Causal Intervention in Concept Bottleneck Models

Zhiyu Zhu[1], Jiayu Zhang[2], Zhibo Jin[1], Xinyi Wang[3], Fang Chen[1], Przemyslaw Biecek[4], and Jianlong Zhou[1⋆]

[1] University of Technology Sydney, Sydney, Australia
[2] Suzhou University of Technology, Suzhou, China
[3] Universiti Malaya, Kuala Lumpur, Malaysia
[4] Warsaw University of Technology, Warsaw, Poland

Abstract. Concept Bottleneck Models (CBMs) are a class of interpretable AI models that enable concept-level control over the decision-making process, allowing humans to directly intervene in model predictions. As a result, CBMs have been widely adopted in tasks requiring interpretability and controllability. Existing research has demonstrated that human-model interaction can significantly enhance CBM performance. However, in practical applications, each intervention requires manual human effort and explicit data provision. Therefore, a central challenge is how to improve model performance while minimizing intervention effort. In this paper, we show that trained CBM parameters encode usable concept-correlation structure. We leverage this structure during interaction by propagating user feedback through the shared input-to-concept pathway, thereby adjusting non-intervened concepts without training an additional realignment model. Experimental results demonstrate that our method improves low-intervention performance across multiple models and datasets. Compared with all tested baselines, our approach improves early-intervention performance by approximately 9%; compared with Concept Realignment, it improves AUC@10 and AUC@20 by 3.90% and 2.88%, respectively, while avoiding extra training. Code is available at `https://github.com/LMBTough/CI`.

Keywords: Concept Bottleneck Models · Causal Intervention · Explainable AI

1 Introduction

As deep learning models are increasingly deployed in high-stakes decision-making scenarios (e.g., medical diagnosis [1,8,12], autonomous driving [4], and financial risk control [2,5]), the demand for model interpretability has grown significantly. Concept Bottleneck Models (CBMs) [9] have shown notable advantages in this regard: by explicitly introducing a concept layer into the network architecture, each concept in this layer is imbued with a clear semantic meaning. Unlike

⋆ Corresponding author: `jianlong.zhou@uts.edu.au`.

traditional "black-box" neural networks, CBMs can output concept vectors that are interpretable to humans during inference, and these concept vectors are subsequently processed by a more transparent classifier (e.g., a linear layer) to yield final predictions. As a result, CBMs not only offer interpretable decision support in high-stakes contexts but also provide a straightforward interface for human-in-the-loop interventions.

Within this conceptual representation framework, CBM intervention has emerged as a vital area of research: when the model exhibits high uncertainty regarding certain concepts, human experts can directly supply the ground-truth values for these concepts, thereby significantly enhancing the final predictive accuracy. For instance, some studies (e.g., [6, 16]) measure uncertainty based on how far concept predictions deviate from 0.5. If a predicted concept value is close to 0.5, the model's confidence in that concept is low, and precise human annotation is more urgently needed. However, each interaction comes with additional manual cost, which is often impractical to incur frequently in real-world scenarios. How to maximize the model's performance with fewer interactions is therefore a key challenge in this field.

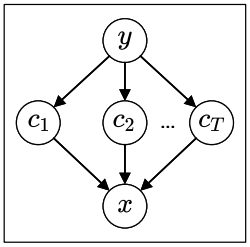


Fig. 1: Illustration of the structural dependency in CBMs. Each concept $c_1, c_2, \ldots, c_T$ is generated from the input x through an encoder network f, while the final prediction y is made from the concept vector. This computational graph motivates using x as a shared upstream carrier for propagating user feedback, without claiming causal identification among real-world semantic concepts.

Early-stage research on concept intervention typically treated each concept independently, ignoring the dependencies among concepts. Concept Realignment (CR) [17] introduced a two-stage approach for the first time: in the first stage, an intervention policy provides the indices of concepts requiring intervention, along with their corresponding ground-truth values supplied by humans; in the second stage, a Concept Realignment operation is applied to further "realign" the concept vectors that have already been manually edited. Specifically, CR needs to train an additional neural network that takes the human-corrected concept vectors as input and outputs adjusted concept vectors. While this method explicitly leverages the correlations among concepts, it also adds complexity by introducing a new network architecture and training procedure, which in-

creases uncertainties related to model initialization, hyperparameter tuning, and convergence. Moreover, this approach only utilizes a small portion of parameters in the concept layer and subsequent modules, overlooking a large body of concept-related information learned by the rest of the CBM. As we will show, these untapped parameters actually capture useful information about correlations among concepts, which can be directly leveraged for realignment without training a separate adjustment model.

Additionally, as illustrated in Figure 1, concept vectors are obtained by mapping the original input x through a concept encoder. Hence, the relationships among concepts may also be embedded within the parameters throughout the network. Focusing solely on the concept layer risks ignoring the distributed representations of these relationships learned by earlier layers. Building on this insight, as shown in Figure 2, we propose a training-free approach called Causal Intervention (CI) to realign concept values in a trained CBM by using x as a shared upstream intervention carrier. Our main idea is that trained CBM parameters encode usable correlations among concepts; therefore, there is no need to train a separate network during the intervention phase. Instead, we use the learned $x \to c$ pathway to adjust the concept vector. This reduces uncertainty introduced by training an additional model and more fully utilizes the concept-level relationships learned by the entire CBM, resulting in improved low-intervention performance. Our main contributions can be summarized as follows:

1. We show that trained CBM parameters encode usable concept-correlation information, enabling concept realignment during intervention without an extra model.
2. We propose Causal Intervention, a training-free method that uses x as a shared upstream carrier to propagate user feedback through the learned $x \to c$ pathway.
3. Through experiments on multiple datasets, we demonstrate a better trade-off between intervention cost and predictive accuracy, especially in low-intervention regimes. Code is available at `https://github.com/LMBTough/CI`.

2 Related Works

2.1 Concept Bottleneck Models (CBMs)

The Concept Bottleneck Model (CBM) [9] is a concept-based interpretable architecture that inserts a concept layer between the input and the final prediction. The encoder first maps an input x to a high-level concept vector, whose dimensions correspond to human-understandable concepts, and a transparent module (e.g., a linear classifier) then maps this vector to the label. This design allows users to inspect concept activations and their contributions to the final decision, yielding human-interpretable explanations.

Because each concept in a standard CBM is represented by a scalar in $[0, 1]$, its expressive capacity can be limited. Concept Embedding Models (CEM) [6]

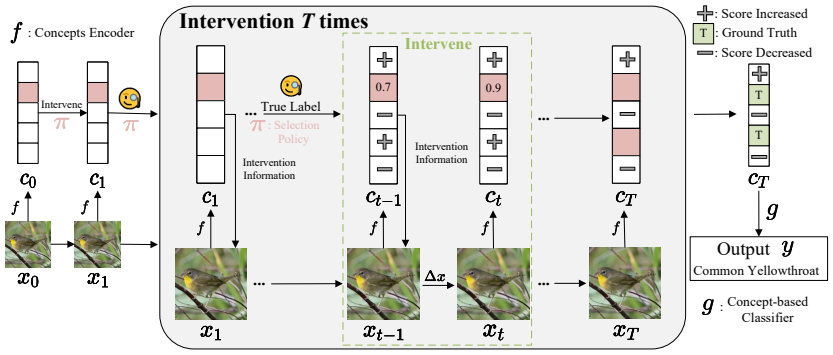


Fig. 2: Overview of the iterative concept-based intervention process. At each round t, the current image x_{t-1} is encoded by f to produce a concept vector c_{t-1}. A policy π selects concept(s) for intervention, and the image is updated to x_t with a small perturbation Δx that moves the selected concepts toward their ground-truth values. After T rounds, the final concept vector c_T is passed to a concept-based classifier g to obtain prediction y. A plus (+) (resp. minus, −) sign indicates an increase (resp. decrease) in the predicted concept score.

address this by assigning two embeddings to each concept, corresponding to its presence or absence, and learning a probability $p \in [0, 1]$ to combine them. The final representation concatenates the embeddings for all concepts. Although CEM often improves accuracy over vanilla CBMs, it introduces more complex concept representations and a larger latent space.

A key advantage of CBMs is that the concept layer supports human-in-the-loop interaction [7]. At inference time, users can overwrite predicted concept values with ground-truth ones to improve predictions [11]. However, each intervention requires additional annotations, so it is important to maximize performance gains from as few interventions as possible.

2.2 CBM Interventions

CBM interventions typically consist of two steps: **(1)** an intervention policy decides which concept(s) should query human feedback; **(2)** the concept vector is adjusted after intervention. Most existing work focuses on step (1), treating concepts independently and omitting explicit realignment in step (2), which limits the ability to exploit correlations between concepts.

Uncertainty of Concept Predictions (UCP) [6] prioritizes concepts whose predicted values are closest to 0.5, i.e., the most uncertain in $[0, 1]$. The importance of concept i is measured by $\frac{1}{|\hat{c}_i - 0.5|}$ (with a small constant added in practice to avoid division by zero).

Contribution of Concept on Target Prediction (CCTP) [16] uses gradient-based attribution [14] on the concept layer. It defines the importance of concept i as $\sum_{j=1}^{M} \left| \hat{c}_i \frac{\partial g_j}{\partial \hat{c}_i} \right|$, where g_j is the j-th logit from the $C \to Y$ module.

Expected Change in Target Prediction (ECTP) [16] measures how much the prediction distribution changes if concept i is clamped to 0 or 1: $(1 - \hat{c}_i)\, D_{\mathrm{KL}}\left(\hat{y}_{\hat{c}_i=0} \| \hat{y}\right) + \hat{c}_i D_{\mathrm{KL}}\left(\hat{y}_{\hat{c}_i=1} \| \hat{y}\right)$, where D_{KL} is the Kullback–Leibler divergence and $\hat{y}_{\hat{c}_i=0}$ (resp. $\hat{y}_{\hat{c}_i=1}$) is the prediction when $\hat{c}_i$ is set to 0 (resp. 1).

Expected Uncertainty Decrease in Target Prediction (EUDTP) [15] quantifies the expected reduction in predictive entropy after intervening on concept i: $(1 - \hat{c}_i)\, H\left(\hat{y}_{\hat{c}_i=0}\right) + \hat{c}_i H\left(\hat{y}_{\hat{c}_i=1}\right) - H(\hat{y})$, where H denotes entropy.

Cooperative Prediction (COOP) [3] combines concept prediction uncertainty with its influence on the final output, producing a joint importance score.

Intervention-aware CEM (Int-CEM) [7] trains a neural network to predict which concept should be intervened on, using concept labels during training, and integrates this network into CBM training so that the model becomes more sensitive to potential interventions.

Step (2), explicit concept realignment, was first emphasized by [17], which observed that concept activations can be strongly correlated. When some concept values are manually corrected, remaining concepts may need to be adjusted consistently. Concept Realignment (CR) [17] trains an additional neural network to realign concept values after intervention, but this extra model introduces additional uncertainty and operates only on the concept layer, ignoring information from other parts of the model and limiting its effectiveness.

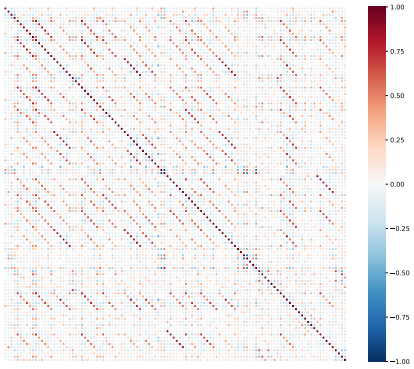
3 Method

3.1 Problem Definition

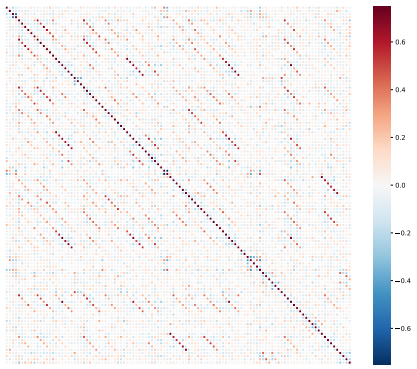
In our setting, each input instance is represented as a sample $x \in \mathbb{R}^n$, and y is a ground-truth label, and we are given a set of ground-truth concepts $c \in \mathbb{R}^k$. A neural network $f : X \to C$ maps the input x to the concept space, and outputs predicted concepts $\hat{c} \in \mathbb{R}^k$. Formally, $\hat{c} = f(x)$. Additionally, a neural network $g : C \to Y$ maps these concepts to a predicted label $\hat{y}$.

We introduce an *intervention policy* $\pi(f, g, x, t, s_t)$ that iteratively refines the concept predictions by selecting and returning a concept index i at each step. Here, $t \in \mathbb{N}$ is the *round index* denoting the current intervention step, and s_t is the *set of concept indices* selected for intervention at round t, where $s_t = \emptyset$ and $s_t = s_{t-1} \cup \{i\}$.

In scenarios where multiple concepts may be intervened upon at once (i.e., *group intervention*), each round returns a set of concepts g_t, such that $s_t = s_{t-1} \cup g_t$. The aim is to progressively improve the concept predictions, and thus the final classification output, by judiciously selecting which concepts to intervene on at each round.



(a) Correlation matrix (M_1) of ground-truth concept annotations.



(b) Correlation matrix (M_2) of learned CBM weights.

Fig. 3: Learned correlations in Concept Bottleneck Models (CBMs) on the Caltech-UCSD Birds-200-2011 (CUB) dataset. (a) Shows the correlation structure of ground-truth concept annotations, while (b) depicts the correlation structure learned by CBMs through the fully connected layer. Although some correlations are diluted in (b), the relative relationships remain highly consistent, suggesting that CBMs capture concept correlations during training. This also highlights that even when the model cannot learn the full correlation structure due to capacity or optimization limitations, it still preserves the most salient correlations among concepts.

3.2 Causal Intervention

Prior to the work of [17], methods such as [6,15,16] treated concept interventions independently, aiming solely to find an optimal concept-selection policy π while overlooking interactions among concepts. As a simple example, consider a bird-classification task with two concepts: "the bird has a gray abdomen" and "the bird has gray feathers." These two concepts are clearly correlated; if one concept is known to be true, the probability of the other being true increases accordingly.

To aid intuitive understanding, this phenomenon can be analogized to a common educational example: suppose E denotes a student's effort, P denotes exam performance, and Q denotes the quality of homework. While there is no direct causal relationship between P and Q, they often exhibit strong statistical correlation in observational data due to both being influenced by the common cause E. That is, students who score high on exams also tend to do homework well—not because one causes the other, but because both are caused by higher effort. When we condition on E, this correlation between P and Q disappears, i.e., $P \perp\!\!\!\perp Q \mid E$.

Concept Realignment (CR) [17] was recently introduced to account for such correlations, training an additional neural network to predict how remaining concepts should be adjusted once certain concept values have been corrected. However, CR only takes the k-dimensional concept vector as both input and output, relying exclusively on the concept layer. In other words, CR's training ignores the information learned by the rest of the CBM, since the concept adjustment

process does not utilize any information from the $X \to C$ pathway, which can be substantial. Moreover, the extra neural network in CR introduces additional uncertainty related to model architecture, parameter initialization, and training strategy. This uncertainty arises because the training of a separate model depends heavily on random initialization, stochastic optimization dynamics, and the specific training samples used—leading to potential instability or variance in behavior across different runs. Motivated by these issues, we propose a **Causal Intervention** method that utilizes existing inter-concept relationships learned by CBMs themselves, without adding new structures. Unlike CR, our method does not require training an additional neural network to perform concept adjustment, resulting in significantly improved computational efficiency and reduced implementation complexity. Interpretability is directly inherited from the underlying CBM: we only introduce a causal interaction procedure at inference time without modifying the CBM itself, so its interpretability is preserved.

Learned Correlations in CBMs. Before describing Causal Intervention, we examine whether trained CBM parameters encode a usable concept-correlation structure. We use two data matrices: one that is entirely independent of CBMs' training process and another that is related to training but cannot be fully derived from the first matrix (i.e., they should not be directly transformable into each other). The first matrix $M_1 \in \mathbb{R}^{n \times k}$ consists of ground-truth concept annotations c for all n training samples. The second matrix $M_2 \in \mathbb{R}^{m \times k}$ is derived from the linear layer that maps the output of a pretrained backbone [6, 9] (of dimensionality m) to the concept space. We extract the trained weight parameters of this linear layer to form M_2. Using Pearson correlation coefficients [13], we compare how correlations among the K concepts are reflected in M_1 versus M_2. As illustrated in Figure 3a and Figure 3b, although the learned weights in M_2 dilute some correlations, the relative relationships remain highly consistent. On CUB-CBM, learned weight correlation aligns strongly with ground-truth concept correlation (signed Spearman 0.849, absolute Spearman 0.635, top-50 overlap 0.86), while a random control is near zero. This observation inspires our Causal Intervention approach, which leverages the CBM's internal parameters to realign concept values after intervention.

Causal Reasoning. Figure 1 shows the CBM computational graph, where each concept is obtained from x via network f. Because there are no lateral links among concepts in this graph, dependencies among concept predictions are mediated by the shared upstream representation. CR [17] uses the concept vector directly, which is partly reasonable when the real concept label c is available as supervision. Indeed, c is also the primary supervision for CR.

However, as discussed in Section 3.2, concepts often exhibit statistical correlation due to a shared upstream carrier (x), while lacking direct influence on each other in the CBM graph. This is analogous to the exam example: exam performance and homework quality are correlated because both are caused by effort, not because one causes the other. Similarly, in CBMs, using x as the in-

tervention carrier propagates feedback through the trained $x \to c$ pathway and avoids architecture-specific edits to an intermediate layer.

Scope of the Causal Claim. We do not claim causal identifiability in the strict Structural Causal Model sense. Instead, CI treats the trained CBM as the object of intervention: the user supplies ground-truth feedback for queried concepts, and the update propagates this feedback through the shared $x \to c$ pathway. This distinction is important because the method exploits dependencies encoded by the model parameters, rather than asserting that semantic concepts have been causally identified from observational data.

Causal Intervention Mechanism. In standard interventions, queried concept dimensions are clamped to the ground-truth values provided by a human before classification. This semantics is retained in CI. Our additional goal is to propagate the feedback to non-queried concepts through the trained encoder f. Since f has learned correlations among concepts and x is their shared upstream carrier, we adjust x to indirectly correct a target concept's value and allow multiple interventions to accumulate through updates to x.

Suppose an intervention policy π selects concept i for intervention, with predicted value $\hat{c}_i(x)$ and ground-truth value c_i. We want to modify x so that the updated $\hat{c}_i(x)$ is closer to c_i. We apply a first-order Taylor expansion at x:

$$\hat{c}_i(x + \Delta x) \approx \hat{c}_i(x) + \Delta x^\top \nabla_x \hat{c}_i(x) + \mathcal{O}(\|\Delta x\|^2), \tag{1}$$

where $\mathcal{O}(\|\Delta x\|^2)$ represents higher-order terms. To move $\hat{c}_i(x + \Delta x)$ closer to c_i, we set Δx according to the sign of the local gradient. In neural networks, the first-order Taylor expansion is a standard local approximation underlying gradient-based optimization, which we further exploit in the following analysis.

$$\Delta x = \begin{cases} \eta \cdot \operatorname{sign}\Big(\frac{\partial \hat{c}_i(x)}{\partial x}\Big) & \text{if } \hat{c}_i < c_i, \\ -\eta \cdot \operatorname{sign}\Big(\frac{\partial \hat{c}_i(x)}{\partial x}\Big) & \text{if } \hat{c}_i > c_i, \end{cases} \tag{2}$$

where η is the step size, and $\operatorname{sign}(\cdot)$ ensures that all dimensions of x are updated consistently to avoid large deviations from the original sample. As a result, after T rounds, $\|x_T - x\|_\infty \leq T \cdot \eta$, which typically translates to modifying only a few pixels for images. A formal proof of efficacy appears in the supplementary material.

We can then consider how this update strategy affects other concepts $j \neq i$:

$$\hat{c}_j(x + \Delta x) = \hat{c}_j(x) \pm \eta \cdot \operatorname{sign}\Big(\frac{\partial \hat{c}_i(x)}{\partial x}\Big) \cdot \frac{\partial \hat{c}_j(x)}{\partial x}. \tag{3}$$

Here, the term $\operatorname{sign}\Big(\frac{\partial \hat{c}_i(x)}{\partial x}\Big) \cdot \frac{\partial \hat{c}_j(x)}{\partial x}$ captures the linkage between concepts i and j through x.

A more unified way to represent this uses a loss function $L(\hat{c}_i, c_i)$, such as binary cross-entropy (BCE) [10]. We let $\Delta x = -\eta \cdot \operatorname{sign}\big(\nabla_x L(\hat{c}_i(x), c_i)\big)$, which

Algorithm 1 Causal Intervention (CI)

Input: Encoder f, Classifier g, Policy π, Steps T, Step size η, Input x
Initialize: $S \leftarrow \emptyset$; **Output:** Refined concepts $\hat{c}$
for $t = 1$ **to** T **do**
  $i \leftarrow \pi(f, g, x, t, S)$; // Human provides c_i
  $S \leftarrow S \cup \{i\}$; $\hat{c} \leftarrow f(x)$
  $\Delta x = -\eta \cdot \text{sign}\left(\sum_{j \in S} \nabla_x \mathcal{L}(\hat{c}_j, c_j)\right)$
  $x \leftarrow x + \Delta x$
end for
$\hat{c} \leftarrow f(x)$; $\hat{c}_j \leftarrow c_j, \forall j \in S$
Return $\hat{c}$

merges the $\hat{c}_i < c_i$ and $\hat{c}_i > c_i$ cases into a single loss-minimization formulation and simplifies implementation.

A single intervention may be insufficient to align $\hat{c}_i$ with c_i. Over multiple rounds, we can incorporate previously intervened concepts S_t. For example, at round t, if concept i is selected, we set

$$\Delta x = -\eta \cdot \text{sign}\Big(\sum_{j \in S_t} \nabla_x L(\hat{c}_j(x), c_j) + \nabla_x L(\hat{c}_i(x), c_i)\Big). \tag{4}$$

If a previously corrected concept $j \in S_t$ is already close to its ground-truth value, its loss approaches zero and does not substantially affect the update. Otherwise, the deviation persists and further influences x so that $\hat{c}_j$ becomes more accurate.

Finally, in scenarios where the number of interactions is limited and may not fully align $\hat{c}_i$ with c_i, we adopt the following step before using the concept-based prediction for y: if a concept has been intervened upon, we simply replace that concept's predicted value with its ground-truth label. When enough intervention steps have brought $\hat{c}_i$ sufficiently close to c_i, this last operation becomes negligible. For details of the implementation of CI, please refer to Algorithm 1.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct our experiments on two datasets: Animals with Attributes 2 (AwA2) [19] and Caltech-UCSD Birds-200-2011 (CUB) [18]. AwA2 contains 85 attributes. In CUB, following [7, 17], we use 112 concepts grouped into 28 concept groups. We evaluate our methods on both *Individual Information* and *Grouped Information* settings. Note that AwA2 does not provide any concept grouping annotations. Hence, as in [17], we do not conduct group-based experiments on AwA2.

Experimental Configuration. We perform a systematic analysis of different concept-intervention strategies under various model architectures, datasets, and learning rates. The three model architectures include: (1) **CBM** (Concept

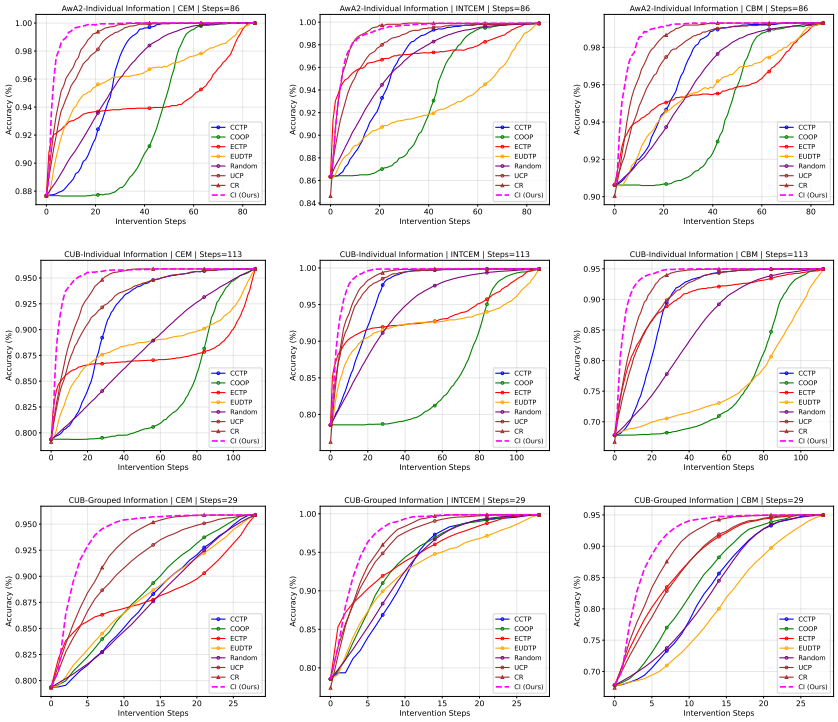


Fig. 4: Overall performance (accuracy vs. number of interventions) on different datasets and model architectures. Our Causal Intervention (CI) method converges faster and achieves higher performance compared to other methods. Each subfigure represents results for a specific dataset-model combination.

Bottleneck Model) (ICML 2020) [9], (2) **CEM** (Concept Embedding Model) (NeurIPS 2022) [6], (3) **Int-CEM** (Intervention-aware CEM) (NeurIPS 2023) [7], each representing a distinct paradigm for concept learning.

We consider six different intervention policies: **ECTP** (Expected Change in Target Prediction) (ICML 2023) [16], **EUDTP** (Expected Uncertainty Decrease in Target Prediction) (NeurIPS 2022) [15], **CCTP** (Contribution of Concept on Target Prediction) (ICML 2023) [16], **COOP** (Cooperative Prediction) (AAAI 2023) [3], **Random** (ICML 2020) [9], and **UCP** (Uncertainty of Concept Predictions) (NeurIPS 2022) [6].

Our main competitor is **Concept Realignment (CR)** [17], a recent approach published at ECCV 2024. For fairness, both CR and our method adopt UCP as the intervention policy. We discuss the impact of different policies on performance in Section 4.6.

To study the effect of optimization, we also test four different learning rates (0.25, 0.125, 0.0625, and 0.03125). We analyze how changing the learning rate influences performance in Section 4.4.

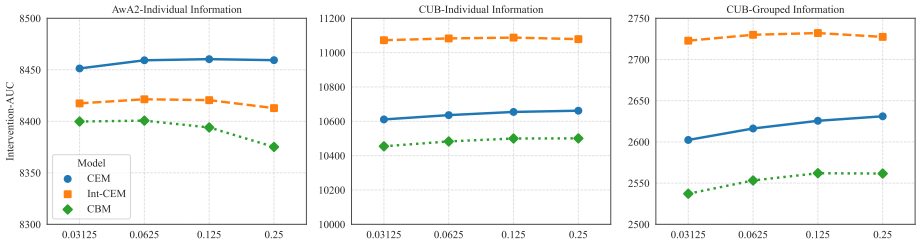


Fig. 5: Comparison of different learning rate across multiple datasets and model architectures.

Table 1: Overall Intervention-AUC comparison for all methods. CI demonstrates the highest AUC in nearly every scenario, with a 5.07% average gain over the mean of competing methods.

Method	AwA2-Individual Information			CUB-Individual Information			CUB-Grouped Information		
	CEM	Int-CEM	CBM	CEM	Int-CEM	CBM	CEM	Int-CEM	CBM
CCTP	8212.86	8199.92	8249.13	10271.53	10827.66	10043.10	2459.31	2608.31	2333.78
COOP	7930.78	7929.28	8030.67	9408.59	9612.37	8552.05	2483.51	2638.15	2382.78
ECTP	8049.32	8269.46	8150.07	9806.57	10485.27	10086.05	2472.20	2654.51	2469.96
EUDTP	8186.47	7880.74	8145.49	9927.85	10346.97	8542.09	2471.51	2600.54	2254.51
Random	8207.47	8210.75	8193.44	9910.29	10594.37	9599.05	2453.72	2619.40	2333.13
UCP	8378.17	8350.76	8320.14	10434.94	10984.10	10196.66	2557.07	2697.99	2463.68
CR	8412.84	**8424.78**	8364.40	10556.64	11016.59	10366.55	2595.53	2707.76	2521.32
CI (Ours)	**8460.33**	8421.38	**8400.64**	**10661.75**	**11087.43**	**10500.56**	**2631.08**	**2732.08**	**2562.02**

Input-change concern. One might worry that updating the input x resembles an adversarial attack. In CI, however, the update is tied to human-provided concept feedback and is bounded by the small step size η. Measured perturbations are small in practice: for CBM, $L_\infty \approx .002$ and SSIM is .999; for grouped CEM, $L_\infty = .025$ and SSIM is .982. Individual CEM can accumulate more change over 112 interventions (SSIM .934), so we report norms and SSIM rather than relying on an imperceptibility claim.

4.2 Evaluation Metric

Following [17], we quantify the cumulative effect of iterative interventions via the *Intervention-AUC*. If a_i denotes the accuracy at step i, then Intervention-AUC is computed by a trapezoidal-rule summation:

$$\text{Intervention-AUC} = \sum_{i=1}^{n-1} \frac{a_i + a_{i-1}}{2} \Delta i \tag{5}$$

where, $\Delta i = i - (i-1) = 1$. This integration of accuracies $\{a_0, a_1, \ldots, a_{n-1}\}$ over n steps yields a single-value metric summarizing the model's performance throughout the intervention process. A larger Intervention-AUC indicates that the intervention method is more effective.

Table 2: Intervention-AUC comparison at the first 10 steps. Our method leads to an 8.59% average improvement over all other methods and outperforms CR by an average of 3.90%. Results for 5, 15, and 20 steps are provided in the supplementary material.

Method	AwA2-Individual Information			CUB-Individual Information			CUB-Grouped Information		
	CEM	Int-CEM	CBM	CEM	Int-CEM	CBM	CEM	Int-CEM	CBM
CCTP	791.35	782.35	820.06	719.12	729.91	618.45	732.02	750.88	636.50
COOP	789.00	777.62	815.62	714.37	707.21	610.29	738.61	775.13	655.75
ECTP	826.87	842.91	836.08	757.68	788.23	679.32	763.54	801.87	706.31
EUDTP	814.81	787.70	819.58	731.64	751.59	616.15	742.64	772.47	626.49
Random	799.87	794.80	820.06	720.49	726.21	622.18	733.21	762.12	643.16
UCP	828.70	821.79	832.60	750.16	781.11	665.16	770.16	811.45	697.75
CR	838.76	842.63	840.37	757.10	781.15	682.99	784.10	815.07	729.09
CI (Ours)	**863.93**	**844.96**	**859.28**	**801.91**	**814.01**	**743.31**	**811.87**	**836.43**	**761.63**

4.3 Results

Because the maximum number of interventions is dictated by the number of concepts available in each dataset, the theoretical upper bound of the Intervention-AUC varies across datasets. Specifically, for AwA2-Individual Information, the maximum number of interventions is 85, implying a maximum Intervention-AUC of 8500. For CUB-Individual Information and CUB-Grouped Information, the maximum intervention rounds are 112 and 28, respectively, giving upper bounds of 11200 and 2800.

As shown in Figure 4, our CI approach outperforms all other methods across nearly all datasets and models, achieving top performance with fewer interventions. In AwA2-Individual and CUB-Individual settings, CI converges within about 20 intervention steps, whereas other methods require significantly more steps. For instance, ECTP, EUDTP, COOP, CCTP, and Random only reach their best performance close to exhausting all concept interventions. Meanwhile, our main competitor, CR, typically needs 10–20 extra steps to approach our results. These findings demonstrate that our method achieves stronger performance with lower computational cost. Notably, when interacting with Int-CEM, CR's retraining procedure leads to performance that is close to ours, yet this scenario is slightly unfair in favor of CR because its design specifically retrains the Int-CEM model for CR. Despite this, our method still yields better results on CUB and achieves comparable performance to CR on AwA2. In Table 1, our method also achieves the best intervention-AUC in almost all experiments.

The supplementary diagnostic ablations further isolate the source of the gain. Direct concept clamping tests whether simply replacing queried concepts is sufficient; a pre-concept hidden carrier tests whether an earlier latent feature can substitute for the input carrier; raw-gradient and no-clamp variants test the update sign and the role of the final ground-truth clamp. Across the intended low-intervention regime, the input carrier consistently improves AUC@20 over the hidden carrier and direct clamping in the CUB diagnostic settings. Removing the final clamp remains competitive in some CEM settings, but the clamp is important for stable late-stage CBM behavior. These results support the design

Table 3: Efficiency comparison on E-CBM (MNIST-Add).

Metric	CCTP	COOP	ECTP	EDUTP	Rand.	UCP	CI (Ours)
AUC (All)	1916.3	1915.6	1933.6	1885.0	1916.6	1942.6	**1957.8**
AUC (10)	583.8	582.2	600.4	586.3	587.8	601.8	**605.1**
Time (s)	15.6	43.1	**13.2**	14.4	25.6	14.3	28.7
Mem (MB)	864	864	864	1296	1422	1296	**1022**

choice of using the shared $x \to c$ pathway to propagate feedback while preserving direct correction of queried concepts. However, as the number of intervention steps increases, all methods tend to converge in the final stages. Therefore, we analyse results of early intervention steps in the next section.

Early Intervention Steps In this section, we report Intervention-AUC results for the first 10 intervention steps in Table 2; additional 5, 15, and 20 step results appear in the supplementary material. As shown in Table 2, CI consistently achieves the best performance across all experiments at 10 intervention steps, surpassing other methods by an average of 8.59%. In particular, we achieve a 3.90% improvement over CR, confirming our efficiency and strong performance in early interventions. This advantage remains as the number of interventions grows. The supplementary 15- and 20-step results show that CI still leads by an average margin of 9.14% and 8.85%, outperforming CR by 3.46% and 2.88%. Effects of different intervention policies are provided in Section 4.6.

4.4 Effect of the Learning Rate

To investigate how the learning rate influences performance, we fix the intervention policy to UCP and evaluate three models (CEM, Int-CEM, CBM) on three datasets (AwA2-Individual, CUB-Individual, CUB-Grouped) using four learning rates: 0.03125, 0.0625, 0.125, and 0.25. As shown in Figure 5, the results reveal a non-monotonic relationship between learning rate and performance. For instance, in CUB-Individual with CEM, performance improves as the learning rate increases (0.03125: 10610.90 $\to$ 0.25: 10661.75), whereas Int-CEM peaks at 0.125 (11087.43). In CUB-Grouped, Int-CEM also achieves its best performance at 0.125 (2732.08), whereas CBM shows a slight decrease when moving from 0.03125 to 0.25 (2537.13 vs. 2561.61). In practice, however, overall sensitivity to learning rate is modest; performance variations remain within a manageable range, indicating that learning rate is not a dominant factor in most scenarios.

4.5 Additional Architectures and Efficiency Analysis

To demonstrate the generalizability of our method beyond standard CBMs and CEMs, we apply our Causal Intervention (CI) update to the recently proposed Energy-based CBM (E-CBM) [20]. We evaluate the models on the MNIST-Add dataset.

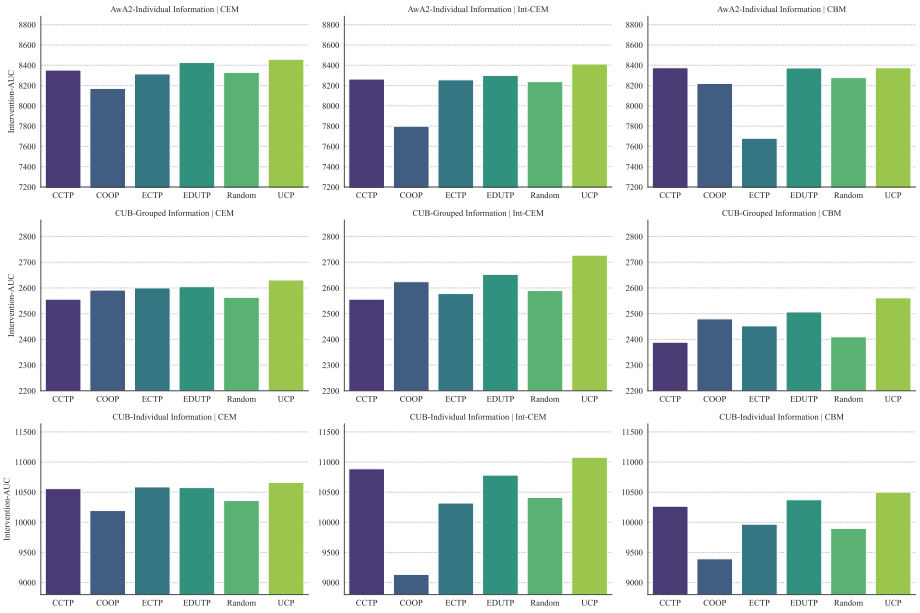


Fig. 6: Comparison of different intervention policies across multiple datasets and model architectures.

Furthermore, we assess the computational efficiency of our method by comparing the runtime (in seconds) and GPU memory footprint (in MB). All measurements are conducted on a single NVIDIA L40S GPU, over 33,000 samples and 32 intervention rounds.

As shown in Tab. 3, CI consistently achieves the highest Intervention-AUC in both the Overall metric and the first 10 steps. Notably, CI only consumes 1022 MB of GPU memory, which is significantly lower than EDUTP/UCP (1296 MB) and Random (1422 MB). The total runtime is 28.65 s, which remains in the same order of magnitude as other methods, with the slight increase attributed to the necessary gradient computation in CI. These results confirm that our method provides substantial performance gains with a moderate and deployable resource footprint.

4.6 Effect of Different Intervention Policies.

We analyze how different policies (CCTP, COOP, ECTP, EUDTP, Random, UCP) affect performance under a fixed learning rate (0.25). We compare CEM, Int-CEM, and CBM on AwA2-Individual, CUB-Individual, and CUB-Grouped. As shown in Figure 6, on AwA2-Individual, UCP significantly outperforms the others (CEM: 8459.34 vs. COOP: 8171.87, a 3.5% increase). In CUB-Grouped, with Int-CEM, UCP also achieves the best AUC (2727.42), whereas under CBM, EUDTP (2506.72) and UCP (2561.61) differ by less than 2%, suggesting the

need to balance policy complexity and benefits in grouped settings. In CUB-Individual, Int-CEM is highly sensitive to policy selection (UCP: 11078.61 vs. COOP: 9135.53, a 21.3% gap), while CBM exhibits larger performance fluctuations (UCP: 10500.56 vs. COOP: 9393.54). Overall, Random maintains moderate but stable performance (e.g., AwA2: 8329.74, CUB-Individual: 9899.59). These findings confirm the effectiveness of carefully designed policies; UCP is generally advantageous for individual attribute modeling, while EUDTP can serve as a lightweight alternative in grouped settings.

5 Conclusion

We introduced *Causal Intervention* (CI), a training-free approach that leverages correlations learned within CBMs to improve the efficiency and effectiveness of concept-based interventions. CI uses the shared upstream input carrier and the trained $x \to c$ pathway to adjust concept predictions while retaining the standard ground-truth clamp for queried concepts. By doing so, CI avoids uncertainty related to additional training and exploits concept-correlation information already embedded within CBMs. We believe our work provides a useful step toward more efficient human-model interaction in high-stakes decision-making contexts.

References

1. Amgad, M., Hodge, J.M., Elsebaie, M.A., Bodelon, C., Puvanesarajah, S., Gutman, D.A., Siziopikou, K.P., Goldstein, J.A., Gaudet, M.M., Teras, L.R., et al.: A population-level digital histologic biomarker for enhanced prognosis of invasive breast cancer. Nature medicine **30**(1), 85–97 (2024)
2. Cao, Y., Chen, Z., Pei, Q., Dimino, F., Ausiello, L., Kumar, P., Subbalakshmi, K., Ndiaye, P.M.: Risklabs: Predicting financial risk using large language model based on multi-sources data. CoRR **abs/2404.07452** (2024)
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)
4. Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
5. Chen, Z.Z., Ma, J., Zhang, X., Hao, N., Yan, A., Nourbakhsh, A., Yang, X., McAuley, J., Petzold, L., Wang, W.Y.: A survey on large language models for critical societal domains: Finance, healthcare, and law. arXiv preprint arXiv:2405.01769 (2024)
6. Espinosa Zarlenga, M., Barbiero, P., Ciravegna, G., Marra, G., Giannini, F., Diligenti, M., Shams, Z., Precioso, F., Melacci, S., Weller, A., et al.: Concept embedding models: Beyond the accuracy-explainability trade-off. Advances in Neural Information Processing Systems **35**, 21400–21413 (2022)
7. Espinosa Zarlenga, M., Collins, K., Dvijotham, K., Weller, A., Shams, Z., Jamnik, M.: Learning to receive help: Intervention-aware concept embedding models. Advances in Neural Information Processing Systems **36**, 37849–37875 (2023)

8. Kim, C., Gadgil, S.U., DeGrave, A.J., Omiye, J.A., Cai, Z.R., Daneshjou, R., Lee, S.I.: Transparent medical image ai via an image–text foundation model grounded in medical literature. Nature Medicine **30**(4), 1154–1165 (2024)
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)
10. Mao, A., Mohri, M., Zhong, Y.: Cross-entropy loss functions: Theoretical analysis and applications. In: International conference on Machine learning. pp. 23803–23828. PMLR (2023)
11. McGrath, T., Kapishnikov, A., Tomašev, N., Pearce, A., Wattenberg, M., Hassabis, D., Kim, B., Paquet, U., Kramnik, V.: Acquisition of chess knowledge in alphazero. Proceedings of the National Academy of Sciences **119**(47), e2206625119 (2022)
12. Richens, J.G., Lee, C.M., Johri, S.: Improving the accuracy of medical diagnosis with causal machine learning. Nature communications **11**(1), 3923 (2020)
13. Sedgwick, P.: Pearson's correlation coefficient. Bmj **345** (2012)
14. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
15. 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)
16. Shin, S., Jo, Y., Ahn, S., Lee, N.: A closer look at the intervention procedure of concept bottleneck models. In: International Conference on Machine Learning. pp. 31504–31520. PMLR (2023)
17. Singhi, N., Kim, J.M., Roth, K., Akata, Z.: Improving intervention efficacy via concept realignment in concept bottleneck models. In: European Conference on Computer Vision. pp. 422–438. Springer (2024)
18. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Tech. rep., California Institute of Technology (2011)
19. 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)
20. Xu, X., Qin, Y., Mi, L., Wang, H., Li, X.: Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. arXiv preprint arXiv:2401.14142 (2024)