Position: In Defense of Information Leakage in Concept-based Models

Mateo Espinosa Zarlenga [1]

Abstract

Concept-based models (CMs), deep neural networks that ground their predictions on representations aligned with human-understandable concepts (e.g., `round`, `stripes`, etc.), have been shown to learn representations that *leak* concept-irrelevant information. As the traditional narrative goes, this leakage is undesirable and should be eradicated as it leads to uninterpretable models. In this paper, we posit that this conventional view of leakage in CMs is not only ill-posed, as the evidence of how leakage makes a model less interpretable is often inconclusive, but also bound to lead to impractical CMs under common real-world constraints. Specifically, we argue that *in real-world settings where concept incompleteness is the norm, some leakage is often necessary for constructing accurate and intervenable CMs*. To this end, we propose that there is such a thing as *benign* leakage and show that, by optimizing a reframing of the typical CM training objective, CMs can encourage and exploit this form of leakage without sacrificing accuracy or intervenability.

1. Introduction

The field of machine learning is teeming with phenomena whose nature has been misconstrued by early incomplete explanations. Batch Normalization (Ioffe & Szegedy, 2015) is still frequently explained as improving optimization by mitigating internal covariate shift, despite substantial empirical and theoretical evidence that this mechanism is, at best, secondary (Santurkar et al., 2018; Bjorck et al., 2018; Lipton & Steinhardt, 2019). Likewise, generalization phenomena in Deep Neural Networks (DNNs), such as grokking (Power et al., 2022), double descent (Belkin et al., 2019), and benign overfitting (Bartlett et al., 2020), are often framed as phenomena unique to DNNs, even though closely related behaviors have been demonstrated across a range of model classes (e.g., see Wilson 2025). A similar pattern appears in common interpretations of attention: attention weights are frequently treated as causal explanations for model predictions, despite strong evidence that they do not, in general, faithfully reflect a model's decision process (Jain & Wallace, 2019; Serrano & Smith, 2019; Wiegreffe & Pinter, 2019). This paper aims to prevent the mischaracterization of another phenomenon that risks a similar trajectory, namely the effect of *information leakage* in concept-based models.

Concept-based models (CMs) (Alvarez Melis & Jaakkola, 2018; Koh et al., 2020; Yuksekgonul et al., 2023; Oikarinen et al., 2023), a rapidly growing class of methods in explainable artificial intelligence (XAI) (Poeta et al., 2023), are models that ground their downstream predictions in representations aligned with human-understandable concepts. A growing body of work has shown that the concept representations learned by CMs are often prone to *information leakage*, in which concept representations encode information that is not strictly attributable to their intended semantics (Margeloiu et al., 2021; Mahinpei et al., 2021). The prevailing narrative treats such leakage as inherently undesirable and calls for its elimination, typically arguing that leakage undermines the interpretability of CMs (Marconato et al., 2022; Havasi et al., 2022; Espinosa Zarlenga et al., 2023a; Parisini et al., 2025). In this paper, we argue that this strict view of leakage is ill-posed and counterproductive.

Importantly, we do not claim that leakage is never harmful. Rather, we argue that treating leakage as a purely negative and uncontrollable property obscures important distinctions and leads to design choices that render CMs impractical under common real-world constraints. In particular, in settings where *concept incompleteness* is the norm, enforcing strictly non-leaky representations can preclude both high task accuracy and effective test-time interventions. As there is significant evidence that both of these properties are attainable even in the presence of leakage, we take the position that **not all forms of leakage are malign, and that a controlled form of *benign* leakage can be necessary and desirable for constructing accurate and intervenable CMs**.

To substantiate this position, we first describe a general framework for analyzing CMs (Sec. 2) and use it to formalize information leakage (Sec. 3). Then, we summarize the dominant arguments against leakage (Sec. 4) and make the case that some degree of leakage, which we define as *benign*

[1]University of Oxford, UK. Correspondence to: Mateo Espinosa Zarlenga <mateo.espinosazarlenga@trinity.ox.ac.uk>.

Preprint. February 8, 2026.

leakage, is often desirable if CMs are to remain useful in incomplete real-world settings (Sec. 5). Finally, we show that this form of benign leakage can be compatible with the core desiderata of CMs, including task fidelity and intervenability, provided that CMs are trained to minimize their task loss when all concepts are intervened (Sec. 6). In doing so, we show that, contrary to common narratives, leakage does not imply that a CM loses the characteristics typically associated with its interpretability and may, instead, be necessary for building accurate CMs in real-world conditions.

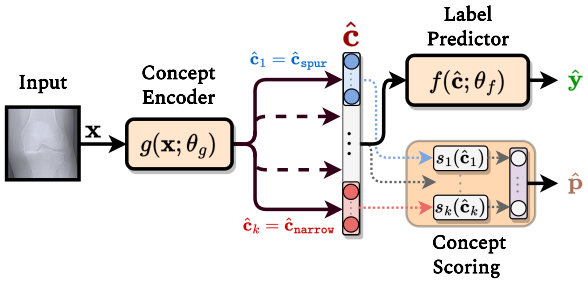

Figure 1. A generalized Concept Bottleneck Model (CBM).

2. Background

A concept-based model (CM) $\eta : \mathcal{X} \to \mathcal{Y} \times \mathcal{C}$ is a DNN that maps features $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ (e.g., pixels) to a *downstream task prediction* $\hat{\mathbf{y}} \in \mathcal{Y}$ (e.g., bird species) and a *concept-based explanation* $\hat{\mathbf{p}} \in \mathcal{C}$ (e.g., `black_wings`, `medium_sized`). Here, without loss of generality, we assume $\hat{\mathbf{y}} \in [0,1]^L$ is a distribution over L labels and $\hat{\mathbf{p}} \in [0,1]^k$ is a vector of concept scores, with $\hat{\mathbf{p}}_i$ approximating the probability that concept C_i is present in $\mathbf{x}$. Traditionally, most CMs assume access to a training dataset $\mathcal{D} = \{(\mathbf{x}^{(j)}, \mathbf{c}^{(j)}, y^{(j)})\}_{j=1}^N$, where each input is annotated with a downstream label $y^{(j)} \in \{1, \ldots, L\}$ and a vector of k binary concept annotations $\mathbf{c}^{(j)} \in \{0,1\}^k$. To obtain this dataset, concept annotations may be provided by experts (e.g., radiologists' notes) or obtained via *label-free* pipelines (Oikarinen et al., 2023; Yang et al., 2023; Rao et al., 2024). Considering this setup, we now describe CBMs, a unifying framework for studying CMs.

Concept Bottleneck Models Given a concept-annotated dataset $\mathcal{D}$, most CMs can be framed as a Concept Bottleneck Model (CBM) (Koh et al., 2020). In its general form, seen in Figure 1, a CBM is a pair of functions $M = (g, f)$, both parameterized as DNNs, supported by a *scoring function* $s : \mathbb{R}^{k \times m} \to [0,1]^k$. The first function $g : \mathbb{R}^n \to \mathbb{R}^{k \times m}$, called the *concept encoder*, maps input features $\mathbf{x}$ to k concept representations $\hat{\mathbf{c}} = (\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k)$, where $\hat{\mathbf{c}}_i \in \mathbb{R}^m$. CBMs learn $\hat{\mathbf{c}}$ such that the i-th output of the scoring function $s(\hat{\mathbf{c}}) = \hat{\mathbf{p}} \in [0,1]^k$, namely $s_i(\hat{\mathbf{c}}) := \hat{\mathbf{p}}_i$, approximates $\mathbb{P}(C_i = 1 \mid X = \mathbf{x})$. Then, the *label predictor* $f : \mathbb{R}^{k \times m} \to [0,1]^L$ takes the representations $\hat{\mathbf{c}}$ and predicts a downstream task label distribution $\hat{\mathbf{y}} = f(\hat{\mathbf{c}})$.

Together, the composition $\eta = (f \circ g)$ forms a predictor $\mathbf{x} \mapsto \hat{\mathbf{y}}$ whose output can be explained by the *concept bottleneck*'s scores $\hat{\mathbf{p}} = s(g(\mathbf{x}))$. Therefore, when trained to align $\hat{\mathbf{p}}$ with $\mathbf{c}$, η is considered interpretable. In practice, this is achieved by learning the parameters of g and f such that a combination of a task loss $\mathcal{L}_y(f(g(\mathbf{x})), y)$ (e.g., cross-entropy) and a concept loss $\mathcal{L}_c(s(g(\mathbf{x})), \mathbf{c})$ (e.g., binary cross-entropy) (Koh et al., 2020) is minimized. These losses may be optimized *independently*, *sequentially* (training g before f), or *jointly* (minimizing a weighted sum).

The simplest instantiation of this framework is a *Sigmoidal CBM* (Koh et al., 2020), where each concept is represented by a single sigmoidal scalar ($m = 1$) and s is the identity. More recent works have explored richer representations and scoring mechanisms, including embedding-based concept representations (Espinosa Zarlenga et al., 2022; Kim et al., 2023; Xu et al., 2024; Espinosa Zarlenga et al., 2025), logit-based (Koh et al., 2020) or unbounded scores (Yuksekgonul et al., 2023; Oikarinen et al., 2023), and bottlenecks augmented with residual or unsupervised side-channels (Mahinpei et al., 2021; Havasi et al., 2022; Hu et al., 2025).

Concept Interventions CMs that can be framed as CBMs support test-time feedback via *concept interventions* (Koh et al., 2020). An intervention on concept i, denoted as $g(\mathbf{x} \mid C_i := v)$, consists of modifying the concept representation $\hat{\mathbf{c}}_i$ such that it corresponds to a desired concept value $v \in [0,1]$, typically by setting $\hat{\mathbf{c}}_i := s_i^{-1}(v)$ when s is invertible. The modified representation is then propagated to the label predictor, which may in turn update its task prediction.

Concept interventions are commonly framed as a mechanism for *correcting mispredicted concepts*: when predicting a high-level concept is easier than predicting the downstream task (e.g., `black_wings` versus bird species), a user can correct erroneous concept predictions, thereby improving task performance. However, this view understates their broader utility. Concept interventions also provide a natural interface for supplying *high-level hints* to the model, expressed in a vocabulary *shared* by experts and the model itself (i.e., the concepts). For example, through an intervention, a radiologist may indicate the presence of "bone spurs" on an X-ray before the CM processes the input features, effectively conditioning the CM's inference on prior knowledge that cannot be easily communicated through low-level feature manipulations (e.g., via pixel-level manipulations). More recent works have explored when and where interventions should be made (Shin et al., 2023; Chauhan et al., 2023; Pugnana et al., 2025), as well as how to increase the effectiveness of interventions (Espinosa Zarlenga et al., 2023b; Steinmann et al., 2024; Vandenhirtz et al., 2024).

Desiderata of CMs CMs are typically designed to achieve

predictive accuracy and interpretability. Although there is a lack of a task-agnostic, measurable definition of "interpretability" (Doshi-Velez & Kim, 2017), the existing literature on CMs commonly evaluates these two properties based on three measurable proxy metrics:

1. *Task Fidelity*: how accurate are the CM's downstream task predictions? (i.e., is $f(g(\mathbf{x})) \approx y$?)
2. *Concept Fidelity*: how accurate are the CM's explanations? (i.e., is $s_i(g(\mathbf{x})) \approx \mathbf{c}_i$ for all $i \in \{1, \cdots, k\}$?)
3. *Intervenability* (Laguna et al., 2024): when provided with ground-truth concepts $\mathbf{c}$, is the CM's task prediction the same as an expert would provide for $\mathbf{c}$? (i.e., does task fidelity increase as we intervene on more concepts?)

While task and concept fidelity are expected desiderata, intervenability plays a distinct role in CMs: it provides an operational test of whether the model correctly uses concepts when making predictions. By examining how predictions change under concept interventions, one can assess whether the model's reasoning aligns with that of the experts who annotated a test set with concepts. Therefore, intervenability allows us to discriminate between a CM that treats concept predictions as outputs disconnected from task inference and a model that reasons about its downstream task prediction based on the high-level concepts it has at its disposal.

Having established a framework and a set of desiderata for studying CMs, we now define *information leakage* in CMs.

3. Defining Information Leakage in CMs

Recent works have shown that a CM's concept representations $\hat{\mathbf{c}} = g(\mathbf{x})$ may encode information that is not strictly attributable to their intended semantics (Margeloiu et al., 2021; Mahinpei et al., 2021). For instance, by analyzing saliency maps of concept encoders in CBMs, Margeloiu et al. (2021) showed that high concept accuracy does not imply that a concept predictor relies exclusively on concept-specific features. Follow-up work by Mahinpei et al. (2021) demonstrated that representing concepts via *soft* (continuous) representations (e.g., logits or probabilities) can allow a label predictor to exploit information about other concepts or the downstream task. Related work further showed that such representations may fail to respect the spatial *locality* of concepts in the input (Raman et al., 2025). Collectively, these observations motivate the view that accurate concept prediction alone does not guarantee that concept representations form an effective bottleneck (Almudévar et al., 2025).

Forms of Leakage Previous works (Parisini et al., 2025; Makonnen et al., 2025) distinguish between two forms of leakage (represented in Figure 2). *Inter-concept leakage* refers to the case where the representation $\hat{\mathbf{c}}_i$ of concept c_i contains information about concept c_j beyond what is present in the ground-truth concept labels. Formally, this implies that $I(\hat{C}_i; C_j) > I(C_i; C_j)$, where $\hat{C}_i$ and C_i denote the random variables associated with the representation and ground-truth label of concept c_i, respectively, and $I(\cdot;\cdot)$ is the mutual information. In contrast, *task leakage* refers to a concept representation encoding information on Y that is not attributable to the concept itself, i.e., when $I(\hat{C}_i; Y) > I(C_i; Y)$, with Y denoting the downstream task label. Notice that, in practice, $\hat{C}_i$ is typically continuous and high-dimensional. Therefore, the quantities characterizing both forms of leakage can only be estimated (Makonnen et al., 2025; Parisini et al., 2025) or assessed via proxies (Mahinpei et al., 2021; Espinosa Zarlenga et al., 2023a).

Leakage via Bypasses and Side Channels Leakage may also arise from modeling choices in the label predictor. In particular, several CMs allow the predictor $f(\cdot)$ to exploit information outside the concept bottleneck $\hat{\mathbf{c}}$ via bypasses or residual connections of the form $f(\hat{\mathbf{c}}, \psi(\mathbf{x}))$, where $\psi(\mathbf{x})$ is an unconstrained representation of the input (Sawada & Nakamura, 2022; Yuksekgonul et al., 2023; Havasi et al., 2022). For example, Hybrid CBMs (Mahinpei et al., 2021) explicitly add k' unsupervised activations $\psi(\mathbf{x}) \in \mathbb{R}^{k'}$ to the bottleneck $\hat{\mathbf{c}}$, which f can exploit to make its label prediction. Such designs are typically motivated as pragmatic responses to *incompleteness* in the concept annotation set $\mathbf{c}$. In those instances, restricting the label predictor to operate on the bottleneck alone can substantially degrade task fidelity (Espinosa Zarlenga et al., 2022). From the perspective of leakage, such bypasses constitute a direct route for task-relevant information to influence predictions. Accordingly, we treat $\psi(\mathbf{x})$ as a *shared concept representation* that may carry both inter-concept and task leakage.

Now that we have formalized what leakage is, we consider how leakage has been framed in the existing literature.

4. Alternative Views: Against Leakage

The CM literature holds the prevailing view that leakage is a purely negative property of CMs that should be avoided whenever possible. This has motivated methods for preventing leakage (Marconato et al., 2022), mitigating its presence (Havasi et al., 2022; Sheth & Ebrahimi Kahou, 2024; Sun et al., 2024; Ragkousis & Parbhoo, 2024; Almudévar et al., 2025), and quantifying it (Espinosa Zarlenga et al., 2023a; Marconato et al., 2024; Parisini et al., 2025; Makonnen et al., 2025; Aysel et al., 2025). Before making our case *for* leakage, we summarize the arguments commonly put forth against leakage by splitting them into three types of claims: intervenability, interpretability, and safety claims.

Intervenability Claims A first line of argument holds that leakage undermines the *intervenability* of CMs (Havasi et al., 2022; Espinosa Zarlenga et al., 2023a; Vandenhirtz

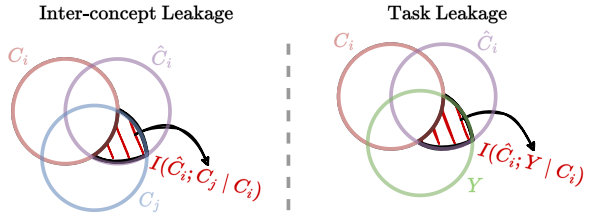

Figure 2. Leakage happens when the concept representation $\hat{C}_i$ encodes information that is not attributable to the ground-truth concept C_i. This additional information can be attributed to **(left)** another concept C_j or **(right)** the downstream task Y.

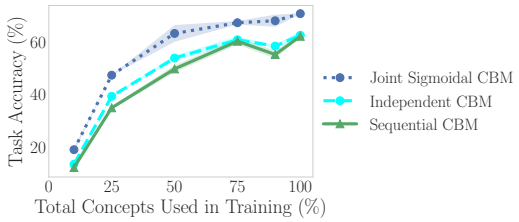

Figure 3. Task fidelity of sigmoidal (i.e., soft) CBMs as we vary the number k of training concepts on the CUB (Wah et al., 2011) dataset (see App. B for details). CBMs that limit task leakage (e.g., independent CBMs) are more affected by incompleteness.

et al., 2024; Ragkousis & Parbhoo, 2024; Sun et al., 2024; Almudévar et al., 2025). These claims argue that, when concept representations encode additional information beyond their intended semantics, interventions may have unpredictable effects on downstream predictions as they may inadvertently modify that latent information (Havasi et al., 2022; Espinosa Zarlenga et al., 2023a; Parisini et al., 2025). From this perspective, leakage seems to threaten the expectation that providing correct concept labels to a CM at test time should reliably improve its task accuracy.

Interpretability Claims A second class of arguments asserts that leakage renders CMs fundamentally "uninterpretable" (Margeloiu et al., 2021; Mahinpei et al., 2021; Marconato et al., 2022; Ragkousis & Parbhoo, 2024; Sinha & Zhang, 2025; Almudévar et al., 2025; Makonnen et al., 2025; Sun et al., 2024; Parisini et al., 2025). These claims argue that, if a concept representation encodes information beyond its corresponding ground-truth concept, the label predictor can no longer be said to reason *only* in terms of known concepts. Nevertheless, given the lack of an agreed, measurable definition of interpretability in the context where these claims are made, these statements are often *ill-posed*. This is because it is impossible to quantify or falsify any interpretability claim without first defining under what view of interpretability we should study leakage. Hence, when evaluating these claims later, we instead measure interpretability using commonly used proxy metrics for CMs (e.g., concept fidelity, intervenability, and label-predictor weights).

Safety Claims Finally, a small number of works argue that leakage poses *safety* risks (Marconato et al., 2022; 2024; Parisini et al., 2025). Specifically, leakage may severely affect intervenability when distribution shifts occur (Espinosa Zarlenga et al., 2025) and it may also facilitate *shortcut learning* (Geirhos et al., 2020), where the label predictor learns to exploit spurious correlations encoded in the representation. Among these claims, concerns about shortcut learning are especially important, as they may lead to models that fail to generalize and violate desirable notions of fairness and privacy (Dwork et al., 2012; Pessach & Shmueli, 2022). Nevertheless, while deserving of study in CMs, shortcuts are not unique to CMs or to concept representations. Rather, they reflect general limitations of DNN-based representation learning systems. Hence, conflating leakage with shortcut learning risks obscuring the true origin of these shortcuts and disconnecting their study from the substantial body of work on robustness (Sagawa et al., 2020; Kim et al., 2019; Sohoni et al., 2020). Because of this, while we believe these concerns are legitimate, we do not frame them as a CM-specific limitation, and we will not aim to disprove their validity in this work. Instead, we focus on CM-specific limitations arising from leakage.

Before rebutting the commonly held claims against leakage presented above, we first put forth our case *for* leakage.

5. The Case for Leakage

It is a common oversimplification to assume that all concept-supervised datasets are equal. However, the nature of the training concept annotations is crucial for the end quality of a CM (Collins et al., 2023; Heidemann et al., 2023; Penaloza et al., 2025). This is because some concepts are (1) inherently more informative about the downstream task (e.g., `stripes` and `fur` are more helpful in describing an animal's species than `quadruped`), (2) redundant (e.g., `black` entails `dark_colour`), and (3) *necessary* for accurately predicting a downstream task (e.g., `scales` is required to classify an animal as a reptile).

Of these constraints, the most limiting is perhaps the latter. When the training concepts $\mathbf{c} \sim \mathbb{P}(C)$ lack a concept crucial for predicting a downstream task (as `scales` is for predicting reptiles), a CM may fail to achieve high task fidelity. This is particularly true if the CM's label predictor is constrained to make its prediction based only on representations that are direct proxies of the concepts' probabilities, as is the case in popular sigmoidal/soft CBMs (see Figure 3).

Therefore, underlying the setup for most CMs, there is an implicit assumption that the set of training concepts $\mathbf{c}$ is a *complete* description (Yeh et al., 2020) of the downstream task y. Formally, this means CMs require that the mutual information between the downstream label $y \sim \mathbb{P}(Y)$ and

the input features $\mathbf{x} \sim \mathbb{P}(X)$ given the concepts $\mathbf{c} \sim \mathbb{P}(C)$ is close to zero (i.e., $I(Y;\ X \mid C) \approx 0$).

In real-world concept-annotated datasets, however, *incompleteness is the typical case*. This is because it is not only costly and difficult to obtain large sets of concepts for every training sample, but it is also challenging to identify, a priori, all the important concepts one may need for future downstream tasks of interest. Moreover, although useful, label-free concept discovery pipelines are not a *sufficient* solution for incompleteness: as seen in these works' own evaluations, label-free models still tend to trail behind opaque DNNs, implying their concept sets C lack information that is present in X (e.g., Table 1 in Yuksekgonul et al. 2023, Table 2 in Oikarinen et al. 2023, and Table 1 in Rao et al. 2024). Furthermore, these approaches depend on the availability of (1) concepts that can be specified in natural language, and (2) foundation models that can reason about even highly domain-specific concepts and modalities, requirements which are not applicable for specialized tasks.

Taking the weak assumption that sacrificing task fidelity is highly undesirable even if this means higher interpretability, something observed across user studies (Papenmeier et al., 2019; Nussberger et al., 2022), we are then confronted by the following challenge: if we want CMs to become practical for real-world tasks where incompleteness is the norm, then these models must have a mechanism for the label predictor $f(\hat{\mathbf{c}})$ to access task-relevant information in the input features $\mathbf{x}$ that can never be present in a "pure" representation of the ground-truth concepts $\mathbf{c}$. In other words, we can only attain useful CMs in real-world incomplete settings if we encourage task leakage to capture $I(Y;\ X \mid C)$.

Should we therefore enable leakage in our concept representations and use it to our advantage? Or should we declare CMs potentially unsuitable for most realistic settings, where incompleteness is the norm? Here, we argue that we should enable leakage, as without it, we cannot construct CMs that can perform the one thing, for better or for worse, expected of statistical models: to be accurate on their downstream task. Concretely, however, we argue that we should encourage a specific kind of leakage that, when properly enabled, can be exploited by a CM without compromising its expected desiderata. We call this form *benign leakage*.

5.1. Benign Leakage

Informally, representations $\hat{C}$ exhibit *benign leakage* if (1) they preserve enough information in them to cover for $I(Y; X \mid C)$ when C is incomplete, and (2) they can be decomposed into a *concept-specific* component $\bar{C}_i$, which is the component one modifies when an intervention is performed, and *residual* component R_i orthogonal to $\bar{C}_i$.

Definition 5.1 (Benign leakage)**.** Let $\hat{C} = (\hat{C}_1, \ldots, \hat{C}_k)$ be the concept representations produced by a CM, and assume that for each i there exists a decomposition $\hat{C}_i \equiv (\bar{C}_i, R_i)$, where $\bar{C}_i$ is the concept-aligned component and R_i is a residual component. Let $R := (R_1, \ldots, R_k)$. We say that $\hat{C}$ exhibits *benign leakage* if the following conditions hold:

i. **Sufficiency:** $I(Y; R \mid C) \approx I(Y; X \mid C)$

ii. **Localization:** $I\Big(Y; \bar{C}_i \mid R_i, \hat{C}_{-i}\Big) \approx I(Y; C_i \mid C_{-i})$

where $C_{-i} = (C_1, \cdots, C_{i-1}, C_{i+1}, \cdots, C_k)$.

Intuitively, sufficiency ensures that all label-relevant information not captured by C is preserved in $\hat{C}$. In contrast, localization ensures that the information about a concept C_i that is informative for predicting Y can be *localized* within the $\bar{C}_i$ component of $\hat{C}_i$. Therefore, when both conditions are satisfied, a label predictor $f(\hat{\mathbf{c}})$ trained on representations with benign leakage can achieve high task fidelity in incompleteness, by exploiting information in both R and $\bar{C} = (\bar{C}_1, \cdots, \bar{C}_k)$, and be intervenable, by ensuring it uses the well-localized variables $\bar{C}$ to access information in C.

We point out that without considering how f operates (or how it learns), we cannot make claims about the intervenability of a CBM based only on properties of $\hat{C}$. For example, notice that a Hybrid CBM may very well exhibit benign leakage, as it cleanly separates the activations that encode concept predictions from those that carry additional leakage. Yet, there is strong evidence that these models, trained only on the standard joint CBM loss, are not intervenable (Espinosa Zarlenga et al., 2022). Because of this, benign leakage, or, for that matter, any property of the concept predictions alone, cannot necessarily imply that a CM is intervenable, as intervenability is a property of the label predictor too. Nevertheless, what benign leakage allows us to ascertain is that well-conditioned label predictors *can* be accurate and intervenable (i.e., it is a *necessary* condition for high task fidelity and intervenability).

Optimizing for Benign Leakage Given our definition of benign leakage, we notice that, under standard well-specification assumptions (see App. A for proof), achieving sufficiency for a CBM (g, f) is equivalent to minimizing the negative task likelihood under full concept interventions:

$$\begin{aligned}\mathcal{L}_{\text{int}} &= \mathbb{E}_{(\mathbf{x},\mathbf{c},y)\sim\mathcal{D}}\big[-\log \mathbb{P}_f\big(y \mid \hat{C} = g(\mathbf{x} \mid C := \mathbf{c})\big)\big] \\ &= \mathbb{E}_{(\mathbf{x},\mathbf{c},y)\sim\mathcal{D}}\Big[\mathcal{L}_y\Big(f\big(g(\mathbf{x} \mid C := \mathbf{c})\big), y\Big)\Big]\end{aligned}$$

Notice that $\mathcal{L}_{\text{int}}$ is similar to the "prior loss" used in (Espinosa Zarlenga et al., 2025) to learn robust, leakage-aware concept representations. The difference lies in the fact that $\mathcal{L}_{\text{int}}$ applies to any CM, not just those that decompose their representations into exogenous and endogenous variables.

In contrast, ensuring localization is not as easy as ensuring sufficiency. When one can explicitly decompose a CM's concept representation $\hat{C}_i$ into a concept-aligned component

$\bar{C}_i$ and a residual "leaky" component R_i (e.g., as in Hybrid CBMs (Mahinpei et al., 2021), Residual PCBMs (Yuksekgonul et al., 2023), and MixCEMs (Espinosa Zarlenga et al., 2025)), localization may be directly optimized if the mutual information terms are quantities that can be estimated (a likely intractable problem if $\hat{C}$ is high-dimensional and not well-constrained). If a CM does not explicitly decompose its concept representations, then optimizing directly for localization becomes even more difficult with an architecture-agnostic regularizer such as $\mathcal{L}_{\text{int}}$. Nevertheless, we can still introduce implicit incentives that encourage localization. As we argue in the next section, due to the tendency of DNNs to learn simpler hypotheses, such an incentive may be a by-product of minimizing the sufficiency regularizer $\mathcal{L}_{\text{int}}$.

6. Revisiting the Arguments Against Leakage

Having argued that benign leakage is necessary for building accurate and intervenable CMs that can operate in incompleteness, we now make the case that information leakage does not *necessarily* affect the CM's intervenability or interpretability (under reasonable proxy metrics).

6.1. Revisiting Intervenability Claims

The first evidence against the common claim that information leakage leads to unintervenable CMs comes from previous works themselves: as recreated in Figure 4 (left), several CMs designed to enable leakage, either by using dynamic embedding representations, such as Concept Embedding Models (CEMs) (Espinosa Zarlenga et al., 2022) and Probabilistic CBMs (ProbCBMs) (Kim et al., 2023), or by using *small* residual bypasses, such as Hybrid CBMs (Mahinpei et al., 2021), are more or similarly intervenable to CBMs commonly agreed to have very minimal leakage (i.e., independently-trained sigmoidal CBMs). Similar results have been observed for other leakage-enabling CMs such as residual Autoregressive CBMs (Havasi et al., 2022), Semi-supervised CBMs (Hu et al., 2025), and further variants of CEMs (Espinosa Zarlenga et al., 2023b; 2025). Therefore, the common claim that leakage *necessarily* precludes a CM's intervenability is false.

It is important to notice, however, that it is true leakage *can* affect intervenability: as seen in Figure 4 (right), some CMs that enable leakage become almost entirely unintervenable when facing incompleteness (something we do not necessarily see when they are deployed in an equivalent, but complete setting as seen in App. C). Therefore, this begs the question: *when is a leakage-enabling CM intervenable?* As discussed next, this is possible when a CM (sometimes implicitly) optimizes for sufficiency (i.e., $\mathcal{L}_{\text{int}}$).

Simplicity Bias and Benign Leakage CMs that enable leakage can remain highly accurate and intervenable in incomplete settings if (1) interventions do not destroy the leakage (e.g., by not overwriting the activations that may be carrying the leaked information), *and* (2) they implicitly encourage forms of benign leakage. This is true, for example, for embedding-based approaches such as CEMs (Espinosa Zarlenga et al., 2022), ProbCBMs (Kim et al., 2023), and MixCEMs (Espinosa Zarlenga et al., 2025), where interventions are operations that swap one dynamic high-dimensional representation for another without destroying leakage. All of these models are randomly intervened on during training, a training-time mechanism called *RandInt* which can be seen as implicitly minimizing the *sufficiency* regularizer $\mathcal{L}_{\text{int}}$. Similar training pipelines with loss terms that more closely resemble $\mathcal{L}_{\text{int}}$ can be found in intervention-aware versions of CEMs (Espinosa Zarlenga et al., 2023b).

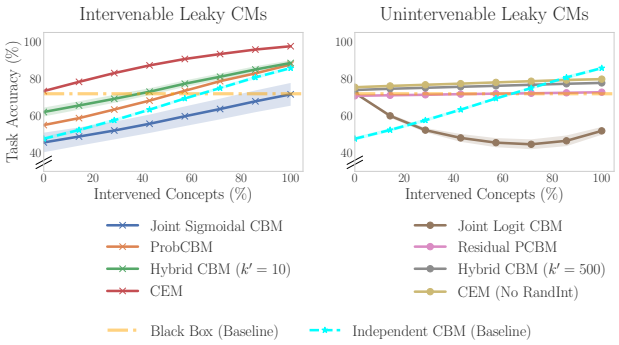

Figure 4. Task accuracy (y-axis) of leaky CMs trained on an incomplete version of CUB (see App. B for details) as we intervene on more concepts at test-time (x-axis). **(left)** Leaky CMs can remain intervenable in incompleteness. **(right)** Some leaky variants, however, lose their intervenability (curves are non-increasing).

However, as discussed in Sec. 5.1, *sufficiency* alone does not guarantee benign leakage. For a label predictor to be intervenable when leakage is present, the variables encoding concept-specific information must be well-localized within $\hat{C}$. Hence, if these leakage-enabling CMs only (implicitly) minimize $\mathcal{L}_{\text{int}}$, what leads them to also appear to attain *localization*? Moreover, even if concepts are localized, what encourages label predictors to properly attend to changes in these variables and become intervenable? Here we hypothesize that both of these effects can be seen as a consequence of the *simplicity bias* of DNNs: when given an option to learn several competing hypotheses, stochastic gradient descent favors simpler hypotheses over more complex ones (Pérez et al., 2019; Shah et al., 2020). Specifically, we argue that localization and intervenability naturally arise when jointly minimizing $\mathcal{L}_{\text{int}}$ because concept encoders and label predictors that achieve these properties are simpler than equally-performing models that do not.

To see this, consider, without loss of generality, what happens when we train a CM to minimize the traditional joint CBM loss together with the regularizer $\mathcal{L}_y(f(g(\mathbf{x} \mid C_i :=$

$\mathbf{c}_i)), y)$ (for some fixed concept index i). Here, whenever we perform an intervention $g(\mathbf{x} \mid C_i := \mathbf{c}_i)$, we are changing in some way the output representation $\hat{\mathbf{c}}_i$ of concept C_i (e.g., by fixing $\hat{\mathbf{c}}_i = c_i$ for sigmoidal CBMs). Therefore, when tasked to minimize the task loss given the ground-truth concept label $\mathbf{c}_i$, a label predictor that extrapolates how $\hat{\mathbf{c}}$ is affected by the intervention $g(\mathbf{x} \mid C_i := \mathbf{c}_i)$ (i.e., how $\hat{\mathbf{c}}$ encodes the true value of $\mathbf{c}_i$) will only require its concept predictor to learn to encode the information $I(Y; X \mid C_i)$ in the rest of its representations to be able to accurately predict Y. Learning this label predictor and concept encoder is arguably simpler than learning a concept encoder that needs to both (a) accurately predict C_i and re-encode its information somewhere new in $\hat{\mathbf{c}}$, and (b) still learn to encode the information $I(Y; X \mid C_i)$ in $\hat{\mathbf{c}}$. Therefore, when $\mathcal{L}_{\text{int}}$ is heavily penalized, both the label predictor and concept encoder will gravitate towards learning the simpler hypothesis whose label predictor (1) effectively extrapolates which variables in $\hat{\mathbf{c}}$ are affected by the intervention on concept C_i (i.e., leading to localization), and (2) uses this information to quickly minimize $\mathcal{L}_{\text{int}}$ (i.e., leading to intervenability).

The Effects of Proper Conditioning We note that indirect evidence for the hypothesis above already exists in the literature: Espinosa Zarlenga et al. (2025) show in their appendix that, when training a leaky CM using a stricter variant of $\mathcal{L}_{\text{int}}$ and a task loss term, the CM learns to implicitly align each ground-truth concept C_i with the set of variables that are affected when concept C_i is intervened on. This can be done even when **no explicit concept loss was used** to train the CM. We show how strong this result is: in Figure 5 we visualize the result of intervening on an otherwise *opaque DNN* that was trained with the regularizer $\mathcal{L}_{\text{int}}$. To achieve this, during training, we selected a set of k neurons in the penultimate layer of the DNN and intervened on them as if they were normal CBM soft concept representations (which they are not, as we do not introduce any concept alignment losses when training this model, and the neurons do not form a bottleneck). We see that this DNN, trained without any concept alignment loss, is intervenable and shows signs of localization as measured by the ROC-AUC between the neuron h_i used to intervene on concept C_i and the ground-truth $\mathbf{c}_i$ (we get a mean concept ROC-AUC of $80.79 \pm 0.11\%$). Hence, even though directly optimizing for *localization* is intractable, by heavily penalizing $\mathcal{L}_{\text{int}}$ we can implicitly encourage benign leakage.

Moreover, we observe that the localization resulting from explicitly minimizing $\mathcal{L}_{\text{int}}$ can be exploited to make leakage-enabling CBMs intervenable. As shown in Figure 6, by adding a strong weight to $\mathcal{L}_{\text{int}}$ when training Joint Logit CBMs, large Hybrid CBMs, and CEMs without RandInt, we can make all of these models intervenable, something we saw, in Figure 4 (right), did not naturally occur (App. D shows similar results for other CMs). Perhaps the most

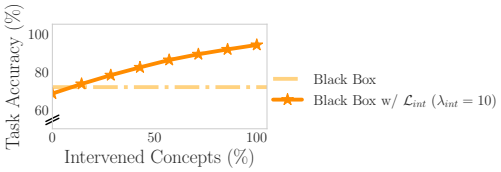

Figure 5. An opaque DNN trained without *any* concept alignment losses becomes intervenable when using regularizer $\mathcal{L}_{\text{int}}$.

surprising result is that even large Hybrid CBMs become highly intervenable when minimizing $\mathcal{L}_{\text{int}}$; all without sacrificing task or concept fidelities (something that *does not* happen when RandInt is introduced in their training, as seen in Espinosa Zarlenga et al. 2022). Therefore, these results strongly suggest that, when CMs are properly conditioned by penalizing $\mathcal{L}_{\text{int}}$, *information leakage does not necessarily lead to a loss of intervenability*. In fact, it may lead to intervenable CMs with higher task fidelities.

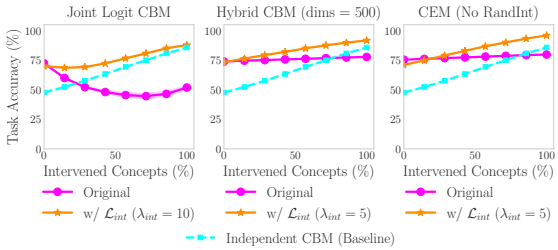

Figure 6. Effect of regularizing previously unintervenable leaky CMs with $\mathcal{L}_{\text{int}}$. We show an independent CBM as a baseline.

6.2. Revisiting Interpretability Claims

We now examine the interpretability claims against leakage. For this, we use as example the Hybrid CBM $M_H = (g_H, f_H)$ trained with $\lambda_{\text{int}} = 5$ in Figure 6. Given the number of unsupervised activations $\psi(\mathbf{x}) \in \mathbb{R}^{k'}$ in M_H's bottleneck $\hat{\mathbf{c}}$ (in this case $k' = 500$, much larger than the set of concept-aligned activations $k = 22$), this model has a strong incentive and mechanism to leak information in $\hat{\mathbf{c}}$. As such, we use M_H to study the validity of interpretability claims against leakage. We summarize our findings below.

It is unclear how leakage prevents interpretability Interpretability claims against leakage typically argue that if a concept representation, or the concept bottleneck $\hat{\mathbf{c}}$ more generally, encodes information beyond their corresponding concepts, then the CM is less "interpretable". However, our experiments above suggest that, under reasonable metrics, this conclusion is not immediately obvious: the Hybrid CBM M_H above has a very similar concept fidelity to the non-leaky Independent CBM M_I baseline (M_H has a mean concept ROC-AUC of $93.65 \pm 0.22\%$ while M_I's is $93.90 \pm 0.11\%$). Moreover, as seen in Figure 6, the Hybrid CBM has (1) a significantly higher task fidelity

than M_I (i.e., $73.31 \pm 0.26\%$ vs $47.61 \pm 1.01\%$), and (2) a significantly higher area under the intervention curve than M_I (i.e., implying the Hybrid CBM is more intervenable). These intervention curves also reveal that M_H is likely allowing leakage in its bottleneck: its task accuracy when it receives several interventions is higher than the task accuracy achieved by the independent CBM when it is intervened on with all the ground-truth concepts (i.e., the rightmost point of the Independent CBM's intervention curve).

The results above suggest that, from the perspective of the evaluation metrics used to study CMs, a leaky model like M_H is likely to be considered "better" than a non-leaky model like M_I. This, however, is not aligned with common views of leakage. It is then natural to wonder under what perspective one can argue that a leaky CM is less interpretable than a non-leaky counterpart. For example, the traditional CM evaluation metrics used above may not detect whether a label predictor operating on leaky representations reasons about a task y using concepts the same way an expert would, a common argument against leakage. Therefore, could we somehow evaluate this property in a different manner?

As the label predictors for M_H and M_I are linear models, we can get a better view of this question by looking at how a leaky CM's task predictor weighs concepts C to predict Y and compare it to how a model we consider "interpretable", such as M_I, performs the same task. Let the label predictor of M_A, for $A \in \{H, I\}$, be $f_A(\hat{\mathbf{c}}) = \text{Softmax}(W_A\hat{\mathbf{c}} + b_A)$, where $W_A \in \mathbb{R}^{L \times \dim(\text{Dom}(f_A))}$. Moreover, let $T_y(M_A)$ be the set of top-5 weights in the y-th row of W_A. By looking at the label predictors of M_H and M_I, we see that, on average, the size of $O = |T_y(M_H) \cap T_y(M_I)|$ is 4.31 ± 0.86. In other words, the set of top-5 most important concepts used by M_I's label predictor to predict a given class y is, on average, almost the same as the set of top-5 most important bottleneck *activations* used by M_H when predicting the same class. To place this within a reasonable frame of reference: the overlap of $T_y(M_I)$ for two differently-seeded Independent CBMs is 4.45 ± 0.75, not statistically different from O. In contrast, the same value when using an ill-conditioned Hybrid CBM with $\lambda_{\text{int}} = 0$ is 2.01 ± 0.21, which is significantly lower than O. Given that the set of potential activations M_H has in its bottleneck is much larger than the set of ground-truth concepts ($(k + k') = 522 > 22 = k$), these results are not only surprising but also a strong indication that M_H's use of concepts is similar to that of M_I, a non-leaky model. Therefore, if models such as M_I are held to be interpretable, why should we not do the same for well-conditioned leaky models like M_H?

Altogether, the simple experiments above suggest that it is unclear under which definition of interpretability leakage leads to less interpretable models. As most interpretability concerns about leakage fall in this ill-specified setup, they are therefore unfalsifiable.

Partial explanations are useful explanations Finally, even if leakage may lead to models that use information not in C to predict Y, they are still *useful*: they can still at least *partially* explain their predictions based on a potentially incomplete set of concepts C, concepts which experts themselves also use to explain, sometimes also partially, their reasoning. Moreover, as seen in Section 6.1, leakage does not preclude intervenability, enabling experts to provide test-time concept-based feedback to leaky but well-conditioned CMs (e.g., CMs that minimize $\mathcal{L}_{\text{int}}$). In the common setting where C is incomplete, the alternative intervenable model would be to use a perfectly non-leaky CM that, at best, has the same power to partially explain its outputs at the cost of a severe drop in its task fidelity. Given a choice between these two options, we struggle to see an argument for selecting the non-leaky CM over the more accurate leaky CM.

7. Conclusion and Call to Action

This paper argued that the prevailing view of information leakage in CMs as a purely negative phenomenon is, in many cases, ill-posed and counterproductive. Specifically, we made our case by showing, through a set of simple experiments and evidence from prior work, that the conventional claims that leakage leads to CMs that are not intervenable or interpretable are, at best, inconclusive. In contrast, we argued that there are good reasons to want some leakage: in realistic concept-incomplete settings, eradicating leakage will only preclude CMs from remaining accurate, thereby making these models impractical. Hence, we showed that minimizing the CM's task loss when all concepts are intervened can lead to leakage that is properly *localized* and *sufficient* to overcome incompleteness. Such a simple regularizer, applicable to all CMs, enables even models previously thought to be unintervenable to remain accurate, intervenable, and interpretable (as measured by their concept accuracies and concept-to-task weights).

Taken together, we hope that the arguments and evidence put forth in this work naturally form a call for the CM community to (1) see past the goal of *entirely* eliminating leakage and instead explore ways to better *control* it, (2) evaluate new CMs on incomplete settings, reporting intervenability under such conditions and providing evidence for how the CM performs as the degree of incompleteness varies, (3) avoid broad claims that leakage or any other property of interpretable models *necessarily* destroys interpretability without specifying falsifiable criteria, and (4) separate CM-specific questions and phenomena (e.g., information leakage) from well-studied, yet more general failure modes of statistical learning (e.g., shortcut learning).

References

Almudévar, A., Hernández-Lobato, J. M., and Ortega, A. There was never a bottleneck in concept bottleneck models. *arXiv preprint arXiv:2506.04877*, 2025.

Alvarez Melis, D. and Jaakkola, T. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

Aysel, H. I., Cai, X., and Prugel-Bennett, A. Concept-based explainable artificial intelligence: Metrics and benchmarks. *arXiv preprint arXiv:2501.19271*, 2025.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Bjorck, N., Gomes, C. P., Selman, B., and Weinberger, K. Q. Understanding batch normalization. *Advances in neural information processing systems*, 31, 2018.

Chauhan, K., Tiwari, R., Freyberg, J., Shenoy, P., and Dvijotham, K. Interactive concept bottleneck models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37 (no. 5), pp. 5948–5955, 2023.

Collins, K. M., Barker, M., Espinosa Zarlenga, M., Raman, N., Bhatt, U., Jamnik, M., Sucholutsky, I., Weller, A., and Dvijotham, K. Human uncertainty in concept-based ai systems. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 869–889, 2023.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Espinosa Zarlenga, M., Pietro, B., Gabriele, C., Giuseppe, M., Giannini, F., Diligenti, M., Zohreh, S., Frederic, P., Melacci, S., Adrian, W., Lio, P., and Jamnik, M. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21400–21413. Curran Associates, Inc., 2022.

Espinosa Zarlenga, M., Barbiero, P., Shams, Z., Kazhdan, D., Bhatt, U., Weller, A., and Jamnik, M. Towards robust metrics for concept representation evaluation. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023a. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i10.26392. URL https://doi.org/10.1609/aaai.v37i10.26392.

Espinosa Zarlenga, M., Collins, K., Dvijotham, K., Weller, A., Shams, Z., and Jamnik, M. Learning to receive help: Intervention-aware concept embedding models. *Advances in Neural Information Processing Systems*, 36, 2023b.

Espinosa Zarlenga, M., Gabriele, D., Pietro, B., Shams, Z., and Jamnik, M. Avoiding leakage poisoning: Concept interventions under distribution shifts. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2025.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Havasi, M., Parbhoo, S., and Doshi-Velez, F. Addressing leakage in concept bottleneck models. In *Advances in Neural Information Processing Systems*, 2022.

Heidemann, L., Monnet, M., and Roscher, K. Concept correlation and its effects on concept-based models. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*, pp. 4780–4788, 2023.

Hu, L., Huang, T., Xie, H., Gong, X., Ren, C., Hu, Z., Yu, L., Ma, P., and Wang, D. Semi-supervised concept bottleneck models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2110–2119, 2025.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Jain, S. and Wallace, B. C. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, 2019.

Kim, E., Jung, D., Park, S., Kim, S., and Yoon, S. Probabilistic concept bottleneck models. In *International Conference on Machine Learning*, pp. 16521–16540. PMLR, 2023.

Kim, M. P., Ghorbani, A., and Zou, J. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 247–254, 2019.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Laguna, S., Marcinkevičs, R., Vandenhirtz, M., and Vogt, J. Beyond concept bottleneck models: How to make black boxes intervenable? *Advances in neural information processing systems*, 37:85006–85044, 2024.

Lipton, Z. C. and Steinhardt, J. Troubling trends in machine learning scholarship: Some ml papers suffer from flaws that could mislead the public and stymie future research. *Queue*, 17(1):45–77, 2019.

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

Makonnen, M., Vandenhirtz, M., Laguna, S., and Vogt, J. E. Measuring leakage in concept-based methods: An information theoretic approach. *arXiv preprint arXiv:2504.09459*, 2025.

Marconato, E., Passerini, A., and Teso, S. GlanceNets: Interpretable, leak-proof concept-based models. *Advances in Neural Information Processing Systems*, 35:21212–21227, 2022.

Marconato, E., Teso, S., Vergari, A., and Passerini, A. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. *Advances in Neural Information Processing Systems*, 36, 2024.

Margeloiu, A., Ashman, M., Bhatt, U., Chen, Y., Jamnik, M., and Weller, A. Do concept bottleneck models learn as intended? *arXiv preprint arXiv:2105.04289*, 2021.

Nussberger, A.-M., Luo, L., Celis, L. E., and Crockett, M. J. Public attitudes value interpretability but prioritize accuracy in artificial intelligence. *Nature communications*, 13 (1):5821, 2022.

Oikarinen, T., Das, S., Nguyen, L. M., and Weng, T.-W. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023.

Papenmeier, A., Englebienne, G., and Seifert, C. How model accuracy and explanation fidelity influence user trust. *arXiv preprint arXiv:1907.12652*, 2019.

Parisini, E., Chakraborti, T., Harbron, C., MacArthur, B. D., and Banerji, C. R. Leakage and interpretability in concept-based models. *arXiv preprint arXiv:2504.14094*, 2025.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. *PyTorch: An imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Penaloza, E., Zhang, T. H., Charlin, L., and Espinosa Zarlenga, M. Preference optimization for concept bottleneck models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2025.

Pérez, G. V., Louis, A. A., and Camargo, C. Q. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Pessach, D. and Shmueli, E. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

Poeta, E., Ciravegna, G., Pastor, E., Cerquitelli, T., and Baralis, E. Concept-based explainable artificial intelligence: A survey. *ACM Computing Surveys*, 2023.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Pugnana, A., Massidda, R., Giannini, F., Barbiero, P., Zarlenga, M. E., Pellungrini, R., Dominici, G., Giannotti, F., and Bacciu, D. Deferring concept bottleneck models: Learning to defer interventions to inaccurate experts. *arXiv preprint arXiv:2503.16199*, 2025.

Ragkousis, A. and Parbhoo, S. Tree-based leakage inspection and control in concept bottleneck models. *arXiv preprint arXiv:2410.06352*, 2024.

Raman, N., Espinosa Zarlenga, M., Heo, J., and Jamnik, M. Do concept bottleneck models respect localities? *Transactions on Machine Learning Research*, 2025.

Rao, S., Mahajan, S., Böhle, M., and Schiele, B. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *European Conference on Computer Vision*, pp. 444–461. Springer, 2024.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts:

On the importance of regularization for worst-case generalization. *International Conference on Learning Representations (ICLR)*, 2020.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

Sawada, Y. and Nakamura, K. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10: 41758–41765, 2022.

Serrano, S. and Smith, N. A. Is attention interpretable? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1282. URL https://aclanthology.org/P19-1282/.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33: 9573–9585, 2020.

Sheth, I. and Ebrahimi Kahou, S. Auxiliary losses for learning generalizable concept-based models. *Advances in Neural Information Processing Systems*, 36, 2024.

Shin, S., Jo, Y., Ahn, S., and Lee, N. A closer look at the intervention procedure of concept bottleneck models. In *International Conference on Machine Learning*, pp. 31504–31520. PMLR, 2023.

Sinha, S. and Zhang, A. A comprehensive survey on the risks and limitations of concept-based models. *arXiv preprint arXiv:2506.04237*, 2025.

Sohoni, N., Dunnmon, J., Angus, G., Gu, A., and Ré, C. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

Steinmann, D., Stammer, W., Friedrich, F., and Kersting, K. Learning to intervene on concept bottlenecks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Sun, A., Yuan, Y., Ma, P., and Wang, S. Eliminating information leakage in hard concept bottleneck models with supervised, hierarchical concept learning. *arXiv preprint arXiv:2402.05945*, 2024.

Vandenhirtz, M., Laguna, S., Marcinkevičs, R., and Vogt, J. Stochastic concept bottleneck models. *Advances in Neural Information Processing Systems*, 37:51787–51810, 2024.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wiegreffe, S. and Pinter, Y. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, 2019.

Wilson, A. G. Deep learning is not so mysterious or different. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2025.

Xu, X., Qin, Y., Mi, L., Wang, H., and Li, X. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. *International Conference on Learning Representations (ICLR)*, 2024.

Yang, Y., Panagopoulou, A., Zhou, S., Jin, D., Callison-Burch, C., and 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.

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

Yuksekgonul, M., Wang, M., and Zou, J. Post-hoc concept bottleneck models. In *ICLR 2022 Workshop on PAIR2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2023.

A. Proof of Sufficiency Optimization

Below we formalize the claim made in Section 5.1 that, under a reasonable set of assumptions, minimizing the task loss when all concepts are intervened is equivalent to achieving *sufficiency* in the learned concept representations $\hat{C}$. For this, we first notice that, due to the data processing inequality, it must be the case that, for localized concept representations $\hat{C} = (\bar{C}, R)$, we must have $I(Y; R \mid C) \leq I(Y; X \mid C)$, since R is a component of $\hat{C}$, which is in turn a function of X. Therefore, to achieve sufficiency, which we defined as $I(Y; R \mid C) \approx I(Y; X \mid C)$, what we want is to maximize $I(Y; R \mid C)$ so that it approaches its upper bound $I(Y; X \mid C)$. Hence, under this perspective, the following theorem says that this maximization is possible by minimizing the task loss when we intervene on all concepts of the bottleneck:

Theorem A.1 (Sufficiency Optimization)**.** *Let $M = \big(g(\cdot\,;\theta_g), f(\cdot\,;\theta_f)\big)$ be a CBM and $(X, C, Y) \sim \mathcal{P}$ be the data-generating distribution. Let $Z := g(X \mid C := C\,;\theta_g)$ denote the random variable corresponding to the* fully intervened *bottleneck. Assume concept interventions are localized so that $Z \equiv (C, R)$ for some residual random variable R. If $\mathcal{L}_y$ is the cross-entropy loss and f is well-specified for $\mathbb{P}(Y \mid Z)$, then:*

$$\underset{\theta_g, \theta_f}{\arg\min} \underbrace{\mathbb{E}_{(\mathbf{x},\mathbf{c},y)\sim\mathcal{P}}\left[\mathcal{L}_y\left(f\Big(g\big(\mathbf{x} \mid C := \mathbf{c}\big)\Big), y\right)\right]}_{\textit{All-concepts-intervened task loss}} \equiv \underset{\theta_g}{\arg\max}\, \underbrace{I\big(Y; R \mid C\big)}_{\textit{Sufficiency}} \tag{1}$$

Proof. Fix θ_g and let $q_{\theta_f}(y \mid z)$ denote the conditional distribution induced by $f(\cdot\,;\theta_f)$. By the standard cross-entropy decomposition, we must have that the LHS of Equation 1 becomes:

$$\mathbb{E}\Big[-\log q_{\theta_f}(Y \mid Z)\Big] = H(Y \mid Z) + \mathbb{E}\Big[D_{\mathrm{KL}}\Big(\mathbb{P}(Y \mid Z) \,\|\, q_{\theta_f}(Y \mid Z)\Big)\Big]$$

Under well-specification, there exists a set of parameters $\theta_f^\star$ such that $q_{\theta_f^\star}(Y \mid Z) = \mathbb{P}(Y \mid Z)$, making the KL divergence term in the expression above zero. Hence, for any fixed θ_g we must have that $\min_{\theta_f}\ \mathbb{E}\big[\mathcal{L}_y(f(Z;\theta_f), Y)\big] = H(Y \mid Z)$.

This implies that the LHS of Equation 1 can be equivalently written as follows:

$$\begin{aligned}
\underset{\theta_g, \theta_f}{\arg\min}\, \mathbb{E}\big[\mathcal{L}_y\big(f(Z;\theta_f), Y\big)\big] &\equiv \underset{\theta_g}{\arg\min}\ H(Y \mid Z) \\
&= \underset{\theta_g}{\arg\min}\, H(Y \mid C, R) && \text{(By our localization assumption)} \\
&= \underset{\theta_g}{\arg\min}\, \Big(H(Y \mid C) - I(Y; R \mid C)\Big) \\
&\equiv \underset{\theta_g}{\arg\min}\, -I(Y; R \mid C) && \big(\text{As } H(Y|C) \text{ does not depend on } \theta_g\big) \\
&\equiv \underset{\theta_g}{\arg\max}\, I(Y; R \mid C)
\end{aligned}$$

This directly yields the equivalency we wished to show in Equation 1. □

B. Experimental Details

In Sections 5 and 6, we made use of some simple experiments to build our case. In this appendix, we discuss details for recreating these results.

B.1. Datasets and Tasks

General Setup In all of our illustrative experiments, we use the CUB-200 dataset (Wah et al., 2011), preprocessed by Koh et al. (2020). Each sample $(\mathbf{x}^{(j)}, \mathbf{c}^{(j)}, y^{(j)})$ in this original dataset corresponds to a normalized RGB image $\mathbf{x}^{(j)} \in [0, 1]^{3\times 299\times 299}$ of a bird that is annotated with one of $L = 200$ bird species $y^{(j)}$. Each sample comes with a set of 112 binary concept annotations $\mathbf{c}^{(j)} \in \{0, 1\}^{112}$ representing visual attributes of the bird (e.g., `black_wing`, `medium_sized`, etc.). By construction, the concept set in this dataset provides a complete and perfect description of the downstream label y (as each task label in the CUB formulation by Koh et al. (2020) is assigned its own "concept profile"). These concepts are grouped into 28 mutually exclusive concept groups (e.g., `wing_color`, `size`). Therefore, when intervening on models trained on this dataset, we apply the intervention to all concepts within the same group simultaneously. Finally, we use the train-validation-test splits from (Koh et al., 2020) and randomly flip and crop images during training.

Used Tasks To recreate the conditions of an incomplete training set, following Espinosa Zarlenga et al. (2023b), we subsample the training concept annotations of the CUB dataset described above to construct an incomplete task we call *CUB-Incomplete*. Specifically, as in (Espinosa Zarlenga et al., 2023b), we subsample, at random, 25% of the 28 original concept annotation groups, yielding an *incomplete version of CUB* whose concept vectors $\mathbf{c}$ are formed by 7 concept groups that have, together, a total of $k = 22$ concepts.

Unless otherwise stated or clear from the context (e.g., the results of Figure 3), all experiments are conducted on CUB-Incomplete. The only exception is the experiments in App. C, which use the concept-complete version of CUB to explore how incompleteness affects the intervenability of leakage-enabling models.

B.2. Baselines

In our illustrative experiments, we train and evaluate the following baselines:

1. **Concept Bottleneck Models** (Koh et al., 2020): We train CBMs with sigmoidal (i.e., $\hat{\mathbf{c}}_i \in [0, 1]$ and $s(\hat{\mathbf{c}}) = \hat{\mathbf{c}}$) and logit (i.e., $\hat{\mathbf{c}}_i \in \mathbb{R}$ and $s(\hat{\mathbf{c}}) = \sigma(\hat{\mathbf{c}})$) bottlenecks. These models are trained *independently*, *sequentially*, and *jointly* (minimizing the combination of $\mathcal{L}_y(\cdot) + \lambda_c \mathcal{L}_c(\cdot)$). In our experiments, we use the independently trained sigmoidal CBM (*Independent CBM*) as a minimal-leakage baseline, since its label predictor is trained on ground-truth concepts.

2. **Hybrid CBMs** (Mahinpei et al., 2021): We include Hybrid CBMs as an example of a leaky CM that introduces a bypass channel in its bottleneck. In practice, we train these models by extending a sigmoidal CBM with k' additional unconstrained activations $\psi(\mathbf{x})$ which are added to the bottleneck. This model is trained by minimizing a joint loss that weights the concept loss term using a hyperparameter λ_c.

3. **Black Box (Baseline)**: We implemented a Black Box DNN baseline using a Hybrid CBM with $k' = 500$ additional neurons in its bottleneck whose concept loss weight during training is set to 0 (i.e, $\lambda_c = 0$). This ensures the Black Box model is given the same capacity as competing models it is compared against (e.g., equivalent Hybrid CBMs).

4. **Concept Embedding Models (CEMs)** (Espinosa Zarlenga et al., 2022): We include, as an example of a highly leaky baseline, a CEM that represents concepts as a mixture of two dynamic embeddings with dimensionality $m = 16$. Unless explicitly stated otherwise, all CEMs are trained using RandInt, as in their original work. That is, during training, we intervene on a concept with probability $p_{\text{int}} = 0.25$.

5. **Probabilistic CBMs (ProbCBMs)** (Kim et al., 2023): As a second embedding-based baseline, we use ProbCBMs, which represent concepts with probabilistic embeddings of size $m = 16$. As suggested by the authors, we train ProbCBMs with RandInt, intervening on a concept during training with probability $p_{\text{int}} = 0.5$.

6. **Post-hoc CBMs (PCBMs)** (Yuksekgonul et al., 2023): Finally, we include as baselines PCBMs trained from the Black Box model baseline. We train both a standard Post-hoc CBM, whose leakage is relatively small, and a Residual Post-hoc CBM, which introduces an explicit residual channel that enables higher task accuracy in incompleteness at the cost of higher leakage.

Model Selection and Implementation All models are implemented in PyTorch (Paszke et al., 2019) and trained using the same backbone architectures, optimizers, and training schedules used in Espinosa Zarlenga et al. (2025). Specifically, we made use of the publicly available code[1] from (Espinosa Zarlenga et al., 2025) to access implementations for all the baselines used in our illustrative examples. To better communicate our position, we avoid fine-tuning baselines and expensive training procedures by reusing the exact hyperparameters and model selection reported in (Espinosa Zarlenga et al., 2025) for each baseline. This means that all baselines were built by using an ImageNet-pretrained ResNet-18 as a backbone for the concept encoder $g(\cdot)$ and a linear layer for the label predictor $f(\cdot)$. For further details on the hyperparameters used for each baseline in our CUB experiments, we refer the reader to Appendix D of (Espinosa Zarlenga et al., 2025).

Sufficiency Regularizer When introducing the sufficiency regularizer $\mathcal{L}_{\text{int}}$, we add it to the training objective with weight $\lambda_{\text{int}} \in \{0, 1, 5, 10\}$ while keeping all other hyperparameters and training configurations of the underlying model constant. The value of λ_{int} is then selected based on the area under the intervention curve on the validation set. This procedure yields $\lambda_{\text{int}} = 10$ for most models with the exception of Hybrid CBMs and CEMs, where the procedure selected $\lambda_{\text{int}} = 5$.

[1] Found at https://github.com/mateoespinosa/cem.

When we train the Black Box baseline model with the regularizer $\mathcal{L}_{\text{int}}$, we select k neurons in the output of the penultimate layer and intervene on those neurons as we would for a traditional sigmoidal CBM (i.e., setting the output of the neuron to $\hat{\mathbf{c}}_i := \mathbf{c}_i$). Furthermore, when using this regularizer for models whose concept representations are unbounded scalars (e.g., logit CBMs (Koh et al., 2020)), we intervene on concepts by setting their respective representations to a high logit value when $\mathbf{c}_i = 1$ (e.g., we use $+5$) and to a low logit value when $\mathbf{c}_i = 0$ (e.g., we use -5).

C. Completeness Experiments

To disentangle the effects of information leakage from those of concept incompleteness, we replicate the intervention experiments of Figure 4 on a complete version of CUB (using the same setup as Koh et al. (2020)).

Our results, summarized in Figure 7, suggest that, when the concept set is complete, all leakage-enabling models remain intervenable, including those that were seen to lose intervenability when the concept set was incomplete (Figure 4). Nevertheless, we observe that even in this instance, the degree of intervenability of the leaky CMs in the right panel of Figure 7 is not the same: for example, although CEMs without RandInt do increase their task accuracies as they are intervened on, the effect is minimal compared to that of other models. This implies that it is likely that incompleteness can lead to several leaky models to become unintervenable, but its effects on leaky CMs can vary.

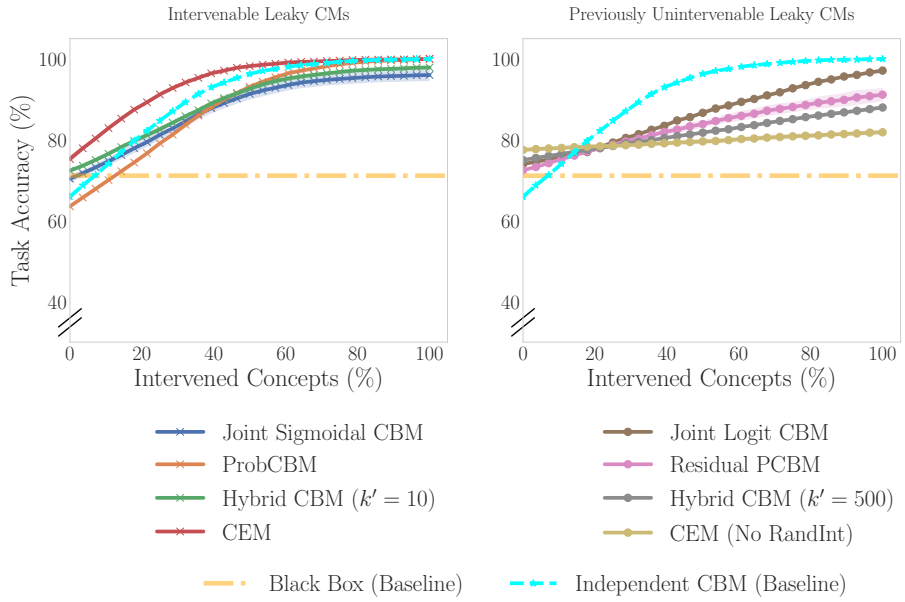

Figure 7. Intervention curves for leakage-enabling CMs on a complete version of CUB. **(left)** Leakage-enabling CMs that were intervenable in the incomplete version of CUB remain highly intervenable here. **(right)** Leakage-enabling CMs that were previously unintervenable in the incomplete version of CUB (Figure 4) can become intervenable when the concept set is complete, although to a lesser degree than the non-leaky baseline (Independent CBMs) variants.

D. Additional Experiments With Sufficiency Regularizer

We further evaluate the effect of the sufficiency regularizer $\mathcal{L}_{\text{int}}$ on models that were previously unintervenable. Specifically, we apply $\mathcal{L}_{\text{int}}$ to Joint Sigmoidal CBMs, Joint Logit CBMs, PCBMs, Residual PCBMs, large Hybrid CBMs ($k' = 500$), and CEMs trained without RandInt.

Figure 8 shows that introducing $\mathcal{L}_{\text{int}}$ consistently improves intervenability across all models, often without degrading task or concept fidelity. These results support the claim that proper conditioning via sufficiency optimization can recover intervenability even in highly leaky architectures.

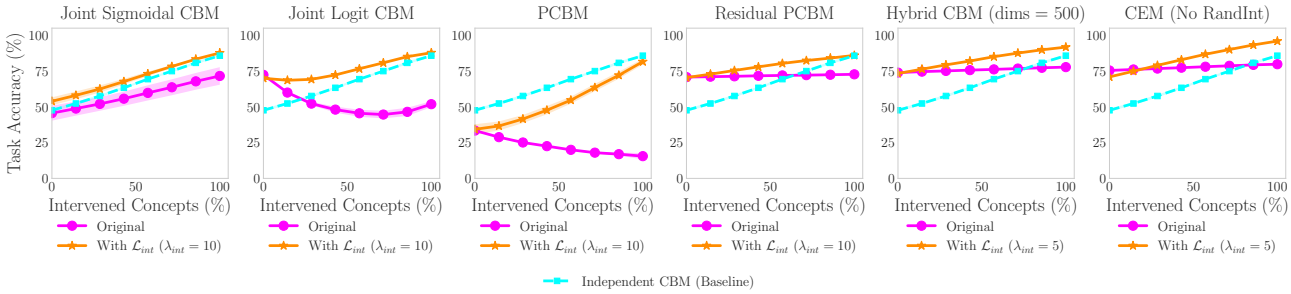

Figure 8. Effect of applying the sufficiency regularizer $\mathcal{L}_{\text{int}}$ to previously unintervenable leakage-enabling models.