On the relationship between disentanglement and multi-task learning

Łukasz Maziarka*[0000−0001−6947−8131], Aleksandra Nowak*[0000−0002−2830−6613], Maciej Wołczyk*[0000−0002−3933−9971], and Andrzej Bedychaj*[0000−0001−9416−3991]

Jagiellonian University
{name.surname}@ii.uj.edu.pl

Abstract. One of the main arguments behind studying disentangled representations is the assumption that they can be easily reused in different tasks. At the same time finding a joint, adaptable representation of data is one of the key challenges in the multi-task learning setting. In this paper, we take a closer look at the relationship between disentanglement and multi-task learning based on hard parameter sharing. We perform a thorough empirical study of the representations obtained by neural networks trained on automatically generated supervised tasks. Using a set of standard metrics we show that disentanglement appears naturally during the process of multi-task neural network training.

Keywords: Multitask learning · Disentangled representation.

1 Introduction

Disentangled representations have recently become an important topic in the deep learning community [12,26,29,35,10]. The main assumption in this problem is that the data encountered in the real world is generated by few independent and explanatory factors of variation. It is commonly accepted that such representations are not only more interpretable and robust but also perform better in tasks related to transfer learning and one-shot learning [3,23,37,28].

Intuitively, a disentangled representation encompasses all the factors of variation and as such can be used for various tasks based on the same input space. On the other hand, non-disentangled representations, such as those learned by vanilla neural networks, might focus only on one or a few factors of variations that are relevant for the current task, while discarding the rest. Such a representation may fail when encountering different tasks that rely on distant aspects of variation which have not been captured.

Exploiting prevalent features and differences across tasks is also the paradigm of multi-task learning. In a standard formulation of a multi-task setting, a model is given one input and has to return predictions for multiple tasks at once. The neural network might be therefore implicitly regularized to capture more

* All authors contributed equally

factors of variation than a network that learns only a single task. Based on this intuition, we hypothesize that disentanglement is likely to occur in the latent representations in this type of problem.

This paper aims to test this hypothesis empirically. We investigate whether the use of disentangled representations improves the performance of a multi-task neural network and whether disentanglement itself is achieved naturally during the training process in such a setting.

Our key contributions are:

- Construction of synthetic datasets that allow studying the relationship between multi-task and disentanglement learning.
- Study of the effect of multi-task learning with hard parameter sharing on the level of disentanglement obtained in the latent representation of the model.
- Analysis of the informativeness of the latent representation obtained in the single- and multi-task training.
- Inspection of the effect of disentangled representations on the performance of a multi-task model.

We verify our hypotheses by training multiple models in single- and multi-task settings and investigating the level of disentanglement achieved in their latent representations. In our experiments, we find that in a hard-parameter sharing scenario multi-task learning indeed seems to encourage disentanglement. However, it is inconclusive whether disentangled representations have a clear positive impact on the models performance, as the obtained by us results in this matter vary for different datasets.

Code for our experiments is available at:
`https://github.com/gmum/disentanglement-multitask`.

2 Related Work

2.1 Disentanglement

Over the recent years, many methods that directly encourage disentanglement have been proposed. This includes algorithms based on variational and Wasserstein auto-encoders [19,13,21,4,39], flow networks [9,38] or generative adversarial networks [8]. The main interest behind disentanglement learning lays in the assumption that such transformation unravels the semantically meaningful factors of variation present in the observations and thus it is desired in training deep learning models. In particular, disentanglement is believed to allow for informative compression of the data that results in a structural, interpretable representation, which is easily adaptable for new tasks [3,23,36,24].

Several of these properties have been experimentally proven in applications in many domains, including video processing tasks [15], recommendation systems [29] or abstract reasoning [41,40]. Moreover, recent research in reinforcement learning concludes that disentangling embeddings of skills allows for faster retraining and better generalization [33]. Finally, disentanglement seems also to be

positively correlated with fairness when sensitive variables are not observed [26]. On the other hand, some empirical studies suggest that one should be cautious while interpreting the properties of disentangled representations. For instance, the latest studies in the unsupervised learning domain point that increased disentanglement does not lead to a decreased sample complexity in downstream tasks [27].

Another key challenge in studying disentangled representations is the fact that measuring the quality of the disentanglement is a nontrivial task [10,12,19], especially in a unsupervised setting [27]. This motivates the research on practical advantages of disentanglement representations and their impact on the studied problem in possible future applications, which is the main focus of our work in the case of multi-task learning.

2.2 Multi-task Learning

Multi-task learning aims at simultaneously solving multiple tasks by exploiting common information [34]. The approaches used predominantly to this problem are soft [11] and hard [6] parameter sharing. In hard parameter sharing the weights of the model are divided into those shared by all tasks, and task-specific. In deep learning, this idea is typically implemented by sharing consecutive layers of the network, which are responsible for learning a joint data representation. In soft parameter sharing each task is given a set of separate parameters. The limitations are then imposed by information-sharing or regularizing the distance between the parameters by adding an applicable loss to the optimization objective.

Multi-task learning is widely used in the Deep Learning community, for instance in applications related to natural language processing [25,30], computer vision [32] or molecular property prediction modeled by graph neural networks [5]. One may observe that the premises of multi-task and disentanglement learning are related to each other and thus it is interesting to investigate whether the joint data representation obtained in a multi-task problem exhibits some disentanglement-related properties.

3 Methods

In this section, we describe the methods and datasets used for conducting the experiments.

3.1 Dataset Creation

In order to investigate the relationship between multi-task learning and disentanglement, we require a dataset that fulfills two conditions:

1. It provides access to the true (disentangled) generative factors z from which the observations x are created.

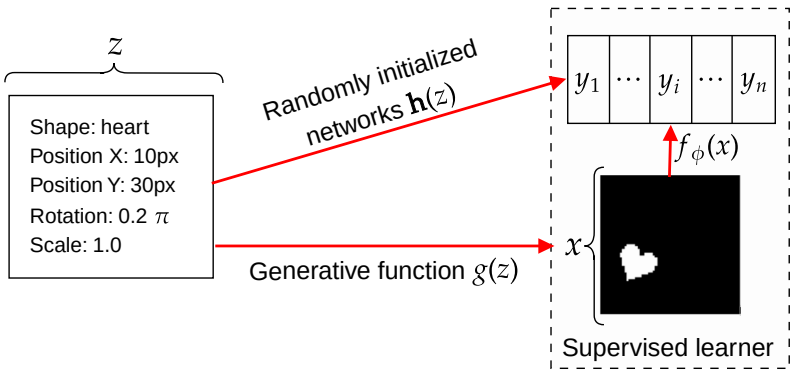


Fig. 1: The setting of our experiments. Given a dataset of pairs (x, z) of observations and their true generative factors, we generate a set of functions $\mathbf{h}(z)_i$ which are aimed to approximate real-world supervised tasks. Then, we train a neural network $f_\phi(x)$ in a multi-task regression setting on pairs $(x, \mathbf{h}(z))$. After the training, we investigate the hidden representations learned by f_ϕ and explore their relation to true factors z.

2. It proposes multiple tasks for a supervised learner by providing labels y_i which non-linearly depend on the true factors z.

The first condition is required in order to measure how well the learned representations approximate the true latent factors z. Access to the true factors allows for full control over the experimental settings and permits a fair comparison through the use of supervised disentanglement metrics. Note that even though unsupervised metrics have been proposed in the literature as well, they typically yield less reliable results, as we further discuss in section 3.3.

The second condition is needed to train a network on multiple nontrivial tasks to approximate the real-world setting of multi-task learning.

To our best knowledge, no nontrivial datasets exist that would abide by both those requirements. Most of the available disentanglement datasets, such as dSprites, Shapes3D, and MPI3D do fulfill the first condition, as they provide pairs (x, z) of observations and their true generative factors. However, those datasets do not offer any type of challenging task on which our model could be trained. On the other hand, many datasets used for supervised multi-task learning fulfill the second condition by providing pairs (x, y), but do not equip the researcher with the latent factors z (ground truth), failing the first condition.

Thus, we aim to create our own datasets which fulfill both conditions by incorporating nontrivial tasks into standard disentanglement datasets. Since in multi-task approaches one often tries to solve tens of tasks at once, designing them by hand is infeasible and as such we decide to generate them automatically in a principled way. In particular, since supervised learning tasks might be

formalized as finding a good approximation to an unknown function $h(x)$ given a set of points $(x, h(x))$, we generate random functions $h(z)$ which are then used to obtain targets for our dataset (see Figure 1).

We require $h(z)$ to be both nontrivial (i.e. non-linear and non-convex) and sufficiently smooth to approximate the nature of real-life tasks. In order to find a family of functions that fulfills those conditions, we take inspiration from the field of extreme learning, which finds that features obtained from randomly initialized neural networks are useful for training linear models on various real-world problems [16]. As such, randomly initialized networks should be able to approximate these tasks up to a linear operation.

In particular, in order to generate the dataset, we define a neural network architecture $h(z, \theta)$. For this purpose, we used an MLP with four hidden layers with 300 units, tanh activations, and an output layer which returns a single number. Then we sample n weight initializations of this network from the Gaussian distribution $\theta_i \sim \mathcal{N}(0, 1)$, where $i \in \{1, \ldots, n\}$. Each of the networks $h(z, \theta_i)$ obtained by random initialization defines a single task in our approach. Thus, for a given dataset $\mathcal{D} = (x, z)$ containing observations and their true generative factors, we obtain a dataset for multi-task supervised learning by applying:

$$\tilde{\mathcal{D}} = \{(x, \mathbf{h}(z)) \mid (x, z) \in \mathcal{D}\} = \{(x, y)\},$$

where $\mathbf{h}(z)$ is a vector of stacked target values for each task, whose element i is given by $\mathbf{h}(z)_i = h(z, \theta_i)$.

We use this data as a regression task, i.e. for a given neural network f_ϕ parameterized by ϕ the goal is to find:

$$\arg\min_{\phi} \sum_{(x,y)\in\tilde{\mathcal{D}}} \|f_\phi(x) - y\|_2^2.$$

We use this process to create multi-task supervised versions of dSprites, Shapes3D, and MPI3D, with 10 tasks for each dataset.

3.2 Models

Multi-task model We investigate the relation between disentanglement and multi-task learning based on a hard parameter sharing approach. In this setting, several consecutive hidden layers of the model are shared across all tasks in order to produce a joint data representation. This representation is then propagated to separate task-specific layers which are responsible for computing the final predictions.

In particular, we use a network consisting of a shared convolutional encoder and separate fully-connected heads for each of the tasks. The encoder learns the joint representation by transforming the inputs into a d-dimensional latent space. [1] The heads are implemented by 4-layer MLPs with ReLU activations, in order

[1] We provide the full model summary in **Appendix ??**. The architecture of the encoder follows the one from [1], which adopts the work of [27] for the `pytorch` package. We use the implementations from `https://github.com/amir-abdi/disentanglement-pytorch`.

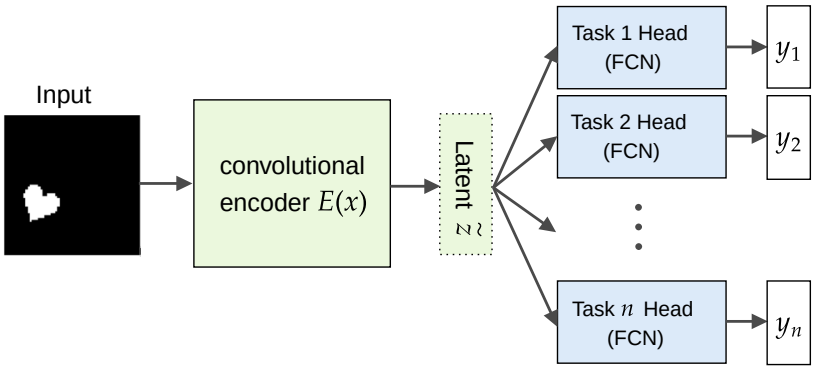


Fig. 2: The model used for multi-task training. The convolutional encoder $E(x)$ transforms the input data x to a latent representation $\tilde{z}$. The parameters of the encoder are shared across all tasks. Next, the produced representation is passed to the task-specific heads, which are implemented by fully-connected networks (FCN).

to match the capacity of the networks used for task generating functions $h_i(x)$. This overview of the model is illustrated in Figure 2.

Auto-encoder model In the second part of our experiments we want to understand if disentangled representation provides some benefits for the multi-task problem. In order to produce disentangled representations, we decided to use three different representation-learning algorithms: a vanilla auto-encoder, the (beta)-variational auto-encoder [20,13] and FactorVAE [19].

All these variants of the auto-encoder architecture encompass a similar framework. An auto-encoder imposes a bottleneck in the network which forces a compressed knowledge representation of the original input. In some variants of those models, we additionally try to constrain the latent variables to be highly informative and independent which further correlates to disentanglement, e.g. in models like β-VAE and FactorVAE. We use latent representations from these models to train task-specific heads and evaluate if disentanglement helped to decrease an error for that task.

The vanilla auto-encoder is also used in Section 4.2, where we add a decoder with transposed convolutions to pre-trained encoders from Section 4.1. This treatment is aimed to decode information for particular encoders in the most efficient way. As such, we find auto-encoders to be a useful tool for investigating disentanglement.

3.3 Disentanglement Metrics

Measuring the qualitative and quantitative properties of the disentanglement representation discovered by the model is a nontrivial task. Due to the fact that the true generating factors of a given dataset are usually unknown, one may assume that decomposition can be obtained only to some extent.

Commonly used unsupervised metrics are based on correlation coefficients which measure the intrinsic dependencies between the latent components. Such measures are widely used in the independent component analysis [17,18,14,4,39,2]. However, uncorrelatedness does not imply stochastical independence. Furthermore, metrics based on linear correlations may not be able to capture higher-order dependencies and are often ineffective in large dimensional or in over-determined spaces. All this makes the use of such unsupervised metrics questionable.

An alternative solution would be to use supervised metrics, which usually are more reliable [27]. This is of course only possible after assuming access to the true generative factors. Such an assumption is rarely valid for real-world datasets, however, it is satisfied for synthetic datasets. Synthetic datasets present therefore a reasonable baseline for benchmarking disentanglement algorithms.

Frequently used metrics which use supervision are mutual information gap (MIG) [7], the FactorVAE metric [19], Separated Attribute Predictability (SAP) score [22] and disenanglement-completness-informativeness (DCI) [12]. In order to comprehensively assess the level of disentanglement in our experiments, we have decided to use all of the above-mentioned metrics to validate our results. A more detailed description of those metrics is available in **Appendix ??**.

4 Results and Discussion

In this section, we describe the performed experiments and discuss the obtained results. For more details on the training regime and experimental setup please refer to **Appendix C**.

4.1 Does Hard Parameter Sharing Encourage Disentanglement?

One of the most common approaches to multi-task learning is hard parameter sharing. The key challenge in this method is to learn a joint representation of the data which is at the same time informative about the input and can be easily processed in more than one task. It is therefore tempting to verify whether disentanglement arises in those representations implicitly, as a consequence of hard parameter sharing.

In order to investigate this problem we build a simple multi-task model described in Section 3.2 and evaluate it on the three datasets discussed in Section 3.1: dSprites, Shapes3D, and MPI3D, each with 10 artificial tasks. After the training is complete, we calculate each of the disentanglement metrics described

in Section 3.3 on the latent representation of the input data[2]. We compare the obtained results with the same metrics computed for an untrained (randomly initialized) network and for single-task models. In all the cases we use the same architecture and training regime. Note that in the single-model scenario we train a separate model for each of the 10 tasks, which is implemented by utilizing only one, dedicated head in the optimization process. We train all models three times, using a different random seed in the parameters initialization procedure. We report the mean results and standard deviations in Figure 3.

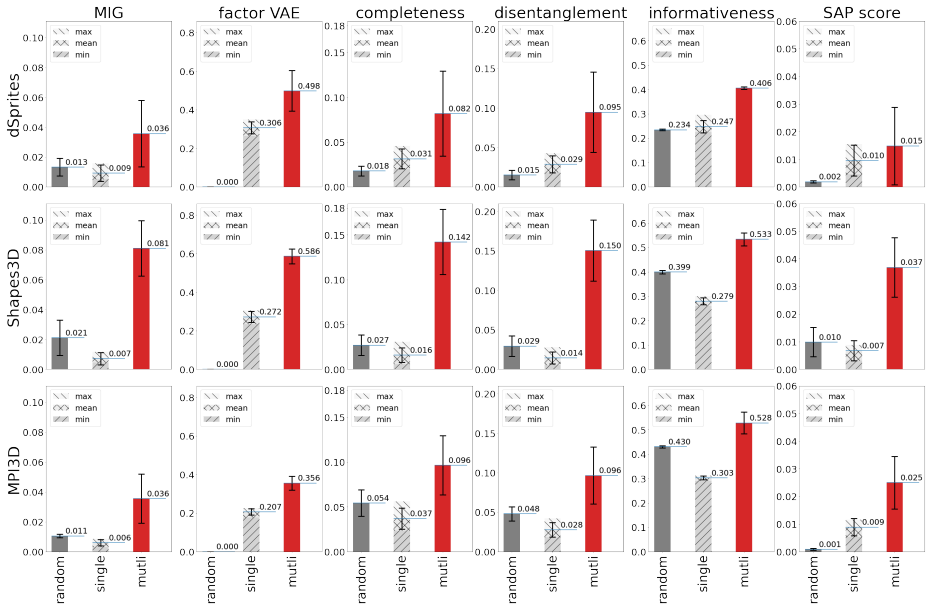


Fig. 3: Different disentanglement metrics computed for random (untrained), single-task and multi-task models evaluated on the three datasets described in Section 3.1. The higher the value the better. For the single-task scenario, we report the mean over all task-specific models. Note that in almost every case the multi-task representations (red bars) outperform the random or single-task representations (dark-gray bars and light gray bars, respectively). Additionally, for single-task models, we report the maximal and minimal values over all tasks to show that the performance on multi-task does not rely on any single 'lucky' task. For tabulated results please refer to **Appendix ??**.

We observe that disentanglement metrics computed for the representations obtained in the multi-task setting are typically significantly better than the values obtained for single-task or random representations. Note that even the

[2] We use the implementations of [27], which are available at `https://github.com/google-research/disentanglement_lib`

maximum mean result over all ten single-task models is in almost every case further than one standard deviation from the multitask mean. Moreover, this is true for all the tested datasets.

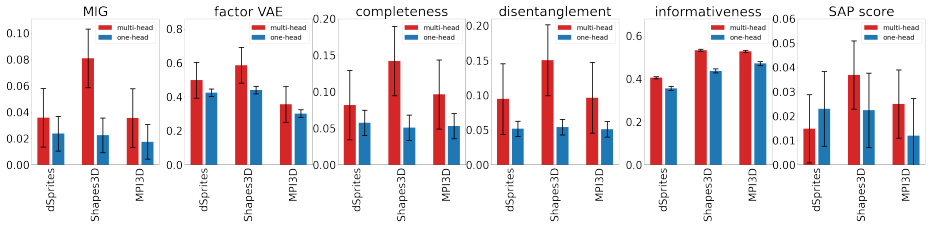


Fig. 4: Different disentanglement metrics computed for the multi-task setting with one head shared between all tasks (one-head) and separate head for every task (multi-head), evaluated on the three datasets described in Section 3.1. The higher the value the better. One may observe that multi-head representations perform better than the ones obtained in the standard, one-head multivariate regression task. For tabulated results please refer to **Appendix ??**.

Let us also point out that instead of using separate heads for each of the tasks in the multi-task model one could simply use one head with the output dimension equal to the number of tasks and perform standard multivariate regression (with no parameter sharing). As presented in Figure 4, the latent representations emerging in such a scenario are less disentangled (in terms of the considered metrics) than the representations obtained when utilizing hard parameter sharing. However, the achieved values are still better than in single-task models. This suggests that even though the increase in the metrics may be partially caused by simply training the network on higher-dimensional targets, the positive influence of hard parameter sharing cannot be ignored. This advocates in favor of the hypothesis that multi-task representations are indeed more disentangled than the ones arising in single-task learning.

4.2 What Are the Properties of the Learned Representations?

The previous section discussed the obtained representations by analyzing quantitative disentanglement metrics. Here, we provide more insights into the characteristics of latent encodings.

UMAP embeddings In order to gain intuition behind the differences between the representations obtained in the previous experiment we compute a 2D-embedding of the latent encodings using the UMAP algorithm [31]. The results are presented in Figure 5.

The embeddings obtained for the multi-task representations are much more semantically meaningful, with easily distinguishable separate clusters. Moreover,

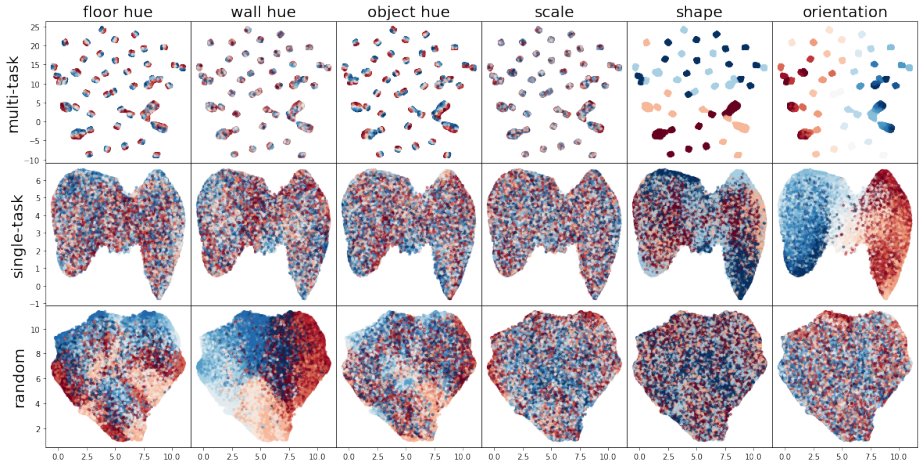


Fig. 5: UMAP embeddings of the latent representations of the Shapes3D test dataset obtained for different models. Change of the color within one subplot presents the change in one particular ground truth component. The embeddings obtained by the multi-task model seem to be most semantically meaningful. See **Appendix ??** for plots for other datasets.

the position and internal structure of the clusters correspond to different values of the true factors. This cannot be observed for the untrained or single-task representations, suggesting that the multi-task representations are indeed more successful in encompassing the information about the real values of the generative sources of the data.

Latent space traversal Providing qualitative results of the retrieved factors is a common practice in disentanglement learning [28,21,35,38,27]. In particular, visual presentation of the interpolations over the latent space allows assessing — from a human perspective — the informativeness and decomposition of the obtained representations. Note that such analysis is possible only after adding and training a suitable decoder network, which maps the retrieved factors back to the image space.

In our setting, the decoder mirrors the architecture of the encoder (the convolutions are replaced by transposed convolutions of the same size — see **Appendix ??**). Given the latent representations as an input, the decoder optimizes the reconstruction error (as measured by MSE) between its outputs and the original images. We train three separate decoders corresponding to the different encoders from the previous section — a randomly initialized encoder, an encoder produced by one of the single-task models, and a multi-task encoder.

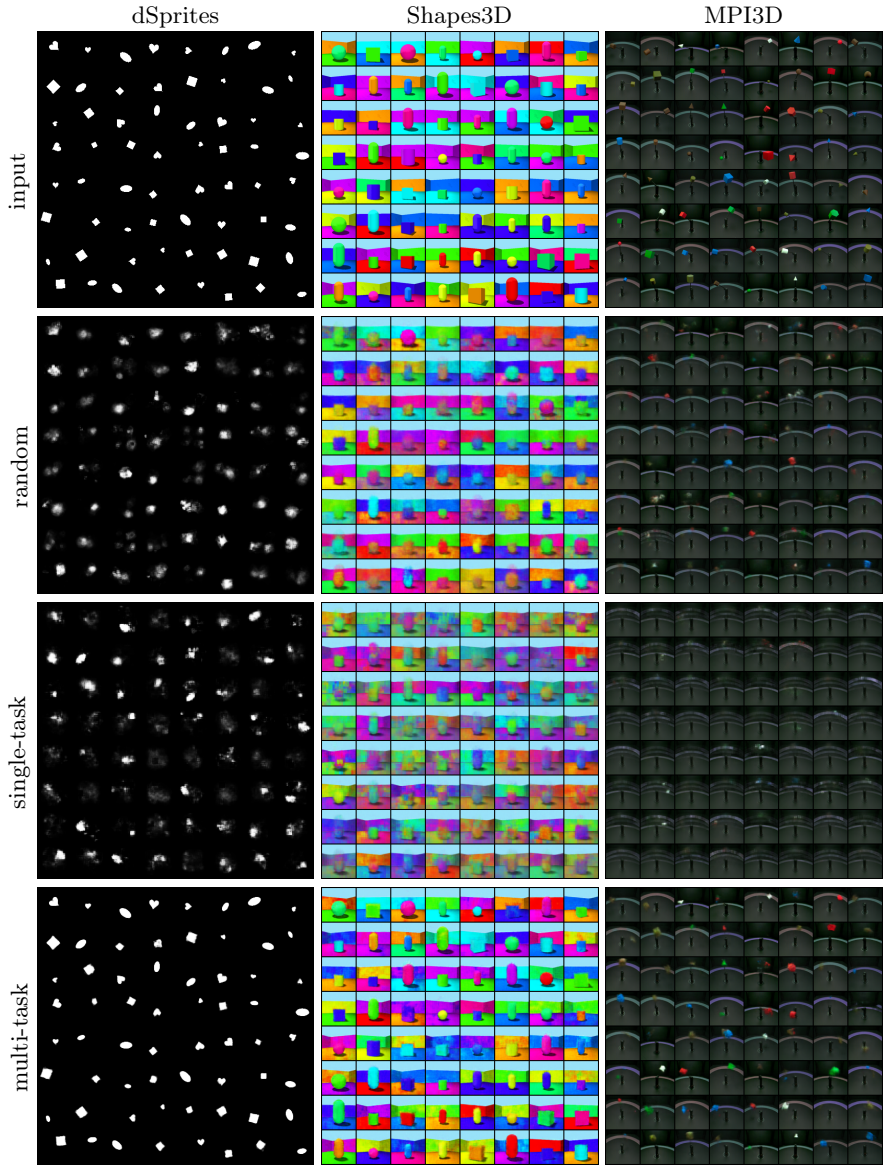
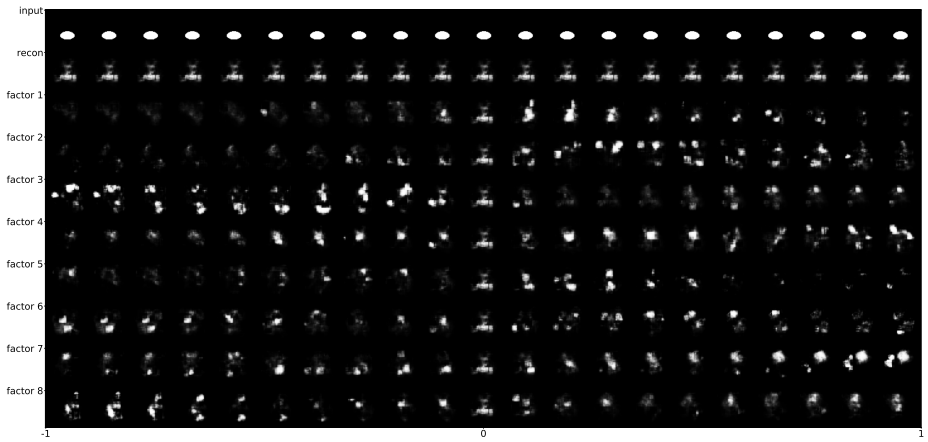
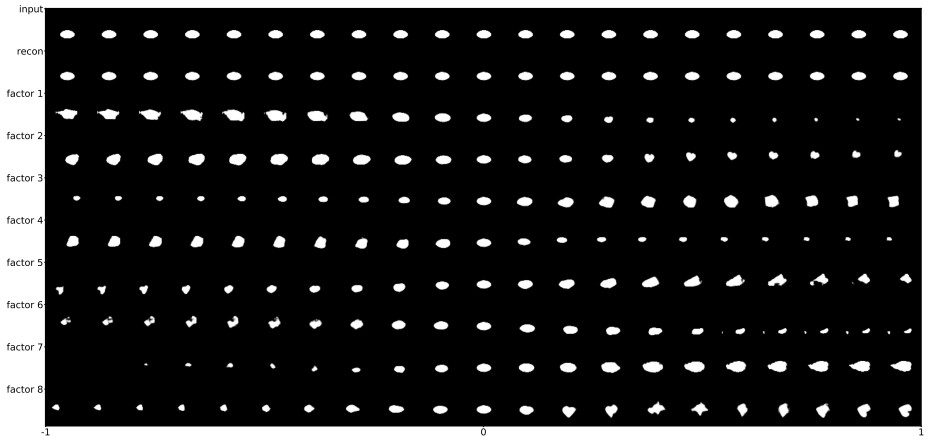


Fig. 6: Reconstructions obtained by the decoders trained on random, single-task, and multi-task encoding. For reference, we provide the original input images in the first row. The quality of the reconstruction for the random and single-task representation is very poor. Contrary, the multi-task encoder provided a latent space that can be successfully decoded into images that closely resemble the corresponding examples from the input. Thus, we conclude that the multi-task representations are more informative about the data and provide better compression.

(a) Random encoder



(b) Single task encoder



(c) Multi-task encoder

Fig. 7: Traverses over latent variable produced for a given architecture. The same example was used in all three traverses. The second row of each image shows how the decoder reconstructed this example in a particular setting. The rest of the factors come from latent space generated from each encoder. Visualization of components from the multi-task encoder are sharp and distinguish the generating factors distinctly. The same cannot be said about the latent factors in single-task and random encoders, which are blurry and disconnected from any interpretable ground truth factors. Please refer to **Appendix ??** for the results of the traversals over other datasets.

First, let us discuss the reconstruction quality achieved by each of the tested decoders. Results of this experiment are presented in Figure 6[3]. Reconstructions produced for the multi-task encodings are clearly superior to the ones obtained for the single-task encodings. In the first case, the resulting images are sharp and contain almost no noise. In contrast, the single task reconstructions are blurry and similar to the ones produced for the randomly initialized encoder. We would like to emphasise that all the decoders used the same architecture and that during their optimization the parameters of the corresponding encoders were kept fixed. Therefore the quality of the reconstruction is an important property of a latent representation, as it allows us to assess the compression capacity of the representation. From this perspective, the compression obtained in the multi-task scenario is much more informative about the input than in the single-task scenario.

Another approach to the visualisation of the latent variables is to perform interpolations (traversals) in the latent space. We start by selecting a random sample from the dataset and compute its encoding $\tilde{z} \in \mathbb{R}^d$. By modifying one of the components of vector $\tilde{z}$ from -1 to 1 with 0.1 step and leaving the $d-1$ unchanged, we produce a traversal along that particular factor. We repeat this procedure for all the factors in order to capture their impact on the decoded example. Results of such traverses for the dSprites dataset are shown in Figure 7.

Note that since the models were not trained directly for disentanglement but only to solve a supervision task, it is not surprising that the representations are not as clearly factorized as in specialized methods such as FactorVAE. However, for the multi-task model, certain latent dimensions still appear to be disentangled and one can easily spot the difference in quality between the single and multi-task representations. In the multi-task traversals, we can notice components that are responsible for the position and scale of a given figure (in Figure 7c, consider the 5th and 7th factors, respectively). In contrast, the results for single task representations demonstrate that even a slight change in any of the single latent dimensions leads to a degradation of the reconstructed examples. As expected, this effect is even more evident for the random (untrained) representations, where the corruption over latent factor is even more prevalent than in the case of a single-task traversal.

4.3 Does Disentanglement Help in Training Multi-task Models?

In the previous sections, we studied whether multi-task learning encourages disentanglement. Here we consider an inverse problem by asking whether using disentangled representation helps in multi-task learning. To investigate this issue, we train an auto-encoder-based model devised specifically to produce disentangled latent representations without access to the true latent factors. Next, we freeze its parameters and use the encoder function to transform the inputs. The obtained latent encodings are then passed directly to the heads of a multi-task

[3] Numerical values for reconstruction errors are presented in **Appendix ??.**

network which minimizes the average regression loss given the target values of the artificial tasks.

We consider three different auto-encoder-based algorithms described in Section 3.2: a vanilla auto-encoder (AE), a variational auto-encoder (VAE), and the FactorVAE. The vanilla auto-encoder does not directly enforce latent disentanglement during the training. In the VAE model, the prior normal distribution with identity covariance matrix implies some disentanglement. Finally, FactorVAE introduces a new module to the VAE architecture that explicitly induces informative decomposition. Therefore, the representations obtained for each subsequent model should be also naturally ordered by the level of the achieved disentanglement. For the exact values of the calculated metrics please refer to **Appendix F**. In addition, we also study a scenario in which we explicitly provide the true source factors. We trained all regression models three times, using a different random seed in the parameters initialization procedure.

Table 1: RMSE of multi-task networks trained on latent representations obtained by different auto-encoder-based methods. For comparison, we added the model trained on ground truth factors. The best results are bolded, and best out of auto-encoder architectures underlined.

Dataset	dSprites	Shapes3D	MPI3D
Ground Truth	150.235 ± 3.754	**72.979 ± 0.193**	**108.568 ± 0.285**
AE	80.062 ± 0.341	114.939 ± 0.160	150.190 ± 0.097
VAE	**63.260 ± 0.260**	132.072 ± 0.169	194.865 ± 15.61
FactorVAE	91.937 ± 0.199	118.396 ± 0.423	151.646 ± 0.336

Table 1 summarizes the performance of the multi-task model trained on the representations obtained for the above-discussed methods. Although the representations obtained from FactorVAE are better (see, for instance, MIG or DCI measures in **Appendix F**) than those from VAE and AE, the encodings produced by the vanilla AE are the best among the tested, exceeding the others on Shapes3D and MPI3D and being second on dSprites. Note that these results coincide with observations presented in the literature. For example, [27] compared different models that enforce disentanglement during the training and showed that even a high value of that property within the factors do not constitute a better model performance. However, in two out of three datasets, the use of the ground true factors seems to significantly improve the obtained results. This may suggest that the representations produced by the considered disentanglement methods are not fully factorized. It is therefore inconclusive whether the discrepancy between the obtained results is due to the shortcomings of the used methods or a manifestation of the impracticality of disentanglement.

5 Conclusions

In this paper, we studied the relationship between multi-task and disentanglement representation learning. A fair evaluation of our hypothesis is impossible on real-world datasets, without provided ground truth factors. To evaluate our results we had to introduce synthetic datasets that contain all necessary properties to be seen as a benchmark in this field. Next, we studied the effects of multi-task learning with hard parameter sharing on representation learning. We found that nontrivial disentanglement appears in the representations learned in a multi-task setting. Obtained factors have intuitive interpretations and correspond to the actual ground truth components. Finally, we inverted the question and investigated the hypothesis that disentangled representation is needed for multi-task learning, the results however are not conclusive. We found out that multi-task models benefit from disentanglement only on specific datasets. However, we cannot name an indicator of when this unambiguously applies.

Acknowledgements The work of Ł. Maziarka was supported by the National Science Centre (Poland) grant no. 2019/35/N/ST6/02125. The work of A. Nowak and M. Wołczyk was supported by Foundation for Polish Science (grant no POIR 04.04.00-00-14DE/18-00) carried out within the Team-Net program co-financed by the European Union under the European Regional Development Fund.

References

1. Abdi, A.H., Abolmaesumi, P., Fels, S.: Variational learning with disentanglement-pytorch. arXiv preprint arXiv:1912.05184 (2019)
2. Bedychaj, A., Spurek, P., Nowak, A., Tabor, J.: Wica: nonlinear weighted ica (2020)
3. Bengio, Y.: Deep learning of representations: Looking forward (2013)
4. Brakel, P., Bengio, Y.: Learning independent features with adversarial nets for non-linear ica (2017)
5. Capela, F., Nouchi, V., Van Deursen, R., Tetko, I.V., Godin, G.: Multitask learning on graph neural networks applied to molecular property predictions. arXiv preprint arXiv:1910.13124 (2019)
6. Caruana, R.: Multitask learning: A knowledge-based source of inductive bias. In: Proceedings of the Tenth International Conference on Machine Learning. pp. 41–48. Morgan Kaufmann (1993)
7. Chen, R.T., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. In: Advances in neural information processing systems. pp. 2610–2620 (2018)
8. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems **29**, 2172–2180 (2016)
9. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014)
10. Do, K., Tran, T.: Theory and evaluation metrics for learning disentangled representations. In: International Conference on Learning Representations (2020), `https://openreview.net/forum?id=HJgK0h4Ywr`

11. Duong, L., Cohn, T., Bird, S., Cook, P.: Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 845–850 (2015)
12. Eastwood, C., Williams, C.K.I.: A framework for the quantitative evaluation of disentangled representations. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=By-7dz-AZ
13. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR (2017)
14. Hirayama, J., Hyvarinen, A., Kawanabe, M.: Splice: Fully tractable hierarchical extension of ica with pooling. In: Proceedings of the International Conference on Machine Learning. vol. 70, pp. 1491–1500. Machine Learning Research (2017)
15. Hsieh, J.T., Liu, B., Huang, D.A., Fei-Fei, L.F., Niebles, J.C.: Learning to decompose and disentangle representations for video prediction. In: Advances in Neural Information Processing Systems. pp. 517–526 (2018)
16. Huang, G.B., Wang, D.H., Lan, Y.: Extreme learning machines: a survey. International journal of machine learning and cybernetics **2**(2), 107–122 (2011)
17. Hyvarinen, A., Morioka, H.: Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In: Advances in Neural Information Processing Systems. pp. 3765–3773 (2016)
18. Hyvarinen, A., Morioka, H.: Nonlinear ica of temporally dependent stationary sources. Proceedings of Machine Learning Research (2017)
19. Kim, H., Mnih, A.: Disentangling by factorising. arXiv preprint arXiv:1802.05983 (2018)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
21. Kumar, A., Sattigeri, P., Balakrishnan, A.: Variational inference of disentangled latent concepts from unlabeled observations. arXiv preprint arXiv:1711.00848 (2017)
22. Kumar, A., Sattigeri, P., Balakrishnan, A.: Variational inference of disentangled latent concepts from unlabeled observations (2018)
23. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and brain sciences **40** (2017)
24. Lipton, Z.C.: The mythos of model interpretability. Queue **16**(3), 31–57 (2018)
25. Liu, X., He, P., Chen, W., Gao, J.: Multi-task deep neural networks for natural language understanding. arXiv preprint arXiv:1901.11504 (2019)
26. Locatello, F., Abbati, G., Rainforth, T., Bauer, S., Schölkopf, B., Bachem, O.: On the fairness of disentangled representations. In: Advances in Neural Information Processing Systems. pp. 14611–14624 (2019)
27. Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations (2019)
28. Locatello, F., Tschannen, M., Bauer, S., Rätsch, G., Schölkopf, B., Bachem, O.: Disentangling factors of variation using few labels. arXiv preprint arXiv:1905.01258 (2019)
29. Ma, J., Zhou, C., Cui, P., Yang, H., Zhu, W.: Learning disentangled representations for recommendation. In: Advances in neural information processing systems. pp. 5711–5722 (2019)

30. Maziarka, Ł., Danel, T.: Multitask learning using bert with task-embedded attention. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–6. IEEE (2021)
31. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
32. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3994–4003 (2016)
33. Petangoda, J.C., Pascual-Diaz, S., Adam, V., Vrancx, P., Grau-Moya, J.: Disentangled skill embeddings for reinforcement learning (2019)
34. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
35. Sanchez, E.H., Serrurier, M., Ortner, M.: Learning disentangled representations via mutual information estimation (2019)
36. Schmidhuber, J.: Learning factorial codes by predictability minimization. Neural computation **4**(6), 863–879 (1992)
37. Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., Mooij, J.: On causal and anticausal learning. arXiv preprint arXiv:1206.6471 (2012)
38. Sorrenson, P., Rother, C., Köthe, U.: Disentanglement by nonlinear ica with general incompressible-flow networks (gin) (2020)
39. Spurek, P., Nowak, A., Tabor, J., Maziarka, Ł., Jastrzębski, S.: Non-linear ica based on cramer-wold metric. In: International Conference on Neural Information Processing. pp. 294–305. Springer (2020)
40. Steenbrugge, X., Leroux, S., Verbelen, T., Dhoedt, B.: Improving generalization for abstract reasoning tasks using disentangled feature representations. arXiv preprint arXiv:1811.04784 (2018)
41. Van Steenkiste, S., Locatello, F., Schmidhuber, J., Bachem, O.: Are disentangled representations helpful for abstract visual reasoning? In: Advances in Neural Information Processing Systems. pp. 14245–14258 (2019)