A. Estimation

We now derive a sampling algorithm to estimate IRS from a observational dataset $\mathcal{D} = \{\boldsymbol{x}^{(i)}, \boldsymbol{g}^{(i)}\}_{i=1,\dots,N}$ where $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$ and $\boldsymbol{g}^{(i)} \in \mathcal{G} = \mathcal{G}_1 \times \cdots \times \mathcal{G}_K$ with each $\mathcal{G}_k$ being discrete and finite. In case of continous $\mathcal{G}_k$ we first need to perform a discretization. The discretization steps trade off bias and variance of the estimate through the number of samples that are available per combination of generative factors.

We will provide an estimation procedure for $\mathrm{EMPIDA}(L|I,J)$ as:

$$\mathbb{E}_{\boldsymbol{g}_I}\left[\sup_{\boldsymbol{g}_J^\triangle} d\left(\mathbb{E}[\boldsymbol{Z}_L|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I)], \mathbb{E}[\boldsymbol{Z}_L|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle)]\right)\right]. \tag{4}$$

From that, also the IRS can be computed. In Section A.1 we provide a simplified version that is sufficient for disentanglement benchmarking based on perfectly crossed noise free datasets. Readers most interested in this application might skip to that part.

The main ingredient for this estimation to work is provided by our constrained causal model (i.e., a disentangled process) that implies that the backdoor criteria can be applied, which we showed in Proposition 1. Further, we already saw in Eq. (3.2) that $p(\boldsymbol{z}_L|\boldsymbol{x}, \mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle)) = p(\boldsymbol{z}_L|\boldsymbol{x})$. This can be used to write the conditional expected value of $\boldsymbol{Z}_L$ as:

$$\begin{aligned}
&\mathbb{E}[\boldsymbol{Z}_L|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle)] \\
&= \int \boldsymbol{z}_L\, p(\boldsymbol{z}_L|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle))\, d\boldsymbol{z}_L \\
&\overset{(3.2)}{=} \int\int \boldsymbol{z}_L\, p(\boldsymbol{z}_L|\boldsymbol{x})\, p(\boldsymbol{x}|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle))\, d\boldsymbol{x}\, d\boldsymbol{z}_L \\
&= \int \left(\int \boldsymbol{z}_L\, p(\boldsymbol{z}_L|\boldsymbol{x})\, d\boldsymbol{z}_L\right) p(\boldsymbol{x}|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle))\, d\boldsymbol{x} \\
&\overset{\text{Prop.1(f)}}{=} \int \left(\int \boldsymbol{z}_L\, p(\boldsymbol{z}_L|\boldsymbol{x})\, d\boldsymbol{z}_L\right) \\
&\qquad \left(\int p(\boldsymbol{x}|\boldsymbol{g}_I, \boldsymbol{g}_J^\triangle, \boldsymbol{g}_{\setminus(I\cup J)})\, p(\boldsymbol{g}_{\setminus(I\cup J)})\, d\boldsymbol{g}_{\setminus(I\cup J)}\right) d\boldsymbol{x} \\
&= \int E(\boldsymbol{x})_L \left(\int p(\boldsymbol{x}|\boldsymbol{g}_I, \boldsymbol{g}_J^\triangle, \boldsymbol{g}_{\setminus(I\cup J)})\, p(\boldsymbol{g}_{\setminus(I\cup J)})\, d\boldsymbol{g}_{\setminus(I\cup J)}\right) d\boldsymbol{x}
\end{aligned} \tag{5}$$

where the elements L of encoding $E(\cdot)$ are defined as:

$$E(\boldsymbol{x})_L := \int \boldsymbol{z}_L\, q_\phi(\boldsymbol{z}_L|\boldsymbol{x})\, d\boldsymbol{z}_L.$$

It is now apparent how this formula can be used to estimate the expected value using the sample mean (or a robust alternative in case outliers in $\boldsymbol{x}$ are to be expected) based on a set of samples $\tilde{\mathcal{D}}$ drawn from $\int p(\boldsymbol{x}|\boldsymbol{g}_I, \boldsymbol{g}_J^\triangle, \boldsymbol{g}_{\setminus(I\cup J)})\, p(\boldsymbol{g}_{\setminus(I\cup J)})\, d\boldsymbol{g}_{\setminus(I\cup J)}$ using the law of large numbers (LLN), i.e.,

$$\mathbb{E}[\boldsymbol{Z}_L|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle)] \overset{\text{LLN}}{\approx} \frac{1}{|\tilde{\mathcal{D}}|} \sum_{\boldsymbol{x} \in \tilde{\mathcal{D}}} E(\boldsymbol{x})_L. \tag{6}$$

However, all we are given are the samples $\mathcal{D}$ drawn from $p(\boldsymbol{x}, \boldsymbol{g}) = p(\boldsymbol{x}|\boldsymbol{g})\, p(\boldsymbol{g})$ where the generative factors could be confounded $p(\boldsymbol{g}) = \int p(\boldsymbol{g}|\boldsymbol{c}) p(\boldsymbol{c})\, d\boldsymbol{c}$. This is why we now provide an importance sampling based adjusted estimation of the expected value of any function of the observations $h(\boldsymbol{X})$ after an intervention on $\boldsymbol{G}_J$ has occured and while conditioning on $\boldsymbol{G}_I$, i.e., $\mathbb{E}[h(\boldsymbol{X})|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle)]$. This procedure can then be used to estimate Eq. (5), as a special case with $h(\cdot) = E(\cdot)_L$, directly from $\mathcal{D}$.

By denoting the Kronecker-delta as δ we obtain:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{X}}[h(\boldsymbol{x})|\mathrm{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^\triangle)] \\
&\overset{\text{derivation of (5)}}{=} \int\int h(\boldsymbol{x})\, p(\boldsymbol{x}|\boldsymbol{g}_I, \boldsymbol{g}_J^\triangle, \boldsymbol{g}_{\setminus(I\cup J)}) \\
&\qquad p(\boldsymbol{g}_{\setminus(I\cup J)})\, d\boldsymbol{g}_{\setminus(I\cup J)}\, d\boldsymbol{x} \\
&\overset{\boldsymbol{g}' = (\boldsymbol{g}'_I, \boldsymbol{g}'_J, \boldsymbol{g}'_{\setminus(I\cup J)})}{=} \int\int h(\boldsymbol{x})\, p(\boldsymbol{x}|\boldsymbol{g}')\, p(\boldsymbol{g}'_{\setminus(I\cup J)})\, \delta(\boldsymbol{g}'_I - \boldsymbol{g}_I) \\
&\qquad \delta(\boldsymbol{g}'_J - \boldsymbol{g}_J^\triangle)\, d\boldsymbol{g}'\, d\boldsymbol{x} \\
&\overset{\text{LLN}}{\approx} \frac{1}{N} \sum_{\boldsymbol{x}^{(i)}, \boldsymbol{g}^{(i)} \in \mathcal{D}} h(\boldsymbol{x}^{(i)}) \\
&\qquad \frac{p(\boldsymbol{x}^{(i)}|\boldsymbol{g}^{(i)}) p(\boldsymbol{g}^{(i)}_{\setminus(I\cup J)}) \delta(\boldsymbol{g}^{(i)}_I - \boldsymbol{g}_I)\, \delta(\boldsymbol{g}^{(i)}_J - \boldsymbol{g}_J^\triangle)}{p(\boldsymbol{x}^{(i)}|\boldsymbol{g}^{(i)}) p(\boldsymbol{g}^{(i)})} \\
&= \sum_{\substack{\boldsymbol{x}^{(i)}, \boldsymbol{g}^{(i)} \in \mathcal{D} \\ \text{where } \boldsymbol{g}^{(i)}_I = \boldsymbol{g}_I \\ \text{and } \boldsymbol{g}^{(i)}_J = \boldsymbol{g}_J}} h(\boldsymbol{x}^{(i)}) \frac{p(\boldsymbol{g}^{(i)}_{\setminus(I\cup J)})}{N\, p(\boldsymbol{g}^{(i)})}
\end{aligned} \tag{7}$$

We can rewrite the weighting term as:

$$w_i := \frac{p(\boldsymbol{g}^{(i)}_{\setminus(I\cup J)})}{N\, p(\boldsymbol{g}^{(i)})} = \frac{1}{N\, p(\boldsymbol{g}^{(i)}_I, \boldsymbol{g}^{(i)}_J | \boldsymbol{g}^{(i)}_{\setminus(I\cup J)})}$$

which gives us the natural interpretation that samples $\boldsymbol{g}_I, \boldsymbol{g}_J$ that would occur more often together with a certain $\boldsymbol{g}_{\setminus(I\cup J)}$ need to be downweighted in order to correct for the confounding effects. We can also see that in case of statistical independence between the generative factors, this reweighting is not needed and we can simply use the sample mean with the subselection of the dataset $\mathcal{D}_{sel} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{g}^{(i)}) \in \mathcal{D} : \boldsymbol{g}^{(i)}_I = \boldsymbol{g}_I \text{ and } \boldsymbol{g}^{(i)}_J = \boldsymbol{g}_J^\triangle\}$.

Since we assume $\boldsymbol{G}$ to be discrete, we can estimate these reweighting factors w_i from observed frequencies. Even though this sampling procedure looks non-trivial, we show

in Section 5 how it can be used to obtain an $\mathcal{O}(N)$ estimation algorithm for $\text{EMPIDA}(L|I,J)$.

A.1. Crossed Dataset without Noise: Benchmarking Disentanglement

In many benchmark datasets for disentanglement (e.g. dsprites) the observations are obtained noise free and the dataset contains all possible crossings of generative factors exactly ones. This makes the estimation of the disentanglement score very efficient, as we have $|\mathcal{D}^{(k,l)}_{I=\{i\},J=\{1,\dots,K\}\setminus\{i\}}| = 1$. Furthermore, since no confounding is present, we can use conditioning to estimate the interventional effect, i.e., $p(\boldsymbol{x}|\text{do}(G_i \leftarrow g_i)) = p(\boldsymbol{x}|g_i)$, as seen in Proposition 1 (g). In order to obtain the disentanglement score of Z_l, as discussed in Eq. (3), we therefore just need to compute the PIDA value:

$$d(\mathbb{E}[\boldsymbol{Z}_l|g_i^{(k)}], E(\tilde{\boldsymbol{x}})_l) \quad \forall \tilde{\boldsymbol{x}} \in \mathcal{D}_i^{(k)}$$

for all generative factors G_i and realizations thereof $\{g_i^{(1)}, \dots, g_i^{(N_i)}\}$. $\mathcal{D}_i^{(k)}$ is the set of observations that was generated with a particular configuration $g_i^{(k)}$. We choose the maximum value w.r.t. $\tilde{\boldsymbol{x}}$ as MPIDA and average over realizations $g_i^{(k)}$ to obtain:

$$\begin{aligned}\text{EMPIDA}_{li} &:= \text{EMPIDA}(\{l\}|\{i\}, \{1,\dots,K\}\setminus\{i\}) \\ &= \frac{1}{N_i}\sum_{k=1}^{N_i} \sup_{\tilde{\boldsymbol{x}}\in\mathcal{D}_i^{(k)}} d\left(\mathbb{E}[Z_l|g_i^{(k)}], E(\tilde{\boldsymbol{x}})\right).\end{aligned}$$

The estimate for the disentanglement score in Eq. (3) for Z_l follows from that:

$$D_l = \max_{i\in\{1,\dots,K\}} \left(1 - \frac{\text{EMPIDA}_{li}}{\sup_{\tilde{\boldsymbol{x}}\in\mathcal{D}} d\left(\mathbb{E}[Z_l], E(\tilde{\boldsymbol{x}})\right)}\right).$$

B. Proof of Proposition 1

Proof. Property (a) directly follows from Definition 1 and the definition of an independent causal mechanism. (b) and (c) can be read off the graphical model (Koller et al., 2009) in Figure 1 which does not contain any arrow from G_i to G_j for $i \neq j$ by Definition 1 of the constrained SCM. This is due to the fact that any distribution implied by an SCM is Markovian with respect to the corresponding graph (Peters et al., 2017, Prop. 6.31). (d) follows from the data processing inequality since we have $\boldsymbol{X} \perp\!\!\!\perp C|\boldsymbol{G}$. The non-existence of a directed path from G_j to G_i implies that there is no total causal effect (Peters et al., 2017, Prop. 6.14). This, in turn, is equivalent to property (e) (Peters et al., 2017, Prop. 6.13). Finally, since there are no arrows between the G_i's, the backdoor criterion (Peters et al., 2017, Prop. 6.41) can be applied to estimate the interventional effects in (f). In particular, $\boldsymbol{G}_{\setminus j}$ blocks all paths from G_j to $\boldsymbol{X}$ entering G_j through the backdoor (i.e., $G_j \leftarrow \cdots \rightarrow \boldsymbol{X}$) but at the same time does not contain any descendents of G_j since by definition $G_j \not\to G_i \; \forall i \neq j$. Property (g) also follows from $G_j \not\to G_i \; \forall i \neq j$ by using parent adjustment (Peters et al., 2017, Prop. 6.41), where in the case no confounding $\boldsymbol{PA}_j = \emptyset$. These properties is why the constrained SCM in Definition 1 is important for further estimation. □

C. Proof of Proposition 2

Proof. The encodings in line 1 requires one pass through the dataset $\mathcal{D}$. So does the estimation of the occurance frequencies in line 1 as one can use a hash table to keep track of the number of occurances of each possible realization. Therefore, the preprocessing steps scale with $\mathcal{O}(N)$.

Further, also the partitioning of the full dataset into $\mathcal{D} = \bigcup_{k=1}^{N_I}\bigcup_{l=1}^{N_{I,J}^{(k)}} \mathcal{D}_{I,J}^{(k,l)}$, which is done in lines 1, 1 and 1, can be done with two passes through the dataset by using hash tables: In the first pass we create buckets with $\boldsymbol{g}_I^{(k)}$ as keys. Consequently, we can pass through all of these buckets to create subbuckets where $\boldsymbol{g}_J^{(l)}$ is used as key. This reasoning is further illustrated in Figure 5 and leads us to the $\mathcal{O}(N)$ complexity of the partitioning.

The remaining computational bottleneck are the computations of `mean` in line 1 and $\texttt{mean}_{\texttt{intv}}$ in line 1. Using Eq. (7) we obtain $\mathbb{E}[\boldsymbol{Z}_L|\text{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I^{(k)})] \approx \sum_{\boldsymbol{x}^{(i)}\in\mathcal{D}_I^{(k)}} w_i\, E(\boldsymbol{x}^{(i)})$ to compute `mean` and $\mathbb{E}[\boldsymbol{Z}_L|\text{do}(\boldsymbol{G}_I \leftarrow \boldsymbol{g}_I^{(k)}, \boldsymbol{G}_J \leftarrow \boldsymbol{g}_J^{(l)})] \approx \sum_{\boldsymbol{x}^{(i)}\in\mathcal{D}_{I,J}^{(k,l)}} \tilde{w}_i\, E(\boldsymbol{x}^{(i)})$ to compute $\texttt{mean}_{\texttt{intv}}$. Since we already computed the encodings as well as the reweighting terms in the preprocessing step, these summations scale as $\mathcal{O}(|\mathcal{D}_I^{(k)}|)$ and $\mathcal{O}(|\mathcal{D}_{I,J}^{(k,l)}|)$. As can be seen in Figure 5, it holds that $\sum_{k=1}^{N_I}|\mathcal{D}_I^{(k)}| = N$ as well as $\sum_{k=1}^{N_I}\sum_{l=1}^{N_{I,J}^{(k)}}|\mathcal{D}_{I,J}^{(k,l)}| = N$ which implies the total computational complexity of $\mathcal{O}(N)$. □

Real World Considerations: Though this estimation procedure scales $\mathcal{O}(N)$ in the dataset size, the required number of observations for a fixed estimation quality (i.e., if $|\mathcal{D}_{I,J}^{(k,l)}|$ should stay constant) might become very large, as we have exponentially growing (in $|I|$ and $|J|$) many possible combinations to consider. This is why some trade-offs need to be made when comparing large sets of factors. The estimation for $|I|, |J| = 1, 2$ or 3, however, usually works well. One trade-off parameter is the discretization step of of g_i's. Partitioning a factor into fewer realizations yields less possible combinations and hence larger sets $\mathcal{D}_{I,J}^{(k,l)}$. In general, the more noise we expect in $\boldsymbol{x}$ the larger the sets $\mathcal{D}_{I,J}^{(k,l)}$ we want to have in order to obtain stable estimates

$\boldsymbol{G}_I$ fixed	$\boldsymbol{G}_J$ fixed	remaining $\boldsymbol{G}_{\setminus(I\cup J)}$		
$\boldsymbol{g}_I^{(1)}$	$\boldsymbol{g}_J^{(1)}$	$\mathcal{D}_{I,J}^{(1,1)}$	$\Big\}\,\vert\mathcal{D}_{I,J}^{(k,l)}\vert$	$\Big\}\,\vert\mathcal{D}_I^{(k)}\vert$	$\Big\}\,\vert\mathcal{D}\vert = N$
	$\cdots$	$\cdots$			
	$\boldsymbol{g}_J^{(N_{I,J}^{(1)})}$	$\mathcal{D}_{I,J}^{(1,N_{I,J}^{(1)})}$			
$\cdots$	$\cdots$	$\cdots$			
	$\cdots$	$\cdots$			
	$\cdots$	$\cdots$			
$\boldsymbol{g}_I^{(N_I)}$	$\boldsymbol{g}_J^{(1)}$	$\mathcal{D}_{I,J}^{(N_I,1)}$			
	$\cdots$	$\cdots$			
	$\boldsymbol{g}_J^{(N_{I,J}^{(N_I)})}$	$\mathcal{D}_{I,J}^{(N_I,N_{I,J}^{(N_I)})}$			

Figure 5. **Partitioning of Dataset:** In order to estimate $\mathrm{EMPIDA}(L|I,J)$ we first partition the dataset according to possible realizations of $\boldsymbol{G}_I$ (first column), where we assume there are N_I many. This partitioning can be done in linear time $\mathcal{O}(N)$ by using hash tables with $\boldsymbol{g}_I^{(i)}$ as keys. For each such partition $D_I^{(k)}$ we can further split these sub-datasets according to realizations of $\boldsymbol{G}_J$ to obtain $\mathcal{D}_{I,J}^{(k,l)} = \{(\boldsymbol{x},\boldsymbol{g}) \in \mathcal{D} \text{ s.t. } \boldsymbol{g}_I = \boldsymbol{g}_I^{(k)}, \boldsymbol{g}_J = \boldsymbol{g}_J^{(l)}\}$, $l = 1,\ldots,N_I^{(k)}$ (illustrated as boxes in third column). We denote with $N_I^{(k)}$ the number of realizations of $\boldsymbol{G}_J$ that occur together with $\boldsymbol{g}_I^{(k)}$ (i.e., can be found in $\mathcal{D}_I^{(k)}$). This takes $\mathcal{O}(|D_I^{(k)}|)$ time per partition $D_I^{(k)}$ or $\mathcal{O}(\sum_{k=1}^{N_I}|D_I^{(k)}|) = \mathcal{O}(N)$ in total by again making use of hash tables.

of the expected values. Also, if we allow for fewer possible realizations in the generative factors, the smaller our dataset can be to cover all relevant combinations. However, larger discretization steps come at the cost of having a less sensitive score. Also note that taking the supremum is in general not vulnerable to outliers in $\boldsymbol{x}$ as we compute distances of *expected* values. When outliers are to be expected, a robust estimate for these expected values can be used. Only when little data is available special care needs to be taken.

D. Details of Experimental Setup

D.1. Validation Methods

We compute the feature importance based disentanglement scores, as discussed by Eastwood & Williams (2018), using random forests with 50 decision trees that are split up to a minimal leaf size of 500. As opposed to Eastwood & Williams (2018), we only use one single feature to 'randomly choose from' at each split, since this guarantees that each feature is equally given the chance to prove itself in reducing the out-of-bag error. When multiple features can be chosen from at each split, it is well possible that features with a mediocre importance are never chosen as there are features always yielding a better split. This would lead to an underestimation of their importance.

For the mutual information metric (Ridgeway & Mozer, 2018) we followed the original proposal of discretizing each latent dimension into 20 buckets and computing the discrete mutual information based on that. We found that using smaller discretization steps (i.e., more buckets) does not change the results notably.

Since we make comparisons to information based evaluation methodologies by Eastwood & Williams (2018) and Ridgeway & Mozer (2018), we here give a more in depth overview of these methods. The validation method of Eastwood & Williams (2018) is based on training a predictor model (e.g. a random forest) which tries to estimate the true generative factors based on the latent encoding. The way disentanglement can be observed is by analyzing the feature importances implicit in this regressor. Intuitively, we expect that in a disentangled representation, each dimension contains information about one single generative factor. In particular, Eastwood & Williams (2018) proceed as follows: Given a labeled dataset with generative factors and observations $\mathcal{D} = \{\boldsymbol{x}^{(i)}, \boldsymbol{g}^{(i)}\}_{i=1,\dots,N}$ and a given encoder E (to be evaluated), they first create the set of features $\{\boldsymbol{z}^{(i)} = E(\boldsymbol{x}^{(i)}) \in \mathbb{R}^{K'} : i = 1, \dots, N\}$. Using these K' features as predictors, they train an individual regressor f_i for each generative factor G_i, i.e., $\hat{G}_i = f_i(\boldsymbol{Z})$. As the basis for further computations, they set up a matrix of relative importances $\boldsymbol{R}$ based on these feature importance values. In particular, R_{ij} denotes the relative importance of the feature Z_i when predicting G_j.

Plotting the matrix $\boldsymbol{R}$ gives a good first impression of the disentanglement capabilities of an encoder. Ideally, we would want to see only one large value per row while the remaining entries should be zero. In our experimental evaluations we plot this matrix (together with similarly interpretable matrices of the other metrics) as is shown for example in Figure 6 on page 16.

To explicitly quantify this visual perspective, Eastwood & Williams (2018) summarize disentanglement as one score value which measures to what extent indeed each latent dimension can only be used to predict one generative factor (i.e., sparse rows). It is obtained by first computing the 'probabilities' of Z_i being important to predict G_j,

$$P_{ij} = R_{ij} / \sum_{k=0}^{K-1} R_{ik}$$

and the entropy of this distribution: $H_K(P_{i\cdot}) = -\sum_{k=0}^{K-1} P_{ik} \log_K P_{ik}$, where $K = \dim(\boldsymbol{g})$ is the number of generative factors. The disentanglement score of variable Z_i is then defined as $D_i = (1 - H_K(P_{i\cdot}))$. For example, if only one generative factor G_u can be predicted with Z_i, i.e., $P_{ij} = \delta_{iu}$, we obtain $D_i = 1$. If the explanatory power spreads over all factors equally, the score is zero. Using relative variable importance $\rho_i = \sum_j R_{ij} / \sum_{ij} R_{ij}$, which accounts for dead or irrelevant components in $\boldsymbol{Z}$, they find an overall disentanglement score as weighted average $S_D = \sum_i \rho_i D_i$. When later plotting the full importance matrices, we also provide information about the individual feature disentanglement scores D_i in the corresponding row labels. These feature-wise scores are better comparable between metrics since all of them have different heuristics to obtain the (weighted) average S_D.

As an additional measure to obtain a more complete picture of the quality of the learned code, they additionally propose the *informativeness* score. It tells us how much information about the generative factors is captured in the latent space and is computed as the out-of-bag prediction accuracy of the regressors $f_1, \dots, f_K$. In our evaluations in Section 6 we will also provide this score, as there is often a trade-off between a disentangled structure and information being preserved.

The mutual information based metric by Ridgeway & Mozer (2018) proceeds in a similar way to Eastwood & Williams (2018). However, instead of relying on a random forest to compute the feature importances, they use an estimate of the mutual information between encodings and generative factors. In particular, they also first compute an importance matrix $\tilde{\boldsymbol{R}}$ where the element $\tilde{R}_{ij}$ corresponds to the mutual information between Z_i and G_j. We also provide plots of this matrix whenever evaluations are made (e.g. Figure 6

on page 16). Another difference to Eastwood & Williams (2018) is that Ridgeway & Mozer (2018) do not compute entropies to measure the deviation from the ideal case of having only one large value per row. Instead, they compute a normalized squared difference between each row and its idealized case where all values except the largest are set to zero. To summarize the disentanglement scores of different dimensions in a feature space they use an unweighted average.

D.2. Disentanglement Approaches

For the disentangling VAE models we made use of existing implementations where this was available. Classic VAE (Kingma & Welling, 2014) and DIP-VAE (Kumar et al., 2018) we implemented ourselves and trained them for 300 epochs using Adam (Kingma & Ba, 2015) with a learning rate of `1e-4` and batch size of 512. We used the same neural network architecture as is described in the appendix of Chen et al. (2018). For DIP-VAE we set the parameters to $\lambda_d = 100, \lambda od = 10$, as is used in the original publication. For the annealed β-VAE approach (Burgess et al., 2018) we used the publicly available third party code from `https://github.com/1Konny/Beta-VAE`, where parameters are set to $C = 20$ and $\gamma = 100$. Also, for FactorVAE (Kim & Mnih, 2018) we used third party code from `https://github.com/1Konny/FactorVAE` with their parameter $\gamma = 6.4$. Chen et al. (2018) provided their own code for β-TCVAE at `https://github.com/rtqichen/beta-tcvae`, which we made use of. We kept their chosen default parameters ($\beta = 6.0$).

E. Visualisations of Importance Matrices

Plots of the full importance matrices for the considered latent spaces and all three validation metrics are included in Figures 8, 9, 10, 11 and 12. The y labels include the disentanglement scores of each individual feature Z_i.

A related visualization possibility to the one we propose in Section 6.3 is that of simple conditioning on different generative factors (without keeping one factor fixed). This is illustrated in Figure 7, where we plot the violin plots (i.e., density estimates) of $p(z_l|g_j)$ for all generative factors G_j (columns) and realizations of them g_j (x axis). This kind of visualization works well to discover simple dependency patterns as well as their noise levels.

F. Visualisations of Interventional Effects

We provide further visualizations of the full latent spaces and their dependency structure (produced by the to be made publicly available code) of a couple of models in Figures 13, 14, 15, 16 and 17.

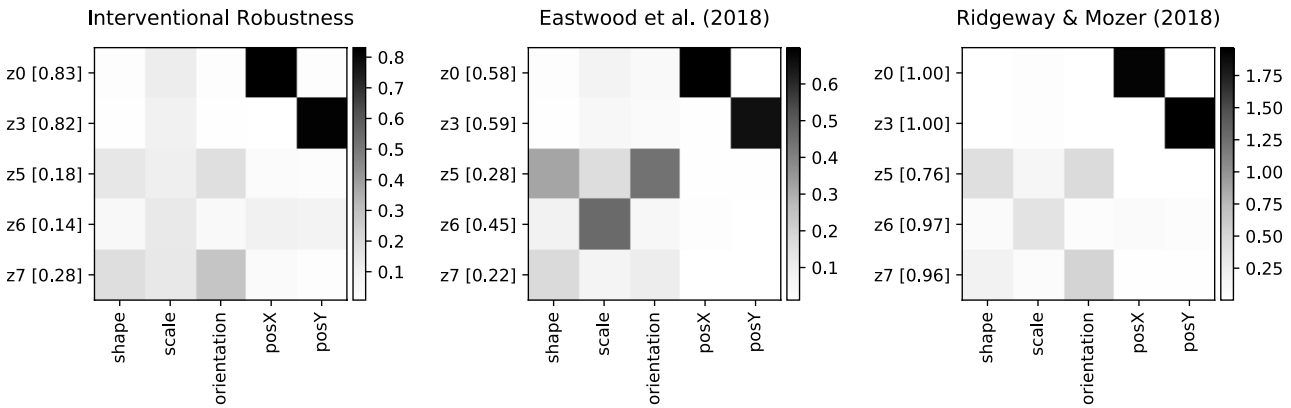

Figure 6. **Dependency Matrices:** These plots illustrate the different dependency structure matrices (of the features learned by the DIP model) that are used by the three discussed evaluation metrics. The rows correspond to the latent space dimensions Z_i (disentanglement score of each feature is given in brackets) and the columns to generative factors G_j (labels indicates their interpretation in the dsprites dataset).

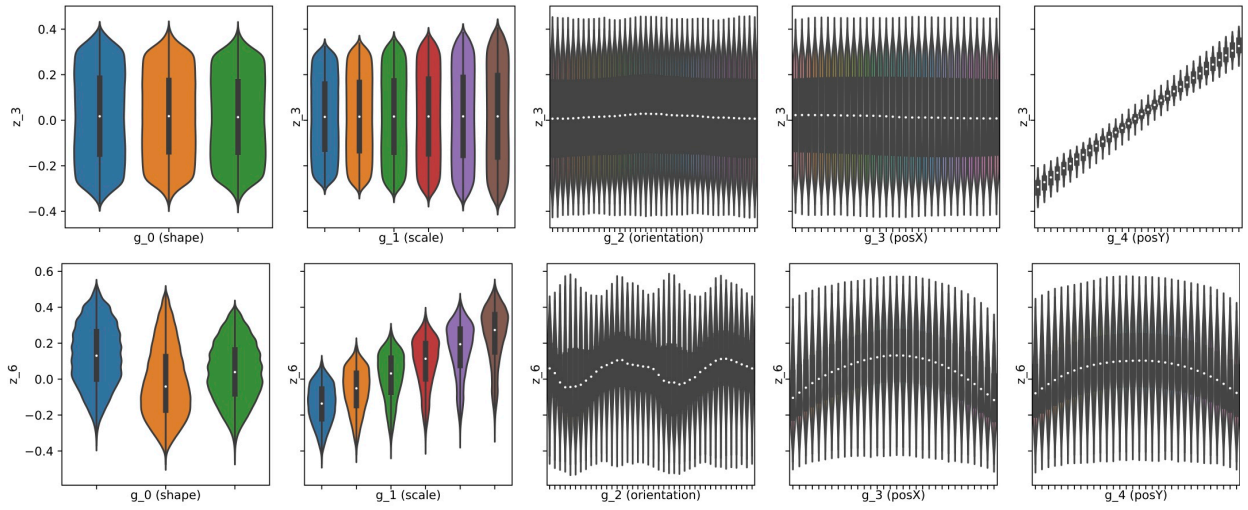

Figure 7. **Visualising Conditional Distributions:** These plots illustrate the violin plots (density estimates) of the conditional distributions $p(z_l|g_j)$ for all generative factors G_j (different boxes) and for all realizations g_j of G_j each (x axis in each plot). The upper plot corresponds to the well disentangled and robust feature Z_3 of the DIP model, the lower to the disentangled (according to MI and FI) but not robust (according to IRS) feature Z_6.

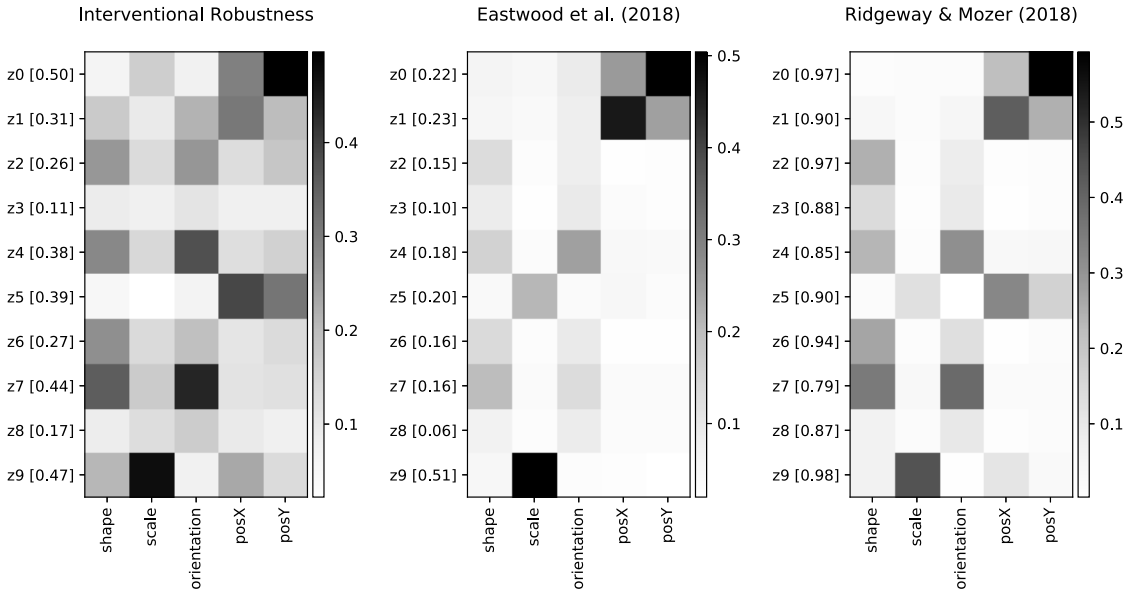

Figure 8. Importance matrices of all three validation metrics for the classic VAE model (Kingma & Welling, 2014).

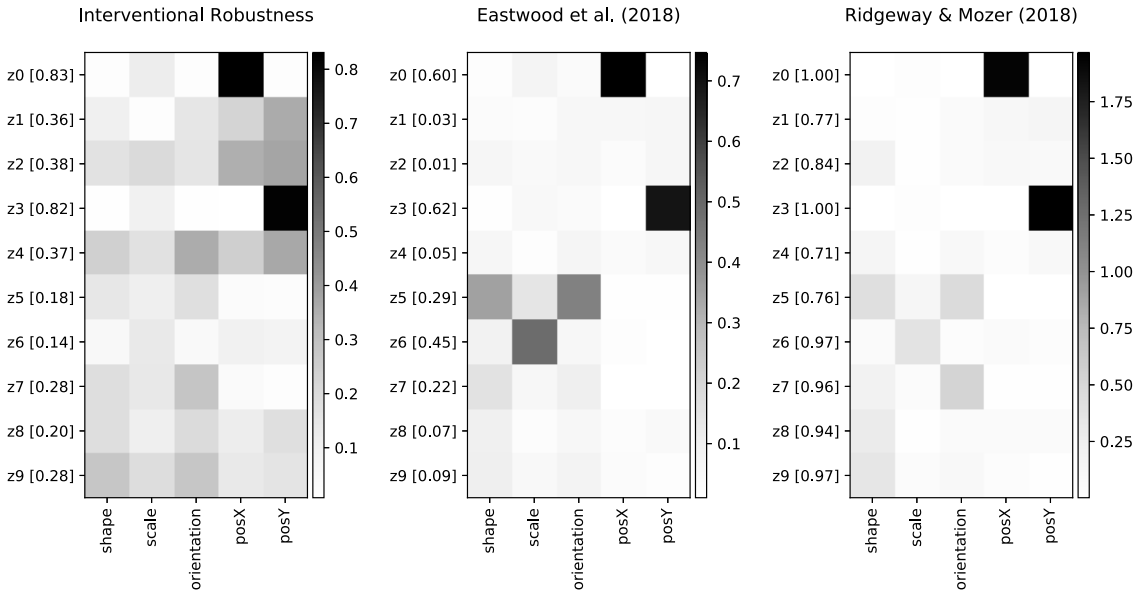

Figure 9. Importance matrices of all three validation metrics for the DIP-VAE model (Kumar et al., 2018).

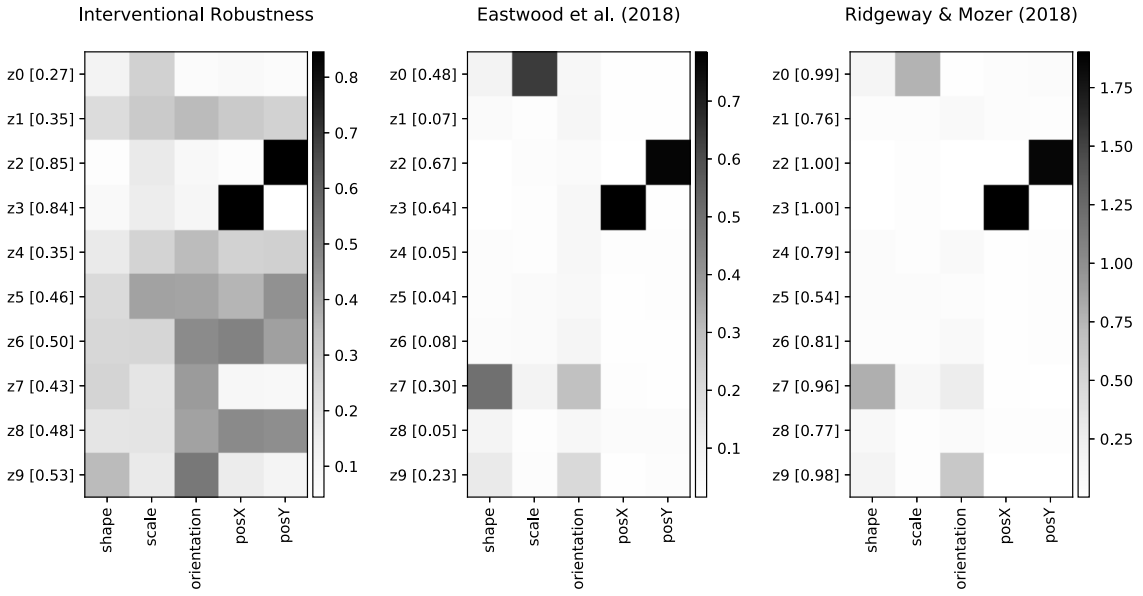

Figure 10. Importance matrices of all three validation metrics for the annealed β-VAE model (Higgins et al., 2017; Burgess et al., 2018).

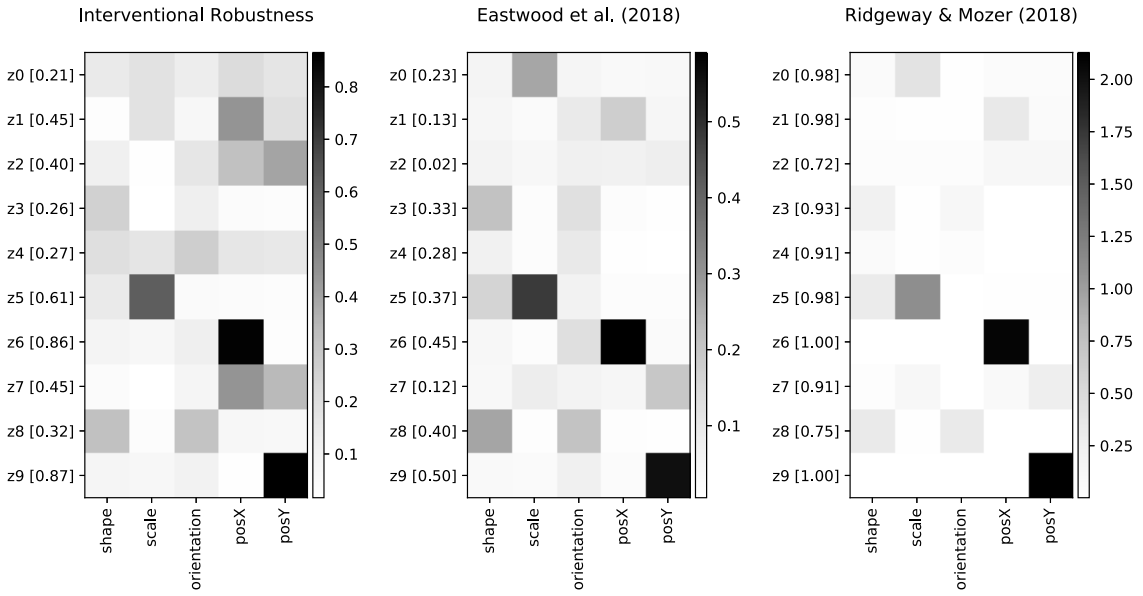

Figure 11. Importance matrices of all three validation metrics for the FactorVAE model (Kim & Mnih, 2018).

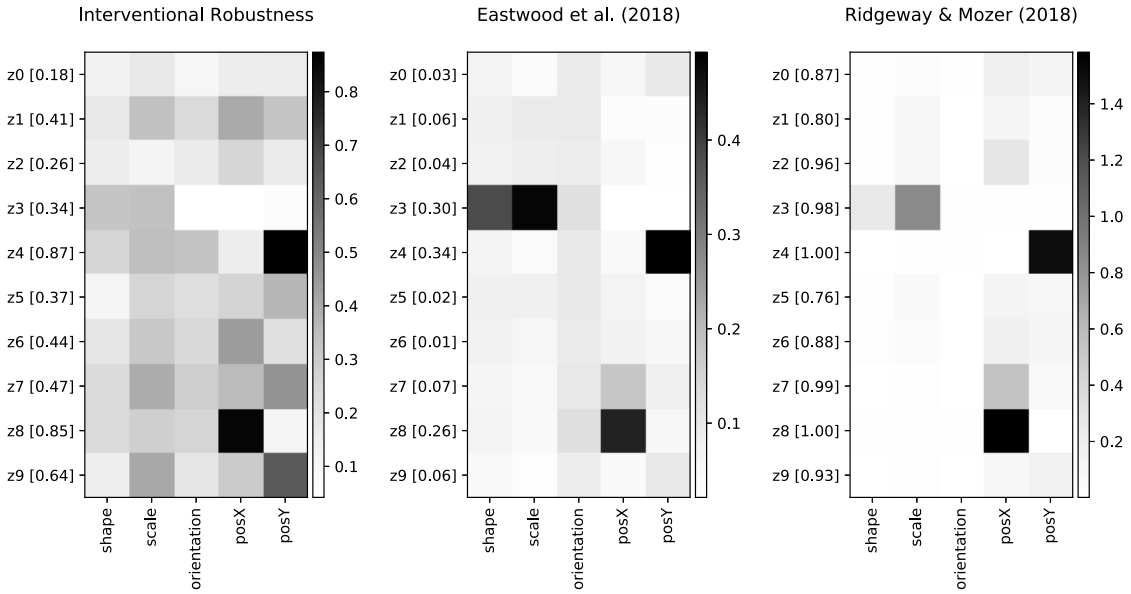

Figure 12. Importance matrices of all three validation metrics for the β-TCVAE model (Chen et al., 2018).

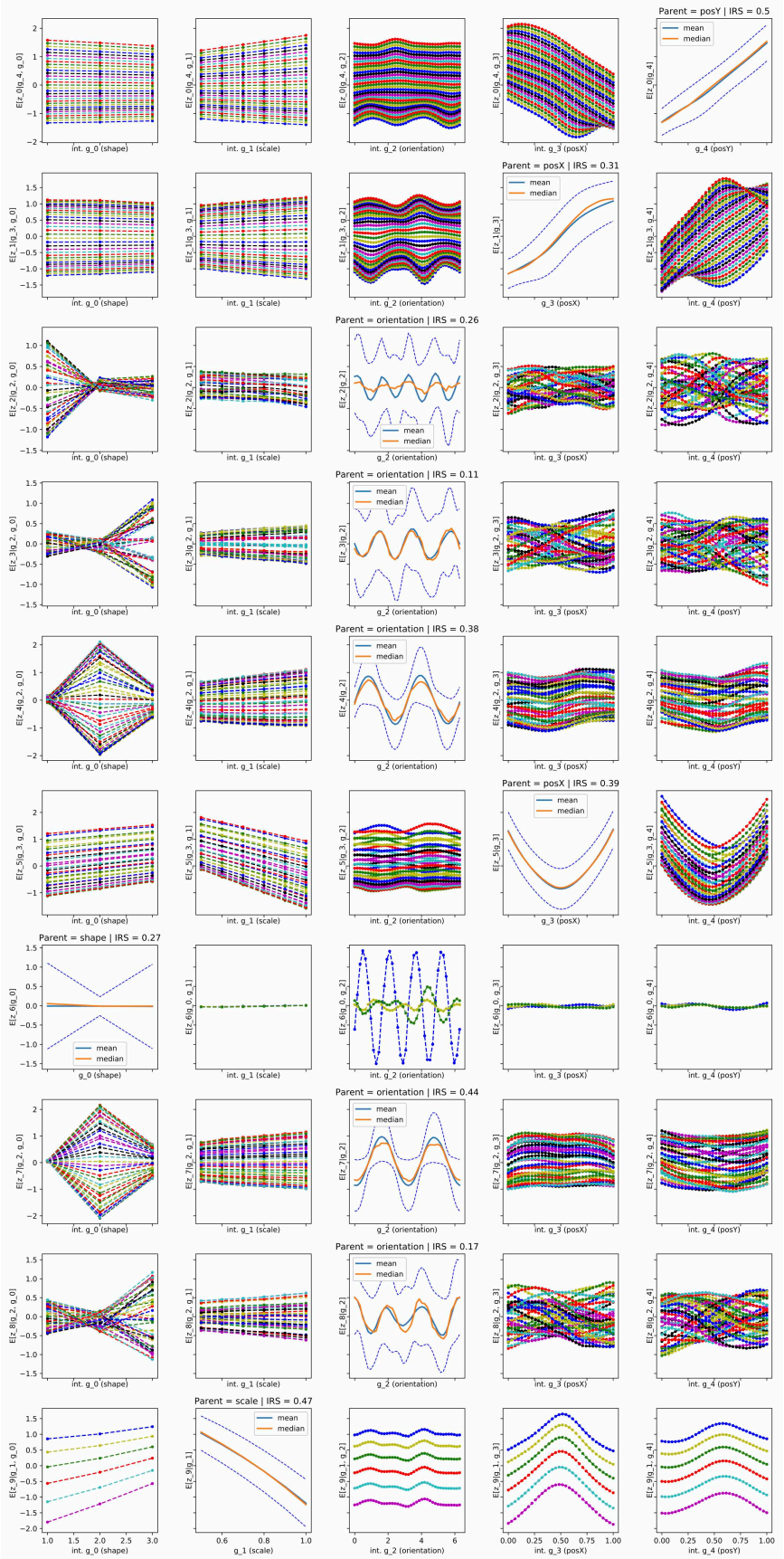

Figure 13. Visualization of interventional effects in the regular VAE model (Kingma & Welling, 2014).

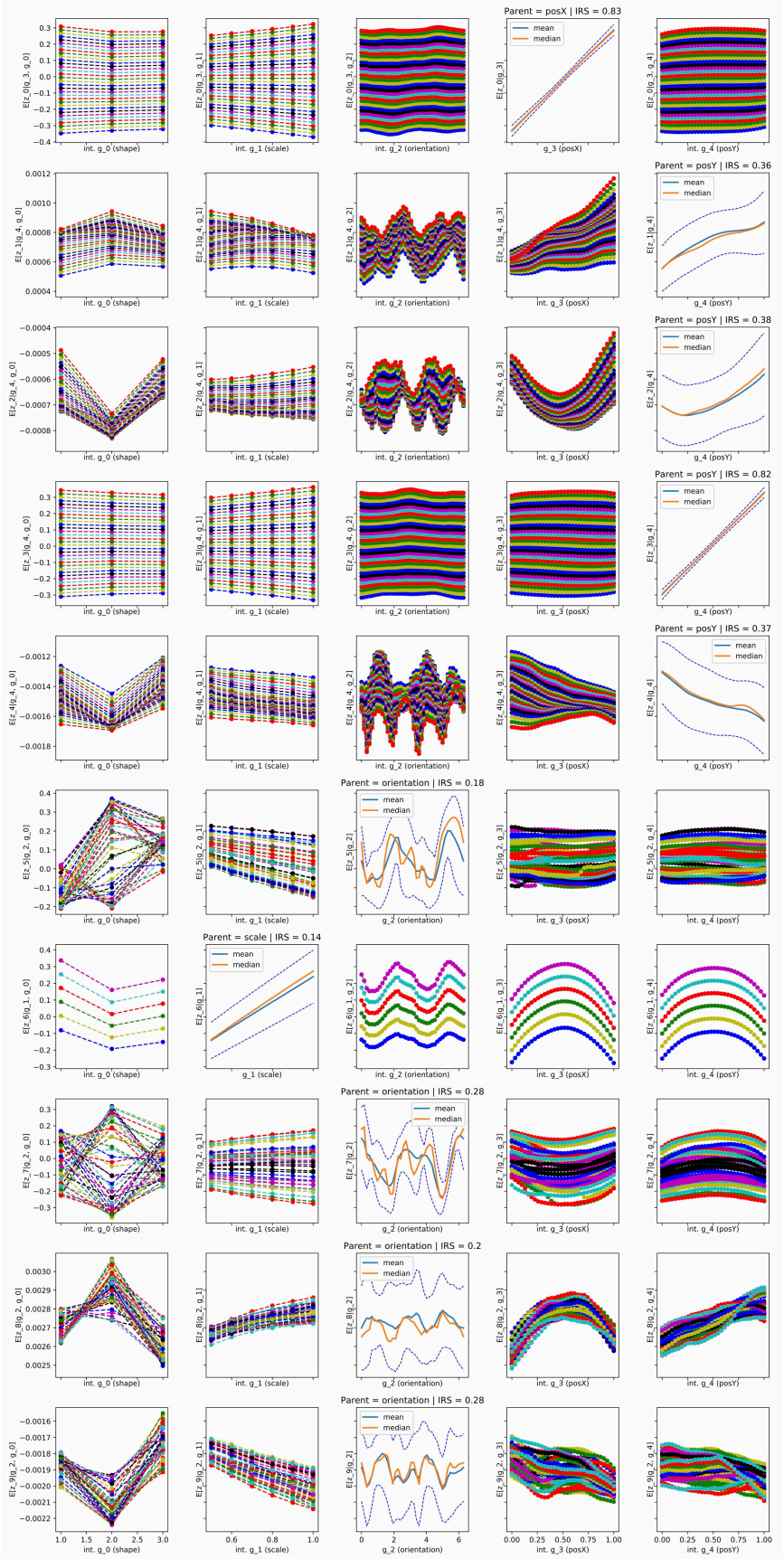

Figure 14. Visualization of interventional effects in the DIP-VAE model (Kumar et al., 2018).

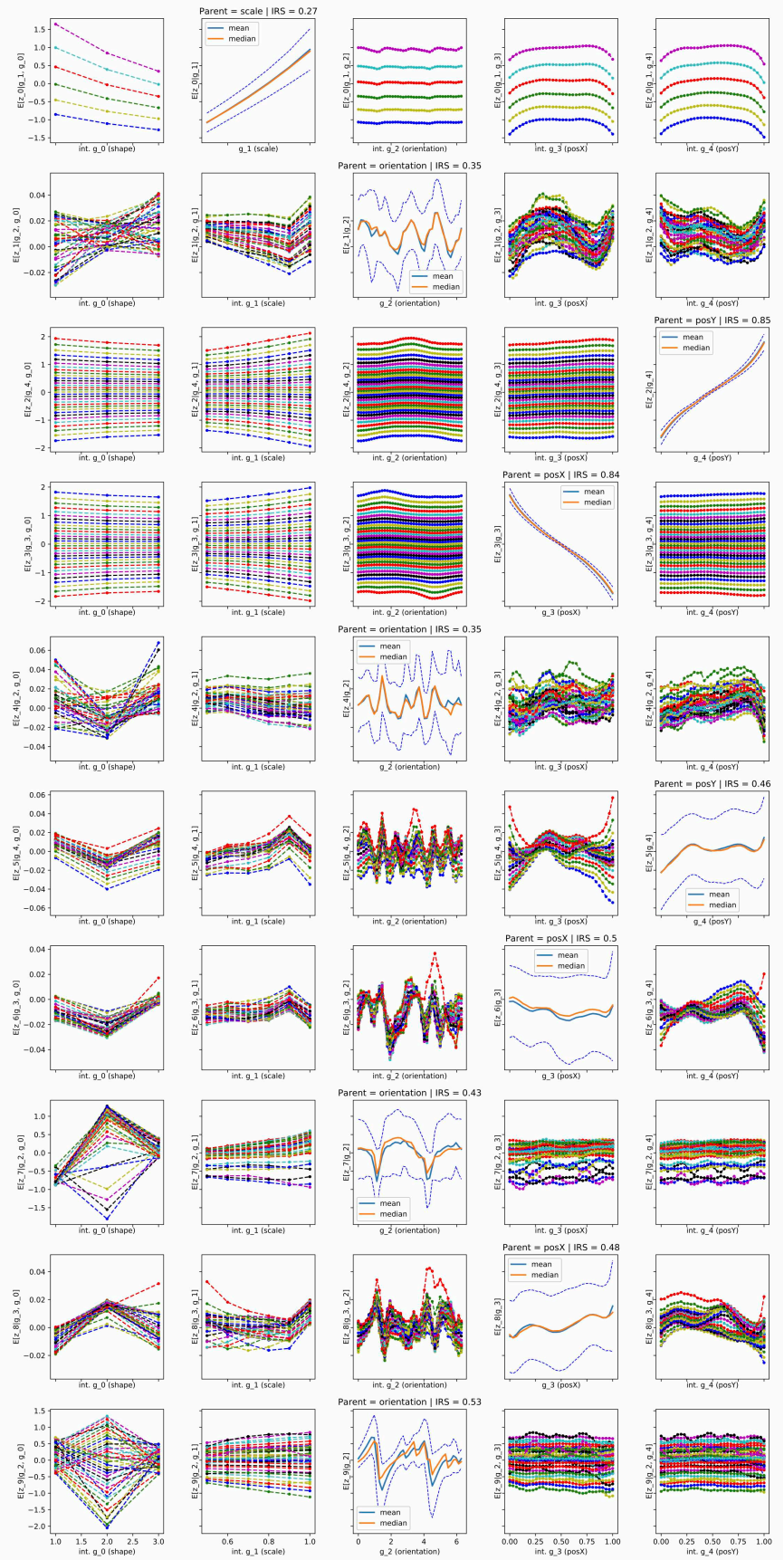

Figure 15. Visualization of interventional effects in the annealed β-VAE model (Higgins et al., 2017; Burgess et al., 2018).

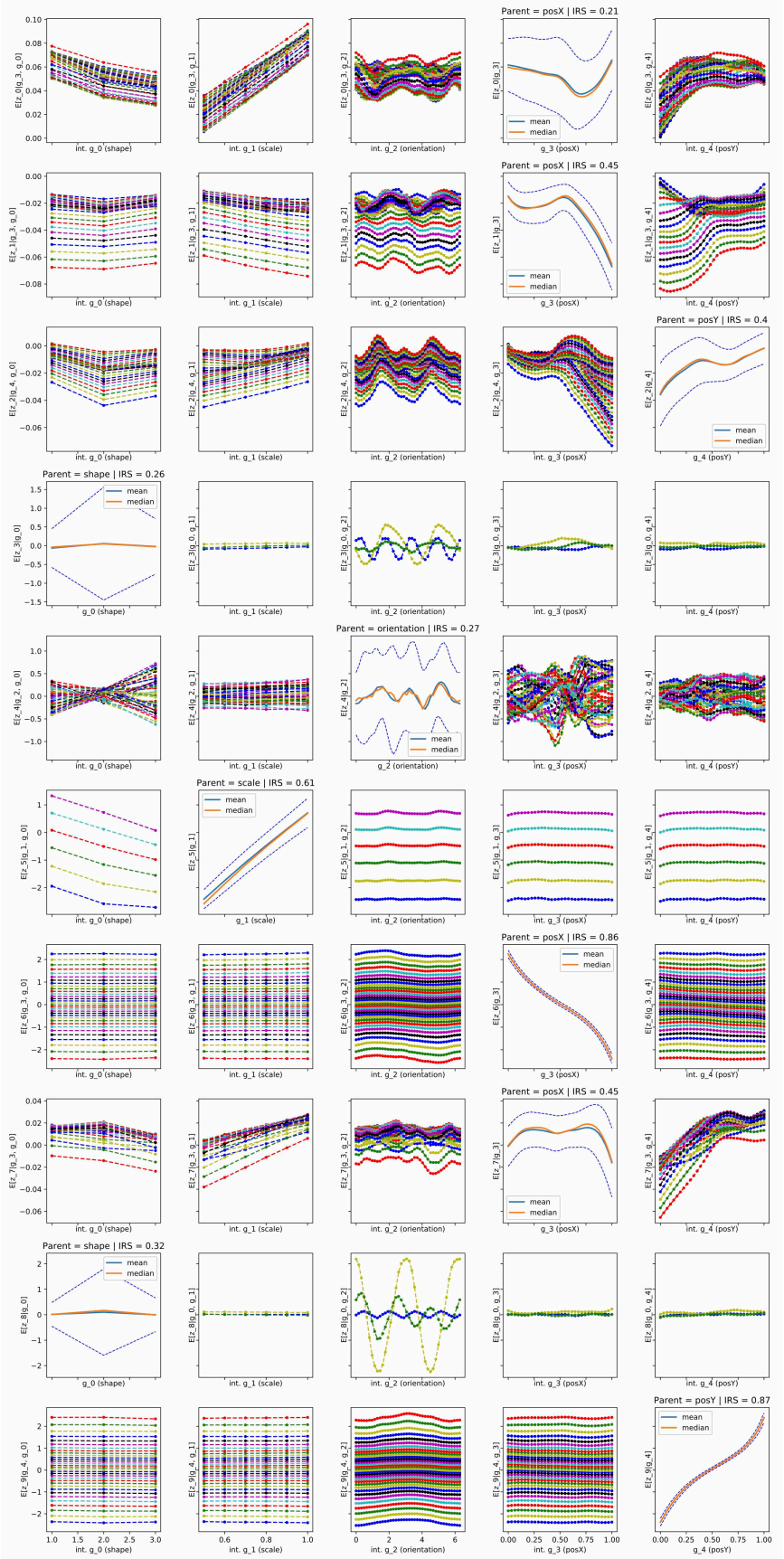

Figure 16. Visualization of interventional effects in the Factor-VAE model (Kim & Mnih, 2018).

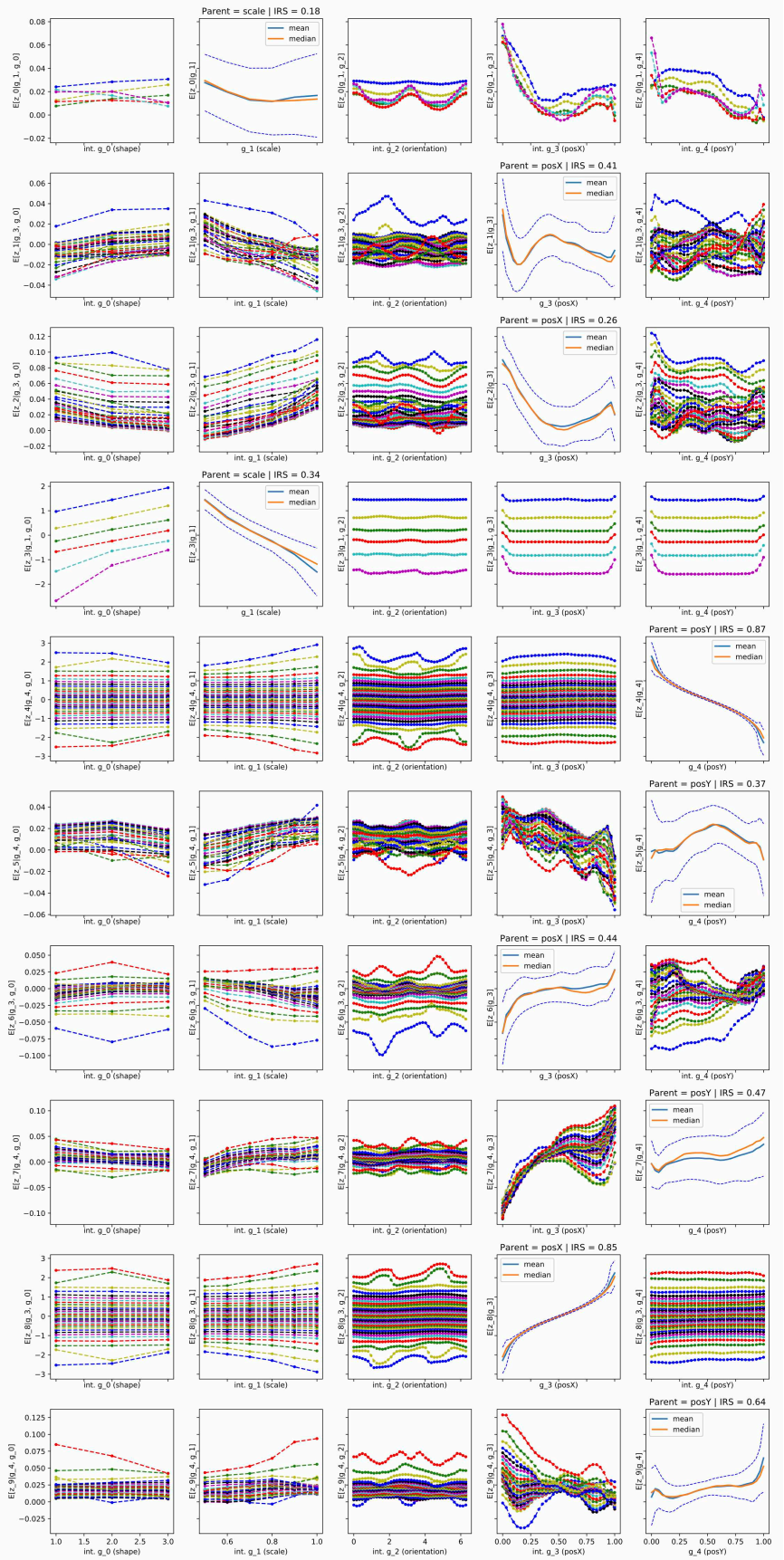

Figure 17. Visualization of interventional effects in the β-TCVAE model (Chen et al., 2018).