Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable?

Ričards Marcinkevičs[*,†] **Sonia Laguna**[*,†] **Moritz Vandenhirtz**[†] **Julia E. Vogt**[†]

Abstract

Recently, interpretable machine learning has re-explored concept bottleneck models (CBM), comprising step-by-step prediction of the high-level concepts from the raw features and the target variable from the predicted concepts. A compelling advantage of this model class is the user's ability to intervene on the predicted concept values, consequently affecting the model's downstream output. In this work, we introduce a method to perform such concept-based interventions on already-trained neural networks, which are not interpretable by design. Furthermore, we formalise the model's *intervenability* as a measure of the effectiveness of concept-based interventions and leverage this definition to fine-tune black-box models. Empirically, we explore the intervenability of black-box classifiers on synthetic tabular and natural image benchmarks. We demonstrate that fine-tuning improves intervention effectiveness and often yields better-calibrated predictions. To showcase the practical utility of the proposed techniques, we apply them to chest X-ray classifiers and show that fine-tuned black boxes can be as intervenable and more performant than CBMs.

1 Introduction

Interpretable and explainable machine learning [8, 28] have seen a renewed interest in concept-based predictive models and approaches to *post hoc* explanation, such as concept bottlenecks [21, 20, 19], contextual semantic interpretable bottlenecks [26], concept whitening layers [5], and concept activation vectors [16]. Moving beyond interpretations defined in the high-dimensional and unwieldy input space, these techniques relate the model's inputs and outputs via additional high-level human-understandable attributes, also referred to as concepts. Typically, neural network models are supervised to predict these attributes in a dedicated bottleneck layer, or *post hoc* explanations are derived to measure the model's sensitivity to a set of concept variables.

This work focuses specifically on the concept bottleneck models [19]. In brief, a CBM $f_{\boldsymbol{\theta}}$, parameterised by $\boldsymbol{\theta}$, is given by $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = g_{\boldsymbol{\psi}}(h_{\boldsymbol{\phi}}(\boldsymbol{x}))$, where $\boldsymbol{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ are covariates and targets, respectively, $h_{\boldsymbol{\phi}} : \mathcal{X} \rightarrow \mathcal{C}$ maps inputs to predicted concepts, *i.e.* $\hat{\boldsymbol{c}} = h_{\boldsymbol{\phi}}(\boldsymbol{x})$, and $g_{\boldsymbol{\psi}} : \mathcal{C} \rightarrow \mathcal{Y}$ predicts the target based on $\hat{\boldsymbol{c}}$, *i.e.* $\hat{y} = g_{\boldsymbol{\psi}}(\hat{\boldsymbol{c}})$. CBMs are trained on labelled data points $(\boldsymbol{x}, \boldsymbol{c}, y)$ annotated by concepts $\boldsymbol{c} \in \mathcal{C}$ and are supervised by the concept and target prediction losses. Note that above, the output of $h_{\boldsymbol{\phi}}$ forms a *concept bottleneck layer*, and thus, the model's final output depends on the covariates $\boldsymbol{x}$ solely through the predicted concept values $\hat{\boldsymbol{c}}$. At inference time, a human user may interact with the CBM by editing the predicted concept values and, as a result, affecting the downstream target prediction. For example, if the user chooses to replace $\hat{\boldsymbol{c}}$ with another $\boldsymbol{c}' \in \mathcal{C}$, the final prediction is given by $\hat{y}' = g_{\boldsymbol{\psi}}(\boldsymbol{c}')$. This act of model editing is known as an *intervention*. User's ability to intervene is a compelling advantage of CBMs over other interpretable models, in that the former allows for human-model interaction.

[*]Equal contribution. Correspondence to {ricardsm,slaguna}@inf.ethz.ch
[†]Department of Computer Science, ETH Zurich, Switzerland

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

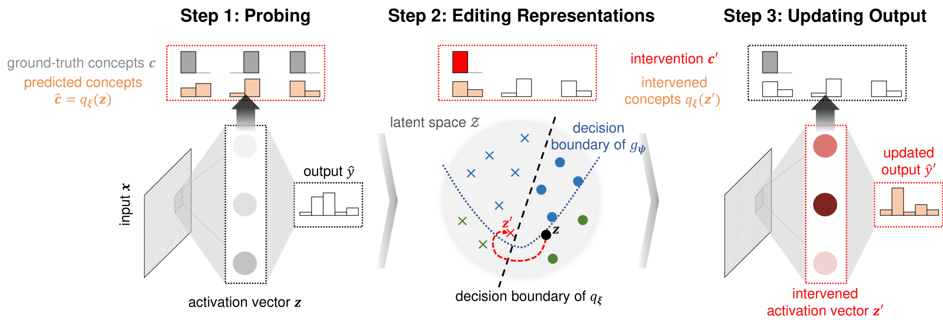


Figure 1: Schematic summary of the concept-based instance-specific intervention procedure for a black-box neural network. (i) A probe $q_{\boldsymbol{\xi}}$ is trained to predict the concepts $\boldsymbol{c}$ from the activation vector $\boldsymbol{z}$. (ii) The representations are edited according to Equation 1. (iii) The final prediction is updated to $\hat{y}'$ based on the edited representations $\boldsymbol{z}'$.

An apparent limitation of the CBMs is that the knowledge of the concepts and annotated data are required at model development. Recent research efforts have been directed at mitigating these limitations by converting pretrained models into CBMs *post hoc* [43] and discovering concept sets automatically in a label-free manner using GPT-3 and CLIP models [31]. However, these works either have not comprehensively investigated the effectiveness of interventions in this setting or have mainly concentrated on global model editing rather than the influence on individual data point predictions. Appendix A includes an overview of other related works. Complementary to the *post hoc* and label-free CBMs, we focus on interventions and explore two related research questions: (i) *Given post hoc a concept set and dataset with concept labels, how can we perform instance-specific interventions directly on a trained black-box model?* (ii) *How can we fine-tune the black-box model to improve the effectiveness of interventions performed on it?* Herein, instance-specific interventions refer to the interventions performed locally, *i.e.* individually for each data point.

Contributions This work contributes to the line of research on concept bottleneck models and concept-based explanations in several ways. (1) We devise a simple procedure (Figure 1) that, given a set of concepts and a labelled dataset, allows performing concept-based instance-specific interventions on an already trained black-box neural network by editing its activations at an intermediate layer. Notably, during training, concept labels are not required and the network's architecture does not need to be adjusted. (2) We formalise *intervenability* as a measure of the effectiveness of the interventions performed on the model and introduce a novel fine-tuning procedure for black-box neural networks that utilises intervenability as the loss. This fine-tuning strategy is designed to improve the effectiveness of concept-based interventions while preserving the original model's architecture and learnt representations. (3) We evaluate the proposed procedures alongside several common-sense baseline techniques on the synthetic tabular, natural image, and medical imaging data. We demonstrate that in practice, for some classification problems, we can improve the predictive performance of already trained black-box models via concept-based interventions. Moreover, the effectiveness of interventions improves considerably when explicitly fine-tuning for intervenability.

2 Methods

In this section, we define a measure for the effectiveness of interventions and present techniques for performing concept-based interventions on black-box neural networks and fine-tuning black boxes to improve the effectiveness of such interventions. Some further remarks and discussion beyond the current scope are provided in Appendix C. In the remainder of this paper, we will adhere to the following notation. Let $\boldsymbol{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, and $\boldsymbol{c} \in \mathcal{C}$ be the covariates, targets, and concepts. Consider a black-box neural network $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$ parameterised by $\boldsymbol{\theta}$ and a slice $\langle g_{\boldsymbol{\psi}}, h_{\boldsymbol{\phi}} \rangle$ [22], defining a layer, s.t. $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = g_{\boldsymbol{\psi}}(h_{\boldsymbol{\phi}}(\boldsymbol{x}))$. We will assume that the black box has been trained end-to-end on the labelled data $\{(\boldsymbol{x}_i, y_i)\}_i$. When applicable, we will use a similar notation for CBMs, as outlined in Section 1. Lastly, for the techniques introduced below, we will assume being given a labelled and annotated validation set $\{(\boldsymbol{x}_i, \boldsymbol{c}_i, y_i)\}_i$.

2.1 Intervening on Black-box Models

Given a black-box model $f_{\boldsymbol{\theta}}$ and a data point $(\boldsymbol{x}, y)$, a human user might desire to influence the prediction $\hat{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ made by the model via high-level and understandable concept values $\boldsymbol{c}'$, *e.g.* think of a doctor trying to interact with a chest X-ray classifier ($f_{\boldsymbol{\theta}}$) by annotating their findings ($\boldsymbol{c}'$) in a radiograph ($\boldsymbol{x}$). To facilitate such interactions, we propose a simple recipe for concept-based instance-specific interventions (detailed in Figure 1) that can be applied to any black-box neural network model. Intuitively, using the given validation data and concept values, our procedure edits the network's representations $\boldsymbol{z} = h_{\boldsymbol{\phi}}(\boldsymbol{x})$, where $\boldsymbol{z} \in \mathcal{Z}$, to align more closely with $\boldsymbol{c}'$ and, thus, affects the downstream prediction. Below, we explain this procedure step-by-step. Pseudocode implementation can be found as part of Algorithm B.1 in Appendix B.

Step 1: Probing To align the network's activation vectors with concepts, the preliminary step is to train a probing function [1, 2], or a probe for short, mapping the intermediate representations to concepts. Namely, using the given annotated validation data $\{(\boldsymbol{x}_i, \boldsymbol{c}_i, y_i)\}_i$, we train a probe $q_{\boldsymbol{\xi}}$ to predict the concepts $\boldsymbol{c}_i$ from the representations $\boldsymbol{z}_i = h_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$: $\min_{\boldsymbol{\xi}} \sum_i \mathcal{L}^{\boldsymbol{c}}(q_{\boldsymbol{\xi}}(\boldsymbol{z}_i), \boldsymbol{c}_i)$, where $\mathcal{L}^{\boldsymbol{c}}$ is the concept prediction loss. Note that, herein, an essential design choice explored in our experiments is the (non)linearity of the probe. Consequently, the probing function can be used to interpret the activations in the intermediate layer and edit them.

Step 2: Editing Representations Recall that we are given a data point $(\boldsymbol{x}, y)$ and concept values $\boldsymbol{c}'$ for which an intervention needs to be performed. Note that this $\boldsymbol{c}' \in \mathcal{C}$ could correspond to the ground-truth concept values or reflect the beliefs of the human subject intervening on the model. Intuitively, we seek an activation vector $\boldsymbol{z}'$, which is similar to $\boldsymbol{z} = h_{\boldsymbol{\phi}}(\boldsymbol{x})$ and consistent with $\boldsymbol{c}'$ according to the previously learnt probing function $q_{\boldsymbol{\xi}}$: $\arg\min_{\boldsymbol{z}'} \; d(\boldsymbol{z}, \boldsymbol{z}'), \text{ s.t. } q_{\boldsymbol{\xi}}(\boldsymbol{z}') = \boldsymbol{c}'$, where d is an appropriate distance function applied to the activation vectors from the intermediate layer. Throughout our experiments, we utilise the Euclidean metric, which is frequently applied to neural network representations, *e.g.* see [29] and [14]. Instead of the constrained problem above, we resort to minimising a relaxed objective:

$$\arg\min_{\boldsymbol{z}'} \; \lambda \mathcal{L}^{\boldsymbol{c}}(q_{\boldsymbol{\xi}}(\boldsymbol{z}'), \boldsymbol{c}') + d(\boldsymbol{z}, \boldsymbol{z}'), \tag{1}$$

where, similarly to the counterfactual explanation [40, 30], hyperparameter $\lambda > 0$ controls the tradeoff between the intervention's validity, *i.e.* the "consistency" of $\boldsymbol{z}'$ with the given concept values $\boldsymbol{c}'$ according to the probe, and proximity to the original activation vector $\boldsymbol{z}$. In practice, we optimise $\boldsymbol{z}'$ for batched interventions using Adam [18].

Step 3: Updating Output The edited representation $\boldsymbol{z}'$ can be consequently fed into $g_{\boldsymbol{\psi}}$ to compute the updated output $\hat{y}' = g_{\boldsymbol{\psi}}(\boldsymbol{z}')$, which could be then returned and displayed to the human subject. For example, if $\boldsymbol{c}'$ are the ground-truth concept values, we would ideally expect a decrease in the prediction error for the given data point $(\boldsymbol{x}, y)$.

2.2 What is Intervenability?

Concept bottlenecks [19] and their extensions are often evaluated empirically by plotting test-set performance or error attained after intervening on concept subsets of varying sizes. Ideally, the model's test-set performance should improve when given more ground-truth attribute values. Below, we formalise this notion of intervention effectiveness, referred to as *intervenability*, for the concept bottleneck and black-box models.

Following the notation from Section 1, for a trained CBM $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = g_{\boldsymbol{\psi}}(h_{\boldsymbol{\phi}}(\boldsymbol{x})) = g_{\boldsymbol{\psi}}(\hat{\boldsymbol{c}})$, we define the intervenability as follows:

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{c},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{c}'\sim\pi}\left[\mathcal{L}^{y}\Big(\underbrace{f_{\boldsymbol{\theta}}(\boldsymbol{x})}_{\hat{y}=g_{\boldsymbol{\psi}}(\hat{\boldsymbol{c}})}, y\Big) - \mathcal{L}^{y}\Big(\underbrace{g_{\boldsymbol{\psi}}(\boldsymbol{c}')}_{\hat{y}'}, y\Big)\right]\right], \tag{2}$$

where $\mathcal{D}$ is the joint distribution over the covariates, concepts, and targets, $\mathcal{L}^{y}$ is the target prediction loss, *e.g.* the mean squared error (MSE) or cross-entropy (CE), and π denotes a distribution over edited concept values $\boldsymbol{c}'$. Observe that Equation 2 generalises the standard evaluation strategy of intervening on a random concept subset and setting it to the ground-truth values, as proposed in the original work on CBMs [19]. Here, the effectiveness of interventions is quantified by the gap between the regular prediction loss and the loss attained after the intervention: the larger the gap between these values, the stronger the effect interventions have.

Note that the definition in Equation 2 can also accommodate more sophisticated intervention strategies, for example, similar to those studied in [37] and [36]. An intervention strategy can be specified via the distribution π, which can be conditioned on $\boldsymbol{x}$, $\hat{\boldsymbol{c}}$, $\boldsymbol{c}$, $\hat{y}$, or even y: $\pi\left(\boldsymbol{c}'|\boldsymbol{x}, \hat{\boldsymbol{c}}, \boldsymbol{c}, \hat{y}, y\right)$. For brevity, we will use π as a shorthand notation for this distribution. Algorithms F.1–F.2 in Appendix F provide concrete examples of intervention strategies utilised in our experiments. Lastly, notice that, in practice, when performing human- or application-grounded evaluation [8], sampling from π may be replaced with the interventions by a human.

Leveraging the intervention procedure described in Section 2.1, analogous to Equation 2, the intervenability for a black-box neural network $f_{\boldsymbol{\theta}}$ at the intermediate layer given by $\langle g_{\boldsymbol{\psi}}, h_{\boldsymbol{\phi}} \rangle$ is

$$\begin{aligned} &\mathbb{E}_{(\boldsymbol{x},\boldsymbol{c},y)\sim\mathcal{D},\, \boldsymbol{c}'\sim\pi}\left[\mathcal{L}^{y}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right), y\right) - \mathcal{L}^{y}\left(g_{\boldsymbol{\psi}}\left(\boldsymbol{z}'\right), y\right)\right], \\ &\text{where } \boldsymbol{z}' \in \arg\min_{\tilde{\boldsymbol{z}}} \lambda \mathcal{L}^{\boldsymbol{c}}\left(q_{\boldsymbol{\xi}}\left(\tilde{\boldsymbol{z}}\right), \boldsymbol{c}'\right) + d\left(\boldsymbol{z}, \tilde{\boldsymbol{z}}\right). \end{aligned} \tag{3}$$

Recall that $q_{\boldsymbol{\xi}}$ is the probe trained to predict $\boldsymbol{c}$ based on the activations $h_{\boldsymbol{\phi}}\left(\boldsymbol{x}\right)$ (step 1, Section 2.1). Furthermore, in the first line of Equation 3, edited representations $\boldsymbol{z}'$ are a function of $\boldsymbol{c}'$, as defined by the second line, which corresponds to step 2 of the intervention procedure (Equation 1).

2.3 Fine-tuning for Intervenability

Since intervenability in Equation 3 is differentiable, a neural network can be fine-tuned by explicitly maximising it using, for example, mini-batch gradient descent. We expect fine-tuning for intervenability to reinforce the model's reliance on the high-level attributes and have a regularising effect. In this section, we provide a detailed description of the fine-tuning procedure (Algorithm B.1, Appendix B), and, afterwards, we demonstrate its practical utility empirically.

To fine-tune an already trained black-box model $f_{\boldsymbol{\theta}}$, we combine the target prediction loss with the weighted intervenability term, which amounts to the following optimisation problem:

$$\begin{aligned} &\min_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{z}'} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{c},y)\sim\mathcal{D},\, \boldsymbol{c}'\sim\pi}\left[\left(1-\beta\right)\mathcal{L}^{y}\Big(g_{\boldsymbol{\psi}}\left(h_{\boldsymbol{\phi}}\left(\boldsymbol{x}\right)\right), y\Big) + \beta\mathcal{L}^{y}\Big(g_{\boldsymbol{\psi}}\left(\boldsymbol{z}'\right), y\Big)\right], \\ &\text{s.t. } \boldsymbol{z}' \in \arg\min_{\tilde{\boldsymbol{z}}} \lambda \mathcal{L}^{\boldsymbol{c}}\left(q_{\boldsymbol{\xi}}\left(\tilde{\boldsymbol{z}}\right), \boldsymbol{c}'\right) + d\left(\boldsymbol{z}, \tilde{\boldsymbol{z}}\right), \end{aligned} \tag{4}$$

where $\beta \in (0, 1]$ is the weight of the intervenability term. Note that for simplicity, we treat the probe's parameters $\boldsymbol{\xi}$ as fixed; however, since the outer optimisation problem is defined w.r.t. parameters $\boldsymbol{\phi}$, ideally, the probe would need to be optimised as the third, inner-most level of the problem. To avoid trilevel optimisation, we consider a special case of Equation 4 under $\beta = 1$. It then simplifies to $\min_{\boldsymbol{\psi},\boldsymbol{z}'} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{c},y)\sim\mathcal{D},\, \boldsymbol{c}'\sim\pi}[\mathcal{L}^{y}(g_{\boldsymbol{\psi}}\left(\boldsymbol{z}'\right), y)]$, s.t. $\boldsymbol{z}' \in \arg\min_{\tilde{\boldsymbol{z}}} \lambda \mathcal{L}^{\boldsymbol{c}}\left(q_{\boldsymbol{\xi}}\left(\tilde{\boldsymbol{z}}\right), \boldsymbol{c}'\right) + d\left(\boldsymbol{z}, \tilde{\boldsymbol{z}}\right)$. Thus, the parameters of $h_{\boldsymbol{\phi}}$ do not need to be optimised, and, hence, the probing function can be left fixed, as activations $\boldsymbol{z}$ are not affected by the fine-tuning. We consider this case to (i) computationally simplify the problem and (ii) keep the network's representations unchanged after fine-tuning for purposes of transfer learning for other downstream tasks.

3 Results

Experimental setup For controlled experiments, we evaluate our methods on the nonlinear synthetic tabular dataset adapted from [24], using diverse data-generating mechanisms (Figure D.1, Appendix D.1). We also utilise the Animals with Attributes 2 (AwA2) [21, 41] natural image dataset, originally intended for attribute-based classification and zero-shot learning. Finally, we study an applied setting with chest radiograph classification using the CheXpert [13] and MIMIC-CXR [15]. Appendix D contains further details on the datasets and preprocessing. We evaluate the proposed methods by conducting interventions and analysing model performance across different intervened concept subset sizes, given by percentages (*e.g.* see Figure 2), using the area under the receiver operating characteristic (AUROC) and precision-recall curves (AUPR) [7]. An improvement in performance with increasing subset sizes is expected if the model is intervenable.

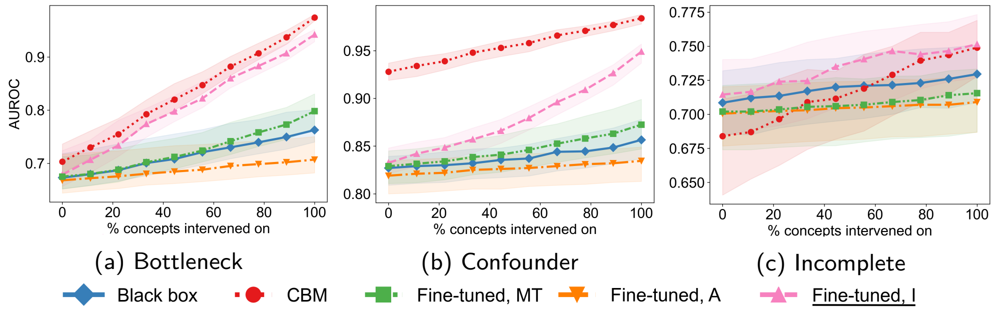


Figure 2: Effectiveness of interventions on black-box, fine-tuned models and CBMs on the synthetic tabular data under three different data-generating mechanisms, described in Figure D.1, Appendix D.1. Interventions were performed on the test set across ten independent simulations. Bold lines correspond to medians, and confidence bands are given by interquartile ranges.

A steeper curve is desirable since it suggests that the performance can be improved with fewer interventions. In addition to AUROC and AUPR, we report Brier scores [3].

In all experiments, we train a standard neural network ($\mathrm{B{\scriptsize LACK\ BOX}}$) without concept knowledge, *i.e.* on the dataset of tuples $\{(\boldsymbol{x}_i, y_i)\}_i$. We utilise our technique for intervening *post hoc* by training a probe to predict concepts and editing the network's activations (Equation 1). As an interpretable baseline, we consider the vanilla concept bottleneck model (CBM) [19] trained using joint optimisation by minimising the weighted sum of the target and concept prediction losses. Finally, as the primary method of interest, we apply our fine-tuning for intervenability technique ($\mathrm{F{\scriptsize INE\text{-}TUNED}}$, I; Equation 4) on the annotated validation set $\{(\boldsymbol{x}_i, \boldsymbol{c}_i, y_i)\}_i$. As a common-sense baseline, we fine-tune the black-box classifier by training a probe to predict the concepts from intermediate representations ($\mathrm{F{\scriptsize INE\text{-}TUNED}}$, MT). This approach amounts to multitask (MT) learning with hard weight sharing [34]. As another baseline method, we fine-tune the black box by appending concepts to the network's activations ($\mathrm{F{\scriptsize INE\text{-}TUNED}}$, A). Detailed formulation and implementation descriptions can be found in Appendices E and F.

Results on Synthetic Data Figure 2 shows intervention results across ten independent simulations for three generative mechanisms on the synthetic tabular data (Appendix D.1). The results, w.r.t. AUROC, demonstrate that the proposed intervention procedure can enhance the predictive performance of a black-box neural network at test time because intervening on larger concept subset sizes leads to an increase in performance. However, interventions are more effective in CBMs compared to untuned black-box classifiers. Models explicitly fine-tuned for intervenability show significant improvements in performance, particularly in the "*bottleneck*" and "*incomplete*" settings. In cases with an incomplete concept set, black boxes outperform CBMs, and fine-tuning with intervenability enhances intervention effectiveness while maintaining the performance gap. Other fine-tuning strategies are less effective or even harmful, resulting in lower AUROCs. Similar results are reported for AUPR in Appendix G.2. Additionally, Table 1 shows that, without interventions, all models achieve comparable AUROCs and AUPRs at the target prediction, while fine-tuning with intervenability leads to better-calibrated probabilistic predictions with lower Brier scores. Thus, fine-tuning helps reduce false overconfidence in neural networks [10], as also suggested by calibration curves in Appendix G.1.

Results on AwA2 We evaluate the effect on natural images using the AwA2 dataset. The intervention results are in Figure 3(a). Despite the dataset's simplicity, which contributes to generally high metrics and superior performance of the CBMs, untuned black-box models also benefit from concept-based interventions. As in the synthetic experiments, fine-tuning for intervenability notably enhances black-box model performance, bridging the gap to the CBMs. Ablation studies on untuned black-box models investigate the impact of different hyperparameters (Figures 3(b-d)). Although interventions are always effective, higher values of the λ-parameter expectedly lead to a steeper performance increase. The choice of intervention strategy [37] also affects the performance. The two strategies compared are explained in Appendix F and defined formally by Algorithms F.1–F.2. Additionally, using a nonlinear probe for interventions results in a notable improvement. Similar ablations are described in Appendix G.2

Table 1: Test-set performance without interventions on the synthetic, AwA2, and chest X-ray datasets. For black-box models, concepts were predicted via a linear probe. Results are reported as averages and standard deviations across ten seeds. For concepts, performance metrics were averaged. Best results are reported in **bold**, second best are in *italics*.

Dataset	Model	Concepts			Target		
		AUROC	AUPR	Brier	AUROC	AUPR	Brier
Synthetic	Black box	0.716±0.018	0.710±0.017	0.208±0.006	0.686±0.043	0.675±0.046	0.460±0.003
	CBM	**0.837±0.008**	**0.835±0.008**	*0.196±0.006*	**0.713±0.040**	**0.700±0.038**	*0.410±0.012*
	Fine-tuned, A	—	—	—	0.682±0.047	0.668±0.046	0.470±0.004
	Fine-tuned, MT	*0.784±0.013*	*0.780±0.014*	**0.186±0.006**	0.687±0.046	0.668±0.043	0.471±0.003
	Fine-tuned, I	0.716±0.018	0.710±0.017	0.208±0.006	*0.695±0.051*	*0.685±0.051*	**0.285±0.014**
AwA2	Black box	*0.987±0.001*	*0.967±0.002*	0.035±0.001	0.987±0.003	0.863±0.009	0.287±0.010
	CBM	**0.993±0.001**	**0.979±0.002**	**0.025±0.001**	*0.988±0.001*	**0.892±0.005**	**0.234±0.009**
	Fine-tuned, A	—	—	—	0.986±0.002	0.857±0.008	0.309±0.012
	Fine-tuned, MT	*0.987±0.001*	*0.967±0.002*	*0.033±0.001*	0.983±0.004	0.828±0.015	0.329±0.024
	Fine-tuned, I	*0.987±0.001*	*0.967±0.002*	0.035±0.001	**0.990±0.003**	*0.881±0.010*	*0.268±0.010*
CheXpert	Black box	0.665±0.003	0.257±0.003	*0.097±0.001*	0.785±0.011	0.911±0.006	0.305±0.009
	CBM	**0.723±0.005**	**0.322±0.003**	0.116±0.001	*0.786±0.009*	*0.919±0.006*	0.375±0.013
	Fine-tuned, A	—	—	—	0.749±0.008	0.891±0.005	0.329±0.013
	Fine-tuned, MT	*0.684±0.003*	*0.275±0.003*	**0.094±0.001**	0.768±0.019	0.901±0.012	*0.297±0.012*
	Fine-tuned, I	0.668±0.004	0.257±0.003	*0.097±0.001*	**0.819±0.009**	**0.938±0.004**	**0.201±0.007**
MIMIC-CXR	Black box	0.743±0.006	0.170±0.004	*0.046±0.001*	*0.789±0.006*	*0.706±0.009*	0.444±0.003
	CBM	*0.744±0.006*	**0.224±0.003**	0.053±0.001	0.765±0.007	0.699±0.006	*0.427±0.003*
	Fine-tuned, A	—	—	—	0.773±0.009	0.665±0.013	0.459±0.004
	Fine-tuned, MT	**0.748±0.008**	*0.187±0.003*	**0.045±0.001**	0.785±0.006	0.696±0.009	0.450±0.008
	Fine-tuned, I	*0.744±0.005*	0.172±0.005	*0.046±0.001*	**0.808±0.007**	**0.733±0.009**	**0.314±0.015**

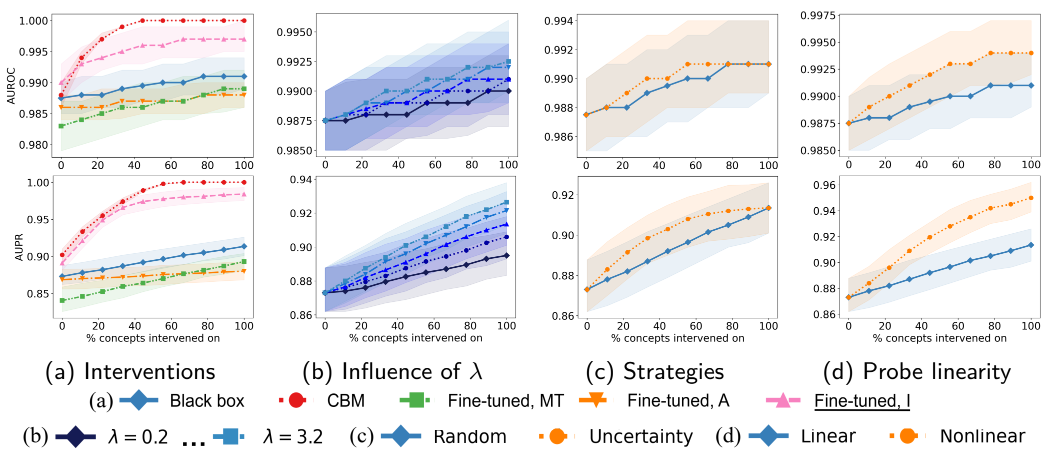
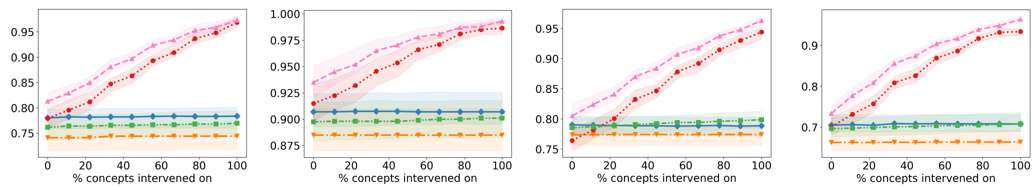


Figure 3: Intervention results on the AwA2 dataset w.r.t. AUROC (*top*) and AUPR (*bottom*). (a) Comparison among black-box, fine-tuned models, and CBMs. (b) Intervention results for the untuned black-box model under varying values of $\lambda \in \{0.2, 0.4, 0.8, 1.6, 3.2\}$ (Equation 3). **Darker** colours correspond to lower values. (c) Comparison between **random-subset** (Algorithm F.1) and **uncertainty-based** (Algorithm F.2) intervention strategies. (d) Comparison between **linear** and **nonlinear** probing functions.

for the synthetic data. Overall, the method has several hyperparameters, which might require careful tuning. These ablations show that the intervention procedure is relatively effective and stable across most values. In the rest of our experiments, we focus on a simpler configuration with the linear probe, random-subset intervention strategy, and a moderate λ-value to provide proof-of-concept results without extensive tuning. Lastly, Table 1 reports evaluation metrics without interventions for target and concept prediction, showing comparable performance across methods due to the dataset size and task simplicity.

Application to Chest X-rays To showcase the practicality of our approach, we present empirical findings in Figure 4 on two chest X-ray datasets, MIMIC-CXR and CheXpert. In both, untuned black-box neural networks are not intervenable. A plausible explanation could be that black



(a) CheXpert, AUROC (b) CheXpert, AUPR (c) MIMIC-CXR, AUROC (d) MIMIC-CXR, AUPR

Black box CBM Fine-tuned, MT Fine-tuned, A Fine-tuned, I

Figure 4: Intervention results w.r.t. AUROC (a, c) and AUPR (b, d) for black-box, fine-tuned models and CBMs on the CheXpert (a, b) and MIMIC-CXR (c, d) datasets.

boxes do not rely on the concept variables in their predictions. On the contrary, after fine-tuning for intervenability, the models' predictive performance and intervention effectiveness improve significantly. However, baselines (Fine-tuned, MT and Fine-tuned, A) are not intervenable, potentially due to the complexity of this task. These data feature instance-level concept labels, and the final predictions of black-box models may not rely as strongly on the attributes. Table 1 further shows that, while CBMs perform well at concept prediction, fine-tuned models excel at target classification.

4 Conclusion

This work has formalised intervenability as a measure of the effectiveness of concept-based interventions. It has also introduced techniques for instance-specific concept-based interventions on neural networks *post hoc* and fine-tuning the black boxes to improve their intervenability.

In contrast to interpretable models such as CBMs [19], our method circumvents the need for concept labels during training, which can be a substantial challenge in practical applications. Unlike recent works on converting black boxes into CBMs *post hoc* [43, 31], we propose an effective intervention method that is faithful to the original architecture and representations. Our approach does not impose restrictions on the size of the bottleneck layer since it uses a probing function defined on the concept set and any chosen layer.

Empirically, we demonstrated that black-box models trained without explicit concept knowledge are intervenable on synthetic tabular and natural image data, given an annotated validation set. We also showed that fine-tuning for intervenability improved the effectiveness of the interventions, bringing black boxes on par with CBMs, and led to better-calibrated predictions. Additionally, we explored the practical applicability of our techniques in chest X-ray classification. In this more realistic setting, black-box classifiers were not directly intervenable. However, the proposed fine-tuning procedure alleviated this limitation.

Acknowledgments and Disclosure of Funding

The authors would like to thank Dr. Ece Ozkan, Department of Brain and Cognitive Sciences, MIT, for insightful discussions. RM is supported by the SNSF grant #320038189096. SL and MV are supported by the Swiss State Secretariat for Education, Research, and Innovation (SERI) under contract number MB22.00047.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2016. *arXiv:1610.01644*.

[2] Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, 2022.

[3] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[4] Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5948–5955, 2023.

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

[6] Katherine M. Collins, Matthew Barker, Mateo Espinosa Zarlenga, Naveen Raman, Umang Bhatt, Mateja Jamnik, Ilia Sucholutsky, Adrian Weller, and Krishnamurthy Dvijotham. Human uncertainty in concept-based ai systems, 2023. *arXiv:2303.12872*.

[7] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233—240, New York, NY, USA, 2006. Association for Computing Machinery.

[8] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017. *arXiv:1702.08608*.

[9] Amirata Ghorbani, James Wexler, James Y. Zou, and Been Kim. Towards automatic concept-based explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 9277—-9286. Curran Associates, Inc., 2019.

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017.

[11] Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[14] Menglin Jia, Bor-Chun Chen, Zuxuan Wu, Claire Cardie, Serge Belongie, and Ser-Nam Lim. Rethinking nearest neighbors for visual classification, 2021. *arXiv:2112.08459*.

[15] Alistair E. W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih-ying Deng, Roger G. Mark, and Steven Horng. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6(1):317, 2019.

[16] 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 Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2668–2677. PMLR, 2018.

[17] Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16521–16540. PMLR, 2023.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[19] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348, Virtual, 2020. PMLR.

[20] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372, Kyoto, Japan, 2009. IEEE.

[21] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009. IEEE.

[22] Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks. In *2018 IEEE International Test Conference (ITC)*. IEEE, 2018.

[23] Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models, 2021. *arXiv:2106.13314*.

[24] Ričards Marcinkevičs, Patricia Reis Wolfertstetter, Ugne Klimiene, Kieran Chin-Cheong, Alyssia Paschke, Julia Zerres, Markus Denzinger, David Niederberger, Sven Wellmann, Ece Ozkan, Christian Knorr, and Julia E. Vogt. Interpretable and intervenable ultrasonography-based machine learning models for pediatric appendicitis, 2023. *arXiv:2302.14460*.

[25] Emanuele Marconato, Andrea Passerini, and Stefano Teso. GlanceNets: Interpretable, leak-proof concept-based models, 2022. *arXiv:2205.15612*.

[26] Diego Marcos, Ruth Fong, Sylvain Lobry, Rémi Flamary, Nicolas Courty, and Devis Tuia. Contextual semantic interpretability. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomás Pajdla, and Jianbo Shi, editors, *Computer Vision - ACCV 2020 - 15th Asian Conference on Computer Vision, Revised Selected Papers, Part IV*, volume 12625 of *Lecture Notes in Computer Science*, pages 351–368. Springer, 2020.

[27] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended?, 2021. *arXiv:2105.04289*.

[28] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.

[29] Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. Deep k-means: Jointly clustering with k-means and learning representations. *Pattern Recognition Letters*, 138:185–192, 2020.

[30] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 607–617, New York, NY, USA, 2020. Association for Computing Machinery.

[31] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The 11th International Conference on Learning Representations*, 2023.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, Red Hook, NY, United States, 2019. Curran Associates, Inc.

[33] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[34] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017. *arXiv:1706.05098*.

[35] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022.

[36] Ivaxi Sheth, Aamer Abdul Rahman, Laya Rafiee Sevyeri, Mohammad Havaei, and Samira Ebrahimi Kahou. Learning from uncertain concepts via test time interventions. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*, 2022.

[37] Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31504–31520. PMLR, 2023.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[39] David Steinmann, Wolfgang Stammer, Felix Friedrich, and Kristian Kersting. Learning to intervene on concept bottlenecks, 2023. *arXiv:2308.13453*.

[40] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31, 2017.

[41] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2019.

[42] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20554–20565. Curran Associates, Inc., 2020.

[43] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The 11th International Conference on Learning Representations*, 2023.

Appendix

A Related Work

In this section, we provide a comprehensive overview of the related works that our method builds on. Particularly, we focus on concept-based models; then, we describe previous efforts on improving concept bottleneck models and their relevant extensions, and, finally, we survey the literature on the intervention procedure in CBMs.

Concept-based Models & Explanations The use of high-level attributes in predictive models has been well-explored in computer vision [21, 20]. Recent efforts have focused on explicitly incorporating concepts in neural network architectures [19, 26], producing high-level *post hoc* explanations by quantifying the network's sensitivity to the given attributes [16], probing [1, 2] and de-correlating and aligning the network's latent space with concept variables [5]. To alleviate the assumption of being given interpretable concepts, some works have explored automated concept discovery prior to *post hoc* explanation [9, 42].

Concept Bottlenecks & Their Extensions Concept bottleneck models [19] have sparked a renewed interest in concept-based classification methods. Many related works have described the inherent limitations of this model class and attempted to address them. For example, it has been observed that CBMs may not always learn meaningful relationships between the concept and input spaces [27]. Similarly, in [23], the authors identify the issue of information leakage in concept predictions. Solutions to this challenge include generative approaches [25] and learning residual relationships between the features and labels that are not captured by the given concept set [11, 35, 43, 24]. Another line of research has investigated modelling uncertainty and probabilistic extensions of the CBMs [6, 17]. Most related to the current work are the techniques for converting already trained black-box neural networks into CBMs *post hoc* [43, 31] by keeping the network's backbone and projecting its activations into the concept bottleneck layer.

Concept-based Interventions As mentioned, CBMs allow for concept-based instance-specific interventions. Several follow-up works have studied interventions in further detail. In [4] and [36], the authors introduce adaptive intervention policies to further improve the predictive performance of the CBMs at the test time. In a similar vein, some works [39] have proposed learning to detect mistakes in the predicted concepts and, thus, learning intervention strategies. Finally, there have been empirical investigations of different intervention procedures across various settings [37].

B Fine-tuning for Intervenability

Algorithm B.1 contains the detailed pseudocode for fine-tuning for intervenability described in Section 2.3. Recall that the black-box model $f_{\boldsymbol{\theta}}$ is fine-tuned using a combination of the target prediction loss and intervenability defined in Equation 3. The implementation below applies to the special case of $\beta = 1$, which leads to the simplified loss. Importantly, in this case, the parameters $\boldsymbol{\phi}$ are treated as fixed, and the probing function $q_{\boldsymbol{\xi}}$ does not need to be fine-tuned alongside the model. Lastly, note that, in Algorithm B.1, interventions are performed for whole batches of data points $\boldsymbol{x}_b$ using the procedure described in Section 2.1.

Algorithm B.1: Fine-tuning for Intervenability

Input: Trained black-box model $f_{\boldsymbol{\theta}} = \langle g_{\boldsymbol{\psi}}, h_{\boldsymbol{\phi}} \rangle$, probing function $q_{\boldsymbol{\xi}}$, concept prediction loss function $\mathcal{L}^{\boldsymbol{c}}$, target prediction loss function $\mathcal{L}^{y}$, validation set $\{(\boldsymbol{x}_i, \boldsymbol{c}_i, y_i)\}_{i=1}^{N}$, intervention strategy π, distance function d, hyperparameter value $\lambda > 0$, maximum number of steps E_I for the intervention procedure, parameter for the convergence criterion $\varepsilon_I > 0$ for the intervention procedure, learning rate $\eta_I > 0$ for the intervention procedure, number of fine-tuning epochs E, mini-batch size M, learning rate $\eta > 0$

Output: Fine-tuned model

Train the probing function $q_{\boldsymbol{\xi}}$ on the validation set,
i.e. $\boldsymbol{\xi} \leftarrow \arg\min_{\boldsymbol{\xi}'} \sum_{i=1}^{N} \mathcal{L}^{\boldsymbol{c}}\left(q_{\boldsymbol{\xi}'}\left(h_{\boldsymbol{\phi}}\left(\boldsymbol{x}_i\right), \boldsymbol{c}_i\right)\right)$ ▷ Step 1: Probing

for $e = 0$ *to* $E - 1$ **do**
  Randomly split $\{1, ..., N\}$ into mini-batches of size M given by $\mathcal{B}$
  for $b \in \mathcal{B}$ **do**
    $\boldsymbol{z}_b \leftarrow h_{\boldsymbol{\phi}}\left(\boldsymbol{x}_b\right)$
    $\hat{y}_b \leftarrow g_{\boldsymbol{\psi}}\left(\boldsymbol{z}_b\right)$
    $\hat{\boldsymbol{c}}_b \leftarrow q_{\boldsymbol{\xi}}\left(\boldsymbol{z}_b\right)$
    Sample $\boldsymbol{c}'_b \sim \pi$
    Initialise $\boldsymbol{z}'_b = \boldsymbol{z}_b$, $\boldsymbol{z}'_{b,\mathrm{old}} = \boldsymbol{z}_b + \varepsilon_I \boldsymbol{e}$, and $e_I = 0$ ▷ Step 2: Editing Representations
    while $\left\|\boldsymbol{z}'_b - \boldsymbol{z}'_{b,\mathrm{old}}\right\|_1 \geq \varepsilon_I$ ***and*** $e_I < E_I$ **do**
      $\boldsymbol{z}'_{b,\mathrm{old}} \leftarrow \boldsymbol{z}'_b$
      $\boldsymbol{z}'_b \leftarrow \boldsymbol{z}'_b - \eta_I \nabla_{\boldsymbol{z}'_b}\left[d(\boldsymbol{z}_b, \boldsymbol{z}'_b) + \lambda \mathcal{L}^{\boldsymbol{c}}\left(q_{\boldsymbol{\xi}}\left(\boldsymbol{z}'_b\right), \boldsymbol{c}'_b\right)\right]$ ▷ Equation 1
      $e_I \leftarrow e_I + 1$
    end
    $\hat{y}'_b \leftarrow g_{\boldsymbol{\psi}}\left(\boldsymbol{z}'_b\right)$ ▷ Step 3: Updating Output
    $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} - \eta \nabla_{\boldsymbol{\psi}} \mathcal{L}^{y}\left(\hat{y}'_b, y_b\right)$ ▷ Equation 4
  end
end

return $f_{\boldsymbol{\theta}}$

C Further Remarks & Discussion

Below, we provide further remarks and discussion on the intervention procedure, intervenability, and the limitations and potential extensions of the current work.

Intervention Procedure The intervention procedure entails a few design choices, including the (non)linearity of the probing function, the distance function in the objective from Equation 1, and the tradeoff between consistency and proximity determined by λ from Equation 1. We explore some of these choices empirically in our ablation experiments (see Figure 3 and Appendix G). Naturally, interventions performed on black-box models using our method are meaningful in so far as the activations of the neural network are correlated with the given high-level attributes and the probing function $q_{\boldsymbol{\xi}}$ can be trained to predict these attribute values accurately. Otherwise, edited representations and updated predictions are likely to be spurious and may harm the model's performance.

Should All Models Be Intervenable? Intervenability (Equation 3), in combination with the probing function, can be used to evaluate the interpretability of a black-box predictive model and help understand whether (i) learnt representations capture information about given human-understandable attributes and whether (ii) the network utilises these attributes and can be interacted with. However, a black-box model does not always need to be intervenable. For instance, when the given concept set is not predictive of the target variable, the black box trained using supervised learning should not and probably would not rely on the concepts. On the other hand, if the model's representations are nearly perfectly correlated with the attributes, providing the ground truth should not significantly impact the target prediction loss. Lastly, the model's intervenability may depend on the chosen intervention strategy, which may not always lead to the expected decrease in the loss.

Limitations & Future Work The current work opens many avenues for future research and improvements. Firstly, the variant of the fine-tuning procedure considered in this paper does not affect the neural network's representations. However, it would be interesting to investigate a more general formulation wherein all model and probe parameters are fine-tuned end-to-end. According to our empirical findings, the choice of intervention strategy, hyperparameters, and probing function can influence the effectiveness of interventions. A more in-depth experimental investigation of these aspects is warranted. Furthermore, we only considered having a single fixed strategy throughout fine-tuning, whereas further improvement could come from learning an optimal strategy alongside fine-tuned weights. Lastly, the proposed techniques rely on the annotated validation data to fit a probe and could benefit from (semi-)automated concept discovery, *e.g.* using multimodal models.

D Datasets

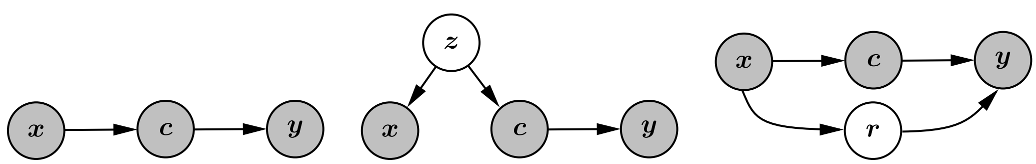
Below, we present further details about the datasets and preprocessing involved in the experiments. AwA2, CheXpert, and MIMIC-CXR datasets are publicly available. Table D.1 provides a brief summary of the datasets. All datasets were divided according to the 60%-20%-20% train-validation-test split. Fine-tuning has been performed on the validation data and evaluation on the test set.

Table D.1: Dataset summary. After any filtering or preprocessing, N is the total number of data points; p is the input dimensionality; and K is the number of concept variables.

Dataset	Data type	N	p	K
Synthetic	Tabular	50,000	1,500	30
AwA2	Image	37,322	224×224	85
CheXpert	Image	49,408	224×224	13
MIMIC-CXR	Image	54,276	224×224	13

D.1 Synthetic Tabular Data

To perform experiments in a controlled manner, we generate synthetic nonlinear tabular data using the procedure adapted from [24]. We explore three settings corresponding to different data-generating mechanisms (Figure D.1): (a) *bottleneck*, (b) *confounder*, and (c) *incomplete*. The first scenario directly matches the inference graph of the vanilla CBM [19]. The *confounder* is a setting wherein $\boldsymbol{c}$ and $\boldsymbol{x}$ are generated by an unobserved confounder $\boldsymbol{z}$ and y is generated by $\boldsymbol{c}$. Lastly, *incomplete* is a scenario with incomplete concepts, where $\boldsymbol{c}$ does not fully explain the variance in y. Here, unexplained variance is modelled as a latent variable $\boldsymbol{r}$ via the path $\boldsymbol{x} \rightarrow \boldsymbol{r} \rightarrow y$. Unless mentioned otherwise, we mainly focus on the simplest scenario shown in Figure D.1(a). Below, we outline each generative process in detail. Throughout this appendix, let N, p, and K denote the number of independent data points $\{(\boldsymbol{x}_i, \boldsymbol{c}_i, y_i)\}_{i=1}^N$, covariates, and concepts, respectively. Across all experiments, we set $N = 50{,}000$, $p = 1{,}500$, and $K = 30$.



(a) Bottleneck (b) Confounder (c) Incomplete

Figure D.1: Data-generating mechanisms for the synthetic dataset summarised as graphical models. Each node corresponds to a random variable. Observed variables are shown in grey.

Bottleneck In this setting, the covariates $\boldsymbol{x}_i$ generate binary-valued concepts $\boldsymbol{c}_i \in \{0, 1\}^K$, and the binary-valued target y_i depends on the covariates exclusively via the concepts. The generative process is as follows:

1. Randomly sample $\boldsymbol{\mu} \in \mathbb{R}^p$ s.t. $\mu_j \sim \mathrm{Uniform}\,(-5, 5)$ for $1 \leq j \leq p$.
2. Generate a random symmetric, positive-definite matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$.
3. Randomly sample a design matrix $\boldsymbol{X} \in \mathbb{R}^{N \times p}$ s.t. $\boldsymbol{X}_{i,:} \sim \mathcal{N}_p\,(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.[3]
4. Let $h : \mathbb{R}^p \rightarrow \mathbb{R}^K$ and $g : \mathbb{R}^K \rightarrow \mathbb{R}$ be randomly initialised multilayer perceptrons with ReLU nonlinearities.
5. Let $c_{i,k} = \mathbf{1}_{\left\{[h(\boldsymbol{X}_{i,:})]_k \geq m_k\right\}}$, where $m_k = \mathrm{median}\left(\left\{[h\,(\boldsymbol{X}_{l,:})]_k\right\}_{l=1}^N\right)$, for $1 \leq i \leq N$ and $1 \leq k \leq K$.
6. Let $y_i = \mathbf{1}_{\{g(\boldsymbol{c}_i) \geq m_y\}}$, where $m_y = \mathrm{median}\left(\{g\,(\boldsymbol{c}_i)\}_{l=1}^N\right)$, for $1 \leq i \leq N$.

[3] $\boldsymbol{X}_{i,:}$ refers to the i-th row of the design matrix, *i.e.* the covariate vector $\boldsymbol{x}_i$

Confounder Another scenario we consider is where $\boldsymbol{x}$ and $\boldsymbol{c}$ are generated by an unobserved confounder:

1. Randomly sample $\boldsymbol{Z} \in \mathbb{R}^{N \times K}$ s.t. $z_{i,k} \sim \mathcal{N}(0, 1)$ for $1 \leq i \leq N$ and $1 \leq k \leq K$.
2. Let $c_{i,k} = \mathbf{1}_{\{z_{i,k} \geq 0\}}$ for $1 \leq i \leq N$ and $1 \leq k \leq K$.
3. Let $h : \mathbb{R}^K \to \mathbb{R}^p$ and $g : \mathbb{R}^K \to \mathbb{R}$ be randomly initialised multilayer perceptrons with ReLU nonlinearities.
4. Let $\boldsymbol{x}_i = h(\boldsymbol{Z}_{i,:})$ for $1 \leq i \leq N$.
5. Let $y_i = \mathbf{1}_{\{\sigma(g(\boldsymbol{c}_i) \geq 1/2)\}}$ for $1 \leq i \leq N$, where σ denotes the sigmoid function.

Incomplete Last but not least, to simulate the incomplete concept set scenario, where a part of concepts are latent, we slightly adjust the procedure from the *bottleneck* setting above:

1. Follow steps 1–3 from the *bottleneck* procedure.
2. Let $h : \mathbb{R}^p \to \mathbb{R}^{K+J}$ and $g : \mathbb{R}^{K+J} \to \mathbb{R}$ be randomly initialised multilayer perceptrons with ReLU nonlinearities, where J is the number of unobserved concept variables.
3. Let $u_{i,k} = \mathbf{1}_{\left\{[h(\boldsymbol{X}_{i,:})]_k \geq m_k\right\}}$, where $m_k = \operatorname{median}\left(\left\{[h(\boldsymbol{X}_{l,:})]_k\right\}_{l=1}^N\right)$, for $1 \leq i \leq N$ and $1 \leq k \leq K + J$.
4. Let $\boldsymbol{c}_i = \boldsymbol{u}_{i,1:K}$ and $\boldsymbol{r}_i = \boldsymbol{u}_{i,(K+1):(K+J)}$ for $1 \leq i \leq N$.
5. Let $y_i = \mathbf{1}_{\{g(\boldsymbol{u}_i) \geq m_y\}}$, where $m_y = \operatorname{median}\left(\{g(\boldsymbol{u}_i)\}_{l=1}^N\right)$, for $1 \leq i \leq N$.

Note that, in steps 3–5 above, $\boldsymbol{u}_i$ corresponds to the concatenation of $\boldsymbol{c}_i$ and $\boldsymbol{r}_i$. Across all experiments, we set $J = 90$.

D.2 Animals with Attributes 2

Animals with Attributes 2[4] dataset [21, 41] serves as a natural image benchmark in our experiments. It comprises 37,322 images of 50 animal classes (species), each associated with 85 binary attributes utilised as concepts. An apparent limitation of this dataset is that the concept labels are shared across whole classes, similar to the Caltech-UCSD Birds experiment from the original work in [19]. Thus, AwA2 offers a simplified setting for transfer learning across different classes and is designed to address attribute-based classification and zero-shot learning challenges. In our evaluation, we used all the images in the dataset without any specialised preprocessing or preselection. All images were rescaled to 224×224 pixels.

D.3 Chest X-ray Datasets

As mentioned, we conducted an empirical evaluation on two real-world chest X-ray datasets: CheXpert [13] and MIMIC-CXR [15]. The former includes over 220,000 chest radiographs from 65,240 patients at the Stanford Hospital.[5] These images are accompanied by 14 binary attributes extracted from radiologist reports using the CheXpert labeller [13], a model trained to predict these attributes. MIMIC-CXR is another publicly available dataset containing chest radiographs in DICOM format, paired with free-text radiology reports.[6] It comprises more than 370,000 images associated with 227,835 radiographic studies conducted at the Beth Israel Deaconess Medical Center, Boston, MA, involving 65,379 patients. Similar to CheXpert, the same labeller was employed to extract the same set of 14 binary labels from the text reports. Notably, some concepts may be labelled as uncertain. Similar to [4], we designate the *Finding/No Finding* attribute as the target variable for classification and utilise the remaining labels as concepts. In our implementation, we remove all the samples that contain uncertain labels and we discard multiple visits of the same patient, keeping only the last acquired recording per subject for both datasets. All images were cropped to a square aspect ratio and rescaled to 224×224 pixels. Additionally,

[4] `https://cvml.ista.ac.at/AwA2/`
[5] `https://stanfordmlgroup.github.io/competitions/chexpert/`
[6] `https://physionet.org/content/mimic-cxr/2.0.0/`

augmentations were applied during training, namely, random affine transformations, including rotation up to 5 degrees, translation up to 5% of the image's width and height, and shearing with a maximum angle of 5 degrees. We also include a random horizontal flip augmentation to introduce variation in the orientation of recordings within the dataset.

E Baselines and Methods

This section details the formulation of the methods compared in this work.

As a starting point, as discussed before, we train a standard neural network without incorporating the concept knowledge, *i.e.* on the dataset of tuples $\{(\boldsymbol{x}_i, y_i)\}_i$. We then apply our *post hoc* intervention technique by training a probe to predict concepts and modifying the network's activations using Equation 1 in Section 2.1. As an interpretable baseline, we consider the vanilla concept bottleneck model [19]. Across all experiments, we restrict ourselves to the joint bottleneck version, which minimises the weighted sum of the target and concept prediction losses: $\min_{\boldsymbol{\phi},\boldsymbol{\psi}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{c},y)\sim\mathcal{D}} \left[\mathcal{L}^y\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right), y\right) + \alpha\mathcal{L}^{\mathbf{c}}\left(h_{\boldsymbol{\phi}}\left(\boldsymbol{x}\right), \boldsymbol{c}\right)\right]$, where $\alpha > 0$ is a hyperparameter controlling the tradeoff between the two loss terms. Finally, as the primary method of interest, we apply our fine-tuning for the intervenability technique (Equation 4, Section 2.3) on the annotated validation set $\{(\boldsymbol{x}_i, \boldsymbol{c}_i, y_i)\}_i$.

In order to better understand the performance of our method, we implement two common-sense baselines. First, we fine-tune the black-box classifier by training a probe to predict the concepts from intermediate representations. This approach amounts to multitask learning with hard weight sharing [34]. Specifically, the model is fine-tuned by minimising the following MT loss: $\min_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\xi}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{c},y)\sim\mathcal{D}} \left[\mathcal{L}^y\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right), y\right) + \alpha\mathcal{L}^{\mathbf{c}}\left(q_{\boldsymbol{\xi}}\left(h_{\boldsymbol{\phi}}\left(\boldsymbol{x}\right)\right), \boldsymbol{c}\right)\right]$. As another baseline method, we fine-tune the black box by appending concepts to the network's activations. At test time, unknown concept values are set to 0.5. To prevent overfitting and handle concept missingness, randomly chosen concept variables are masked during training. Formally, the objective is given by $\min_{\tilde{\boldsymbol{\psi}}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{c},y)\sim\mathcal{D}}[\mathcal{L}^y(\tilde{g}_{\tilde{\boldsymbol{\psi}}}\left(\left[h_{\boldsymbol{\phi}}\left(\boldsymbol{x}\right), \boldsymbol{c}\right]\right), y)]$, where $\tilde{g}$ takes as input concatenated activation and concept vectors. Note that, for this baseline, the parameters $\boldsymbol{\phi}$ remain fixed during fine-tuning.

F Implementation Details

This section provides implementation details, such as network architectures and intervention and fine-tuning procedure hyperparameter configurations. All models and procedures were implemented using PyTorch (v 1.12.1) [32] and scikit-learn (v 1.0.2) [33].

Network & Probe Architectures For the synthetic tabular data, we utilise a fully connected neural network (FCNN) as the black-box model. Its architecture is summarised in Table F.1 in PyTorch-like pseudocode. For this classifier, probing functions are trained and interventions are performed on the activations of the third layer, *i.e.* the output after line **2** in Table F.1. For natural and medical image datasets, we use the ResNet-18 [12] with random initialisation followed by four fully connected layers and the sigmoid or softmax activation. Probing and interventions are performed on the activations of the second layer after the ResNet-18 backbone. For the CBMs, to facilitate fair comparison, we use the same architectures with the exception that the layers mentioned above were converted into bottlenecks with appropriate dimensionality and activation functions.

For fine-tuning, we utilise a single fully connected layer with an appropriate activation function as a linear probe and a multilayer perceptron with a single hidden layer as a nonlinear function. For evaluation on the test set (Table 1), we fit a logistic regression classifier from scikit-learn as a linear probe. The logistic regression is only used for evaluation purposes and not interventions.

Table F.1: Fully connected neural network architecture used as a black-box classifier in the experiments on the synthetic tabular data. `nn` stands for `torch.nn`; `F` stands for `torch.nn.functional`; `input_dim` corresponds to the number of input features.

	FCNN Classifier
1	`nn.Linear(input_dim, 256)`
	`F.relu()`
	`nn.Dropout(0.05)`
	`nn.BatchNorm1d(256)`
2	`for l in range(2):`
	`    nn.Linear(256, 256)`
	`    F.relu()`
	`    nn.Dropout(0.05)`
	`    nn.BatchNorm1d(256)`
3	`out = nn.Linear(256, 1)`
4	`torch.sigmoid()`

Interventions Unless mentioned otherwise, interventions on black-box models were performed using linear probes, the random-subset intervention strategy, and under $\lambda = 0.8$ (Equation 1). Recall that Figures 3 and G.3 provide ablation results on the influence of this hyperparameter. Despite some variability, the analysis shows that higher values of λ expectedly lead to more effective interventions. The choice of λ for our experiments was meant to represent the "average case", and no tuning was performed for this hyperparameter.

Similarly, we have mainly used a linear probing function and the simple random-subset intervention strategy to provide proof-of-concept results without extensive optimisation of the intervention strategy or the need for nonlinear probing. Thus, our primary focus was on demonstrating the intervenability of black-box models and showcasing the effectiveness of the fine-tuning method rather than an exhaustive hyperparameter search.

Intervention Strategies In ablation studies, we compare two intervention strategies (Figure 3) inspired by [37]: (i) random-subset and (ii) uncertainty-based. Herein, we provide a more formal definition of these procedures described as pseudocode in Algorithms F.1–F.2. Recall that given a data point $(\boldsymbol{x}, \boldsymbol{c}, y)$ and predicted values $\hat{\boldsymbol{c}}$ and $\hat{y}$, an intervention strategy defines a distribution over intervened concept values $\boldsymbol{c}'$. Random-subset strategy (Algorithm F.1) replaces predicted values for several concept variables (k) chosen uniformly at random with the ground truth. By contrast, the uncertainty-based strategy (Algorithm F.2) samples concept variables to be replaced with the ground-truth values without replacement with initial probabilities proportional to the concept prediction uncertainties, denoted by $\boldsymbol{\sigma}$. In our experiments, the components of $\hat{\boldsymbol{c}}$ are the outputs of the sigmoid function, and the uncertainties are computed as $\sigma_i = 1/\left(|\hat{c}_i - 0.5| + \varepsilon\right)$ [37] for $1 \leq i \leq K$, where $\varepsilon > 0$ is small.

Algorithm F.1: Random-subset Intervention Strategy

Input: A data point $(\boldsymbol{x}, \boldsymbol{c}, y)$, predicted concept values $\hat{\boldsymbol{c}}$, the number of concept variables to be intervened on $1 \leq k \leq K$
Output: Intervened concept values $\boldsymbol{c}'$

$\boldsymbol{c}' \leftarrow \hat{\boldsymbol{c}}$
Sample $\mathcal{I}$ uniformly at random from $\{\mathcal{S} \subseteq \{1, \ldots, K\} : |\mathcal{S}| = k\}$
$\boldsymbol{c}'_{\mathcal{I}} \leftarrow \boldsymbol{c}_{\mathcal{I}}$
return $\boldsymbol{c}'$

Algorithm F.2: Uncertainty-based Intervention Strategy

Input: A data point $(\boldsymbol{x}, \boldsymbol{c}, y)$, predicted concept values $\hat{\boldsymbol{c}}$, the number of concept variables to be intervened on $1 \leq k \leq K$
Output: Intervened concept values $\boldsymbol{c}'$

$\sigma_j \leftarrow 1/\left(|\hat{c}_j - 0.5| + \varepsilon\right)$ for $1 \leq j \leq K$, where $\varepsilon > 0$ is small
$\boldsymbol{\sigma} \leftarrow (\sigma_1 \quad \cdots \quad \sigma_K)$
$\boldsymbol{c}' \leftarrow \hat{\boldsymbol{c}}$
Sample k indices $\mathcal{I} = \{i_j\}_{j=1}^{k}$ s.t. each i_j is sampled without replacement from $\{1, \ldots, K\}$ with initial probabilities given by $(\boldsymbol{\sigma} + \varepsilon) / \left(K\varepsilon + \sum_{i=1}^{K} \sigma_i\right)$, where $\varepsilon > 0$ is small
$\boldsymbol{c}'_{\mathcal{I}} \leftarrow \boldsymbol{c}_{\mathcal{I}}$
return $\boldsymbol{c}'$

Fine-tuning for Intervenability The fine-tuning procedure outlined in Section 2.3 and detailed in Algorithm B.1 necessitates intervening on the representations throughout the optimisation. During fine-tuning, we utilise the random-subset intervention strategy (Algorithm F.1), *i.e.* interventions are performed on a subset of the concept variables by providing the ground-truth values. More concretely, interventions are performed on 50% of the concept variables chosen uniformly at random.

Fine-tuning Baselines The baseline methods described in the main text incorporate concept information in distinct ways. On the one hand, the multitask learning approach, FINE-TUNED, MT, utilises the entire batch of concepts at each iteration during fine-tuning. For this procedure, we set $\alpha = 1.0$ (recall that α controls the tradeoff between the target and concept prediction loss terms). On the other hand, the FINE-TUNED, A approach, which appends the concepts to the network's activations, does not use the complete concept set for each batch. In particular, before appending, concept values are randomly masked and set to 0.5 with a probability of 0.5. This practical trick is reminiscent of the dropout [38] and is meant to help the model remain intervenable and handle missing concept values.

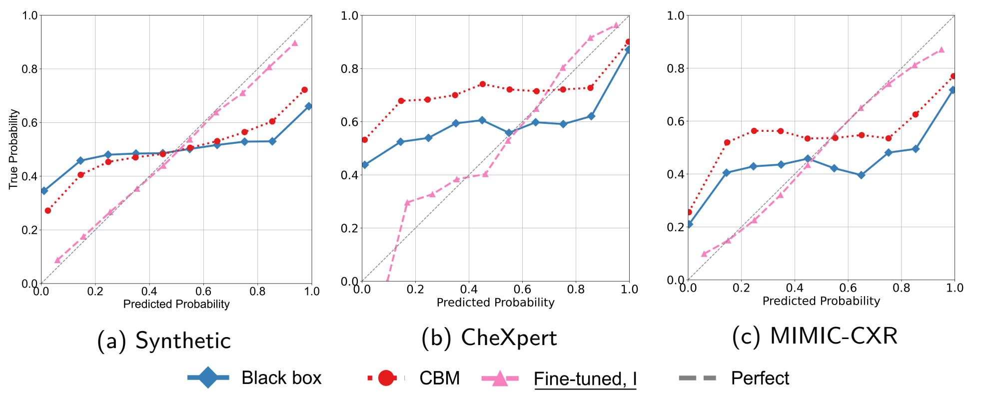
Hyperparameters Below, we list key hyperparameter configurations; the remaining details are documented in our code. For the synthetic data, CBMs and black-box classifiers are trained for 150 and 100 epochs, respectively, with a learning rate of 10^{-4} and a batch size of 64. Across all other experiments, CBMs are trained for 350 epochs and black-box models for 300 epochs with a learning rate of 10^{-4} halved midway through training and a batch size of 64. CBMs are trained using the joint optimisation procedure [19] under $\alpha = 1.0$, where α controls the tradeoff between the concept and target prediction losses. All probes were trained on the validation data for 150 epochs with a learning rate of 10^{-2} using the stochastic gradient descent (SGD) optimiser. Finally, all fine-tuning procedures were run for 150 epochs with a learning rate of 10^{-4} and a batch size of 64 using the Adam optimiser. At test time, interventions were performed on batches of 512 data points.

G Further Results

This section contains supplementary results and ablation experiments not included in the main body of the text.

G.1 Calibration Results

The fine-tuning approach introduced leads to better-calibrated predictions (Table 1), possibly, due to the regularising effect of intervenability. In this section, we further support this finding by visualising calibration curves for the binary classification tasks, namely, for the synthetic tabular data and chest radiograph datasets. Figure G.1 shows calibration curves for the fine-tuned model, untuned black box, and CBM averaged across ten seeds. We have omitted fine-tuning baselines for legibility since their predictions were comparably ill-calibrated as for the black box. The fine-tuned model consistently and considerably outperforms both the untuned black box and the CBM in all three binary classification tasks, as its curve is the closest to the diagonal, which corresponds to perfect calibration.



(a) Synthetic (b) CheXpert (c) MIMIC-CXR

Figure G.1: Analysis of the probabilities predicted by the **black box**, **fine-tuned black box**, and **CBM** on the (a) synthetic, (b) CheXpert, and (c) MIMIC-CXR. The calibration curves, averaged across ten seeds, display for each bin the true empirical probability of $y = 1$ against the probability predicted by the model. The gray dashed line corresponds to perfectly calibrated predictions.

G.2 Further Results on Synthetic Data

Figure G.2 supplements the intervention experiment results in Figure 2, Section 3, showing intervention curves w.r.t. AUPR under the three generative mechanisms for the synthetic data. The overall patterns and conclusions are similar to those observed w.r.t. AUROC (Figure 2).

Figure G.3 provides ablation experiment results obtained on the synthetic tabular data under the *bottleneck* generative mechanism shown in Figure D.1(a), similar to the results reported for AwA2 in Figure 3, Section 3. In Figure G.3(a), we plot black-box intervention results across varying values of the hyperparameter λ (Equation 1). As for AwA2, higher λs result in more effective interventions: this finding is expected since λ is the weight of the term penalising the inconsistency of the concept values predicted by the probe with the given values and, in the current implementation, interventions are performed using the ground truth. Interestingly, in Figure G.3(b), we observe no difference between the random subset and uncertainty-based intervention strategies. This could be explained by the fact that, in the synthetic dataset, all concepts by design are, on average, equally hard to predict and equally helpful in predicting the target variable (see Appendix D.1). Hence, the uncertainty score should not be as informative in this dataset, and the order of intervention on the concepts should have little effect. Finally, Figure G.3(c) suggests that nonlinear probing improves intervention effectiveness.

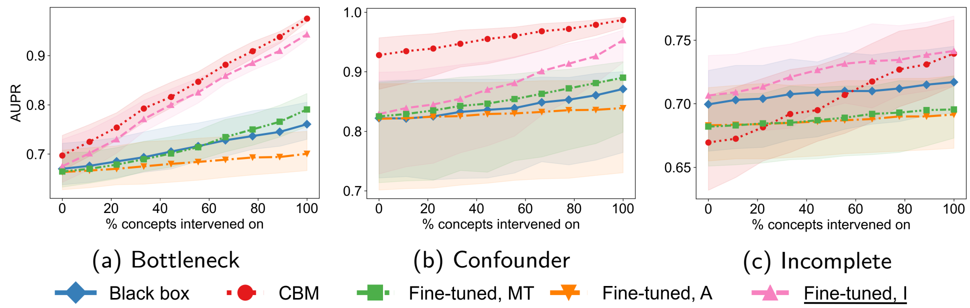


Figure G.2: Effectiveness of interventions w.r.t. AUPR on black-box and fine-tuned models and CBMs on the synthetic tabular data under three different data-generating mechanisms (Appendix D.1). Interventions were performed on the test set across ten independent simulations. Bold lines correspond to medians, and confidence bands are given by interquartile ranges.

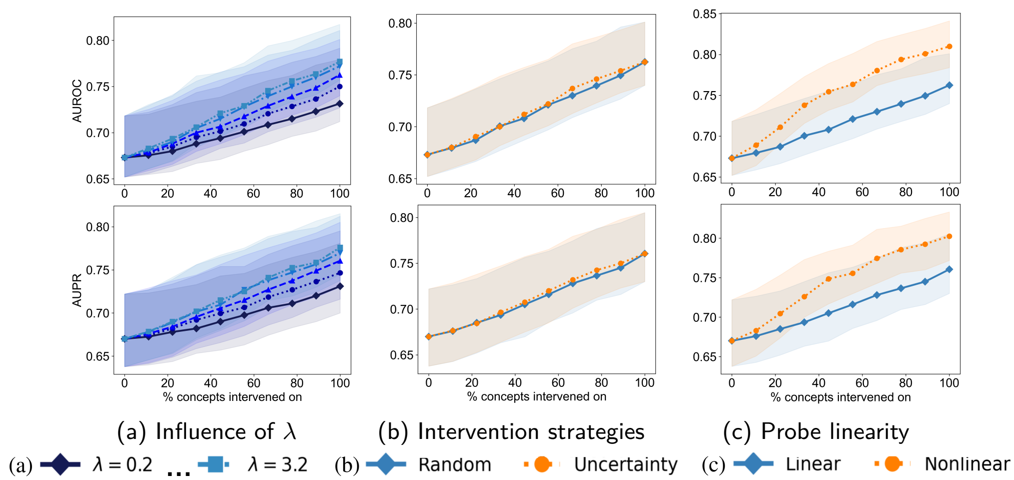


Figure G.3: Ablation study results w.r.t. AUROC (*top*) and AUPR (*bottom*) on the synthetic dataset. Bold lines correspond to medians, and confidence bands are given by interquartile ranges across ten independent simulations. (a) Intervention results for the untuned black-box model under varying values of $\lambda \in \{0.2, 0.4, 0.8, 1.6, 3.2\}$ (Equation 3). **Darker** colours correspond to lower values. (b) Comparison between **random-subset** and **uncertainty-based** intervention strategies. (c) Comparison between **linear** and **nonlinear** probing functions.