Learning disentangled representations via product manifold projection - Supplementary Material

Marco Fumero [1] **Luca Cosmo** [1 2] **Simone Melzi** [1] **Emanuele Rodolà** [1]

Abstract

This document contains the supplementary material for the paper *"Learning disentangled representations via product manifold projection"*. We provide details about the experiments of the main paper, including details on the architecture, hyper-parameters and training strategy together with an ablation study on the loss terms. We also theoretically motivate the choice of the sum operation as aggregation function for our latent subspaces.

1. Implementation details

For the experiments on the datasets *DSprites* (Higgins et al., 2017), *Shapes3D* (Kim & Mnih, 2018), *Cars3D* (Reed et al., 2015), *SmallNORB* (LeCun et al., 2004), we implement a simple convolutional architecture for both the encoder and the decoder. We report the detailed parameters in Table 1, where d refers to the dimensionality of the latent space $\mathcal{Z}$, which bounds the maximum dimensionality of each of the k latent subspaces $\mathcal{S}_1 \dots \mathcal{S}_k$. The architecture of the nonlinear projectors P_i is described in Table 3. For the *FAUST* dataset we employ a PointNet (Qi et al., 2017) based architecture for the encoder and a simple MLP for the decoder. Details are reported in Table 2

Table 1: Convolutional architecture used in image datasets.

Encoder	**Decoder**
Input : $64 \times 64 \times$ number of channels	Input : $\mathbb{R}^d$
4×4conv, 32 ReLU, stride 2, padding 1	FC, 256, ReLU
4×4conv, 32 ReLU, stride 2, padding 1	FC, 256, ReLU
4×4conv, 64 ReLU, stride 2, padding 1	FC, $64 \times 4 \times 4$, ReLU
4×4conv, 64 ReLU, stride 2, padding 1	4×4upconv, 64 ReLU, stride 2, padding 1
FC, 256, ReLU	4×4upconv, 32 ReLU, stride 2, padding 1
FC, d	4×4upconv, 32 ReLU, stride 2, padding 1
-	4×4upconv, number of channels, stride 2, padding 1

1.1. Experimental settings

For the comparisons with (Locatello et al., 2020) and its top performer model Ada-GVAE presented in Tables 1-4 in the main paper, we set the dimensionality d of the latent space $\mathcal{Z}$ to 10, and the number of subspaces k to 10. This puts us in a setting that is as close as possible to (Locatello et al., 2020), where the latent space is 10-dimensional and the subspaces are 1-dimensional by construction. For all the quantitative experiments we trained 5 times the same model with different random seeds, and report the median results on each dataset. A summary of the hyperparameters are in Table 4.

[1] Sapienza, University of Rome, Rome, Italy [2] Università della Svizzera italiana, Lugano, Switzerland. Correspondence to: Marco Fumero <fumero@di.uniroma1.it>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

Table 2: PointNet - MLP architecture used in FAUST dataset.

Encoder	Decoder
Input : 2500$\times$ 3	Input : $\mathbb{R}^d$
$1 \times$ 1conv, 32, BatchNorm, ReLU,	FC, 1024, LeakyReLU
$1 \times$ 1conv, 128, BatchNorm, ReLU,	FC, 2048, LeakyReLU
$1 \times$ 1conv, 256, BatchNorm, ReLU,	FC, 2500 $\times$ 3, ReLU
$1 \times$ 1conv, 512,	-
MaxPooling,	-
FC, 512, BatchNorm, ReLU,	-
FC, 256, BatchNorm, ReLU,	-
FC, 128, BatchNorm, ReLU,	-
FC, d,	-

Table 3: Projectors architecture.

P
Input : $\mathbb{R}^d$
FC d, ReLU
FC d

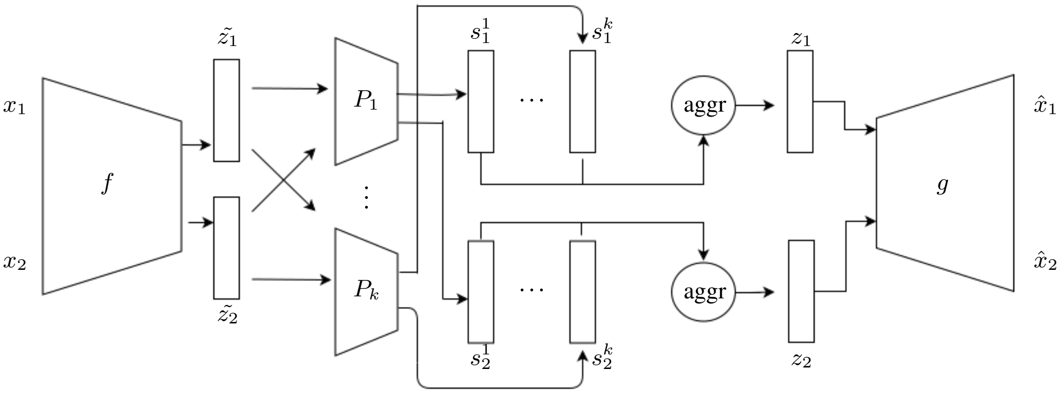

Figure 2: The architecture of our model. We process data in pairs (x_1, x_2), which are embedded into an intermediate lower dimensional space $\tilde{\mathcal{Z}}$ via a siamese network f. The image (z_1, z_2) is then mapped into k smaller spaces $\mathcal{S}_1, \dots, \mathcal{S}_k \subset \mathcal{Z}$ via the nonlinear operators P_i. The resulting vectors are aggregated in $\mathcal{Z}$, with $aggr = +$, and mapped back to the input data space by the decoder g. As we do not impose any constraint on f and g, the intermediate module of the proposed architecture can be in principle attached to any autoencoder model.

Parameter	Value
d	10
k	10
β_1	0.1
β_2	100
β_3	0.0001
Batch Size	32
Optimizer	Adam
Learning rate	0.0005
Adam: (beta1, beta2, epsilon)	(0.9,0.99,1e-8)

Table 4: Hyper-parameter settings for the experiments in Table 1-4 of the main paper.

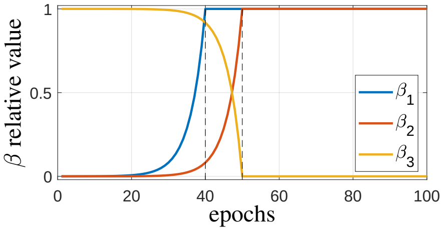

Figure 1: Evolution of the regularization parameters $\beta_i, i = 1, 2, 3$ as a function of the epoch number. Here, the parameters are all scaled to have a maximal value of one.

2. Training process

We split the training process in two stages: (i) a *reconstruction phase*, and (ii) a *disentanglement phase*. This strategy helps in obtaining better results; this is due to the fact that our distance loss $\mathcal{L}_{dis}$ needs to operate in a latent space $\mathcal{Z}$ already structured, where the distances are meaningful. Moreover, our consistency loss makes use of the reconstructed observations, that have to be well formed to make it relevant. We stress that the two phases are not completely separated, since the space $\mathcal{Z}$ continues to be optimized during the disentanglement phase.

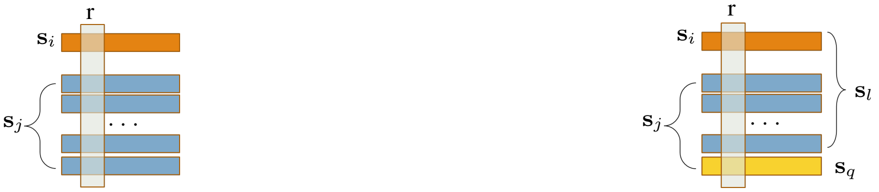

Figure 3: A visualization of the main variables involved in our proof.

In practice, we implement this by back-propagating only through the reconstruction loss for the first 20% of the training iterations. Then, the losses enter one after the other in the following order: $\mathcal{L}_{reg}, \mathcal{L}_{dis}, \mathcal{L}_{cons}$ in a slow-start mode. This is obtained by exponentially increasing the regularization parameters β_i for $i = 1, 2$ during the training, until they reach their maximal value (as reported in Table 4), with β_2 being shifted in time (number of iterations/epochs) with respect to β_1 . Conversely, we set $\beta_3 = (1 - \beta_2)$, so it exponentially decays until it reaches zero; indeed, the regularization loss prevents the subspace from collapsing until the other losses are active at full capacity. We show an example of the behavior of the β's in Figure 1.

3. Subspace structure

3.1. The latent subspace structure

The model architecture, shown in Figure 3 of the main paper and reported also here in Figure 2 for convenience, imposes a factorized structure on the latent space $\tilde{\mathcal{Z}}$ into subspaces $\mathcal{S}_i, i = 1..k$. In principle, the aggregator function depicted could be any linear or nonlinear aggregation operation. In our experiments we simply choose to *sum* all the subspaces, for the following reason: due to the sparsity induced on the subspaces by the loss $\mathcal{L}_{spar}$, the sum operation provides us with an approximation of the cartesian product, leading to $\tilde{\mathcal{Z}} \approx \mathcal{S}_1 \times .. \times \mathcal{S}_k$. More precisely, if the sparsity contraint holds (*i.e.* $\mathcal{L}_{spar} = 0$), the sum operation will be equivalent to taking the cartesian product on the latent subspace vectors, since on each dimension $r \in 1 \dots d$ such that $\mathbf{s}_i[r] \neq 0$, for an $i \ \in 1, \dots, k$ the loss $\mathcal{L}_{spar}$ enforces the latent vectors to have $\mathbf{s}_q[r] = 0 \ \ \forall q \neq i \ \in 1, \dots, k$. We prove this in the following:

Sketch of proof. We prove that the sparsity imposes the structure of a product space on the latent subspace vectors. We do this by studying the first order optimality conditions for $\mathcal{L}_{spar}$, $\frac{\partial \mathcal{L}_{spar}(\mathbf{s}_i)}{\partial \mathbf{s}_i} = 0$, where with $\mathbf{s}_i = P_i(f(x))$ we denote a latent vector in the subspace $\mathcal{S}_i$. Indicating with $\odot$ the element-wise product, we can write:

$$\mathcal{L}_{spar} = \sum_{i=1}^{k} \mathcal{L}_{spar}^{\mathbf{s}_i} = \sum_{i=1}^{k} \|\mathbf{s}_i \odot \sum_{j\neq i}^{k} \mathbf{s}_j\|_1 \,, \quad \text{where } \mathcal{L}_{spar}^{\mathbf{s}_i} = \| \underbrace{\mathbf{s}_i \odot \sum_{j\neq i}^{k} \mathbf{s}_j}_{Q} \|_1 \,. \tag{1}$$

We aim to study $\frac{\partial \mathcal{L}_{spar}^{\mathbf{s}_i}}{\partial \mathbf{s}_i} = 0, \forall i \in 1, \dots, k$ that is equivalent to:

$$\frac{\partial \mathcal{L}_{spar}^{\mathbf{s}_i}}{\partial \mathbf{s}_i} = \frac{\partial \|Q\|_1}{\partial Q} \frac{\partial Q}{\partial \mathbf{s}_i} = \Big(sign(\mathbf{s}_i \odot \sum_{j\neq i}^{k} \mathbf{s}_j) \Big) (\sum_{j\neq i}^{k} \mathbf{s}_j) = 0, \ \forall i \in 1, \dots, k \,. \tag{2}$$

W.l.o.g. we fix a dimension $r \in 1, \dots, d$ in the latent space. By indicating with $\mathbf{s}_i[r]$ the r-th entry of $\mathbf{s}_i$ we can write:

$$\frac{\partial \mathcal{L}_{spar}^{\mathbf{s}_i}}{\partial \mathbf{s}_i}[r] = \Big(sign(\mathbf{s}_i[r] \sum_{j\neq i}^{k} \mathbf{s}_j[r]) \Big) (\sum_{j\neq i}^{k} \mathbf{s}_j[r]) = 0 \,. \tag{3}$$

To satisfy Eq. (3), we have three possible cases:

- *Case 1:* $\mathbf{s}_i[r] = 0$ and $\sum_{j\neq i}^{k} \mathbf{s}_j[r] \neq 0$

- *Case 2:* $\mathbf{s}_i[r] \neq 0$ and $\sum_{j\neq i}^{k} \mathbf{s}_j[r] = 0$

  Since we are optimizing $\forall i$ we can consider $\frac{\partial \mathcal{L}_{spar}^{\mathbf{s}_q}}{\partial \mathbf{s}_q}[r] = 0$ for every other $q \in 1 \dots k, q \neq i$, Therefore we have:

$$\frac{\partial \mathcal{L}_{spar}^{\mathbf{s}_q}}{\partial \mathbf{s}_q}[r] = \Big(sign(\mathbf{s}_q[r] \sum_{l\neq q}^{k} \mathbf{s}_l[r])\Big)(\sum_{l\neq q}^{k} \mathbf{s}_l[r]) = 0 \tag{4}$$

  We can split this latter case in two subcases:

  - *Case 2.1:* $\mathbf{s}_q[r] = 0, \forall q \neq i$
    Therefore, satisfying our thesis.
  - *Case 2.2:* $\mathbf{s}_q[r] \neq 0$ for at least one $q \neq i$.
    In this situation, we can write:

$$\sum_{j'\neq q,i}^{k} \mathbf{s}_{j'}[r] + \mathbf{s}_q[r] = 0 \text{ and thus } \sum_{j'\neq q,i}^{k} \mathbf{s}_{j'}[r] = -\mathbf{s}_q[r]. \tag{5}$$

    From which we have:

$$\sum_{j'\neq q}^{k} \mathbf{s}_{j'}[r] = \mathbf{s}_i[r] + \sum_{j'\neq q,i}^{k} \mathbf{s}_{j'}[r] = \mathbf{s}_i[r] - \mathbf{s}_q[r]. \tag{6}$$

    Substituting in Eq.4 (by replacing j' with l) we get:

$$\frac{\partial \mathcal{L}_{spar}^{\mathbf{s}_q}}{\partial \mathbf{s}_q}[r] = \Big(sign(\mathbf{s}_q[r](\mathbf{s}_i[r] - \mathbf{s}_q[r]))\Big)(\mathbf{s}_i[r] - \mathbf{s}_q[r]) \tag{7}$$

    Now because we have that $\mathbf{s}_q[r] \neq 0$, this implies:

$$\frac{\partial \mathcal{L}_{spar}^{\mathbf{s}_q}}{\partial \mathbf{s}_q}[r] = 0 \iff \mathbf{s}_i[r] - \mathbf{s}_q[r] = 0 \implies \mathbf{s}_i[r] = \mathbf{s}_q[r] \tag{8}$$

    and this holds $\forall q \in 1, \dots, k$ and $q \neq i$ such that $\mathbf{s}_q[r] \neq 0$. (referring to Figure 3 may help the reader).
    This allows us to conclude that: $\sum_{j\neq i}^{k} \mathbf{s}_j[r] = \alpha \mathbf{s}_i[r] \neq 0$, with α being an integer between 1 and $k-1$. Therefore we get a contradiction with our hypothesis of *Case 2* $\mathbf{s}_i[r] \neq 0$ and $\sum_{j\neq i}^{k} \mathbf{s}_j[r] = 0$, and thus the unique possible subcase is the former *Case 2.1*.

- *Case 3:* $\mathbf{s}_i[r] = 0$ and $\sum_{j\neq i}^{k} \mathbf{s}_j[r] = 0$
  Performing the same analysis done in *Case 2*, in this case we get that $\forall i \in 1 \dots k$ $\mathbf{s}_i[r] = 0$. Therefore, all the latent subspace vectors will have the same dimension r set to zero. In this case, we can consider recursively the other $k-1$ dimensions. The case where all dimensions $r \in 1 \dots k$ are zero, for all $\mathbf{s}_i, i = 1 \dots k$ is theoretically possible, but we stress this is rather an exotic case that cannot happen in practice, as we comment in the last paragraph below.

  Since we have chosen r w.l.o.g., the same s true for all dimensions in $1 \dots d$. Therefore, we have that each vector $\mathbf{s}_i$ will be nonzero in the $l > 0$ dimensions where the other $\mathbf{s}_j$ are zero. Now setting $aggr = +$, we have that the sum corresponds to concatenating the latent subspace vectors along the nonzero dimensions, i.e. taking the cartesian product of the subspace to get an element of $\tilde{\mathcal{Z}}$.

Degenerate case In the proof we mentioned the degenerate case in which $\mathbf{s}_i[r] = 0$ $\forall i \in 1 \dots k,$ $\forall r \in 1 \dots d$. This would mean that the latent subspaces have collapsed to the same point (a vector made of zeros). This exotic case is never reached in practice, due to the other losses such as the reconstruction loss, the consistency losses, and the contrastive term of the distance loss.

References

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Kim, H. and Mnih, A. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2654–2663. PMLR, 2018.

LeCun, Y., Fu Jie Huang, and Bottou, L. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pp. II–104 Vol.2, 2004.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6348–6359. PMLR, 2020.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 77–85. IEEE Computer Society, 2017.

Reed, S. E., Zhang, Y., Zhang, Y., and Lee, H. Deep visual analogy-making. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 1252–1260, 2015.