

IS DISENTANGLEMENT ALL YOU NEED? COMPARING CONCEPT-BASED & DISENTANGLEMENT APPROACHES

**Dmitry Kazhdan^a, Botty Dimanov^a, Helena Andres Terre^a,
Mateja Jamnik^a, Pietro Liò^a, Adrian Weller^{a, b}**

^aThe University of Cambridge, Cambridge, UK

^bThe Alan Turing Institute, London, UK

{dk525, ha376}@cam.ac.uk

ABSTRACT

Concept-based explanations have emerged as a popular way of extracting human-interpretable representations from deep discriminative models. At the same time, the disentanglement learning literature has focused on extracting similar representations in an unsupervised or weakly-supervised way, using deep generative models. Despite the overlapping goals and potential synergies, to our knowledge, there has not yet been a systematic comparison of the limitations and trade-offs between concept-based explanations and disentanglement approaches. In this paper, we give an overview of these fields, comparing and contrasting their properties and behaviours on a diverse set of tasks, and highlighting their potential strengths and limitations. In particular, we demonstrate that state-of-the-art approaches from both classes can be data inefficient, sensitive to the specific nature of the classification/regression task, or sensitive to the employed concept representation.

1 INTRODUCTION

Recent years have witnessed a dramatic increase in work on Explainable Artificial Intelligence (XAI) approaches, exploring ways of improving the explainability of deep learning models, and making them more transparent and trustworthy (Guidotti et al., 2018; Carvalho et al., 2019; Adadi & Berrada, 2018). A novel strand of XAI approaches relies on *concept-based* explanations, presenting explanations in terms of higher-level, human-understandable units (e.g., a prediction for an image of a car may rely on the presence of wheel and door concepts) (Kim et al., 2018; Kazhdan et al., 2020; Koh et al., 2020; Chen et al., 2020). At the same time, recent work on *disentanglement learning* aims to learn input data representations that have useful properties, such as better predictive performance (Locatello et al., 2019), fairness (Creager et al., 2019), and interpretability (Adel et al., 2018). These approaches assume that high-dimensional real-world data (e.g., images) is generated from a relatively-smaller number of *latent factors* of variation (Bengio et al., 2013).

Despite their overlapping goals and potential synergies, these two fields are typically not considered together. In this paper, we give an overview of these two fields and argue that concepts are a type of interpretable factors of variations. We conduct a systematic comparison between disentanglement learning and concept-based explanation approaches, showcasing their limitations and trade-offs to underscore their underlying assumptions and causes of failures due to training, labelling, and choice of objective functions.

We present a comprehensive *library*¹, implementing a range of state-of-the-art concept-based, and disentanglement learning approaches. Using this library, we compare approaches across a variety of tasks and datasets. In particular, we demonstrate that state-of-the-art approaches can be data inefficient, sensitive to the specific nature of the end task, or sensitive to the nature of the concept values. All experimental setups will be released as part of the library, and will serve as a foundation for future research aiming to address the limitations of these approaches.

¹<https://github.com/dmitrykazhdan/concept-based-xai>

2 STATE OF THE ART: ANALYSIS AND LIMITATIONS

Concept Bottleneck Models (CBMs): CBMs (Koh et al., 2020; Chen et al., 2020) assume that the training data consists of inputs \mathbf{x}_i , task labels y_i , and additional *concept annotations* \mathbf{c}_i : $(\mathbf{x}_1, \mathbf{c}_1, y_1), \dots, (\mathbf{x}_N, \mathbf{c}_N, y_N)$. CBMs are models which are interpretable by design, separating input processing into two distinct steps: (i) predicting concepts from the input, and (ii) predicting task labels from the concepts. These can be represented as a concept predictor function g_{cp} , and a label predictor function f_{lp} , such that $f_{lp}(g_{cp}(\mathbf{x})) : \mathcal{X} \rightarrow \mathcal{C} \rightarrow \mathcal{Y}$, where $\mathcal{X}, \mathcal{C}, \mathcal{Y}$ denote the input, concept, and output representations, respectively. Further details can be found in Appendix A.2.1.

Limitations: (i) CBMs typically require a large number of concept annotations during training, making them data inefficient. (ii) If the downstream task is dependent on information not contained in the provided concepts, then CBMs are either incapable of accurately predicting task labels, or their concept representations contain information “impurities” that are not related to the concepts (see Appendix A.2.1 for further details). The former hinders performance, while the latter hinders interpretability, as it becomes unclear whether a model is using concept or non-concept information.

Post-hoc Concept Extraction (CME): CME approaches (Kazhdan et al., 2020; Yeh et al., 2019; Kim et al., 2018) extract concepts from pre-trained models. Given a trained model $f(\mathbf{x}) = y$, CME relies on f to obtain a feature extractor $\phi(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{H}$, computing a higher-level representation of the input. Using this feature extractor, CME extracts models $f_{lp}(g_{cp}(\phi(\mathbf{x}))) : \mathcal{X} \rightarrow \mathcal{H} \rightarrow \mathcal{C} \rightarrow \mathcal{Y}$. Similarly to CBMs, these models compute a concept representation using g_{cp} , and then use it to compute the task labels via f_{lp} . Importantly, instead of using the raw input \mathbf{x} , these methods rely on a reduced representation returned by ϕ , typically making them more data-efficient. Further details can be found in Appendix A.2.2.

Limitations: (i) CME approaches assume that concepts can be reliably predicted from the hidden space of a trained model (i.e., that the representation computed by ϕ is informative of the concepts). Thus, CME requires a powerful pre-trained model in order to extract concepts reliably. (ii) Existing CME approaches either extract concepts in a semi-supervised fashion (Kazhdan et al., 2020), which requires the concepts to be known beforehand, or in an unsupervised fashion (Yeh et al., 2019), which requires the extracted concepts to be manually inspected and annotated afterwards.

Unsupervised Disentanglement Learning: Disentanglement learning approaches, such as VAEs (Kingma & Welling, 2014; Higgins et al., 2016), assume that input data \mathbf{x} is generated by a set of causal latent factors \mathbf{z} , sampled from a distribution $p(\mathbf{z})$, such that the input \mathbf{x} is sampled from $p(\mathbf{x}|\mathbf{z})$. Consequently, they aim to *automatically* learn mappings between the input data \mathbf{x} and lower-dimensional space \mathbf{z} , that describes the corresponding factors of variations.

Limitations: Recently, Locatello et al. (2019) showed that unsupervised VAEs can learn infinitely many different possible latent factor solutions that are not guaranteed to learn factors with desirable properties, such as human-interpretability, fairness, or high downstream task performance, without explicit supervision.

Weakly-supervised Disentanglement Learning: Weak supervision can be used to induce desirable properties on the learned factors, where a (small) set of data-points is labelled with ground-truth latent factor annotations (Locatello et al., 2020). Importantly, knowledge of the ground-truth data generative process is not available in most practical scenarios. Thus, a human annotator usually relies on domain knowledge of the task to provide labels of his/her *perception* of the factors of variation, instead of the ground-truth latent generative factors (i.e., factors of variation used by the environment). Consequently, we argue that these labels correspond to *concepts* (i.e., human perceived factors of variation dictated by domain knowledge). Hence, the latent factors \mathbf{z} used in weakly-supervised disentanglement learning can be seen as concepts \mathbf{c} . Further details can be found in Appendix A.2.3.

Limitations: (i) The weak supervision requirement of WVAE approaches imply that a user must have some knowledge of the underlying latent factors. (ii) VAE-based approaches predominantly rely on a combination of reconstruction loss and regularisation loss (the latter encourages disentanglement of the latent space). This implies that WVAEs may exhibit *bias*, since concepts having a larger variability in the input space (which we refer to as *loud* concepts) will have a relatively larger effect on the reconstruction loss, and will therefore be learned first. This has important fairness implications, particularly with respect concepts such as colour, as will be explored later on.

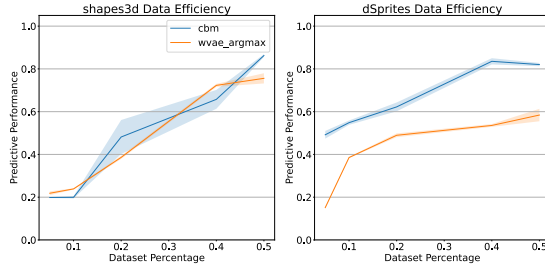


Figure 1: Average concept predictive accuracy (y-axis) of WVAE and CBM for shapes3D (left) and dSprites (right) across a progressively larger number of available concept annotations as a percentage of the total number of concept annotations (x-axis).

3 EXPERIMENTS

Datasets We evaluate the compared methods on datasets commonly-used for benchmarking disentanglement learning approaches (Locatello et al., 2020), in which the underlying generative factors describe all possible data variations: *dSprites* (Matthey et al., 2017), and *shapes3D* (Burgess & Kim, 2018). These datasets allow fine-grained control over the utilised concept representations and higher-level tasks, and have been previously used in work on concept explanations, including (Kazhdan et al., 2020). Further details regarding the datasets can be found in Appendix A.1.

Benchmarks We compare and evaluate the following state-of-the-art approaches described in Section 2: multi-task CBMs (Koh et al., 2020), CME with gradient boosted tree (GBT) (Hastie et al., 2009) concept extraction functions (Kazhdan et al., 2020), unsupervised VAEs, and Adaptive-ML-VAE (WVAEs) Locatello et al. (2020), for which weak supervision is provided by feeding in pairs of data samples, differing in only a few concepts at a time. While CME and CBM predict concepts explicitly, in the case of WVAEs we train a GBT, as in (Locatello et al., 2020), to predict concept values from the latent representation. Further details regarding these benchmarks can be found in Appendix A.2.

4 RESULTS & DISCUSSION

Data Efficiency As discussed in Section 2, CBMs assume that all input data-points are explicitly labelled with concept annotations (i.e. have the associated concept labels for every data point), while WVAEs assume this information is provided implicitly via pairs of points. In practice, obtaining concept annotations may be expensive or infeasible, especially for tasks with a large number of concepts. Consequently, it is important to consider the amount of annotations required by these approaches for accurate concept extraction (we refer to this as *data efficiency*).

In this experiment we compare the amount of supervision necessary to extract concept representation for: (1) CBM and (2) WVAE. We train both models with progressively smaller labelled datasets containing labels of *all* concepts, observing the corresponding change in concept predictive accuracy, averaged over all concepts. Figure 1 demonstrates that the concept predictive accuracy for both approaches decreases significantly with the labelled dataset sizes, and that both CBMs and WVAEs require almost 40% of all training points to be labelled to achieve above 60% average predictive accuracy. Overall, CBMs are more data efficient than WVAEs, which is likely due to the fact that they are explicitly trained to optimise for concept predictive accuracy. Importantly, unsupervised VAEs do not rely on concept annotations at all, and CME is highly sensitive to the end task (as will be shown below). Thus, these approaches are not included in this setup.

Concept-to-Task Dependence In contrast to both CBMs and WVAEs, CME is explicitly designed to be data efficient. However, CME relies on the hidden representation quality of its corresponding pre-trained model, as discussed in Section 2. Here, we compare the concept predictive accuracy of CME across tasks with a varying dependence on the provided concepts, as shown in Figure 2. In all cases, the CME models were trained with merely 500 labelled datapoints (approximately 1% of the concept annotations). We also provide results for CBM models trained on the entire datasets of

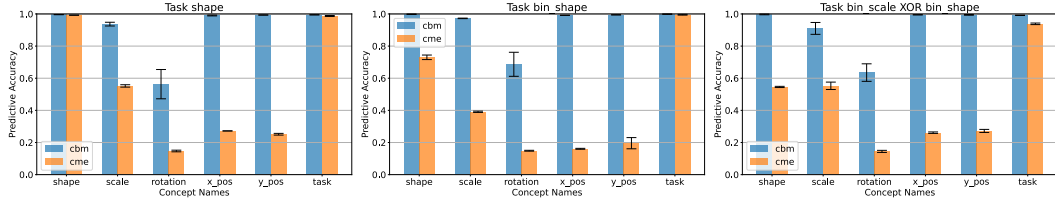


Figure 2: Comparison between the per-concept and downstream task predictive accuracy for CME and CBM across tasks of varying dependence on the concept values: full dependence on one concept (left), partial dependence on one concept (center), partial dependence on two concepts (right). For every subplot, the rightmost column (Task) represents the predictive performance for the task labels.

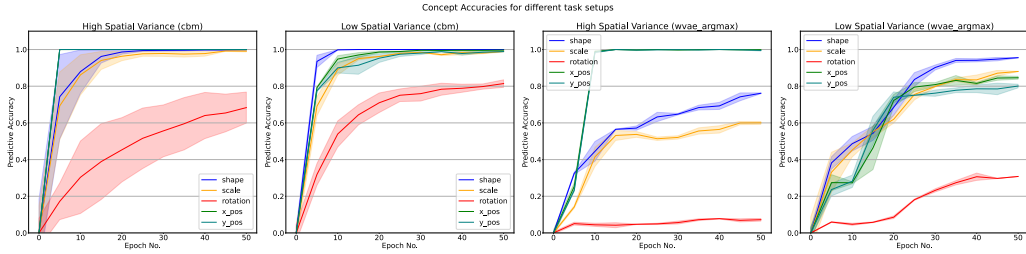


Figure 3: Comparison between different concept loudness setups. ‘High Spatial Variance’ refers to a dataset having x_pos and y_pos concepts significantly louder than the rest, whereas ‘Low Spatial Variance’ refers to a dataset where these concepts have similar loudness to the other concepts. The subfigures on the left show the performance of CBM, whilst subfigures on the right show performance of WVAE. Importantly, the other 3 concepts are significantly affected by the loudness of x_pos and y_pos in the case of WVAE.

concept annotations (which defines an upper bound for the concept predictive accuracy). Further details can be found in Appendix A.3.1 and Appendix B.1.

Firstly, Figure 2 demonstrates that CME successfully extracted the concept information necessary to achieve high predictive accuracy on the end task (indicated by the high ‘Task’ performance on all three tasks). However, concept predictive accuracy progressively declined as the task labels had progressively less reliance on the specific concept values. Thus, the CME predictive performance of each concept is highly dependant on how much this concept affects the end task label.

Concept Variance Fragility As discussed in Section 2, WVAE approaches are implicitly biased toward concepts whose values have a higher variance in the input space (the *louder concepts*). Consequently, we hypothesise that it is harder and significantly slower for WVAEs to learn the *quieter* concepts. We compare the effect of *concept loudness* on the ability of WVAEs and CBMs to learn concept representations, by inspecting how the predictive accuracy for every concept changes over time as the model is trained. Figure 3 illustrates that relative “concept loudness” can have a significant effect on the WVAEs ability to learn concepts (i.e., the performance of quieter concepts degrades significantly). On the other hand, CBMs are significantly less affected by this phenomenon. Further details regarding this experimental setup can be found in Appendix A.3.3. Importantly, colour combinations can have an effect on concept loudness. Hence, a concept representing colour (e.g., the colour of an object) can vary in loudness, depending on the specific colour combinations represented by that concept. This is explored further in Appendix A.3.3 and Appendix B.2.

Conclusions We analyse existing concept-based and disentanglement learning approaches with respect to three important properties: *data efficiency*, *concept-to-task dependence*, and *concept loudness*. We demonstrate that WVAE and CBM approaches are data inefficient, whilst CME is sensitive to the dependence between the concepts and the end task. Additionally, WVAE can be sensitive to the relative loudness of concepts. Overall, we believe these findings can be used for selecting the most suitable concept extraction approach, as well as for highlighting important shortcomings of state-of-the-art methods, that should be addressed by future work.

ACKNOWLEDGEMENTS

AW acknowledges support from a Turing AI Fellowship under grant EP/V025379/1, The Alan Turing Institute under EPSRC grant EP/N510129/1 and TU/B/000074, and the Leverhulme Trust via CFI.

REFERENCES

- Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6, 2018.
- Tameem Adel, Zoubin Ghahramani, and Adrian Weller. Discovering interpretable representations for both deep generative and discriminative models. In *International Conference on Machine Learning*, pp. 50–59. PMLR, 2018.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35, 2013.
- Chris Burgess and Hyunjik Kim. 3d shapes dataset. <https://github.com/deepmind/3dshapes-dataset/>, 2018.
- Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.
- Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition, 2020.
- Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement. In *International Conference on Machine Learning*, pp. 1436–1445. PMLR, 2019.
- Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
- 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. 2016.
- Dmitry Kazhdan, Boty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. Now you see me (CME): concept-based model extraction. In Stefan Conrad and Ilaria Tiddi (eds.), *Proceedings of the CIKM 2020 Workshops co-located with 29th ACM International Conference on Information and Knowledge Management (CIKM 2020), Galway, Ireland, October 19-23, 2020*, volume 2699 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020. URL <http://ceur-ws.org/Vol-2699/paper02.pdf>.
- Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2673–2682. PMLR, 2018. URL <http://proceedings.mlr.press/v80/kim18d.html>.
- Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL <http://arxiv.org/abs/1312.6114>.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of Machine Learning and Systems 2020*, pp. 11313–11323. International Conference on Machine Learning, 2020.

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*, pp. 4114–4124. PMLR, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pp. 6348–6359. PMLR, 2020.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. <https://github.com/deepmind/dsprites-dataset/>, 2017.

Chih-Kuan Yeh, Been Kim, Sercan O Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *arXiv preprint arXiv:1910.07969*, 2019.

A IMPLEMENTATION

A.1 DATASETS

A.1.1 DSPRITES

dSprites (Matthey et al., 2017) consists of 2D 64×64 pixel black-and-white shape images, procedurally generated from all possible combinations of 6 ground truth independent concepts (color, shape, scale, rotation, x and y position) for a total of $1 \times 3 \times 6 \times 40 \times 32 \times 32 = 737280$ total images. Table 1 lists the concepts and their corresponding values, while Figure 4 (a) presents a few examples.

Table 1: dSprites concepts and values

Concept name	Values
Color	white
Shape	square, ellipse, heart
Scale	6 values linearly spaced in $[0.5, 1]$
Rotation	40 values in $[0, 2\pi]$
Position X	32 values in $[0, 1]$
Position Y	32 values in $[0, 1]$

A.1.2 3D SHAPES

The 3dshapes (Burgess & Kim, 2018) dataset comprises of 64×64 pixel coloured 3D shape images, generated from all possible combinations of six latent factors – floor hue, wall hue, object hue, scale, shape, and orientation. Table 2 lists the concepts, and corresponding values, while Figure 4 (b) presents some examples.

A.2 BENCHMARKS

A.2.1 CBMs

As discussed in Section 2, CBM-based approaches train models that process input data-points in two distinct steps: (i) predicting concepts from the input, and (ii) predicting the task labels from the concepts, which are represented as a concept predictor function g_{cp} , and a label predictor function

Table 2: 3dshapes concepts and values

Concept name	Values
Floor hue	10 values linearly spaced in $[0, 1]$
Wall hue	10 values linearly spaced in $[0, 1]$
Object hue	10 values linearly spaced in $[0, 1]$
Scale	8 values linearly spaced in $[0, 1]$
Shape	4 values in $[0, 1, 2, 3]$
Orientation	15 values linearly spaced in $[-30, 30]$

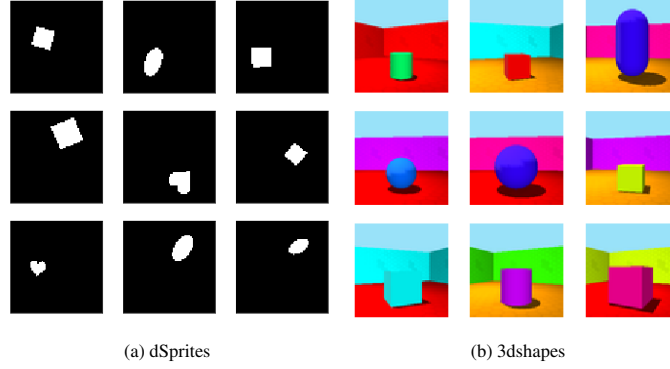


Figure 4: Sample images from dSprites (a) and 3dshapes (b).

f_{lp} , such that $f_{lp}(g_{cp}(\mathbf{x})) : \mathcal{X} \rightarrow \mathcal{C} \rightarrow \mathcal{Y}$, where $\mathcal{X}, \mathcal{C}, \mathcal{Y}$ denote the input, concept, and output spaces respectively.

Koh et al. (2020) introduce several different training regimes for learning f_{lp} and g_{cp} from the input data, concept, and task labels:

Independent: $\hat{g}_{ind} = \arg \min_g \sum_i \mathcal{L}_c(g(\mathbf{x}_i), \mathbf{c}_i)$, and $\hat{f}_{ind} = \arg \min_f \sum_i \mathcal{L}_y(f(\mathbf{c}_i), y_i)$

Sequential: $\hat{g}_{seq} = \arg \min_g \sum_i \mathcal{L}_c(g(\mathbf{x}_i), \mathbf{c}_i)$, and $\hat{f}_{seq} = \arg \min_f \sum_i \mathcal{L}_y(f(\hat{g}_{seq}(\mathbf{x}_i)), y_i)$

Joint: $\hat{g}_{joint}, \hat{f}_{joint} = \arg \min_{f,g} \sum_i \mathcal{L}_y(f(g(\mathbf{x}_i)), y_i) + \lambda \mathcal{L}_c(g(\mathbf{x}_i), \mathbf{c}_i)$. In this case the λ parameter trades off the importance of concept information contained in \mathbf{c} , with other information contained in the data which is relevant for predicting the labels y .

In the above, \mathcal{L}_c and \mathcal{L}_y refer to the concept prediction loss and task label loss, respectively. Further details can be found in Koh et al. (2020).

Importantly, the Independent and Sequential regimes require the task labels to be highly-dependent on the concept information, as it is the only information available to the label predictor. The Joint regime alleviates this issue, by allowing non-concept information to pass through the bottleneck as well. Whilst this modification can improve predictive performance of a CBM, it implies that concept and non-concept information are both mixed in the bottleneck of the CBM. This harms interpretability, as it becomes unclear how a CBM uses concept and non-concept information when producing task labels.

Multi-task: In the original CBM work (Koh et al., 2020), all concepts are assumed to be binary. Consequently, the concept predictor is implemented as a single-output model with a sigmoid output layer, having $|\text{neurons}| = |\text{concepts}|$ (i.e., one neuron per concept).

However, many other datasets (including those used in this work) contain multi-valued concepts. Consequently, we propose a new type of bottleneck model, namely a *multi-task CBM* model, where the bottleneck layer consists of several output heads, and each concept c_i is learned in a separate

output head. Each output head consists of n_i softmax units, where n_i represents the number of unique concept values for concept c_i . For example, dSprites has 6 concepts; hence, our multi-task model will have 6 different outputs, each with an individual sparse-categorical cross-entropy loss.

Importantly, the concept outputs are all concatenated and fed together into the concept-to-output model. Thus, similarly to the original CBM, the concepts are trained jointly. However, in contrast to the classical CBM setting, this set-up ensures that concept values of the same concept are mutually-exclusive (i.e., an image cannot be both a square and a heart), since they are now separated across multiple outputs. We found that this setup gave better performance than the standard CBM, for datasets with multiple multi-valued concepts (including dSprites and shapes3d).

A.2.2 CME

CME trains concept predictor and label predictor models in a semi-supervised fashion, by relying on the hidden space of a pre-trained model. Given a pre-trained model $f(\mathbf{x}) = y : \mathcal{X} \rightarrow \mathcal{Y}$ predicting task labels y from input \mathbf{x} , CME uses the hidden space \mathcal{H} of f to train a concept predictor model $\hat{g}_{cp}(\mathbf{h}) = \mathbf{c} : \mathcal{H} \rightarrow \mathcal{C}$, using a provided set of concept annotations \mathbf{c} (which is assumed to be small). The optimisation problem is defined as $\hat{g}_{cp} = \arg \min_g \sum_i \mathcal{L}_c(g(\phi(\mathbf{x}_i)), \mathbf{c}_i)$, where $\phi : \mathcal{X} \rightarrow \mathcal{H}$ is assumed to be a feature extractor function, returning activations of f at a given layer. Next, CME trains a task labeller function $\hat{f}_{lp}(\mathbf{c}) = y : \mathcal{C} \rightarrow \mathcal{Y}$, which is trained using $\hat{f}_{lp} = \arg \min_f \sum_i \mathcal{L}_y(f(\hat{g}_{cp}(\phi(\mathbf{x}_i))), y_i)$.

In the above, \mathcal{L}_c and \mathcal{L}_y refer to the concept prediction loss and task label loss, respectively. Further details, including the choice of layer(s) to use for ϕ , can be found in Kazhdan et al. (2020).

A.2.3 WEAKLY-SUPERVISED DISENTANGLEMENT LEARNING

The standard optimization objective adopted by many existing disentanglement learning approaches is shown in Equation 1. Here, the aim is to jointly learn an encoder model (representing $q_\phi(\mathbf{z}|\mathbf{x})$), and a decoder model (representing $p_\theta(\mathbf{x}|\mathbf{z})$), which model the underlying data generative process.

$$\max_{\phi, \theta} \mathbb{E}_{p(\mathbf{x})} [\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))] \quad (1)$$

As discussed in Section 2, work in Locatello et al. (2019) showed that without supervision, unsupervised VAE-based approaches can learn an infinite amount of possible solutions, guided by implicit biases that are not explicitly controlled for. Consequently, a range of recent work explored methods to semi/weakly-supervised disentanglement learning.

In particular, work in Locatello et al. (2020) assumes that input data is provided as pairs of points $(\mathbf{x}_1, \mathbf{x}_2)$ during training, with every pair of points differing by only a few factors of variation at a time. Consequently, the optimization objective is modified as shown in Equation 2.

$$\begin{aligned} \max_{\phi, \theta} \mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2)} \mathbb{E}_{\hat{q}_\phi(\hat{\mathbf{z}}|\mathbf{x}_1)} \log p_\theta(\mathbf{x}_1|\hat{\mathbf{z}}) + \mathbb{E}_{\hat{q}_\phi(\hat{\mathbf{z}}|\mathbf{x}_2)} \log p_\theta(\mathbf{x}_2|\hat{\mathbf{z}}) \\ - \beta D_{KL}(\hat{q}_\phi(\hat{\mathbf{z}}|\mathbf{x}_1) || p(\hat{\mathbf{z}})) - \beta D_{KL}(\hat{q}_\phi(\hat{\mathbf{z}}|\mathbf{x}_2) || p(\hat{\mathbf{z}})) \end{aligned} \quad (2)$$

In the above, \hat{q} refers to a modified version of the approximate posterior q , defined as show in Equation 3, and a refers to an averaging function.

$$\begin{aligned} \hat{q}_\phi(\hat{z}_i|\mathbf{x}_1) &= a(\hat{q}_\phi(\hat{z}_i|\mathbf{x}_1), \hat{q}_\phi(\hat{z}_i|\mathbf{x}_2)), \quad \forall i \in \hat{S} \\ \hat{q}_\phi(\hat{z}_i|\mathbf{x}_1) &= \hat{q}_\phi(\hat{z}_i|\mathbf{x}_1), \quad \text{else} \end{aligned} \quad (3)$$

The update rule for $\hat{q}_\phi(\hat{z}_i|\mathbf{x}_2)$ is obtained in an analogous manner. In the above, set \hat{S} is the set of chosen latent factors, which is obtained by selecting latent factors with smallest $D_{KL}(\hat{q}_\phi(\hat{z}_i|\mathbf{x}_1) || \hat{q}_\phi(\hat{z}_i|\mathbf{x}_2))$. Further details can be found in Locatello et al. (2020).

A.3 EXPERIMENTAL DETAILS

A.3.1 CONCEPT-TO-TASK DEPENDENCE

dSprites For dSprites, we use a reduced version of the dataset introduced in Kazhdan et al. (2020), namely: we select 16 of the 32 values for *Position X* and *Position Y* (keeping every other value only), and select 8 of the 40 values for *Rotation* (retaining every 5th value) to decrease the 737280 samples to $3 * 6 * 8 * 16 * 16 = 36864$ samples.

We define 3 separate tasks: *shape*, *bin_shape*, and *bin_scale XOR bin_shape*. The *shape* task consists of predicting the value of the shape concept (i.e. $y = c_{shape}$). The *bin_shape* task consists of predicting whether the shape concept is a heart or not (i.e., the concept is now binarized into two values, instead of three: $y = ((c_{shape} == 0) \text{ OR } (c_{shape} == 1))$). The *bin_scale XOR bin_shape* task is defined as: $y = ((c_{shape} == 2) \text{ XOR } (c_{scale} > 2))$

shapes3d For shapes3d, we relied on a reduced version of the dataset: we select every other value from all concepts, except *shape*, for which we keep all values, for a total of: $5 * 5 * 5 * 4 * 4 * 7 = 14000$ samples

Similarly to dSprites, we define 3 tasks: The *shape* task consists of predicting the value of the shape concept (i.e. $y = c_{shape}$). The *bin_shape* task is defined as $y = (c_{shape} \geq 2)$ (i.e., whether it's the first two values, or the last two values). The task *bin_scale XOR bin_shape* task is defined as: $y = ((c_{shape} \geq 2) \text{ XOR } (c_{scale} > 2))$.

For both datasets, the 3 tasks represent a gradual progression from a task which loses no information about one of the concepts (Task 1 is equivalent to the *shape* concept), to a task which loses information about all concepts (Task 3 only loosely depends on the exact *shape* and *scale* concept values).

A.3.2 DATA EFFICIENCY

For the data efficiency experiments, we rely on the same dataset setups as in the Concept-to-Task Dependence experiments, for both dSprites and shapes3d.

A.3.3 CONCEPT VARIANCE FRAGILITY

dSprites We define two tasks for the spatial variance experiment. For the first one, we sub-select the following concept values: *shape* $\in [0, \dots, 3]$, *scale* $\in [0, \dots, 5]$, *range* $\in [0, \dots, 19]$, *x_pos* $\in [0, 10, 20, 30]$, *y_pos* $\in [0, 10, 20, 30]$. For the second task, we sub-select the following concept values: *shape* $\in [0, \dots, 3]$, *scale* $\in [0, \dots, 5]$, *range* $\in [0, \dots, 19]$, *x_pos* $\in [0, \dots, 3]$, *y_pos* $\in [0, \dots, 3]$. Importantly, both tasks consist of exactly the same concepts and have the same number of values per concept. However, the input variance of the first task wrt. the *x_pos* and *y_pos* concepts is significantly larger than for the second task (i.e., *x_pos* and *y_pos* are relatively louder in the first setup), since the difference in the input space between the different concept values is now larger.

For the colour variance experiment, we introduce an extra *colour* concept to dSprites, which can take on two values. Consequently, we define three tasks, with the colour concept taking on different colour combinations. For all three tasks, we set the first value to be the colour *green* (RGB [0, 1, 0]). Next, we vary the value of the second colour to be *purple* (RGB [1, 0, 1]), *blue* (RGB [0, 0, 1]), and *turquoise* (RGB [0, 1, 0.85]). Importantly, these pairs of values gradually decrease in distance in the RGB space. For the remaining concepts, all three tasks had the same sub-selected concept values: *shape* $\in [0, \dots, 3]$, *scale* $\in [0, \dots, 2]$, *range* $\in [0, 2, 4, 6]$, *x_pos* $\in [0, 2, 4, \dots, 16]$, *y_pos* $\in [0, 2, 4, \dots, 16]$. Thus, this experiment introduces tasks in which the loudness of the colour concept is gradually reduced.

shapes3d We define three tasks with the following sub-selected values:

- **Task 1:** *floor_hue* $\in [0, 2, \dots, 8]$, *wall_hue* $\in [0, 2, \dots, 8]$, *object_hue* $\in [0, 2, \dots, 8]$, *scale* $\in [0, 2, \dots, 6]$, *shape* $\in [0, \dots, 3]$, *orientation* $\in [0, \dots, 15]$
- **Task 2:** *floor_hue* $\in [0, \dots, 10]$, *wall_hue* $\in [0, \dots, 10]$, *object_hue* $\in [0, \dots, 10]$, *scale* $\in [0, \dots, 3]$, *shape* $\in [0, \dots, 3]$, *orientation* $\in [0]$

- **Task 3:** $floor_hue \in [0, \dots, 10]$, $wall_hue \in [0, \dots, 10]$, $object_hue \in [0, \dots, 10]$, $scale \in [4, \dots, 7]$, $shape \in [0, \dots, 3]$, $orientation \in [0]$

Importantly, all three tasks have the same total number of samples (16000 samples). Task 1 differs from Task 2 and Task 3 by having fewer values in the first five concepts, and more values in the last concept. Task 2 differs from Task 3 by using lower values for the *scale* concept, instead of higher values (hence, making it quieter).

B ADDITIONAL RESULTS

B.1 CONCEPT-TO-TASK DEPENDENCE

This section includes further experiments on Concept-to-Task Dependence. Figure 5 shows the concept-to-task dependence results for the shapes3d dataset.

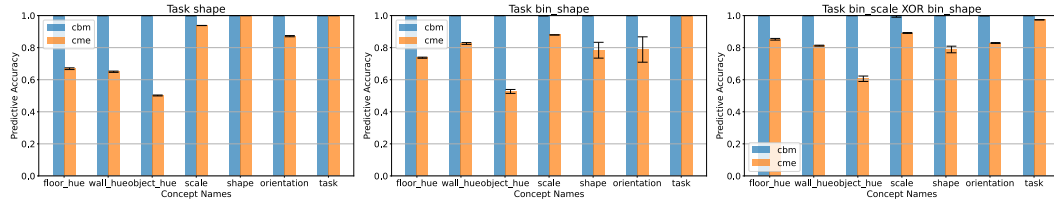


Figure 5: Comparison between the per-concept and downstream task predictive accuracy for CME and CBM across tasks of varying dependence on the concept values: full dependence on one concept (left), partial dependence on one concept (center), partial dependence on two concepts (right).

B.2 CONCEPT VARIANCE FRAGILITY

This section includes further experiments on Concept Variance Fragility. Figure 6 shows concept variance fragility for the shapes3d dataset.

Figure 7 shows concept variance fragility for the dSprites colour experiment.

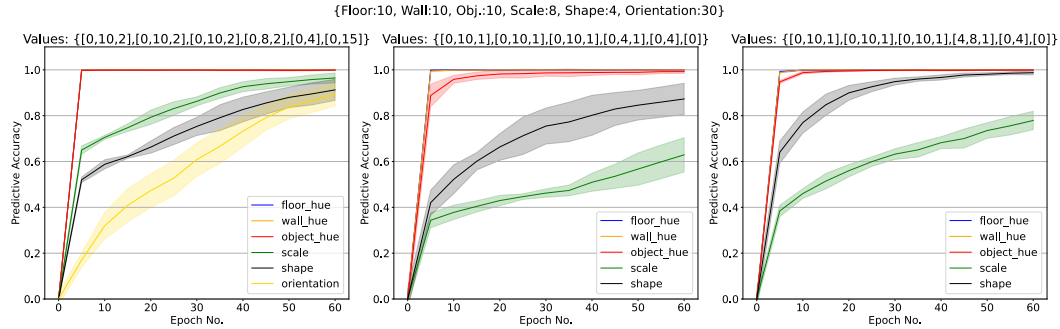


Figure 6: Comparison between different concept loudness setups for the shapes3d tasks.

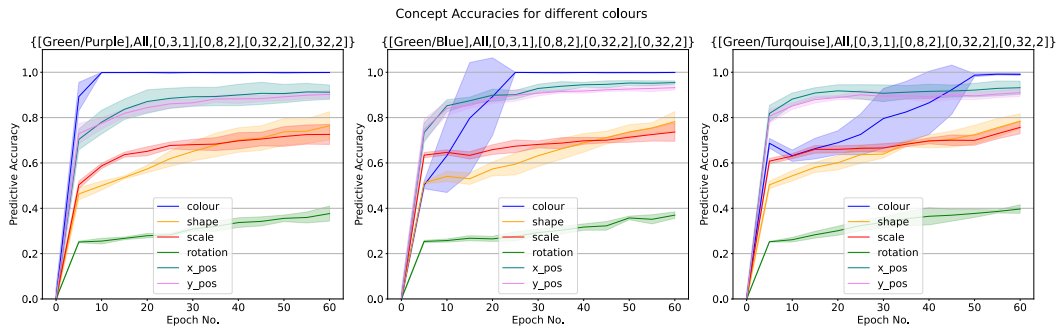


Figure 7: Comparison between different concept loudness setups for the dSprites colour tasks. Overall, the performance of the colour concept is significantly affected by its specific colour combinations.