RESIDUAL STOCHASTIC CONCEPT BOTTLENECK MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept-based models (CMs) learn human-interpretable intermediate representations that enable users to understand the model's reasoning and steer its predictions through test-time concept interventions. However, in realistic settings, their performance can be limited by concept incompleteness, where the training concepts fail to capture all task-relevant information. Adding an unconstrained residual pathway to the model's predictive process can overcome the consequences of incompleteness. However, this introduces an inscrutable, leak-prone prediction pathway that may weaken concept interventions, as these often leave residual variables unchanged. To address this limitation, we introduce Residual Stochastic Concept Bottleneck Models (Res-SCBMs), a family of concept-based models that jointly model concept and residual variables through a multivariate Gaussian distribution. Res-SCBMs learn a covariance structure that explicitly captures dependencies between the known concepts and learnt residual variables, allowing interventions to propagate to statistically dependent residual variables through Gaussian conditioning. This extends the intervention capabilities of previous CMs without requiring concept completeness during training. Our results show that, across multiple concept-incomplete benchmarks, Res-SCBMs improve predictive performance over competing baselines while retaining effective test-time concept interventions.

1 INTRODUCTION

As deep learning models are increasingly deployed across a wide range of domains, understanding the reasoning underlying their predictions has become more important. Concept-based models (CMs) have emerged as a promising approach to incorporate interpretability by design (Koh et al., 2020; Espinosa Zarlenga et al., 2022; Yuksekgonul et al., 2023; Vandenhirtz et al., 2024). Rather than predicting the task output $\hat{y}$ directly from the input $\boldsymbol{x}$, CMs first predict a set of human-interpretable concept representations $\hat{\boldsymbol{c}} = g(\boldsymbol{x})$, which are then used to produce the final prediction $\hat{y} = f(\hat{\boldsymbol{c}})$. These intermediate representations let users inspect the concepts a model uses in its reasoning and, importantly, intervene on intermediate concepts at test time. Through these interventions, users can correct mispredicted concepts in the model's intermediate reasoning and communicate high-level hints as concepts (Laguna et al., 2024; Chauhan et al., 2023; Espinosa Zarlenga et al., 2023).

A central challenge for CMs is *concept incompleteness* (Yeh et al., 2020; Mahinpei et al., 2021), where the predefined concept set fails to capture all information relevant to the downstream task. In practice, constructing a complete concept set can be difficult, and obtaining reliable concept annotations may require substantial effort, whether through manual labelling, domain expertise, or automated annotation pipelines (Oikarinen et al., 2023; Yuksekgonul et al., 2023). As a result, models constrained to operate solely through the concept bottleneck may suffer from limited predictive performance (Espinosa Zarlenga et al., 2022; Havasi et al., 2022; Yuksekgonul et al., 2023).

To address this limitation, several methods (Yuksekgonul et al., 2023; Shang et al., 2024) have explored augmenting an interpretable concept representation with additional residual representations designed to capture task-relevant information not explained by the predefined concepts. However, introducing a residual channel creates a tension between predictive capacity and intervenability (Espinosa Zarlenga et al., 2022; Parisini et al., 2025). To address concept incompleteness, the residual

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

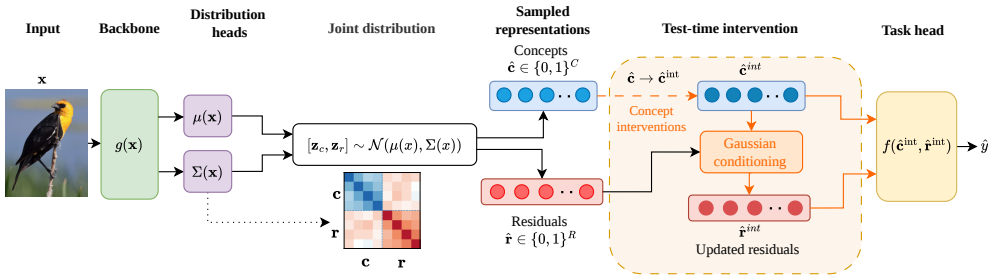


Figure 1: **Overview of Res-SCBMs.** Concept and residual variables are jointly modelled through a multivariate Gaussian distribution. At test time, concept interventions propagate to statistically dependent residual variables through Gaussian conditioning, updating the residual representation before prediction.

representation must carry task-relevant information. Yet, because this provides an alternative pathway to the downstream predictor, concept corrections may have less influence on the final prediction (Zarlenga, 2026; Parisini et al., 2025). Existing work has shown that additional latent representations can coexist with effective concept interventions (Havasi et al., 2022; Zarlenga, 2026), but these representations are typically treated as separate information pathways. In particular, intervening on a predefined concept does not explicitly update the residual representation to reflect dependencies between the two, potentially leaving correlated residual information unchanged after the intervention. A key challenge is therefore not only to retain intervenability with a residual channel, but to ensure that the residual representation itself responds consistently when concepts are corrected.

To address this, we introduce *Residual Stochastic Concept Bottleneck Models* (Res-SCBMs), a concept-based architecture that introduces a residual channel designed to capture task-relevant information missing from the predefined concept set while preserving effective concept interventions (Figure 1). Rather than modelling concepts and residuals independently, Res-SCBMs draw inspiration from Stochastic Concept Bottleneck Models (SCBMs) (Vandenhirtz et al., 2024) and represent concepts and residuals jointly via a multivariate Gaussian distribution, whose covariance structure captures dependencies between the two representations. Consequently, concept interventions, both on known concepts and on the residual variables, can propagate through Gaussian conditioning to other statistically dependent variables, which may include residuals. We achieve this by introducing an intervention-aware training objective, inspired by Zarlenga (2026), that encourages the model to retain predictive performance after concept correction. Our results suggest that Res-SCBMs recover the predictive performance lost by competing approaches such as SCBMs under concept incompleteness, while retaining effective concept interventions.

Therefore, our main contributions can be summarised as follows:

- Res-SCBMs, a probabilistic concept-residual architecture that retains task-relevant information under concept incompleteness and propagates concept interventions to statistically dependent residual variables.
- An empirical evaluation demonstrating that Res-SCBMs improve predictive performance on concept-incomplete tasks while maintaining effective concept interventions across different residual dimensionalities.

2 Background & Related Work

Concept Bottleneck Models. Most CMs can be expressed as a Concept Bottleneck Model (CBM) (Koh et al., 2020). A CBM consists of two functions, a concept predictor $g(\mathbf{x})$ and a task predictor $f(\hat{\mathbf{c}})$, which are typically parameterised by neural networks. Given an input $\boldsymbol{x} \in \mathbb{R}^D$, the concept predictor maps $\boldsymbol{x}$ to a vector of C human-interpretable concepts, $\hat{\boldsymbol{c}} = g(\boldsymbol{x}) \in [0,1]^C$. The task predictor then takes the predicted concept representation as input and produces the final task prediction, $\hat{y} = f(\hat{\boldsymbol{c}})$. CBMs are trained on datasets of the form $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{c}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^D$ are the raw input features, $\boldsymbol{c}_i \in \{0,1\}^C$ is the corresponding vector of human-interpretable concept

annotations, and $y_i \in \mathcal{Y}$ is the downstream task label. While these concept annotations are traditionally provided by human annotators, recent variants obtain concepts automatically through label-free or unsupervised concept discovery pipelines (Oikarinen et al., 2023; Schrodi et al., 2024). Several extensions of the standard CBM formulation have been proposed, including post-hoc CBMs, which construct concept representations from pretrained models (Yuksekgonul et al., 2023), probabilistic CBMs (Vandenhirtz et al., 2024), models with richer concept representations (Espinosa Zarlenga et al., 2022), and hybrid or residual CBMs that introduce additional capacity beyond the predefined concept bottleneck (Mahinpei et al., 2021; Shang et al., 2024).

Modelling concept dependencies. A limitation of conventional CBMs is that they do not explicitly model dependencies between concepts (Vandenhirtz et al., 2024; Dominici et al., 2025). Such dependencies naturally arise in real-world concept sets. For example, `CUB` (Wah et al., 2011), a popular benchmark for CMs, contains attributes describing related regions of a bird, including breast colour and underparts colour, which may exhibit strong statistical dependencies. Ignoring these dependencies can make test-time interventions less efficient (Singhi et al., 2025), as correcting one concept does not automatically update other related concepts, potentially requiring users to correct them individually. Several approaches account for such dependencies. ECBMs (Xu et al., 2024) model dependencies between inputs, concepts, and task labels through a joint energy-based formulation, allowing concept corrections to affect related concepts. Causal approaches instead explicitly model directed relationships between concepts. For example, Causal Concept Graph Models (Dominici et al., 2025) organise concepts according to a causal graph, allowing interventions to propagate through the modelled causal structure. Stochastic CBMs (SCBMs) (Vandenhirtz et al., 2024) instead model the concepts jointly using a multivariate Gaussian distribution. Its covariance structure captures dependencies between concepts, allowing interventions on one concept to propagate to correlated concepts through Gaussian conditioning.

Residual capacity in CBMs. To address concept incompleteness, several approaches have explored increasing the information capacity of concept-based models beyond the predefined concept set. Hybrid CBMs (Mahinpei et al., 2021) augment the interpretable concept bottleneck with an additional residual representation, allowing task-relevant information not captured by the concepts to bypass the bottleneck. Similarly, Hybrid Post-hoc Concept Bottleneck Models (PCBM-h) (Yuksekgonul et al., 2023) introduce a residual predictor that operates on the original backbone representation and adds its output to the concept-based prediction, allowing information missing from the concept bank to contribute to the downstream task. Incremental Residual Concept Bottleneck Models (Shang et al., 2024) introduce residual directions to capture missing information and subsequently attempt to associate them with additional human-interpretable concepts. In contrast, Concept Embedding Models (CEMs) (Espinosa Zarlenga et al., 2022) increase the capacity of the concept bottleneck itself by representing each concept with a high-dimensional embedding rather than a single scalar value. This richer representation retains task-relevant information beyond binary concept labels while preserving a concept-structured bottleneck. Increasing information capacity beyond the predefined concepts can increase the risk of leakage, potentially reducing both interpretability and intervenability (Parisini et al., 2025). Rather than treating all leakage as undesirable, Zarlenga (2026) distinguishes between harmful and benign leakage, showing that benign leakage can retain additional task-relevant information while preserving effective concept interventions.

Despite these advances, existing work has largely considered these two directions separately, either modelling dependencies within the concept representation or introducing additional capacity for information missing from the concept set. Existing methods do not jointly model concepts and residual variables in a way that allows concept interventions to propagate to statistically dependent residual variables. Our work addresses this limitation by jointly modelling concepts and residual variables, allowing interventions to propagate between them, while an intervention-aware training objective encourages the model to remain responsive to concept corrections.

3 Residual Stochastic CBMs

We introduce Residual Stochastic Concept Bottleneck Models (Res-SCBMs), a class of CMs that extend SCBMs with residual representations to capture task-relevant information not encoded by the

observed concepts. Concepts and residuals are modelled jointly through a multivariate Gaussian, whose covariance structure captures dependencies between the two representations (see Figure 1).

3.1 Model Formulation

Given an input $\boldsymbol{x} \in \mathbb{R}^D$, a Res-SCBM models predicted concept logits $\boldsymbol{z}_c \in \mathbb{R}^C$ and residual logits $\boldsymbol{z}_r \in \mathbb{R}^R$ through a joint multivariate Gaussian distribution parameterised as:

$$\begin{bmatrix} \boldsymbol{z}_c \\ \boldsymbol{z}_r \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_c(\boldsymbol{x}) \\ \boldsymbol{\mu}_r(\boldsymbol{x}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{cc}(\boldsymbol{x}) & \boldsymbol{\Sigma}_{cr}(\boldsymbol{x}) \\ \boldsymbol{\Sigma}_{cr}(\boldsymbol{x})^\top & \boldsymbol{\Sigma}_{rr}(\boldsymbol{x}) \end{bmatrix} \right). \tag{1}$$

Here, the cross-covariance $\boldsymbol{\Sigma}_{cr}$ captures statistical dependencies between the concept and residual representations. For Res-SCBMs to be most effective, residual variables should be neither independent of the concepts nor merely redundant with them. If residual variables are independent of concepts, concept interventions induce little or no change in the residual representation through Gaussian conditioning. Conversely, if residual variables are too strongly dependent on the concepts, they may largely replicate information already captured by the concept representation. Ideally, the residual variables capture complementary task-relevant information missing from the observed concept set while retaining dependencies with the concepts that allow interventions to propagate to the residual representation.

Following Vandenhirtz et al. (2024), the sampled logits are mapped to stochastic binary concept and residual representations by applying a sigmoid followed by Bernoulli sampling:

$$\begin{bmatrix} \hat{\boldsymbol{c}} \\ \hat{\boldsymbol{r}} \end{bmatrix} \sim \text{Bernoulli}\left(\sigma\left(\begin{bmatrix} \boldsymbol{z}_c \\ \boldsymbol{z}_r \end{bmatrix} \right) \right) \in \{0,1\}^{C+R}. \tag{2}$$

In practice, we estimate gradients during optimisation using a straight-through estimator. The concept and residual representations are concatenated and passed jointly through the same task head, $\hat{y} = f([\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}])$, where $[\cdot, \cdot]$ denotes vector concatenation.

3.2 Concept interventions

Modelling concepts and residuals jointly is central to Res-SCBMs. The learned covariance structure allows an intervention on a concept to update not only correlated concepts, as in SCBMs (Vandenhirtz et al., 2024), but also residual variables that are statistically dependent on the intervened concept. Following the intervention procedure of SCBMs and logit-based CBMs (Koh et al., 2020), we map binary concept corrections using empirical percentiles of the predicted concept probabilities on the training set. A corrected concept with $c_i^{\text{int}} = 0$ is mapped to the 5th percentile, whereas $c_i^{\text{int}} = 1$ is mapped to the 95th percentile. The corresponding intervention logit is given by

$$p_i^{\text{int}} = (1 - c_i^{\text{int}}) q_{0.05}^{(i)} + c_i^{\text{int}} q_{0.95}^{(i)}, \qquad z_{c,i}^{\text{int}} = \sigma^{-1}\left(p_i^{\text{int}}\right), \tag{3}$$

where $q_{0.05}^{(i)}$ and $q_{0.95}^{(i)}$ denote the 5th and 95th percentiles of the empirical distribution of predicted probabilities for concept i over the training set.

Notice then that when all concepts are intervened on with corrected concept logits $\boldsymbol{z}_c^{\text{int}}$, the conditional distribution over the residual logits is given by:

$$\boldsymbol{z}_r \mid \boldsymbol{z}_c^{\text{int}} \sim \mathcal{N}\left(\boldsymbol{\mu}_r + \boldsymbol{\Sigma}_{rc}\boldsymbol{\Sigma}_{cc}^{-1}(\boldsymbol{z}_c^{\text{int}} - \boldsymbol{\mu}_c),\ \boldsymbol{\Sigma}_{rr} - \boldsymbol{\Sigma}_{rc}\boldsymbol{\Sigma}_{cc}^{-1}\boldsymbol{\Sigma}_{cr}\right). \tag{4}$$

Consequently, interventions propagate through the learned cross-covariance $\boldsymbol{\Sigma}_{rc}$, updating residual variables statistically dependent on the intervened concepts. The conditioned residual logits are subsequently binarised as in Eq. 2 and passed to the task head together with the corrected concepts.

3.3 Training Objective

We train Res-SCBM using a base objective $\mathcal{L}_{\text{base}}$ and introduce an additional intervention-aware loss $\mathcal{L}_{\text{int}}$ to encourage the model to remain responsive to concept corrections despite the additional residual capacity.

Base objective. We adopt the original SCBM training objective (Vandenhirtz et al., 2024) and extend it to the joint concept–residual representation. The resulting base objective is given by

$$\begin{aligned}\mathcal{L}_{\text{base}} = & -\log \frac{1}{M}\sum_{m=1}^{M}\exp\left(\sum_{i=1}^{C} -\text{BCE}\left(\boldsymbol{c}_i, \boldsymbol{p}_i^{(m)}\right)\right) \\ & + \lambda_1 \,\text{CE}\left(y, \frac{1}{M}\sum_{m=1}^{M} f_\psi\left([\hat{\boldsymbol{c}}^{(m)}, \hat{\boldsymbol{r}}^{(m)}]\right)\right) \\ & + \lambda_2 \frac{1}{(C+R)(C+R-1)}\sum_{i\neq j}\left|\left[\boldsymbol{\Sigma}(\boldsymbol{x})^{-1}\right]_{ij}\right|.\end{aligned} \tag{5}$$

Here, M is the number of Monte Carlo samples and $p_i^{(m)} = \sigma(z_{c,i}^{(m)})$ is the concept probability for sample m. The three terms correspond to the concept prediction loss, downstream task loss and precision loss, respectively. The precision loss regularises the off-diagonal entries of the precision matrix, encouraging a sparse conditional dependency structure and reducing overfitting to spurious dependencies (Vandenhirtz et al., 2024). We retain this regularisation for the joint concept and residual distribution.

Intervention-aware objective. While the residual channel can improve predictive performance under concept incompleteness, it introduces an alternative pathway from the input to the task prediction. If this pathway dominates the prediction, concept interventions may have little effect on the downstream output.

To encourage responsiveness to concept interventions, we introduce an auxiliary loss $\mathcal{L}_{\text{int}}$ inspired by the intervention objective of Zarlenga (2026). This objective penalises task mispredictions when all observed concepts are intervened on. Specifically, we replace the predicted concept values with their ground-truth values $\boldsymbol{c}$ and condition the joint Gaussian on the corresponding corrected concept logits. We then sample the updated residual representation $\hat{\boldsymbol{r}}_{\boldsymbol{c}}^{\text{cond}}$ from the resulting conditional distribution. The intervention-aware loss is defined as

$$\mathcal{L}_{\text{int}} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{c},y)\sim\mathcal{D}}\left[\ell_y\left(f_\psi([\boldsymbol{c}, \hat{\boldsymbol{r}}_{\boldsymbol{c}}^{\text{cond}}]), y\right)\right]. \tag{6}$$

The combined training objective is

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \lambda_{\text{int}}\mathcal{L}_{\text{int}}, \tag{7}$$

where λ_{int} controls the contribution of the intervention-aware loss.

By optimising task performance after concept correction, $\mathcal{L}_{\text{int}}$ encourages the model to use corrected concepts together with the conditioned residual representation, thus preserving the influence of concept interventions on the final prediction. It may further promote localisation of concept-specific information within the concept representation (Zarlenga, 2026), leaving the residual channel to capture complementary task-relevant information beyond the observed concepts.

4 Experimental setup

Datasets. We evaluate Res-SCBMs across four vision datasets: `CUB`, `CUB-Incomplete`, `AwA2`, and `AwA2-Incomplete`. `CUB` (Wah et al., 2011) contains 11,788 images from 200 bird species and provides 312 binary concepts per image. Following Koh et al. (2020), we retain a subset of 112 concepts. We treat `CUB` as concept-complete, as the concept set is perfectly predictive of the downstream target. To evaluate the models under concept incompleteness, we construct `CUB-Incomplete` following the concept-removal procedure of Espinosa Zarlenga et al. (2023), reducing the concept set from 112 to 22 concepts. In contrast, `AwA2` (Xian et al., 2019) contains 37,322 images from 50 animal classes, each described by 85 concepts. `AwA2` has task-specific concept profiles, implying it is a complete concept set. Hence, we similarly construct `AwA2-Incomplete` by following the concept-removal procedure of Espinosa Zarlenga et al. (2025), reducing the concept set from 85 concepts to an incomplete set with only 9 concepts. See Appendix A.1 for further details.

Baselines. We compare Res-SCBMs against (1) Hard CBMs (Havasi et al., 2022), an extension of the original CBM with binarised concept predictions, (2) SCBM (Vandenhirtz et al., 2024), the closest previous work to Res-SCBMs, and (3) CEMs (Espinosa Zarlenga et al., 2022), a family of leaky models also designed to address incompleteness. Hard CBMs and SCBMs serve as non-leaky baselines, as their binarised concept bottlenecks restrict downstream predictions to the predicted concept values. In contrast, CEM represents each concept with a high-dimensional embedding rather than a single scalar value, providing additional representational capacity. We select CEMs over Soft CBMs as our leaky baseline, given their explicit design to address concept incompleteness while supporting concept interventions. We additionally include a standard DNN as a black-box baseline, providing a reference for predictive performance without the constraints imposed by an interpretable concept bottleneck. Since all models use linear task heads, we also train a linear classifier on ground-truth concepts to assess how much predictive information is available in the concept representation alone. For comparability with our intervention-aware model, and following Espinosa Zarlenga et al. (2022), we train CEMs by randomly intervening on 25% of the concepts during training (a process the authors call *RandInt*). We retain *RandInt* rather than explicitly optimising $\mathcal{L}_{\text{int}}$, as Zarlenga (2026) show that both approaches can yield intervenable CEMs, with *RandInt* implicitly optimising a similar objective.

Metrics. We evaluate predictive performance using task accuracy and concept prediction performance using mean concept ROC-AUC. To evaluate intervenability, we measure task accuracy w.r.t. the fraction of corrected concepts (selected at random). This yields intervention curves that quantify how effectively improvements to the concept representation propagate to the downstream prediction.

Implementation Details. For all residual models, we use a residual dimension of 20 unless stated otherwise. For the training objectives of our models, as defined in Equations 5 and 7, we set $\lambda_1 = \lambda_2 = \lambda_{\text{int}} = 1$. We use amortised covariance parametrisation for SCBMs and Res-SCBMs and a random intervention policy for all models. Further implementation details are provided in Appendix A.3.

5 Results

We evaluate two variants: Res-SCBM w/o $\mathcal{L}_{\text{int}}$, trained using only the base objective, and Res-SCBM, which additionally incorporates the intervention-aware loss $\mathcal{L}_{\text{int}}$. We address four main research questions:

RQ1. Do our residual models improve predictive performance compared with models restricted to the concept representation?

RQ2. Do our residual models retain effective concept interventions despite the added capacity?

RQ3. Does intervention-aware training preserve intervenability as residual capacity increases?

RQ4. What do the learned concept–residual covariances reveal about the residual representation?

5.1 Predictive Performance (RQ1)

We first look at Res-SCBM's task accuracy across tasks when no interventions are performed. Our results summarised in Table 1, show that, on concept-complete tasks, Res-SCBM is slightly stronger than concept-based baselines, while on concept-incomplete tasks, Res-SCBM substantially outperforms non-leaky baselines (i.e., Hard CBMs and SCBMs). This latter improvement is achieved while maintaining a competitive, and sometimes better (e.g., on `CUB-Incomplete`), accuracy than high-capacity CEMs. We see this even when we train Res-SCBMs without $\mathcal{L}_{\text{int}}$: for example, on `AwA2-Incomplete`, both variants of Res-SCBM improve task accuracy by more than 23 percentage points over Hard CBM and SCBM, while performing comparably to CEMs. Crucially, both Res-SCBM variants outperform a linear classifier trained to map ground-truth observed concepts to the task labels on `AwA2-Incomplete`, suggesting that their predictions exploit task-relevant information beyond the incomplete concept set. Finally, we observe that Res-SCBMs are also competitive w.r.t. unconstrained DNNs: Res-SCBM outperforms the black-box DNN on both `CUB` tasks and performs similarly to both `AwA2` tasks. On the concept-complete datasets, the gains over existing CBMs are comparatively small, consistent with the observed concepts already containing sufficient information for the downstream task.

Table 1: Task accuracy (%) before concept interventions. Values are means over seeds, with standard deviations in subscript. Best/second-best concept-based results are bold/underlined. The DNN is reported as a predictive baseline, while the linear head uses ground-truth concepts and is reported as a reference. Residual models outperform non-leaky baselines under concept incompleteness.

Method	CUB	CUB Incomplete	AwA2	AwA2 Incomplete
DNN (black box)	$68.60_{\pm 0.44}$	$64.77_{\pm 0.15}$	$92.93_{\pm 0.15}$	$92.50_{\pm 0.10}$
Linear head (GT concepts)	100.00	$84.91_{\pm 0.72}$	100.00	$76.20_{\pm 0.03}$
Hard CBM	$69.63_{\pm 0.90}$	$47.87_{\pm 1.00}$	$92.30_{\pm 0.10}$	$68.20_{\pm 0.87}$
CEM	$70.50_{\pm 0.36}$	$\underline{64.77_{\pm 1.71}}$	$\underline{92.63_{\pm 0.25}}$	$\mathbf{91.87_{\pm 0.06}}$
SCBM	$\underline{71.30_{\pm 0.00}}$	$49.03_{\pm 0.71}$	$91.87_{\pm 0.15}$	$68.43_{\pm 0.31}$
Res-SCBM w/o $\mathcal{L}_{\text{int}}$ (ours)	$70.87_{\pm 0.15}$	$64.43_{\pm 0.72}$	$92.60_{\pm 0.17}$	$\mathbf{91.87_{\pm 0.06}}$
Res-SCBM (ours)	$\mathbf{71.80_{\pm 0.70}}$	$\mathbf{68.20_{\pm 0.69}}$	$\mathbf{92.77_{\pm 0.06}}$	$\underline{91.77_{\pm 0.12}}$

Table 2: Mean concept ROC-AUC (%) before concept interventions. Values are means over seeds, with standard deviations in subscript. Best/second-best concept-based results are bold/underlined. Residual models remain competitive with baseline concept prediction performance.

Method	CUB	CUB Incomplete	AwA2	AwA2 Incomplete
Hard CBM	$\underline{97.83_{\pm 0.12}}$	$97.00_{\pm 0.17}$	$\underline{99.30_{\pm 0.00}}$	$\underline{99.10_{\pm 0.00}}$
CEM	$\underline{97.83_{\pm 0.06}}$	$97.13_{\pm 0.15}$	$\mathbf{99.37_{\pm 0.06}}$	$\mathbf{99.27_{\pm 0.06}}$
SCBM	$\mathbf{97.97_{\pm 0.06}}$	$97.17_{\pm 0.06}$	$99.17_{\pm 0.06}$	$\underline{99.10_{\pm 0.00}}$
Res-SCBM w/o $\mathcal{L}_{\text{int}}$ (ours)	$\mathbf{97.97_{\pm 0.06}}$	$\underline{97.33_{\pm 0.06}}$	$99.00_{\pm 0.00}$	$\underline{99.10_{\pm 0.00}}$
Res-SCBM (ours)	$\mathbf{97.97_{\pm 0.06}}$	$\mathbf{97.53_{\pm 0.12}}$	$99.03_{\pm 0.06}$	$\underline{99.10_{\pm 0.00}}$

Similar trends can be observed in concept prediction performance. Specifically, as shown in Table 2, the mean concept ROC-AUC of Res-SCBMs remains competitive compared with previous baselines. This holds for complete and incomplete tasks, suggesting that the residual pathway in Res-SCBMs may indeed capture complementary task-relevant information not in the concept set.

5.2 Test-Time Concept Interventions (RQ2)

Next, we examine how interventions affect the downstream task accuracies and concept ROC-AUC of Res-SCBMs. Our results, summarised in Figure 2, show that, across all datasets, Res-SCBM remains responsive to concept corrections despite its additional residual pathway. For example, in `CUB-Incomplete` we see that Res-SCBM's task accuracy improves substantially as more concepts are corrected, reaching above 90% task accuracy under full intervention and approaching CEM's intervention performance. Notably, Res-SCBM outperforms CEM when only a few concepts are corrected. This is particularly beneficial in practice, where the cost and effort of obtaining ground-truth concept corrections often limit the number of feasible interventions.

Crucially, propagating interventions across Res-SCBM's bottleneck increases its concept ROC-AUC above that of CEMs on both `CUB` datasets, with a marked improvement on the complete version. On both `AwA2` datasets, Res-SCBM shows a slight initial decrease in concept ROC-AUC before recovering as more concepts are corrected. A similar trend is observed for SCBM on complete `AwA2`, indicating that this behaviour is not unique to the residual extension. On complete `AwA2`, Res-SCBM also remains below the ground-truth-concept reference under full intervention. Nevertheless, Res-SCBM is designed primarily for concept-incomplete settings, where it achieves strong predictive performance while retaining responsiveness to concept interventions.

Finally, we observe that Res-SCBM is also consistently more responsive than Res-SCBM w/o $\mathcal{L}_{\text{int}}$, with the largest separation on `CUB-Incomplete` and a smaller but consistent difference on `AwA2-Incomplete`. This suggests that intervention-aware training increases the influence

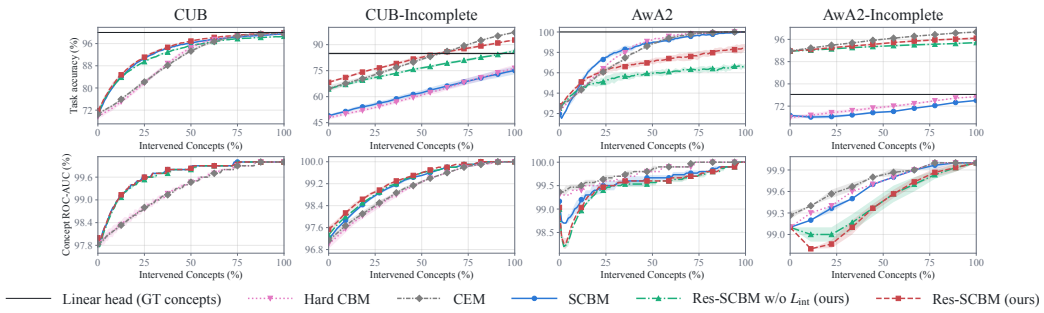


Figure 2: Task accuracy (top) and mean concept ROC-AUC (bottom) as the fraction of corrected concepts increases on `CUB`, `CUB-Incomplete`, `AwA2`, `AwA2-Incomplete`. Res-SCBM remains responsive to concept corrections despite its residual pathway.

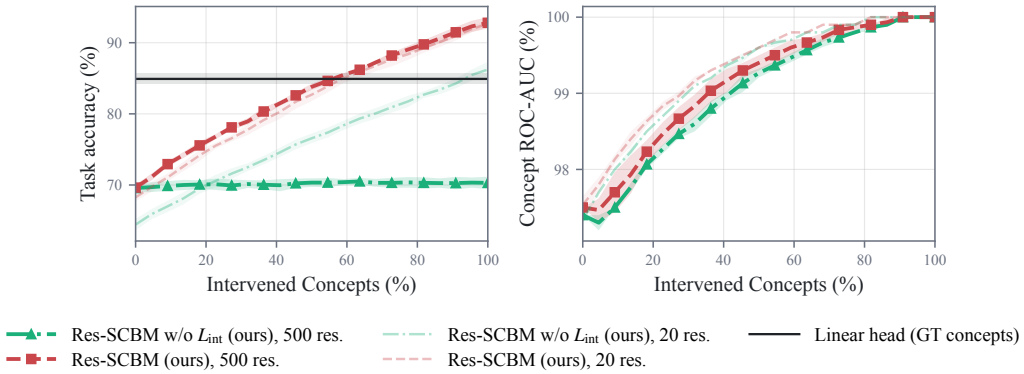


Figure 3: **Effect of intervention-aware training.** Task accuracy (left) and mean concept ROC-AUC (right) under concept interventions on `CUB-Incomplete`, with/without $\mathcal{L}_{\text{int}}$ and residual dimensions $R \in \{20, 500\}$. The horizontal line shows linear classifier accuracy on ground-truth concepts. Intervention-aware training preserves responsiveness at increased residual capacity.

of corrected concept information on the downstream prediction. Together, these results suggest that, at the default residual capacity, both variants remain responsive to concept corrections, while intervention-aware training strengthens this response.

5.3 Effect of Intervention-Aware Training (RQ3)

To examine how intervention-aware training interacts with residual capacity, we compare Res-SCBM with and without $\mathcal{L}_{\text{int}}$ using residual dimensions $R = 20$ and $R = 500$. The larger residual dimension serves as a stress test in which the residual pathway has substantially greater capacity to provide an alternative route to the downstream prediction. Our results, shown in Figure 3, suggest that residual capacity has a pronounced effect when we remove intervention-aware training. Specifically, we observe that, while Res-SCBM w/o $\mathcal{L}_{\text{int}}$ remains responsive to concept corrections when $R = 20$, increasing the residual dimension to $R = 500$ leaves task accuracy almost unchanged as additional concepts are corrected. In contrast, Res-SCBM remains strongly responsive for both $R = 20$ and $R = 500$, with similar intervention trajectories across both.

Importantly, the concept ROC-AUC trajectories are similar across the model variants. This suggests that the divergence in task accuracy reflects how effectively corrected concept information influences the downstream prediction, rather than differences in concept prediction quality. Moreover, Res-SCBM eventually exceeds the linear classifier using ground-truth observed concepts, suggesting that the model combines corrected concept information with complementary information retained in the residual representation.

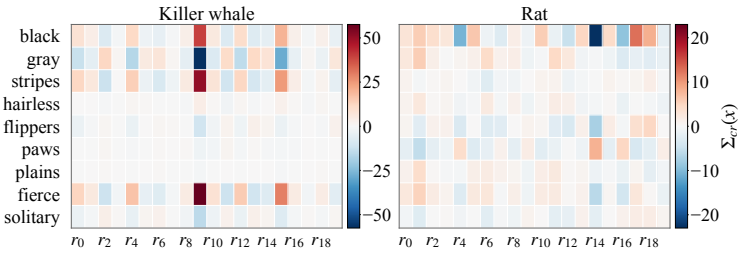


Figure 4: Learned concept–residual cross-covariance for two `AwA2-Incomplete` test samples (concepts: rows; residuals: columns). Distinct patterns across residual dimensions and samples reveal input-dependent concept–residual dependencies.

These results indicate that increasing residual capacity can substantially weaken intervenability when the model is trained without an intervention-aware objective. The effect of $\mathcal{L}_{\text{int}}$ therefore becomes particularly important as residual capacity increases, preserving the influence of concept corrections despite a high-capacity residual pathway.

5.4 Concept–Residual Covariance Analysis (RQ4)

Figure 4 illustrates the learned concept–residual cross-covariance for two `AwA2-Incomplete` test samples. Different residual dimensions exhibit distinct dependency patterns with the observed concepts, which also vary across inputs. These dependencies are central to Res-SCBM's intervention mechanism, as Gaussian conditioning uses the learned covariance structure to update statistically dependent residual variables following concept corrections. Inspecting the cross-covariance therefore provides a mechanistic perspective on how concepts and residuals interact and which residual dimensions may respond to particular concept corrections. Comparing these dependencies across samples could also help infer what individual residual dimensions might encode. A more detailed investigation of these relationships and their potential for improving residual interpretability is left for future work.

6 Limitations and Conclusion

Limitations & Future Work. As in SCBMs, modelling a joint Gaussian distribution requires learning a covariance matrix whose size scales quadratically with the number of variables. Moreover, the effectiveness of intervention propagation depends on the learned concept–residual cross-covariance. Concept and residual predictions may covary across inputs through their predicted means without substantial within-input cross-covariance, limiting the effect of Gaussian conditioning on the residual representation. Future work could therefore investigate methods to encourage stronger concept–residual cross-covariance to improve intervention propagation. A natural extension would also be to leverage the learned covariance structure to better understand what the residual channel encodes and how it interacts with observed concepts.

Conclusion. We introduced Res-SCBMs, a concept-based architecture for addressing concept incompleteness while preserving effective test-time interventions. Res-SCBMs augment the observed concept representation with residual variables and jointly model both through a multivariate Gaussian distribution. The learned concept–residual covariance allows concept interventions to propagate to statistically dependent residual variables through Gaussian conditioning. This joint representation also offers future research directions, including exploiting the learned dependencies to better understand the information captured by residual variables and potentially discover missing concepts. Our experiments show that Res-SCBMs improve predictive performance on concept-incomplete tasks while maintaining effective test-time interventions. Importantly, our model's intervention-aware training helps preserve responsiveness to concept corrections even when the residual representation has substantially greater capacity. These findings demonstrate that additional residual capacity can be combined with effective concept interventions, providing a step towards concept-based models that remain both predictive and intervenable in real-world incomplete settings.

AI USE STATEMENT

In this work, we used generative AI tools to review the manuscript for clarity, grammar, and readability. Generative AI was also used to identify relevant literature, with every identified paper manually read and evaluated by the authors. Generative AI was used to assist with code development, primarily by generating skeleton code, assisting with plot formatting, and identifying areas of the codebase that could be cleaned up or reorganised. Additionally, generative AI was consulted to check aspects of the proposed methodology. These checks were independently verified by the authors through a manual review of the relevant literature. All AI-assisted code was manually inspected and tested for correctness, and all plots were manually inspected and verified. The manuscript was written, thoroughly reviewed, and revised by the authors.

Generative AI was not used to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for mathematical proofs, or assist in writing proofs. It was not used to propose or refine research hypotheses, design experiments, or develop the research methodology beyond the methodological checks described above. No generative AI tools were used for translation, dataset cleaning or reformatting, qualitative or thematic data analysis, or interpretation of experimental results.

All research ideas, experimental design, execution of experiments, analysis, and interpretation of results were carried out by the authors. All AI-assisted material was reviewed, and the authors take full responsibility for the final content of this work, including its results, claims, and conclusions.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide a detailed description of our model formulation, intervention procedure, and training objectives in Section 3. Appendix A.1 describes the datasets, preprocessing, and construction of the concept-incomplete variants. Appendix A.2 provides implementation details and computational resources, while Appendix A.3 documents the model architectures, training procedures, hyperparameters, and test-time intervention protocols. All experiments were conducted using three random seeds, with results reported as mean and standard deviation. Our implementation is publicly available in an anonymous repository.[1]

REFERENCES

Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*, pp. 5948–5955. AAAI Press, 2023. doi: 10.1609/AAAI.V37I5.25736. URL https://doi.org/10.1609/aaai.v37i5.25736.

Gabriele Dominici, Pietro Barbiero, Mateo Espinosa Zarlenga, Alberto Termine, Martin Gjoreski, Giuseppe Marra, and Marc Langheinrich. Causal concept graph models: Beyond causal opacity in deep learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lmKJ1b6PaL.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lió, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 21400–21413. Curran Associates, Inc., 2022. doi: 10.52202/068431-1555. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/867c06823281e506e8059f5c13a57f75-Paper-Conference.pdf.

Mateo Espinosa Zarlenga, Katie Collins, Krishnamurthy Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. Learning to receive help: Intervention-aware concept embedding models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 37849–37875. Curran Associates, Inc., 2023. doi: 10.52202/

[1] https://anonymous.4open.science/r/residual_scbm-DFF9/README.md

075280-1648. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/770cabd044c4eacb6dc5924d9a686dce-Paper-Conference.pdf.

Mateo Espinosa Zarlenga, Gabriele Dominici, Pietro Barbiero, Zohreh Shams, and Mateja Jamnik. Avoiding leakage poisoning: Concept interventions under distribution shifts. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15564–15595. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/espinosa-zarlenga25a.html.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23386–23397. Curran Associates, Inc., 2022. doi: 10.52202/068431-1699. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/944ecf65a46feb578a43abfd5cddd960-Paper-Conference.pdf.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Sonia Laguna, Ričards Marcinkevičs, Moritz Vandenhirtz, and Julia E. Vogt. Beyond concept bottleneck models: How to make black boxes intervenable? In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 85006–85044. Curran Associates, Inc., 2024. doi: 10.52202/079017-2699. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9a439efaa34fe37177eba00737624824-Paper-Conference.pdf.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. *CoRR*, abs/2106.13314, 2021. URL https://arxiv.org/abs/2106.13314.

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FlCg47MNvBA.

Enrico Parisini, Tapabrata Chakraborti, Chris Harbron, Ben D. MacArthur, and Christopher R. S. Banerji. Leakage and interpretability in concept-based models. *CoRR*, abs/2504.14094, 2025. doi: 10.48550/ARXIV.2504.14094. URL https://doi.org/10.48550/arXiv.2504.14094.

Simon Schrodi, Julian Schur, Max Argus, and Thomas Brox. Concept bottleneck models without predefined concepts. *CoRR*, abs/2407.03921, 2024. doi: 10.48550/ARXIV.2407.03921. URL https://doi.org/10.48550/arXiv.2407.03921.

Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental residual concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11030–11040, June 2024.

Nishad Singhi, Jae Myung Kim, Karsten Roth, and Zeynep Akata. Improving intervention efficacy via concept realignment in concept bottleneck models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 422–438, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73347-5.

Moritz Vandenhirtz, Sonia Laguna, Ričards Marcinkevičs, and Julia E. Vogt. Stochastic concept bottleneck models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 51787–51810. Curran Associates, Inc., 2024. doi: 10.52202/079017-1641. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5c7894ac8788555f1cecf536f1e0fd35-Paper-Conference.pdf.

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

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2251–2265, 2019. doi: 10.1109/TPAMI.2018.2857768. URL https://doi.org/10.1109/TPAMI.2018.2857768.

Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=I1quoTXZzc.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20554–20565. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ecb287ff763c169694f682af52c1f309-Paper.pdf.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nA5AZ8CEyow.

Mateo Espinosa Zarlenga. In defense of information leakage in concept-based models. *CoRR*, abs/2606.10669, 2026. doi: 10.48550/ARXIV.2606.10669. URL https://doi.org/10.48550/arXiv.2606.10669.

A Appendix

A.1 Dataset Details

CUB. The Caltech-UCSD Birds-200-2011 (CUB) (Wah et al., 2011) dataset comprises 11,788 images across 200 bird classes, annotated with 312 binary attributes organised into 28 concept groups. Following the preprocessing procedure of Koh et al. (2020), we retain 112 binary class-level concepts, where all images belonging to the same bird class share the same ground-truth concept labels.

CUB-Incomplete. Following Espinosa Zarlenga et al. (2023), we construct `CUB-Incomplete` by retaining 25% of the original concept groups from `CUB`. This results in 22 binary concepts across seven groups: bill shape, head pattern, breast colour, bill length, wing shape, tail pattern, and bill colour. The dataset retains all 200 bird classes and uses the same train, validation, and test splits as `CUB`.

AwA2. We use the Animals with Attributes 2 (AwA2) dataset (Xian et al., 2019), comprising 37,322 images across 50 animal classes. Each class is annotated with 85 binary attributes. Following Espinosa Zarlenga et al. (2025), the attributes are organised into 28 concept groups. Unlike instance-level annotations, the attributes are defined at the class level, meaning that all images belonging to the same animal class share the same ground-truth concept labels.

AwA2-Incomplete. Following Espinosa Zarlenga et al. (2025), we use a concept-incomplete variant of `AwA2` that retains nine of the original 85 concepts: black, gray, stripes, hairless, flippers, paws, plains, fierce, and solitary. The dataset retains all 50 animal classes and uses the same train, validation, and test splits as `AwA2`.

Dataset overview. Table 3 summarises the properties of all datasets used in our experiments.

Table 3: High-level properties of the datasets used in our experiments. Train, val., and test indicate the number of samples in each split. All input images have dimensions $3 \times 224 \times 224$.

Dataset	Train	Val.	Test	Classes	Concepts	Groups
`CUB`	4,796	1,198	5,794	200	112	28
`CUB-Incomplete`	4,796	1,198	5,794	200	22	7
`AwA2`	22,393	7,464	7,465	50	85	28
`AwA2-Incomplete`	22,393	7,464	7,465	50	9	6

A.2 Implementation Details

All models were implemented in PyTorch v2.4.1 and trained on a GPU cluster equipped with NVIDIA GeForce RTX 2080 GPUs. Each training job was allocated one GPU and two CPU cores. All experiments were conducted using three random seeds, and results are reported as mean $\pm$ standard deviation across seeds.

A.3 Model Architecture and Hyperparameters

Architecture. For the `CUB`-based datasets, we use an ImageNet-pretrained ResNet-18 (He et al., 2016) as the image backbone. For the `AwA2`-based datasets, we use frozen, 2048-dimensional embeddings extracted using an ImageNet-pretrained ResNet-101, following Xian et al. (2019). All models use a linear downstream task head.

SCBM and Res-SCBM. The original SCBM formulation considers both global and amortised covariance parametrisations (Vandenhirtz et al., 2024). The global variant learns a single covariance matrix shared across the dataset, whereas the amortised variant predicts an input-dependent covariance matrix for each sample. For SCBM and both Res-SCBM variants, we use linear output

heads to parameterise the Gaussian distribution and employ the amortised covariance parametrisation throughout. We use $M = 100$ Monte Carlo samples to approximate the training objectives. The loss coefficients defined in Eqs 5 and 7 are set to $\lambda_1 = \lambda_2 = \lambda_{\text{int}} = 1$, where applicable.

Unless otherwise specified, we use $R = 20$ residual variables across all experiments. For the experiment investigating the effect of intervention-aware training in Section 5.3 under increased residual capacity, we increase the residual dimension to $R = 500$. Both Res-SCBM variants share the same architecture and hyperparameters, differing only in whether the intervention-aware loss $\mathcal{L}_{\text{int}}$ is included.

CEM. For CEM, we use an embedding dimension of 16 per concept, following Espinosa Zarlenga et al. (2022). We additionally employ random intervention training (RandInt) with an intervention probability of 0.25, where randomly selected concepts are corrected to their ground-truth values during training (Espinosa Zarlenga et al., 2022).

DNN (black box). We use a black-box classifier with the same encoder as the corresponding concept-based models, followed by a linear layer, a LeakyReLU activation, and a linear classification head. To match the bottleneck capacity of both Res-SCBM variants, we set the hidden dimension to $d = C + 20$, where C is the number of observed concepts and 20 is the residual dimension. Unlike the concept-based models, all hidden units are unsupervised, and the DNN is trained solely using task cross-entropy.

Training. All models were trained for a maximum of 200 epochs, with early stopping patience of 75 epochs for `CUB` and `CUB-Incomplete` and 30 epochs for `AwA2` and `AwA2-Incomplete`. We use a learning rate of 10^{-4} and a batch size of 64 across all models.

Concept interventions. For all test-time intervention experiments, we use a random intervention policy (Koh et al., 2020; Vandenhirtz et al., 2024), selecting a random subset of concepts for correction at each intervention step.

For SCBM and both Res-SCBM variants, we compute corrected concept logits using the empirical percentile strategy described in Eq. 3. We then condition the joint Gaussian distribution on these logits, allowing interventions to propagate to statistically dependent concepts and, in our models, residual variables.

For Hard CBM (Havasi et al., 2022) and CEM (Espinosa Zarlenga et al., 2022), we instead replace the predicted concept probabilities with their ground-truth binary values. In Hard CBM, this corrects the corresponding entries in the binary concept bottleneck. In CEM, it selects the corresponding negative or positive concept embedding instead of the original probability-weighted combination of the two.