Disentangled Latent Spaces Facilitate Data-Driven Auxiliary Learning

Geri Skenderi[1,2*] Luigi Capogrosso[3] Andrea Toaiari[3] Matteo Denitto[4]
Franco Fummi[3] Simone Melzi[5]

[1]Bocconi Institute for Data Science and Analytics (BIDSA) [2]Bocconi University [3]University of Verona
[4]HUMATICS - SYS-DAT Group [5]University of Milano-Bicocca

Abstract

*Training a neural network on additional auxiliary tasks can improve learning when data is scarce or the principal task is highly complex. This idea is primarily inspired by the improved generalization induced by solving multiple tasks simultaneously, which leads to a more robust shared representation. However, selecting optimal auxiliary tasks often requires hand-crafted solutions or costly meta-learning approaches. We propose **Detaux**, a novel framework that discovers an auxiliary classification task through weakly supervised disentanglement in a product manifold. Our method isolates variation in the data related to the principal task in a dedicated subspace while producing orthogonal subspaces with high separability. A clustering procedure in the most disentangled subspace generates discrete auxiliary labels that, along with the original data, can be used within any Multi-Task Learning (MTL) framework. Theoretical evidence on the linear independence of task representations, alongside experiments on synthetic and real data, demonstrates the potential to link disentangled representations and MTL.*

1. Introduction

Human learning is often considered a combination of processes (*e.g.*, acquired high-level skills, and evolutionary encoded physical perception) that are used together and can be transferred from one problem to another. Inspired by this, Multi-Task Learning (MTL) [5] represents the machine learning paradigm in which multiple tasks are learned together to improve the generalizability of a model by using shared knowledge that derives from considering different aspects of the input. Specifically, this is achieved by jointly optimizing the model's parameters across different tasks, allowing the model to learn task-specific and task-shared representations simultaneously. As a result, MTL can lead to better generalization, improved efficiency at inference time, and improved performance on individual tasks by exploiting their underlying relationships.

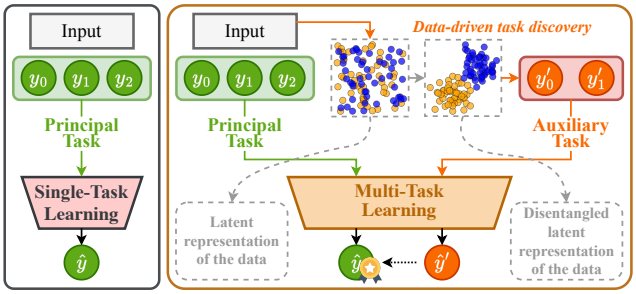

Figure 1. Overview of our data-driven auxiliary task discovery. The figure illustrates the difference between Single-Task Learning (STL) (left) and MTL (right) with **Detaux**, which uses an auxiliary task generated from visual concepts in the data to improve performance on the principal task.

A particular form of this learning approach, referred to as *auxiliary learning*, has garnered considerable interest in recent years [25]. Auxiliary learning consists of using an additional set of auxiliary tasks that operate on the same input data and lead to a shared representation that boosts performance on the principal task, *i.e.*, the only task of interest. In the literature, auxiliary tasks are mainly generated via meta-learning [28, 39], which requires a prior definition of the hierarchy of the desired auxiliary tasks and is usually computationally inefficient. Thus, the question we try to answer in this work is: *Can we discover, with no prior knowledge, an additional auxiliary task directly from the data structure in order to improve the performance of the principal task?*

We explore this problem by proposing **Detaux**, a weakly supervised strategy that discovers auxiliary classification tasks that allow solving a STL classification problem in a MTL fashion, as shown in Figure 1. Specifically, **Detaux** is capable of individuating unrelated auxiliary tasks. Unrelatedness in MTL means having tasks whose features have *no* semantic intersection and has been proven to be effective in learning multiple tasks simultaneously [20, 27, 42, 49, 52, 53]. Our method takes root in the idea of [42], where two groups of tasks, the principal task and the auxiliary tasks,

*Correspondence: geri.skenderi@unibocconi.it

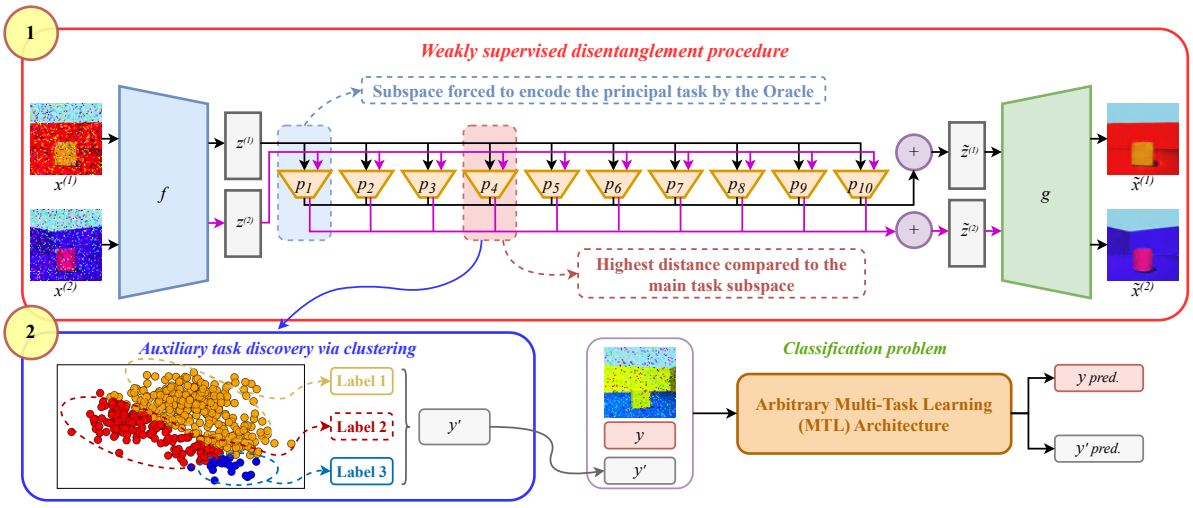

Figure 2. **Detaux** involves two steps: 1) First, we use weakly supervised disentanglement to isolate the structural features specific to the principal task in one subspace (red rectangle at the top). 2) Next, we identify the subspace with the most disentangled factor of variation related to the principal task, and through a clustering module, we obtain new labels (blue rectangle in the bottom left). These can be used to create a new classification task that can be combined with the principal task in any MTL model (bottom right). Best viewed in color.

are known to be unrelated, and assumes the claim that joint learning of unrelated tasks can improve the performance on the principal task. They propose to generate a shared low-dimensional representation for both tasks, forcing these representations to be orthogonal. The procedure from [42] exploits a linear classifier and requires the knowledge of the labels for both the principal task and the auxiliary tasks. Our method extends this process, bypassing the need for supervision, and parametrizing the problem using neural networks. Specifically, it generates auxiliary tasks so that their labels implicitly drive an MTL network to understand the unrelatedness between the tasks. Our idea is to work in a specific representation space, a product manifold, to reveal the auxiliary tasks for a given principal task. We get inspiration from [13], who discovered the product manifold as a convenient representation basis for disentanglement. In particular, as shown in Figure 2, we first extract task-specific features using a *weakly supervised disentanglement procedure* that implements projections in orthogonal subspaces of the latent representation. Then, we identify a subspace where the respective projections are maximally separated. Finally, we generate new labels via a clustering module to enable integration with the principal task through *any* MTL model, as shown in the bottom right of Figure 2. In this way, any MTL model can be chosen depending on several factors in addition to performance, such as efficiency or scalability. The rest of the manuscript provides both a general explanation of our method and experimental evidence on four real datasets with three different MTL models. Finally, we provide theoretical and additional experimental evidence in the Appendix to support our claims.

2. Methodology

2.1. Mathematical Preliminaries

In this section, we provide a high-level overview of the mathematical formalization of disentanglement used in this work. Due to space constraints, we provide an extended explanation in Section 6 of the Appendix, with more extensive treament of all the varous formulas. Disentangled representation learning aims to learn a representation of the data where different latent factors, *i.e.*, visual concepts, are represented independently of the others. In this work, we rely on the definition of disentanglement proposed by [13]. Under the manifold assumption and assuming that independent factors generate the data, it becomes reasonable to see the data manifold as a product manifold: $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2 \times \ldots \times \mathcal{M}_k$ where each $\mathcal{M}_i, i \in \{1 \ldots k\}$, can be intuitively considered orthogonal to the others and represent one latent visual concept. As a result, given a pair of data (x_1, x_2) known to differ in the h-th latent factor only, their learned representations are considered fully disentangled if they have fixed projections in all the submanifolds $\{\mathcal{M}_i\}_{i=1}^{k}$, except for the h-th. Consider, for example, a product manifold composed of k 1D, connected submanifolds (without boundary). This structure can be embedded in the latent space as $\mathbb{R}^k$, with each axis displaying a visual concept. In general, the main advantage of this approach is that each submanifold can have different dimensionalities, with the sum of the dimensions of the submanifolds equal to that of the product manifold. Thus, we can generalize the intuitive idea of an "axis of variation" and look for auxiliary tasks in a higher-dimensional space instead of being limited to 1D axes as in VAE-based methods [17, 21].

In practice, we consider a finite-dimensional normed vector space $\mathbf{Z} \subseteq \mathbb{R}^d$, containing the disentangled latent representation, obtained as the output of an encoder network $f : \mathcal{M} \to \mathcal{Z}$. Therefore, our latent representation takes the form of a Cartesian product space $\mathbf{Z} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_k$, such that $\forall i, j \in \{1 \ldots k\}$, $\mathcal{S}_i \cap \mathcal{S}_j = \{0\}$ if $i \neq j$. Finally, the aggregated latent code is fed into a decoder g that approximates the inverse of f. To wrap up, the representation framework operates in the following way:

$$x \xrightarrow{f} z \xrightarrow[i=1\ldots k]{\{p_i\}} \{s_i\} \xrightarrow[i=1\ldots k]{\sum_i} \tilde{z} \xrightarrow{g} \tilde{x} \ , \tag{1}$$

where the p_i's are nonlinear operators, and $\tilde{z}$ and $\tilde{x}$ are the aggregated latent representation and the reconstructed input, respectively.

The visual representation of this process is shown in Figure 2, within the red rectangle. Given a data pair, this framework can be made equivalent to solving the following optimization problem:

$$\mathcal{L} = \mathcal{L}_{rec} + \beta_1(\mathcal{L}_{dist} + \mathcal{L}_{spar}) + \beta_2\mathcal{L}_{cons} + \beta_3\mathcal{L}_{reg} \ , \tag{2}$$

where β_1, β_2, and β_3 are Lagrange multipliers controlling the influence of the constraints. $\mathcal{L}_{rec}$ is a reconstruction loss to learn the global manifold. $\mathcal{L}_{dist}$ is a contrastive loss based on an oracle $\mathcal{O}$, which selects the subspace $\mathcal{S}_i$ with maximal separation for the given image pair while encouraging similarity in the remaining subspaces. $\mathcal{L}_{spar}$ is an $L1$ sparsity term that promotes orthogonality among subspaces, enforcing a direct sum structure, $\mathcal{L}_{cons}$ encourages each projector p_i to be invariant to variations in other subspaces $\mathcal{S}_j$, for $j \neq i$. Finally, $\mathcal{L}_{reg}$ is a regularization term that ensures that the oracle $\mathcal{O}$ distributes the choices uniformly across subspaces to prevent collapse.

2.2. Auxiliary Task Discovery

We assume the existence of a labeled image dataset $D = \{\, (x^{(i)}, y^{(i)}) \,|\, \forall i \in \{1 \ldots N\},\, x^{(i)} \in \mathcal{R}^{w \times h \times c},\, y^{(i)} \in \mathbb{N}\}$, where w is the width, h the height, c the number of channels, and N the number of tuples (image, label). We consider a classification task whose objective is to learn a mapping $x^{(i)} \to y^{(i)}$ that maps each image to its corresponding label. To discover auxiliary tasks from new visual concepts, we must have a way to accommodate the (known) variation related to the principal task in one subspace and fix it there. To achieve this, we define a principal task oracle $\hat{\mathcal{O}} : \mathbf{Z} \times \mathbf{Z} \to \{1, \ldots, k\}$, which ensures that the α-th subspace will contain all the variation in the data corresponding to pairs $(x^{(1)}, x^{(2)})$ that differ in their principal task label. Note that we do not inject direct knowledge of these labels, but only whether or not they differ, as follows:

$$\hat{\mathcal{O}}(z^{(1)}, z^{(2)}) = \begin{cases} \alpha & \text{if } y^{(1)} \neq y^{(2)} \\ \underset{i \in \{1,\ldots,k\} \setminus \alpha}{\operatorname{argmax}} \; d(s_i^{(1)}, s_i^{(2)}) & \text{otherwise} \end{cases} \tag{3}$$

where $\alpha \in \{1 \ldots k\}$ is the index for the subspace of choice and $d(s_i^{(1)}, s_i^{(2)})$ is the distance between the projections of $(z^{(1)}, z^{(2)})$ in the i-th subspace $\mathcal{S}_i$. The variation in the data is encoded in $\mathcal{S}_\alpha$ if $y^{(1)} \neq y^{(2)}$, and in a different subspace otherwise.

We set $\alpha = 1$, but any other value in $\{1...k\}$ is perfectly suitable. The choice of the subspace for the case $y^{(1)} = y^{(2)}$ is made by looking at where the distance between the projections is maximal, as this is where the difference between the pair in that latent factor will be encoded.

In practice, we approximate the argmax operator in Equation 3 by applying a $\operatorname{softmax}$ at low temperature to the normalized pairwise Euclidean distances. This produces a discrete probability distribution over the distances, which can then be used to weigh the contributions of the projections in each subspace.

After this step, we can find new auxiliary tasks in the remaining $\{1 \ldots k\} \setminus \alpha$ subspaces. Let $\mathcal{S}_j$ be the subspace where the distances are maximally preserved (minimal $\mathcal{L}_{dist}$ in Equation 8). We apply a clustering algorithm to the latent representation found in $\mathcal{S}_j$, as shown inside the blue rectangle of Figure 2. This gives rise to a set of discrete pseudo-labels, determining the new auxiliary classification task. Given that the disentanglement procedure already forces each subspace to cluster embeddings (through $\mathcal{L}_{dist}$ and $\mathcal{L}_{cons}$, Equation 10), choosing subspace $\mathcal{S}_j$ provides a great advantage for the clustering procedure.

Although it is possible to use an arbitrary clustering algorithm, one would prefer it to support clusters of arbitrary shapes and not to specify directly the number of clusters. Therefore, we utilize HDBSCAN [4] because it allows us to group data points based on their density without explicitly specifying the number of groups. Finally, HDBSCAN can associate points that cannot be assigned to any cluster to a "noise" cluster, which we can retain as an additional label of the auxiliary task, making it more robust. If HDBSCAN finds only one cluster, we denote the run as unsuccessful and stop the procedure, as training in it would lead to MTL trivial results. Otherwise, we have discovered a novel task and its corresponding labels $y' \in \mathbb{N}$, which can be used with *any MTL model* alongside the principal task, as shown in the blue rectangle of Figure 2. In this work, we limit ourselves to finding only one auxiliary task, as this is common practice [24] and because the use of multiple auxiliary tasks increases the computational load of MTL. Scaling on more tasks is the subject of future work. At this stage, we have enriched our dataset with an additional set of labels, obtaining $D' = \{\, (x_i, y_i, y'_i) \,|\, \forall i \in \{1 \ldots N\},\, x_i \in \mathcal{R}^{w \times h \times c},\, y_i, y'_i \in \mathbb{N}\}$. In Section 7 of the Appendix, we present a theoretical analysis as to why this setup provides uncorrelated representations that ultimately lead HDBSCAN to produce different and uncorrelated clustering.

Table 1. Classification accuracy on the FACES [11], CIFAR-10 [23], SVHN [40], and Cars [22] datasets. (*) indicates results reported in the original paper due to reproducibility challenges with the available code. In **bold**, the best results. Underlined the second best.

Learning Paradigm	FACES ↑	CIFAR-10 ↑	SVHN ↑	Cars ↑
STL	0.915	0.844	0.956	0.711
MAXL [28]	0.933	0.868	0.953	0.638
AuxiLearn [39]	0.915	0.811	0.943	0.644*
MTL-HPS [5] + **Detaux**	0.951	0.848	0.954	0.789
NDDR [14] + **Detaux**	0.932	0.872	0.952	0.712
MTI [48] + **Detaux**	**0.978**	**0.910**	**0.961**	**0.807**

3. Experiments

Due to lack of space, we provide implementation details in Section 8 of the Appendix. As mentioned previously, the image pairs are sampled based on the principal task labels. For the FACES [11] dataset, this corresponds to the person's facial expression. For CIFAR-10 [23], SVHN [40], and Cars [22], it corresponds to the only available task. We used a ResNet-18 [16] encoder-decoder architecture for the disentanglement phase (Figure 2 (1)) to obtain a high-fidelity reconstruction. We compare our approach with two different auxiliary learning methods, *i.e.*, MAXL [28] and AuxiLearn [39]. Unlike these approaches based on meta-learning, our discovered auxiliary task can be exploited interchangeably with any MTL model. To have an evaluation that is as fair as possible and focuses on the benefits of the auxiliary task, we choose parameter-sharing MTL networks. This enforces the idea that the gains are due to the new task and not advanced learning dynamics, a dichotomy also discussed in [26]. We select three different models: the standard Hard Parameter Sharing for MTL (MTL-HPS) [5], NDDR [14], and MTI [48]. The main loss term in all these models is a summation of each task's classification loss.

Table 1 summarizes the results. MTI, with our generated auxiliary labels, displays the best performance. Furthermore, even simple ConvNet-based models, such as MTL-HPS and NDDR, achieve superior results compared to MAXL and AuxiLearn. Most notably, we outperform STL with at least one of the MTL+**Detaux** models *in all the datasets*, while MAXL and AuxiLearn have performance discrepancies. For fairness, we report that we exploit pre-trained backbones for the MTL models on Cars, which contains very complex images and is categorized as a fine-grained classification dataset. For the disentanglement phase, we change the encoder f so that it does not produce a dense representation in the bottleneck layer but a compressed feature map via 1×1 convolution. Finally, to show the ideal scenarios of our method, we provide additional results with a synthetic setup in the Appendix 8.

How does Detaux extract auxiliary visual concepts? This experiment aims to show how disentanglement effectively extracts task labels from the underlying data structure. On the FACES dataset, we compare the auxiliary task generated by **Detaux** with the auxiliary task resulting from the clustering in the latent space of an autoencoder that only learns to reconstruct. Without disentanglement, MTL-HPS can only reach 0.9 accuracy, worse than the 0.915 obtained by STL. This reveals that performing an auxiliary task mining on the entangled autoencoder space provides a less informative auxiliary task to the MTL network compared to our approach. We provide further qualitative evidence of this observation in Figure 5 of the Appendix.

Is it possible to use other clustering algorithms? One immediate question that may come to mind with regard to **Detaux** is its flexibility w.r.t. the clustering method. As mentioned in Section 2.2, we rely on HDBSCAN due to its nice properties. We aim to show that our pipeline can improve downstream performance even with other (and simpler) clustering algorithms. In particular, we use two versions of the KMeans algorithm [1, 29], which assume a flat geometry, and MeanShift [8], which works well even in non-flat geometries. The results are presented in Table 2 and clearly show that our method is flexible to the choice of the clustering algorithm, improving performance in all cases when compared to STL.

Are the generated labels correlated? To empirically verify that our pipeline design is aligned with the underlying theoretical analysis, we calculate the Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI) between the principal task and the auxiliary task labels generated by **Detaux**. These metrics are used in the clustering literature to measure the agreement between two label assignments, independently of the order [43]. The results in Table 3 indicate that the two sets of labels are almost uncorrelated, with the labels of FACES showing minimal correlation. To further confirm our claim, we run a contingency-based test χ^2 with the null hypothesis that the two groups have no significant differences. For all datasets, the p-values are essentially 0 (the largest one being 9.06×10^{-17}), allowing us to reject the null hypothesis with high confidence.

4. Conclusion

In this paper, we propose a novel outlook on the utility of disentangled representations, utilizing them as a proxy for auxiliary learning in order to improve the accuracy of a principal task, originally solvable only in a STL fashion. Our proposed pipeline facilitates the weakly supervised discovery of a new classification task from a factorized representation, which can be incorporated into any MTL framework, offering better performance w.r.t. the principal task.

Acknowledgments

Geri Skenderi is funded by the European Union through the Next Generation EU - MIUR PRIN PNRR 2022 Grant P20229PBZR. The views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible.

References

[1] David Arthur and Sergei Vassilvitskii. k-means++: The Advantages of Careful Seeding. Technical Report, Stanford InfoLab, 2006. 4, 7

[2] Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798–1828, 2013. 1

[3] Burgess, Chris and Kim, Hyunjik. 3D Shapes Dataset. https://github.com/deepmind/3dshapes-dataset/, 2018. Accessed: 2024-11-03. 2, 4, 5

[4] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2013. 3, 6, 7

[5] Rich Caruana. Multitask Learning. *Machine learning*, 28: 95–133, 1998. 1, 4, 7

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 1

[7] Roberto Cipolla, Yarin Gal, and Alex Kendall. Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[8] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. 4, 7

[9] Lucio M Dery, Paul Michel, Mikhail Khodak, Graham Neubig, and Ameet Talwalkar. AANG: Automating Auxiliary Learning. In *International Conference on Learning Representations (ICLR)*, 2022. 1

[10] Cian Eastwood and Christopher KI Williams. A Framework for the Quantitative Evaluation of Disentangled Representations. In *International Conference on Learning Representations (ICLR)*, 2018. 1

[11] Natalie C Ebner, Michaela Riediger, and Ulman Lindenberger. FACES—A database of facial expressions in young, middle-aged, and older women and men: Development and validation. *Behavior Research Methods*, 42(1):351–362, 2010. 4, 5, 7

[12] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently Identifying Task Groupings for Multi-Task Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[13] Marco Fumero, Luca Cosmo, Simone Melzi, and Emanuele Rodolà. Learning disentangled representations via product manifold projection. In *International Conference on Machine Learning (ICML)*, 2021. 2, 1, 4, 5

[14] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. NDDR-CNN: Layerwise Feature Fusing in Multi-Task CNNs by Neural Discriminative Dimensionality Reduction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4, 1

[15] Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[17] 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. In *International Conference on Learning Representations (ICLR)*, 2017. 2, 1

[18] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a Definition of Disentangled Representations. *arXiv preprint arXiv:1812.02230*, 2018. 1

[19] Daniella Horan, Eitan Richardson, and Yair Weiss. When Is Unsupervised Disentanglement Possible? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[20] Dinesh Jayaraman, Fei Sha, and Kristen Grauman. Decorrelating Semantic Visual Attributes by Resisting the Urge to Share. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[21] Hyunjik Kim and Andriy Mnih. Disentangling by Factorising. In *International Conference on Machine Learning (ICML)*, 2018. 2, 5

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *International Conference on Computer Vision Workshops (ICCVW)*, 2013. 4, 7

[23] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. Technical Report, University of Toronto, 2009. 4, 7

[24] Yong Li and Shiguang Shan. Meta Auxiliary Learning for Facial Action Unit Detection. *IEEE Transactions on Affective Computing*, 14(3):2526–2538, 2023. 3, 1

[25] Lukas Liebel and Marco Körner. Auxiliary Tasks in Multi-task Learning. *arXiv preprint arXiv:1805.06334*, 2018. 1

[26] Baijiong Lin and Yu Zhang. LibMTL: A Python Library for Deep Multi-Task Learning. *The Journal of Machine Learning Research*, 24(1):9999–10005, 2023. 4

[27] Cheng Liu, Chu-Tao Zheng, Sheng Qian, Si Wu, and Hau-San Wong. Encoding sparse and competitive structures

among tasks in multi-task learning. *Pattern Recognition*, 88: 689–701, 2019. 1

[28] Shikun Liu, Andrew Davison, and Edward Johns. Self-Supervised Generalisation with Meta Auxiliary Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 4, 5

[29] S. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 4, 7

[30] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *International Conference on Machine Learning (ICML)*, 2019. 1

[31] Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling Factors of Variation Using Few Labels. *arXiv preprint arXiv:1905.01258*, 2019. 5

[32] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-Supervised Disentanglement Without Compromises. In *International Conference on Machine Learning (ICML)*, 2020. 1

[33] Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling factors of variation using few labels. In *International Conference on Learning Representations (ICLR)*, 2020. 1

[34] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2018. 4

[35] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing Sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2018. Accessed: 2024-11-03. 2

[36] Łukasz Maziarka, Aleksandra Nowak, Maciej Wołczyk, and Andrzej Bedychaj. On the Relationship Between Disentanglement and Multi-task Learning. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2023. 2

[37] Qingjie Meng, Nick Pawlowski, Daniel Rueckert, and Bernhard Kainz. Representation Disentanglement for Multi-task Learning with Application to Fetal Ultrasound. In *Smart Ultrasound Imaging and Perinatal, Preterm and Paediatric Image Analysis*, 2019. 1

[38] Jaehyun Nam, Jihoon Tack, Kyungmin Lee, Hankook Lee, and Jinwoo Shin. STUNT: Few-shot Tabular Learning with Self-generated Tasks from Unlabeled Tables. In *International Conference on Learning Representations (ICLR)*, 2023. 1

[39] Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. Auxiliary Learning by Implicit Differentiation. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 4

[40] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, and Andrew Y Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *Advances in Neural Information Processing Systems Workshop (NeurIPSW)*, 2011. 4, 5, 7

[41] Utkarsh Ojha, Krishna Kumar Singh, Cho-Jui Hsieh, and Yong Jae Lee. Elastic-InfoGAN: Unsupervised Disentangled Representation Learning in Class-Imbalanced Data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[42] Bernardino Romera Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil. Exploiting Unrelated Tasks in Multi-Task Learning. In *15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012. 1, 2, 4

[43] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Adjusting for Chance Clustering Comparison Measures. *Journal of Machine Learning Research*, 17 (134):1–32, 2016. 4

[44] K Simonyan and A Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[45] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. FineGAN: Unsupervised Hierarchical Disentanglement for Fine-Grained Object Generation and Discovery. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[46] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which Tasks Should Be Learned Together in Multi-task Learning? In *International Conference on Machine Learning (ICML)*, 2020. 1

[47] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 5th edition, 2016. 4

[48] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. MTI-Net: Multi-scale Task Interaction Networks for Multi-task Learning. In *European Conference on Computer Vision (ECCV)*, 2020. 4, 1

[49] Hongcheng Wang and Ahuja. Facial expression decomposition. In *International Conference on Computer Vision (ICCV)*, 2003. 1

[50] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing Knowledge in Neural Networks. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[51] Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning (IJCAI). In *28th International Joint Conference on Artificial Intelligence*, 2019. 1

[52] Yu Zheng, Jianping Fan, Ji Zhang, and Xinbo Gao. Exploiting Related and Unrelated Tasks for Hierarchical Metric Learning and Image Classification. *IEEE Transactions on Image Processing*, 29:883–896, 2020. 1

[53] Dengyong Zhou, Lin Xiao, and Mingrui Wu. Hierarchical Classification via Orthogonal Transfer. In *International Conference on Machine Learning (ICML)*, 2011. 1

Disentangled Latent Spaces Facilitate Data-Driven Auxiliary Learning

Appendix

5. Related Work

5.1. MTL and Auxiliary Learning

Multi-Task Learning (MTL), *i.e.*, the procedure through which we can solve multiple learning problems at the same time [5], can help us reduce the inference time, improve the precision, and increase the efficiency of the data [46]. When the adopted dataset contains annotation for multiple tasks, the challenges to face concern which tasks may work well together [12, 46, 51] or how to weigh the losses of different tasks [7] to create a better joint optimization objective. Numerous methods have recently emerged that address the simultaneous resolution of multiple tasks [5, 14, 48].

A different problem arises when we would like to use a MTL method, but the given dataset contains annotations for only one task. The learning of additional tasks aims to maximize the prediction performance on a principal task by supervising the model to learn other tasks, as shown in [28, 39]. Therefore, auxiliary tasks are tasks of minor interest, or even irrelevant compared to the principal task we want to solve, and thus can be seen as regularizers if learned simultaneously with the task of interest [25]. For example, [42] suggests that the use of two unrelated groups of tasks, where one of them hosts the principal task, can lead to better performance, where unrelated means that an orthogonal set of features defines the two groups of tasks. In [25], the authors use seemingly unrelated tasks to help the learning on one principal task, this time without imposing any constraint on the feature structure. With **Detaux**, we are working in the product manifold space, which has already been shown by [13] to be effective for separating embedding subspaces that are orthogonal by design.

Moreover, recent emerging techniques leverage meta-learning to select the most appropriate auxiliary tasks or even autonomously create novel ones. Both [28] and [24] train two neural networks simultaneously: a label-generation model to predict the auxiliary labels and a multi-task model to train the primary task alongside the auxiliary task. In contrast with our approach, these require the a priori definition of a hierarchy binding the auxiliary labels to the principal task labels and present conflicting ideas on the possible semantic interpretation of the generated labels. Furthermore, they are computationally inefficient: meta-learning is a resource-intensive technique that requires retraining the entire architecture to change the employed multi-task method. [38] also used meta-learning, presenting a novel framework to generate new auxiliary objectives to address the niche problem of few-shot semi-supervised tabular learning. Finally, [9] proposes to deconstruct existing natural language processing objectives within a unified taxonomy, identifying connections between them, and generating new ones by selecting the best combinations from a Cartesian product of the available options. To the best of our knowledge, we are not aware of any other method that proposes a systematic approach to generate new labels from a disentangled latent space to enable MTL classification when only the annotations for one task are given in the considered dataset; thus **Detaux** represents the first effort in this sense.

5.2. Learning Disentangled Representations

Representing data in a space where different components are independent is a long-standing research topic in machine learning. The rise of deep learning, which is based on learning representations, has made this concept even more relevant and useful in understanding the latent space [2].

Recent literature has proposed several characterizations of disentanglement, whether that is in terms of group theory [18], metric and product spaces [13], or permutations of element-wise, nonlinear functions [19]. [17] demonstrates that variational auto-encoders could learn to disentangle by enforcing the ELBO objective, while [6] relies on generative adversarial networks and an information-theoretic view of disentanglement. Later works, such as [10, 41, 45], extensively explored different directions and use cases. [30] showed that completely unsupervised disentanglement is not possible due to the inability of the models to identify factors of variation. Soon after, the authors proposed weak supervision and access to few labels to bypass this limitation [32, 33]. In **Detaux**, we place ourselves in the same setting of [13] but control and force the disentanglement by supervision only on the known (principal) task.

5.3. MTL and Disentanglement

[37] reports a connection between disentangled representations and MTL, showing that disentangled features can improve the performance of multi-task networks, especially on data with previously unseen properties. Disentanglement is obtained by adversarial learning, forcing the encoded features to be minimally informative about irrelevant tasks. In this case, the tasks to be disentangled are known a priori, while in our case, only the principal task task is known.

[50] proposes a novel concept called "Knowledge Factorization". Exploiting the knowledge contained in a pre-trained multi-task network (called teacher), the idea is to train disentangled single-task networks (called students) to reduce the computational effort required by the final single-task network. The factorization of the teacher knowledge is

dual: they provide structural factorization and representation factorization. In structural factorization, they split the net into a common-knowledge network and a task-specific network based on mutual information.

Finally, [36] explores the degree of disentanglement of MTL models in a controlled, semi-synthetic setting. Initially, a set of task labels is created using a randomly initialized Multi-Layer Perceptron (MLP) starting from the latent factors of parametric disentanglement datasets [3, 15, 35]. The authors successively train a separate neural network to solve these artificially created tasks and understand how disentangled the representations are, w.r.t. the original latent factors. The reported results may be seen as inconclusive, as they do not clearly indicate how disentangled representations directly impact MTL performance.

In this work, we show that disentanglement in a representation space can be used as a general prior for MTL. After using disentanglement to mine for auxiliary tasks, an MTL model extracts a model-specific embedding that takes advantage of the combination of the principal and newly discovered labels, improving downstream performance on the principal task.

6. Detailed Mathematical Background

6.1. Disentanglement Framework

At a high level, disentangled representation learning aims to learn a representation of the data where different latent factors are represented independently of the others; that is, we have a factorization (a.k.a. disentanglement) of the representation. There are different ways to properly formalize this general concept. In this work, we rely on the definition and approach of disentanglement proposed by [13].

The primary assumption behind this framework is the manifold hypothesis, *i.e.*, that high-dimensional data lies near a lower-dimensional manifold. Building upon this idea and assuming that independent factors generate the data, it becomes reasonable to see the manifold as a product manifold: $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2 \times \ldots \times \mathcal{M}_k$. In such a topological structure, each $\mathcal{M}_i, i \in \{1 \ldots k\}$, is orthogonal to the others, and thus, we would like it to represent at most one latent factor of the data. This concept is adequately formalized by relying on the topological construct of a metric space and employing what we call a weak isometry between the data and the learned product manifolds, defined as follows.

Definition 6.1 (Product Manifold Disentanglement [13])**.** Let $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2 \times \ldots \times \mathcal{M}_k$ be the data product manifold, embedded in a high-dimensional space $\mathcal{X}$. Furthermore, let us assume that we have access to some metric that endows these two spaces with the properties of a metric space. A representation z in some product space $\mathcal{Z} = S_1 \times \ldots \times S_k$, such that $dim(\mathcal{Z}) \ll dim(\mathcal{X})$, is *disentangled* with respect to $\mathcal{M}$ if there exists a diffeomorphism (a bijection with a smooth inverse) $\tilde{g} : \mathcal{Z} \to \mathcal{M}$ such that $\forall x_1, x_2 \in \mathcal{M}; \forall i \in 1, \ldots, k$:

$$d_{\mathcal{M}_i}(x_1^i, x_2^i) > 0 \implies d_{\mathcal{S}_i}(s_1^i, s_2^i) > 0 \,,$$
$$d_{\mathcal{M}_i}(x_1^i, x_2^i) = 0 \implies d_{\mathcal{S}_i}(s_1^i, s_2^i) = 0 \implies s_1^i = s_2^i \,,$$

where x_j^i is the projection of x_j on $\mathcal{M}_i$ and $s_j^i \Pi_i \tilde{g}^{-1}(x_j)$ with Π_i being the projection onto the subspace $\mathcal{S}_i \subset \mathcal{Z}$.

As a result, according to Definition 6.1, given a pair of data (x_1, x_2) known to differ in the h-th latent factor only, their learned representations are considered fully disentangled if they have fixed projections in all the submanifolds $\{\mathcal{M}_i\}_{i=1}^k$, except for the h-th.

To provide a pictorial understanding of the above definition, we provide the following example to the reader: Consider the simple case where the data live in $M = \mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$, embedded in an ambient space $\mathcal{X}$ of arbitrary (but finite) dimension. We can see this data manifold as the Cartesian plane and label the two submanifolds as the well-known x and y axes. Given that both are diffeomorphic to open subsets of the real number line, our goal is to learn a latent representation where the x-coordinate is embedded into one subspace and the y coordinate into the other, such that they remain separate. In this simple example, the disentangled representation would correspond to an intuitive change of basis in $\mathbb{R}^2$. Very similarly, any product manifold composed of n 1D, connected, non-compact submanifolds without boundary, could be represented in latent space as $\mathbb{R}^n$ while respecting Definition 6.1. This approach comes with a great advantage when it comes to its application in generating auxiliary tasks, which is that each submanifold can have a different dimensionality. Therefore, we can generalize the intuitive idea of an "axis of variation" and look for auxiliary tasks in a higher-dimensional space instead of being limited to 1D representation axes as in Variational Auto-Encoders (VAE)-based methods [17, 21].

6.2. Disentanglement Training Procedure

In practice, we consider a finite-dimensional normed vector space $\mathbf{Z} \subseteq \mathbb{R}^d$, containing the disentangled latent representation, obtained as the output of an encoder network $f : \mathcal{M} \to \mathcal{Z}$. Note that $\mathbf{Z}$ is a particular case of a manifold. Therefore, our latent disentangled representation takes the form of a Cartesian product space $\mathbf{Z} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_k$, such that $\forall i, j \in \{1 \ldots k\}$, with $i \neq j$, $\mathcal{S}_i \cap \mathcal{S}_j = \{0\}$. As mentioned above, each subspace encodes a generalized notion of a "axis of variation". The representations in each subspace are then aggregated, and a decoder g maps the resulting vectors back to the input data space. More specifically, each $\mathbf{S}_i \subseteq \mathcal{R}^d$ is defined in such a way that it has the same ambient dimensionality as the product space $\mathbf{Z}$. Using a specific regularization (defined in Equation 9), each subspace will have only a few non-zero entries, and the non-zero entries in one subspace will be zero in the others. This

encourages orthogonal and sparse representations for each $\mathbf{S}_i$, which can then be summed to produce a latent code. Subsequently, this latent code is fed into a decoder g that approximates the inverse of f. Thus, the decoder is the approximation of the function $\tilde{g}$ in Definition 6.1.

To wrap up, the representation framework operates in the following way:

$$x \xrightarrow{f} z \xrightarrow[i=1\dots k]{\{p_i\}} \{s_i\} \xrightarrow[i=1\dots k]{\sum_i} \tilde{z} \xrightarrow{g} \tilde{x} \ , \tag{4}$$

where the p_i's are nonlinear operators, and $\tilde{z}$ and $\tilde{x}$ are the aggregated latent representation and the reconstructed input, respectively. The visual representation of this process is shown in Figure 2, within the red rectangle. In the following, we describe how it is possible to parameterize this framework with neural networks and train it end-to-end.

The maps f and g are approximated using an autoencoder architecture. The encoder f receives non-i.i.d data pairs $(x^{(1)}, x^{(2)})$ and produces the latent representations $(z^{(1)}, z^{(2)})$, with the decoder g that approximates the inverse of f. The reason for training with input pairs is to have a sampling procedure designed to induce weak supervision, requiring a pair of images known to vary in at least one latent factor (this is crucial to later isolate the change from $x^{(1)}$ to $x^{(2)}$ in one subspace). Additionally, a set of k neural networks $p_i, i \in \{1 \dots k\}$ called projectors are trained simultaneously to map the latent codes in the subspaces $\{\mathcal{S}_i\}_{i=1}^k$, each of which contains the corresponding submanifold $\{\mathcal{M}_i\}_{i=1}^k$.

An initial warm-up phase trains f and g only to minimize the data reconstruction error, which is needed to learn the global data manifold $\mathcal{M}$. After this warm-up phase, four differentiable constraints that regard different aspects of the desiderata defined in Section 6.1 are added, posing the following optimization problem:

$$\mathcal{L} = \mathcal{L}_{rec} + \beta_1(\mathcal{L}_{dist} + \mathcal{L}_{spar}) + \beta_2\mathcal{L}_{cons} + \beta_3\mathcal{L}_{reg} \ , \tag{5}$$

where β_1, β_2, and β_3 are Lagrange multipliers. $\mathcal{L}_{rec}$ corresponds to a *reconstruction loss*, implemented in practice as the squared error between the input and the reconstructed images following the aggregation operation of the subspaces in the latent space. It is defined as:

$$\mathcal{L}_{rec} = \|x - \bar{x}\|_2^2 \ , \tag{6}$$

$$\bar{x} = g\Big(\sum(p_1(f(x)), \dots, p_k(f(x))\Big) \ . \tag{7}$$

This term is necessary to learn the global structure of the manifold $\mathcal{M}$.

The *distance loss*, $\mathcal{L}_{dist}$, is a contrastive loss term that follows the oracle $\mathcal{O}$, defined in Section 6.1, which calculates the subspace $\mathcal{S}_i$ where the projections of the images in the pair (x_1, x_2) differ the most and encourages the projection representation of the two input images onto the subspaces not selected by $\mathcal{O}$ to be as close as possible. It is defined as:

$$\mathcal{L}_{dist} = \sum_{i=1}^{k}(1 - \lambda_i)\delta_i^2 + \lambda_i \max(m - \delta_i, 0)^2 \ , \tag{8}$$

where $\lambda_i = 1$ if $\mathcal{O}(z^{(1)}, z^{(2)}) = i$ and 0 otherwise, while m is a hyperparameter that constrains the points to be at least at a distance m from each other.

$\mathcal{L}_{spar}$ is a *L1* constraint which promotes sparsity and orthogonality between the subspaces. It is defined as:

$$\mathcal{L}_{spar} = \sum_{i=1}^{k} \|p_i(f(x)) \odot \sum_{j\neq i}^{k} p_j(f(x))\|_1 \ . \tag{9}$$

This constraint allows the disentanglement framework to use the sum operation to aggregate the subspaces. The minimization of $\mathcal{L}_{spar}$ promotes sparsity and orthogonality between the subspaces, encouraging each one to have a few non-zero entries that will be zero in the others. In our finite-dimensional setting, this loss is equivalent to imposing that the product space is a direct sum of the subspaces.

$\mathcal{L}_{cons}$, namely the *consistency loss*, encourages each projector p_i to be invariant to changes in subspaces $\mathcal{S}_j, j \neq i$. It is defined as:

$$\mathcal{L}_{cons} = \sum_{i=1}^{k} ||p_i(f_\theta(\hat{x}_{s_i})) - s_i||_2^2 \ , \tag{10}$$

with $s_i = p_i f(x_1)$ and $\hat{x}_{s_i} = g(\sum(p_i f(x_1), p_{j\neq i} f(x_2)))$. Along with $\mathcal{L}_{dist}$, this constraint encourages a metric definition of disentanglement, *i.e.*, given a pair of images that are different in image space w.r.t. to a particular factor, they should be equally different in the latent representation of that attribute, hosted only in one submanifold which composes the global, product manifold of the latent representation.

Finally, the *regularization loss* $\mathcal{L}_{reg}$ introduces a penalty that ensures the choice of the oracle $\mathcal{O}$ is uniformly distributed among the subspaces to avoid the collapse of information. This is necessary given the initial warm-up period with only the reconstruction loss being active, as there is no guarantee that information will be equally spread out among the subspaces. It is defined as:

$$\mathcal{L}_{reg} = \sum_{j=1}^{k}\left(\frac{1}{N}\sum_{n=1}^{N}\mathbf{A}_{n,j} - \frac{1}{k}\right)^2 \ , \tag{11}$$

with $\mathbf{A} \in \mathbb{R}^{N\times k}$ being the practical implementation of the oracle indicator variables of Equation 8 in a batch of N pairs, obtained by applying a weighted softmax to the distance matrix of pairs in each of the k subspaces.

7. Theoretical analysis

Under the assumption that tasks living in orthogonal spaces help increase MTL performance [42], we now show why our method regularizes the learning procedure and implicitly guides it towards orthogonal feature spaces for each task. For the rest of the paragraphs, we assume perfect disentanglement *i.e.*, $\mathcal{L} = 0$ in Equation 5, to study the behavior of the system at a minima of the problem. Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the vectorized representation of the dataset ($d = w \times h \times c$), $\mathcal{S}_\alpha \in \mathbb{R}^{N \times h}$ be the subspace that contains the representation of the principal task, forced by Equation 3, and $\mathcal{S}_j \in \mathbb{R}^{N \times h}$ be the subspace that contains the representation of the auxiliary task, as described previously. We then obtain the following results:

Proposition 7.1. The representations $\mathcal{S}_\alpha$ and $\mathcal{S}_j$ are not correlated. Furthermore, given the respective distance matrices $\mathbf{\Delta}^\alpha_{ab} = ||s^{(a)}_\alpha - s^{(b)}_\alpha||_2$, $\mathbf{\Delta}^j_{ab} = ||s^{(a)}_j - s^{(b)}_j||_2$, $a, b \in \{1, \dots, N\}$, and some scalar $\gamma \in \mathbb{R}$, we have $\mathbf{\Delta}^j \neq \gamma \mathbf{\Delta}^\alpha$.

Proof. The uncorrelatedness of the representations is a direct consequence of the complete minimization of the sparsity and orthogonality constraint $\mathcal{L}_{spar}$ (Equation 9). For any vector $x \in \mathbf{X}$, we have that:

$$\|p_\alpha(f(x)) \odot p_j(f(x))\|_1 = 0 \,, \tag{12}$$

which directly implies that $\langle \mathcal{S}_\alpha, \ \mathcal{S}_j \rangle_F = 0$. Assuming without loss of generality that the representations are centered at 0, this leads to the conclusion that the two representations are uncorrelated as they have 0 covariance.

Similarly, the second part of the proposition is a direct consequence of the complete minimization $\mathcal{L}_{spar}$ and the consistency constraint $\mathcal{L}_{cons}$ (Equation 10). The complete minimization of $\mathcal{L}_{cons}$ makes the non-linear operator p_i invariant to changes in the subspaces $S_j, \forall j \neq i$. [13]. Now, we proceed by contradiction. Assume that the two distance matrices are proportional to each other's scalar multiple. Then, having proportional pairwise distances would imply that there exists a linear function $\omega : \mathbf{Z} \longrightarrow \mathbf{Z}$ $p_j(f(x)) = \omega(p_\alpha(f(x)))$, implying that the vector in the auxiliary subspace is a function of $p_\alpha(f(x))$. A straightforward example of this would be a permutation followed by scaling. If this were the case, $p_j(\cdot)$ would not be invariant to the changes in S_α, as it directly depends on it, so by contradiction, we can conclude that $\mathbf{\Delta}^j \neq \gamma \mathbf{\Delta}^\alpha$. □

Proposition 7.2. Let the overlap matrix of two square matrices $\mathcal{A}$ and $\mathcal{B}$ be defined as $\mathcal{V}_{AB} = \mathcal{A}^T \mathcal{B}$. Then, the overlap matrix between the eigenvectors of the Gram matrices $\mathcal{G}_\alpha = \mathcal{S}_\alpha \mathcal{S}_\alpha^T$ and $\mathcal{G}_j = \mathcal{S}_j \mathcal{S}_j^T$ is different from the Identity matrix $\mathcal{I}_n$, *i.e.*, they have different eigenvectors.

Proof. From Proposition 7.1, we have that $\mathcal{G}_\alpha \not\propto \mathcal{G}_j$, meaning that the pairwise similarities in both spaces are not proportional (intended as equal or different up to a scalar factor). Given that both Gram matrices are Symmetric and Positive Semi-Definite, by the Spectral Theorem [47], we can diagonalize them and obtain a set of n orthonormal eigenvectors with real eigenvalues:

$$\mathcal{G}_\alpha = \mathcal{U}_\alpha \Lambda_\alpha \mathcal{U}_\alpha^T \,, \tag{13}$$

$$\mathcal{G}_j = \mathcal{U}_j \Lambda_j \mathcal{U}_j^T \,. \tag{14}$$

The reason why we are interested in these eigenvectors is that they contain orthogonal directions of pairwise similarity, thus indicating the pairwise groupings present in the dataset. By simply calculating the overlap matrix on the above eigendecomposition, it is straightforward to see that:

$$\begin{aligned} \mathcal{V}_{\mathcal{G}_\alpha \mathcal{G}_j} &= (\mathcal{U}_\alpha \Lambda_\alpha \mathcal{U}_\alpha^T)^T \mathcal{U}_j \Lambda_j \mathcal{U}_j^T \,. \\ &= \mathcal{U}_\alpha \Lambda_\alpha \mathcal{U}_\alpha^T \mathcal{U}_j \Lambda_j \mathcal{U}_j^T \,. \\ &\neq \mathcal{I}_n \,. \end{aligned}$$

Therefore, the eigenvectors do not perfectly align, and thus, the fixed directions of pairwise similarities are different in the two subspaces. □

Proposition 7.1 implies that the relationship between $d(s^{(a)}_\alpha, s^{(b)}_\alpha)$ will not influence the one between $d(s^{(a)}_j, s^{(b)}_j)$, given that the information encoded in each subspace is different. Furthermore, due to Proposition 7.2, the structure of the pairwise similarity between points (given by the eigenvectors of the Gram matrices) is different in the two subspaces. Thus, a clustering algorithm that relies on pairwise similarity, such as HDBSCAN, will produce different clusterings.

8. Additional Experiments

Implementation details. Our code is written within the PyTorch Lightning framework. We fix the batch size to 32 and the learning rate to 0.0005 for all the experiments and use the AdamW [34] optimizer. The disentanglement model is trained for 40 epochs on 3D Shapes [3] and 400 epochs on FACES [11], CIFAR-10 [23], SVHN [40], and Cars [22]. The first quarter of the epochs is used as a warm-up period where only $\mathcal{L}_{rec}$ is active. The multipliers $\beta_1, \beta_2, \beta_3$ follow an exponential warm-up routine after the reconstruction-only phase, such that the constraints they modulate are gently introduced in the optimization procedure. The projectors $p_i, i \in \{1 \dots k\}$ are implemented as two layers MLPs. Finally, all the MTL models were trained for 150 epochs. All experiments were performed on NVIDIA RTX 3090 GPUs.

Synthetic data. To showcase the capabilities of **Detaux**, we begin our experimental validation with the 3D Shapes

dataset, a common benchmark in the disentanglement literature [13, 21, 31]. 3D Shapes comprises six generative factors: hue of the floor, hue of the wall, hue of the object, scale, shape, and orientation. It is parametrically generated through the Cartesian product between these factors, resulting in 480,000 images. To adapt it to our case, we treat the classification of one generative factor as principal task and pretend that we do not know the others.

Due to the synthetic nature of the images in 3D Shapes, the execution of classification tasks with a neural network can be excessively easy, leaving a limited possibility for improvement through MTL. Specifically, using a simple VGG16 [44] model, we achieve perfect accuracy in each of the six possible tasks. Thus, to render this setting slightly more complicated, we add salt-and-pepper noise to 15% of the image pixels. With the presence of noise, the classification of the object scale (4 classes) becomes challenging. Hence, we have chosen it as the primary task for our experiments. The number of subspaces k is set to 10 as in [13].

As described in Section 2.2, we cluster the most disentangled subspace (not considering the one dedicated to principal task) according to the loss of disentanglement. The HDBSCAN minimum cluster size hyperparameter is set to 2% of the number of data points N. In this experiment, the subspace chosen for the clustering coincides with the one encoding the information regarding the hue of the object (10 classes). Given the optimal disentanglement on 3D Shapes, the auxiliary labels generated by the clustering procedure almost perfectly match the ground-truth object hue labels, having homogeneity and completeness scores of 0.999.

We feed the noisy 3D Shapes images and the enriched label set into an MTL hard parameter-sharing architecture with a VGG16 [44] as the backbone and compare Single-Task Learning (STL) vs MTL. For this comparison, we need to perform a train-test split on 3D Shapes, which is non-trivial since the possible combinations of the latent factors in the dataset are present exactly once. Therefore, we split the dataset based on the floor and wall hue labels, allocating the images that contain 5 of the 10 values for both factors only to the test set, resulting in a 75-25 train-test split. On the principal task, MTL achieves an accuracy of **0.889**, outperforming the 0.125 obtained by STL by a large margin, *i.e.* **+0.746**.

Why return to image space for MTL? One may ask why we did not work directly in the latent feature space found by the disentanglement procedure. We did some preliminary experiments in this direction, but they yielded inconclusive results and raised implementation issues that are beyond the scope of this paper. The reason is that most MTL frameworks for image classification require convolution, which is not well-defined for feature vectors living in the latent space. Another reason is that **Detaux** works at a representation level, regardless of any classification aim induced by a specific classification framework. Its sole purpose is to reveal, together with the subspace dedicated to the principal task determined by the initial labels, other orthogonal complementary subspaces, which can be assumed as tasks if they admit clustering. The output of **Detaux** is an enriched set of labels that can be exploited with any MTL model. In addition, **Detaux** allows us to visualize and interpret the disentangled subspaces since it reconstructs the images. This procedure allowed us to understand that, in the toy example in 3D Shapes [3], the additional task corresponds to the hue of the object (one of the generative factors). Unfortunately, in more complex real cases, clear interpretation becomes more challenging, barely revealing the gender as an additional task in the FACES [11] benchmark. In the other cases, we had no clue. However, it should be noted that we focused on producing a framework that transforms a single task classification problem into a MTL one. We left eventual interpretability analyses for future work.

Why only use a single auxiliary task? In our experiments, we always use a single auxiliary task extracted from the most disentangled subspace (excluding the one allocated for the principal task). We made this choice to be able to test our research question - can disentanglement help us discover at least one subspace from which to extract a good auxiliary task? - while keeping the presentation of the various stages of the pipeline as straightforward as possible. Furthermore, the number of additional tasks places a non-trivial computational burden on the parameter-sharing models we implement for MTL. The scalability of such models is an interesting research direction, which we believe is beyond the scope of this work. Hence, the use of more auxiliary tasks is deferred to future work.

Are the MTL results statistically significant? To empirically validate if the results presented in Table 1 are statistically significant, we focus on the SVHN [40] dataset, where only MTI + **Detaux** outperforms the STL baseline. Therefore, we compare with the best competitor, MAXL [28], over five different seeds. For comparison, we conducted a two-sample t-test on the results to check if the means are significantly different from each other. Considering a significance level of 0.05, we obtain a p-value of 0.0004, which confirms that the results are significant. On average (over these five runs), MTI + **Detaux** reports a 1.1% gain in accuracy compared to MAXL.

What does disentanglement look like from a qualitative perspective? Figure 3 and Figure 4 provide a qualitative perspective of the disentanglement on the 3D Shapes and FACES datasets. In both visualizations, the outermost columns (far left and far right) represent the two images

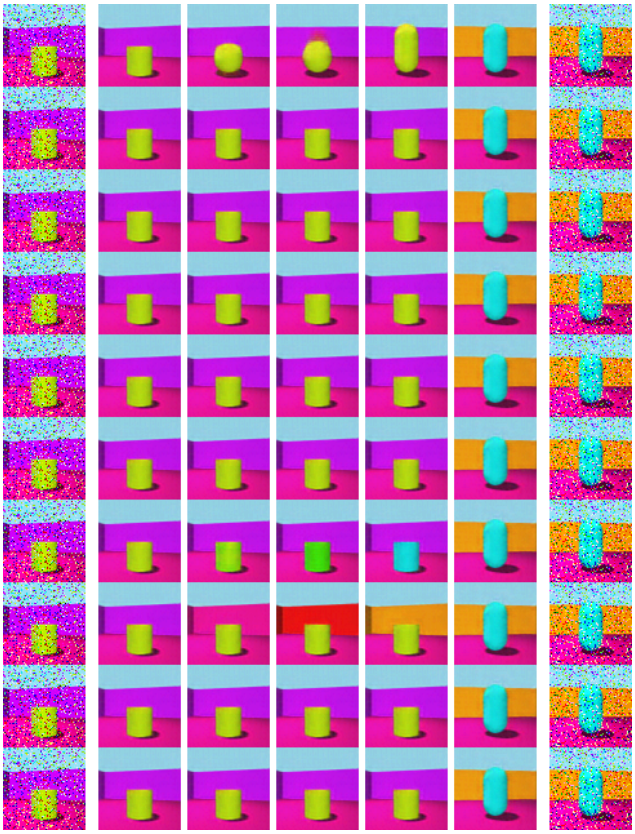

Figure 3. Visual interpretation of the disentanglement procedure on 3D Shapes with noisy input. The disentanglement model factorizes the representation and forces the principal task (*i.e.*, the object scale) in the first subspace, albeit with other factors of variation. The remaining factors (floor, wall, and object hues) are all disentangled in different subspaces and can be used to discover additional auxiliary tasks. Best viewed in color.

composing an input image pair, respectively. The adjacent columns (near the outermost) depict the reconstruction of the images. The three central columns display variations corresponding to specific factors encoded in individual subspaces. This is done by linearly interpolating between the representations of the pair and then reconstructing the result. Each row highlights a distinct subspace, showcasing how different generative factors are disentangled and isolated for targeted analysis.

In Figure 3, we can see how the disentanglement model factorizes the representation and forces the principal task (*i.e.*, the scale of the object) in the first subspace, although with other factors of variation. Furthermore, we can see that setting a higher number of subspaces than generative factors is not an issue, since it is possible for the model to collapse the variation in certain subspaces.

Figure 4 shows how the disentanglement procedure behaves when used on real data. Specifically, it becomes clear

Figure 4. Visual interpretation of the disentanglement on FACES. With real data, it becomes more difficult to factorize and visualize true generative factors. The naked eye can definitely realize that the variations between the image pair are contained in different subspaces. The first row shows the effect of our supervised oracle, which forces the principal task (*i.e.*, the person's facial expression) in the first subspace. At the same time, other variations arise in the other subspaces, allowing us to mine for auxiliary tasks.

that it is more difficult to factorize and visualize true generative factors. One can notice how only the eyes and mouth, related to smiling and being happy, are altered, while the rest of the face remains almost identical. In the second row, we can see a candidate auxiliary task, where the subject's gender seems to change and display different traits. These traits are indeed diverse from the ones dealing with the change in emotion, isolated in the first subspace, showing how we can extract orthogonal auxiliary tasks.

Is disentanglement crucial for auxiliary task discovery? (cont.d) Figure 5 highlights from a qualitative point of view the significance of disentanglement for discovering auxiliary tasks. The visualizations show the feature spaces of the FACES dataset in two different settings. Subfigure *(a)* illustrates the entangled feature space, where representations remain highly mixed, leading to less discernible clusters. Conversely, subfigure *(b)* depicts the most disentangled subspace, where the features are clearly grouped into distinct and interpretable groups. The high-dimensional feature representations are reduced to 3D space using PCA, and the clusters are identified using the HDBSCAN [4] algorithm. The evident separation in the disentangled subspace underscores its importance for auxiliary task min-

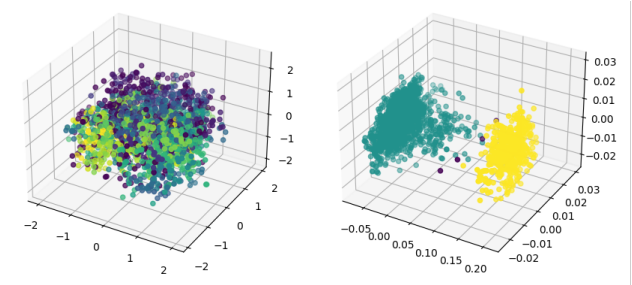

Figure 5. 3D visualization of the discovered auxiliary task in the entangled autoencoder feature space *(a)* and the most disentangled subspace *(b)*, on the FACES dataset. The high-dimensional representations are projected to 3D space using Principal Component Analysis (PCA). Different colors mean different clusters found by HDBSCAN. The representation in *(a)* is highly entangled, while the one in the disentangled representation space *(b)* displays a clear and reasonable grouping. Best viewed in color.

Table 2. Classification accuracy on the FACES and Cars datasets when using different clustering algorithms to generate the auxiliary task labels of **Detaux**. MTL results are obtained using HPS [5]. In parentheses, the change in performance over STL.

	Clustering	**FACES [11]** ↑	**Cars [22]** ↑
STL	–	0.915	0.711
MTL	HDBSCAN [4]	0.951	0.789
MTL	KMeans [29]	0.953 (+0.038)	0.789 (+0.078)
MTL	KMeans++ [1]	0.934 (+0.019)	0.790 (+0.079)
MTL	MeanShift [8]	0.963 (+0.048)	0.783 (+0.072)

Table 3. Normalized and Adjusted Mutual Information between the principal and auxiliary task labels generated using **Detaux**.

Dataset	**Normalized MI** ↓	**Adjusted MI** ↓
FACES [11]	0.1405	0.1390
CIFAR-10 [23]	0.0033	0.0031
SVHN [40]	0.0033	0.0030
Cars [22]	0.0311	0.0085

ing. These results also emphasize from a qualitative point of view that disentanglement not only simplifies representation learning but also facilitates structured auxiliary task discovery.