Probabilistic Concept Bottleneck Models

Eunji Kim [1] **Dahuin Jung** [1] **Sangha Park** [1] **Siwon Kim** [1] **Sungroh Yoon** [1 2]

Abstract

Interpretable models are designed to make decisions in a human-interpretable manner. Representatively, Concept Bottleneck Models (CBM) follow a two-step process of concept prediction and class prediction based on the predicted concepts. CBM provides explanations with high-level concepts derived from concept predictions; thus, reliable concept predictions are important for trustworthiness. In this study, we address the ambiguity issue that can harm reliability. While the existence of a concept can often be ambiguous in the data, CBM predicts concepts deterministically without considering this ambiguity. To provide a reliable interpretation against this ambiguity, we propose Probabilistic Concept Bottleneck Models (ProbCBM). By leveraging probabilistic concept embeddings, ProbCBM models uncertainty in concept prediction and provides explanations based on the concept and its corresponding uncertainty. This uncertainty enhances the reliability of the explanations. Furthermore, as class uncertainty is derived from concept uncertainty in ProbCBM, we can explain class uncertainty by means of concept uncertainty. Code is publicly available at https://github.com/ejkim47/prob-cbm.

1. Introduction

As deep learning systems have been increasingly used in various applications and fields, ensuring transparency of the systems' decision-making has become a significant challenge (Esteva et al., 2019; Miller, 2019). Numerous post-hoc explanation methods have been introduced to explain the decision-making of already-trained deep neural networks (Simonyan et al., 2014; Kim et al., 2018a; Goyal et al., 2019). However, post-hoc methods cannot entirely explain the model's prediction (Zhou et al., 2018) and provide approximate explanations in a human-understandable form, which may lead to incorrect explanations (Rudin, 2019). In contrast, interpretable models are designed to make decisions through human-interpretable processes, ensuring interpretability and not requiring an external explanation method to account for their decisions. Accordingly, research on building interpretable models has been actively conducted (Melis & Jaakkola, 2018; Chen et al., 2019; Koh et al., 2020; Jung et al., 2020; Bohle et al., 2021).

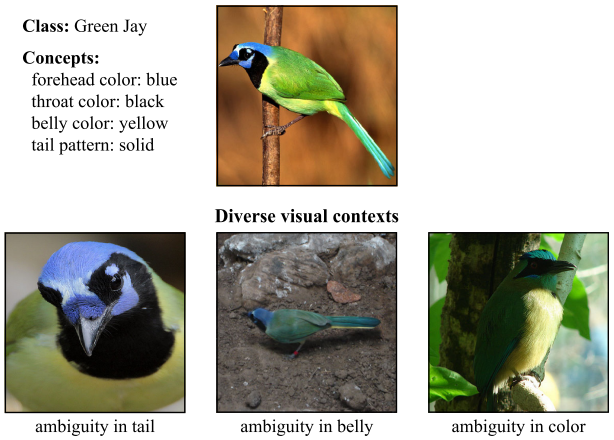


Figure 1. Examples of ambiguous cases in the existence of the concept. Images have diverse visual contexts, where partial concepts may become invisible and unclear.

A concept-based model makes a decision and provides an explanation based on high-level concepts (Koh et al., 2020; Chen et al., 2020). Here, the concepts have semantic meanings that align with human understanding and can be expressed via imagery or language (Kim et al., 2018a). Concept Bottleneck Models (CBM) (Koh et al., 2020), which are widely used concept-based models, adopt concept prediction in the middle of the decision-making process of the black-box model. In CBM, the final decision is made based on the predicted concepts; thus, the concept prediction serves as an explanation. Accordingly, concept prediction is important for ensuring interpretability in CBM. The concept prediction in CBM is trained as deterministic binary classification by using a dataset that includes concept labels indicating the existence (1) or non-existence (0) of a concept.

[1]Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea [2]Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul, Korea. Correspondence to: Sungroh Yoon <sryoon@snu.ac.kr>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

However, the existence of a concept is often ambiguous in some cases where the deterministic concept prediction can harm the reliability of the concept explanation. We conjecture *diverse visual contexts* as the origin of ambiguity in concepts. Figure 1 illustrates examples belonging to the same class with different visual contexts. In contrast to the top image, the bottom images either lack the tail or belly or exhibit a different color tone, despite belonging to the same class. Deterministic predictions on those properties may harm the faithfulness of the concept explanation. This issue can be further exacerbated when training with discrete concept labels. To alleviate the burden of individually annotating concept labels, images belonging to the same class are normally assigned the same concept labels (Koh et al., 2020). Some instances may not actually contain the labeled concepts. Furthermore, data augmentation techniques (*e.g.,* random cropping) are commonly used to enhance prediction performance but can introduce diverse visual contexts that lead to the ambiguity issue.

To reflect the aforementioned ambiguity in concept prediction, we propose Probabilistic Concept Bottleneck Models (ProbCBM). ProbCBM exploits probabilistic embeddings (Oh et al., 2019; Shi & Jain, 2019; Chun et al., 2021) in the concept embedding space and reflects uncertainty in concept prediction. Figure 2 visualizes the concept and class embedding spaces of ProbCBM with examples. Depending on the uncertainty in concept prediction originating from *diverse visual contexts*, ProbCBM maps an image to the concept embeddings with probabilistic distributions, which model concept uncertainties. The concept embeddings from all concepts are projected to form class embeddings; thus, the final class prediction is derived from concept prediction.

ProbCBM explains its prediction with the predicted concepts and the estimated concept uncertainties, ensuring the reliability of the explanation. It is also capable of providing class uncertainty drawn by concept uncertainty, which means the class uncertainty can be explained with concept uncertainty. Through various empirical analyses, we illustrate uncertainty estimation and prediction of ProbCBM. We examine how concept uncertainty varies across different visual contexts and show that the ambiguity introduced by image transformation promotes an increase in uncertainty. We also explore the practical application of estimated uncertainty in human-model interactions in ProbCBM.

Our main contributions are as follows:

- To the best of our knowledge, we pose the ambiguity issue in concept prediction for the first time and address it with uncertainty modeling.
- We propose ProbCBM, an interpretable model that provides explanations based on the concept and its corresponding uncertainty by exploiting probabilistic embeddings.

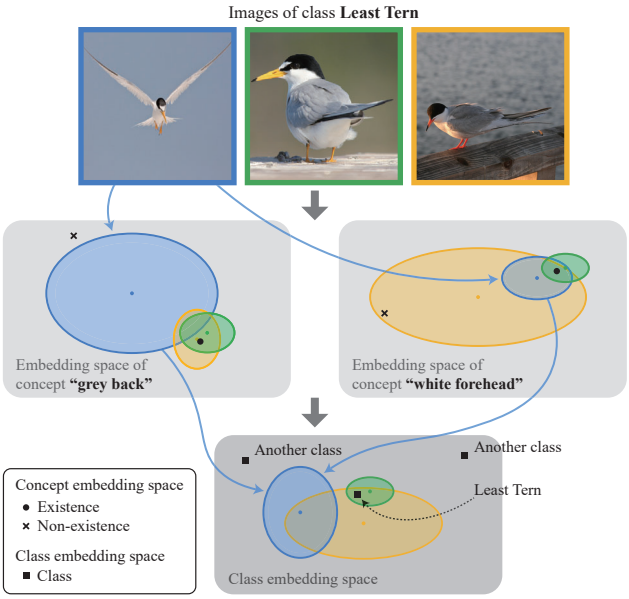


Figure 2. Probabilistic embeddings in ProbCBM. Images are mapped as probabilistic embeddings in the concept embedding space and the probabilistic concept embeddings are mapped to the class embedding space. Arrows represent those mappings. For simplicity, only arrows of the leftmost image are drawn. The same color represents the same image. The images with more ambiguity in concept prediction are mapped as embeddings with larger ellipses. The anchor points represent the existence and non-existence of the concepts in the concept embedding space and classes in the class embedding space.

- We analyze the estimated uncertainty through various experimental results, mainly focusing on the aforementioned origin of ambiguity.

2. Related Work

2.1. Interpretable Neural Networks

Interpretable neural networks are built to ensure interpretability (Melis & Jaakkola, 2018). One way to build an interpretable prediction process is the use of a concept-based explanation, where models learn human interpretable concepts and make predictions based on the learned concepts (Koh et al., 2020; Chen et al., 2019). Since classification models are trained solely with class labels, there have been studies focusing on learning prototypes that represent distinctive properties of each class in an unsupervised manner (Melis & Jaakkola, 2018; Chen et al., 2019).

In contrast, CBM (Koh et al., 2020) utilizes both concept and class labels. It first predicts the concept for a given input and then proceeds to predict the class. Owing to its simple structure based on human-defined concepts, there have been studies on building improved interpretable models in the framework of CBM (Sarkar et al., 2022). To overcome the

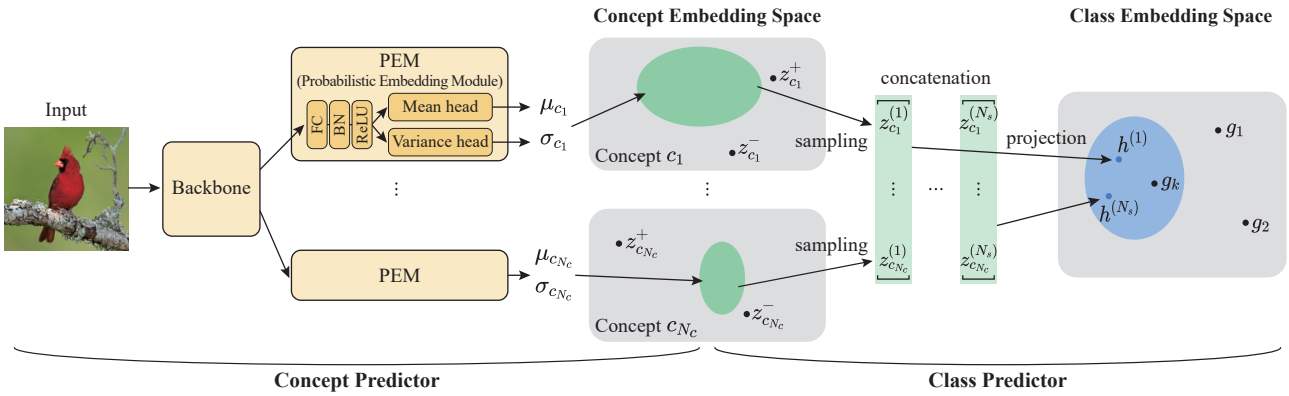


Figure 3. Prediction flow of ProbCBM. Ellipses represent probabilistic embeddings of a given input.

challenges associated with concept labeling and make use of well-trained models that lack interpretability, Post-hoc CBM (PCBM) was proposed by Yuksekgonul et al. (2023), allowing the conversion of pre-trained models into CBM. Zarlenga et al. (2022) posed a trade-off issue between accuracy and interpretability in CBM and proposed a novel model called Concept Embedding Models (CEM) by introducing embeddings to leverage high-level features in task prediction, which represent more information beyond the existence of concepts. Also, some efforts have been made to enhance the reliability of CBM (Havasi et al., 2022; Marconato et al., 2022), focusing on an information leakage issue, where unintended information is utilized by the class predictor (Mahinpei et al., 2021). However, no research has been performed to mitigate the ambiguity in concept prediction, which is the primary focus of this study.

2.2. Uncertainty and Probabilistic Embeddings

Uncertainty modeling is used to improve the interpretability and robustness (Blundell et al., 2015; Gal & Ghahramani, 2016; Kendall & Gal, 2017). Uncertainty is mainly divided into two types: model uncertainty and data uncertainty. Model uncertainty comes from the model parameters, whereas data uncertainty comes from the noise of the data (Gal & Ghahramani, 2016).

An approach with probabilistic embeddings is a general method that mainly considers data uncertainty, where the representations of input samples are expressed as probabilistic distributions. Shi & Jain (2019) proposed probabilistic face embeddings to address feature ambiguity in real-world face images. Oh et al. (2019) proposed Hedged Instance Embeddings (HIB) and defined the matching probability between a pair via Monte-Carlo estimation. Chun et al. (2021) extended HIB to the joint embedding space of image and text and solved a cross-modal retrieval problem. We extend HIB to the concept embedding space for concept prediction and build an interpretable model that makes decisions based on the predicted concept embeddings.

3. Preliminary: Concept Bottleneck Models

CBM (Koh et al., 2020) consists of two predictors: a concept predictor and a class predictor. Given an input x, the concept predictor g maps it to a concept space to predict a set of the existences of concepts $\mathcal{C}$ ($\hat{\mathcal{C}} = g(x)$). The class predictor f estimates the class y from $\hat{\mathcal{C}}$, *i.e.,* the concepts predicted by the concept predictor ($\hat{y} = f(\hat{\mathcal{C}})$). CBM's decision-making process is expressed as $\hat{y} = f(g(x))$, where the concept space serves as an interpretable bottleneck for the class prediction. To learn the mapping g, the data pairs $(x, \mathcal{C})$ are required. Thus, CBM requires the dataset $\{(x^{(i)}, \mathcal{C}^{(i)}, y^{(i)})\}_{i=1}^{N}$ while the conventional classifier that directly maps x to y require the dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, where N represents the number of data pairs.

CBM has two strengths with regard to interpretability, which make it useful as an interpretable model. First, it can provide the concept information that it discovers in an input. Because CBM makes the final prediction according to the predicted concept information, the predicted concept information is a valid explanation for the model's decision. Second, concept intervention enables further understanding of the model. In CBM, changes in concept prediction modify the classification result. When the predicted concept is incorrect (not aligned with human understanding), humans can debug the model by intervening in the concept prediction and changing the model's decision. The relationship between a concept and a class can be analyzed by observing the result of concept intervention, which provides counterfactual explanations (Abid et al., 2022). Because we build our model, *i.e.,* ProbCBM, on the basis of CBM's framework, ProbCBM inherits the strengths of CBM.

4. Method

Overview. We propose ProbCBM, an interpretable model that exploits probabilistic embedding in prediction. It provides concept prediction and concept uncertainty as explanations. We extend the probabilistic modeling in HIB (Oh

et al., 2019) to build ProbCBM. Figure 3 shows the overall prediction flow of ProbCBM. Similar to CBM, ProbCBM comprises a concept predictor and a class predictor. The concept predictor generates concept embeddings and the class predictor generates class embeddings from the predicted concept embeddings. With these embeddings, concept and class prediction problems are solved as matching problems with concept and class anchors, respectively. We train ProbCBM with the dataset $\{\left(x^{(i)}, \mathcal{C}^{(i)}, y^{(i)}\right)\}_{i=1}^{N}$.

4.1. Probabilistic Concept Modeling

Probabilistic concept embedding. Given an input x, the concept predictor makes probabilistic concept embedding for each concept $c \in \mathcal{C}$, which is formulated as a normal distribution with a mean vector and a diagonal covariance matrix.

$$p(z_c|x) \sim \mathcal{N}(\mu_c, \text{diag}(\sigma_c)), \tag{1}$$

where $\mu_c, \sigma_c \in \mathbb{R}^{D_c}$ and D_c represents the dimension of the concept embedding space. μ_c and σ_c are predicted by a probabilistic embedding module (PEM), which is described in Sec. 4.5. As shown in Figure 3, the generation of probabilistic concept embedding is performed for every concept using a shared backbone and individual PEMs.

Concept prediction. With N_s representations $\{z_c^{(n)}\}_{n=1}^{N_s}$ sampled from $p(z_c|x)$, the probability of the existence of concept c ($c = 1$) is obtained via Monte-Carlo estimation:

$$p(c = 1|x) \approx \frac{1}{N_s} \sum_{n=1}^{N_s} p(c = 1|z_c^{(n)}), \tag{2}$$

$$p(c = 1|z_c^{(n)}) = s\left(a\left(||z_c^{(n)} - z_c^-||_2 - ||z_c^{(n)} - z_c^+||_2\right)\right), \tag{3}$$

where $a > 0$ is a learnable parameter and $s(\cdot)$ represents a sigmoid function. We define trainable anchor points z_c^+ and z_c^- in $\mathbb{R}^{D_c}$, which represent the existence and non-existence of concept c, respectively. For a given sampled representation $z_c^{(n)}$, the probability that the concept exists is defined by the Euclidean distance with z_c^+ and z_c^-. If $z_c^{(n)}$ becomes closer to z_c^+ than z_c^-, the probability of existence increases, and if it becomes farther away, the probability decreases. More explanations on design of concept prediction are provided in Appendix A.

4.2. Probabilistic Class Modeling

Class embedding. The class predictor generates a class embedding from the concept embeddings. We concatenate sampled concept representations from all concepts $\mathcal{C} = \{c_1, c_2, \ldots, c_{N_c}\}$, where N_c represents the number of concepts. The concatenation is projected to the class embedding space with a fully connected (FC) layer.

$$h^{(n)} = \mathbf{w}^T \left(\left[z_{c_1}^{(n)}, z_{c_2}^{(n)}, \ldots, z_{c_{N_c}}^{(n)}\right]\right) + \mathbf{b}, \tag{4}$$

where $h^{(n)}$ is a class representation and $\mathbf{w}$ and $\mathbf{b}$ represent the weight and bias of the FC layer, respectively.

Class prediction. The logit for class k is defined by the Euclidean distance between the class embedding for the image and a trainable anchor point for class k, $g_k \in \mathbb{R}^{D_y}$ where D_y represents the dimension of the class embedding space. We obtain the class probabilities by applying softmax to the logits for overall classes. The classification probability is obtained via Monte-Carlo estimation:

$$p(y_k = 1|x) \approx \frac{1}{N_s} \sum_{n}^{N_s} \frac{\exp(-d\|h^{(n)} - g_k\|_2)}{\sum_{k'} \exp(-d\|h^{(n)} - g_{k'}\|_2)}, \tag{5}$$

where $d > 0$ is a learnable parameter.

4.3. Training and Inference

Training objective. We use a binary cross-entropy loss ($\mathcal{L}_{\text{BCE}}$) with the concept probability (Eq. 2). Following HIB (Oh et al., 2019), we additionally use a KL divergence loss between the predicted concept embedding distributions and the standard normal distribution. This prevents the variances from collapsing to zero and makes the distribution $\mathcal{N}(\mu_c, \text{diag}(\sigma_c))$ have only salient information for predicting the probability that the concept c exists.

$$\mathcal{L}_{\text{KL}}(c) = \text{KL}\left(\mathcal{N}(\mu_c, \text{diag}(\sigma_c))||\mathcal{N}(0, I)\right). \tag{6}$$

Thus, the overall training loss for the concept predictor is expressed as:

$$\mathcal{L}_{\text{concept}} = \mathcal{L}_{\text{BCE}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}, \tag{7}$$

where λ_{KL} is a balancing factor.

We use a cross-entropy loss for training class predictor ($\mathcal{L}_{\text{class}}$).

Training scheme. We train the concept predictor and class predictor separately. First, the concept predictor is trained with $\mathcal{L}_{\text{concept}}$. Then, the class predictor is trained with $\mathcal{L}_{\text{class}}$ using the concept embeddings predicted by the concept predictor. During the training of the class predictor, a sampled concept embedding $z_c^{(n)}$ is replaced with the concept anchor of a ground-truth concept label (z_c^+ and z_c^- for positive and negative labels, respectively) with the predefined probability p_{replace}. This prevents the class predictor from learning with incorrect concepts, improving the final classification performance and the reliability of the class predictor. See Algorithm 1 in the Appendix for details.

Inference. Inference can be done by approximating the probabilities via Monte-Carlo sampling (Eqs. 3 and 5) or using μ_c as z_c without sampling. The inference method is discussed in Sec 5.2.2.

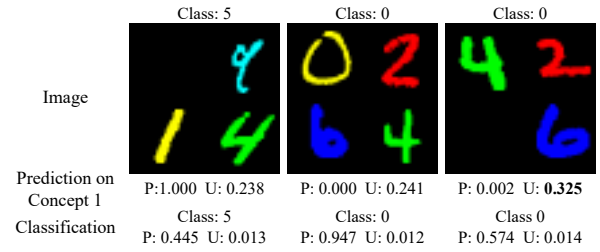


Figure 4. Examples of ProbCBM's prediction in diverse visual contexts. P and U represent the probability and uncertainty, respectively. The highest value of concept uncertainty is bolded.

4.4. Derivation of Uncertainty

ProbCBM leverages probabilistic modeling, enabling us to estimate uncertainty directly from the predicted probabilistic distribution without the need for sampling. We quantify uncertainty using the determinant of the covariance matrix, which represents the volume of the probabilistic distribution. Because the distribution of the concept embedding is parameterized with a diagonal covariance matrix, the uncertainty of each concept c can be calculated as the geometric mean of the diagonal elements σ_c. The class embedding is a linear transformation of the concatenation of concept embeddings, and the class embeddings follow $\mathcal{N}(\mathbf{w}^T\mu + \mathbf{b}, \mathbf{w}^T\Sigma\mathbf{w})$, where $\mu = \left[\mu_{c_1}, \mu_{c_2}, ..., \mu_{c_{N_c}}\right]$ and $\Sigma = \text{diag}\left(\left[\sigma_{c_1}, \sigma_{c_2}, ..., \sigma_{c_{N_c}}\right]\right)$. Hence, the determinant of $\mathbf{w}^T\Sigma\mathbf{w}$ serves as an uncertainty measure of class prediction.

4.5. Architecture

As shown in Figure 3, with a shared backbone (*e.g.,* ResNet (He et al., 2016)), there is one PEM for each concept. PEM predicts the mean and diagonal covariance vector for the corresponding concept from a feature map extracted from the backbone. Following the work of Chun et al. (2021), we use self-attention-based mean and variance head submodules in PEM. To reduce the feature dimension, we add an FC layer followed by batch normalization (Ioffe & Szegedy, 2015) and ReLU activation (Nair & Hinton, 2010) before the mean and variance head modules.

5. Experiments

We evaluate ProbCBM with one synthetic dataset and two real-world datasets. With these datasets, we demonstrate how ProbCBM effectively models uncertainty under diverse visual contexts presented in raw images. Additionally, we conduct an analysis to examine the ambiguity induced by image transformations.

5.1. Analysis with Synthetic Dataset

5.1.1. Dataset and Experimental Setting

Dataset. We create synthetic data using the MNIST dataset (LeCun et al., 2010) which contains images of 10 digits. We utilize each digit as a single concept. We first divide the 10 digits into five groups: (0, 1), (2, 3), (4, 5), (6, 7), and (8, 9). We then generate a single image with four digits from four different groups and annotate a new class label out of 12 classes, depending on the combination of digits. The digits in the same group are colorized the same color. To add diversity to the concept combinations for each class, we randomly drop one of the four digits in the images. There is no image that can be considered as multiple classes owing to the drop because no class shares more than three concepts with any other class. To simulate ambiguity in the training data, we use cutout (DeVries & Taylor, 2017) augmentation method. The synthetic dataset provides instance-level concept annotations. See Appendix C for details.

Setting. We use a shallow convolutional network consisting of two convolutional layers with 3×3 filters, followed by batch normalization and ReLU activation as a backbone. ProbCBM achieves accuracies of 99.8% and 99.8% for concept prediction and classification, respectively. See Appendix D for experimental details.

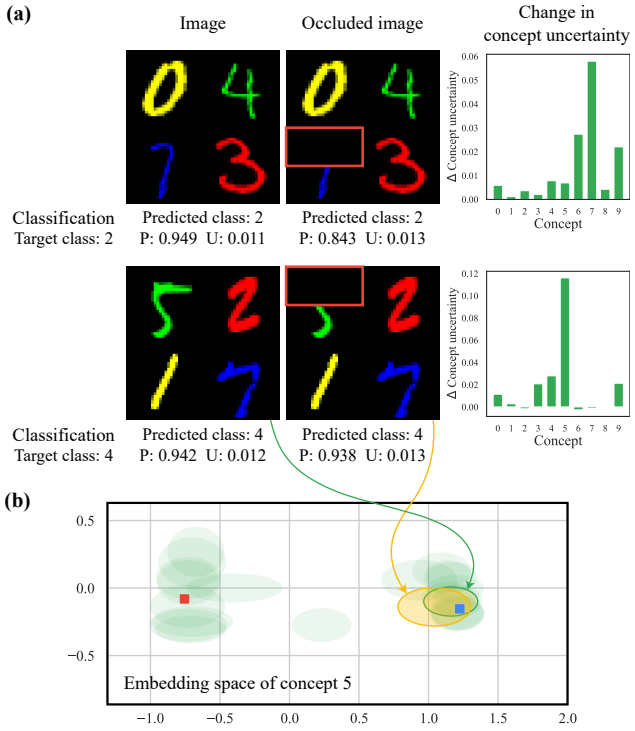


Figure 5. (a) Examples of changes in the concept and class uncertainties after occlusion of parts of images. Red bounding boxes denote the occluded parts. P and U represent the probability and uncertainty, respectively. (b) Visualization of the embedding space of concept 5. Blue and red dots present positive and negative anchors, respectively. The probabilistic concept embeddings are visualized as green and yellow ellipses. Blue and red points represent positive and negative anchor points, respectively.

Table 1. Prediction accuracy for the CUB and AwA2 datasets. Results include mean values with standard deviation from three experiment repetitions. "w/o prob." denotes ProbCBM with deterministic embeddings. "w/o sampling" denotes inference without sampling.

DATA	METHOD	CONCEPT	CLASS
CUB	IMAGE: 224 × 224		
	CBM	0.950±0.001	0.670±0.006
	PROBCBM	0.949±0.001	0.680±0.004
	W/O PROB.	0.950±0.001	0.677±0.004
	W/O SAMPLING	0.949±0.001	0.679±0.003
	IMAGE: 299 × 299		
	CBM	0.956±0.001	0.708±0.006
	CEM	0.954±0.001	0.759±0.002
	PROBCBM	0.956±0.001	0.718±0.005
AWA2	CBM	0.975±0.001	0.877±0.004
	PROBCBM	0.975±0.000	0.880±0.002
	W/O PROB.	0.975±0.000	0.880±0.002
	W/O SAMPLING	0.975±0.000	0.880±0.001

5.1.2. UNDERSTANDING ESTIMATED UNCERTAINTY

Ambiguity in diverse visual contexts. We compare the estimated uncertainties of multiple images which have diverse visual contexts. Figure 4 shows examples of ProbCBM's predictions for concept 1, where the left two images are predicted with smaller uncertainties, and the rightmost image is predicted with a larger uncertainty. Because the leftmost image contains concept 1, ProbCBM predicts that concept 1 exists, with a small uncertainty. Additionally, ProbCBM confidently predicts that concept 1 does not exist in the center image. Note that concepts 0 and 1 are in the same group; thus, the existence of concept 0 implies the non-existence of concept 1. The rightmost image contains neither concept 0 nor concept 1; thus, the estimated uncertainty for concept 1 is large. This indicates that the estimated concept uncertainty depends on the visual context. The estimated concept uncertainty is large when there is ambiguity in the prediction of the corresponding concept, leading to large uncertainty in classification.

Introducing ambiguity via transformation. We artificially introduce ambiguity in the data by occluding the upper half region of one of four digits in images. Figure 5(a) shows examples of the changes in concept uncertainty after the occlusion of part of the images. When part of a digit is occluded, the uncertainty of the corresponding concept increases significantly (concept 7 in 1^{st} row and concept 5 in 2^{nd} row), increasing the class uncertainty.

Visualization in concept embedding space. We visualize the predicted probabilistic concept embeddings and positive and negative anchors of concept 5, as shown in Figure 5(b). We use principal component analysis (PCA) (Pearson, 1901) to visualize them in a two-dimensional (2D) space. Each ellipse represents the predicted concept embedding of an image, whose size denotes the uncertainty. The embeddings of the samples are distributed near the positive or negative anchors and have various uncertainty values. The concept embedding of the occluded image is represented by an ellipse larger than that of the original image. This indicates that ProbCBM successfully models concept uncertainty by using probabilistic embeddings.

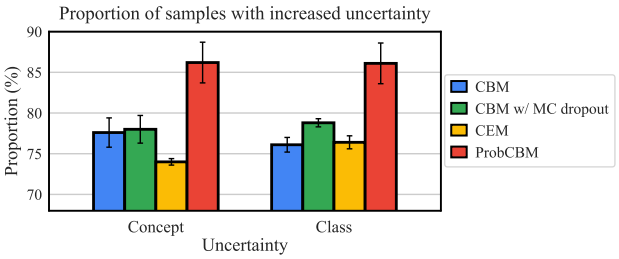


Figure 6. Proportion (%) of samples whose uncertainty increases after occlusion for the CUB dataset. The experiments are conducted with an image size of 299 × 299. Results include mean values with standard deviation from three experiment repetitions.

5.2. Analysis with Real-world Datasets

5.2.1. DATASETS AND EXPERIMENTAL SETTING

CUB. Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011) contains 11,788 images from 200 bird species, which are annotated with 312 attributes. Koh et al. (2020) denoised the attribute annotations and 112 attributes remained, where the images belonging to the same class have the same attribute annotations. We use the attribute annotations and dataset splits that Koh et al. (2020) provided. The attribute annotations are used as concepts.

AwA2. Animal with Attributes (AwA2) dataset (Xian et al., 2018) contains 37,322 images from 50 animal classes, which are annotated with 85 attributes. Because the list of attributes includes invisible attributes (*e.g.,* fast, slow), we retain 45 attributes that are visible in images and use them as concepts. See Appendix C for the remaining attribute labels.

Setting. We use ResNet18 (He et al., 2016) as a backbone, which is pretrained with ImageNet-1K (Russakovsky et al., 2015). We use color jittering, random horizontal flips, and random resized cropping for data augmentation. Note that we apply the same data augmentation to all methods, which is a weaker level of augmentation than that used in other works (CBM and CEM (Zarlenga et al., 2022)), to reduce the distortion of concept labels due to data augmentation[1]. For most experiments, the images are resized to 224 × 224. We additionally conduct experiments with the image size of 299 × 299 to compare ProbCBM with other methods for the CUB dataset (Sec. 5.2.2). See Appendix D for more details.

[1]This makes a difference in the reported performance of prediction of other methods.

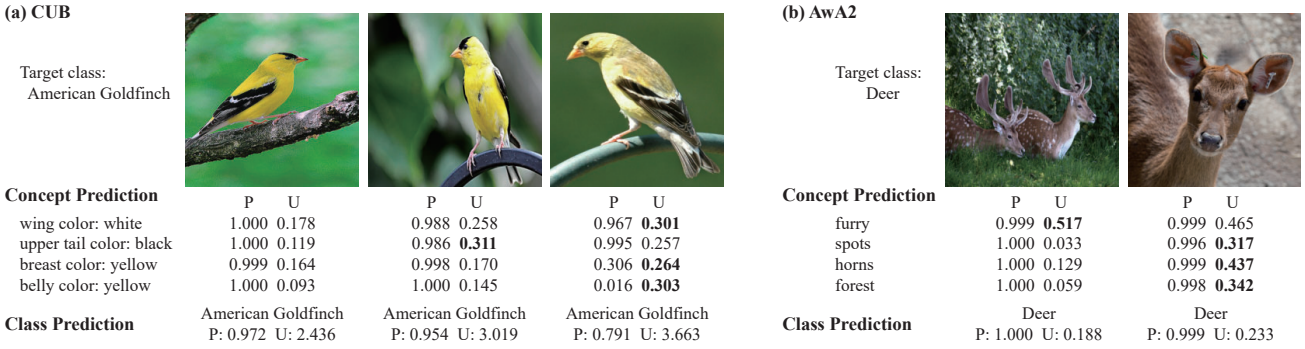


Figure 7. Examples of ProbCBM's prediction in diverse visual contexts for the (a) CUB and (b) AwA2 datasets. P and U represent the probability and uncertainty, respectively. The highest value of uncertainty in each concept is bolded.

5.2.2. Comparisons

Prediction performance. It is important to enhance the reliability of the model while avoiding a drop in prediction performance. To evaluate this aspect, we compare the concept and class prediction performances of ProbCBM with CBM and CEM. CEM is chosen as a comparison method because it also utilizes concept embeddings.

The results are presented in Table 1. ProbCBM demonstrates superior classification accuracy compared to CBM while maintaining a comparable concept prediction accuracy. This indicates that introducing uncertainty with probabilistic embedding does not drop prediction performance. ProbCBM achieves lower class prediction performance but slightly higher concept prediction performance compared to CEM. This is due to their distinct training approaches. In CEM, concept embeddings are trained to represent other information in addition to the existence or non-existence of concepts, aiming to improve classification performance through joint training of concept and class predictors. In contrast, ProbCBM focuses on training concept embeddings to represent uncertainties without explicit consideration of classification through sequential training. They prioritize different aspects of prediction. Additionally, it is worth noting that CEM has a larger number of parameters (15.6M) compared to ProbCBM (12.5M). The comparison of prediction performances with other models shows that ProbCBM does not compromise prediction performance as well as provides uncertainty-based interpretation.

To examine the effect of probabilistic embedding, we also evaluate the performance of the model that exploits deterministic concept and class embeddings, which we refer to as "w/o prob.," as shown in Table 1. The classification accuracy of ProbCBM is higher than or comparable to that of the model exploiting deterministic embeddings.

Efficient inference. Inference of ProbCBM can be done with and without Monte-Carlo sampling. To reduce the computational cost for sampling, we can use the predicted means instead of sampled representations. As shown in Table 1, the difference between inference with sampling and inference without sampling is minor. The anchor points can be considered as the distributions following a normal distribution with zero variance. Thus, using the means can be viewed as directly using the distance between two distributions – probabilistic class embeddings and class anchors – in prediction.

Uncertainty estimation. To evaluate the capability of estimating uncertainty, we intentionally design scenarios that would induce an increase in uncertainty and quantitatively compare the ability to detect this increase. Motivated by a common evaluation protocol (Oh et al., 2019; Shi & Jain, 2019), we occlude a patch of size 64×64 from the center of images. This occlusion induces a loss of information, resulting in an increase in uncertainty. We then analyze the proportion of samples that exhibit higher uncertainty than before. In addition to CBM and CEM, we compare our method with MC dropout (Gal & Ghahramani, 2016), a well-known approach for uncertainty estimation, by adopting it to CBM (CBM with MC dropout). Uncertainties in the models, except for ProbCBM, are estimated using the entropy of the predicted concept and class probabilities.

Figure 6 depicts the proportion of samples exhibiting increased uncertainty after occlusion. The results clearly demonstrate that ProbCBM is effective in detecting an increase in both concept and class uncertainties across a wide range of samples. While MC dropout improves the detection of increased uncertainty in concept and class, its performance is inferior to that of ProbCBM. Notably, CEM, despite employing concept embeddings, diverges significantly in uncertainty estimation due to disparate objectives and strategies compared to ProbCBM. These findings affirm the effectiveness of ProbCBM in uncertainty estimation.

5.2.3. Understanding Estimated Uncertainty

Ambiguity in diverse visual contexts. Figure 7 shows examples of images belonging to the same class in diverse visual contexts. Regarding Figure 7(a), the concepts are

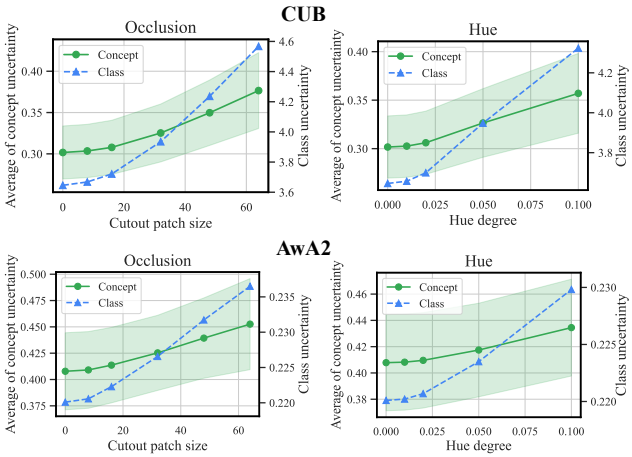


Figure 8. Changes in uncertainty after the introduction of ambiguity via image transformations to various degrees. For the concept uncertainty, line and shade represent the mean and standard deviation over all concepts, respectively.

well visible in the leftmost image; thus, the estimated concept uncertainties are small, inducing small uncertainty in class prediction. The center and rightmost images mainly show the bird's belly and back, respectively, which exhibit different uncertainties. Regarding Figure 7(b), the left image contains spots and horns. The right image does not contain them, but the concept predictor is trained to predict images with deer to have horns and spots; thus, the predictions for these concepts exhibit high uncertainties. Note that class-level concept annotations are used to train ProbCBM, where images belonging to the same class are annotated with the same concepts.

Introducing ambiguity via transformation. We artificially add ambiguity to images by applying transformations: cutout and hue. We cut out a square patch from an image or change the hue of an image. Because the sets of defined concepts in the CUB and AwA2 datasets contain descriptions of parts of objects (*e.g.,* neck, leg) and colors (*e.g.,* wing color), these transformations can remove or distort information about concepts in the images. As shown in Figure 8, a larger size of the patch cut out or degree of change in the hue makes the bigger correspond to larger increases in the concept and class uncertainties, indicating that uncertainty increases as the ambiguity increases.

Visualization in embedding space. We visualize predicted probabilistic embeddings in the 2D space via PCA. Figure 9 shows the embedding space of the concept `leg color: buff`. The image with the buff-colored leg is located near the positive anchor, and the image with the red-colored leg is located near the negative anchor. Two images where the leg is invisible are visualized as larger ellipses between two anchors, representing larger uncertainties. Such a large concept uncertainty can induce a large uncertainty in classification, as shown in Figure 20 in the Appendix.

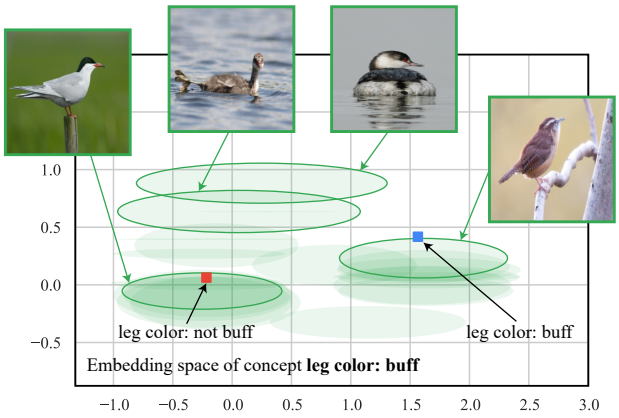


Figure 9. Visualization of the embedding space of the concept `leg color: buff`. Each green ellipse represents the concept embedding of the corresponding sample.

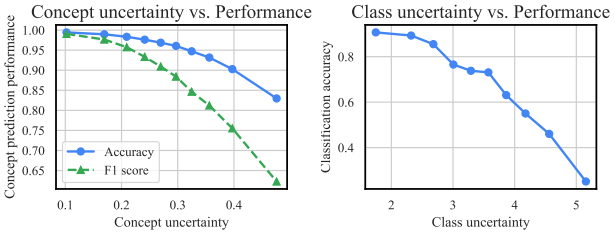


Figure 10. Correlation between performance and uncertainty for the CUB dataset. A point represents one group of samples with similar uncertainties and the x-axis indicates the median uncertainty for each group.

Correlation between uncertainty and performance. The estimated uncertainty can be used to estimate the probability of failure in prediction. We divide the test data into 10 groups according to the degree of uncertainty, where each group has the same number of samples, and evaluate the prediction performance for each group. As shown in Figure 10, groups with larger uncertainties exhibit worse performances for concept and class prediction.

5.2.4. Concept Intervention

Concept intervention is performed to revise incorrect concept predictions for modifying model prediction (Koh et al., 2020). In ProbCBM, concept intervention can be done by replacing sampled concept points with anchors of the ground-truth concept. Following Koh et al. (2020), we group the concepts belonging to a similar category and intervene in the concepts in the same group at once (28 groups for the CUB dataset). As shown in Figure 11, intervening incorrectly predicted concepts, even in random order, consistently increases the classification accuracy, which eventually reaches 100%. This indicates that, during debugging, humans can fix wrong model predictions by intervening in incorrect concepts in ProbCBM.

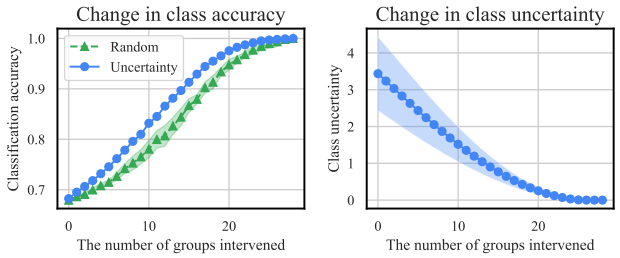


Figure 11. Results of concept intervention for the CUB dataset. Intervention in a random order is conducted five times and is visualized with the means and standard deviations. On the right plot, intervention is conducted in an uncertainty-based order, and the means and standard deviations for all test samples are shown.

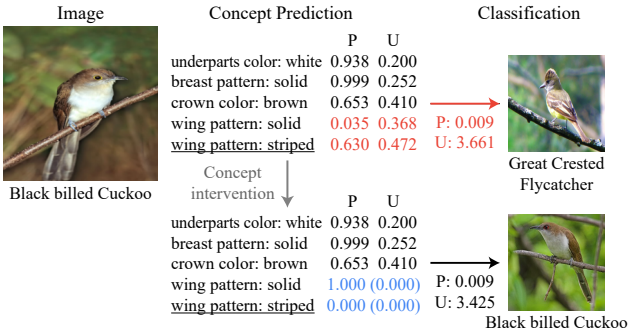


Figure 12. Examples of concept intervention in ProbCBM. P and U represent the probability and uncertainty, respectively. The concept labels for most concepts are positive, except those that are underlined. Red text denotes wrong predictions and blue text denotes the change after concept intervention.

Practically, it is challenging to ascertain the correctness of concept prediction for all concepts. Therefore, it is necessary to develop an efficient strategy that can effectively identify concepts to intervene first to quickly revise the final predictions. Estimated uncertainty serves as a valuable tool for efficient concept intervention (Shin et al., 2022; Chauhan et al., 2022; Sheth et al., 2022). The concepts are intervened in descending order of the maximum uncertainty values of concepts within a group. Figure 11 demonstrates that the order based on uncertainty yields a faster improvement in class accuracy than random orders in ProbCBM. Figure 12 provides an example of concept intervention where we intervene in the concepts related to the wing pattern, which exhibit the highest uncertainties. Consequently, the class prediction is fixed and the class uncertainty decreases.

Likewise, we can make more certain class predictions via concept intervention in ProbCBM. Intervening in concepts makes concept embeddings deterministic; thus, the uncertainty of the intervened concepts becomes zero. Concept intervention in ProbCBM includes not only intervening in the prediction of the existence of concepts but also intervening in the concept uncertainties, affecting the class uncertainty. As shown in Figure 11, the class uncertainty decreases with intervention in more groups of concepts.

6. Limitations

ProbCBM is built upon CBM, aiming to enhance the reliability of concept prediction by incorporating uncertainty. Consequently, it inherits some limitations from CBM. One such limitation is the necessity of concept labels. Thus, in CBM, the accuracy of concept prediction is heavily influenced by the quality of human-annotated concept labels. ProbCBM utilizes concept labels, but by introducing uncertainty to address the ambiguity issue, we can reduce the dependency on the correctness of concept labels. Another limitation is information leakage (Mahinpei et al., 2021), the usage of unintended information in task prediction. The motivation for the concept bottleneck is to build an interpretable prediction process using task prediction only based on human-interpretable concepts. However, the leakage of information that is not interpretable by humans can harm reliability. While we improved reliability in terms of uncertainty, information leakage is also a problem that needs to be resolved to enhance reliability in the framework of CBM.

7. Conclusion

In this paper, we first pose the ambiguity issue that can harm the reliability of concept prediction in the concept bottleneck model. We propose ProbCBM which can successfully reflect the ambiguity as concept uncertainty by introducing probabilistic embeddings. Since the class embedding is derived from the probabilistic concept embedding, ProbCBM is also able to provide uncertainty in class prediction. Various analysis has been presented to help understand ProbCBM and show the improved reliability of the explanations.

Acknowledgements

The authors would like to express their appreciation to Sanghyuk Chun and Sangdoo Yun at NAVER AI Lab for their significant help in the ideation and experimental designs of this work. Many aspects of this work were devised and carried out during the first author's internship at Naver AI Lab, utilizing the Naver Smart Machine Learning (NSML) platform (Kim et al., 2018b).

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (Ministry of Science and ICT, MSIT) (2022R1A3B1077720 and 2022R1A5A708390811), Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (2022-0-00959 and 2021-0-01343: AI Graduate School Program, SNU), and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2023.

References

Abid, A., Yuksekgonul, M., and Zou, J. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning*, pp. 66–88. PMLR, 2022.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622. PMLR, 2015.

Bohle, M., Fritz, M., and Schiele, B. Convolutional dynamic alignment networks for interpretable classifications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10029–10038, 2021.

Chauhan, K., Tiwari, R., Freyberg, J., Shenoy, P., and Dvijotham, K. Interactive concept bottleneck models. *arXiv preprint arXiv:2212.07430*, 2022.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., and Su, J. K. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, pp. 8930–8941, 2019.

Chen, Z., Bei, Y., and Rudin, C. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

Chun, S., Oh, S. J., De Rezende, R. S., Kalantidis, Y., and Larlus, D. Probabilistic embeddings for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8415–8424, 2021.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., and Dean, J. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pp. 1050–1059. PMLR, 2016.

Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., and Lee, S. Counterfactual visual explanations. In *International Conference on Machine Learning*, pp. 2376–2384. PMLR, 2019.

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Heo, B., Chun, S., Oh, S. J., Han, D., Yun, S., Kim, G., Uh, Y., and Ha, J.-W. Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights. In *International Conference on Learning Representations*, 2021.

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.

Jung, D., Lee, J., Yi, J., and Yoon, S. icaps: An interpretable classifier via disentangled capsule networks. In *European Conference on Computer Vision*, pp. 314–330. Springer, 2020.

Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pp. 2668–2677. PMLR, 2018a.

Kim, H., Kim, M., Seo, D., Kim, J., Park, H., Park, S., Jo, H., Kim, K., Yang, Y., Kim, Y., et al. Nsml: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018b.

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.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Mahinpei, A., Clark, J., Lage, I., Doshi-Velez, F., and Wei-Wei, P. Promises and pitfalls of black-box concept learning models. In *Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI, ICML*, 2021.

Marconato, E., Passerini, A., and Teso, S. Glancenets: Interpretable, leak-proof concept-based models. In *Advances in Neural Information Processing Systems*, 2022.

Melis, D. A. and Jaakkola, T. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pp. 7775–7784, 2018.

Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–814, 2010.

Oh, S. J., Gallagher, A. C., Murphy, K. P., Schroff, F., Pan, J., and Roth, J. Modeling uncertainty with hedged instance embeddings. In *International Conference on Learning Representations*, 2019.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Pearson, K. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572, 1901.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Sarkar, A., Vijaykeerthy, D., Sarkar, A., and Balasubramanian, V. N. A framework for learning ante-hoc explainable models via concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10286–10295, 2022.

Sheth, I., Rahman, A. A., Sevyeri, L. R., Havaei, M., and Kahou, S. E. Learning from uncertain concepts via test time interventions. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS*, 2022.

Shi, Y. and Jain, A. K. Probabilistic face embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6902–6911, 2019.

Shin, S., Jo, Y., Ahn, S., and Lee, N. A closer look at the intervention procedure of concept bottleneck models. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS*, 2022.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations*, 2014.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2018.

Yuksekgonul, M., Wang, M., and Zou, J. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023.

Zarlenga, M. E., Barbiero, P., Ciravegna, G., Marra, G., Giannini, F., Diligenti, M., Shams, Z., Precioso, F., Melacci, S., Weller, A., Lio, P., and Jamnik, M. Concept embedding models. In *Advances in Neural Information Processing Systems*, 2022.

Zhou, B., Sun, Y., Bau, D., and Torralba, A. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 119–134, 2018.

A. Usage of two anchors for concept prediction

Concept prediction can be exclusively performed using a positive anchor, where the distance between the predicted concept embedding and the positive anchor represents the probability of the concept's non-existence. This approach trains the model to generate embeddings close to the positive anchor for samples with the concept (positive samples) and far from the positive anchor for samples without the concept (negative samples). As a result, the magnitude of σ of negative samples significantly exceeds that of positive samples. Consequently, the volume of embedding becomes an unreliable estimate of uncertainty when the positive anchor is exclusively used. On the other hand, using separate positive and negative anchors provides targets for both proximity and distance, allowing σ of probabilistic embeddings to represent uncertainty reliably.

B. Discussion on data augmentation and ambiguity

CBM heavily relies on concept labels, and the accuracy and reliability of concept prediction are highly contingent on the quality of human-annotated concept labels. In addition to the incompleteness inherent in concept labels, data augmentation can exacerbate this incompleteness. Concept labels are more susceptible to being influenced by data augmentation techniques compared to class labels. For instance, in datasets like CUB, there are concepts related to specific object parts, and random cropping can inadvertently remove or obscure those parts. Similarly, concepts related to colors can be affected by strong color jittering, resulting in distorted color representation. Therefore, it is vital to employ meticulous data augmentation techniques that preserve the integrity of concept labels, considering different types of concepts.

Likewise, data augmentation introduces a wide range of images with diverse visual contexts, which can result in ambiguity issues. The introduction of ambiguity brings a challenge to reliable concept prediction, as we posed in this paper. ProbCBM effectively tackles the ambiguity issue by modeling and estimating uncertainty. By doing so, it mitigates the adverse effects of diverse visual contexts, thereby enhancing the reliability of concept prediction. Through the incorporation of uncertainty estimation, ProbCBM offers a dependable approach to concept prediction in the presence of ambiguity.

C. Details of datasets

C.1. Synthetic dataset based on MNIST dataset

We create a synthetic dataset using the MNIST dataset (LeCun et al., 2010), which consists of 60,000 images for training and 10,000 images for testing. As described in Sec. 5.1.1, each synthetic image consists of four digits, and its size becomes 56×56. Table 2 presents the 12 class labels and the corresponding concepts. The order (positions) of the four digits in each synthetic image is randomly determined. When the synthetic images are generated, one of the four digits is dropped with a probability of 0.5. There are 68,017 synthetic images for training and 11,244 synthetic images for testing, which are generated with images in the official corresponding splits. Figure 13 shows examples from the synthetic dataset.

For data augmentation, we use cutout (DeVries & Taylor, 2017) with a size 7×7.

Table 2. Pairs of class labels and concepts in the synthetic dataset based on the MNIST dataset.

CLASS	CONCEPTS
0	0, 2, 4, 6
1	0, 2, 5, 9
2	0, 3, 4, 7
3	0, 3, 6, 8
4	1, 2, 5, 7
5	1, 2, 4, 9
6	1, 3, 5, 6
7	1, 3, 7, 8
8	1, 4, 6, 8
9	3, 5, 7, 9
10	2, 4, 7, 8
11	2, 5, 6, 8

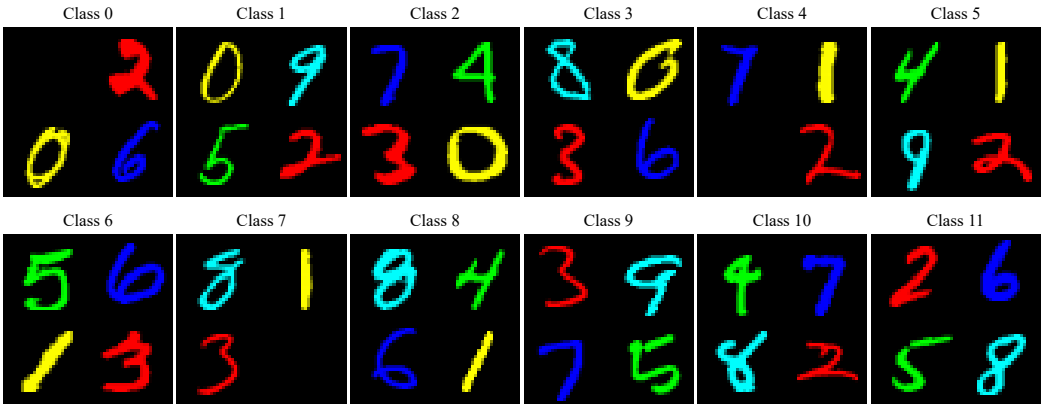


Figure 13. Example images from the synthetic dataset based on the MNIST dataset.

C.2. CUB

During training, we use color jittering, horizontal flipping, and random scaling and cropping as data augmentation methods. Because strong random scaling and cropping can produce extreme noise in concept labels, we first resize the images to 256×256 and apply random scaling and cropping with a scale of $(0.8, 1.0)$. The final size of the images for training is 224×224. When we train the model with an image size of 299×299, the images are resized to 341×341 and cropped. For inference, the images are resized to 256×256 (341×341) and center-cropped to 224×224 (299×299).

Following Koh et al. (2020), we group concepts by using a common prefix of concept names in the dataset. The groups are used in concept intervention.

C.3. AwA2

Table 3 presents the remaining 45 attributes and how we group them for concept intervention. During training, we use the same data augmentation methods that are employed for the CUB dataset, except for resizing. For the AwA2 dataset, the images are not resized before cropping. For inference, the images are resized to 224×224.

Table 3. The groups of attributes in AwA2 dataset.

GROUP	ATTRIBUTE NAME
COLOR	BLACK, WHITE, BLUE, BROWN, GRAY, ORANGE, RED, YELLOW
PATTERN	PATCHES, SPOTS, STRIPES
HAIR	FURRY, HAIRLESS
SKIN	TOUGHSKIN
SIZE	BIG, SMALL
FAT	BULBOUS, LEAN
HAND	FLIPPERS, HANDS, HOOVES, PADS, PAWS
LEG	LONG LEG
NECK	LONG NECK
TAIL	TAIL
HORN	HORNS
CLAW	CLAWS
TUSK	TUSKS
WALK	BIPEDAL, QUADRAPEDAL
LIVE	ARCTIC, COASTAL, DESERT, BUSH, PLAINS, FOREST, FIELDS, JUNGLE, MOUNTAINS, OCEAN, GROUND, WATER, TREE, CAVE

D. Experimental details

D.1. Details for ProbCBM

For the synthetic dataset, we set D_c and D_y as 16 and 32, respectively. For the real-world datasets, D_c and D_y are set as 16 and 128, respectively. N_s is set as 50. With the synthetic dataset, we train the concept predictor for 30 epochs and the class predictor for 20 epochs. With the real-world datasets, we train the concept predictor for 50 epochs and the class predictor for 20 epochs. For stable training, we fix the pretrained weights with ImageNet-1K (Russakovsky et al., 2015) for the first five epochs. We initialize a and d, which are learnable parameters for scaling given by Eqs. 3 and 5, as 5 and 10, respectively.

We use AdamP optimizer (Heo et al., 2021) with the cosine learning rate scheduler (Loshchilov & Hutter, 2017). The learning rate is set to 10^{-3} for the pretrained weights and 10^{-2} for the randomly initialized weights and learnable parameters. λ_{KL} is set as 5×10^{-5}. All experiments are implemented using PyTorch (Paszke et al., 2017).

D.2. Details for other models

ResNet18 is used as a backbone in all models. For CBM, we incorporate an FC layer after the backbone to build a concept predictor. In addition, we utilize a separate FC layer as a class predictor. To apply MC dropout to CBM, we add a dropout layer with a dropout rate of 0.2 after every convolutional block in a backbone. During inference, the predictions were obtained with 50 samples same as ProbCBM. We train CBM and CBM with MC dropout in the same manner as our training scheme. The concept predictor outputs the probabilities that each concept exists, and the probabilities that the concept predictor generates are fed into the class predictor. We first train the concept predictor and then train the class predictor in a sequential way. During the training of the class predictor, the output of the concept predictor is replaced with a ground-truth concept label (1 and 0 for positive and negative labels, respectively) with probability p_{replace}. Equal to the case of training ProbCBM, p_{replace} is set as 0.5. CEM is trained with the official code[2]. Please note that we use the same data augmentation techniques across all models while keeping other configurations at their default settings.

E. Training scheme

Algorithm 1 provides detailed steps of our training scheme for the class predictor, which is described in Sec. 4.3.

Algorithm 1 Training Scheme of Class Predictor

Input: Training data $\mathcal{D}$, Trained concept predictor
repeat
  for $x \in \mathcal{D}$ **do**
    for concept $c \in \{c_1, c_2, ..., c_{N_c}\}$ **do**
      Get $p(z_c|x)$ from concept predictor
      for $n = 1$ **to** N_s **do**
        Initialize representation set $\mathcal{Z}^{(n)} = \{\}$
        Sample $z_c^{(n)} \sim p(z_c|x)$
        Sample replace strategy $r \sim Bernoulli(p_{\text{replace}})$
        if $r = true$ **then**
          Append $\mathbf{1}(c = 1)z_c^+ + \mathbf{1}(c = 0)z_c^-$ to $\mathcal{Z}^{(n)}$
        else
          Append $z_c^{(n)}$ to $\mathcal{Z}^{(n)}$
        end if
      end for
      Concatenate $\mathcal{Z}^{(n)}$ and get $h^{(n)}$ with Eq. 4
    end for
    Get $p(y_k = 1)$ with Eq. 5
    Calculate $\mathcal{L}_{\text{class}}$ and update the class predictor with $\mathcal{L}_{\text{class}}$
  end for
until the class predictor converges

[2] https://github.com/mateoespinosa/cem

F. Additional experimental results

F.1. Ablation studies

We conduct ablations studies on the effect of the probability p_{replace} and the embedding dimensions D_c and D_y. The experiments are conducted with the CUB dataset.

Probability p_{replace}. Figure 14 shows the classification performance with respect to p_{replace}. As shown, there is no significant difference except when p_{replace} is 1. Training with $p_{\text{replace}} = 1$ is the same as the independent learning strategy in CBM, wherein the class predictor is trained with ground-truth concept and class labels regardless of the concept predicted by the concept predictor.

Embedding dimensions D_c and D_y. Figure 15 shows the concept prediction and classification performance with respect to the embedding dimensions D_c and D_y. D_c is the dimension of the concept embedding space and D_y is the dimension of the class embedding space. The concept prediction performance is the same in the ablation study on D_y. The performance is not significantly affected by D_c or D_y. Additionally, we compare the proportions of samples that exhibit increased uncertainty after occlusion across different values of D_c, as depicted in Figure 16. The results indicate that when the concept embedding dimension D_c is small ($D_c = 4$), the ability to detect increased uncertainty diminishes. However, as D_c increases over a certain threshold, it shows convergence. This finding indirectly suggests that utilizing high-dimensional concept embeddings has a positive impact on enhancing the ability for uncertainty estimation.

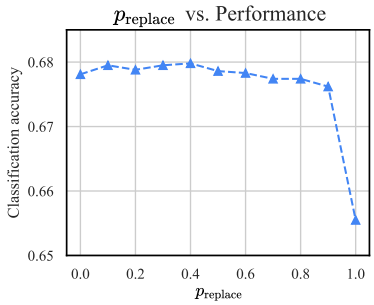


Figure 14. Classification performances of ProbCBM trained with different p_{replace} for the CUB dataset.

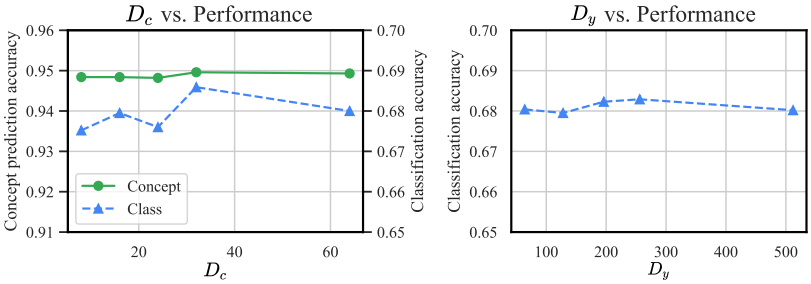


Figure 15. Concept prediction and classification performance of ProbCBM trained with different D_c and D_y values for the CUB dataset.

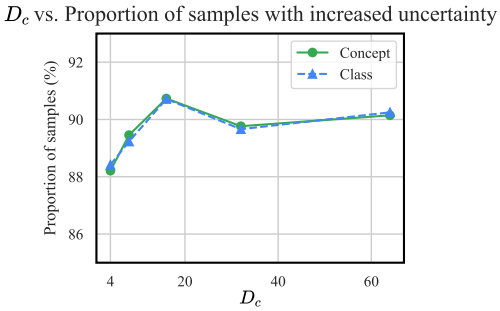


Figure 16. Proportion (%) of samples whose uncertainty increases after occlusion of ProbCBM trained with different D_c values for the CUB dataset.

F.2. Additional results

Figure 17 presents the results of occlusion experiments conducted on the AwA2 dataset, similar to those performed on the CUB dataset in Section 5.2.2. We generate occlusion by covering a patch of size 64×64 with gray color at the center of the images. The figure illustrates the proportion of samples that exhibit increased uncertainty after occlusion. The results demonstrate the effectiveness of ProbCBM in detecting an increase in both concept and class uncertainties across a diverse set of samples.

Figure 18 shows examples of the transformed images for the analysis presented in Sec. 5.2.3. Figure 19 presents the changes in concept uncertainty after image transformations. As shown, the concept uncertainty increases for most samples. As the degree of image transformation increases, the proportion of samples with significantly increased uncertainty increases.

As shown in Figure 20, we visualize class embeddings in the 2D space, with the four images presented in Figure 9. The images with large concept uncertainties have large uncertainties in classification.

Figure 21 shows the correlation between the performance and the estimated uncertainty for the AwA2 dataset. Figure 22 shows the results of concept intervention for the AwA2 dataset. As described in Sec. 5.2.4, we intervene in the concepts in the same group at once (15 groups for the AwA2 dataset).

F.3. Additional examples

Figure 23 shows additional examples of changes in the predicted concept uncertainty of ProbCBM after occlusion of parts of images for the synthetic dataset. Figures 24 and 25 show additional examples of ProbCBM's prediction for the real-world datasets.

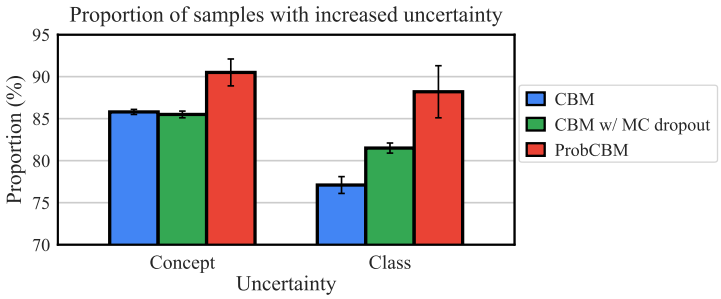


Figure 17. Proportion (%) of samples whose uncertainty increases after occlusion for the AwA2 dataset. Results include mean values with standard deviation from three experiment repetitions.

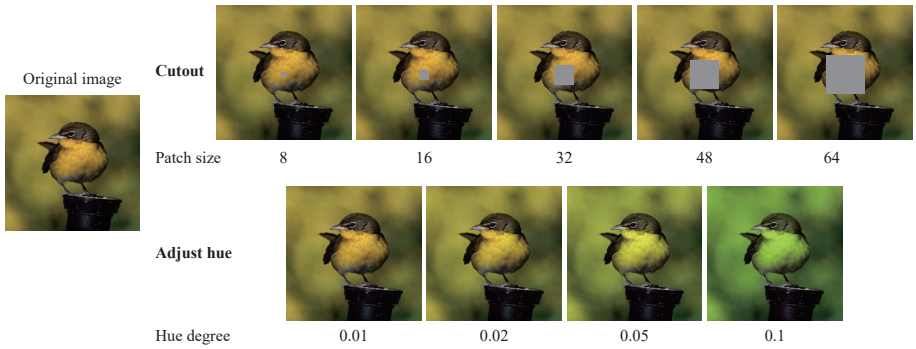


Figure 18. Examples of the resulting images after image transformations.

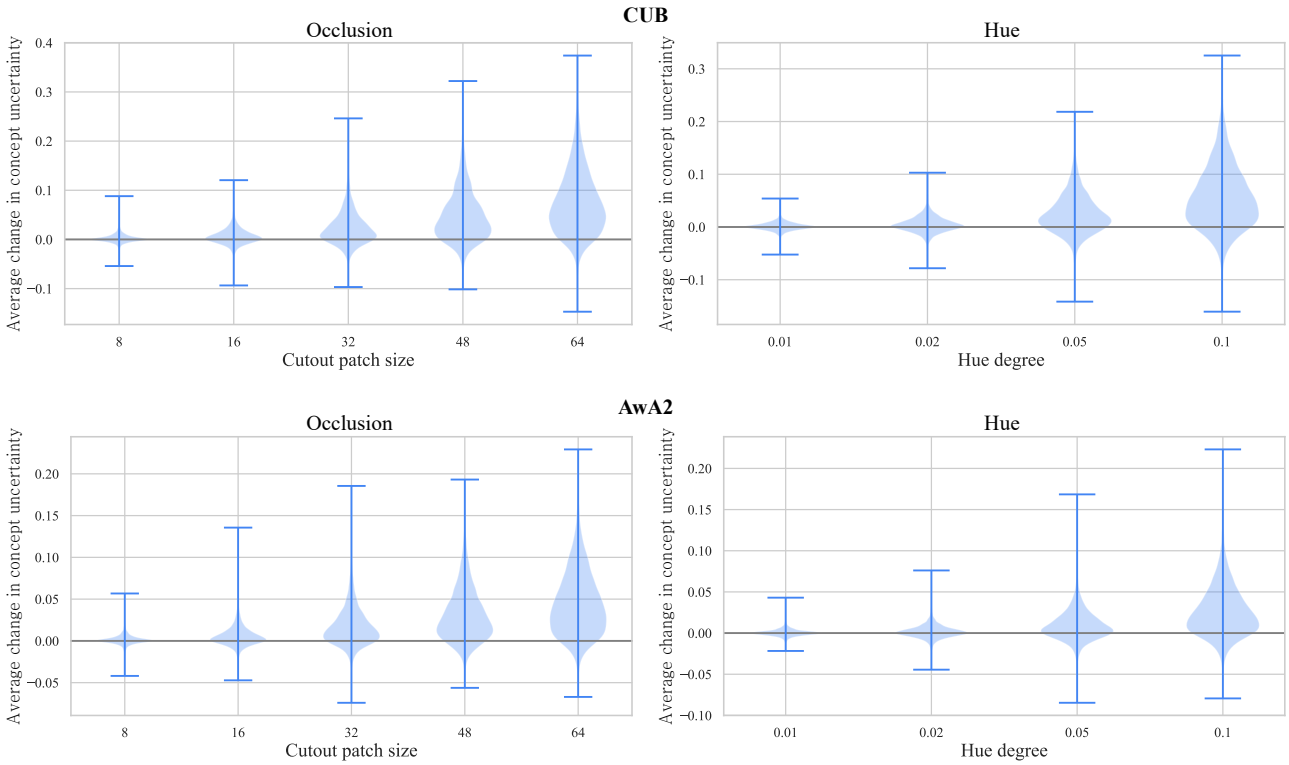


Figure 19. Violin plots of changes in the concept uncertainty after image transformations.

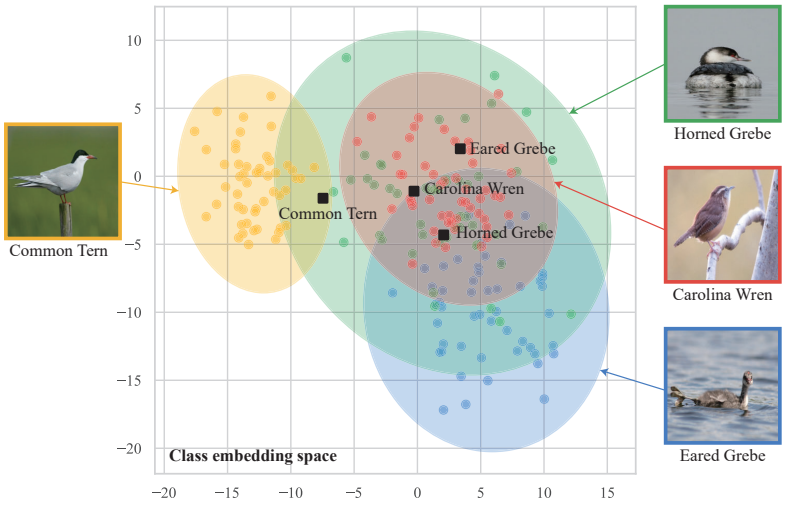


Figure 20. Visualization of the class embedding space.

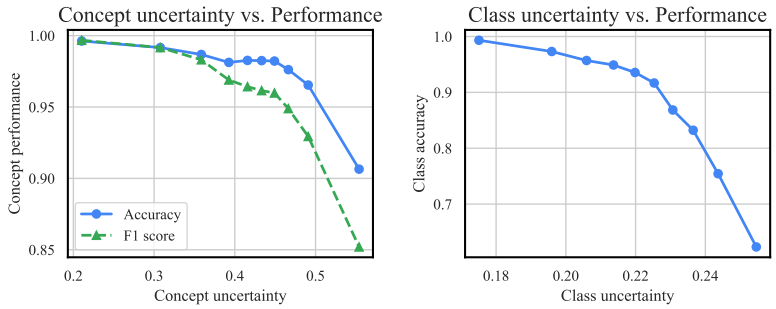


Figure 21. Correlation between the performance and uncertainty for the AwA2 dataset. A point represents one group of samples with similar uncertainties, and the x-axis indicates the median uncertainty for each group.

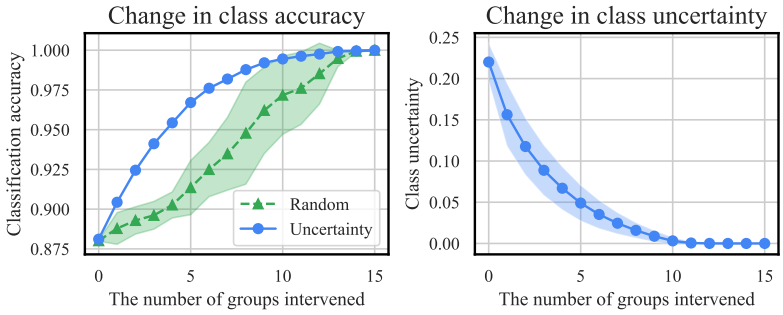


Figure 22. Results of concept intervention for the AwA2 dataset. Intervention in a random order is conducted five times and is visualized with the means and standard deviations. On the right plot, intervention is conducted in an uncertainty-based order, and the means and standard deviations of the class uncertainty for all test samples are shown.

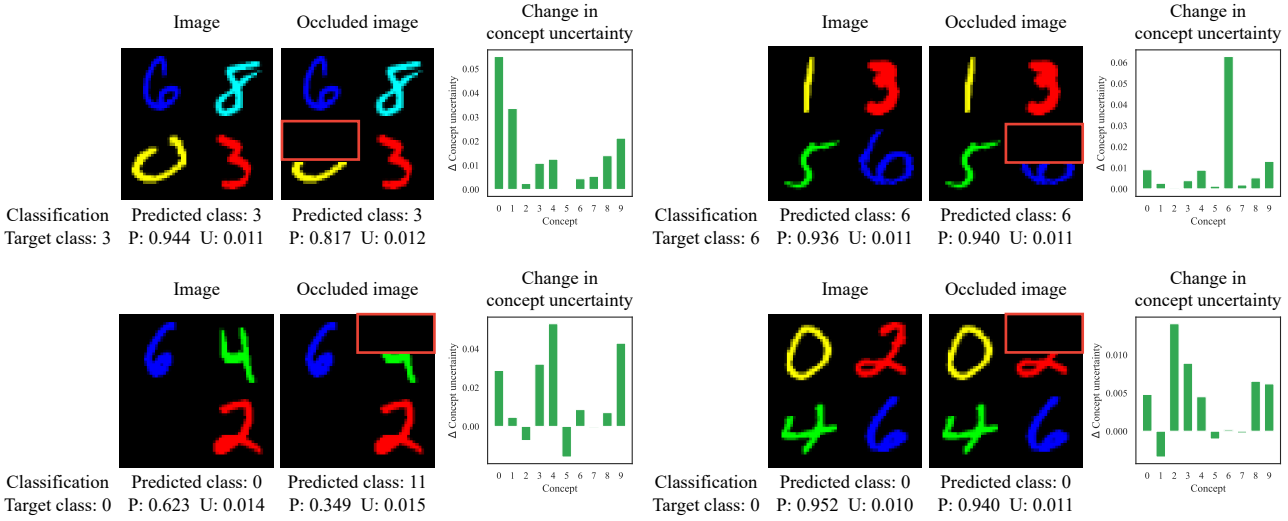


Figure 23. Examples of changes in the concept and class uncertainties after occlusion of parts of images. P and U present the probability and uncertainty, respectively. Red bounding boxes denote the occluded parts.

Target class: Hooded Oriole

Concept Prediction	P	U	P	U	P	U
wing color: white	0.993	0.285	0.003	**0.320**	0.999	0.176
belly color: yellow	0.883	0.278	0.999	**0.294**	1.000	0.122
underparts color: yellow	0.965	**0.268**	0.999	0.207	1.000	0.089
Class Prediction	Hooded Oriole P: 0.940 U: 3.710		Hooded Oriole P: 0.156 U: 4.209		Hooded Oriole P: 0.973 U: 2.845	

Target class: Baltimore Oriole

Concept Prediction	P	U	P	U
wing color: black	0.984	**0.418**	0.999	0.247
throat color: black	0.871	**0.310**	0.999	0.232
crown color: black	0.011	**0.387**	0.999	0.147
Class Prediction	Baltimore Oriole P: 0.648 U: 3.989		Baltimore Oriole P: 0.958 U: 2.904	

Target class: Black billed Cuckoo

Concept Prediction	P	U	P	U	P	U
wing shape: pointed-wings	0.901	**0.447**	0.998	0.226	0.999	0.149
back color: brown	0.680	**0.486**	0.996	0.409	0.998	0.299
upper tail color: brown	0.951	**0.335**	0.998	0.171	0.999	0.122
Class Prediction	Yellow billed Cuckoo P: 0.612 U: 4.210		Black billed Cuckoo P: 0.906 U: 3.580		Black billed Cuckoo P: 0.900 U: 3.018	

Target class: American Redstart

Concept Prediction	P	U	P	U
wing color: black	0.257	**0.532**	0.999	0.170
upperparts color: black	0.890	**0.473**	1.000	0.168
breast pattern: multi-colored	0.001	**0.256**	0.995	0.170
Class Prediction	Blue Headed Vireo P: 0.259 U: 4.125		American Redstart P: 0.734 U: 2.525	

Target class: Blue Jay

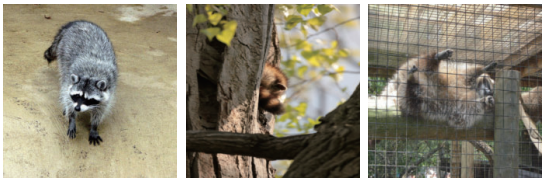
Concept Prediction	P	U	P	U	P	U
forehead color: blue	0.996	0.217	0.001	**0.293**	0.999	0.135
under tail color: white	0.979	0.304	0.008	**0.393**	0.996	0.223
leg color: grey	0.999	0.190	0.026	**0.229**	1.000	0.098
Class Prediction	Blue Jay P: 0.945 U: 3.947		Mockingbird P: 0.821 U: 4.134		Blue Jay P: 0.978 U: 2.581	

Target class: Bobolink

Concept Prediction	P	U	P	U
wing color: black	0.999	0.188	0.992	**0.355**
upperparts color: black	0.997	**0.271**	0.996	0.270
breast pattern: multi-colored	1.000	0.171	0.993	**0.285**
Class Prediction	Bobolink P: 0.968 U: 2.860		Bobolink P: 0.560 U: 3.812	

Figure 24. Examples of predictions of ProbCBM on the CUB dataset. P and U denote probability and uncertainty, respectively. The highest value of uncertainty in each concept is bolded.

Target class: Raccoon



Concept Prediction	P	U	P	U	P	U
gray	1.000	0.275	0.943	0.524	0.963	**0.634**
spots	1.000	0.068	0.053	0.428	0.005	**0.469**
paws	1.000	0.204	0.732	**0.285**	0.987	0.226
Class Prediction	Raccoon P: 0.999 U: 0.177		Squirrel P: 0.374 U: 0.235		Spider monkey P: 0.972 U: 0.232	

Target class: Weasel

Concept Prediction	P	U	P	U
brown	0.998	**0.381**	0.999	0.335
tail	0.999	**0.474**	1.000	0.318
forest	0.999	0.396	0.997	**0.423**
Class Prediction	Spider monkey P: 0.920 U: 0.218		Weasel P: 0.845 U: 0.210	

Target class: Walrus

Concept Prediction	P	U	P	U	P	U
toughskin	0.999	0.330	1.000	0.392	0.965	**0.621**
tusks	1.000	0.168	0.952	0.345	0.925	**0.422**
arctic	1.000	0.231	0.999	0.281	0.987	**0.437**
Class Prediction	Walrus P: 0.998 U: 0.185		Walrus P: 0.979 U: 0.204		Walrus P: 0.992 U: 0.249	

Target class: Sheep

Concept Prediction	P	U	P	U
white	0.999	**0.386**	1.000	0.206
horns	0.000	**0.353**	0.000	0.187
plains	0.999	**0.400**	1.000	0.147
Class Prediction	Sheep P: 1.000 U: 0.209		Sheep P: 1.000 U: 0.192	

Figure 25. Examples of predictions of ProbCBM on the AwA2 dataset. P and U denote probability and uncertainty, respectively. The highest value of uncertainty in each concept is bolded. The labels of the most of concepts are positive except for underlined concepts.