Improving Intervention Efficacy via Concept Realignment in Concept Bottleneck Models

Nishad Singhi[1] Jae Myung Kim[1] Karsten Roth[1] Zeynep Akata[2]
Primary contact: nishadsinghi@gmail.com

[1]Tübingen AI Center, University of Tübingen [2]Helmholtz München, TU München

Abstract. Concept Bottleneck Models (CBMs) ground image classification on human-understandable concepts to allow for interpretable model decisions as well as human interventions, in which expert users can modify misaligned concept choices to interpretably influence the decision of the model. However, existing approaches often require numerous human interventions per image to achieve strong performances, posing practical challenges in scenarios where obtaining human feedback is expensive. In this paper, we find that this is driven by an independent treatment of concepts during intervention, wherein a change of one concept does not influence the use of other ones. To address this issue, we introduce a trainable concept intervention realignment module, which leverages concept relations to realign concept assignments post-intervention. Across standard benchmarks, we find that concept realignment significantly improves intervention efficacy and reduces the number of interventions needed to reach a target classification performance or concept prediction accuracy. Moreover, it easily integrates into existing concept-based architectures without requiring changes to the models themselves. This reduced cost of human-model collaboration is crucial to enhance the feasibility of CBMs in resource-constrained environments. Our code is available at https://github.com/ExplainableML/concept_realignment.

Keywords: Concept Bottlenecks · Trustworthiness · Interventions

1 Introduction

Despite tremendous progress of Deep Learning (DL) techniques in research and applications, their adoption to high-stakes scenarios has been limited [9, 27, 29]. This is in large part due to unpredictable failure cases of deep models when transferring to unseen data or complex & ambiguous cases. The black-box nature of typical DL models further exacerbates this problem, as it makes understanding and debugging the decision-making process of these models difficult. Consequently, it becomes hard for human practitioners to trustworthily operate these models in scenarios with significant legal [5, 21] or ethical [4, 17] constraints. To foster trust, transparency in the decision-making process, and the ability to operate alongside expert feedback are required. In order to incorporate these desiderata into the design space of deep models, *Koh et al.* [9] introduced Concept Bottleneck Models (CBMs). These models break the decision process into

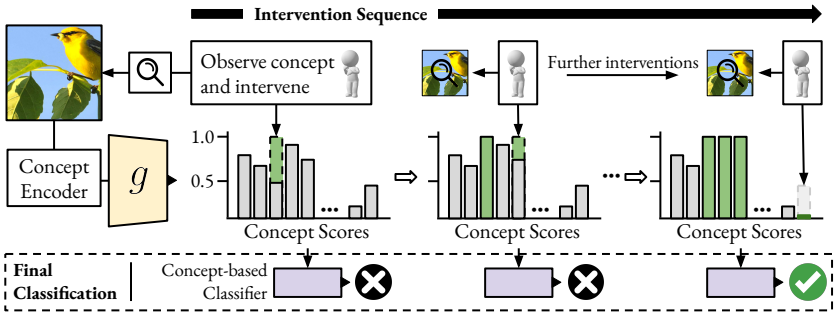


Fig. 1: Concept-based classification models allow for human intervention, where a human expert can correct specifically assigned concepts. However, to achieve satisfactory performance, concept-based classification models often require a large number of interventions, where each additional intervention requires costly human interaction.

the extraction of human-interpretable concepts (such as "white wings" and "orange beak" when classifying a seagull) from a given input, and a subsequent concept-grounded classifier operating on top of these concept predictions. While this allows users to peek into the model decision process - maybe even more importantly, it also uniquely allows for human-guided intervention and feedback integration at test time. This is done through concept interventions, wherein an expert user analyses predicted concepts and optionally replaces those they deem incorrect with ground-truth information (Fig. 1). Such interventions can significantly raise the performance and reliability of these models [9, 20, 27, 29], while offering a natural interface for human-AI collaboration. However, human annotation is expensive, especially when resources and access to expert knowledge are limited. Ideally, such concept models should operate well with minimal human input. This becomes particularly prevalent as CBMs (as well as follow-up extensions such as Concept Embedding Models (CEMs, [27])) often require numerous interventions in order to significantly boost model performance [9,27,29], as the set of concepts these models operate on can often be rather extensive. For example, on the widely used CUB benchmark (bird classification, [22]), it takes 13 interventions per image on overage to raise the accuracy of a baseline CBM model from around 68% to 90% (Fig. 4). In this work, we posit that a large part of this limited intervention efficacy can be traced back to the independent nature of how concept interventions are treated. This means that correcting for one concept (or a set of concepts) does not affect which other concepts are predicted for the same image. However, the occurrence of concepts in real life is often correlated, and informing the model about one concept should consequently influence the use of related ones. Not doing so means that we do not leverage human feedback to its full extent - intervening on one specific concept naturally gives additional context about the potential occurrence of other concepts, which should be taken into account in the final classification process. In particular, we study the extent of this crucial aspect when operating with concept-based models. Our study highlights how the use of a **simple *concept intervention realignment* module**, which learns from statistical concept relations, can effec-

tively and automatically realign concept values after an intervention (or multiple) have been performed. Our experiments reveal how our concept realignment seamlessly integrates into and improves existing CBM approaches (e.g. default CBMs [9], CEMs [27] or intervention-aware CEMs [29]), and can be deployed both jointly during the initial training of the concept model, and as a post-hoc trained realignment mechanism. Across three standard, real-world benchmarks (CUB [22], CelebA [11] and AwA2 [23]), we showcase consistent, in parts very significant improvements in intervention efficacy. Across concept prediction and overall classification accuracy, performance increases more rapidly with interventions as compared to baselines without realignment (reducing the number of interventions needed to reach a target performance by over 70%). Combined with its versatile usage and minimal additional resource requirements, we believe our insights into concept intervention realignment to be of high practical relevance, helping to drive down the cost of human-model collaboration and facilitate the corresponding practical deployment of concept-based models.

2 Related Works

Concept Bottleneck Models (CBMs) have been extensively studied since their introduction by [9]. [27] proposed Concept Embedding Models as a generalization, utilizing embedding vectors for concepts rather than scalar probabilities, thus enhancing task performance while maintaining interpretability. Recent efforts have explored methods to enhance CBMs without requiring explicit concept supervision during training, leveraging pre-trained vision backbones and language guidance [15, 25, 26]. [2] introduced Self-explaining Neural Networks (SENNs) for unsupervised concept learning, while [18] proposed CBM-AUC combining SENNs with CBMs. Probabilistic CBMs [8] were proposed to model uncertainty in concepts and final predictions. [12, 14] addressed concept leakage, while [6,13] aimed to alleviate it. Our work complements these efforts by enabling CBMs to update predictions of all concepts after human intervention.
Interventions on CBMs. [9] showed that intervening on random concepts enhances classification in CBMs. [3] and [19] proposed uncertainty-based strategies for interventions. [20] extensively studied concept selection strategies, focusing on task performance and execution cost. [29] introduced interventions during training to enhance model receptiveness to test-time interventions. We complement existing methods by updating predictions of all concepts following expert interventions. Concurrently, [24] proposed Energy-based CBMs to automate concept prediction updates. In comparison, our method benefits from higher simplicity, improved performance, and seamless integration with existing CBMs.

3 Methods

3.1 Background and Preliminaries

Concept Bottleneck Models. A Concept-Bottleneck Model (CBM) can be viewed as a composition of two models, $h = f(g(x)) : \mathcal{X} \to \mathcal{Y}$, with concept en-

coder $g : \mathcal{X} \to \mathcal{C}$, and concept-based classification head $f : \mathcal{C} \to \mathcal{Y}$. $\mathcal{X}, \mathcal{Y}$, where $\mathcal{C}$ denote input, class label, and concept sets, respectively. CBMs get their name from an inherently bifurcated optimization process: While the concept encoder $g(x)$ is trained to predict concepts $\hat{c} \in \mathbb{R}^k$ from the concept set with $\|\mathcal{C}\| = k$ concepts given an image $x \in \mathbb{R}^d$, the classification head $f(\cdot)$ is optimized to predict final target labels $y \in \mathcal{Y} \in \mathbb{R}^M$ solely based on concept assignments produces by g. CBM training data is thus given as $\mathcal{D} := \{x^{(i)}, c^{(i)}, y^{(i)}\}_{i=1}^N$, where $x^{(i)}, c^{(i)}, y^{(i)}$ are the inputs, ground-truth concepts, and ground-truth labels, respectively. Following existing works [9, 27–29], the concept encoder g is trained using a (weighted) binary cross-entropy loss ($\mathcal{L}_{\text{concept}}(\hat{c}, c)$), while the classification head f utilizes a cross-entropy classification objective ($\mathcal{L}_{\text{task}}(\hat{y}, y) = \mathcal{L}_{\text{CE}}(\hat{y}, y)$). Overall, there are three established schemes [9] for training CBMs: (1) *Independent training:* the concept encoder and classification head are trained entirely independently, with ground-truth concepts c provided as inputs to the classification head during training. (2) *Sequential training:* the concept encoder g is trained first, followed by the classification head f trained using the concepts predicted by g. (3) *Joint training:* both the concept encoder g and the classification head f are trained together using a combination of $\mathcal{L}_{\text{concept}}$ and $\mathcal{L}_{\text{task}}$, respectively. In all cases, this means that the classification head leverages only information on concept (co-)occurences to predict final class labels, making it easy to ground the final classification decision on interpretable concept assignments.

Concept Embedding Models. The flow of information in a CBM is bottlenecked by the set of user-defined concepts. This can potentially limit the processing capacity of the model, especially when the concepts do not contain all the information that is needed to perform the downstream task. To overcome this issue, [27] proposed Concept Embedding Models (CEMs) as a generalization of CBMs wherein every concept i is represented by a pair of high-dimensional vectors, $\hat{\mathbf{c}}_i^+$ and $\hat{\mathbf{c}}_i^-$ (as opposed to scalar concepts in CBMs). These embeddings are generated by passing x through concept-specific networks ϕ_i^+ and ϕ_i^-, and represent the concept being present and absent, respectively. The probability $\hat{p}_i$ of the concept i being in x is then simply computed by passing $\hat{\mathbf{c}}_i^+$ and $\hat{\mathbf{c}}_i^-$ to a scoring function s as $\hat{p}_i = s([\hat{\mathbf{c}}_i^+, \hat{\mathbf{c}}_i^-])$. Similarly, both embeddings can also be combined as $\hat{\mathbf{c}}_i = \hat{p}_i \hat{\mathbf{c}}_i^+ + (1 - \hat{p}_i)\hat{\mathbf{c}}_i^-$ to parameterize a joint embedding for concept i. The final concept embedding which represents the full image x and is passed to the classification head is then given as $\hat{\mathbf{c}} := [\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, ..., \hat{\mathbf{c}}_k]$. Notice the higher dimensionality, as we *concatenate* k concept-specific embeddings.

Concept Interventions. Both CBMs and CEMs allow users to intervene on concepts at test time. Concretely, starting from the concept predictions of the model, $\hat{c}$, the user sequentially intervenes on $T \leq k$ concepts. As a human expert has to both investigate concept predictions and compare against input data, interventions are difficult to parallelize, effectively equating concept intervention into a trajectory of T concept intervention steps [20] (see also Fig. 1 for intuition). Let $\mathcal{S}_t$ represent the set of concepts that have been intervened on up to time $t \leq T$. The corresponding concept embedding at time t is then given as $\tilde{c}_t = \{c_{\mathcal{S}_t}, \hat{c}_{\setminus \mathcal{S}_t}\}$, where $c_{\mathcal{S}_t}$ denotes the ground truth values of the intervened concepts,

and $\hat{c}_{\setminus \mathcal{S}_t}$ are the model's predictions of non-intervened concepts. Intervening on concepts in this way updates the final prediction of the model from $\hat{y}$ to $\tilde{y} = f(\tilde{c})$. In the case of a CEM, intervening on concept i to update its value from $\hat{p}_i$ to p_i changes its embedding from $\hat{\mathbf{c}}_i$ to $\tilde{\mathbf{c}}_i = p_i \hat{\mathbf{c}}_i^+ + (1 - p_i)\hat{\mathbf{c}}_i^-$. After each intervention t, we use a concept intervention policy $\pi(\tilde{c}_t)$ to decide which concept to intervene on next. While π can simply suggest random concepts for intervention, it is often much better to leverage heuristics that rank concepts in the order of importance (by some measure). A commonly deployed policy is UCP [10,20], which uses the uncertainty of concepts; selecting concepts with the highest uncertainty (concept predictions closest to 0.5).

Intervention-aware CEMs. While test-time interventions typically improve performance, this is not always guaranteed. In fact, recent works have shown that concept interventions can in some cases even hurt the model's performance [20, 28]. [29] noted that this stems from the lack of training incentive for the model to perform well under intervention. To address this, they proposed Intervention-aware CEMs (IntCEMs), which introduce interventions during the training process to improve the model's receptiveness to interventions at test time, outperforming all existing methods in the intervention setting. In particular, they train a CEM to minimize the following objective:

$$\mathcal{L}_{\text{IntCEM}}(x, c, y, \mathcal{T}) = \mathcal{L}_{\text{pred}}(x, c, y, \hat{c}, \tilde{c}_t) + \lambda_{\text{conc}}\mathcal{L}_{\text{conc}}(\hat{c}, c) + \lambda_{\text{roll}}\mathcal{L}_{\text{roll}}(x, c, y, \mathcal{T}) \tag{1}$$

$$\mathcal{L}_{\text{pred}}(x, c, y, \tilde{c}_0, \kappa_t) = \frac{\text{CE}(f(\hat{c}, y) + \gamma^T \text{CE}(f(\tilde{c}_t), y)}{1 + \gamma^T}$$

$\mathcal{L}_{\text{pred}}$ is the prediction loss for y, $\mathcal{L}_{\text{conc}}$ the concept prediction loss, $\mathcal{L}_{\text{roll}}$ the rollout loss incentivizing the model to predict the most informative concept for intervention. λ_{conc} and λ_{roll} are user-defined weights corresponding to $\mathcal{L}_{\text{conc}}$ and $\mathcal{L}_{\text{pred}}$ respectively, while $\mathcal{T}$ denotes the intervention trajectory. $\mathcal{L}_{\text{pred}}$ penalizes the model for incorrect predictions both before and after the intervention, and $\gamma \geq 1$ is a scaling term that prioritizes correct predictions after intervention.

3.2 Concept Intervention Realignment

Previous works incrementally improve on predecessor methods by better parameterizing concept representations or introducing an intervention-aware training objective. However, all these works still treat concept interventions independently. This means that an intervention on one specific concept has no effect on the assignment of other concepts. This disregards relationships between concepts, which in practice do not occur independently (e.g., "white wing" and "white belly" are more likely to co-occur). As a result, the existing intervention process does not utilize human feedback optimally, as information about the verified existence of one concept should naturally guide the prediction of other concepts. While this aspect is naturally important to ensure that an accurate concept representation is passed to the label classifier, it is also crucial when utilizing concept-selection criteria such as UCP because intervening on one concept

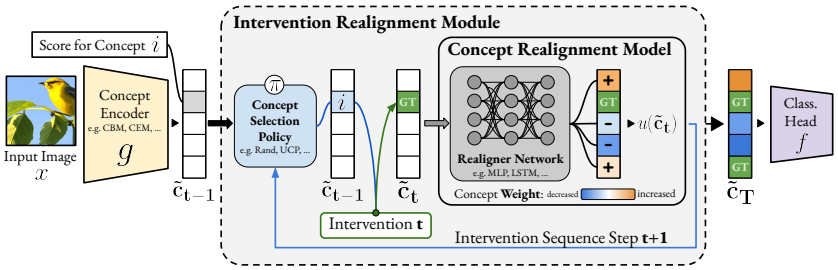


Fig. 2: Illustration of the concept intervention realignment module. Given the concept encoding $g(x)$, we intervene on the concept i selected by a concept selection policy π. This concept is replaced with a ground-truth (GT) value ($\in \{0, 1\}$ depending on whether it is present in a given image or not) to obtain $\tilde{c}_t$ (representing intervention step $t \in \{1, ..., T\}$). This intervened concept representation is then passed into the concept realignment module (leveraging e.g. an MLP or LSTM reweighting mode), which outputs the realigned $u(\tilde{c}_t)$. To ensure that the ground-truth values provided by the user are not overwritten during realignment, $u(\tilde{c}_t)$ retains ground-truth corrections. The final concept vector is then based into a concept-based classifier f.

should consequently reduce the chances of intervening on other closely related, likely co-occurring concepts, while also raising the probability that uncertain and unrelated concepts get intervened on.

Intervention Realignment Module. To address this, we propose a *concept intervention realignment module* (**CIRM**), which consists of two interdependent components: (a) a *concept realignment model* (**CRM**), $u : \mathcal{C} \to \mathcal{C}$. After a user intervenes on a subset of concepts $\mathcal{S}$, the remaining concepts ($\setminus\mathcal{S}$) are updated by a realigner network; and (b) an ***intervention policy*** π. The concepts predicted by the realignment model are fed to the policy to suggest which concept to intervene on next. Both components are interdependent, and together form the overall concept intervention realignment module, as also visualized in Figure 2. The training of the full CIRM comprising both selection policy and concept realignment model aims to simulate the complete intervention process. It thus starts from the concept predictions of the base model, $\hat{c}$, where we sequentially intervene on concepts for $T \leq k$ time steps by following a policy of choice, π (in our case UCP by default, which we experimentally find to outperform random intervention significantly; See supp. §B).

As in §3.1, let $\mathcal{S}_t$ denote the set of intervened concepts and $\tilde{c}_t = \{c_{\mathcal{S}_t}, \hat{c}_{\setminus\mathcal{S}_t}\}$ denote the concepts at time t, respectively. At every intervention time step, we feed $\tilde{c}_t$ to the realignment model to obtain updated concept predictions as $\kappa_t = u(\tilde{c}_t)$, which in turn are utilized by $\pi(\kappa_t)$ to produce intervention recommendations for $t+1$. Finally, we train u with the ground-truth labels as targets using the loss $\mathcal{L}(u) = (\sum_{t=0}^{T} \text{CE}(u(\tilde{c}_t), c))/T$.

Using this simple objective, the concept realignment model u learns to take concept representations and leverage intervened concepts $\mathcal{S}_t$ to predict an updated concept distribution, i.e., $p(i; \hat{c}, \mathcal{S}_t)$. Note that this training objective uti-

lizes standard CBM training information (i.e., concept annotations, [9, 20, 27–29]); so no additional information beyond the standard CBM pipeline is required.

The overall training pipeline can still follow the standard CBM training paradigms (see previous section), with the intervention realignment module being trained independently on top of a pre-trained frozen CBM/CEM as a posthoc realignment method, or jointly with the CBM/CEM to introduce an explicit realignment objective during training. For posthoc realignment, we first train the backbone f and the classification head g. Subsequently, we freeze those components and train the realignment model u.

Realignment Models. As shown in Fig. 2, we parameterize our concept realignment model with a neural network v. To ensure that u does not overwrite the ground-truth concepts provided by the user, we also keep track of the already intervened concepts $\mathcal{S}_t$. Using this information, we replace the output of the realigned concept embedding with the user-provided values for concepts in $\mathcal{S}_t$. Hence, the final output of u for the i^{th} concept is given as

$$u(\tilde{c}_t, \mathcal{S}_t)^{(i)} = v(\tilde{c}_t)^{(i)} \quad \text{if } i \notin \mathcal{S}_t \text{ else} \quad \tilde{c}_t^{(i)} \tag{2}$$

Depending on the assumptions made on the realignment process, v is either a simple MLP or a recurrent model (such as an LSTM [7]). The former parametrizes our default concept intervention realignment model, which only passes the set of intervened and un-intervened concepts at intervention step t to the concept realignment model consisting of a simple MLP. The set of concepts fed into the MLP may either be the original concept embedding $\tilde{c}_0$, where all intervened concepts up to and including step t have been replaced with ground-truth values, or the previously realigned κ_{t-1} with similarly updated intervened concepts (c.f. Fig. 2, "GT"). Note that in either case, κ_{t-1} informs the selection process of the subsequent concept to intervene on. After all interventions, the final concept embedding fed into the classifier is always $u(\tilde{c}_T)$. Practically, we found using $\tilde{c}_t$ to work slightly better than κ_{t-1}. Both cases above however only pass the final set of concepts at time t to the realignment model. Given the sequential nature of interventions, however, it may also be beneficial to account for the entire intervention history to inform future concept realignment. As a result, we also introduce a recurrent realignment variant, u_{rec}, which employs an LSTM model to retain the entire history of interventions until time t. An algorithmic summary is provided in supplementary §A.

End-to-End Realignment. In order to jointly train the CIR module and the base model $f \circ g$, we will perform interventions while also training the base model. This naturally combines with the IntCEM framework [29], which incorporates train-time interventions, and as such is our default choice for joint model and realignment module training.

Concretely, we modify IntCEMs such that after t interventions, concepts $\tilde{c}_t$ are corrected post-intervention to obtain $\kappa_t = u(\tilde{c}_t)$, which is then fed to the

classifier f. The new training objective, then, is:

$$\mathcal{L}_{\text{IntCEM-ReA}}(x, c, y, \mathcal{T}) = \mathcal{L}_{\text{pred}}(x, c, y, \tilde{c}_0, \kappa_t) + \lambda_{\text{conc}} \mathcal{L}_{\text{conc-ReA}}(\hat{c}, c, \kappa_0, \kappa_t) + \lambda_{\text{roll}} \mathcal{L}_{\text{roll}} \tag{3}$$

$$\mathcal{L}_{\text{conc-ReA}}(\hat{c}, c, \kappa_0, \kappa_t) = \frac{1}{2} \left(\mathcal{L}_{\text{conc}}(\hat{c}, c) + \frac{\text{CE}(\kappa_0, c) + \gamma^T \text{CE}(\kappa_T, c)}{1 + \gamma^T} \right) \tag{4}$$

where $\mathcal{L}_{\text{conc-ReA}}$ is the modified concept prediction loss which trains both the backbone g of the base model (first term) as well as the CRM (second term). We use the same γ as in $\mathcal{L}_{\text{pred}}$ to prioritize correct predictions by the CRM after intervention, and the same λ_{conc} and λ_{roll} as in Eq. 1.

4 Experiments

4.1 Preliminaries

Datasets & Implementation. We perform experiments on three datasets: (1) **Caltech-UCSD Birds-200-2011 (CUB)** [22] containing $n = 11,788$ bird images over 200 classes. Following the original CBM paper [9], we use 112 concepts grouped into 28 concept groups with the same splits. (2) **Large-scale CelebFaces Attributes (CelebA)** [11] contains over 200,000 celebrity images annotated with 40 attribute labels, including noisy characteristics such as gender and age. Following [27, 29], we use only the most balanced 8 concepts in our experiments, resulting in $2^8 = 256$ classes. (3) **Animals with Attributes 2 (AwA2)** [23] is a collection of $n = 37,322$ animal images over 50 classes annotated with 85 attributes such as species, color, and behavior. We perform experiments on CEMs, IntCEMs, and three types of CBMs (sequential, independent, and joint). For all models and datasets, we follow the hyperparameters used in [29]. During CIRM training, we sequentially intervene on concepts $T = k$ times. By default, we use UCP both during training and inference, and if not stated otherwise, use a multi-layered perceptron (MLP) for concept realignment. We use the predictions of the base CBM ($\tilde{c}_t$) as its input for un-intervened concept representations. We perform a small, standard hyperparameter using Optuna [1] with 50 trials to search over the number of hidden layers $\in \{1, 2, 3\}$ and units $\in \{k, 2k, k/2\}$, the learning rate $\in [10^{-5}, 10^{-1}]$ and weight decay $\in [10^{-6}, 5 \times 10^{-5}]$, and use the same batch size as used to train the base model. We employ early stopping and learning rate decay on the validation loss. For joint training, we instantiate the realigner MLP 2 hidden layers containing k neurons each. Experiments are conducted using PyTorch [16].

4.2 Concept Realignment Improves Intervention Efficacy

To probe the efficacy of our concept intervention realignment module, we evaluate both the change in concept prediction loss as well as overall classification accuracy as a function of intervened concept counts. These are visualized in Fig. 3

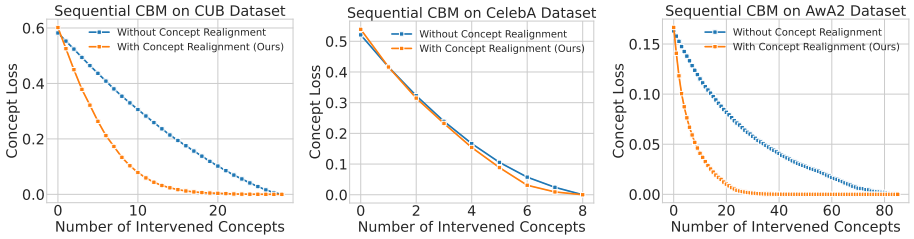


Fig. 3: Concept prediction loss vs. the number of intervened concepts with and without concept realignment. Concept realignment consistently improves concept predictions.

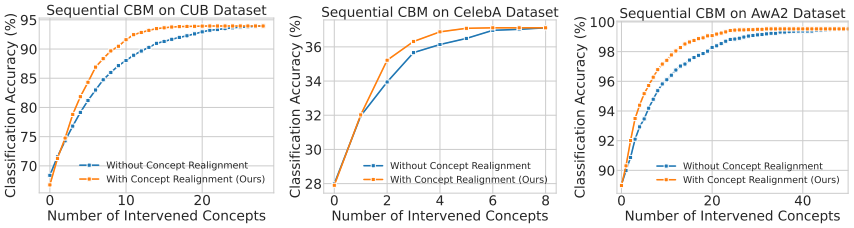


Fig. 4: Classification accuracy vs. the number of intervened concepts with and without concept realignment. Realignment consistently improves classification accuracy.

and Fig. 4 for sequentially trained CBMs (see §3.1), respectively, for all benchmark test sets - CUB, CelebA and AwA2. Note that for AwA2, we only show the first 50 interventions for visual clarity, as performance beyond that heavily plateaus since sufficient concepts have been intervened on to perfectly solve the test data. Table 1 numerically summarizes these results via AUC scores and provides additional scores for independently trained CBMs, jointly trained CBMs as well as Concept Embedding Models (CEMs). Runs in Tab. 1 & Figs. 3, 4 all utilize the stronger UCP selection policy as opposed to the weaker random selection policy (§B) to measure intervention efficacy at the highest level, and train the concept realignment module on top of already trained concept models.

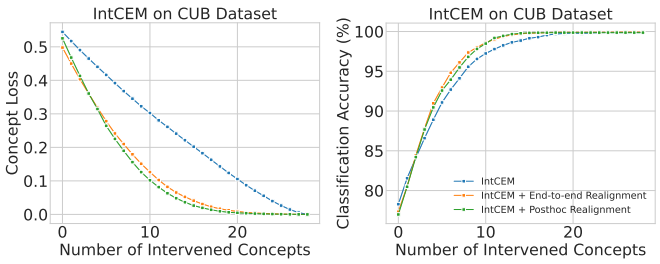
Improved concept attribution through intervention. Across all datasets, we can observe a consistent, in parts vast reduction in concept prediction loss, which measures the correct assignment of concepts for each input (using the concept loss described in §3.1). For example on CUB, a *tenfold* reduction of the original unintervened concept loss (~ 0.6 to ~ 0.06) can be achieved with half the number of interventions (11 with concept realignment, 23 without). This effect becomes even more prevalent on AwA2, where a tenfold reduction ($\sim 0.17 \rightarrow \sim 0.017$) is achieved after around 16 interventions with realignment versus more than 60 without; *marking a more than* 70% *reduction in intervention efforts*. This is also reflected in Tab. 1, where concept loss AUC drops by in parts more than half for CUB and from 4.26 to 1.13 on AwA2. We find this significant improvement in concept attribution persists across all CBMs and CEMs, as well as random seed initializations (see Supp. Tab. 2 and 3)

Table 1: Area Under Curve (AUC) of Concept Prediction Loss and Classification Accuracy with/without CIRM (non-averaged sum). We use the same backbone throughout. CIRM improves performance across all models and datasets. Intervention curves share long saturation plateaus for high intervention counts. Accuracy AUC scores are thus saturated, and best combined with performance graphs in Figs. 3, 4.

Base Model	Realigned	Concept Loss AUC ↓			Accuracy AUC ↑		
		CUB	CelebA	AwA2	CUB	CelebA	AwA2
Sequential CBM	×	6.71	1.59	4.26	2460.8	280.7	8364.0
	✓	**3.15**	**1.52**	**1.13**	**2510.9**	**284.3**	**8397.6**
Independent CBM	×	6.71	1.59	4.26	2653.4	280.2	8403.4
	✓	**3.15**	**1.52**	**1.13**	**2678.3**	**282.1**	**8437.0**
Joint CBM	×	5.93	3.06	4.77	2580.3	273.1	8276.4
	✓	**3.67**	**1.76**	**1.48**	**2609.0**	**273.9**	**8327.4**
CEM	×	5.99	1.61	4.90	2521.4	396.3	8429.3
	✓	**3.20**	**1.46**	**1.69**	**2558.4**	**400.1**	**8433.9**

We do find that for CelebA with a much more restrictive concept bottleneck than e.g. CUB and AwA2, due to significantly fewer (note that in CUB concepts are already grouped, see §4.1) and noisier concepts, that the overall gain in concept accuracy is smaller. This is also reflected in the notably weaker performance of the base CBM (c.f. Fig. 4, middle - less than 38% accuracy when intervening on *all* concepts), which strongly points towards overall insufficient concept information provided in the CelebA training data. Overall, however, we find very clear evidence that the concept intervention realignment module allows practitioners to leverage human intervention feedback to a much larger extent to attribute the correct concepts to respective inputs. This means that the subsequent classifier will operate on a much more accurate set of concepts, thereby improving the overall interpretability of the final classification decision.

Improved overall classification through intervention. On top of that, we also find that the significant gain in intervention efficacy on a concept attribution level also translates to subsequent gains in intervention efficacy for the overall classification performance (Fig. 4). For example on CUB, the final classification accuracy after intervening on all concepts is 93.9%, which is achieved already after ~ 16 intervention steps. A comparable performance without concept intervention realignment requires nearly complete, ~ 24 intervention steps, marking a 50% increase. The same can be seen on CelebA and AwA2 as well, where the upper-bound performance can be achieved with much fewer interventions (particularly without the need to intervene on *all* concepts). Even intermediate performance targets are achieved much earlier; a classification accuracy target of e.g. 98% on AwA2 requires only 12 concept interventions with realignment, while the non-aligned baseline needs 19 interventions on average. We find these results to be also reflected numerically in Tab 1, where accuracy AUC increases from e.g. 2460.8 to 2510.9 on CUB. We do point towards high numerical saturation given the larger performance plateaus at higher intervention counts, and high starting accuracies (e.g. $\sim 90\%$ on AwA2). Numerical results are thus best considered alongside the intervention trajectories in Figs. 3 and 4.



(a) **(b)**

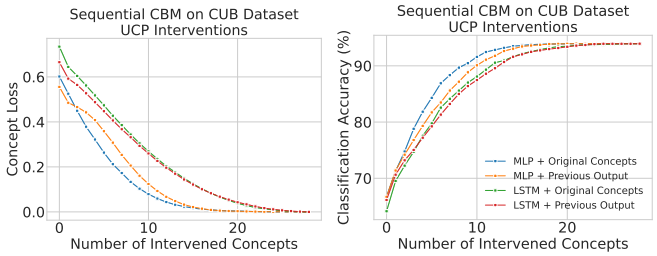
Fig. 5: Concept Intervention Realignment in intervention-aware CEMs. (a) Concept prediction loss and (b) classification accuracy with jointly and post-hoc trained CIRMs. In both cases, significant benefits can be seen, especially for correct concept attribution after intervention - both for jointly and posthoc trained realignment modules.

Together, our experiments provide strong evidence that concept intervention realignment is crucial to best leverage human feedback in concept-based decision systems; allowing to significantly reduce intervention budgets by in parts over 70% to achieve a desired target performance. These gains can also be achieved *after* concept models have been trained, allowing for versatile applicability.

4.3 Intervention Realignment for Intervention-aware CEMs

In this section, we investigate training the CIRM during the training process of an already intervention-regularized concept model; namely the recently proposed, state-of-the-art intervention-aware CEM [29] (see also §3.1). Following the objective described in Eq. 3, we operate and train the concept intervention module in conjunction with the intervention objective proposed by [29].

Our results are shown in Fig. 5. First, we find that explicit concept intervention realignment can significantly improve correct concept attribution, even in intervention-aware training setups (c.f. Fig. 5a). While not as significant as improvements over standard CBM models, for specific target concept prediction losses (such as a *fivefold* reduction from 0.5 to 0.1), half the number of intervention steps are needed (11 versus 20). The improved concept attribution is also reflected in higher intervention accuracies as seen in Fig. 5b, albeit the overall (still notable!) improvement is less reflective of the significant gains on a concept level (additional results can be found in supp. §C). Overall, however, our experiments highlight that even when applied to state-of-the-art approaches that specifically simulate the intervention process during training, improved intervention efficacy can be found. Importantly, the consistently significant improvements on a concept attribution level mean that classification decisions are much better grounded on correct concept attributions, which is crucial for interpretability [9,27] of classification results. Finally, we find that concept intervention realignment can be applied both as a regularization mechanism during training, as well as adapted entirely posthoc, while still offering consistent benefits. This supports the high versatility of CIRMs as a general-purpose tool to increase intervention efficacy.



(a) **(b)**

Fig. 6: (a) Concept prediction loss and (b) classification accuracy for various realigner architectures alongside UCP policy. Using an MLP with concept predictions of the base model works better than compounding refinements and accounting for intervention trajectories using LSTMs.

4.4 Realignment Module Ablations

Realignment Model Architectures. In this section, we study the effect of various design choices for the realignment module along two dimensions: **(1) Recurrent vs. Feedforward Networks:** Since we intervene on concepts sequentially, it is possible that the realignment module can benefit from the overall order and history of interventions to make more accurate concept predictions. To do this, we instantiate the concept realignment network using an LSTM [7]. We compare this against our default MLP. **(2) Previous Output vs. Original Concepts:** By default, the realignment module takes as input a combination of ground-truth concepts provided by the user and values predicted by the base model at $t = 0$ for the concepts that have not been intervened on (see also §3.2). Due to the sequential nature of interventions, one may also directly feed the output of the realignment module at time $t - 1$ as input to it at time t in order to compound the refinements over multiple time steps. Combining both axes results in four recombinations, which we compare in Fig. 6. As can be seen, there is limited gain when accounting for the complete intervention history using an LSTM realigner network. Similarly, we find that applying the MLP primarily for concept selection alongside UCP and as final input to the classification head works better than compounding refinements over intervention steps.

Intervention Policy Transfer. In this section, we study the importance of aligning intervention policies used during training with those deployed at test time. In particular, we operate on the base setup, which deploys the CBM and the concept intervention realignment module using only the much weaker random intervention policy at test time. However, we change the policy used to train the concept intervention realignment module. Our results are visualized in Fig. 7. As can be seen, while a realignment module trained with UCP can still be effective when deployed with a random intervention policy, it is notably outperformed by the weaker random policy at test-time when the realignment module has been trained on the same random policy as well. This means that the realignment module adapts to the selection policy used during training. Thus to get the

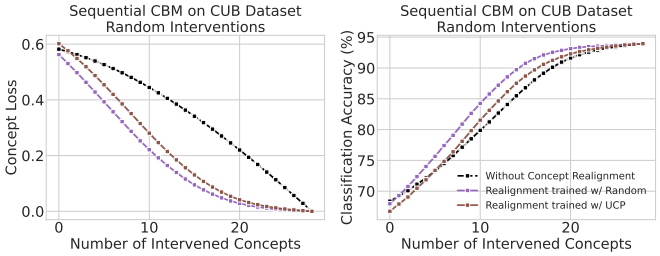


Fig. 7: Concept prediction loss and classification accuracy under random interventions for realignment modules trained with random and UCP policy, respectively. Results indicate that alignment of policy used during training and deployment is important.

most benefits out of concept intervention realignment, selection policies should align during training and deployment.

Alignment b/w Realignment Module Components. Finally, we study how important the alignment between the concept realignment model and intervention policy (i.e., UCP) is to form the overall concept intervention realigment module. To accomplish this, we employ two module variations: (a) an *original policy* denoted as $\pi(\hat{c}_0)$, which only applies the UCP criterion to the original concept predictions generated by the base model without any concept realignment (i.e., the policy does not change over time), and (b) our default setup (*updated policy*), which informs the intervention policy using realigned concept values ($\pi(\kappa_t)$). Note that in both cases, the classification head still receives realigned concept embeddings, as we only want to study the importance of alignment between the concept realignment model and the intervention policy. Results in Fig. 8 clearly reveal that while simple realignment on its own can already help improve intervention efficacy, much larger efficacy gains are unlocked when both policy and the realignment model are utilized in conjunction.

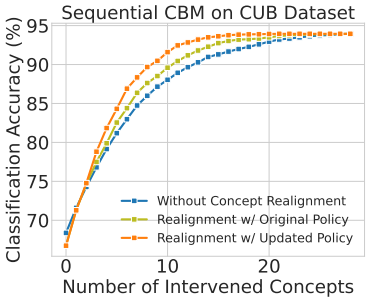


Fig. 8: Classification accuracy vs concept interv. counts, showing our updated selection policies improving over the static one.

A Closer Look at Concept Realignment. To understand the impact of the realignment process qualitatively, we also provide examples in Fig. 9. In this figure, we showcase the impact of interventions on the top 10 concepts with the highest prediction errors, and the specific number of interventions required to predict the correct label. For both examples, we find that intervention on a single concept is insufficient to flip incorrect class predictions. However, as we intervene on more concepts, we can clearly see that concept realignment jointly allows concept prediction error - even on the initially worst predicted concepts - to be significantly reduced, while also reaching correct image classification with

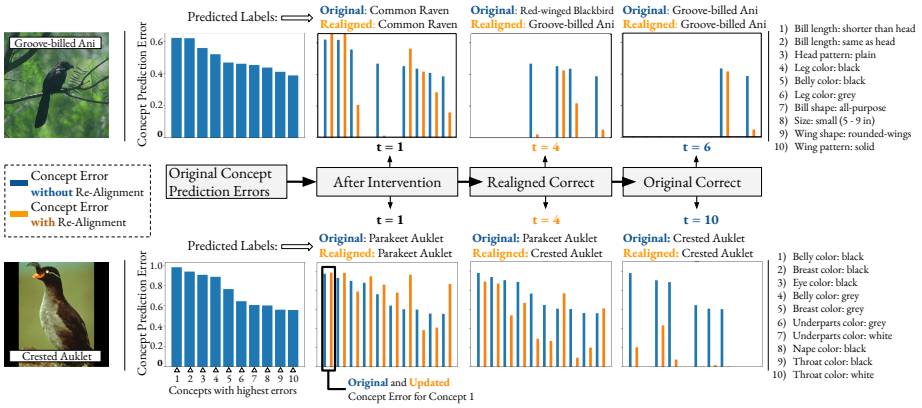


Fig. 9: Examples for the improved intervention efficiency of CIRM. We show the change in concept prediction errors of the ten worst predicted concepts, as a function of concept intervention steps t: concept realignment allows concept error even for strongly mispredicted concepts to be significantly reduced with interventions, achieving correct label classification after much fewer interventions compared to non-realigned baselines.

in parts less than half the number of interventions (for *Crested Auklet*). These results conceptually support the quantitative benefits of concept realignment seen in previous benchmark experiments.

5 Conclusion

In this work, we identify the independent treatment of concepts during test-time interventions in CBMs as a cause for reduced intervention efficacy. To remedy this problem, we propose a concept intervention realignment module - a simple and lightweight technique to automatically update concept assignments after human intervention on one or multiple concepts. Our experiments demonstrate significant gains in concept attribution as well as overall classification accuracy of concept-based models under intervention. We show that our approach is versatile and can be applied to a wide range of concept-based models, intervention policies, and training schemes. We believe that the reduction in required human interventions to reach performance targets facilitates the practical deployment of concept-based models even in resource-constrained environments.

Acknowledgements

Karsten Roth and Jae Myung Kim thank the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program and the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for support. This work was supported by DFG project number 276693517, by BMBF FKZ: 01IS18039A, by the ERC (853489 - DEXIM), by EXC number 2064/1 – project number 390727645. The authors would like to thank Shyamgopal Karthik and Leander Gerrbach for their helpful feedback on the manuscript.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2623–2631 (2019) 8
2. Alvarez Melis, D., Jaakkola, T.: Towards robust interpretability with self-explaining neural networks. Advances in neural information processing systems **31** (2018) 3
3. Chauhan, K., Tiwari, R., Freyberg, J., Shenoy, P., Dvijotham, K.: Interactive concept bottleneck models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 5948–5955 (2023) 3
4. Durán, J.M., Jongsma, K.R.: Who is afraid of black box algorithms? on the epistemological and ethical basis of trust in medical ai. Journal of Medical Ethics **47**(5), 329–335 (2021). https://doi.org/10.1136/medethics-2020-106820, https://jme.bmj.com/content/47/5/329 1
5. EUGDPR: Gdpr. general data protection regulation (2017) 1
6. Havasi, M., Parbhoo, S., Doshi-Velez, F.: Addressing leakage in concept bottleneck models. Advances in Neural Information Processing Systems **35**, 23386–23397 (2022) 3
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997) 7, 12
8. Kim, E., Jung, D., Park, S., Kim, S., Yoon, S.: Probabilistic concept bottleneck models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 16521–16540. PMLR (23–29 Jul 2023), https://proceedings.mlr.press/v202/kim23g.html 3
9. Koh, P.W., Nguyen, T., Tang, Y.S., Mussmann, S., Pierson, E., Kim, B., Liang, P.: Concept bottleneck models. In: International conference on machine learning. pp. 5338–5348. PMLR (2020) 1, 2, 3, 4, 7, 8, 11
10. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Machine learning proceedings 1994, pp. 148–156. Elsevier (1994) 5
11. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015) 3, 8
12. Mahinpei, A., Clark, J., Lage, I., Doshi-Velez, F., Pan, W.: Promises and pitfalls of black-box concept learning models. arXiv preprint arXiv:2106.13314 (2021) 3
13. Marconato, E., Passerini, A., Teso, S.: Glancenets: Interpretable, leak-proof concept-based models. Advances in Neural Information Processing Systems **35**, 21212–21227 (2022) 3
14. Margeloiu, A., Ashman, M., Bhatt, U., Chen, Y., Jamnik, M., Weller, A.: Do concept bottleneck models learn as intended? arXiv preprint arXiv:2105.04289 (2021) 3
15. Oikarinen, T., Das, S., Nguyen, L.M., Weng, T.W.: Label-free concept bottleneck models. In: The Eleventh International Conference on Learning Representations (2022) 3
16. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances

in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf 8
17. Piano, S.L.: Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward. Palgrave Communications **7**(1), 1–7 (2020), https://EconPapers.repec.org/RePEc:pal:palcom:v:7:y:2020:i:1:d:10.1057_s41599-020-0501-9 1
18. Sawada, Y., Nakamura, K.: Concept bottleneck model with additional unsupervised concepts. IEEE Access **10**, 41758–41765 (2022) 3
19. Sheth, I., Rahman, A.A., Sevyeri, L.R., Havaei, M., Kahou, S.E.: Learning from uncertain concepts via test time interventions. In: Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022 (2022) 3
20. Shin, S., Jo, Y., Ahn, S., Lee, N.: A closer look at the intervention procedure of concept bottleneck models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 31504–31520. PMLR (23–29 Jul 2023), https://proceedings.mlr.press/v202/shin23a.html 2, 3, 4, 5, 7
21. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr (2018) 1
22. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011) 2, 3, 8
23. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: 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 (2018) 3, 8
24. Xu, X., Qin, Y., Mi, L., Wang, H., Li, X.: Energy-based concept bottleneck models. In: The Twelfth International Conference on Learning Representations (2023) 3
25. Yang, Y., Panagopoulou, A., Zhou, S., Jin, D., Callison-Burch, C., Yatskar, M.: Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19187–19197 (2023) 3
26. Yuksekgonul, M., Wang, M., Zou, J.: Post-hoc concept bottleneck models. In: The Eleventh International Conference on Learning Representations (2022) 3
27. Zarlenga, M.E., Barbiero, P., Ciravegna, G., Marra, G., Giannini, F., Diligenti, M., Shams, Z., Precioso, F., Melacci, S., Weller, A., et al.: Concept embedding models. arXiv preprint arXiv:2209.09056 (2022) 1, 2, 3, 4, 7, 8, 11
28. Zarlenga, M.E., Barbiero, P., Shams, Z., Kazhdan, D., Bhatt, U., Weller, A., Jamnik, M.: Towards robust metrics for concept representation evaluation. arXiv preprint arXiv:2301.10367 (2023) 4, 5, 7
29. Zarlenga, M.E., Collins, K.M., Dvijotham, K.D., Weller, A., Shams, Z., Jamnik, M.: Learning to receive help: Intervention-aware concept embedding models. In: Thirty-seventh Conference on Neural Information Processing Systems (2023) 1, 2, 3, 4, 5, 7, 8, 11