0.9017)	0.1718	(0.1117, 0.2319)
NK	CoMP	0.25	0.0158	(0.0075, 0.0241)	-0.0043	(-0.0049, -0.0036)
NK	CoMP	0.50	0.0233	(0.0045, 0.0420)	-0.0048	(-0.0056, -0.0041)
NK	CoMP	1.00	0.0721	(0.0488, 0.0953)	-0.0025	(-0.0035, -0.0015)
NK	CoMP	5.00	0.6378	(0.5259, 0.7497)	0.0058	(0.0018, 0.0097)
NK	CoMP	10.00	0.5422	(0.4403, 0.6440)	0.0135	(0.0094, 0.0177)
NK	CoMP	15.00	0.4105	(0.3194, 0.5016)	0.0134	(0.0083, 0.0185)
NK	CoMP	20.00	0.5637	(0.4606, 0.6667)	0.0372	(0.0152, 0.0592)
T	CoMP	0.25	0.0052	(0.0025, 0.0079)	-0.0066	(-0.0076, -0.0057)
T	CoMP	0.50	0.0054	(0.0021, 0.0086)	-0.0059	(-0.0066, -0.0052)
T	CoMP	1.00	0.0009	(0.0003, 0.0016)	-0.0062	(-0.0074, -0.0050)
T	CoMP	5.00	0.3430	(0.2440, 0.4420)	0.0087	(0.0041, 0.0134)
T	CoMP	10.00	0.4024	(0.3032, 0.5015)	0.0113	(0.0067, 0.0159)
T	CoMP	15.00	0.2605	(0.1846, 0.3364)	0.0030	(0.0005, 0.0055)
T	CoMP	20.00	0.4979	(0.3883, 0.6076)	0.0542	(0.0224, 0.0860)

Table 9: Counterfactual reconstruction by cell type: Pearson correlation coefficient metrics for all genes ( $r_{all}$ ) and the top 50 DE genes ( $r_{DE}$ ). Metrics represent the mean across 10 random seeds for each model. SEM represents standard error of the mean.

Cell type	Model	$r_{all}$	$r_{all} \pm \text{SEM}$	$r_{DE}$	$r_{DE} \pm \text{SEM}$
B	VAE	0.8854	(0.8850, 0.8857)	0.8170	(0.8165, 0.8175)
B	CVAE	0.9499	(0.9481, 0.9516)	0.9153	(0.9125, 0.9181)
B	VFAE	0.9908	(0.9901, 0.9915)	0.9880	(0.9866, 0.9893)
B	trVAE	0.9877	(0.9868, 0.9886)	0.9833	(0.9817, 0.9849)
B	CoMP	0.9986	(0.9984, 0.9988)	0.9985	(0.9982, 0.9987)
CD14 Mono	VAE	0.7488	(0.7485, 0.7491)	0.4896	(0.4891, 0.4900)
CD14 Mono	CVAE	0.7529	(0.7520, 0.7538)	0.4958	(0.4938, 0.4977)
CD14 Mono	VFAE	0.9954	(0.9951, 0.9958)	0.9928	(0.9921, 0.9935)
CD14 Mono	trVAE	0.9830	(0.9804, 0.9856)	0.9650	(0.9586, 0.9714)
CD14 Mono	CoMP	0.9954	(0.9915, 0.9992)	0.9920	(0.9848, 0.9993)
CD16 Mono	VAE	0.8304	(0.8301, 0.8307)	0.7135	(0.7131, 0.7140)
CD16 Mono	CVAE	0.8351	(0.8341, 0.8360)	0.7223	(0.7203, 0.7243)
CD16 Mono	VFAE	0.9912	(0.9909, 0.9915)	0.9910	(0.9904, 0.9916)
CD16 Mono	trVAE	0.9881	(0.9873, 0.9889)	0.9821	(0.9802, 0.9839)
CD16 Mono	CoMP	0.9990	(0.9985, 0.9994)	0.9989	(0.9986, 0.9993)
CD4 T	VAE	0.8975	(0.8971, 0.8978)	0.8366	(0.8360, 0.8372)
CD4 T	CVAE	0.9697	(0.9682, 0.9712)	0.9514	(0.9492, 0.9537)
CD4 T	VFAE	0.9977	(0.9975, 0.9979)	0.9983	(0.9982, 0.9985)
CD4 T	trVAE	0.9915	(0.9908, 0.9922)	0.9905	(0.9893, 0.9918)
CD4 T	CoMP	0.9990	(0.9990, 0.9991)	0.9988	(0.9987, 0.9989)
CD8 T	VAE	0.9108	(0.9104, 0.9112)	0.8719	(0.8713, 0.8724)
CD8 T	CVAE	0.9726	(0.9715, 0.9736)	0.9613	(0.9598, 0.9628)
CD8 T	VFAE	0.9923	(0.9920, 0.9927)	0.9935	(0.9931, 0.9939)
CD8 T	trVAE	0.9828	(0.9810, 0.9846)	0.9808	(0.9781, 0.9836)
CD8 T	CoMP	0.9927	(0.9917, 0.9937)	0.9950	(0.9945, 0.9955)
DC	VAE	0.8156	(0.8153, 0.8159)	0.5809	(0.5802, 0.5816)
DC	CVAE	0.8213	(0.8205, 0.8221)	0.5943	(0.5925, 0.5961)
DC	VFAE	0.9885	(0.9879, 0.9892)	0.9894	(0.9887, 0.9901)
DC	trVAE	0.9743	(0.9702, 0.9783)	0.9502	(0.9396, 0.9608)
DC	CoMP	0.9946	(0.9925, 0.9966)	0.9931	(0.9899, 0.9962)
NK	VAE	0.8918	(0.8910, 0.8926)	0.8304	(0.8292, 0.8316)
NK	CVAE	0.9539	(0.9520, 0.9558)	0.9269	(0.9237, 0.9301)
NK	VFAE	0.9870	(0.9865, 0.9874)	0.9864	(0.9855, 0.9873)
NK	trVAE	0.9393	(0.9259, 0.9526)	0.9290	(0.9113, 0.9466)
NK	CoMP	0.9917	(0.9904, 0.9929)	0.9899	(0.9881, 0.9917)
T	VAE	0.8848	(0.8843, 0.8853)	0.7469	(0.7457, 0.7480)
T	CVAE	0.9516	(0.9500, 0.9533)	0.8960	(0.8926, 0.8994)
T	VFAE	0.9849	(0.9841, 0.9856)	0.9763	(0.9750, 0.9777)
T	trVAE	0.9567	(0.9498, 0.9637)	0.9368	(0.9294, 0.9443)
T	CoMP	0.9941	(0.9936, 0.9946)	0.9934	(0.9928, 0.9940)

Table 10: Counterfactual reconstruction by cell type: Mean squared error metrics for all genes ( $MSE_{all}$ ) and the top 50 DE genes ( $MSE_{DE}$ ). Metrics represent the mean across 10 random seeds for each model. SEM represents standard error of the mean.

Cell type	Model	$MSE_{all}$	$MSE_{all} \pm SEM$	$MSE_{DE}$	$MSE_{DE} \pm SEM$
B	VAE	0.0085	(0.0085, 0.0085)	0.3230	(0.3221, 0.3239)
B	CVAE	0.0038	(0.0037, 0.0039)	0.1398	(0.1347, 0.1448)
B	VFAE	0.0008	(0.0007, 0.0008)	0.0199	(0.0178, 0.0220)
B	trVAE	0.0010	(0.0009, 0.0010)	0.0276	(0.0250, 0.0301)
B	CoMP	0.0001	(0.0001, 0.0001)	0.0024	(0.0020, 0.0028)
CD14 Mono	VAE	0.0483	(0.0483, 0.0484)	1.7942	(1.7923, 1.7962)
CD14 Mono	CVAE	0.0476	(0.0475, 0.0478)	1.7624	(1.7563, 1.7684)
CD14 Mono	VFAE	0.0014	(0.0013, 0.0015)	0.0343	(0.0314, 0.0371)
CD14 Mono	trVAE	0.0044	(0.0038, 0.0051)	0.1422	(0.1190, 0.1654)
CD14 Mono	CoMP	0.0011	(0.0002, 0.0019)	0.0245	(0.0023, 0.0468)
CD16 Mono	VAE	0.0301	(0.0301, 0.0302)	1.1255	(1.1234, 1.1276)
CD16 Mono	CVAE	0.0294	(0.0293, 0.0295)	1.0933	(1.0878, 1.0989)
CD16 Mono	VFAE	0.0017	(0.0017, 0.0018)	0.0223	(0.0204, 0.0242)
CD16 Mono	trVAE	0.0029	(0.0027, 0.0031)	0.0861	(0.0790, 0.0932)
CD16 Mono	CoMP	0.0002	(0.0001, 0.0003)	0.0031	(0.0017, 0.0044)
CD4 T	VAE	0.0060	(0.0059, 0.0060)	0.2274	(0.2266, 0.2282)
CD4 T	CVAE	0.0018	(0.0017, 0.0019)	0.0677	(0.0644, 0.0709)
CD4 T	VFAE	0.0001	(0.0001, 0.0002)	0.0021	(0.0018, 0.0023)
CD4 T	trVAE	0.0005	(0.0005, 0.0006)	0.0126	(0.0107, 0.0144)
CD4 T	CoMP	0.0001	(0.0001, 0.0001)	0.0015	(0.0014, 0.0016)
CD8 T	VAE	0.0058	(0.0058, 0.0058)	0.2187	(0.2178, 0.2196)
CD8 T	CVAE	0.0019	(0.0019, 0.0020)	0.0684	(0.0659, 0.0709)
CD8 T	VFAE	0.0005	(0.0005, 0.0006)	0.0098	(0.0093, 0.0103)
CD8 T	trVAE	0.0012	(0.0011, 0.0013)	0.0332	(0.0284, 0.0381)
CD8 T	CoMP	0.0005	(0.0004, 0.0006)	0.0074	(0.0066, 0.0082)
DC	VAE	0.0332	(0.0331, 0.0332)	1.2308	(1.2292, 1.2324)
DC	CVAE	0.0322	(0.0321, 0.0324)	1.1887	(1.1834, 1.1939)
DC	VFAE	0.0024	(0.0023, 0.0025)	0.0303	(0.0287, 0.0318)
DC	trVAE	0.0056	(0.0048, 0.0064)	0.1758	(0.1436, 0.2081)
DC	CoMP	0.0011	(0.0007, 0.0016)	0.0161	(0.0083, 0.0239)
NK	VAE	0.0091	(0.0091, 0.0092)	0.3395	(0.3370, 0.3420)
NK	CVAE	0.0043	(0.0041, 0.0044)	0.1535	(0.1477, 0.1593)
NK	VFAE	0.0014	(0.0013, 0.0014)	0.0345	(0.0327, 0.0362)
NK	trVAE	0.0053	(0.0042, 0.0064)	0.1455	(0.1100, 0.1810)
NK	CoMP	0.0008	(0.0007, 0.0010)	0.0204	(0.0169, 0.0238)
T	VAE	0.0077	(0.0077, 0.0077)	0.2799	(0.2786, 0.2811)
T	CVAE	0.0033	(0.0032, 0.0034)	0.1126	(0.1088, 0.1164)
T	VFAE	0.0011	(0.0010, 0.0011)	0.0237	(0.0223, 0.0252)
T	trVAE	0.0030	(0.0025, 0.0034)	0.0674	(0.0583, 0.0764)
T	CoMP	0.0004	(0.0004, 0.0005)	0.0066	(0.0060, 0.0073)

Table 11: Effect of  $\gamma$  on counterfactual data reconstruction: Mean Pearson correlation coefficient for all and DE genes across 10 random seeds.

Cell type	Model	$\gamma$	$r_{\text{all}}$	$r_{\text{all}} \pm \text{SEM}$	$r_{\text{DE}}$	$r_{\text{DE}} \pm \text{SEM}$
B	CoMP	0.25	0.9987	(0.9986, 0.9988)	0.9986	(0.9984, 0.9987)
B	CoMP	0.50	0.9987	(0.9985, 0.9988)	0.9985	(0.9983, 0.9988)
B	CoMP	1.00	0.9986	(0.9984, 0.9988)	0.9985	(0.9982, 0.9987)
B	CoMP	5.00	0.9577	(0.9432, 0.9722)	0.9623	(0.9510, 0.9736)
B	CoMP	10.00	0.9233	(0.9023, 0.9443)	0.9397	(0.9239, 0.9555)
B	CoMP	15.00	0.9106	(0.8925, 0.9287)	0.9299	(0.9163, 0.9435)
B	CoMP	20.00	0.8974	(0.8841, 0.9107)	0.9080	(0.8966, 0.9195)
CD14 Mono	CoMP	0.25	0.9906	(0.9866, 0.9945)	0.9828	(0.9746, 0.9909)
CD14 Mono	CoMP	0.50	0.9948	(0.9917, 0.9980)	0.9910	(0.9844, 0.9977)
CD14 Mono	CoMP	1.00	0.9954	(0.9915, 0.9992)	0.9920	(0.9848, 0.9993)
CD14 Mono	CoMP	5.00	0.9892	(0.9831, 0.9954)	0.9823	(0.9723, 0.9922)
CD14 Mono	CoMP	10.00	0.9536	(0.9316, 0.9757)	0.9439	(0.9217, 0.9662)
CD14 Mono	CoMP	15.00	0.9224	(0.8933, 0.9515)	0.9174	(0.8886, 0.9463)
CD14 Mono	CoMP	20.00	0.9034	(0.8737, 0.9332)	0.8966	(0.8673, 0.9258)
CD16 Mono	CoMP	0.25	0.9983	(0.9977, 0.9989)	0.9982	(0.9976, 0.9988)
CD16 Mono	CoMP	0.50	0.9989	(0.9985, 0.9992)	0.9988	(0.9985, 0.9991)
CD16 Mono	CoMP	1.00	0.9990	(0.9985, 0.9994)	0.9989	(0.9986, 0.9993)
CD16 Mono	CoMP	5.00	0.9847	(0.9805, 0.9890)	0.9801	(0.9753, 0.9850)
CD16 Mono	CoMP	10.00	0.9420	(0.9168, 0.9672)	0.9503	(0.9313, 0.9693)
CD16 Mono	CoMP	15.00	0.9017	(0.8702, 0.9332)	0.9222	(0.8999, 0.9444)
CD16 Mono	CoMP	20.00	0.8563	(0.8289, 0.8838)	0.8850	(0.8668, 0.9031)
CD4 T	CoMP	0.25	0.9989	(0.9989, 0.9990)	0.9987	(0.9986, 0.9988)
CD4 T	CoMP	0.50	0.9991	(0.9990, 0.9991)	0.9989	(0.9988, 0.9990)
CD4 T	CoMP	1.00	0.9990	(0.9990, 0.9991)	0.9988	(0.9987, 0.9989)
CD4 T	CoMP	5.00	0.9925	(0.9899, 0.9951)	0.9901	(0.9863, 0.9939)
CD4 T	CoMP	10.00	0.9948	(0.9933, 0.9962)	0.9970	(0.9954, 0.9985)
CD4 T	CoMP	15.00	0.9944	(0.9931, 0.9958)	0.9979	(0.9975, 0.9983)
CD4 T	CoMP	20.00	0.9782	(0.9689, 0.9875)	0.9738	(0.9584, 0.9892)
CD8 T	CoMP	0.25	0.9963	(0.9961, 0.9964)	0.9965	(0.9964, 0.9966)
CD8 T	CoMP	0.50	0.9955	(0.9951, 0.9960)	0.9962	(0.9959, 0.9965)
CD8 T	CoMP	1.00	0.9927	(0.9917, 0.9937)	0.9950	(0.9945, 0.9955)
CD8 T	CoMP	5.00	0.9666	(0.9626, 0.9705)	0.9790	(0.9765, 0.9814)
CD8 T	CoMP	10.00	0.9559	(0.9499, 0.9620)	0.9757	(0.9727, 0.9787)
CD8 T	CoMP	15.00	0.9528	(0.9468, 0.9589)	0.9745	(0.9715, 0.9774)
CD8 T	CoMP	20.00	0.9455	(0.9397, 0.9512)	0.9605	(0.9516, 0.9694)
DC	CoMP	0.25	0.9959	(0.9955, 0.9962)	0.9945	(0.9942, 0.9949)
DC	CoMP	0.50	0.9966	(0.9963, 0.9970)	0.9956	(0.9949, 0.9962)
DC	CoMP	1.00	0.9946	(0.9925, 0.9966)	0.9931	(0.9899, 0.9962)
DC	CoMP	5.00	0.9694	(0.9576, 0.9811)	0.9671	(0.9528, 0.9814)
DC	CoMP	10.00	0.9219	(0.8966, 0.9472)	0.9265	(0.9031, 0.9499)
DC	CoMP	15.00	0.8867	(0.8549, 0.9184)	0.8955	(0.8686, 0.9224)
DC	CoMP	20.00	0.8547	(0.8288, 0.8806)	0.8676	(0.8428, 0.8924)
NK	CoMP	0.25	0.9962	(0.9959, 0.9964)	0.9955	(0.9951, 0.9959)
NK	CoMP	0.50	0.9949	(0.9942, 0.9957)	0.9938	(0.9927, 0.9950)
NK	CoMP	1.00	0.9917	(0.9904, 0.9929)	0.9899	(0.9881, 0.9917)
NK	CoMP	5.00	0.9567	(0.9491, 0.9643)	0.9399	(0.9296, 0.9501)
NK	CoMP	10.00	0.8916	(0.8654, 0.9178)	0.8670	(0.8361, 0.8979)
NK	CoMP	15.00	0.8780	(0.8501, 0.9060)	0.8518	(0.8187, 0.8850)
NK	CoMP	20.00	0.8767	(0.8517, 0.9018)	0.8477	(0.8189, 0.8765)
T	CoMP	0.25	0.9951	(0.9947, 0.9954)	0.9945	(0.9940, 0.9950)
T	CoMP	0.50	0.9950	(0.9947, 0.9952)	0.9945	(0.9940, 0.9949)
T	CoMP	1.00	0.9941	(0.9936, 0.9946)	0.9934	(0.9928, 0.9940)
T	CoMP	5.00	0.9682	(0.9584, 0.9779)	0.9690	(0.9627, 0.9753)
T	CoMP	10.00	0.9462	(0.9336, 0.9588)	0.9595	(0.9519, 0.9672)
T	CoMP	15.00	0.9402	(0.9277, 0.9527)	0.9569	(0.9495, 0.9642)
T	CoMP	20.00	0.9239	(0.9136, 0.9342)	0.9254	(0.9092, 0.9416)

Table 12: Effect of  $\gamma$  on counterfactual data reconstruction: Mean of the mean squared error (MSE) for all and DE genes across 10 random seeds.

Cell type	Model	$\gamma$	MSE <sub>all</sub>	MSE <sub>all</sub> $\pm$ SEM	MSE <sub>DE</sub>	MSE <sub>DE</sub> $\pm$ SEM
B	CoMP	0.25	0.0001	(0.0001, 0.0001)	0.0023	(0.0021, 0.0025)
B	CoMP	0.50	0.0001	(0.0001, 0.0001)	0.0023	(0.0019, 0.0026)
B	CoMP	1.00	0.0001	(0.0001, 0.0001)	0.0024	(0.0020, 0.0028)
B	CoMP	5.00	0.0032	(0.0022, 0.0043)	0.0672	(0.0479, 0.0864)
B	CoMP	10.00	0.0056	(0.0041, 0.0071)	0.1046	(0.0768, 0.1325)
B	CoMP	15.00	0.0065	(0.0053, 0.0078)	0.1201	(0.0960, 0.1442)
B	CoMP	20.00	0.0077	(0.0067, 0.0086)	0.1666	(0.1442, 0.1891)
CD14 Mono	CoMP	0.25	0.0023	(0.0014, 0.0032)	0.0576	(0.0309, 0.0843)
CD14 Mono	CoMP	0.50	0.0012	(0.0005, 0.0020)	0.0276	(0.0068, 0.0483)
CD14 Mono	CoMP	1.00	0.0011	(0.0002, 0.0019)	0.0245	(0.0023, 0.0468)
CD14 Mono	CoMP	5.00	0.0034	(0.0013, 0.0056)	0.0935	(0.0330, 0.1541)
CD14 Mono	CoMP	10.00	0.0123	(0.0066, 0.0180)	0.3157	(0.1724, 0.4591)
CD14 Mono	CoMP	15.00	0.0196	(0.0124, 0.0268)	0.4825	(0.3056, 0.6594)
CD14 Mono	CoMP	20.00	0.0245	(0.0173, 0.0316)	0.6022	(0.4302, 0.7742)
CD16 Mono	CoMP	0.25	0.0003	(0.0002, 0.0005)	0.0054	(0.0034, 0.0074)
CD16 Mono	CoMP	0.50	0.0002	(0.0001, 0.0003)	0.0036	(0.0024, 0.0048)
CD16 Mono	CoMP	1.00	0.0002	(0.0001, 0.0003)	0.0031	(0.0017, 0.0044)
CD16 Mono	CoMP	5.00	0.0040	(0.0024, 0.0056)	0.0876	(0.0451, 0.1300)
CD16 Mono	CoMP	10.00	0.0120	(0.0070, 0.0170)	0.2751	(0.1522, 0.3979)
CD16 Mono	CoMP	15.00	0.0194	(0.0134, 0.0254)	0.4529	(0.3058, 0.6001)
CD16 Mono	CoMP	20.00	0.0284	(0.0233, 0.0335)	0.6688	(0.5421, 0.7955)
CD4 T	CoMP	0.25	0.0001	(0.0001, 0.0001)	0.0015	(0.0014, 0.0017)
CD4 T	CoMP	0.50	0.0001	(0.0001, 0.0001)	0.0013	(0.0013, 0.0014)
CD4 T	CoMP	1.00	0.0001	(0.0001, 0.0001)	0.0015	(0.0014, 0.0016)
CD4 T	CoMP	5.00	0.0005	(0.0003, 0.0006)	0.0133	(0.0081, 0.0185)
CD4 T	CoMP	10.00	0.0003	(0.0002, 0.0004)	0.0040	(0.0019, 0.0062)
CD4 T	CoMP	15.00	0.0003	(0.0003, 0.0004)	0.0027	(0.0022, 0.0032)
CD4 T	CoMP	20.00	0.0013	(0.0008, 0.0019)	0.0381	(0.0155, 0.0607)
CD8 T	CoMP	0.25	0.0003	(0.0003, 0.0003)	0.0049	(0.0046, 0.0051)
CD8 T	CoMP	0.50	0.0003	(0.0003, 0.0003)	0.0052	(0.0048, 0.0056)
CD8 T	CoMP	1.00	0.0005	(0.0004, 0.0006)	0.0074	(0.0066, 0.0082)
CD8 T	CoMP	5.00	0.0023	(0.0020, 0.0025)	0.0329	(0.0290, 0.0369)
CD8 T	CoMP	10.00	0.0030	(0.0026, 0.0033)	0.0366	(0.0322, 0.0411)
CD8 T	CoMP	15.00	0.0032	(0.0028, 0.0035)	0.0387	(0.0344, 0.0431)
CD8 T	CoMP	20.00	0.0038	(0.0034, 0.0042)	0.0717	(0.0512, 0.0921)
DC	CoMP	0.25	0.0009	(0.0008, 0.0009)	0.0135	(0.0122, 0.0149)
DC	CoMP	0.50	0.0007	(0.0006, 0.0008)	0.0102	(0.0082, 0.0123)
DC	CoMP	1.00	0.0011	(0.0007, 0.0016)	0.0161	(0.0083, 0.0239)
DC	CoMP	5.00	0.0075	(0.0044, 0.0106)	0.1189	(0.0580, 0.1798)
DC	CoMP	10.00	0.0165	(0.0113, 0.0217)	0.2549	(0.1607, 0.3490)
DC	CoMP	15.00	0.0228	(0.0169, 0.0288)	0.3663	(0.2603, 0.4723)
DC	CoMP	20.00	0.0299	(0.0250, 0.0347)	0.4809	(0.3844, 0.5773)
NK	CoMP	0.25	0.0004	(0.0004, 0.0004)	0.0089	(0.0080, 0.0097)
NK	CoMP	0.50	0.0005	(0.0004, 0.0006)	0.0122	(0.0099, 0.0146)
NK	CoMP	1.00	0.0008	(0.0007, 0.0010)	0.0204	(0.0169, 0.0238)
NK	CoMP	5.00	0.0041	(0.0035, 0.0048)	0.1169	(0.0988, 0.1350)
NK	CoMP	10.00	0.0090	(0.0069, 0.0110)	0.2419	(0.1869, 0.2969)
NK	CoMP	15.00	0.0100	(0.0078, 0.0122)	0.2687	(0.2101, 0.3273)
NK	CoMP	20.00	0.0103	(0.0084, 0.0122)	0.2874	(0.2376, 0.3371)
T	CoMP	0.25	0.0004	(0.0003, 0.0004)	0.0053	(0.0048, 0.0058)
T	CoMP	0.50	0.0004	(0.0003, 0.0004)	0.0053	(0.0049, 0.0058)
T	CoMP	1.00	0.0004	(0.0004, 0.0005)	0.0066	(0.0060, 0.0073)
T	CoMP	5.00	0.0022	(0.0016, 0.0029)	0.0409	(0.0311, 0.0507)
T	CoMP	10.00	0.0037	(0.0029, 0.0046)	0.0579	(0.0454, 0.0703)
T	CoMP	15.00	0.0041	(0.0033, 0.0050)	0.0624	(0.0501, 0.0747)
T	CoMP	20.00	0.0053	(0.0046, 0.0060)	0.1005	(0.0829, 0.1181)

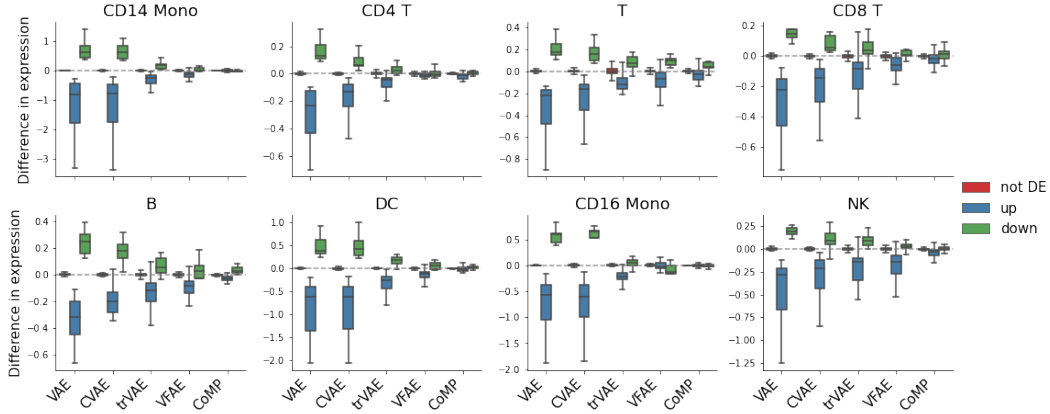


Figure 8: The difference in gene expression values for 1950 non-differentially expressed genes (red) and the top 50 differentially expressed genes (up-regulated: blue, down-regulated: green) between IFN- $\beta$  stimulated cells and counterfactually stimulated control cells for each cell type. The difference in expression for a gene is the gene mean expression across stimulated cells of a cell type minus the mean reconstructed gene expression for counterfactually stimulated control cells of the same cell type.

## D.8 CO<sub>2</sub> emissions

Experiments were conducted using private infrastructure, which has an estimated carbon efficiency of 0.188 kgCO<sub>2</sub>eq/kWh. An estimated cumulative 1900 hours of computation were performed on hardware of type Tesla V100. Total emissions are estimated to be 107 kgCO<sub>2</sub>eq. Estimations were conducted using the Machine Learning Impact calculator presented in [53].

Table 13: CoMP architecture and hyperparameters for the Tumour / Cell Line dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	9468		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	512	[Input, Conditions]	BatchNorm1D, LeakyReLU
FC_2	512	FC_1	BatchNorm1D, LeakyReLU
FC_3	512	FC_2	BatchNorm1D, LeakyReLU
Z_mean	16	FC_3	
Z	16	[Z_mean, 0.1]	Normal()
<b>Decoder</b>			
FC_1	512	Z	BatchNorm1D, LeakyReLU
FC-2	512	FC_1	BatchNorm1D, LeakyReLU
FC-3	512	FC-2	BatchNorm1D, LeakyReLU
$\hat{X}$ _mean	9468	FC-3	
$\hat{X}$ _scale	1	FC-3	
$\hat{X}$	9468	[\hat{X}_mean, \hat{X}_scale]	Normal()
<b>Penalty</b>			
CoMP penalty		[Z, Conditions]	
Optimiser	Adam		
Learning rate	1e-4		
Batch size	5500		
Epochs	4000		
$\beta$	1e-7		
$\gamma$	0.5		
LeakyReLU slope	0.01		

Table 14: VFAE architecture and hyperparameters for the Tumour / Cell Line dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	9468		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	512	[Input, Conditions]	BatchNorm1D, LeakyReLU
FC_2	512	FC_1	BatchNorm1D, LeakyReLU
FC_3	512	FC_2	BatchNorm1D, LeakyReLU
Z_mean	16	FC_3	
Z_scale	16	FC_3	
Z	16	[Z_mean, Z_scale]	Normal()
<b>Decoder</b>			
FC_1	512	Z	BatchNorm1D, LeakyReLU
FC-2	512	FC_1	BatchNorm1D, LeakyReLU
FC-3	512	FC-2	BatchNorm1D, LeakyReLU
$\hat{X}$ _mean	9468	FC-3	
$\hat{X}$ _scale	1	FC-3	
$\hat{X}$	9468	[\hat{X}_mean, \hat{X}_scale]	Normal()
<b>Penalty</b>			
MMD		[FC1, Conditions]	Multi-scale RBF kernel
Optimiser	Adam		
Learning rate	1e-03		
Batch size	5550		
Epochs	4000		
$\beta$	1e-7		
$\gamma$	4		
LeakyReLU slope	0.01		

Table 15: trVAE architecture and hyperparameters for the Tumour / Cell Line dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	9468		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	512	[Input, Conditions]	BatchNorm1D, LeakyReLU
FC_2	512	FC_1	BatchNorm1D, LeakyReLU
FC_3	512	FC_2	BatchNorm1D, LeakyReLU
Z_mean	16	FC_3	
Z_scale	16	FC_3	
Z	16	[Z_mean, Z_scale]	Normal()
<b>Decoder</b>			
FC_1	512	Z	BatchNorm1D, LeakyReLU
FC-2	512	FC_1	BatchNorm1D, LeakyReLU
FC-3	512	FC-2	BatchNorm1D, LeakyReLU
$\hat{X}_{\text{mean}}$	9468	FC-3	
$\hat{X}_{\text{scale}}$	1	FC-3	
$\hat{X}$	9468	$[\hat{X}_{\text{mean}}, \hat{X}_{\text{scale}}]$	Normal()
<b>Penalty</b>			
MMD		[FC1, Conditions]	Multi-scale RBF kernel
Optimiser	Adam		
Learning rate	3e-4		
Batch size	5550		
Epochs	4000		
$\beta$	1e-7		
$\gamma$	10		
LeakyReLU slope	0.01		

Table 16: CoMP architecture and hyperparameters for the single-cell PBMC dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	2000		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	512	[Input, Conditions]	BatchNorm1D, LeakyReLU
FC_2	512	FC_1	BatchNorm1D, LeakyReLU
FC_3	512	FC_2	BatchNorm1D, LeakyReLU
Z_mean	40	FC_3	
Z	40	[Z_mean, 0.1]	Normal()
<b>Decoder</b>			
FC_1	512	Z	BatchNorm1D, LeakyReLU
FC-2	512	FC_1	BatchNorm1D, LeakyReLU
FC-3	512	FC-2	BatchNorm1D, LeakyReLU
$\hat{X}_{\text{mean}}$	2000	FC-3	
$\hat{X}_{\text{scale}}$	1	FC-3	
$\hat{X}$	2000	$[\hat{X}_{\text{mean}}, \hat{X}_{\text{scale}}]$	Normal()
<b>Penalty</b>			
CoMP penalty		[Z, Conditions]	
Optimiser	Adam		
Learning rate	1e-06		
Batch size	512		
Epochs	10000		
$\beta$	1e-7		
$\gamma$	1		
LeakyReLU slope	0.01		

Table 17: VFAE architecture and hyperparameters for the single-cell PBMC dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	2000		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	512	[Input, Conditions]	BatchNorm1D, LeakyReLU
FC_2	512	FC_1	BatchNorm1D, LeakyReLU
FC_3	512	FC_2	BatchNorm1D, LeakyReLU
Z_mean	40	FC_3	
Z	40	[Z_mean, 0.1]	Normal()
<b>Decoder</b>			
FC_1	512	Z	BatchNorm1D, LeakyReLU
FC-2	512	FC_1	BatchNorm1D, LeakyReLU
FC-3	512	FC-2	BatchNorm1D, LeakyReLU
$\hat{X}_{\text{mean}}$	2000	FC-3	
$\hat{X}_{\text{scale}}$	1	FC-3	
$\hat{X}$	2000	$[\hat{X}_{\text{mean}}, \hat{X}_{\text{scale}}]$	Normal()
<b>Penalty</b>			
MMD		[FC1, Conditions]	Multi-scale RBF kernel
Optimiser	Adam		
Learning rate	1e-4		
Batch size	512		
Epochs	10000		
$\beta$	1e-7		
$\gamma$	1		
LeakyReLU slope	0.01		

Table 18: trVAE architecture and hyperparameters for the single-cell PBMC dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	2000		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	512	[Input, Conditions]	BatchNorm1D, LeakyReLU
FC_2	512	FC_1	BatchNorm1D, LeakyReLU
FC_3	512	FC_2	BatchNorm1D, LeakyReLU
Z_mean	40	FC_3	
Z	40	[Z_mean, 0.1]	Normal()
<b>Decoder</b>			
FC_1	512	Z	BatchNorm1D, LeakyReLU
FC-2	512	FC_1	BatchNorm1D, LeakyReLU
FC-3	512	FC-2	BatchNorm1D, LeakyReLU
$\hat{X}_{\text{mean}}$	2000	FC-3	
$\hat{X}_{\text{scale}}$	1	FC-3	
$\hat{X}$	2000	$[\hat{X}_{\text{mean}}, \hat{X}_{\text{scale}}]$	Normal()
<b>Penalty</b>			
MMD		[FC1, Conditions]	Multi-scale RBF kernel
Optimiser	Adam		
Learning rate	5e-4		
Batch size	512		
Epochs	6000		
$\beta$	1e-7		
$\gamma$	10		
LeakyReLU slope	0.01		

Table 19: CoMP architecture and hyperparameters for the UCI Adult Income dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	82		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	64	[Input, Conditions]	BatchNorm1D, LeakyReLU
FC_2	64	FC_1	BatchNorm1D, LeakyReLU
Z_mean	16	FC_2	
Z	16	[Z_mean, 0.1]	Normal()
<b>Decoder</b>			
FC_1	64	Z	BatchNorm1D, LeakyReLU
FC-2	64	FC_1	BatchNorm1D, LeakyReLU
$\hat{X}_{\text{mean}}$	82	FC-2	
$\hat{X}_{\text{scale}}$	1	FC-2	
$\hat{X}$	82	$[\hat{X}_{\text{mean}}, \hat{X}_{\text{scale}}]$	Normal()
<b>Penalty</b>			
CoMP penalty		[Z, Conditions]	
Optimiser	Adam		
Learning rate	1e-04		
Batch size	4096		
Epochs	10000		
$\beta$	1		
$\gamma$	0.5		
LeakyReLU slope	0.01		

Table 20: VFAE architecture and hyperparameters for the UCI Adult Income dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	82		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	64	[Input, Conditions]	BatchNorm1D, LeakyReLU
Z_mean	16	FC_1	
Z_scale	16	FC_1	
Z	16	[Z_mean, Z_scale]	Normal()
<b>Decoder</b>			
FC_1	64	Z	BatchNorm1D, LeakyReLU
$\hat{X}_{\text{mean}}$	82	FC_1	
$\hat{X}_{\text{scale}}$	1	FC_1	
$\hat{X}$	82	$[\hat{X}_{\text{mean}}, \hat{X}_{\text{scale}}]$	Normal()
<b>Penalty</b>			
MMD		[Z, Conditions]	
Optimiser	Adam		
Learning rate	1e-04		
Batch size	512		
Epochs	10000		
$\beta$	1		
$\gamma$	1000		
RBF scale	2		
LeakyReLU slope	0.01		

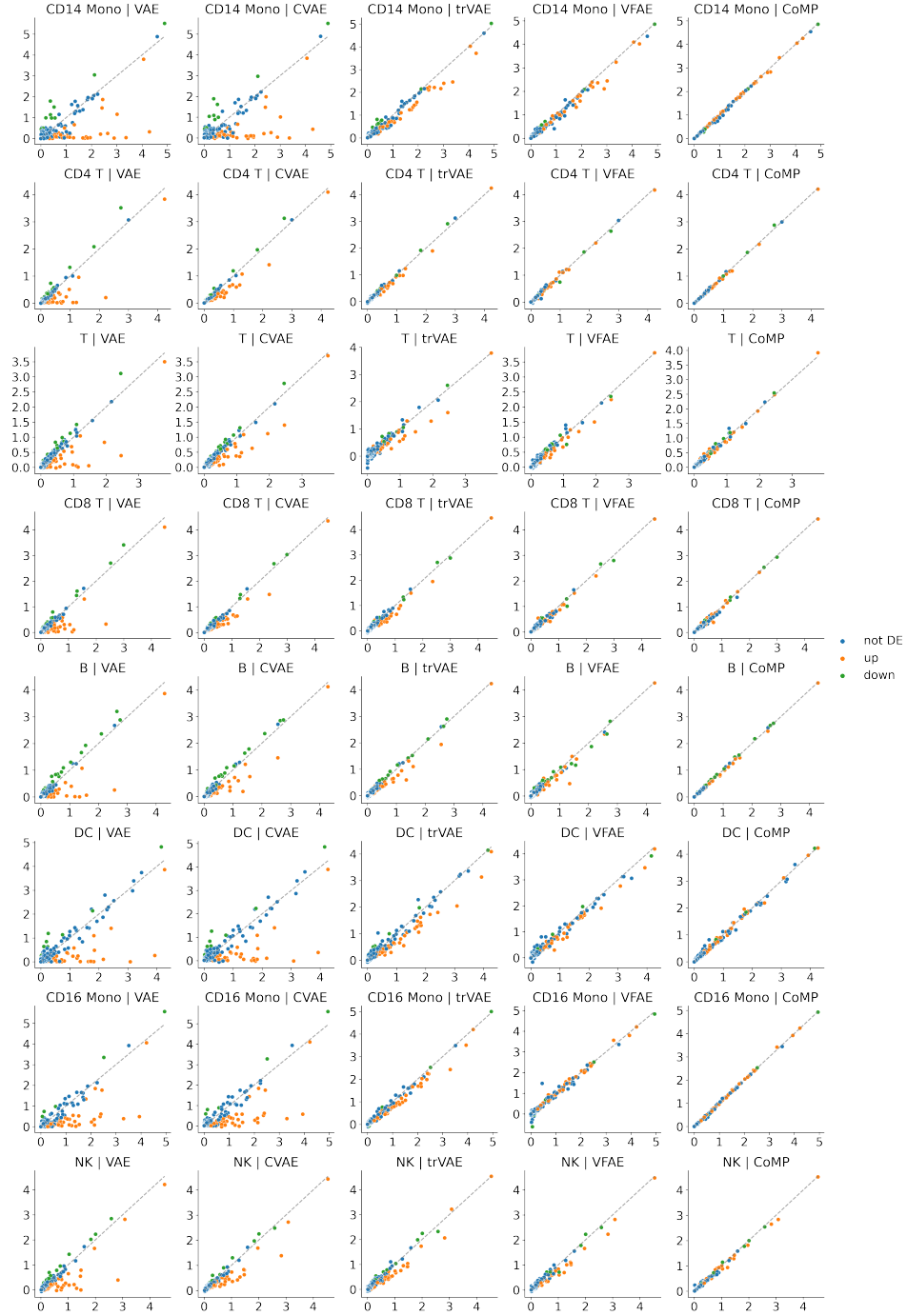


Figure 9: Mean gene expression of actual stimulated cells against the mean gene expression of counterfactually stimulated control cells for each cell type and model.

Table 21: trVAE architecture and hyperparameters for the UCI Adult Income dataset.

Layer	Output Dim	Inputs	Notes
<b>Input</b>	82		
<b>Conditions</b>	2		
<b>Encoder</b>			
FC_1	32	[Input, Conditions]	BatchNorm1D, LeakyReLU
FC_2	32	FC_1	BatchNorm1D, LeakyReLU
Z_mean	8	FC_2	
Z	8	[Z_mean, 0.1]	Normal()
<b>Decoder</b>			
FC_1	32	Z	BatchNorm1D, LeakyReLU
FC-2	32	FC_1	BatchNorm1D, LeakyReLU
$\hat{X}_{\text{mean}}$	82	FC-2	
$\hat{X}_{\text{scale}}$	1	FC-2	
$\hat{X}$	82	$[\hat{X}_{\text{mean}}, \hat{X}_{\text{scale}}]$	Normal()
<b>Penalty</b>			
MMD		[FC1, Conditions]	Multi-scale RBF kernel
Optimiser	Adam		
Learning rate	1e-04		
Batch size	4096		
Epochs	10000		
$\beta$	0.001		
$\gamma$	10		
LeakyReLU slope	0.01		

## References

- [1] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature protocols*, 13(4):599–604, 2018.
- [2] Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, et al. Science forum: the human cell atlas. *Elife*, 6:e27041, 2017.
- [3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [4] Danilo Jimenez Rezende, S. Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.
- [5] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.
- [6] Gregory P Way and Casey S Greene. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium*, pages 80–91. World Scientific, 2018.
- [7] Dongfang Wang and Jin Gu. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics*, 16(5):320–331, 2018.
- [8] Christopher Heje Grønbech, Maximillian Fornitz Vording, Pascal N Timshel, Casper Kaae Sønderby, Tune H Pers, and Ole Winther. scvae: Variational auto-encoders for single-cell gene expression data. *Bioinformatics*, 36(16):4415–4422, 2020.
- [9] Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scGen predicts single-cell perturbation responses. *Nature methods*, 16(8):715, 2019.
- [10] Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian J Theis, and F Alexander Wolf. Conditional out-of-sample generation for unpaired data using trVAE. *arXiv preprint arXiv:1910.01791*, 2019.
- [11] Matthew Amodio, D. V. Dijk, R. Montgomery, Guy Wolf, and Smita Krishnaswamy. Out-of-sample extrapolation with neuron editing. *arXiv: Quantitative Methods*, 2018.
- [12] W. Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8 1:118–27, 2007.
- [13] J. Leek and John D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3, 2007.
- [14] Laleh Haghverdi, A. Lun, Michael D. Morgan, and J. Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36:421–427, 2018.
- [15] Allison Warren, Andrew Jones, Tsukasa Shibue, William C Hahn, Jesse S Boehm, Francisca Vazquez, Aviadi Tsherniak, and James M McFarland. Global computational alignment of tumor and cell line transcriptional profiles. *Nature Communications*, 12(22), 2021.
- [16] Kihyuk Sohn, Honglak Lee, and Xinchun Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.
- [17] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.
- [18] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.
- [19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.
- [21] Allison Warren, Yejia Chen, Andrew Jones, Tsukasa Shibue, William C Hahn, Jesse S Boehm, Francisca Vazquez, Aviadi Tsherniak, and James M McFarland. Global computational alignment of tumor and cell line transcriptional profiles. *Nature Communications*, 12(1):1–12, 2021.
- [22] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [23] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- [24] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

- [25] Emile Mathieu, Tom Rainforth, N. Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412. PMLR, 2019.
- [26] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018.
- [27] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [28] Kenneth A. Bollen. *Structural equation models*. Wiley, 2005.
- [29] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029, 2016.
- [30] Jonas Peters, Joris Mooij, Dominik Janzing, and Bernhard Schölkopf. Identifiability of causal graphs using functional models. *arXiv preprint arXiv:1202.3757*, 2012.
- [31] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.
- [32] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *arXiv preprint arXiv:1703.06856*, 2017.
- [33] Craig A Glastonbury, Michael Ferlaine, Christoffer Nellåker, and Cecilia M Lindgren. Adjusting for confounding in unsupervised latent representations of images. *arXiv preprint arXiv:1811.06498*, 2018.
- [34] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel methods in computational biology*, 47:35–70, 2004.
- [35] Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087*, 2018.
- [36] Romain Lopez, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Information constraints on auto-encoding variational bayes. In *Advances in Neural Information Processing Systems*, pages 6114–6125, 2018.
- [37] Kaspar Märtens and Christopher Yau. Neural decomposition: Functional anova with variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 2917–2927. PMLR, 2020.
- [38] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pages 6446–6456, 2017.
- [39] H. Kim, Seungjae Shin, Joonho Jang, Kyungwoo Song, Weonyoung Joo, Wanmo Kang, and Il-Chul Moon. Counterfactual fairness with disentangled causal effect variational autoencoder. In *AAAI*, 2021.
- [40] Stephen R. Pfohl, Tony Duan, Daisy Yi Ding, and Nigam H. Shah. Counterfactual reasoning for fair clinical risk prediction. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 325–358, Ann Arbor, Michigan, 09–10 Aug 2019. PMLR.
- [41] Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, et al. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1):89, 2018.
- [42] Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell rna-seq batch correction. *Nature methods*, 16(1):43–49, 2019.
- [43] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [44] Christian P. Robert and Judith Rousseau. How Principled and Practical Are Penalised Complexity Priors? *Statistical Science*, 32(1):36–40, February 2017.
- [45] Murray Aitkin. Posterior Bayes Factors. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(1):111–142, 1991.
- [46] Paul Fearnhead and Dennis Prangle. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474, 2012.
- [47] Adam Foster, Martin Jankowiak, Matthew O’Meara, Yee Whye Teh, and Tom Rainforth. A unified stochastic gradient approach to designing bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR, 2020.
- [48] J. Weinstein, E. Collisson, G. Mills, K. Shaw, B. Ozenberger, Kyle Ellrott, I. Shmulevich, C. Sander, and Joshua M. Stuart. The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics*, 45:1113–1120, 2013.

- 802 [49] DS Gerhard, S Hunger, C Lau, J Maris, P Meltzer, S Meshinchi, E Perlman, J Zhang, J Guidry-Auvil, and  
803 M Smith. Therapeutically applicable research to generate effective treatments (target) project: Half of  
804 pediatric cancers have their own "driver" genes. In *PEDIATRIC BLOOD & CANCER*, volume 65, pages  
805 S45–S45. WILEY 111 RIVER ST, HOBOKEN 07030-5774, NJ USA, 2018.
- 806 [50] Mary J Goldman, Brian Craft, Mim Hastie, Kristupas Repečka, Fran McDade, Akhil Kamath, Ayan  
807 Banerjee, Yunhai Luo, Dave Rogers, Angela N Brooks, et al. Visualizing and interpreting cancer genomics  
808 data via the xena platform. *Nature biotechnology*, 38(6):675–678, 2020.
- 809 [51] M. Ghandi, F. Huang, J. Jané-Valbuena, G. Kryukov, Christopher Lo, E. McDonald, J. Barretina, E. Gelfand,  
810 C. Bielski, Haoxin Li, Kevin Hu, Alexander Y. Andreev-Drakhlin, J. Kim, J. Hess, B. Haas, F. Aguet,  
811 B. Weir, M. Rothberg, B. Paolella, M. Lawrence, Rehan Akbani, Y. Lu, Hong L. Tiv, P. Gokhale, Antoine  
812 de Weck, Ali Amin Mansour, C. Oh, J. Shih, Kevin Hadi, Yanay Rosen, J. Bistline, K. Venkatesan,  
813 Anupama Reddy, Dmitriy Sonkin, Manway Liu, J. Lehár, J. Korn, D. Porter, M. Jones, J. Golji, G. Capon-  
814 igro, Jordan E. Taylor, C. Dunning, Amanda L Creech, Allison Warren, James M. McFarland, Mahdi  
815 Zamanighomi, A. Kauffmann, Nicolas Stransky, M. Imieliński, Y. Maruvka, A. Cherniack, Aviad Tsher-  
816 niak, F. Vazquez, J. Jaffe, A. A. Lane, D. Weinstock, C. Johannessen, Michael P. Morrissey, F. Stegmeier,  
817 R. Schlegel, W. Hahn, G. Getz, G. Mills, J. Boehm, T. Golub, L. Garraway, and W. Sellers. Next-generation  
818 characterization of the Cancer Cell Line Encyclopedia. *Nature*, 569:503–508, 2019.
- 819 [52] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression  
820 data analysis. *Genome biology*, 19(1):1–5, 2018.
- 821 [53] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon  
822 emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.