AnyCBMs: How to Turn Any Black Box into a Concept Bottleneck Model[*]

Gabriele Dominici[1,*,†], Pietro Barbiero[1,†], Francesco Giannini[2], Martin Gjoreski[1] and Marc Langeinrich[1]

[1]*Università della Svizzera Italiana, Lugano, Switzerland*

[2]*Università di Siena, Siena, Italy*

Abstract

Interpretable deep learning aims at developing neural architectures whose decision-making processes could be understood by their users. Among these techniqes, Concept Bottleneck Models enhance the interpretability of neural networks by integrating a layer of human-understandable concepts. These models, however, necessitate training a new model from the beginning, consuming significant resources and failing to utilize already trained large models. To address this issue, we introduce "AnyCBM", a method that transforms any existing trained model into a Concept Bottleneck Model with minimal impact on computational resources. We provide both theoretical and experimental insights showing the effectiveness of AnyCBMs in terms of classification performances and effectivenss of concept-based interventions on downstream tasks.

Keywords

Interpretability, Explainable AI, Concept Learning, Concept Bottleneck Models

1. Introduction

Numerous national and international regulatory frameworks underscore the transformative potential of artificial intelligence (AI). However, they also warn of the inherent risks associated with such powerful technology, emphasizing the importance of careful monitoring and strict protections. For instance, the recent AI Act [1] aims to implement detailed regulations for AI systems, ensuring their safety, transparency, and accountability. Similarly, in the US, the federal government issued an executive order that proposes principles for trustworthy AI. Hence, interpretable AI has become a crucial aspect of modern machine learning to address concerns over the opaque nature of deep learning (DL) models [2, 3]. The quest for transparency has been driven by the need to understand the decision-making processes of AI systems, particularly in critical areas where ethical [4] and legal [5] implications of these systems' decisions are significant.

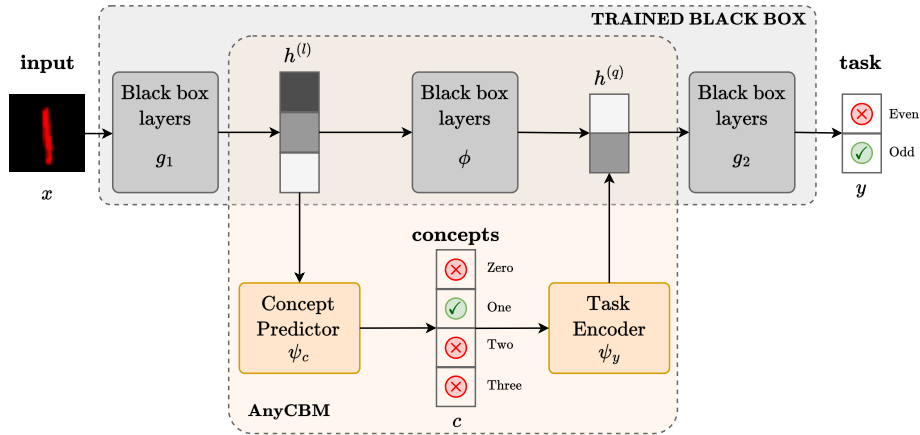**Figure 1:** Any Concept Bottleneck Models (AnyCBMs) transform any black box neural architecture into an interpretable CBM mapping black box embeddings into a set of supervised concepts and then mapping the predicted concepts back to black box embeddings. This allows AnyCBMs to be applied to any layer of a trained black box and to perform concept-based interventions as in standard CBMs.

Concept Bottleneck Models (CBMs) [6] are a family of differentiable models aiming to increase DL interpretability [7]. These models map input data (e.g., pixel intensities) to human-understandable concepts (e.g., shapes, colors), and then use these concepts to predict labels of a downstream classification task. However, existing CBMs necessitate training a new model from the beginning even in settings where trained or fine-tuned models already exist. In these scenarios, current CBM architectures would consume significant resources in re-training or fine-tuning again possibly large models. As a result, this limitation restricts CBMs' ability to be adopted in new domains. To bridge this gap, we introduce Any Concept Bottleneck Models (AnyCBMs, Figure 1), a method to transform any black-box neural architecture into an interpretable CBM. The key innovation of AnyCBMs lies in a neural model mapping black-box embeddings into a set of supervised concepts and then mapping the predicted concepts back to black-box embeddings. This allows AnyCBMs to be applied to any layer of a trained black box and to perform concept-based interventions as in standard CBMs. Results demonstrate that AnyCBMs match black-box performance in classification accuracy in downstream tasks and CBM performance in concept accuracy. In addition, AnyCBM could steer the behaviour of a black-box model acting on human-understandable concepts as effectively as in CBMs.

2. Background

Concept-based models $f : C \rightarrow Y$ learn a map from a concept space C to a task space Y [8]. If concepts are semantically meaningful, then humans can interpret this mapping by tracing back predictions to the most relevant concepts [7]. When the features of the input space are hard for humans to reason about (such as pixel intensities), concept-based models work on the output of a concept-encoder mapping $g : X \rightarrow C$ from the input space X to the concept space C [9]. These architectures are known as Concept Bottleneck Models (CBMs) [6]. In general,

training a CBM model may require a dataset where each sample consists of input features $x \in X \subseteq \mathbb{R}^n$ (e.g., an image's pixels), k ground truth concepts $c \in C \subseteq \{0,1\}^k$ (i.e., a binary vector with concept annotations, when available) and o task labels $y \in Y \subseteq \{0,1\}^o$ (e.g., an image's classes). During training, a CBM is encouraged to align its predictions to task labels i.e., $y \approx \hat{y} = f(g(x))$. Similarly, a concept predictor can be supervised when concept labels are available i.e., $c \approx \hat{c} = g(x)$. We indicate concept and task predictions as $\hat{c}_i = (g(x))_i$ and $\hat{y}_j = (f(\hat{c}))_j$ respectively. When concept labels are not available, they can still be extracted in with unsupervised techniques [9, 10, 11], which make CBMs applicable to a wide range of applications.

3. AnyCBM: Turning Black Boxes into Concept Bottleneck Models

AnyCBM (Figure 1) is a method designed to convert any opaque neural network architecture into a Concept Bottleneck Model (CBM) that is interpretable. The fundamental innovation of AnyCBMs involves the use of an external model that processes embeddings from a trained black box model. These embeddings, denoted as $h^{(l)} \in H^{(l)} \subseteq \mathbb{R}^l$, are encoded into a set of supervised concepts $c \in C$. Subsequently, these concepts are mapped back into embeddings $h^{(q)} \in H^{(q)} \subseteq \mathbb{R}^q$. This process allows for the embedding space of the black box model to be translated into a more understandable and interpretable form, where each concept represents a meaningful feature or characteristic that explains the decision-making process of the neural network. The following definition formalizes AnyCBMs.

Definition 3.1 (AnyCBM). Given a black box model $\phi : H^l \to H^q$ and a set of concepts C, a AnyCBM is a tuple of models (ψ_c, ψ_y) such that, the following diagram commutes:

$$
\begin{array}{ccc}
H^{(l)} & \xrightarrow{\quad \phi \quad} & H^{(q)} \\
& \searrow_{\psi_c} \qquad \nearrow_{\psi_y} & \\
& C &
\end{array}
$$
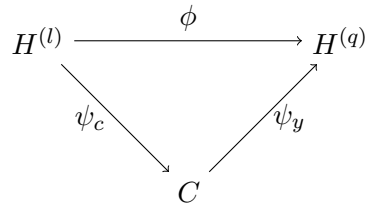
More specifically, the concept predictor $\psi_c : H^{(l)} \to C$ encodes black box embeddings into concepts, and the task encoder $\psi_y : C \to H^{(q)}$ maps concepts back into black box embeddings. In practice, the commutative diagram describes how the interpretable mapping through C via ψ_c and ψ_y should be consistent with the direct transformation of the black box ϕ. Also properties and capabilities of AnyCBMs can directly be derived from the commutative diagram as it constraints the relationships among the transformations ψ_c, ψ_y, and ϕ.

In the following we present two practical case studies.

Case 1: ϕ is the identity function on H   When ϕ is the identity function, $\phi(h^{(l)}) = h^{(l)}$ for all $h^{(l)} \in H^{(l)}$, and $H^{(l)} = H^{(q)}$. The diagram simplifies, and we have:

$$\psi_y \circ \psi_c = \text{id}_H$$

Theorem 3.2. *If ϕ is the identity function on H, then ψ_y is injective:*

$$\phi = id_H \implies \psi_y : C \hookrightarrow H^{(q)} \tag{1}$$

Proof. Assume $\psi_y(c_1) = \psi_y(c_2)$. Since ψ_c is surjective, there exist $h_1, h_2 \in H^{(l)}$ such that $\psi_c(h_1) = c_1$ and $\psi_c(h_2) = c_2$. Then,

$$h_1 = \psi_y(\psi_c(h_1)) = \psi_y(c_1) = \psi_y(c_2) = \psi_y(\psi_c(h_2)) = h_2.$$

Thus, $c_1 = c_2$, proving that ψ_y is injective. □

Significance: This property implies that ψ_y can uniquely reconstruct elements of $H^{(l)}$ from C, despite ψ_c not being injective. For example, if ψ_c represents a lossy compression, then ψ_y could be an error-correcting decoding where no information is lost despite compression.

Case 2: independent training   In many practical cases, concept predictors and task encoders are independently trained to reduce concept leakage [12]. In this common setting, we can prove another property of AnyCBMs task encoders.

Theorem 3.3. *If ψ_c and ψ_y are independently trained and ϕ is a multi-layer neural network, then ψ_y cannot be surjective.*

Proof. Assume for contradiction that ψ_y is surjective. The surjectivity of ψ_y would require that every point in $H^{(q)}$ is the image of some point in C. Given the independent training, the domain of ψ_y is finite, specifically 2^k. Since $H^{(q)} \subseteq \mathbb{R}^q$, the mapping $\psi_y : C \to H^{(q)}$ must pull from a set with finite cardinality 2^k to cover $\mathbb{R}^q$ which is a contradiction. Hence, ψ_y cannot be surjective. □

Significance: This theorem indicates that the surjectivity of ψ_y depends on the way we train the concept bottleneck. This means that, under independent training, AnyCBMs are not invertible, even when ϕ represents an invertible transformation.

4. Experiments

Our experiments aim to answer the following questions:

- How is AnyCBMs classification performance on concepts and downstream tasks compared to standard CBMs and black boxes?
- How effective are concept interventions in AnyCBM compared to concept interventions in CBM?
- Is it possible to train AnyCBM with a dataset slightly different from the one used to train the black-box model?

This section describes essential information about the experiments.

4.1. Data & task setup

In our experiments, we use two different datasets commonly used to evaluate CBMs: MNIST even/odd [13], where the task is to predict whether handwritten digits are even or odd; and CUB [14], where the task is to predict bird species based on bird characteristics.

4.2. Evaluation

In our analysis, we use ROC-AUC scores to measure classification performance in concepts and downstream tasks and to measure the effectiveness of concept-based interventions in improving classification performance in downstream tasks. To measure the effectiveness of interventions, we follow a similar approach to the one described by Espinosa Zarlenga et al. [15]. First, we perturb the latent embeddings by adding a small random noise a few layers before predicting concepts both in AnyCBM and CBM. Then, we intervene on a portion of the concepts with the ground truth. Finally, we test our assumption about the possibility of training AnyCBM with a different dataset with concepts. We train the black-box model with an MNIST even/odd dataset with RGB images. Then, we train AnyCBM with a version of MNIST that contains greyscale images with associated concepts. All results are reported using the mean and standard error over five different runs with different parameter initializations.

4.3. Baselines

In our experiments, we compare AnyCBMs with standard CBMs and with an end-to-end black-box model in terms of generalisation performance. We compare AnyCBMs' interventions with the effectiveness of interventions in standard CBMs.

5. Key findings

AnyCBMs match black box and CBM performances in terms of classification accuracy on concepts and downstream tasks (Table 1), AnyCBMs perform just as well as the original black-box models on which they are based when it comes to accurately completing tasks. Additionally, the accuracy with which these models handle concepts is equal to that of other similar Concept Bottleneck Model architectures. This suggests that AnyCBMs could be a valuable tool for making existing black-box models easier to understand. Using AnyCBMs, we might be able to explain how these complex models work and, in particular, which encoded information is inside the layers of the models, making them more transparent and accessible for further analysis and improvement.

AnyCBM interventions are as effective as in Concept Bottleneck Models (Figure 2) AnyCBMs are as responsive to concept-based interventions as standard CBMs. This means that when concepts predicted by AnyCBMs are manually changed by human experts at test time, they effectively impact the downstream task accuracy. This finding underlines the ability of AnyCBMs to interact with domain experts as it would have been expected by CBMs. In

	MNIST EVEN/ODD		CUB	
	Task ROC AUC	Concept ROC AUC	Task ROC AUC	Concept ROC AUC
Black box	99.8 ± 0.0	-	90.5 ± 0.3	-
CBM	99.8 ± 0.0	99.8 ± 0.0	90.0 ± 0.2	83.0 ± 0.2
Black box + AnyCBMs	99.6 ± 0.0	98.8 ± 0.3	90.3 ± 0.2	84.8 ± 0.3

Table 1

Downstream task and concept ROC AUC of AnyCBMs compared to CBMs and a black box model on MNIST and CUB datasets.
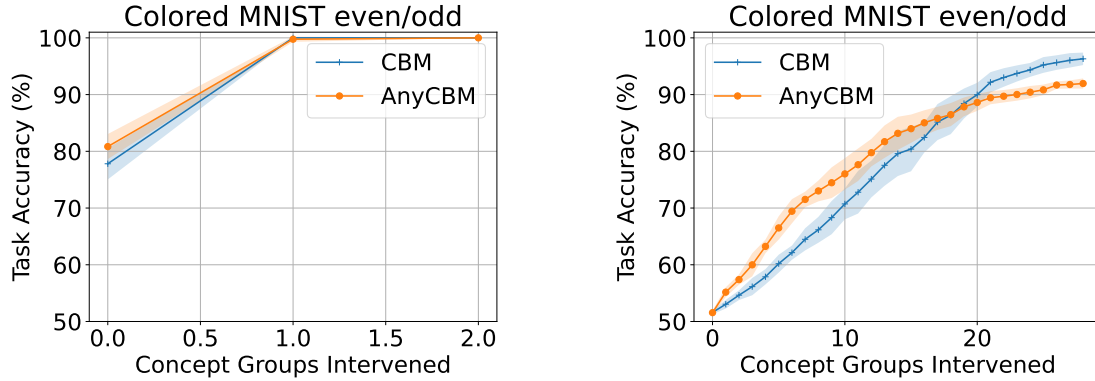

Figure 2: Task accuracy of AnyCBMs compared to CBMs after intervening on an increasing number of family of concepts on the MNIST and CUB dataset.

	MNIST EVEN/ODD RGB		MNIST EVEN/ODD GREY	
	Task	Concept	Task	Concept
Black box	98.6 ± 0.1	-	89.7 ± 1.7	-
CBM	74.4 ± 3.1	88.7 ± 0.7	99.3 ± 0.0	98.6 ± 0.0
Black box + AnyCBMs	98.6 ± 0.0	90.9 ± 1.2	89.8 ± 1.6	94.1 ± 0.2

Table 2

Downstream task and concept accuracy of AnyCBMs (trained on MNIST Greyscale) compared to CBMs (trained on MNIST Greyscale) and a black box model (trained on MNIST RGB)

addition, this represents a successful method to steer the behaviour of the model modifying human-understandable concepts.

AnyCBM can be trained with a different dataset from the one used to train the black-box model (Table 2) One can initially train a black box model with a dataset, which could be larger or more beneficial for addressing the downstream task. Subsequently, the AnyCBM module can be trained on a slightly different dataset that includes concept annotations. As demonstrated in Table 2, this approach does not compromise the model's performance in terms of task accuracy when both the black-box model and AnyCBM are utilised in the original dataset. It also partially accurately predicts concepts in the original dataset, even when there is a distribution shift. This indicates that AnyCBM can alleviate a significant constraint of CBMs,

which is the requirement for concept annotations in the dataset used to train the entire model. In addition, the dataset used to train the AnyCBM module could contain only input and concept annotations, without the need for label annotations.

6. Discussion

Advantages In the age of Large Models with billions of parameters, the development of solutions that do not require retraining to enhance their capabilities is crucial. AnyCBMs successfully meet this need, as they do not require the alteration of the weights of a pre-trained black-box model. This enables any black-box model to acquire the extra features of CBMs, such as the interpretability of the latent space and the capacity to change the model's behaviour through concept interventions. Furthermore, we believe that AnyCBM can be trained using a dataset that is smaller than the one used to train the original black-box because it has a consistently smaller number of parameters. Interestingly, the dataset can even be distinct (for instance, we might train the model with a dataset without concepts while training AnyCBM with a slightly different dataset that has only concept annotations), mitigating the CBMs' constraint of needing concept annotations for the training set used to train the model. Under these circumstances, it might be intriguing to determine whether certain concepts can be accurately predicted from the latent embeddings of black-box models. If some concepts are unpredictable, this could suggest that the black-box models did not grasp that particular concept in the prior training, either due to the dataset employed or its irrelevant role in task prediction.

Limitations Although the model gains the benefits of CBMs, it also takes on some of their drawbacks. The primary constraint is the necessity for concept data to train the AnyCBM component, although this is somewhat alleviated by the reduced need for concept annotations and the option to utilise an alternate dataset for their extraction.

Future work We underscore the importance of delving deeper into AnyCBM and its benefits, while also trying to mitigate its drawbacks. For example, it would be intriguing to examine its application in multimodal contexts, where automatic concept extraction could be feasible, as suggested in [11].

7. Conclusion

This paper introduces Any Concept Bottleneck Models (AnyCBMs), a method for transforming opaque neural networks into interpretable Concept Bottleneck Models (CBMs), allowing for insights into the decision-making process in terms of concept-based explanations and interventions. This paper analyses practical case studies that demonstrate the properties and limitations of AnyCBMs in enhancing interpretability while maintaining high classification performances from both a theoretical and an experimental perspective. These results suggest how AnyCBMs could represent a computationally effective solution to enhance the interpretability of existing trained or fine-tuned black-box neural networks, allowing also for concept-based interventions in the black-box latent space.

Acknowledgments

References

[1] T. Madiega, Artificial intelligence act, European Parliament: European Parliamentary Research Service (2021).

[2] A. Bussone, S. Stumpf, D. O'Sullivan, The role of explanations on trust and reliance in clinical decision support systems, in: 2015 international conference on healthcare informatics, IEEE, 2015, pp. 160–169.

[3] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215.

[4] J. M. Durán, K. R. Jongsma, Who is afraid of black box algorithms? On the epistemological and ethical basis of trust in medical AI, Journal of Medical Ethics 47 (2021) 329–335.

[5] S. Lo Piano, Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward, Humanities and Social Sciences Communications 7 (2020) 1–7.

[6] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, P. Liang, Concept bottleneck models, in: International Conference on Machine Learning, PMLR, 2020, pp. 5338–5348.

[7] A. Ghorbani, A. Abid, J. Zou, Interpretation of neural networks is fragile, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 3681–3688.

[8] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, P. Ravikumar, On completeness-aware concept-based explanations in deep neural networks, Advances in Neural Information Processing Systems 33 (2020) 20554–20565.

[9] A. Ghorbani, J. Wexler, J. Zou, B. Kim, Towards automatic concept-based explanations, arXiv preprint arXiv:1902.03129 (2019).

[10] L. C. Magister, D. Kazhdan, V. Singh, P. Liò, Gcexplainer: Human-in-the-loop concept-based explanations for graph neural networks, arXiv preprint arXiv:2107.11889 (2021).

[11] T. Oikarinen, S. Das, L. M. Nguyen, T.-W. Weng, Label-free concept bottleneck models, 2023. `arXiv:2304.06129`.

[12] A. Mahinpei, J. Clark, I. Lage, F. Doshi-Velez, W. Pan, Promises and pitfalls of black-box concept learning models, arXiv preprint arXiv:2106.13314 (2021).

[13] P. Barbiero, G. Ciravegna, F. Giannini, P. Lió, M. Gori, S. Melacci, Entropy-based logic explanations of neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, 2022, pp. 6046–6054.

[14] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200-2011 dataset, 2011.

[15] M. Espinosa Zarlenga, P. Barbiero, G. Ciravegna, G. Marra, F. Giannini, M. Diligenti, Z. Shams, F. Precioso, S. Melacci, A. Weller, et al., Concept embedding models, Advances in Neural Information Processing Systems 35 (2022).