Concept Correlation and Its Effects on Concept-Based Models

Lena Heidemann, Maureen Monnet, Karsten Roscher
Fraunhofer IKS
Munich, Germany
{firstname.lastname}@iks.fraunhofer.de

Abstract

Concept-based learning approaches for image classification, such as Concept Bottleneck Models, aim to enable interpretation and increase robustness by directly learning high-level concepts which are used for predicting the main class. They achieve competitive test accuracies compared to standard end-to-end models. However, with multiple concepts per image and binary concept annotations (without concept localization), it is not evident if the output of the concept model is truly based on the predicted concepts or other features in the image. Additionally, high correlations between concepts would allow the model to predict a concept with high test accuracy by simply using a correlated concept as a proxy. In this paper, we analyze these correlations between concepts in the CUB and GTSRB datasets and propose methods beyond test accuracy for evaluating their effects on the performance of a concept-based model trained on this data. To this end, we also perform a more detailed analysis on the effects of concept correlation using synthetically generated datasets of 3D shapes. We see that high concept correlation increases the risk of a model's inability to distinguish these concepts. Yet simple techniques, like loss weighting, show promising initial results for mitigating this issue.

1. Introduction

Using high-level concepts for explaining predictions of deep neural networks (DNNs) has gained increasing attention in recent years. While post-hoc methods like Testing with Concept Activation Vectors (TCAV) [9] try to explain a model's prediction without modifying the model or the training process, there have been recent efforts in building inherently interpretable concept models (e.g. [10, 12, 13, 3]). These methods try to enforce an interpretable intermediate layer, the so-called concept bottleneck, which outputs predefined concepts. The concept predictions are learned based on image-level concept annotations.

Besides interpretation, such inherent concept models also enable intervention during test-time by correcting potentially incorrect concept predictions. These corrections could additionally improve the accuracy on the main classification task. Concept models could also be helpful for evaluating if a certain class prediction seems plausible or should not be trusted. Such test-time intervention and plausibility checks are particularly useful in safety-critical domains like medical imaging or autonomous driving.

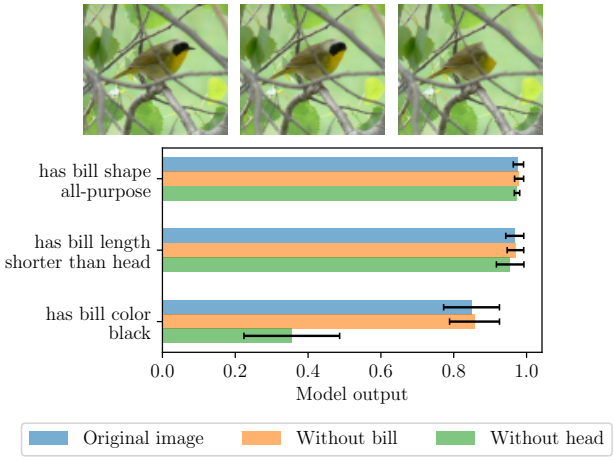


Figure 1: Output of a concept model trained on CUB for images where the bill or the whole head of the bird is removed.

Concept Bottleneck Models (CBMs) [10], as an example for inherently interpretable concept models, have achieved accuracies which are competitive with standard end-to-end models for image classification tasks. However, it is rather difficult to provide evidence that the concept predictions truly represent these concepts in the image. Each image is annotated with multiple concepts but without information on where in the image they are located. Therefore, it is theoretically possible for a model to leverage correlations in the dataset to predict certain concepts with high test accuracy without having actually learned what the concept looks like.

Figure 1 shows an example of this issue. Concepts describing the bill of a bird are still predicted with high confidence for an image where the bill is removed. For two concepts that is even true for an image where the whole head of the bird is not visible. This example shows that concept predictions may not be based on the concept in the image and for some not even based on features close to where the concept is located. This is particularly problematic for applications with ethical or safety-related implications, which typically rely on interpretable models.

In this paper, we therefore make the following contributions:

- We propose methods beyond test accuracy for evaluating concept-based models, in particular CBMs, with regard to whether a concept has truly been learned by the model, and show that only reporting the test accuracy for a concept model is usually not sufficient.
- We analyze concept correlations in two real-world datasets (CUB and GTSRB) and investigate their potential effects on concept-based models via synthetically generated datasets of 3D shapes. We find that high concept correlation may lead to the model learning only one of the concepts and using this to predict the other.
- Additionally, we present simple mitigation techniques. In particular, we train one model for each concept and weight the loss of images where only one of the concepts is present. This yields promising initial results.

2. Related Work

While the construction of concept-based models has received significant attention, little work has, to our knowledge, focused on whether the dataset used in the study actually permits concept learning, nor which impact the dataset's properties have on the learning process. In this section, we first give a brief overview on concept-based models, before outlining what has been done so far to highlight these pitfalls.

Research incorporating human-understandable concepts into black-box image recognition models boils down to two categories:

The first one consists of *post-hoc concept models*, where explanations for the network's predictions are found post training. These methods often make use of the latent space of a trained CNN to find mappings to one or a combination of predefined concepts [23, 5, 9, 24].

The second stream of research consists of *inherent concept models*, which already include the desired concepts in the learning phase, making these models interpretable by design. Instead of relying on properties the latent space may not have, inherent concept models focus on constraining the latter to account for a set of concepts, using a concept whitening layer [3], a bottleneck of intermediate concepts [10, 12, 13], or additional information such as image descriptions [21].

The use of automatically found concepts without the need for concept annotation has also been investigated in post-hoc methods [6, 4] and inherent models [2, 16]. In this work, we focus on methods which leverage image-level data annotations since this is where concept correlations become relevant. In particular, we focus on Concept Bottleneck Models (CBMs) [10].

Most criticism regarding inherent concept models relies on the fact that concepts are actually used as a proxy to learn the final label, and thus contain more information than the mere concept. Margeloiu et al. [15] show that this is the case for CBMs with jointly trained concept and task models. Mahinpei et al. [14] extend the claim to all models using soft concept representations, i.e., concepts that are represented with intermediate nodes corresponding to the confidence in the presence of the concept.

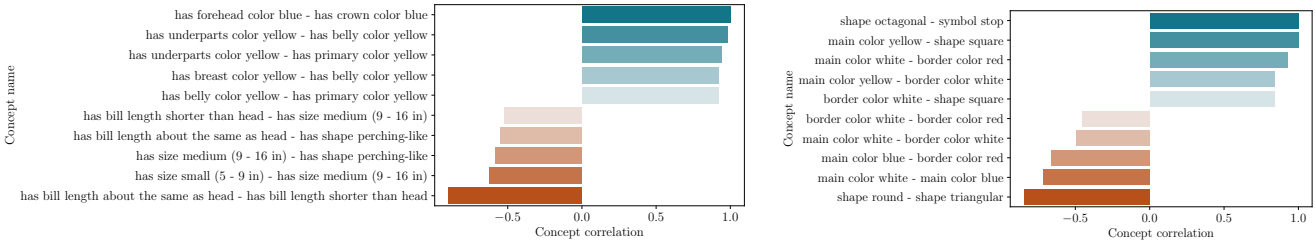
Using saliency maps, Margeloiu et al. [15] also show that the learned concepts do not appear to correspond to anything semantically meaningful.

We on the other hand take an upstream approach and analyze whether the concepts can be learned correctly in the first place, given a labeled dataset and the interplay of the concept correlations, without taking into account any downstream task model.

3. Concept Bottleneck Models

We are interested in the basic principles of concept-based models which rely on image-level annotations for learning predefined concepts which in turn are used as input for the main classification task. In our experiments we focus on Concept Bottleneck Models (CBMs) [10] as an example of this class of models. It follows a simple architecture but its performance is competitive with standard end-to-end models which makes it a suitable candidate for representing concept-based models for image classification.

Instead of training a model end-to-end from input to output class, CBMs allow by construction to first learn a set of concepts which are then used to predict the output class. To this end, two losses are defined, namely a task loss – designed to learn the final class – and a concept loss – enforcing concept learning. The concept model is designed as a single convolutional neural network (CNN) with multi-label outputs, while the task model often is a small multi-layer perceptron (MLP). These two models can be trained independently, sequentially, or jointly. The three training methods may differ in task error but achieve similar concept errors. Since our focus for these experiments is on concept learning, we only train the concept model, irrespective of



(a) CUB

(b) GTSRB

Figure 2: Concept pairs with the top and bottom 5 correlation values.

the task. We can infer that a better concept model will lead to a better task model since the concept model's output is the task model's input. This is particularly true, when both models are trained independently or sequentially.

The dataset needed to train such models is required to have image-level concept annotations, with a binary label (concept is present or not) for each defined concept and image input. While this type of annotation is relatively easy and cheap to obtain (as opposed to localized concept labels), the binary annotations may permit spurious correlations between concepts.

4. Evaluation Methods

In this section we introduce the evaluation methods used in the following experiments. These methods should help to evaluate whether a model has truly learned a concept or bases its prediction on other features in the image.

4.1. Concept Removal

A straight-forward approach to evaluating if a certain concept has been learned correctly would be to remove this concept from the image and observe if and how the prediction changes (see Figure 1). For real-world datasets this is usually only possible by editing images manually, which makes it more suited for an initial qualitative analysis. Yet in our experiments with synthetically generated datasets, we also perform a quantitative analysis over the whole test set, as the individual concepts can be easily removed in the image generation process. The associated metric we report is the concept removal accuracy which is defined as the number of samples for which the model's prediction changes from *present* to *not present* when the concept is removed over the number of all true positive samples.

4.2. Pointing Game

The Pointing Game [22] evaluates saliency maps in a quantitative manner. In addition to the saliency map, it requires the ground truth mask of a given concept. A sample is defined as a hit when the maximum point of the saliency map lies within the ground truth mask, and as a miss otherwise. The Pointing Game accuracy is then calculated as the number of hits over the number of all samples. In our experiments we use Guided Grad-CAM [17] to produce the saliency maps. However, due to the known limitations of saliency maps (e.g. [1]) the results of this evaluation method should be treated with caution and used in combination with other evaluation metrics.

4.3. Difference in Test Accuracy

In case ground truth concept masks are not available, we can compare the test accuracy of each concept on two different subsets of the test data. These subsets are specific to a pair of concepts: One subset contains images where both concepts are present or both are absent (dependent samples), while the other subset contains images where only one of the concepts is present (independent samples). We would expect these accuracies to be similar, as the prediction for a concept should not be influenced by the presence or absence of other concepts. The main drawback of this evaluation method is that, given a high correlation between two concepts, there might only be a few or no independent samples at all. Nevertheless, even with few samples the difference in test accuracy between these two subsets could serve as an indicator for whether a concept has been learned based on the presence of that concept or based on another concept.

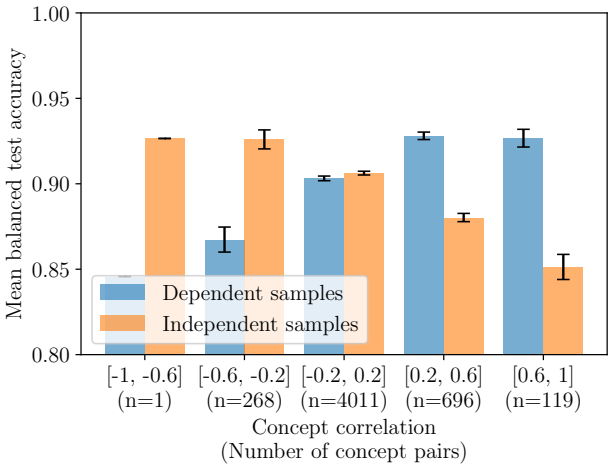
5. Experimental Setup

The concept models are trained on two real-world datasets, the Caltech-UCSD Birds-200-2011 (CUB) [20] and the German Traffic Sign Recognition Benchmark (GTSRB) dataset [18], and a number of synthetically generated datasets of 3D shapes. The CUB dataset comprises 11,788 images of 200 bird species with attribute annotations. GTSRB contains 50,000 images of 43 different classes of German traffic signs without concept-level annotations. Similarly to [11], we define 43 concepts for the GTSRB dataset, which consist of colors, shapes, numbers, and symbols, and assign them to the respective classes.

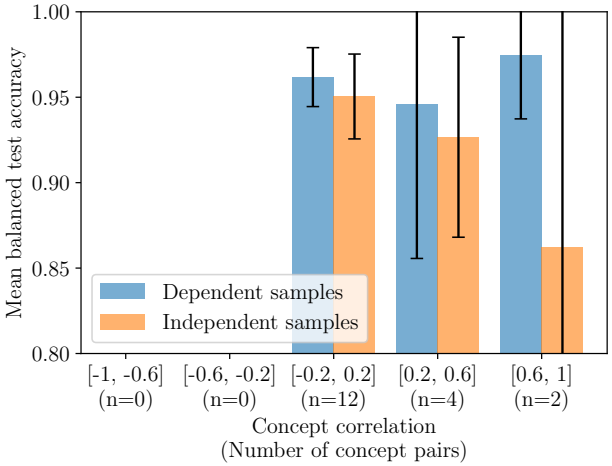
Since we want to investigate the impact of concept correlation, we also show the concept pairs with the highest and lowest correlation coefficients for each dataset, CUB and GTSRB, in Figure 2. The correlation is calculated as the Pearson correlation coefficient of the concept labels in the dataset. Not surprisingly, for CUB we see high correlations between the same attributes of different parts which are located close to each other (e.g. blue forehead & crown, yellow underparts & belly). Similarly, attributes which are contradicting, like a small and medium size of the bird, have a strong negative correlation. In the GTSRB dataset we have strong correlations between concepts which only belong to one specific traffic sign (e.g. the stop sign with an octagonal shape and the word 'stop') or often appear together irrespective of the class (e.g. main color white and border color red). Similar to CUB, we have high negative correlations for contradicting concepts, as well as concepts which rarely appear together, like main color blue and border color red. Please refer to the supplementary material for a full overview of the correlation coefficients for CUB and GTSRB.

In addition to real-world datasets, we also generate datasets of 3D shapes using the CLEVR dataset generation [8]. We define 6 shapes as concepts (cube, sphere, cylinder, cone, torus, icosphere) and vary their size, material, and color randomly. For a full 3D shapes dataset we generate 10,000 images, where each image contains 2 - 4 shapes.

For training on CUB, we closely follow the implementation of the original CBM paper, i.e., we use the same 112 binary bird attributes and fine-tune a pretrained Inception-v3 model [19]. For GTSRB and the 3D shapes datasets, we use a ResNet-18 model [7] to learn the concepts. If not stated otherwise, mean and standard deviation are reported based on 5 models trained with different seeds for CUB and GTSRB, and 3 models for 3D shapes due to the computational cost associated with training on this variety of datasets.



(a) CUB



(b) GTSRB

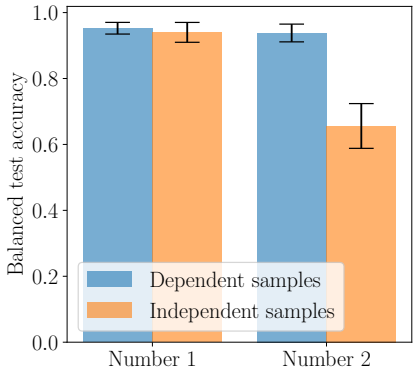
Figure 3: Mean balanced accuracy on the dependent and independent subsets of the test data for different correlation values. Error bars show the 95% confidence intervals.

6. Concept-level Evaluation of Concept Models

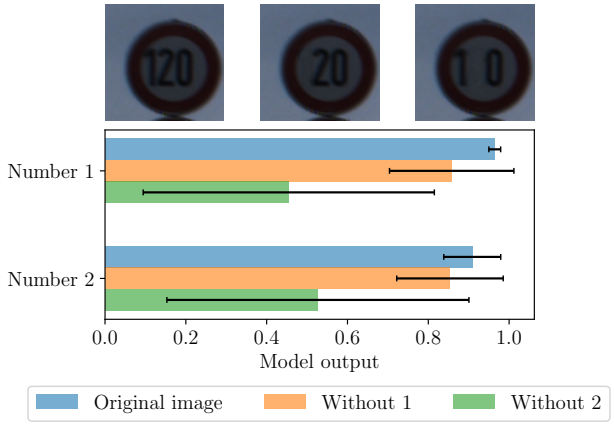
Reporting the test accuracy is usually not enough for evaluating concept models. We want to know whether each output of the concept model is truly based on the predicted concept or other concepts in the image. To this end, we apply methods from Section 4 to concept models trained on CUB and GTSRB. As ground truth masks are not available for the concepts in these datasets, we focus on the difference in test accuracy and examples of concept removal.

We report the difference in test accuracy between a subset of samples where both concepts are present/absent (dependent samples) and a subset where just one of the concepts is present (independent samples). In order to filter out effects due to an imbalance in the data, we calculate the balanced test accuracies. Figure 3 shows the results for CUB and GTSRB for different values of correlation. Although the sample sizes differ, for CUB we see a decrease of accuracy for independent samples with higher correlation and the inverse effect for high negative correlations. Therefore, the difference in balanced test accuracy between these subsets is the largest for concepts with a correlation close to 1 and close to -1.

Compared to CUB, there are fewer concepts in GTSRB and they are less shared among the classes. As a result, in many cases not enough dependent or independent samples are available for calculating an accuracy. In Figure 3b we can only plot 18 out of the 903 possible concept pairs, since these are the only pairs for which there are at least one positive and one negative sample for each concept in



(a) Difference in accuracy between dependent and independent subsets of the test data for concepts *number 1* and *number 2* of the GTSRB dataset.



(b) Model outputs for GTSRB images where concepts *number 1* and *number 2* are removed.

Figure 4: Evaluation of the concept pair *number 1* and *number 2* of GTSRB. Their correlation coefficient is 0.64.

each subset. However, even with these few samples and the resulting large confidence intervals, the results hint to a similar trend. The difference in balanced test accuracy seems to be larger for correlation values close to 1.

Due to the small number of concept pairs available for evaluation, we additionally perform a qualitative analysis of one correlated concept pair. We pick the pair with the highest correlation among those 18 available and show the difference in balanced accuracy and qualitative results for removing each concept (see Figure 4). Concept *number 2* has a lower test accuracy on the independent samples, while the accuracies for concept *number 1* are quite similar. Though by removing each concept individually, we see that *number 1* does not seem relevant for either concept predictions, as they barely differ from the predictions for the original image. Only the absence of *number 2* changes the predictions for both concepts. This seems to contradict the

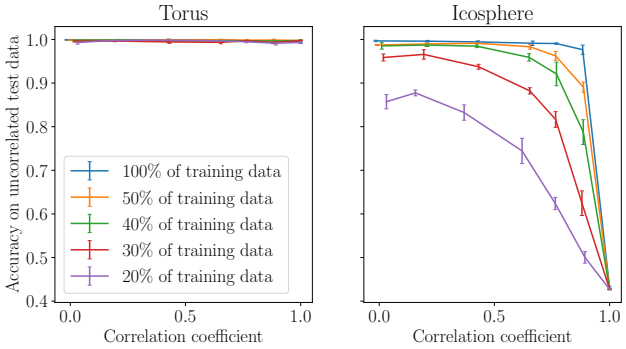


Figure 5: Test accuracy on an uncorrelated dataset for models trained on 3D shapes datasets with different values of correlation and different proportions of the training data.

results of the test accuracy differences. One reason for this might be more complex correlations and correlations with other concepts or features which are not captured by the pairwise concept correlations. In support of that, the large variation of predictions for the image without *number 2* also indicates that there are models where none of the two concepts seem to have an influence on the prediction. Rather other correlated features might be relevant for predicting these concepts.

7. Analyzing the Impact of Concept Correlation

In this section we perform a more in-depth analysis of concept correlation and its impact on concept model performance. To this end, we create datasets of 3D shapes with different degrees of correlation between concepts, which are 6 different shapes in this case. The reported correlation values might take seemingly arbitrary values but this is due to non-deterministic elements in the dataset generation process. Additionally, we generate a test set of uncorrelated data which enables a simple evaluation of the model's ability to distinguish all concepts. All datasets are designed to be balanced in order to reduce the effects of other factors on the results.

7.1. Varying the Degree of Concept Correlation

We first select a concept pair, namely *torus* and *icosphere*, and generate datasets with different degrees of correlation between the two concepts. All other concepts are uncorrelated and the datasets are balanced. Figure 5 shows the accuracies for both concepts on an uncorrelated test set for correlations ranging from 0 to 1. The test accuracy for the concept *torus* is not affected by an increasing correlation, whereas the *icosphere* accuracy drops for correlations of about 0.9 or higher. When we use less and less training data, this threshold decreases further and smaller values of

Concept pair with correlation = 1	Cube	Sphere	Cylinder	Cone	Torus	Icosphere
Cube & Sphere	66.1 ± 3.6	78.4 ± 3.6	97.6 ± 0.3	100.0 ± 0.0	100.0 ± 0.1	98.2 ± 0.3
Cube & Cylinder	70.5 ± 0.3	99.4 ± 0.1	74.4 ± 0.3	100.0 ± 0.0	99.9 ± 0.1	98.9 ± 0.4
Cube & Cone	45.1 ± 0.1	99.4 ± 0.3	$\mathbf{80.7 \pm 2.0}$	99.8 ± 0.1	100.0 ± 0.1	99.3 ± 0.5
Cube & Torus	44.0 ± 0.0	99.5 ± 0.2	$\mathbf{77.2 \pm 2.1}$	99.9 ± 0.1	100.0 ± 0.0	99.3 ± 0.2
Cube & Icosphere	66.2 ± 1.6	99.4 ± 0.5	98.3 ± 0.2	100.0 ± 0.1	99.9 ± 0.1	76.9 ± 1.6
Sphere & Cylinder	92.1 ± 2.2	71.7 ± 1.2	70.9 ± 1.2	99.9 ± 0.1	100.0 ± 0.1	98.3 ± 0.3
Sphere & Cone	99.7 ± 0.2	52.0 ± 1.7	99.6 ± 0.1	91.1 ± 1.7	99.9 ± 0.1	95.4 ± 2.0
Sphere & Torus	99.1 ± 0.3	43.3 ± 0.2	99.7 ± 0.2	100.0 ± 0.0	99.3 ± 0.2	$\mathbf{84.8 \pm 1.7}$
Sphere & Icosphere	98.9 ± 0.3	72.5 ± 0.6	99.6 ± 0.1	99.9 ± 0.1	99.9 ± 0.1	72.0 ± 0.6
Cylinder & Cone	$\mathbf{74.2 \pm 0.8}$	99.7 ± 0.3	44.5 ± 0.8	98.5 ± 0.8	99.9 ± 0.1	99.5 ± 0.4
Cylinder & Torus	$\mathbf{74.6 \pm 0.7}$	99.4 ± 0.4	46.9 ± 0.5	100.0 ± 0.0	99.3 ± 0.5	99.4 ± 0.2
Cylinder & Icosphere	95.6 ± 1.0	98.3 ± 0.7	70.3 ± 0.4	100.0 ± 0.1	100.0 ± 0.1	73.6 ± 0.4
Cone & Torus	99.2 ± 0.2	99.6 ± 0.2	99.7 ± 0.1	67.2 ± 2.0	77.5 ± 1.9	99.4 ± 0.4
Cone & Icosphere	99.6 ± 0.2	95.1 ± 1.2	100.0 ± 0.1	87.1 ± 4.6	100.0 ± 0.0	57.8 ± 4.6
Torus & Icosphere	99.3 ± 0.5	$\mathbf{82.3 \pm 2.9}$	99.7 ± 0.2	99.9 ± 0.1	99.5 ± 0.3	43.1 ± 0.3

Table 1: Concept accuracy (in %) on an uncorrelated test dataset for CBM concept models trained on different datasets. Each row denotes a dataset with a correlation of 1 between the two concepts. Accuracies for the correlated concepts are highlighted in gray. Accuracies of seemingly uncorrelated concepts which drop below 90% are marked in bold.

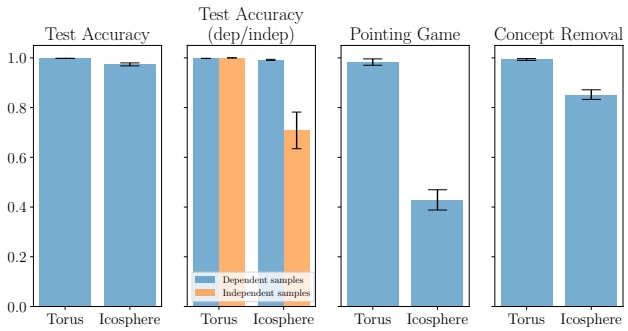


Figure 6: Overview of evaluation metrics for a 3D shapes dataset with a correlation of 0.89 between *torus* and *icosphere*, trained on 50% of the training data.

correlation already lead to a lower accuracy for *icosphere*. We see that high concept correlation can lead to a model only learning one of the concepts properly. Furthermore, the impact of correlation gets bigger the less training data is available. By comparing the accuracies of both concepts at a correlation value close to 0, it seems that the *icosphere* is more difficult to learn than the *torus* and might require more training samples. Therefore, when little training data is available, the bigger impact of correlation is apparently also due to the decreasing absolute number of independent samples available to the model to learn the concept.

Taking a closer look at one of the models, we see that only reporting the test accuracy is not enough to evaluate concept models. Out of the models from the previous experiment, we choose the one trained on 50% of the training data with a correlation of 0.89 between *torus* and *icosphere*. For this dataset the impact of correlation seems big enough to be detected, but small enough to test the metrics. Figure 6 shows different evaluation metrics for a model trained on this dataset. We know from the test accuracy on the uncorrelated dataset (see Figure 5) that with these settings the model struggles to correctly classify the *icosphere*. This uncorrelated test dataset, however, is usually not available for real-world datasets. We see that the usually reported accuracy on a standard test set with the same correlations as the training data does not reveal the insufficiency in the concept model. Yet, the metrics presented in Section 4, namely concept removal accuracy, pointing game accuracy, and the difference in accuracy between dependent and independent samples, do indicate that the *icosphere* has not been learned properly.

7.2. Evaluating All Concept Pairs

The example of *torus* and *icosphere* shows that with a high correlation the model might focus on learning only one of the concepts. This might raise the question, if that is true for other concept pairs as well and if it is always the same concept that the model focuses on. With the aim to answer this, we create datasets for each possible combination of concept pairs with a correlation of 1 and train concept models on these. In Table 1 we report the concept accuracy on an uncorrelated test dataset for each of the combinations. We see that, besides only one concept dropping in accuracy, there are cases where both correlated concepts have a lower

accuracy (see *cube* & *sphere*).

We also see a difference between concepts in how prone they are to drops in accuracy. The concepts *cone* and *torus* seem to be quite robust, in that they retain a high accuracy even with high correlations. This might be due to the fact that these shapes have very unique features and are therefore easier to learn.

Interestingly, we also see effects on other concepts which are not part of the correlation pair but seem to be related to one of the correlated concepts. These side effects are strong for *cube* and *cylinder*, where *cylinder* also drops in accuracy when *cube* has not been learned properly and vice versa. Similar but less severe effects can be observed for *sphere* and *icosphere*. The reason for that might be a visual similarity, which makes the model learn shared features for both concepts in the uncorrelated case, but when no features are learned for one of them due to a correlation, these are missing for the other concept as well.

8. Mitigating the Effects of Concept Correlation

Having analyzed the effects of concept correlation, we now try to mitigate these and report initial results on the 3D shapes dataset. We propose two simple methods, of which one tries to contain the side effects on concepts which are not part of the correlation and the other one aims to make the model learn both concepts despite high correlations.

8.1. Training One Model for Each Concept

Since we assume that the shared feature extractor causes the side effects on seemingly unrelated concepts, we conduct the same experiments as in Section 7.2 with the sole difference of training one model for each concept as a binary classification task instead of training a single model for predicting all concepts. The single concept models have the same architecture as the shared concept model. We see in Figure 7 that we get similar results for the correlated concepts (apart from differences in which of the two concepts has the higher accuracy) but less severe drops in accuracy for uncorrelated concepts (see *cube* and *cylinder*), which supports our assumption.

8.2. Weighting the Loss of Independent Samples

In order to tackle the issue of a model focusing on only one concept of a concept pair with high correlation, we try to balance the focus by putting more weight on the loss of samples where only one of the concepts is present. We want to isolate the effects of loss weighting in a controlled setting by using 3D shapes datasets with only one correlated concept pair. We train single concept models for each concept and loss weight. Figure 8 shows the results for two concepts on datasets with different degrees of correlation with another concept. By weighting the loss for independent samples we increase the accuracy for *icosphere* from 93.4% to 97.0% and for *cube* from 91.9% to 95.2% (both: weight = 12) for correlation values of 0.88 and 0.89, respectively. On a dataset with an even higher correlation of 0.94, loss weighting improves the accuracy for *icosphere* from 71.9% to 80.6% (weight = 14) and for *cube* from 75.6% to 83.2% (weight = 18). Although we do not achieve the same accuracies as models trained on a dataset without correlations, we do see considerable improvement by weighting the loss for independent samples.

	Cube	Sphere	Cylinder	Cone	Torus	Icosphere
Cube & Sphere	-5.9	+6.8	+1.9	-0.0	0.0	+1.0
Cube & Cylinder	-0.8	+0.3	+1.8	-0.1	-0.1	+0.6
Cube & Cone	-0.1	+0.2	+18.3	0.0	-0.0	+0.1
Cube & Torus	+0.1	+0.2	+21.1	-0.0	-0.1	+0.2
Cube & Icosphere	+3.6	+0.3	+1.0	0.0	-0.0	-4.4
Sphere & Cylinder	+6.8	-8.9	+8.4	+0.1	-0.1	+1.1
Sphere & Cone	-0.4	-5.0	-0.3	+7.0	+0.0	+3.9
Sphere & Torus	+0.0	+0.0	-0.6	0.0	+0.4	+14.7
Sphere & Icosphere	+0.5	+0.1	-0.3	+0.1	-0.0	-0.6
Cylinder & Cone	+19.3	+0.1	+3.4	-1.5	+0.1	-0.2
Cylinder & Torus	+15.7	+0.1	-0.3	0.0	+0.3	+0.1
Cylinder & Icosphere	+3.8	+1.4	+14.3	-0.0	-0.1	-15.7
Cone & Torus	-0.3	+0.1	-0.1	-11.2	+14.6	+0.1
Cone & Icosphere	-0.0	+4.4	-0.5	+10.4	0.0	-8.0
Torus & Icosphere	+0.1	+17.2	-0.3	-0.1	+0.3	-0.4

Figure 7: Difference in concept accuracy on an uncorrelated test dataset between training CBMs and training a single model for each concept. Each row denotes a dataset with a correlation of 1 between the two concepts. A positive value denotes a higher accuracy for individual models compared to CBMs, and vice versa.

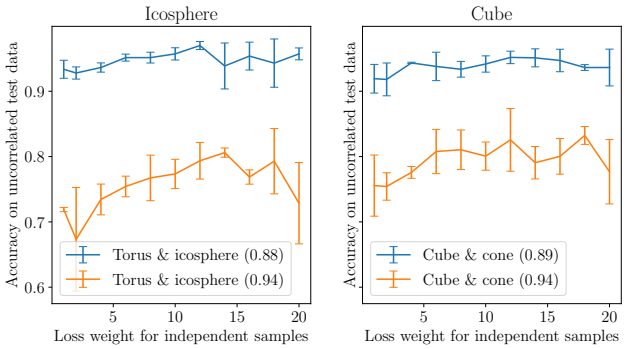


Figure 8: Concept accuracy on an uncorrelated test dataset over different loss weights for independent samples. Models are trained for each concept individually. Each training dataset has a correlated concept pair with the correlation coefficient shown in brackets.

9. Discussion & Conclusion

In this work, we study concept correlations and their effects on concept-based models. We find that test accuracy alone does not provide an adequate evaluation for concept models. We need additional methods for judging whether a concept has been learned correctly by the model. Our results suggest that the methods we propose are able to detect when a concept is predicted based on other concepts or features. Although these methods have limitations and requirements (e.g. concept masks for the pointing game accuracy), each of them can at least give an indication on whether there is an issue with learning this concept and when used together, these methods can provide a more meaningful evaluation.

Furthermore, we evaluate concept models trained on CUB and GTSRB. We see that they struggle to learn some concepts with high correlation. The difference between the balanced test accuracy on dependent and independent samples is higher for correlations close to 1 and -1. Additionally, we present examples where removing the concept from an image does not affect the model's output for that concept.

Using datasets of 3D shapes, for which we can control the image generation process, we perform a deeper analysis on the impact of concept correlation. We find that for at least one concept of a pair the accuracy on uncorrelated test data decreases with higher correlation and with less training data. With a correlation of 1, the concept model either learns only one of the concepts or both of them but only to an extent. Furthermore, we see side effects of concept correlations on other seemingly unrelated concepts.

We show that by training one model per concept instead of using a shared model we are able to contain these side effects on other concepts. Additionally, we achieve substantial improvement of the concept model's performance on concepts which suffer from a high correlation with another concept by putting more weight on the loss of samples where only one of the concepts is present. We acknowledge that training one model per concept is not particularly efficient and therefore limits its application in fields which require real-time capability. However, in safety-critical applications like medical image analysis, an increase in computational time is accepted if it leads to safer predictions and better interpretability.

Since the issues presented in this paper would prevent concept models from being applied in domains where they would be most useful (e.g. medicine, autonomous driving), these issues have to be addressed. To this end, we suggest the following directions for future work: Although pairwise correlation probably covers the main effects, we should additionally analyze more complex relations which can be used by models to predict a concept based on this relation and not on its presence in an image. Furthermore, the mitigation techniques presented in this paper work well for 3D shapes datasets with a single concept pair with high correlation. Real-world datasets usually have more than one highly correlated concept pair. More complex weighting or upsampling methods, generating synthetic data with independent concepts, or guiding a model to learn a concept by providing localization information for a few samples could be future directions for mitigating the effects of concept correlation on models trained on real-world datasets.

Acknowledgments

This work was funded by the Bavarian Ministry for Economic Affairs, Regional Development and Energy as part of a project to support the thematic development of the Institute for Cognitive Systems.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[2] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[3] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[4] Zhengqing Fang, Kun Kuang, Yuxiao Lin, Fei Wu, and Yu-Feng Yao. Concept-based Explanation for Fine-grained Images and Its Application in Infectious Keratitis Classification. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 700–708, 2020.

[5] Ruth Fong and Andrea Vedaldi. Net2Vec: Quantifying and Explaining How Concepts Are Encoded by Filters in Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8730–8738, 2018.

[6] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards Automatic Concept-based Explanations. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.

[9] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings*

of the 35th International Conference on Machine Learning, pages 2668–2677, 2018.

[10] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept Bottleneck Models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5338–5348, 2020.

[11] Jan Kronenberger and Anselm Haselhoff. Dependency Decomposition and a Reject Option for Explainable Models. *arXiv:2012.06523*, 2020.

[12] Chi Li, M. Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D. Hager, and Manmohan Chandraker. Deep Supervision with Intermediate Concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1828–1843, 2019.

[13] Max Losch, Mario Fritz, and Bernt Schiele. Interpretability Beyond Classification Output: Semantic Bottleneck Networks. *arXiv:1907.10882*, 2019.

[14] Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and Pitfalls of Black-Box Concept Learning Models. *arXiv:2106.13314*, 2021.

[15] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do Concept Bottleneck Models Learn as Intended? *arXiv:2105.04289*, 2021.

[16] Meike Nauta, Ron van Bree, and Christin Seifert. Neural Prototype Trees for Interpretable Fine-Grained Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021.

[17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[18] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pages 1453–1460, 2011.

[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[20] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Computation & Neural Systems Technical Report 2010-001, California Institute of Technology, Pasadena, CA, 2011.

[21] Sandareka Wickramanayake, Wynne Hsu, and Mong Li Lee. Comprehensible Convolutional Neural Networks via Guided Concept Learning. *arXiv:2101.03919*, 2021.

[22] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-Down Neural Attention by Excitation Backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[23] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting Deep Visual Representations via Network Dissection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2131–2145, 2019.

[24] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable Basis Decomposition for Visual Explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.