Partially Shared Concept Bottleneck Models

Delong Zhao[1,*], **Qiang Huang**[1,*], **Di Yan**[1], **Yiqun Sun**[2], **Jun Yu**[1,3,†]

[1]Harbin Institute of Technology (Shenzhen), Shenzhen, China
[2]National University of Singapore, Singapore,
[3]Peng Cheng Laboratory, Shenzhen, China
{zhaodelong,yandi}@stu.hit.edu.cn, {huangqiang,yujun}@hit.edu.cn, sunyq@comp.nus.edu.sg

Abstract

Concept Bottleneck Models (CBMs) enhance interpretability by introducing a layer of human-understandable concepts between inputs and predictions. While recent methods automate concept generation using Large Language Models (LLMs) and Vision-Language Models (VLMs), they still face three fundamental challenges: poor visual grounding, concept redundancy, and the absence of principled metrics to balance predictive accuracy and concept compactness. We introduce **PS-CBM**, a **P**artially **S**hared **CBM** framework that addresses these limitations through three core components: (1) a multimodal concept generator that integrates LLM-derived semantics with exemplar-based visual cues; (2) a Partially Shared Concept Strategy that merges concepts based on activation patterns to balance specificity and compactness; and (3) Concept-Efficient Accuracy (CEA), a post-hoc metric that jointly captures both predictive accuracy and concept compactness. Extensive experiments on eleven diverse datasets show that PS-CBM consistently outperforms state-of-the-art CBMs, improving classification accuracy by 1.0%–7.4% and CEA by 2.0%–9.5%, while requiring significantly fewer concepts. These results underscore PS-CBM's effectiveness in achieving both high accuracy and strong interpretability.

Code — https://github.com/7494zdl/PS-CBM

1 Introduction

Deep neural networks have achieved remarkable success across a wide range of domains, including computer vision, natural language processing, and speech recognition. Yet, their opaque decision-making process poses a critical barrier to deployment in high-stakes domains such as healthcare and autonomous driving (Chauhan et al. 2023). A promising direction for enhancing interpretability is the Concept Bottleneck Model (CBM) (Koh et al. 2020), which inserts an intermediate layer of human-understandable concepts between inputs and outputs.

While concept-based modeling improves transparency, most early CBMs rely on manually annotated concepts curated by domain experts, which is labor-intensive and difficult to scale (Sawada and Nakamura 2022; Yun et al. 2023; Yuksekgonul, Wang, and Zou 2023). To address this, recent work automates concept construction using either LLMs to generate class-specific semantic descriptions (Yang et al. 2023; Oikarinen et al. 2023; Yan et al. 2023; Srivastava, Yan, and Weng 2024), or VLMs to select visual concepts based on image-text alignment (Rao et al. 2024; He et al. 2025b). These methods reduce annotation effort and perform competitively at scale. However, as summarized in Table 1, they still fall short in addressing three fundamental challenges:

- **Poor Visual Grounding.** LLM-generated concepts offer semantic richness but often lack alignment with actual visual content (Yang et al. 2023; Oikarinen et al. 2023). Conversely, VLM-based methods improve visual fidelity but sacrifice class-level semantic coherence and incur higher computational costs (Rao et al. 2024; He et al. 2025b). It reflects a persistent semantic-visual gap that weakens both accuracy and interpretability.
- **Concept Redundancy.** As depicted in Figure 1(a) and Figure 1(b), some methods generate concepts independently for each class, resulting in semantic duplication

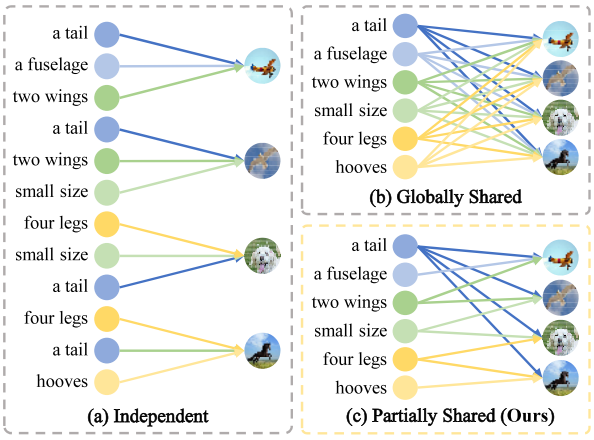

Figure 1: Illustration of different concept sharing strategies: (a) Independent, where redundant concepts exist across classes; (b) Globally Shared, where predictions are affected by irrelevant concepts; and (c) Partially Shared (Ours), which reduces the number of concepts while avoiding interference from irrelevant ones.

*Equal contribution.
†Corresponding author.

Method	Semantic-Visual Grounding	Low Concept Redundancy	Principled Metrics
LaBo (Yang et al. 2023)	×	×	△
LF-CBM (Oikarinen et al. 2023)	×	×	△
LM4CV (Yan et al. 2023)	△	✓	×
DN-CBM (Rao et al. 2024)	✓	×	×
Res-CBM (Shang et al. 2024)	×	✓	✓
VLG-CBM (Srivastava, Yan, and Weng 2024)	△	△	△
V2C-CBM (He et al. 2025b)	✓	×	△
DCBM (Prasse et al. 2025)	✓	△	×
PS-CBM	✓	✓	✓

Table 1: Comparison of representative CBMs on three key limitations: Poor Visual Grounding, Concept Redundancy, and Inadequate Metrics. Symbols indicate well (✓), partial (△), or no (×) improvement.

and overlapping terms (Yang et al. 2023; Srivastava, Yan, and Weng 2024; He et al. 2025b); Others adopt global deduplication, which compresses redundancy but forces unrelated classes to share a fixed pool of concepts, undermining class discrimination (Oikarinen et al. 2023; Yuksekgonul, Wang, and Zou 2023; Shang et al. 2024). Both strategies hinder model clarity and training stability.

- **Inadequate Metrics.** Most CBMs are evaluated solely on classification accuracy, ignoring the interpretability cost of large and redundant concept sets (Yang et al. 2023; Rao et al. 2024; He et al. 2025b; Prasse et al. 2025). As shown in Table 1, few models address this concern explicitly. Without principled metrics to capture the trade-off between accuracy and concept efficiency, performance gains may come at the expense of usability.

To address these challenges, we propose **PS-CBM**, a unified framework for interpretable and scalable **C**oncept **B**ottleneck **M**odeling based on a novel **P**artially **S**hared concept strategy. Our core contributions are as follows:

- **Multimodal Concept Generation.** We integrate the semantic richness of LLMs with visual grounding from exemplar images, generating concept sets that are both semantically meaningful and visually faithful, bridging the semantic-visual gap.
- **Partially Shared Concept Strategy.** We introduce a novel strategy that merges concepts with similar activation patterns and assigns them across all relevant classes. As depicted in Figure 1(c), this partially shared strategy combines the specificity of per-class concepts with the compactness of global sharing, reducing redundancy without sacrificing discriminative expressiveness.
- **Concept-Efficient Accuracy (CEA).** We propose a task-aware, post-hoc metric that jointly quantifies classification accuracy and concept compactness. CEA is interpretable, bounded, model-agnostic, and requires no changes to model training, enabling fair comparison across CBM designs.

As shown in Table 1, unlike previous approaches that only partially (or fail to) address these challenges, PS-CBM achieves comprehensive coverage across all dimensions (indicated by ✓), showcasing its robustness and design coherence. To validate the effectiveness of PS-CBM, we conduct extensive experiments across eleven diverse real-world datasets, covering a broad spectrum of classification tasks, from general-purpose to fine-grained and domain-specific challenges. PS-CBM consistently surpasses state-of-the-art CBMs in classification accuracy (+1.0%–7.4%) and CEA (+2.0%–9.5%). More importantly, it does so with significantly fewer concepts, underscoring its ability to deliver high predictive performance while preserving transparency.

2 Related Work

2.1 Concept Bottleneck Models

Concept Bottleneck Models (CBMs) enhance model interpretability by introducing human-understandable concepts as an intermediate representation between inputs and outputs (Koh et al. 2020). Existing approaches largely fall into two categories: those using a **globally shared** concept pool and those employing **independent**, class-specific pools.

Globally Shared Concept Pools. These methods use a unified concept set shared across *all* classes, enabling reusability and scalability (Oikarinen et al. 2023; Yuksekgonul, Wang, and Zou 2023; Shang et al. 2024; Rao et al. 2024; Midavaine et al. 2024; Prasse et al. 2025; Luyten and van der Schaar 2024; Panousis, Ienco, and Marcos 2024; Vandenhirtz et al. 2024; Penaloza et al. 2025; Zarlenga et al. 2025; Schmalwasser et al. 2025; Xie et al. 2025). For instance, **LF-CBM** (Oikarinen et al. 2023) generates class-level concepts using GPT-3 (Brown et al. 2020) to reduce annotation cost, while **Res-CBM** (Shang et al. 2024) incrementally expands the concept space through residual learning. **DN-CBM** (Rao et al. 2024) employs sparse autoencoders to discover transferable visual concepts, and **DCBM** (Prasse et al. 2025) extracts multi-granular concepts via segmentation foundation models. Despite their scalability, these methods often suffer from *concept redundancy*, where irrelevant or overlapping concepts impair discriminative capacity and clarity.

Independent Concept Pools. Alternatively, class-specific models such as **LaBo** (Yang et al. 2023), **VLG-CBM** (Srivastava, Yan, and Weng 2024), and **V2C-CBM** (He et al. 2025b) generate tailored concepts for each class, enhancing semantic relevance. However, this design introduces *semantic duplication*, where similar concepts are redundantly assigned to multiple classes, leading to inefficiency and potential information leakage. To overcome the limitations of both extremes, we introduce a **Partially Shared Concept Strategy** that adaptively merges semantically similar concepts, balancing compactness and discriminative power.

Alternative Interpretability Strategies. Beyond concept pooling, other methods explore interpretability through different mechanisms (Yan et al. 2023; Delfosse et al. 2024; Bhalla et al. 2024; Laguna et al. 2024; Tan, Zhou, and Chen 2024; Liu, Wang, and Ji 2024; Huang et al. 2024; Dominici et al. 2025; Benou and Raviv 2025; Yu et al. 2025; Penaloza et al. 2025; Hu et al. 2025; Liu, Zhang, and Gu 2025; Yamaguchi and Nishida 2025). For example, **LM4CV** (Yan et al. 2023) uses LLMs to retrieve image-specific concepts, but lacks consistent concept-class mappings, limiting coherence and interpretability. **XBM** (Yamaguchi and Nishida

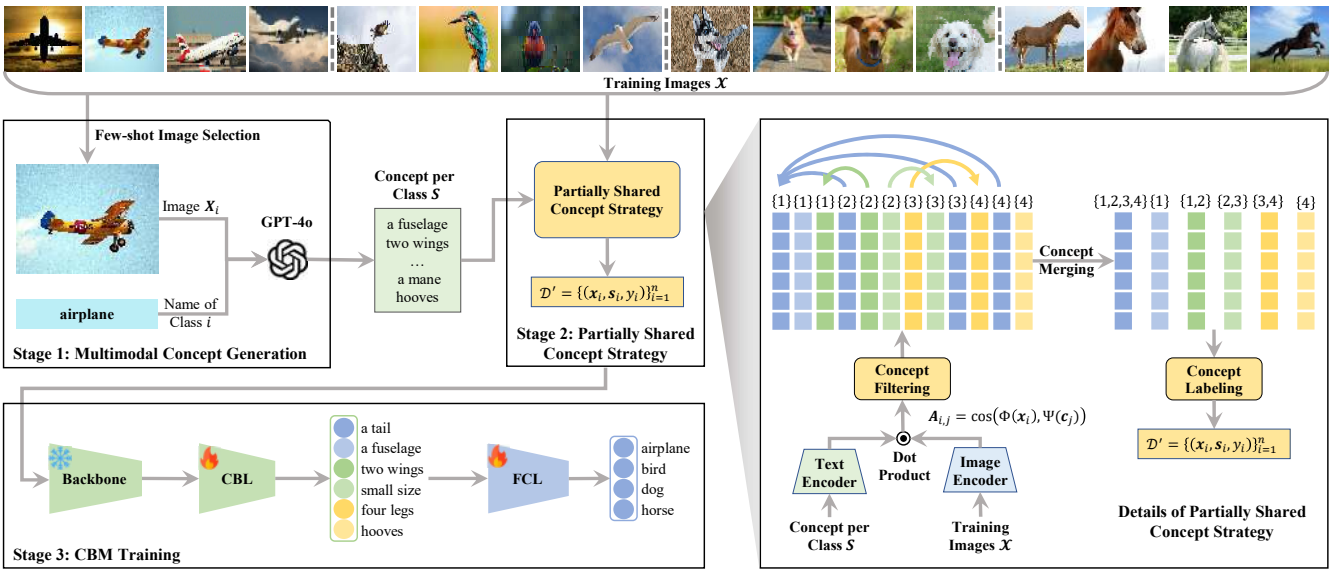

Figure 2: Overview of the PS-CBM pipeline. Stage 1: Generate multimodal concepts $\mathcal{S}$ by aligning LLM-derived semantics with exemplar images. Stage 2: Apply the Partially Shared Concept Strategy based on activation patterns to construct a concept-labeled dataset $\mathcal{D}'$. Stage 3: Train a transparent sequential predictor on $\mathcal{D}'$ for interpretable image classification.

2025) generates textual explanations from embeddings using LLMs. While expressive, it lacks concept-level transparency and requires large backbones, making it impractical for lightweight models such as ResNet50 (He et al. 2016). Recently, **Chat-CBM** (He et al. 2025a) enhances human intervention through language-driven editing, yet concept redundancy and weak grounding remain open challenges.

2.2 Metrics for Concept-based Models

CBMs are typically evaluated by accuracy alone, often overlooking the cost of large or redundant concept sets. To address this, **Concept Utilization Efficiency (CUE)** (Shang et al. 2024) penalizes verbose concept sets but lacks a clear upper bound and is sensitive to textual formatting. **Number of Effective Concepts (NEC)** (Srivastava, Yan, and Weng 2024) quantifies concept sparsity but requires training-time changes and hyperparameter tuning. We propose **Concept-Efficient Accuracy (CEA)**, a post-hoc, task-aware metric that balances accuracy and concept compactness. CEA is text-invariant, training-free, and enables fair comparison across different CBM architectures.

3 The PS-CBM Framework

We propose **PS-CBM**, a unified framework for constructing interpretable concept bottlenecks via a three-stage pipeline:

1. **Multimodal Concept Generation**, which leverages both language and vision modalities to produce semantically meaningful and visually grounded concepts;
2. **Partially Shared Concept Strategy**, which adaptively assigns concepts to classes by merging semantically overlapping concepts based on activation patterns;
3. **CBM Training**, which learns a transparent prediction model through concept supervision.

The overall architecture is illustrated in Figure 2.

Problem Setup. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a dataset with n samples, where $\boldsymbol{x}_i$ is an image from class $y_i \in \mathcal{Y} = \{1, \cdots, l\}$. Each class i has its own image set $\mathcal{X}_i$, with $\mathcal{X} = \bigcup_{i=1}^l \mathcal{X}_i$, and candidate concept set $\mathcal{S}_i$, with $\mathcal{S} = \bigcup_{i=1}^l \mathcal{S}_i$ and $|\mathcal{S}| = m$. The goal is to learn a classification function:

$$\boldsymbol{f} \circ \boldsymbol{g} \circ \boldsymbol{\phi} : \mathcal{X} \to \mathcal{Y},$$

where predictions are mediated through a binary concept space for interpretability.

3.1 Multimodal Concept Generation

We generate interpretable and grounded concepts using both semantic prompts and exemplar images.

Few-shot Image Selection. To anchor concepts visually, we construct a diverse, few-shot exemplar set $\boldsymbol{X}_i \subset \mathcal{X}_i$ for each class i ($1 \le i \le l$) using CLIP embedding (Radford et al. 2021). Specifically, we initialize with a random image and iteratively select additional exemplars that maximize cosine distance from those already selected. For noisy datasets (e.g., Food101 (Bossard, Guillaumin, and Van Gool 2014)), we employ random sampling.

Concept Generation. For each class i, we construct a prompt by combining a text description with the selected exemplars $\boldsymbol{X}_i$ and query GPT-4o twice to reduce randomness. After deduplication, we obtain a candidate concept set $\mathcal{S} = \bigcup_{i=1}^l \mathcal{S}_i$. Each concept $\boldsymbol{c}_j$ ($1 \le j \le m$) is associated with a class set $\mathcal{C}_j$.

3.2 Partially Shared Concept Strategy

To reduce redundancy and enhance interpretability, we refine $\mathcal{S}$ in three steps:

Algorithm 1: Concept Merging

Input: filtered concepts $\boldsymbol{S}$; filtered affinity matrix $\boldsymbol{A}$; merge threshold τ_{merge};

Output: The final concept set $\hat{\mathcal{S}}$ after merging;

1 $m = |\mathcal{S}|$;
2 **for** $i = 1$ **to** m **do**
3 **for** $j = 1$ **to** m **do**
4 $\boldsymbol{Q}_{i,j} = \frac{\boldsymbol{A}_{:,i}^{\top} \boldsymbol{A}_{:,j}}{\|\boldsymbol{A}_{:,i}\| \|\boldsymbol{A}_{:,j}\|}$;
5 $\hat{\mathcal{S}} \leftarrow \emptyset$;
6 **while** $\mathcal{S} \neq \emptyset$ **do**
7 $\mathcal{S}_j \leftarrow \{\boldsymbol{c}_i \in \mathcal{S} \mid \boldsymbol{Q}_{i,j} > \tau_{\text{merge}}\}$ **foreach** $\boldsymbol{c}_j \in \mathcal{S}$;
8 Retrieve $\boldsymbol{c}_{\text{max}} \leftarrow \arg\max_{\boldsymbol{c}_j \in \mathcal{S}} |\mathcal{S}_j|$ and $\mathcal{S}_{\text{max}}$;
9 $\hat{\mathcal{S}} \leftarrow \hat{\mathcal{S}} \cup \{\boldsymbol{c}_{\text{max}}\}$;
10 $\mathcal{S} \leftarrow \mathcal{S} \setminus (\{\boldsymbol{c}_{\text{max}}\} \cup \mathcal{S}_{\text{max}})$;
11 **return** $\hat{\mathcal{S}}$;

Step 1: Concept Filtering. Let $\Phi(\cdot)$ and $\Psi(\cdot)$ denote the image and text encoders, respectively. We compute the affinity matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ between images and concepts:

$$\boldsymbol{A}_{i,j} = \cos(\Phi(\boldsymbol{x}_i), \Psi(\boldsymbol{c}_j)).$$

We retain concept $\boldsymbol{c}_j$ if its average top-4 alignment with class images exceeds the confidence threshold τ_{conf}.

Step 2: Concept Merging. Algorithm 1 outlines the concept merging process. We begin by computing a correlation matrix $\boldsymbol{Q}$ over filtered concepts (Lines 1–4) and greedily merge those exceeding a threshold τ_{merge}) (Lines 5–11). Merged concepts $\hat{\mathcal{S}}$ inherit the union of original class sets. To limit redundancy, we retain only the top K exclusive concepts per class, i.e., those associated with a single class.

Step 3: Concept Labeling. Let $\hat{m} = |\hat{\mathcal{S}}|$. The one-hot encoded concept label vector $\boldsymbol{s}_i = [s_{i,j}] \in \{0,1\}^{\hat{m}}$ for each image $\boldsymbol{x}_i$ is defined as:

$$s_{i,j} = \begin{cases} 1, & \text{if } y_i \in C_j \text{ and } \boldsymbol{A}_{i,j} > \tau_{\text{conf}}, \\ 0, & \text{otherwise.} \end{cases}$$

This yields the concept-labeled dataset $\mathcal{D}'$ for training CBM:

$$\mathcal{D}' = \{(\boldsymbol{x}_i, \boldsymbol{s}_i, y_i)\}_{i=1}^{n}.$$

3.3 CBM Training

Training Concept Bottleneck Layer (CBL). Given the concept-labeled dataset $\mathcal{D}'$, we train the CBL to predict multi-label concept annotations. Let $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^d$ denote a *frozen* backbone encoder mapping each image $\boldsymbol{x}$ to an embedding $\boldsymbol{z} = \boldsymbol{\phi}(\boldsymbol{x})$. The CBL $\boldsymbol{g} : \mathbb{R}^d \to \mathbb{R}^{\hat{m}}$ projects embeddings to concept logits. We optimize $\boldsymbol{g}$ by minimizing Binary Cross-Entropy (BCE) loss:

$$\min_{\boldsymbol{g}} \mathcal{L}_{\text{CBL}} = \frac{1}{n} \sum_{i=1}^{n} \text{BCE}(\boldsymbol{g}(\boldsymbol{\phi}(\boldsymbol{x}_i)), \boldsymbol{s}_i), \tag{1}$$

where $\boldsymbol{s}_i$ denotes a vector of the binary concept labels.

Training Final Classification Layer (FCL). At last, we train a sparse linear classifier $\boldsymbol{f} : \mathbb{R}^{\hat{m}} \to \mathbb{R}^{l}$ with weight matrix $\boldsymbol{W}_F$ and bias $\boldsymbol{b}_F$, to map concept logits to class predictions. For each sample, we first compute concept logits via the trained, frozen CBL $\boldsymbol{g}$ and normalize them using training set statistics. The optimization objective combines Cross-Entropy (CE) loss and elastic-net regularization (Zou and Hastie 2005):

$$\min_{\boldsymbol{f}} \mathcal{L}_{\text{FCL}} = \frac{1}{n} \sum_{i=1}^{n} \text{CE}(\boldsymbol{f}(\hat{\boldsymbol{g}}(\boldsymbol{x}_i)), y_i) + \lambda R_\alpha(\boldsymbol{W}_F), \tag{2}$$

where $\hat{\boldsymbol{g}}(\boldsymbol{x}_i)$ denotes the normalized concept logits, and

$$R_\alpha(\boldsymbol{W}_F) = (1-\alpha)\|\boldsymbol{W}_F\|_2^2 + \alpha\|\boldsymbol{W}_F\|_1.$$

We solve this objective using the GLM-SAGA (Zou and Hastie 2005) optimizer.

3.4 Concept-Efficient Accuracy (CEA)

CEA is grounded in Shannon's information theory, which establishes that distinguishing among l classes using binary (0/1) concept signals requires at least $k = \lceil \log_2 l \rceil$ bits of information (Shannon 1948). CEA then quantifies how efficiently a model achieves its accuracy relative to this theoretical bound, penalizing redundant concept usage. Let m denote the number of concepts used. We define CEA as:

$$\text{CEA} = \frac{\text{ACC}}{(\log_k m)^\beta}, \tag{3}$$

where $\text{ACC} \in [0,1]$ denotes the model's classification accuracy, and $\beta \geq 0$ is a temperature parameter controlling the penalty on concept complexity. A smaller β makes CEA focus more on accuracy, whereas a larger β emphasizes concept compactness. CEA possesses three desirable properties:

- **Optimal Efficiency**: CEA approaches 1 as accuracy nears 1 ($\text{ACC} \to 1$) and concept usage approaches the theoretical minimum ($m \to k$).
- **Adaptive Scaling**: The base-k logarithmic scaling ensures that the penalty adapts to task complexity: more classes allow moderately larger concept sets without excessive penalization.
- **Theoretical Foundation**: The formulation aligns with Shannon's information theory, encouraging parsimonious yet accurate explanations.

In summary, CEA provides a unified measure of both accuracy and interpretability, enabling fair, task-aware comparisons across CBMs with varying concept complexities.

4 Experiments

We conduct extensive experiments to evaluate PS-CBM's classification performance, interpretability, and robustness.

4.1 Experimental Setup

Datasets. We evaluate PS-CBM on 11 publicly available real-world datasets spanning multiple domains: (1) General image classification: **CIFAR10**, **CIFAR100** (Krizhevsky 2009), and **ImageNet** (Deng et al. 2009); (2) Fine-grained classification: **Aircraft** (Maji et al. 2013), **CUB200** (Wah et al. 2011), **Flower102** (Nilsback and Zisserman 2008), and

Method	Aircraft		CIFAR10		CIFAR100		CUB200		DTD		Flower102		Food101		HAM10000		ImageNet		Resisc45		UCF101	
	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑
Linear Probe	42.5	–	88.5	–	69.8	–	67.4	–	71.8	–	97.4	–	82.7	–	79.8	–	72.6	–	85.4	–	81.0	–
LaBo	40.3	27.9	87.5	60.1	68.1	47.1	66.8	46.1	71.3	49.4	96.6	66.7	82.2	56.8	77.1	50.7	72.1	49.0	84.1	58.4	79.9	55.2
LF-CBM	36.2	28.9	87.3	63.1	68.8	49.9	58.6	45.7	68.5	52.8	94.4	73.1	77.7	58.8	70.2	56.8	67.5	48.8	84.5	62.2	80.0	58.2
LM4CV	38.5	28.8	81.4	62.8	65.9	49.3	65.6	48.6	70.8	53.6	94.6	70.7	81.1	60.6	66.8	49.8	69.6	50.2	81.3	61.7	78.9	59.0
DN-CBM	42.1	28.7	88.2	55.2	68.6	46.7	66.6	46.2	74.5	49.8	96.6	65.8	82.3	56.1	80.5	47.6	73.1	52.0	85.7	57.3	81.1	55.3
Res-CBM	36.9	28.9	87.6	61.9	65.7	49.6	59.1	46.9	64.4	49.5	93.8	73.1	79.3	61.6	77.5	52.7	68.3	52.2	81.8	61.5	75.4	58.6
VLG-CBM	45.6	34.6	89.6	67.4	68.3	50.8	68.0	51.0	72.2	55.3	97.1	72.5	81.6	60.3	79.8	59.9	65.7	47.6	84.9	65.1	81.3	60.8
V2C-CBM	37.1	25.6	87.3	60.0	68.3	47.2	63.8	44.0	70.8	49.1	96.6	68.2	81.2	56.1	75.4	49.6	73.0	49.6	83.7	58.1	78.3	54.1
DCBM	36.4	25.8	85.7	55.9	62.3	44.3	59.8	43.2	70.1	48.8	94.0	66.8	79.3	56.4	75.3	46.4	57.0	42.3	81.1	56.4	75.3	53.6
PS-CBM	**47.0**	**34.9**	**89.8**	**68.5**	**72.1**	**53.6**	**70.1**	**53.3**	**75.1**	**59.0**	**97.9**	**74.9**	**83.0**	**61.7**	**83.4**	**61.3**	**74.0**	**54.6**	**87.5**	**65.7**	**83.0**	**61.8**

Table 2: ACC and CEA on 11 datasets using `CLIP_RN50` backbone. **Bold** and underline denote the best and second-best results.

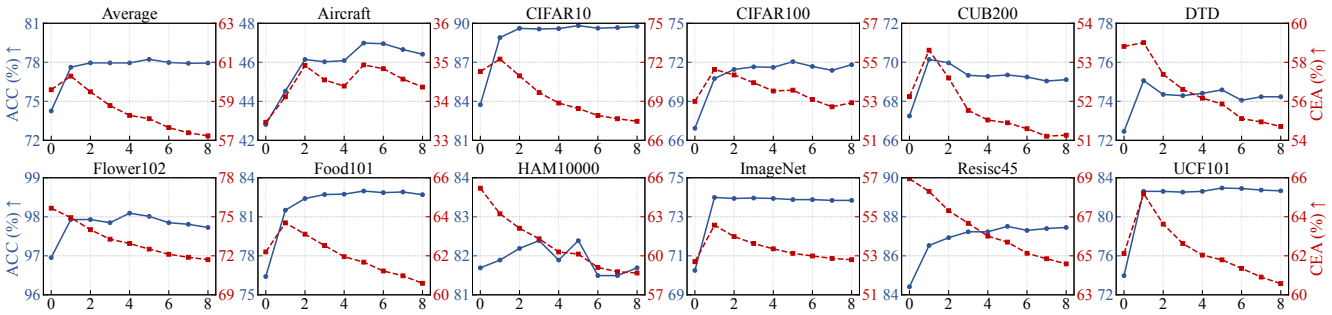

Figure 3: Variations of ACC and CEA with the upper limit K on the number of exclusive concepts per class across datasets.

Food101 (Bossard, Guillaumin, and Van Gool 2014); (3) Domain-specific tasks: **DTD** (Cimpoi et al. 2014) (textures), **HAM10000** (Tschandl, Rosendahl, and Kittler 2018) (skin tumor classification), **Resisc45** (Cheng, Han, and Lu 2017) (remote sensing), and **UCF101** (Soomro, Zamir, and Shah 2012) (action recognition).

Metrics. We evaluate performance using three key metrics:

- **Classification Accuracy (ACC):** Serving as the standard measure of predictive performance.
- **Concept-Efficient Accuracy (CEA):** Our proposed metric balancing accuracy and concept compactness.
- **CLIP Score (in ablations):** Assessing concept–image alignment, especially for domain-specific datasets like DTD, Resisc45, and UCF101, where it is crucial.

Baselines. We assess PS-CBM against two categories of models. The first comprises leading CBMs, including **LaBo** (Yang et al. 2023), **LF-CBM** (Oikarinen et al. 2023), **LM4CV** (Yan et al. 2023), **DN-CBM** (Rao et al. 2024), **Res-CBM** (Shang et al. 2024), **VLG-CBM** (Srivastava, Yan, and Weng 2024), **V2C-CBM** (He et al. 2025b), and **DCBM** (Prasse et al. 2025). The second is the **Linear Probe** (Yang et al. 2023), a logistic regression model trained on CLIP features, serving as a strong black-box baseline without explicit interpretability.

Implementation Details. All methods share identical train/dev/test splits and backbones (`CLIP_RN50` and `CLIP_ViT-L/14`); results for the latter appear in Appendix B. Image-text similarities use `CLIP_ViT-B/16`. Concept filtering and labeling uses $\tau_{\text{conf}} = 0.20$. Concept merging uses $\tau_{\text{merge}} \in [0.9996, 0.9999]$ (step 0.0001), and the pruning parameter K from $\{0, 1, \cdots, 8\}$. The CEA temperature parameter β is set to 0.25. Due to ImageNet's scale, only 10% of its training images are sampled for merging. Full configurations are provided in Appendix A.

Experiment Environment. All experiments are run on Ubuntu 22.04 with PyTorch 2.3.0, CUDA 12.1, and a single NVIDIA RTX 3080 Ti (12GB), using an Intel® Xeon® 4214R CPU (12-core, 2.40 GHz) and 90 GB RAM.

4.2 Classification Accuracy

Table 2 presents the classification accuracy (ACC) of PS-CBM and all baseline models across 11 datasets. PS-CBM consistently achieves the highest accuracy, demonstrating robust generalization across a wide range of domains, including fine-grained, texture, and remote sensing tasks. This result underscores PS-CBM's ability to maintain strong predictive performance even in challenging settings.

To further contextualize PS-CBM's performance, Table 3 presents the average classification accuracy alongside the number of concepts used by each method. PS-CBM surpasses state-of-the-art CBMs by 1.0%–7.4% in average ACC, while significantly reducing concept usage.

Notably, PS-CBM uses an average of 7,647 fewer concepts per dataset than DN-CBM, achieving higher ACC and CEA. As LM4CV (Fig. 3) shows, large concept pools can inflate accuracy even with random concepts, indicating spurious correlations. PS-CBM's strong performance with fewer concepts demonstrates tighter concept grounding and reduced leakage risk.

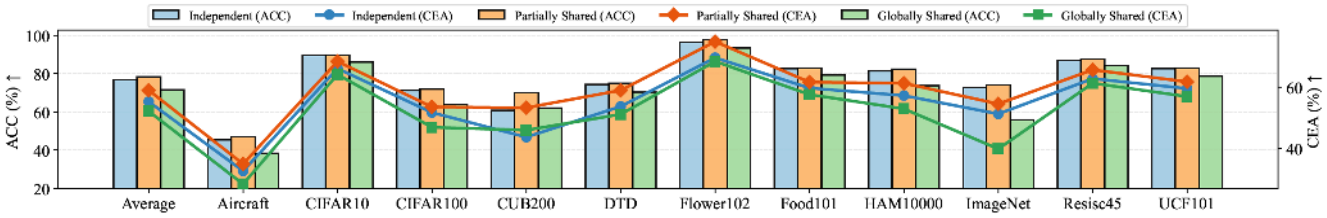

Figure 4: Ablation study on different concept bottleneck strategies, comparing their impact on ACC and CEA.

Figure 5: Case study of PS-CBM predictions. Green highlights correct predictions, yellow denotes partially accurate or ambiguous outputs, and red indicates incorrect results.

Method	Avg. ACC (%) ↑	Avg. # Concepts ↓	Avg. CEA (%) ↑
LaBo	72.8	7,900	51.6
LF-CBM	72.9	718	55.2
LM4CV	73.4	873	56.4
DN-CBM	<u>77.3</u>	8,192	53.4
Res-CBM	71.8	**291**	56.7
VLG-CBM	75.2	732	<u>57.0</u>
V2C-CBM	72.8	7,500	51.2
DCBM	70.9	2,048	49.5
PS-CBM	**78.3**	<u>545</u>	**59.0**

Table 3: Comparison of methods by average ACC, number of concepts used, and CEA. **Bold** and <u>underlined</u> indicate the best and second-best results, respectively.

4.3 Explanability

Alongside accuracy, Table 2 also reports CEA, a metric designed to capture the trade-off between predictive performance and interpretability. PS-CBM achieves the highest CEAs across all datasets, indicating that it delivers accurate predictions with a minimal and meaningful set of concepts.

Table 3 presents the average CEA scores of all CBM methods across 11 datasets. PS-CBM surpasses state-of-the-art CBMs by 2.0%–9.5% in average CEA, highlighting its ability to balance classification performance and concept compactness. In contrast, methods such as DN-CBM, LaBo, and V2C-CBM, despite achieving high accuracy, rely on excessively large concept sets, often an order of magnitude larger than others. This redundancy compromises interpretability and limits human oversight. Their lower CEA values further emphasize the importance of jointly evaluating performance and interpretability, as captured by CEA.

Method	DTD	Resisc45	UCF101	Concept Generation	Concept Pools
LaBo	0.227	<u>0.222</u>	0.230	Language	Independent
LF-CBM	0.225	0.208	0.199	Language	Globally Shared
DN-CBM	0.192	0.187	0.187	Vision	Globally Shared
V2C-CBM	<u>0.246</u>	0.216	<u>0.247</u>	Vision	Independent
PS-CBM	**0.249**	**0.255**	**0.265**	Language+Vision	Partially Shared

Table 4: CLIP Score (↑) comparison across three domain-specific datasets. **Bold** and <u>underlined</u> indicate the best and second-best results, respectively.

To evaluate concept–image alignment, Table 4 reports the average CLIP score at the class level. Compared to LaBo and LF-CBM, which use only LLMs for concept generation, and DN-CBM and V2C-CBM, which rely solely on visual alignment, PS-CBM achieves significantly better alignment on domain-specific datasets such as DTD, Resisc45, and UCF101. These results showcase the benefits of PS-CBM's multimodal concept generation strategy in producing concepts that are both semantically rich and visually grounded.

4.4 Ablation Study

To better understand the design choices in PS-CBM, we conduct several ablation studies. Figure 3 illustrates how ACC and CEA vary with the maximum number of exclusive concepts per class (K). Accuracy improves sharply when K increases from 0 to 1 and stabilizes beyond $K = 2$. Meanwhile, CEA peaks at $K = 1$ and declines as K increases.

τ_{conf}	Avg. ACC (%) ↑	Avg. # Concepts ↓	Avg. CEA (%) ↑
0.10	76.20	548	57.41
0.15	76.14	548	57.36
0.20	**78.35**	545	**59.02**
0.25	72.71	458	55.16
0.30	57.55	**145**	46.84

Table 5: Effect of varying confidence threshold τ_{conf} on average ACC, number of concepts used, and CEA across all datasets. **Bold** highlights the best-performing setting.

These results suggest that allowing a small number of class-specific concepts is beneficial, but excessive exclusivity undermines concept efficiency. This confirms that PS-CBM effectively leverages shared concepts while maintaining the discriminative power of a minimal set of exclusive ones.

Figure 4 compares three concept-sharing strategies: Independent, Partially Shared, and Globally Shared, implemented within the PS-CBM framework. The Partially Shared variant achieves the best performance in both ACC and CEA. This validates the core idea of PS-CBM: selectively sharing concepts among semantically similar classes leads to more accurate and interpretable models than either fully disjoint or globally shared approaches.

Table 5 explores the impact of the confidence threshold τ_{conf} used in concept filtering and labeling, with thresholds ranging from 0.10 to 0.30. We observe that a threshold of 0.20 yields the best balance between ACC, # Concepts, and CEA. This result reinforces the robustness of PS-CBM under different hyperparameter settings.

4.5 Case Study

To further illustrate how PS-CBM supports interpretable decision-making, we present a case study in Figure 5 following the visualization protocol from VLG-CBM (Srivastava, Yan, and Weng 2024). For each prediction, we compute the contribution of each concept based on its activation and corresponding weight, highlighting the top-5 concepts and aggregating the rest as "sum of other concepts."

Models with high interpretability tend to have a large share of their predictions explained by the top few concepts, making decisions easier to interpret and verify. As shown in Figure 5, PS-CBM and VLG-CBM exhibit more concentrated contributions from their top concepts compared to DN-CBM and LaBo. In particular, PS-CBM benefits from both its multimodal concept generation and partially shared concept strategy, producing more relevant and semantically meaningful explanations. This leads to reduced concept redundancy and a lower risk of information leakage, thereby enhancing the overall transparency of the model.

4.6 Effect of Partially Shared Concept Strategy

To visualize the effect of the Partially Shared Concept Strategy (PSCS), we plot the concept-class map on CIFAR10 in Figure 6, using $K = 1$ and a concept merge threshold of 0.9996. The visualization reveals that PSCS selectively shares concepts across semantically related classes while preserving class-specific ones where necessary.

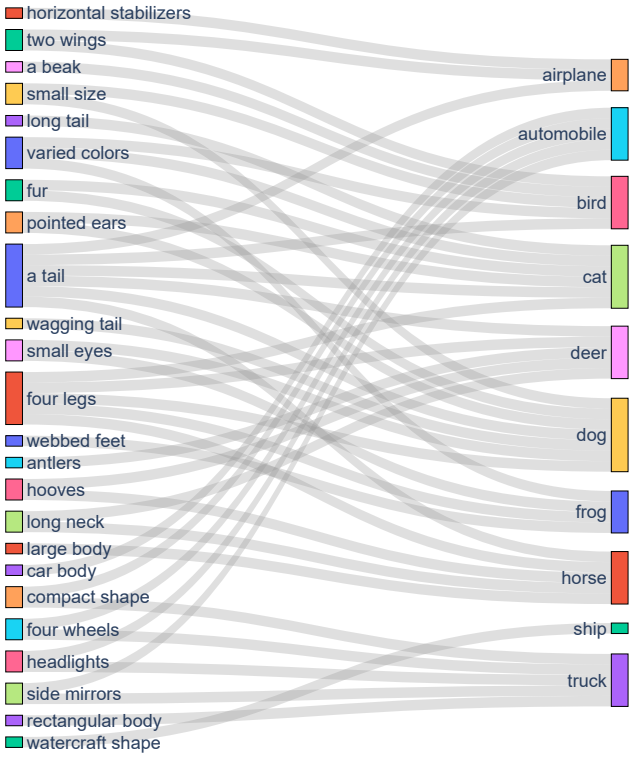

Figure 6: Concept-class map on CIFAR10 with $K = 1$ and $\tau_{\text{merge}} = 0.9996$. Partially shared concepts appear across related classes, while others remain class-specific.

For example, *horizontal stabilizers* and *a beak* appear only for airplanes and birds, respectively. Meanwhile, shared concepts such as *two wings*, *a tail*, and *four legs* span related classes, reflecting common visual features. Other shared attributes, such as *hooves* or *long neck*, apply to both deer and horses, while automotive-related features like *headlights* and *side mirrors* are shared between cars and trucks.

These patterns highlight PSCS's ability to learn compact, semantically coherent concepts, enhancing interpretability and robustness as a core contributor to PS-CBM's success.

5 Conclusions

In this paper, we introduced PS-CBM, a unified and scalable CBM framework for interpretable image classification. PS-CBM introduces three key innovations: (1) a multimodal concept generation module that improves both relevance and interpretability; (2) a Partially Shared Concept Strategy that adaptively merges and reassigns concepts across classes based on activation patterns, effectively balancing specificity and compactness; and (3) a new metric, CEA, that jointly captures accuracy and concept efficiency. Extensive experiments on 11 diverse datasets show that PS-CBM consistently surpasses state-of-the-art CBMs in both accuracy and interpretability, while requiring significantly fewer concepts. Overall, PS-CBM offers a practical and extensible solution for building interpretable vision systems, laying a strong foundation for future research in scalable, multimodal, and concept-efficient learning.

Acknowledgements

We sincerely thank the anonymous reviewers for their insightful and constructive feedback, which greatly improved this paper. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 62125201 and U24B20174.

References

Benou, I.; and Raviv, T. R. 2025. Show and Tell: Visually Explainable Deep Neural Nets via Spatially-Aware Concept Bottleneck Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 30063–30072.

Bhalla, U.; Oesterling, A.; Srinivas, S.; Calmon, F.; and Lakkaraju, H. 2024. Interpreting CLIP with Sparse Linear Concept Embeddings (SpLiCE). In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, 84298–84328.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–Mining Discriminative Components with Random Forests. In *European Conference on Computer Vision (ECCV)*, 446–461.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 1877–1901.

Chauhan, K.; Tiwari, R.; Freyberg, J.; Shenoy, P.; and Dvijotham, K. 2023. Interactive Concept Bottleneck Models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 5948–5955.

Cheng, G.; Han, J.; and Lu, X. 2017. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 105(10): 1865–1883.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing Textures in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3606–3613.

Delfosse, Q.; Sztwiertnia, S.; Rothermel, M.; Stammer, W.; and Kersting, K. 2024. Interpretable Concept Bottlenecks to Align Reinforcement Learning Agents. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, 66826–66855.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255.

Dominici, G.; Barbiero, P.; Giannini, F.; Gjoreski, M.; Marra, G.; and Langheinrich, M. 2025. Counterfactual Concept Bottleneck Models. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

He, H.; Zhu, L.; Li, K.; Zhang, X.; Hu, J.; Fu, O.; Yao, Z.; and Lu, Y. 2025a. Chat-CBM: Towards Interactive Concept Bottleneck Models with Frozen Large Language Models. *arXiv preprint arXiv:2509.17522*.

He, H.; Zhu, L.; Zhang, X.; Zeng, S.; Chen, Q.; and Lu, Y. 2025b. V2C-CBM: Building Concept Bottlenecks with Vision-to-Concept Tokenizer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 3401–3409.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hu, L.; Ren, C.; Hu, Z.; Lin, H.; Wang, C.-L.; Tan, Z.; Lyu, W.; Zhang, J.; Xiong, H.; and Wang, D. 2025. Editable Concept Bottleneck Models. In *International Conference on Machine Learning (ICML)*.

Huang, Q.; Song, J.; Hu, J.; Zhang, H.; Wang, Y.; and Song, M. 2024. On the concept trustworthiness in concept bottleneck models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 21161–21168.

Koh, P. W.; Nguyen, T.; Tang, Y. S.; Mussmann, S.; Pierson, E.; Kim, B.; and Liang, P. 2020. Concept Bottleneck Models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 5338–5348.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's Thesis, University of Tront*.

Laguna, S.; Marcinkevičs, R.; Vandenhirtz, M.; and Vogt, J. E. 2024. Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable? In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, 85006–85044.

Liu, J.; Wang, F.; and Ji, J. 2024. Concept-Level Causal Explanation Method for Brain Function Network Classification. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 3087–3096.

Liu, Y.; Zhang, T.; and Gu, S. 2025. Hybrid Concept Bottleneck Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 20179–20189.

Luyten, M. R.; and van der Schaar, M. 2024. A Theoretical Design of Concept Sets: Improving the Predictability of Concept Bottleneck Models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, 100160–100195.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-Grained Visual Classification of Aircraft. *arXiv preprint arXiv:1306.5151*.

Midavaine, N.; Go, G. H. T.; Canez, D.; Simion, I.; and Chatterji, S. 2024. [Re] On the Reproducibility of Post-Hoc Concept Bottleneck Models. *Transactions on Machine Learning Research (TMLR)*.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 722–729.

Oikarinen, T.; Das, S.; Nguyen, L.; and Weng, L. 2023. Label-free Concept Bottleneck Models. In *International Conference on Learning Representations (ICLR)*.

Oikarinen, T.; and Weng, T.-W. 2022. Clip-dissect: Automatic description of neuron representations in deep vision networks. *arXiv preprint arXiv:2204.10965*.

Panousis, K. P.; Ienco, D.; and Marcos, D. 2024. Coarse-to-Fine Concept Bottleneck Models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, 105171–105199.

Penaloza, E.; Zhan, T. H.; Charlin, L.; and Zarlenga, M. E. 2025. Addressing Concept Mislabeling in Concept Bottleneck Models Through Preference Optimization. In *International Conference on Machine Learning (ICML)*.

Prasse, K.; Knab, P.; Marton, S.; Bartelt, C.; and Keuper, M. 2025. DCBM: Data-Efficient Visual Concept Bottleneck Models. In *International Conference on Machine Learning (ICML)*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 8748–8763.

Rao, S.; Mahajan, S.; Böhle, M.; and Schiele, B. 2024. Discover-then-Name: Task-Agnostic Concept Bottlenecks via Automated Concept Discovery. In *European Conference on Computer Vision (ECCV)*, 444–461.

Sawada, Y.; and Nakamura, K. 2022. Concept Bottleneck Model with Additional Unsupervised Concepts. *IEEE Access*, 10: 41758–41765.

Schmalwasser, L.; Penzel, N.; Denzler, J.; and Niebling, J. 2025. FastCAV: Efficient Computation of Concept Activation Vectors for Explaining Deep Neural Networks. In *International Conference on Machine Learning (ICML)*.

Shang, C.; Zhou, S.; Zhang, H.; Ni, X.; Yang, Y.; and Wang, Y. 2024. Incremental residual concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11030–11040.

Shannon, C. E. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3): 379–423.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv preprint arXiv:1212.0402*.

Srivastava, D.; Yan, G.; and Weng, T.-W. 2024. VLG-CBM: training concept bottleneck models with vision-language guidance. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, 79057–79094.

Tan, A.; Zhou, F.; and Chen, H. 2024. Explain via Any Concept: Concept Bottleneck Model with Open Vocabulary Concepts. In *European Conference on Computer Vision (ECCV)*, 123–138.

Tschandl, P.; Rosendahl, C.; and Kittler, H. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1): 1–9.

Vandenhirtz, M.; Laguna, S.; Marcinkevičs, R.; and Vogt, J. 2024. Stochastic Concept Bottleneck Models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, 51787–51810.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset.

Xie, Y.; Zeng, Z.; Zhang, H.; Ding, Y.; Wang, Y.; Wang, Z.; Chen, B.; and Liu, H. 2025. Discovering Fine-Grained Visual-Concept Relations by Disentangled Optimal Transport Concept Bottleneck Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 30199–30209.

Yamaguchi, S.; and Nishida, K. 2025. Explanation Bottleneck Models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 21886–21894.

Yan, A.; Wang, Y.; Zhong, Y.; Dong, C.; He, Z.; Lu, Y.; Wang, W. Y.; Shang, J.; and McAuley, J. 2023. Learning Concise and Descriptive Attributes for Visual Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3090–3100.

Yang, Y.; Panagopoulou, A.; Zhou, S.; Jin, D.; Callison-Burch, C.; and Yatskar, M. 2023. Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19187–19197.

Yu, L.; Han, H.; Tao, Z.; Yao, H.; and Xu, C. 2025. Language Guided Concept Bottleneck Models for Interpretable Continual Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 14976–14986.

Yuksekgonul, M.; Wang, M.; and Zou, J. 2023. Post-hoc Concept Bottleneck Models. In *International Conference on Learning Representations (ICLR)*.

Yun, T.; Bhalla, U.; Pavlick, E.; and Sun, C. 2023. Do Vision-Language Pretrained Models Learn Composable Primitive Concepts? *Transactions on Machine Learning Research (TMLR)*.

Zarlenga, M. E.; Dominici, G.; Barbiero, P.; Shams, Z.; and Jamnik, M. 2025. Avoiding Leakage Poisoning: Concept Interventions Under Distribution Shifts. In *International Conference on Machine Learning (ICML)*.

Zou, H.; and Hastie, T. 2005. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2): 301–320.

A Implementation Details

A.1 Dataset Details

To rigorously evaluate the performance, interpretability, and generalizability of PS-CBM, we conduct experiments across 11 publicly available datasets, which are selected to reflect a wide range of visual recognition challenges, including general-purpose object classification, fine-grained recognition, and domain-specific tasks such as medical diagnosis and satellite imagery analysis. Below, we provide detailed descriptions and the unique characteristics of each dataset.

General Image Classification. This category includes benchmarks designed to evaluate a model's ability to recognize common visual concepts across broad object categories. These datasets are commonly used in computer vision to validate generalization on everyday object classes:

- **CIFAR10 (Krizhevsky 2009):** A classic benchmark of 60,000 low-resolution (32×32) color images evenly distributed across 10 object categories–airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset provides 6,000 images per class, split into 50,000 for training and 10,000 for testing, making it a foundational testbed for general-purpose image classification.
- **CIFAR100 (Krizhevsky 2009):** A more fine-grained variant of CIFAR10, this dataset contains the same number of total images (60,000) but subdivides them into 100 distinct classes grouped under 20 superclasses. Each class contains 600 images. The increased granularity introduces additional complexity, making it a stronger benchmark for evaluating a model's discriminative capabilities across visually similar categories.
- **ImageNet (Deng et al. 2009):** A large-scale benchmark composed of over 1.2 million high-resolution training images and 50,000 validation images, spanning 1,000 object categories. As one of the most influential datasets in deep learning, ImageNet has been instrumental in driving progress in image classification, representation learning, and transfer learning. Its scale, diversity, and high intra-class variability make it a robust test of generalization.

Fine-Grained Classification. This task requires distinguishing between visually similar subcategories within a broader class (e.g., bird species, airplane models). Such datasets pose greater challenges due to subtle inter-class differences, requiring models to capture fine-grained and nuanced visual features.

- **Aircraft (Maji et al. 2013):** Comprising approximately 10,000 images categorized into 100 aircraft variants, this dataset requires models to differentiate fine-grained structural differences in aircraft such as engines, wings, and fuselage shape. It poses a significant challenge due to inter-class similarity and intra-class variability caused by different angles, lighting, and environments.
- **CUB200 (Wah et al. 2011):** Contains 11,788 images spanning 200 bird species, each annotated with detailed part locations (e.g., beak, wings, tail). It is a widely used benchmark in the fine-grained classification community, offering part-based and explainable modeling. Recognition in this dataset often hinges on subtle color patterns, feather textures, and pose variations.
- **Flower102 (Nilsback and Zisserman 2008):** Comprises 8,189 images from 102 flower species, posing challenges due to high inter-class visual similarity and variations in lighting, scale, and background clutter. It serves as a standard benchmark for evaluating fine-grained classification models in the botanical domain.
- **Food101 (Bossard, Guillaumin, and Van Gool 2014):** Consists of 101,000 images across 101 food categories, with each class containing 750 training and 250 test images. The dataset exhibits large intra-class variability due to differences in cooking styles, presentation, and lighting, making it an excellent testbed for evaluating model robustness and generalization in the context of fine-grained food recognition.

Domain-Specific Tasks. These datasets cover more specialized and domain-intensive classification tasks, such as texture analysis, medical diagnosis, and remote sensing, providing a rigorous test of a model's adaptability to non-standard visual domains:

- **DTD (Cimpoi et al. 2014):** Comprising 5,640 images grouped into 47 texture classes (e.g., striped, bubbly, porous), DTD focuses on recognizing mid-level perceptual patterns in textures. Unlike traditional object-centric datasets, it evaluates models on their ability to capture material properties and fine-grained textural information, often independent of object identity.
- **HAM10000 (Tschandl, Rosendahl, and Kittler 2018):** A medically oriented dataset of 10,015 dermatoscopic images labeled into 7 diagnostic classes of skin lesions (e.g., melanoma, basal cell carcinoma). HAM10000 is widely used in the medical imaging community for training and evaluating models in skin cancer classification. It provides a realistic setting for assessing model reliability, especially in safety-critical applications.
- **Resisc45 (Cheng, Han, and Lu 2017):** Composed of 31,500 satellite images across 45 scene categories such as airports, harbors, and residential areas, this dataset evaluates models' capabilities in scene understanding from aerial imagery. The scenes vary widely in scale, orientation, and structural layout, making Resisc45 a comprehensive test for domain adaptation and robustness.
- **UCF101 (Soomro, Zamir, and Shah 2012):** Originally designed for video-based human action recognition, UCF101 contains 13,320 videos spanning 101 action categories. For our purposes, we extract representative still frames from these videos and repurpose the dataset for image-based action classification. This approach allows us to evaluate the capacity of static models to infer dynamic actions, a challenging task due to the temporal nature of the source content.

A.2 Dataset Splits

To ensure fair and consistent evaluation across all baselines and experimental settings, we adopt standard or widely accepted dataset splits, summarized in Table 6. The splitting strategies are as follows:

- For **Aircraft**, **DTD**, **Flower102**, **Food101**, **Resisc45**, and **UCF101**, we use the official splits provided by LaBo (Yang et al. 2023), which balance class representation and enable reproducibility.
- For **CUB200**, we randomly sample 10 training images per class to construct a development set, maintaining class balance with a minimal validation budget.

Dataset	# Classes	Train	Development	Test
Aircraft	100	3,334	3,333	3,333
CIFAR10	10	45,000	5,000	10,000
CIFAR100	100	45,000	5,000	10,000
CUB200	200	3,994	2,000	5,794
Flower102	102	4,093	1,633	2,463
Food101	101	50,500	20,200	30,300
DTD	47	2,820	1,128	1,692
HAM10000	7	8,010	1,000	1,005
ImageNet	1,000	1,281,167	50,000	–
Resisc45	45	3,150	3,150	25,200
UCF101	101	7,639	1,898	3,783

Table 6: Summary of dataset statistics and data splits used in our experiments, including the number of classes and the distribution across training, development, and test sets.

- For **CIFAR10** and **CIFAR100**, we randomly reserve 10% of the training data for development, following standard practice in few-shot and semi-supervised settings.
- For **HAM10000**, we apply a stratified 80/10/10 split per class for training, development, and test, preserving diagnostic class balance.
- For **ImageNet**, we evaluate only on the development set due to the scale and computational cost of full training.

A.3 Prompt Design

To generate visually grounded concepts for each category, we design a unified prompting strategy that integrates concise textual instructions with exemplar images. The textual component is crafted to elicit descriptions of distinctive and observable visual features, deliberately avoiding abstract reasoning or domain-specific jargon.

For each category, we interact with GPT using *four* representative images and issue *two* separate queries. The resulting concept sets are then *merged and deduplicated* to produce a concise and diverse list of relevant concepts. This dual-query setup helps capture a broader range of semantic attributes while ensuring consistency across responses.

Figure 7 provides an illustrative example of the prompt used for the class "dog" in CIFAR10, highlighting the structure and effectiveness of multimodal concept generation.

A.4 Hyperparameter Settings

Parameter Settings for PS-CBM. In the experiments, we set $\tau_{\text{conf}} = 0.20$ for concept filtering and labeling and $\beta = 0.25$ when computing the CEA. For the rest hyperparameters, we summarize the settings in Table 7.

Parameter Settings for Baseline Methods. Table 11 summarizes the hyperparameter configurations used for all baseline methods. Below, we detail the specific settings for each:

- **LaBo (Yang et al. 2023):** We use the concept set provided by the original authors and enable the `remove_cls_name` option to exclude class names from candidate concepts. For CIFAR100, the learning rate is increased from 1e-5 to 1e-4. All datasets are trained for extended epochs to ensure convergence, while other hyperparameters remain at their defaults.

Image Input
A small set of representative images from the target category are encoded as base64 and appended to the prompt as image URLs.

Text Prompt
You are shown several sample images from "{dog}".
Examine their visual patterns and summarize the key features that commonly define this category in its image domain.
Please focus on traits that are typically present across most samples, even if not in every image.

- Describe each feature as a short, concrete visual concept without abstraction or inference.
- Ensure that each concept is concise (≤35 characters) and output as a plain list without numbering or full sentences.

Output Format Example:
1. two wings
2. a black cap
3. long gray runways
4. a baseball
5. a pattern of spots

Now generate the feature list:

Output
[✓] dog: ['a snout', 'a wagging tail', 'brown or white coat', 'dark eyes', 'floppy ears', 'four legs', 'fur texture', 'long tail', 'pointed ears', 'wet nose']

Figure 7: Example of the multimodal concept generation prompt for the class "dog" in the CIFAR10 dataset. The prompt combines representative images with a natural language instruction to elicit visually grounded, semantically meaningful concepts from GPT.

- **LF-CBM (Oikarinen et al. 2023):** Concepts are generated using `GPT-3.5-turbo-instruct`. We tune key hyperparameters such as `lam`, `clip_cutoff`, and `interpretability_cutoff` via grid search. Other training configurations are standardized to ensure efficiency and consistency across datasets.
- **LM4CV (Yan et al. 2023):** Concepts are generated using `GPT-3.5-turbo-instruct`. We set the number of attributes to five times the number of classes for each dataset. The learning rate is fixed at 0.01, batch size at 4,096, and both training and linear probing epochs at 3,000, with early stopping enabled. Mahalanobis distance regularization is activated with strength 1.
- **DN-CBM (Rao et al. 2024):** Concepts are drawn from a 20,000-word vocabulary adopted by CLIP-Dissect (Oikarinen and Weng 2022), publicly available online.[1]. We adjust the number of probe epochs and learning rate to balance training efficiency and quality. All other settings follow default values.
- **Res-CBM (Shang et al. 2024):** Concepts come from dictionaries, which are built from a base concept bank combining standard TCAV concepts and ConceptNet-derived

[1]https://github.com/first20hours/google-10000-english/blob/master/20k.txt

Dataset	Backbone	K	τ_{merge}	CBL				FCL		
				Batch Size	Max Steps	LR	Weight Decay	Batch Size	Max Iterations	λ
Aircraft	CLIP_RN50	5	0.9997	32	20,000	5e-4	0	256	10,000	7e-4
	CLIP_ViT-L/14	4	0.9998	32	20,000	5e-4	0	256	10,000	7e-4
CIFAR10	CLIP_RN50	5	0.9996	1,024	20,000	4e-4	0	256	10,000	7e-4
	CLIP_ViT-L/14	5	0.9999	32	10,000	5e-4	0	256	10,000	7e-4
CIFAR100	CLIP_RN50	5	0.9997	512	20,000	5e-4	0	256	10,000	7e-4
	CLIP_ViT-L/14	5	0.9997	32	20,000	2e-4	0	256	10,000	7e-4
CUB200	CLIP_RN50	1	0.9998	32	30,000	2e-4	1e-5	256	10,000	8e-4
	CLIP_ViT-L/14	2	0.9998	32	30,000	1e-4	1e-5	256	10,000	9e-4
DTD	CLIP_RN50	1	0.9996	32	10,000	5e-4	0	256	10,000	7e-4
	CLIP_ViT-L/14	5	0.9996	32	5,000	1e-3	0	256	10,000	7e-4
Flower102	CLIP_RN50	1	0.9997	512	20,000	5e-4	0	256	10,000	7e-4
	CLIP_ViT-L/14	0	0.9996	32	30,000	1e-4	1e-5	256	10,000	2e-4
Food101	CLIP_RN50	5	0.9998	32	20,000	5e-4	0	256	10,000	7e-4
	CLIP_ViT-L/14	2	0.9996	32	20,000	1e-4	1e-5	256	10,000	7e-4
HAM10000	CLIP_RN50	3	0.9996	512	20,000	1e-3	0	256	10,000	7e-4
	CLIP_ViT-L/14	1	0.9996	32	20,000	5e-4	0	256	10,000	7e-4
ImageNet	CLIP_RN50	1	0.9997	1,024	60	1e-3	0	512	80	5e-4
	CLIP_ViT-L/14	1	0.9997	1,024	30	1e-4	0	512	80	1e-4
Resisc45	CLIP_RN50	5	0.9997	32	20,000	1e-3	0	256	10,000	7e-4
	CLIP_ViT-L/14	2	0.9998	32	5,000	4e-4	0	256	10,000	7e-4
UCF101	CLIP_RN50	5	0.9997	32	20,000	5e-4	0	256	10,000	7e-4
	CLIP_ViT-L/14	2	0.9997	32	20,000	1e-4	1e-5	256	10,000	7e-4

Table 7: Hyperparameter settings for the Concept Bottleneck Layer (CBL) and Final Classification Layer (FCL) in **PS-CBM** across all datasets, using CLIP_RN50 and CLIP_ViT-L/14 backbones. Note: LR is short for Learning Rate; for ImageNet, Max Steps and Max Iterations indicate the number of epochs.

associations, together with a larger candidate pool extracted from Visual Genome. The residual dimension is set to 30 for ImageNet and 15 for other datasets. Following the original paper, similarity regularization is set to 1 and the candidate number to 5. Additional tuning ensures timely convergence.

- **VLG-CBM (Srivastava, Yan, and Weng 2024):** Concepts are generated by GPT-3.5-turbo-instruct, with ImageNet annotations provided by the authors. The number of hidden layers in the concept bottleneck layer is fixed to one to improve expressiveness and accuracy. The regularization parameter saga lam is set to 0.0007 (or 0.0005 for ImageNet). Automatic weighting is enabled with a positive weight of 0.2 to address class imbalance. The probability of cropping to concept varies: 0.3–0.5 for fine-grained datasets (Aircraft, Flower, Food, HAM10000, and CUB), and 0 otherwise. Additional parameters such as learning rate, epochs, and batch size are tuned per dataset.
- **V2C-CBM (He et al. 2025b):** Concepts (500 per class) are provided by the authors. Weight matrices are randomly initialized, and each class uses 50 concepts uniformly. All other settings follow the original implementation.
- **DCBM (Prasse et al. 2025):** This method uses the same 20,000-word vocabulary as DN-CBM and employs Mask R-CNN for concept generation. Concepts are clustered into 2,048 groups using k-means with median-based centroids. For ImageNet, the batch size is increased to 256 to accommodate the dataset scale.

A.5 Error Bar Analysis on CLIP_RN50 Backbone

To evaluate the robustness and statistical significance of the main results, we conduct additional experiments using five random seeds on three datasets with the smallest performance margins—CIFAR10, Flower102, and Food101—under the CLIP_RN50 backbone. As shown in Table 10, PS-CBM consistently surpasses competing CBMs with minimal variance across runs. Specifically, PS-CBM still outperforms DN-CBM by +1.12% in ACC and +9.22% in CEA, and exceeds VLG-CBM by +0.73% in ACC and +1.60% in CEA, with all standard deviations below 0.2%.

Method	Aircraft		CIFAR10		CIFAR100		CUB200		DTD		Flower102		Food101		HAM10000		ImageNet		Resisc45		UCF101	
	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑
Linear Probe	62.9	–	98.1	–	87.1	–	84.5	–	81.1	–	99.4	–	93.2	–	81.5	–	84.3	–	93.3	–	89.8	–
LaBo	61.4	42.4	97.7	67.1	85.9	59.4	82.0	56.5	77.0	53.4	99.3	68.6	92.4	63.9	80.5	53.0	84.1	57.1	91.5	63.5	90.2	62.3
LF-CBM	53.2	42.3	97.1	69.9	86.1	62.3	77.4	60.2	76.6	59.0	98.1	75.8	91.8	69.5	72.3	58.1	79.8	57.7	91.8	67.6	88.4	64.3
LM4CV	54.7	40.9	96.7	74.6	84.5	63.2	79.3	58.7	79.4	60.1	98.8	73.8	92.2	68.9	66.8	49.8	**84.7**	61.1	92.3	70.0	89.3	66.8
DN-CBM	62.9	43.2	97.7	61.7	86.3	59.3	83.3	58.2	81.6	54.9	**99.5**	68.4	92.2	63.4	81.2	48.4	79.1	56.7	93.4	62.9	89.8	61.7
Res-CBM	54.4	42.7	97.5	68.9	83.0	62.7	75.8	60.1	73.6	56.5	98.6	76.9	89.8	69.8	76.8	52.2	80.6	61.6	91.1	68.5	88.0	68.5
VLG-CBM	63.5	48.3	97.9	73.6	87.0	64.7	84.5	63.4	77.4	59.3	99.5	74.2	92.8	68.6	82.0	61.5	82.5	59.8	92.4	70.8	90.2	67.4
V2C-CBM	57.9	40.0	98.0	67.3	86.1	59.5	80.8	55.7	79.8	55.3	99.2	70.0	92.2	63.7	78.9	51.9	84.3	57.3	92.6	64.3	88.2	61.0
DCBM	57.9	41.2	97.5	63.7	85.2	60.6	80.8	58.4	78.4	54.6	99.1	70.4	92.0	65.4	78.0	48.1	76.5	56.7	91.4	63.6	87.5	62.2
Ours	**65.1**	**48.5**	**98.0**	**74.8**	**87.3**	**64.9**	**85.3**	**64.1**	**82.1**	**61.5**	99.5	**77.7**	**93.0**	**71.1**	**82.8**	**63.9**	84.5	**62.3**	**93.4**	**72.4**	**90.4**	**69.7**

Table 8: ACC and CEA on 11 datasets using `CLIP_ViT-L/14` backbone. **Bold** and underline denote the best and second-best results, respectively.

Method	Avg. ACC (%) ↑	Avg. # Concepts ↓	Avg. CEA (%) ↑
LaBo	85.6	8,241	58.8
LF-CBM	83.0	787	62.4
LM4CV	83.5	824	62.5
DN-CBM	86.1	6,144	58.1
Res-CBM	82.7	**282**	62.6
VLG-CBM	86.4	739	64.7
V2C-CBM	85.3	8,009	58.7
DCBM	84.0	2,048	58.6
PS-CBM	**87.4**	497	**66.4**

Table 9: Comparison of methods by average ACC, number of concepts used, and CEA using `CLIP_ViT-L/14` backbone. **Bold** and underlined indicate the best and second-best results, respectively.

Method	CIFAR10		Flower102		Food101	
	ACC ↑	CEA ↑	ACC ↑	CEA ↑	ACC ↑	CEA ↑
DN-CBM	88.2	55.2	96.5	65.8	82.3	56.1
VLG-CBM	89.5	67.4	97.1	72.4	81.5	60.3
PS-CBM	**89.7**	**68.4**	**97.8**	**74.8**	**82.8**	**61.5**

Table 10: Error bar analysis on the `CLIP_RN50` backbone using five random seeds (std $< 0.2\%$).

These results confirm the stability and statistical reliability of PS-CBM's performance improvements.

B Results with `CLIP_ViT-L/14` Backbone

To validate the robustness of our findings across architectures, we replicate the main experiments using the more powerful `CLIP_ViT-L/14` backbone, in addition to the default `CLIP_RN50`. The performance of PS-CBM and all baselines is summarized in Tables 8 and 9.

Consistent with the RN50-based results, PS-CBM continues to demonstrate strong generalization and concept efficiency. As shown in Table 9, on average across 11 datasets, it outperforms state-of-the-art CBMs by 1.0%–4.7% in classification accuracy (ACC) and by 1.7%–8.3% in Concept-Efficient Accuracy (CEA). From Table 8, at the dataset level, PS-CBM consistently achieves the highest CEA across all benchmarks, underscoring its ability to deliver interpretable predictions without compromising performance. It also secures the top ACC score on most datasets, with only minor deviations on Flower102 and ImageNet, where it ranks second with marginal gaps.

Importantly, PS-CBM accomplishes these results while using substantially fewer concepts compared to methods like LM4CV and DN-CBM, which rely on much larger concept sets to reach similar accuracy. This demonstrates that PS-CBM not only maintains competitive predictive performance but also ensures greater interpretability and compactness, reducing redundancy and mitigating the risk of concept leakage.

Method	Backbone	Param	Datasets										
			Aircraft	**CIFAR10**	**CIFAR100**	**CUB200**	**DTD**	**Flower102**	**Food101**	**HAM10000**	**ImageNet**	**Resisc45**	**UCF101**
LaBo	CLIP_RN50	BS	256	512	512	512	512	256	1,024	256	2,048	256	256
		LR	5e-5	1e-4	1e-4	5e-5	1e-4	1e-5	1e-5	5e-4	1e-5	5e-5	1e-5
		E/S	50,000	15,000	50,000	15,000	15,000	50,000	50,000	20,000	3,000	15,000	50,000
	CLIP_ViT-L/14	BS	256	512	512	512	512	256	1,024	256	2,048	256	256
		LR	5e-5	1e-4	1e-5	5e-5	1e-4	1e-5	1e-5	5e-4	1e-5	5e-5	1e-5
		E/S	10,000	15,000	10,000	5,000	15,000	50,000	5,000	10,000	3,000	15,000	50,000
LF-CBM	CLIP_RN50	BS	256	256	256	256	256	256	256	256	256	256	256
		LR	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
		E/S	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000
	CLIP_ViT-L/14	BS	256	256	256	256	256	256	256	256	256	256	256
		LR	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
		E/S	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000
LM4CV	CLIP_RN50	BS	4,096	4,096	4,096	4,096	4,096	4,096	4,096	4,096	4,096	4,096	4,096
		LR	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2
		E/S	3,000	3,000	3,000	3,000	3,000	3,000	3,000	3,000	3,000	3,000	3,000
	CLIP_ViT-L/14	BS	4,096	4,096	4,096	4,096	4,096	4,096	4,096	4,096	4,096	4,096	4,096
		LR	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2
		E/S	3,000	3,000	3,000	3,000	3,000	3,000	3,000	3,000	3,000	3,000	3,000
DN-CBM	CLIP_RN50	BS	512	512	512	512	512	512	512	512	512	512	512
		LR	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2
		E/S	500	500	500	500	500	500	500	500	1,000	500	500
	CLIP_ViT-L/14	BS	512	512	512	512	512	512	512	512	512	512	512
		LR	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2
		E/S	500	500	500	500	500	500	500	500	1,000	500	500
Res-CBM	CLIP_RN50	BS	128	128	128	128	128	128	128	128	1,024	128	128
		LR	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2
		E/S	50	50	50	50	50	50	50	50	50	50	50
	CLIP_ViT-L/14	BS	128	128	128	128	128	128	128	128	1,024	128	128
		LR	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2
		E/S	50	50	50	50	50	50	50	50	50	50	50
VLG-CBM	CLIP_RN50	BS	32	32	32	32	32	32	32	32	64	32	32
		LR	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	1e-2	5e-4	5e-4
		E/S	300	100	100	100	100	300	200	100	20	100	100
	CLIP_ViT-L/14	BS	32	32	32	32	32	32	32	32	64	32	32
		LR	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	1e-2	5e-4	5e-4
		E/S	100	10	10	100	20	100	15	100	15	100	100
V2C-CBM	CLIP_RN50	BS	256	512	512	512	512	256	1,024	256	1,024	256	256
		LR	5e-6	1e-4	1e-5	5e-5	1e-4	5e-5	1e-5	5e-4	1e-5	5e-5	1e-5
		E/S	10,000	50,000	50,000	15,000	15,000	20,000	50,000	20,000	2,000	15,000	50,000
	CLIP_ViT-L/14	BS	256	512	512	512	512	256	1,024	256	1,024	256	256
		LR	5e-5	1e-4	1e-5	5e-5	1e-4	1e-5	1e-5	5e-4	1e-5	5e-5	1e-5
		E/S	10,000	50,000	50,000	15,000	15,000	20,000	50,000	20,000	2,000	15,000	50,000
DCBM	CLIP_RN50	BS	128	128	128	128	128	128	128	128	256	128	128
		LR	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
		E/S	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	CLIP_ViT-L/14	BS	128	128	128	128	128	128	128	128	256	128	128
		LR	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
		E/S	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000

Table 11: Hyperparameter settings for all baselines across all 11 datasets, including Batch Size (BS), Learning Rate (LR), and Epochs/ Steps (E/S). Each method is evaluated with two backbones (CLIP_RN50 and CLIP_ViT-L/14), and each backbone is represented by three rows corresponding to different training components.