Semi-supervised Concept Bottleneck Models

Jeeon Bae [1] **Sungbin Shin** [2] **Namhoon Lee** [2]

Abstract

Concept bottleneck models (CBMs) enhance the interpretability of deep neural networks by adding a concept layer between the input and output layers. However, this improvement comes at the cost of labeling concepts, which can be prohibitively expensive. To tackle this issue, we develop a semi-supervised learning (SSL) approach to CBMs that can make accurate predictions given only a handful of concept annotations. Our approach incorporates a strategy for effectively regulating erroneous pseudo-labels within the standard SSL approaches. We conduct experiments on a range of labeling scenarios and present that our approach can reduce the labeling cost quite significantly without sacrificing the prediction performance.

1. Introduction

Deep neural networks are becoming a critical component of modern computing (Jordan & Mitchell, 2015; LeCun et al., 2015). Despite their widespread usage, however, it still remains challenging to explain the inner-workings of their decision-making process (Gilpin et al., 2018). One way to remedy this issue is to introduce human-readable concepts and make use of them at inference (Kim et al., 2018; Koh et al., 2020).

Concept bottleneck models (CBMs) are the first approach to utilize the idea of concept-based predictions (Koh et al., 2020). Specifically, CBMs secure interpretability by having a layer of concepts in the network, by which the model first makes prediction for the high-level concepts (*e.g.*, wing color of a bird) and then predicts the target class (*e.g.*, a bird species) based on the predicted concept values. This stands in stark contrast to the standard end-to-end neural network models where the learned representations used to make predictions are not necessarily interpretable.

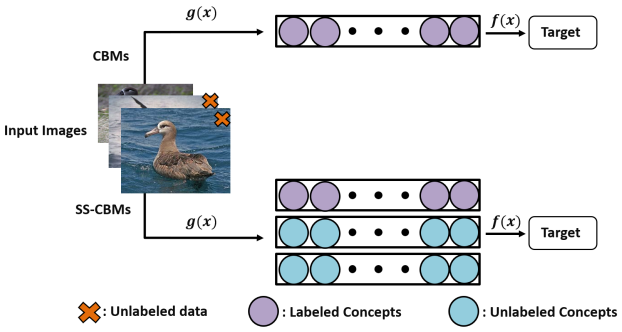

Figure 1. Comparison between the standard CBMs and SS-CBMs. Unlike standard CBMs, SS-CBMs can utilize unlabeled data.

However, CBMs require the ground-truth annotations for concepts to be trained. Compared to the standard end-to-end learning models, this labeling cost can be quite significant, which undermines their utility in practice.

In this work, we suggest a new way to tackle this issue via semi-supervised learning (SSL), *i.e.*, learning with only a small number of labeled examples while having access to potentially a large amount of unlabeled data (see Figure 1). To be more specific, we propose a semi-supervised concept bottleneck model (SS-CBM) that can learn the underlying relationship between concepts and target distributions as a form within existing state-of-the-art SSL approaches, such that it could regulate erroneous pseudo-labels.

We demonstrate that our approach can significantly reduce the labeling cost required to learn CBMs while maintaining the prediction accuracy. In particular, we show that with only a single labeled example per class SS-CBM can achieve more than 50% prediction accuracy for a 200-way classification task. Also, using only 20% of labeling SS-CBM shows its competitive prediction performance against the case of learning with fully labeled data.

2. Related Work

While CBMs are interpretable and intervenable (Shin et al., 2023), it has the clear disadvantage of demanding concept

[1]Industrial & Management Systems Engineering, University of Kyung Hee, Seoul, Korea. his work was done during the internship at POSTECH [2]Computer Science and Engineering, POSTECH, Pohang, Korea. Correspondence to: Jeeon Bae <jeeon98b@gmail.com>.

AI & HCI Workshop at the 40th International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

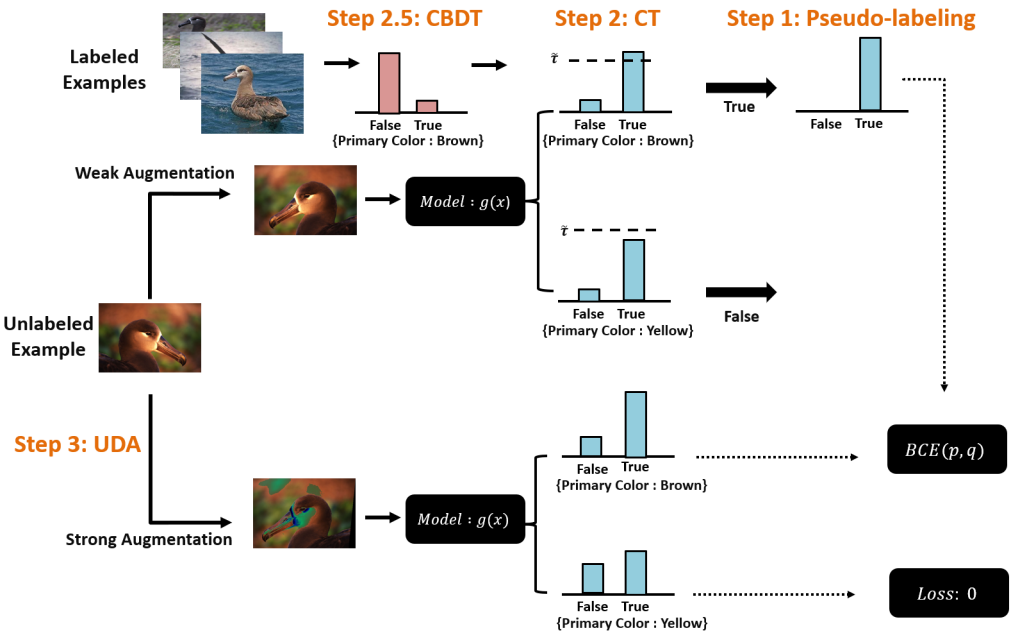

Figure 2. Overview of SS-CBM. The algorithm applies the following 3 steps for the concept-unlabeled data: step 1 where it generates pseudo concept labels for the weakly augmented inputs, step 2 where the dynamic confidence threshold is calculated to regulate erroneous pseudo-labels, and step 3 where it applies consistency regularization between the weakly- and strongly-augmented inputs.

labels by human experts. A few recent works have suggested addressing this issue via automatically generating the concepts and their annotations using generative models; one approach is P-CBMs (Yuksekgonul et al., 2023), which use the so-called concept activation vector (Kim et al., 2018) or multi-modal CLIP models (Radford et al., 2021); another approach, LF-CBMs (Oikarinen et al., 2023), uses large language processing models (Brown et al., 2020).

However, the concepts generated by aforementioned approaches are not necessarily reliable in practice. For example, these models often fail to generate accurate concepts for highly complex domains such as medical imaging. Also, they may require a nontrivial pre-processing step to filter out noisy and redundant concept annotations. In contrast, we make use of unlabeled data in the form of a semi-supervised learning in order to reduce the labeling cost, which is arguably much easier to access in practical scenarios.

3. Preliminary

3.1. Concept Bottleneck Models

CBMs have two-step prediction mechanisms. Firstly, they use a neural network, denoted as $g : \mathbb{R}^d \to \mathbb{R}^k$, to predict vector of k binary concepts $c \in \mathbb{R}^k$ from input $x \in \mathbb{R}^d$. Secondly, they use a neural network denoted as $f : \mathbb{R}^k \to \mathbb{R}$ to predict the target $y \in \mathbb{R}$ from the predicted concept.

The learning process of CBMs aims to find the optimal functions $\hat{g}$ and $\hat{f}$. These functions are determined by solving the following optimization problems, where L_c represents Binary Cross Entropy Loss (BCE) and L_y represents Cross Entropy Loss (CE), given m data points:

$$\hat{g} = \arg\min_{g} \sum_{n=1}^{m} \sum_{k=1}^{K} L_c(g(x_n^k), c_n^k)$$

$$\hat{f} = \arg\min_{f} \sum_{n=1}^{m} L_y(f(c_n), y_n).$$

As you can see, CBMs require not only a single task label per image but also k concept labels. This can be extremely expensive to annotate, especially for data that require domain-specific knowledge for labeling the concepts (*e.g.*, medical images).

To address these challenges, we propose semi-supervised CBMs (SS-CBMs). SS-CBMs can be trained with only a handful of annotated data, even when concept labels or both concept labels and target labels do not exist.

3.2. FixMatch

There are many exceptional SSL methods, but we select FixMatch (Sohn et al., 2020) as the cornerstone for integrating SSL into CBMs because of its outstanding performance, simplicity, and wide use of state-of-the-art SSL methods.

FixMatch can be decomposed into three core algorithms: Pseudo-labeling (Lee et al., 2013), Confidence Threshold-

ing, and Unsupervised Data Augmentation (Xie et al., 2020).

Pseudo-labeling is a technique for generating artificial labels. It works by taking the $\arg\max$ of the probability distribution predicted by a model for unlabeled data.

Confidence thresholding is a technique that uses a predefined threshold to determine which unlabeled data points to use to train a model. If the confidence exceeds the threshold, then the pseudo-label and the corresponding data point are used to train the model.

Unsupervised Data Augmentation is a consistency regularization method used in SSL that applies augmentation to unlabeled data in two directions: one with weak augmentation and the other with strong augmentation. It then calculates the consistency loss between the predictions obtained from these two augmented inputs.

4. Semi-supervised CBMs

In this section, we explain our framework SS-CBM which applies semi-supervised learning approaches to CBMs (see Figure 2 for the overview of the algorithm). We consider the two labeling scenarios for SS-CBM: "easy case" and "hard case" depending on the availability of the class labels for the concept-unlabeled dataset. We denote a concept-labeled dataset of n samples as $\mathcal{X}_\mathcal{L} = \{(x_i, c_i, y_i)\}_{i=1}^n$ and a concept-unlabeled dataset of m samples as $\mathcal{X}_\mathcal{U} = \{(x_j, y_j)\}_{j=1}^m$ (in easy case) or $\mathcal{X}_\mathcal{U} = \{(x_j)\}_{j=1}^m$ (in hard case).

4.1. Step 1: Pseudo concept labeling

Our algorithm first generates pseudo concept labels for $\mathcal{X}_\mathcal{U}$ by applying weak augmentation α_w to the input. Specifically, the pseudo label for k-th concept of j-th sample is calculated as $\hat{c}_j^k = \mathbb{1}[q_j^k > 0.5]$ where $q_j^k = p(c_j^k|\alpha_w(x_j))$ represents the corresponding concept prediction value.

4.2. Step 2: Confidence Tresholding

In SS-CBM, whether to include an unlabeled data point in the training process is based on the absolute difference between 0.5 and q_j^k. If this absolute difference exceeds a threshold τ (the value between 0 and 0.5), the data point x_j is included in the training process. Then, the unsupervised loss L_u with confidence thresholding can be expressed as follows:

$$L_u = \frac{1}{m}\sum_{j=1}^{m}\sum_{k=1}^{K}\mathbb{1}[|0.5 - q_j^k| > \tau] \cdot L_c(q_j^k, \hat{c}_j^k)$$

4.2.1. Step 2.5: Class-based Dynamic Thresholding

The quality of step 2 can be improved if the class labels are provided for $\mathcal{X}_\mathcal{U}$ as in the easy case. We propose a method called Class-Based Dynamic Thresholding (CBDT) to leverage this additional information by adopting dynamic threshold values rather than a fixed value in FixMatch. The method is based on the assumption that concept values within the same class should be more similar to each other than those from different classes. CBDT consists of the following three steps:

Investigating: It investigates the probability distributions of each concept c within each class y in $\mathcal{X}_\mathcal{L}$. We denote the set of all probability distributions as V.

Searching: It searches for the probability distribution $P(c_i^j|y_i)$ in V that corresponds to the i-th class and j-th concept. We denote the identified concept probability distribution's $\arg\max$ as $\tilde{c}_i^j$ and its true probability as ρ_i^j.

Determining threshold: It dynamically determines the confidence threshold based on ρ_i^j.

Taken together, the CBDT is formulated as follows:

$$\tilde{\tau} = \begin{cases} \tau + \rho_i^j(0.5 - \tau) & \text{if } \hat{c}_i^j \neq \tilde{c}_i^j \\ \tau + (1 - \rho_i^j)(0.5 - \tau) & \text{otherwise.} \end{cases}$$

We remark that $\tilde{\tau}$ becomes larger when $\hat{c}_i^j \neq \tilde{c}_i^j$ thus making the concept difficult to be included in the training process. The high-level idea behind this is that pseudo-concept label $\hat{c}_i^j$ is more likely to be incorrect in this case; hence, excluding these concepts from the training process would increase the model performance.

In the hard case, however, the class labels for $\mathcal{X}_\mathcal{U}$ are not available. To apply our approach in this situation, we generate pseudo class labels $\hat{y}$ from the auxiliary end-to-end neural network which predicts the class label from the inputs. The auxiliary network is trained with $\mathcal{X}_\mathcal{L}$ and $\hat{y}$ is taken as the $\arg\max$ of the predicted class values.

4.3. Step 3: Unsupervised Data Augmentation

Finally, we apply consistency regularization between the weakly- and strongly-augmented samples.

In CBMs, if we denote the predicted probability distribution with strong augmentation applied as $\hat{q}_j^k = p(c_j^k|\alpha_s(x_j))$, the expression for the unsupervised loss L_u with UDA + CBDT can be expressed as follows:

$$L_u = \frac{1}{m}\sum_{j=1}^{m}\sum_{k=1}^{K}\mathbb{1}[|0.5 - q_j^k| > \tilde{\tau}] \cdot L_c(\hat{q}_j^k, \hat{c}_j^k)$$

After the above steps, the total loss is calculated as $L = L_s + \lambda L_u$ where L_s, L_u each represent the supervised and unsupervised loss and λ is a hyperparameter. By minimizing this total loss, we can find the optimal $\hat{g}$.

Method	Target accuracy (%)					Concept accuracy (%)				
	$n=1$	$n=2$	$n=3$	$n=6$	$n=30$	$n=1$	$n=2$	$n=3$	$n=6$	$n=30$
CBM (No SSL)	17.47	28.27	36.50	51.07	76.70	87.09	89.07	90.35	92.83	96.60
CBM w/ FixMatch	15.48	26.48	39.25	56.83	–	87.60	89.18	91.05	93.80	–
SS-CBM (easy)	**53.33**	**65.46**	**68.32**	**71.40**	–	**94.03**	**95.50**	**95.80**	**96.30**	–
SS-CBM (hard)	**19.34**	**38.91**	**49.74**	**64.16**	–	**88.95**	**92.05**	**93.48**	**95.27**	–

Table 1. Target and concept accuracy of different models for the CUB. Here, n represents the number of labeled examples per class ($n = 30$ corresponds to the fully supervised case). The results are the mean over 3 different random seeds.

Model	$n=1$	$n=2$	$n=3$	$n=6$
EfficientNet V2	30.00	44.41	51.81	63.76

Table 2. Target accuracy of the auxiliary network for the CUB. Here, n represents the number of labeled examples per class. The results are the mean over 3 different random seeds.

5. Experiments

5.1. Setup

Dataset We evaluate SS-CBM on the CUB dataset (Welinder et al., 2010) which is used as the standard benchmark to study CBMs. The dataset contains $11,788$ images of 200 bird species classes, $5,994$ for training and $5,794$ for testing. Each class has 30 labeled images, and each image has fine-grained annotations for 112 concepts after pre-processing. We follow Koh et al. (2020) to pre-process the dataset as with other existing work. We also perform data augmentation using the same strategy as in Sohn et al. (2020).

Labeling Denoting the number of labeled images per class as n, we vary this number to be $n = 1, 2, 3, 6, 30$ in order to simulate the scenario of a limited labeling budget. We let a SSL-based approach to make use of the rest of unlabeled images. This means that, when $n = 1$ for example, a SSL approach is provided with 1 labeled image per class and 29 unlabeled images per class.

Model We use the Inception V3 model (Szegedy et al., 2016) as the backbone architecture, which is pre-trained on the ImageNet dataset (Deng et al., 2009). We also use the EfficientNet V2 Large model (Tan & Le, 2021) as the auxiliary network for generating pseudo-labels. This model is first pre-trained on ImageNet and then fine-tuned on the target dataset for 30 epochs.

Hyperparameter: In CBMs with FixMatch and SS-CBM use an identical set of hyperparameters. The losses for labeled and unlabeled data are given equal weight (*i.e.* $\lambda = 1$). The batch size for the unlabeled dataset is set to 64, while the batch size for the labeled dataset is 32. Additionally, the threshold value τ is set to 0.4.

5.2. Evaluation of SS-CBM

We evaluate SS-CBM for varying labeling cost and present the results in Table 1. We find that SS-CBM consistently outperforms the other baselines across all settings. Specifically in the easy case, our algorithm achieves a target accuracy of 53.33%, which is 35.85% higher than fully supervised CBMs when only a single label per class was used (*i.e.*, 3.3% of total labeling cost). In addition, SS-CBM performs relatively closely to the fully supervised CBMs, when using only 6 labels per class (*i.e.*, 20% of total labeling cost).

5.3. Analysis of SS-CBM

We observe that SS-CBM improves as the accuracy of the pseudo-label for the class provided by the auxiliary network increases. The knowledge about the relationship between concept and target can still be helpful even when the target accuracy of the auxiliary network is low (see table 2). This improvement can be attributed to two key factors:

Indirect use: SS-CBMs do not directly use class knowledge. Instead, they indirectly incorporate it into the threshold. This means that SS-CBMs can prevent the training of accurate pseudo-label in the worst-case scenario. When we associate this with the fact that SS-CBM yields better results than CBMs with FixMatch, it becomes evident that prioritizing the avoidance of inaccurate pseudo-labels is more crucial, even if it leads to a smaller amount of unlabeled data being learned.

Similar class has similar features: We speculate that even when predicting an incorrect pseudo class label, the model would tend to predict classes with features similar to the correct class label. Consequently, similar classes are likely to have comparable concept labels, leading to similarity with the concept set of the correct class.

5.4. Target Accuracy vs Concept Accuracy

As we can see from Table 1, a high concept accuracy does not always guarantee a high target accuracy, especially when the concept accuracy is low. This finding suggests that there are certain concepts or combinations of concepts that are more important for predicting certain classes than others.

6. Conclusion

We address the problem of high annotation cost in learning CBMs via a semi-supervised learning approach. To this end, we propose a novel CBM called SS-CBM, which combines existing semi-supervised learning methods with techniques to regulate inaccurate pseudo-labels by utilizing the essence of CBMs. Our approach not only encompasses various label scenarios but also offers valuable insights on effectively utilizing unlabeled data for CBMs. As a result, SS-CBM are able to attain a target accuracy of 93% and a concept accuracy of 99.6% with a mere 20% subset of the full dataset.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH), and No.2022-0-00959, (part2) Few-Shot learning of Causal Inference in Vision and Language for Decision Making)

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *NeurIPS*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. 2009.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. Explaining explanations: An overview of interpretability of machine learning. *DSAA*, 2018.

Jordan, M. I. and Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, 2015.

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). *ICML*, 2018.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. *ICML*, 2020.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 2015.

Lee, D.-H. et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on challenges in representation learning, ICML*, 2013.

Oikarinen, T., Das, S., Nguyen, L. M., and Weng, T.-W. Label-free concept bottleneck models. *ICLR*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. *ICML*, 2021.

Shin, S., Jo, Y., Ahn, S., and Lee, N. A closer look at the intervention procedure of concept bottleneck models. *arXiv preprint arXiv:2302.14260*, 2023.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 2020.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *CVPR*, 2016.

Tan, M. and Le, Q. Efficientnetv2: Smaller models and faster training. *ICML*, 2021.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-ucsd birds 200. 2010.

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. *NeurIPS*, 2020.

Yuksekgonul, M., Wang, M., and Zou, J. Post-hoc concept bottleneck models. *ICLR*, 2023.