This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Language Guided Concept Bottleneck Models for Interpretable Continual Learning

Lu Yu [1] Haoyu Han [1] Zhe Tao [1] Hantao Yao [2] Changsheng Xu [3]

[1]School of Computer Science and Engineering, Tianjin University of Technology
[2]School of Information Science and Technology, University of Science and Technology of China
[3]State Key Laboratory of Multimodal Artificial Intelligence Systems,
Institute of Automation, University of Chinese Academy of Sciences
{luyu@email, fishercat@stud, tz@stud}.tjut.edu.cn,
yaohantao@ustc.edu.cn, csxu@nlpr.ia.ac.cn

Abstract

Continual learning (CL) aims to enable learning systems to acquire new knowledge constantly without forgetting previously learned information. CL faces the challenge of mitigating catastrophic forgetting while maintaining interpretability across tasks. Most existing CL methods focus primarily on preserving learned knowledge to improve model performance. However, as new information is introduced, the interpretability of the learning process becomes crucial for understanding the evolving decision-making process, yet it is rarely explored. In this paper, we introduce a novel framework that integrates language-guided Concept Bottleneck Models (CBMs) to address both challenges. Our approach leverages the Concept Bottleneck Layer, aligning semantic consistency with CLIP models to learn human-understandable concepts that can generalize across tasks. By focusing on interpretable concepts, our method not only enhances the model's ability to retain knowledge over time but also provides transparent decision-making insights. We demonstrate the effectiveness of our approach by achieving superior performance on several datasets, outperforming state-of-the-art methods with an improvement of up to 3.06% in final average accuracy on ImageNet-subset. Additionally, we offer concept visualizations for model predictions, further advancing the understanding of interpretable continual learning. Code is available at https://github.com/FisherCats/CLG-CBM.

1. Introduction

In dynamic and ever-changing environments, models trained on fixed datasets often struggle to adapt to new tasks or data without losing previously acquired knowledge, facing the challenge known as catastrophic forgetting [32]. Continual learning [8, 31, 71] addresses this limitation by allowing systems to retain and incrementally build upon past knowledge, resulting in greater flexibility. This capability is especially critical in real-world applications like robotics, autonomous systems, and personalized healthcare, where models must continuously learn from ongoing experiences and data streams to stay effective. Additionally, continual learning improves the transfer of knowledge across tasks, minimizing the need for frequent retraining and enhancing computational efficiency. These benefits make continual learning a promising approach for advancing intelligent and adaptive AI systems in real-world.

Existing continual learning methods [26, 39, 42, 56, 64] focus on mitigating the issue of catastrophic forgetting yet the underlying mechanisms behind this phenomenon remain largely unexplained. As these models continuously update their knowledge, it becomes crucial to understand what they learn and how they retain previous information to prevent unintended behavior and enhance human interpretability. Furthermore, understanding the internal mechanisms of continual learning models can guide improvements in their design, leading to more robust and effective systems. Despite the importance of this challenge, few papers have explored the interpretability of continual learning models. ICICLE [44] is one such work, which interprets concept drift in continual learning through a prototypical-part network and introduces interpretability regularization to minimize changes in prototype similarities.

Concept Bottleneck Models (CBMs) [22, 54] are a form of interpretable machine learning model that enhance transparency by organizing the learning process around human-understandable concepts. Rather than directly mapping

inputs to predictions, CBMs incorporate an intermediate 'bottleneck' layer composed of high-level, explicitly defined concepts that are meaningful to humans. Some recent research [36, 47, 58, 61] integrates language-guided approaches with CBMs to further align the model's decision-making process with human cognition, particularly for tasks involving abstract or semantic knowledge. Leveraging the inherent interpretability and structured knowledge representation of CBMs, this paper investigates their potential application to continual learning, with the goal of enhancing both interpretability and adaptability while ensuring the retention of previously acquired knowledge.

With the rapid advancements in large-scale multimodal models, we can harness their robust zero-shot encoding capabilities to establish a strong foundation for extracting high-quality features. In this paper, we integrate the pre-trained CLIP [38] model to obtain bottleneck concepts within a continual learning framework. Specifically, we utilize pre-trained language models (e.g., ChatGPT [5]) to generate human-understandable concepts for each category. These concepts are subsequently encoded into concept embeddings through the CLIP text encoder, after which the most informative and expressive concepts are selected to form a task-specific bottleneck. We construct the Concept Bottleneck Layer (CBL) by aligning the concept score matrix with CLIP concept activiation matrix, enhancing model's interpretability. Furthermore, we leverage semantic knowledge to augment prototypes, effectively mitigate catastrophic forgetting. Finally, our approach achieves superior performance on seven benchmark datasets, consistently maintaining interpretability throughout the process.

Our main contributions can be summarized as follows:

- We introduce a novel framework that leverages language-guided Concept Bottleneck Models to enhance both interpretability and the ability to mitigate catastrophic forgetting in continual learning.
- We construct the Concept Bottleneck Layer by aligning semantic consistency with CLIP models, enabling the learning of human-understandable concepts across tasks.
- Our method outperforms state-of-the-art methods on the several datasets. Additionally, we provide concept visualizations for model predictions to enhance the understanding of interpretable continual learning.

2. Related Work

2.1. Continual Learning

There are three main types of existing continual learning methods. *Regularization-based* methods [2, 10, 14, 17, 21, 26, 29, 51, 66–68, 72] typically alleviate catastrophic forgetting either by constraining the model's outputs through knowledge distillation or by penalizing changes to critical parameters based on their assessed importance. *Rehearsal-based* methods [6, 16, 28, 41, 42, 60] involve replaying data from previous tasks by storing a subset of past data or by generating synthetic data, to prevent forgetting. The replayed data can be used at training stage with the current task data, maintaining the model's performance on previous tasks. *Architecture-based* methods [30, 59, 63] adapt the model's structure dynamically as new tasks are introduced, such as activating different part of model parameters for different tasks, using modular designs to compartmentalize learning and minimize interference between tasks.

Methods based on pre-trained models (PTMs) have made significant progress recently by leveraging robust features extracted from large-scale PTMs. Among these methods, approaches [13, 27, 49, 55, 56] based on *Parameter-Efficient Tuning* typically construct and train a set of PEF modules, improving performance while reducing computational costs. *Representation-based* methods [52, 69, 70], on the other hand, usually utilize the robust features of PTMs to construct classifiers.

The methods discussed above effectively alleviate the problem of catastrophic forgetting, a central challenge in continual learning. However, the decision-making process of those methods is still a black box and not transparent, and the rationality for model predictions shifts over time due to catastrophic forgetting, making it difficult to improve interpretability in continual learning. To address this issue, ICICLE [44] has explored interpretable continual learning methods by introducing a prototypical parts-based approach, they further proposed interpretability regularization and proximity-based prototype initialization, regularizing model to activate similar prototypical parts learned previously to mitigate Interpretability Concept Drift (ICD). Although ICICLE provides interpretability, it substantially limits the plasticity of the model, highlighting the need for methods that are both interpretable and effective.

2.2. Interpretable Deep Learning

There are two types of interpretable models that have been well-developed in the domain of deep learning explanations: post-hoc models and self-explainable models. Post-hoc models aim to elucidate the reasoning process of black-box methods, using techniques such as saliency maps [45, 46], concept activation vectors [18, 22, 62], counterfactual examples [1, 15, 33] and prototype similarity [7, 11, 43], etc. While self-explainable models [3, 19, 34] aim to make the model intrinsically interpretable by faithfully representing the entire classification behavior, they are often more complex than post-hoc models.

Among these interpretable models, Concept Bottleneck Models [22, 48, 58, 61, 65] provide explanations of the model's decision-making process in a straightforward manner. CBMs are designed to be interpretable, incorporating an intermediate Concept Bottleneck Layer (CBL), where

each neuron represents a high-level, expressive concepts. Image features are projected into Concept Space to acquire concept score vectors, which are utilized to make final classification. Recent research [36, 58, 61] has integrated textual knowledge with CBMs, addressing the challenges of obtaining class concepts and annotations by querying LLMs (e.g. chatGPT) and utilizing VLM (e.g. CLIP) to encode images and concepts. The similarity between the image and the concepts is calculated to determine the probability of each concept's presence in the image, producing a similarity vector that serves for interpretation and classification, making the decision making process more transparent.

3. Preliminary

3.1. Class-Incremental Learning

We focus on class incremental learning (CIL) in this paper, where the learning process unfolds across multiple tasks. For CIL scenario with n tasks, let $\mathcal{X}_t$ denote the input space, and $\mathcal{Y}_t$ represent the set of labels or classes observed by the model at task t. The data associated with task t can be denoted as $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$, where $x_i \in \mathcal{X}_t$ and $y_i \in \mathcal{Y}_t$. At each task t, the model is introduced to a new set of classes $\mathcal{Y}_t$, which it learns to recognize these new classes while retaining knowledge of previously encountered classes.

The sequence of tasks is denoted as $\mathcal{T} = \{1, 2, \ldots, n\}$. In CIL, classes included in separate tasks are non-overlapping, where $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset \quad (\forall\ i \neq j)$. The entire class space after task t is represented as: $\mathcal{Y}_{1:t} = \mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \cdots \cup \mathcal{Y}_t$. During training for task t, the model has access only to the current task's dataset $\mathcal{D}_t$, but it needs to predict labels from the cumulative set of classes $\mathcal{Y}_{1:t}$.

3.2. Concept Bottleneck Models

Compared to conventional deep learning architectures, Concept Bottleneck Models (CBMs) [22, 54] introduce a Concept Bottleneck Layer (CBL) positioned between feature extractor and classifier. Concept Bottleneck Layer comprises neurons that represent human-understandable concepts. These neurons convert image features into concept scores, which quantify the degree of alignment between the image and predefined concepts. The classifier then leverages these concept scores to make predictions, thereby enhancing the interpretability of the decision-making process. Language-Guided CBMs [36, 47, 58, 61] extend this framework by integrating natural language processing (NLP) with concept-based modeling to further enhance interpretability and performance. A notable example is the CLIP-based CBMs [58, 61], which employs the pre-trained CLIP model as its backbone. For instance, CLIP-based CBM method [58] initially gathers relevant concepts by querying ChatGPT, followed by a concept selection module which is developed to identify the most informative and discriminative concepts. Specifically, a simple Multi-Layer Perceptron is trained in [58] to capture the semantic knowledge embedded in images, guided by both Cross-Entropy loss and Mahalanobis loss. The acquired semantic knowledge is subsequently utilized for concept selection, constructing a concept pool $\mathcal{C}$ to facilitate model interpretability. The text features of $\mathcal{C}$ serves as the CBL in CLIP-based CBMs, and the size of CBL is determined by the quantity of selected concepts for all seen tasks.

The CLIP concept activation matrix is computed as the dot product between the image features $f_I(\mathcal{X})$ and text features $f_T(\mathcal{C})$ as follows, enabling the model to align visual inputs with their corresponding semantic concepts:

$$E_{clip} = f_I(\mathcal{X}) \cdot f_T(\mathcal{C})^\top \tag{1}$$

The final predication $\hat{y}$ can be calculated as follows:

$$\hat{y} = \text{argmax}\ \sigma(E_{clip} \cdot W_l^\top), \tag{2}$$

where W_l refers to the weight of classifier, σ refers to sigmoid function.

4. Method

4.1. Overview Framework

We introduce Language-Guided Concept Bottleneck Models as an efficient framework for achieving interpretable continual learning. CBMs are designed to learn human-understandable intermediate representations, known as concepts, which serve as the basis for predictions and enhance transparency. However, while CBMs improve interpretability, applying them directly in a continual learning setting can still lead to catastrophic forgetting. To mitigate this, we propose a *semantic-guided prototype augmentation module* to generate pseudo-features for old classes by leveraging semantic similarity with new data. Additionally, we introduce a *concept alignment module* during the learning of the Concept Bottleneck Layer (CBL) to enhance neuron interpretability. This alignment enforces neurons to activate in response to target concepts, resulting in clearer concept representation while simultaneously improving predictive accuracy. Together, these modules aim to preserve past knowledge and improve both interpretability and performance in continual learning scenarios.

As shown in Figure 1, we use the frozen CLIP image encoder f_I and text encoder f_T to extract image and concept features. When a new task arrives, we generate key concepts for each class using ChatGPT, selecting the most expressive and informative concepts $\mathcal{C}_t$ to form a task-specific "concept bottleneck" using a learning-to-search algorithm, as proposed in [58]. We save the bottlenecks that learned at each task for continual interpretability. We then construct

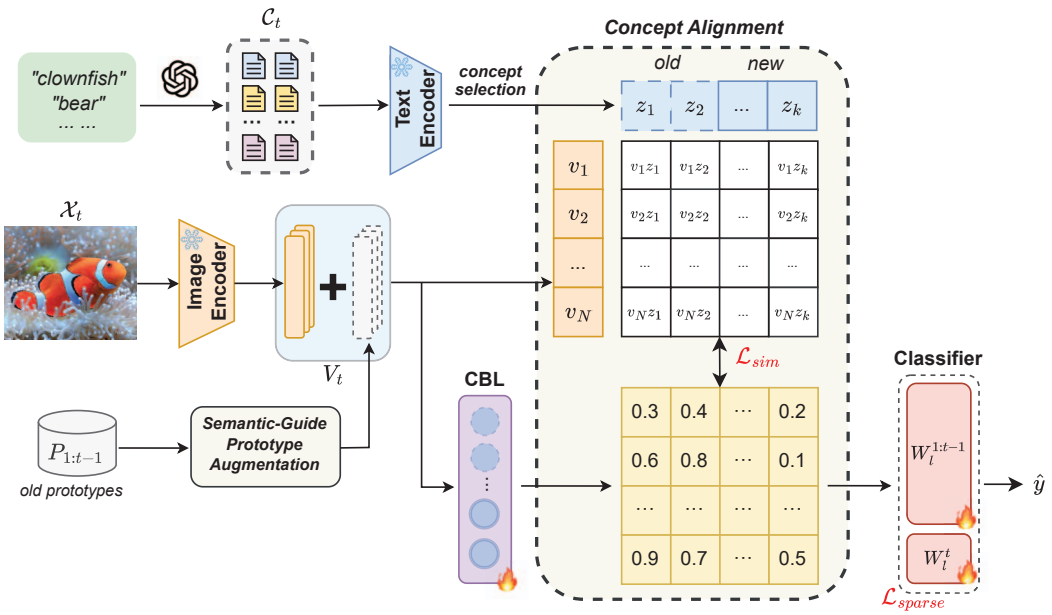


Figure 1. The framework of our method for task t. The concept alignment module aligns the concept score matrix with the CLIP concept activation matrix. The semantic-guided prototype augmentation module leverages semantic knowledge to identify the most semantically similar class within $\mathcal{Y}_t$, generating pseudo-features for previously learned classes to mitigate catastrophic forgetting.

the Concept Bottleneck Layer (CBL) to represent the relationships between learned classes and concepts, using both the current features and generated pseudo features of old classes. Inspired by [36], the concept score matrix output by the CBL is aligned with the CLIP concept activation matrix via a similarity loss $\mathcal{L}_{\text{sim}}$. Finally, the classifier maps these concept scores to predicted classes using cross-entropy loss $\mathcal{L}_{\text{ce}}$ and a sparsity loss $\mathcal{L}_{\text{sparse}}$, inspired by [57], to encourage learning of key features and interpretable classifier.

4.2. Model Training

Concept Bottleneck Layer Construction. We construct the Concept Bottleneck Layer (CBL) using the bottlenecks collected across tasks to explain our model's behavior while enhancing its performance. The bottlenecks from tasks $\mathcal{T}_{1:t}$ are represented as $B_t = [z_1, z_2, \ldots, z_k] \in \mathbb{R}^{|\mathcal{C}_{1:t}| \times D}$, $k = |\mathcal{C}_{1:t}|$, where D refers to the dimension of feature embedding. We save all previously learned bottlenecks, creating a cumulative concept space that preserves essential semantic knowledge to mitigate catastrophic forgetting. These bottlenecks also improve generalization to new tasks by leveraging a broader set of interpretable features accumulated across tasks. We use the frozen CLIP image encoder f_I to extract image features $f_I(\mathcal{X}_t)$ and project these into the CBL to produce concept scores E, capturing the model's understanding of relationships between images and concepts, as formulated below:

$$E = f_I(\mathcal{X}_t) \cdot W_C^\top, \; E \in \mathbb{R}^{N_t \times |\mathcal{C}_{1:t}|} \tag{3}$$

where W_C represents the weight matrix of CBL, which consists of a single linear layer. Finally, the classifier g maps E to the class space, yielding the predicted labels $\hat{y}$ for the images.

Since the concepts are acquired incrementally under CIL scenario, both the CBL and the classifier need to expand to accommodate the learning of new concepts and categories, ensuring that they accurately map the embeddings while retaining the learned knowledge from the previous task.

Concept Alignment. To guide the CBL toward human-understandable bottlenecks and enhance interpretability, we align the learned concept scores with CLIP concept activation scores. This alignment maintains semantic consistency across tasks and improves robustness by anchoring learning to stable, meaningful concepts. By using CLIP, we can easily acquire the CLIP concept activation matrix E_{clip} by equation (1). We train CBL to align with the CLIP concept score matrix E_{clip}, transferring the general knowledge to the output E of CBL by a similarity loss $\mathcal{L}_{sim}$, defined as:

$$\begin{aligned} \mathcal{L}_{sim} &= -\frac{1}{|\mathcal{C}_{1:t}|} \sum_{i=1}^{|\mathcal{C}_{1:t}|} \cos(\hat{E}^i, \hat{E}^i_{clip}) \\ &= -\frac{1}{|\mathcal{C}_{1:t}|} \sum_{i=1}^{|\mathcal{C}_{1:t}|} \frac{\hat{E}^i \cdot \hat{E}^i_{clip}}{||\hat{E}^i||_2 \cdot ||\hat{E}^i_{clip}||_2} \end{aligned} \tag{4}$$

Here, $\hat{E} = E^3$ and $\hat{E}_{clip} = E^3_{clip}$ to further sharp the concept scores distribution following [36].

The classifier in CBM connects the concept space and

the class space, can be understood as the contribution of concepts to distinguish categories. To encourage more interpretable classifier and transparent decision-making process, we regularize the final layer of our model using elastic-net penalty proposed by [57] as shown below:

$$\mathcal{L}_{sparse} = \phi||W_l||_1 + \frac{1}{2}(1-\phi)||W_l||_F \tag{5}$$

where W_l is the weight of final layer, ϕ is a trade-off parameter, we set ϕ to 0.99 for our method.

Semantic-guided Prototype Augmentation. Prototype augmentation or compensation is commonly employed in class-incremental learning (CIL) to retain previously acquired knowledge, as prototypes from earlier tasks may become inaccurate over time due to feature drift. Research in [37, 64, 72] demonstrates that feature drift within feature extractor can be estimated using data from the current task, resulting in significant performance gains. Building on this insight, we propose leveraging semantic knowledge to augment prototypes, thereby mitigating catastrophic forgetting and align concepts across all categories. Specifically, we propose a semantic-guided prototype augmentation strategy to generate pseudo-features for previously learned classes. We assume that classes with similar semantic information exhibit similar embedding distributions. Therefore, we first identify the category most similar to a previous class $j \in \mathcal{Y}_{1:t-1}$ by calculating similarity scores between the embedding of the old class name and the prototypes of newly learned classes as follows:

$$h = \underset{i \in \mathcal{Y}_t}{\arg\max}(\cos(f_T(y^j), p^i)) \tag{6}$$

where y^j refers to the class name of class j, p_i refers to the prototype of class i.

Then we compute the discrepancy between the image features V^h of class h in the current task t and the prototype of class h. We add this discrepancy to the prototype of class j to obtain the generated pseudo features $\tilde{V}^j$, where j is the class most semantically similar to h. The generated pseudo features $\tilde{V}^j$ of previous class j can formulates as follows:

$$\tilde{V}^j = p^j + V^h - p^h \tag{7}$$

After augmenting prototypes, we train our model on the current task data $\mathcal{D}_t$ along with the generated pseudo features guiding the learning process with cross-entropy loss as follows:

$$\hat{y} = V_t \cdot W_C^\top \cdot W_l^\top \tag{8}$$

where V_t represents the feature set which involves the image features of classes in $\mathcal{Y}_t$ and augmented prototypes of classes in $\mathcal{Y}_{1:t-1}$. Overall, the final loss function can be described as:

$$\mathcal{L} = \mathcal{L}_{ce}(\hat{y}, y) + \lambda\mathcal{L}_{sim} + \sigma\mathcal{L}_{sparse} \tag{9}$$

where $y \in \mathcal{Y}_{1:t}$, λ and σ are trade-off weights for $\mathcal{L}_{sim}$ and $\mathcal{L}_{sparse}$ respectively.

5. Experiments

5.1. Experiments Setup

Evaluation Benchmarks. We evaluated our method on three coarse-grained datasets: CIFAR-100 [24], Tiny-ImageNet [25], ImageNet-subset [9]. The first two datasets contain 100 classes each, while the latter contains 200 classes. In addition, we performed comprehensive evaluations on four fine-grained datasets: CUB-200 [53], Flower [35], Food-101 [4], Stanford-cars [23]. There are 200 classes in CUB-200, 101 classes in Food-101, 102 classes in Flower, 196 classes in Stanford-cars.

We adopt the exemplar-free class incremental learning setting, where no exemplars are retained for previously learned classes. The dataset is split according to the format "B-*m* Inc-*n*", where *m* denotes the number of classes included in the initial task, and *n* represents the number of classes included in each incremental task. For each dataset, we perform experiments using two different splitting strategies: (1) large *m* with small *n*, and (2) *m* = *n*. All experiments we conducted are based on same random seed 1993 for fair comparison.

Implementation Details. We developed our method and reproduced other method with Pilot [50]. All experiments were conducted on an NVIDIA RTX 3090. Unless otherwise specified, we ran all experiments using CLIP ViT-B/16 as the backbone for all methods. CLIP RN-50 was used as backbone when comparing to ICICLE, which is designed for CNN-based model. We train our model using the Adam optimizer [20] with a batch size of 64, a learning rate of 0.001, and a total of 60 training epochs. We set the trade-off weight of similarity loss λ to 1 and that of sparsity loss σ to 0.001. The prompts we used to query ChatGPT and examples of generated concepts are provided in Supp. Mat. 8.1. More implementation details can be found in Supp. Mat. 8.2, including the implementation of the concept selection module and the pseudo code of the algorithm.

Evaluation Metric. Following the evaluation protocol of previous works [56, 69, 70], we report the average incremental accuracy, denoted as $\bar{A} = \frac{1}{n}\sum_1^n A_t$, where A_t represents the mean accuracy on all learned categories after learning task *t*, n denotes the number of tasks. Additionally, we also use A_{last} to represent the average accuracy of all categories after learning the last task n.

5.2. Comparison with State-of-the-Art Methods

Evaluation on Coarse-grained Datasets. In Table 1, we report the average incremental accuracy $\bar{A}$ and the final average accuracy A_{last} of our method compared with state-of-the-art methods on three datasets: CIFAR-100, Tiny-

Table 1. Performance comparison on three coarse-grained datasets, the best performance is shown in bold, the second-best performance is underlined. All methods are implemented without using exemplars. We replace the backbones of all methods to CLIP ViT-B/16.

Methods	CIFAR-100				Tiny-ImageNet				ImageNet-subset			
	B-10 Inc-10		B-50 Inc-5		B-10 Inc-10		B-100 Inc-10		B-10 Inc-10		B-50 Inc-5	
	$\bar{A}$	A_{last}	$\bar{A}$	A_{last}	$\bar{A}$	A_{last}	$\bar{A}$	A_{last}	$\bar{A}$	A_{last}	$\bar{A}$	A_{last}
L2P[56]	76.56	65.75	61.85	44.19	69.66	60.36	62.29	55.13	71.28	52.24	65.89	49.08
DualPrompt[55]	81.41	70.34	64.05	43.86	74.06	66.08	65.35	56.15	72.86	54.20	64.64	49.48
CODA-Prompt[49]	82.13	72.34	65.49	49.72	75.18	66.65	61.47	48.31	71.17	52.98	64.64	41.40
CPP[27]	75.73	67.50	69.57	66.26	68.70	61.23	66.61	63.48	83.45	75.80	79.74	73.78
LAE[13]	82.65	72.60	67.35	50.96	76.81	68.98	63.38	49.52	78.29	62.94	64.25	47.90
Continual-CLIP[52]	75.15	66.68	70.79	66.68	63.63	55.91	58.68	55.91	84.98	75.40	81.35	75.40
SLCA[69]	83.13	72.01	**80.07**	71.24	68.99	54.50	60.34	48.53	83.19	69.44	80.81	73.78
EASE[70]	**85.07**	**77.31**	76.72	70.50	**79.88**	**72.99**	70.42	64.12	84.80	70.82	63.74	56.48
Ours	84.49 ± 0.26	76.82 ± 0.50	79.07 ± 0.34	**75.91** ± 0.50	79.28 ± 0.86	71.98 ± 0.25	**75.64** ± 0.17	**71.97** ± 0.09	**86.83** ± 1.32	**78.97** ± 0.39	**81.85** ± 0.76	**78.21** ± 0.29

Table 2. Average incremental accuracy comparison on four fine-grained datasets, the best performance is shown in bold, the second-best is underlined. All methods are implemented without using exemplars. We replace the backbones of all methods to CLIP ViT-B/16.

Methods	CUB-200		Flower		Stanford-cars		Food-101	
	B-10 Inc-10	B-100 Inc-10	B-10 Inc-10	B-50 Inc-5	B-14 Inc-14	B-100 Inc-10	B-10 Inc-10	B-50 Inc-5
L2P[56]	62.08	59.38	84.37	76.70	64.42	61.82	79.48	69.72
DualPrompt[55]	64.95	61.85	89.64	79.16	76.94	68.46	86.27	69.66
CODA-Prompt[49]	67.22	59.82	88.57	77.70	76.44	60.80	87.76	67.22
CPP[27]	83.60	75.03	94.95	93.30	84.75	77.49	90.21	86.60
LAE[13]	66.45	59.98	86.79	77.55	77.25	80.28	88.41	66.26
Continual-CLIP[52]	69.41	60.35	78.72	74.06	86.43	69.79	92.04	89.96
SLCA[69]	80.53	76.85	92.77	82.14	84.74	70.59	74.49	76.28
EASE[70]	83.87	66.14	94.86	77.05	86.22	64.32	91.74	81.72
Ours	**85.40** ± 0.61	**82.20** ± 0.65	**95.58** ± 0.20	**94.53** ± 0.20	**88.60** ± 0.55	**85.07** ± 0.82	**92.25** ± 0.32	**90.97** ± 0.42

ImageNet, ImageNet-subset. As shown, our approach demonstrates either the best or second-best performance across all settings and datasets. On ImageNet-subset, we achieve at least a 1.79% improvement on $\bar{A}$ and 3.06% on A_{last} compared to existing methods. Our results on CIFAR-100 and Tiny-ImageNet are comparable to those methods that prioritize performance improvement without considering interpretability. In contrast, our method achieves a balanced approach, enhancing both performance and interpretability.

Evaluation on Fine-grained Datasets. We also conduct experiments on four fine-grained datasets: CUB-200, Flowers, Stanford Cars, and Food-101. As shown in Table 2, it is clear that our method achieves superior results across various data splits and datasets, particularly on CUB-200 and Stanford Cars, where the gains reach 4.43% and 4.69%, respectively. This suggests that our language-guided CBM-based approach provides significant benefits for fine-grained dataset classification in a continual learning setting.

Comparison with Interpretable Methods. We further compared our proposed method with ICICLE[44], a recently proposed interpretability-driven continual learning method. The original ICICLE utilizes a ResNet pre-trained on ImageNet as its backbone. To ensure a fair comparison, we replace the backbone in both ICICLE and our method with the pre-trained CLIP RN50. As shown in Table 3, we can observe that our method significantly outperforms ICICLE on both benchmark datasets and demonstrates greater stability to various data split.

Table 3. Average incremental accuracy comparison with ICICLE, the best performance is shown in bold. We replace the backbone of our method and ICICLE to pre-trained CLIP RN50.

Methods	CUB-200		Stanford-cars	
	B-50 Inc-50	B-20 Inc-20	B-49 Inc-49	B-14 Inc-14
ICICLE [44]	39.31	20.85	49.58	24.22
Ours	**66.21**	**69.72**	**73.91**	**75.26**

Moreover, our method requires fewer trainable parameters than ICICLE, which necessitates training the entire model. Additionally, our method offers greater flexibility, as it is architecture-agnostic. Unlike ICICLE, a prototypical-part-based approach constrained to CNN architectures, our method can utilize either a Vision Transformer [12] or ResNet as the backbone.

Additional experimental results are provided in Supp. Mat.8.3, including dynamic performance curves under different settings, results obtained using various versions of description generation models, comparisons with joint training, and ablation studies on different modules.

5.3. Further Analysis

Performance-Parameter Comparison. Figure 2 demonstrates the Performance-Parameter comparison between our

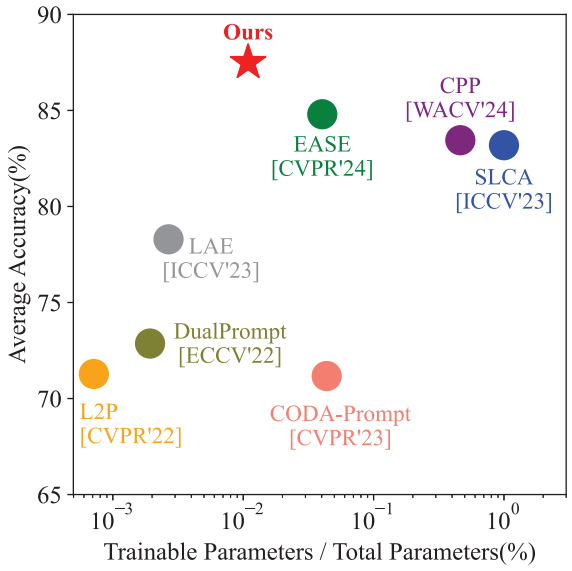


Figure 2. Performance-Parameter comparison of our method and benchmark methods on ImageNet-subset B-10 Inc-10.

method and other state-of-the-art methods after training on ImageNet-subset B-10 Inc-10. Methods with fewer trainable parameters and higher accuracy are considered superior. From Figure 2 we can observe that all the compared methods, our method achieves best performance with relatively less trainable parameters. Since our model is designed to select concepts from a concept pool, approximately half of its trainable parameters are allocated to learning to identify expressive and informative concepts. The parameter requirement could be further reduced by using pre-defined concepts instead of training concept selection for each task. We also provide the computational overhead of the concept selection module in Supp. Mat. 8.4.

Effect of Concept Quantity. Table 4 manifests the effect of different concept quantities to the performance. We report $\bar{A}$ of our method with 10,20,50 and 100 concepts per task on CUB-200 B-10 Inc-10 and ImageNet-subset B-10 Inc-10. We observe that performance improves as the number of concepts learned increases. On CUB-200 dataset, performance shows a dramatic improvement with more concepts per task, eventually plateauing. For ImageNet-subset, the results remain relatively stable as the number of concepts increases. Based on these observations, we choose to adapt 100 concepts per task across all datasets, striking a balance between concept diversity and computational efficiency.

5.4. Understanding Model Interpretability

Interpretable Model Prediction. Inspired by [36, 40, 58], We analyze the decision-making process of our method by highlighting the contribution of each concept to the final classification result. Given an input image x of class k, we

Table 4. The effect of the number of concepts on the performance of the method.

Concepts per task	CUB-200 B-10 Inc-10	ImageNet-subset B-10 Inc-10
10	79.32	86.25
20	83.70	86.80
50	85.70	87.24
100	**86.07**	**87.40**

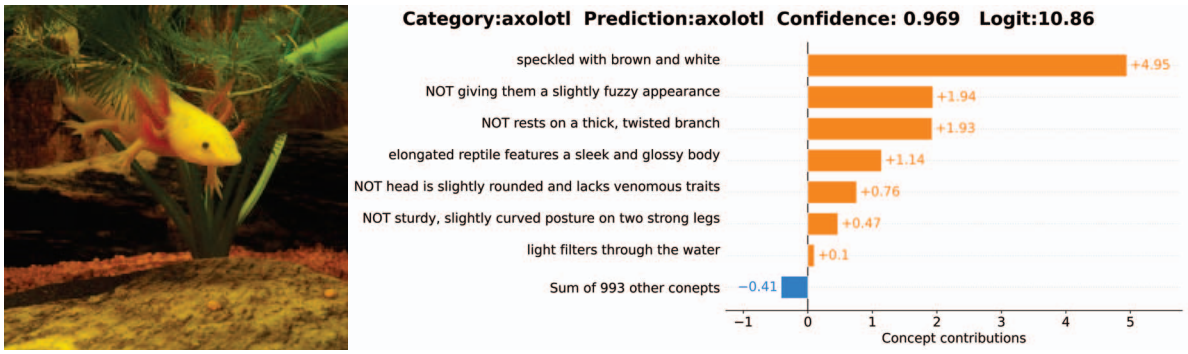
denote the contribution of concept c_i to class k as $Con_{i,k}$, which is computed as follows:

$$Con_{i,k} = f_I(x) \cdot (W_C^i)^\top \cdot W_l^{k,i} \tag{10}$$

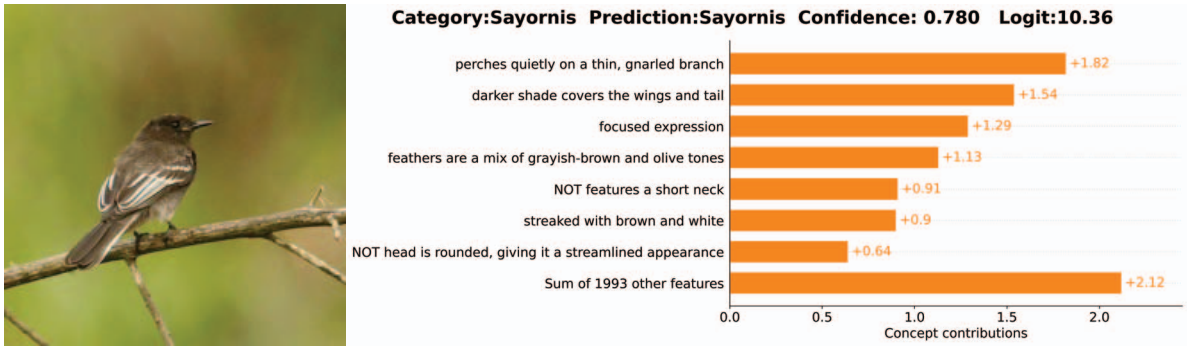
Here, W_C^i denotes the i-th row of the weight matrix of the CBL, and $W_l^{k,i}$ represents the connection weight between class k and concept i in the final layer. The value of $Con_{i,k}$ can be either positive or negative, indicating positive or negative relevance to class k, respectively. A higher absolute value of $Con_{i,k}$ signifies greater importance of the concept for that class.

As illustrated in Figure 3, we present the top-7 most relevant concepts for an *axolotl* image from ImageNet-subset and a *sayornis* image from CUB-200 after the entire training process. Concepts are listed in descending order of absolute contribution, with concepts labeled as "NOT" indicating a negative association to the image. We can see that the concept "speckled with brown and white" contributes most to the class *axolotl* as shown in Figure 3a, and "perches quietly on a thin, gnarled branch" is highly activated to class *sayornis* in Figure 3b, the decision-making process is illustrated through human-interpretable concepts, providing insights into the model's reasoning. Furthermore, lower relevant concepts have minimal impact, as demonstrated by the cumulative negative contribution of 993 other features in Figure 3a, indicating a weak association with *axolotl*. Although the cumulative contribution of 1993 additional features in Figure 3b exceeds that of the top-7 concepts, the contribution of each individual feature within the 1993 concepts is marginal. Notably, negative concepts also play a role in enhancing model performance and confidence, reflecting their significance in the interpretability framework. More examples of interpretable model predictions are provided in Supp. Mat. 8.5.

Concepts Change Across Tasks. As illustrated in Figure 4, we present the variation in concept relevance for the class *gazania* on the Flower B-25 Inc-25, category *gazania* is learned at first task. Concepts are ordered from highest to lowest based on their contribution values. Our method demonstrates robust interpretability, as concepts with high contribution scores in the initial task maintain relatively high contributions even after the completion of the final task. Additionally, the negative concepts also show in-



(a) Contribution visualization on ImageNet-subset



(b) Contribution visualization on CUB-200

Figure 3. (a) Contribution visualization of *axolotl* after training on ImageNet-subset B-10 Inc-10. (b) Contribution visualization of *sayornis* after training on CUB-200 B-10 B-10.

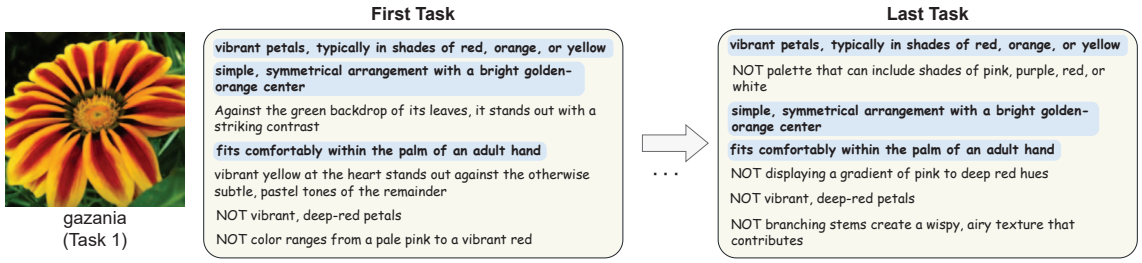


Figure 4. Top-7 concepts change of *gazania* after training the first and all the tasks on Flower B-25 Inc-25.

creased contribution compared to the initial task, thereby alleviating forgetting of previous knowledge. It indicates that our framework help alleviate catastrophic forgetting by focusing on stable concepts that are shared across tasks, decoupling task-specific information, and promoting the retention of useful knowledge over time.

6. Conclusion

In this paper, we introduce a novel framework based on Language-Guided Concept Bottleneck Models for interpretable continual learning. Our approach enhances interpretability by aligning the concept score matrix generated by the concept bottleneck layer to the CLIP activation score matrix, while also learning a sparse linear classifier. Additionally, we propose a semantic-guided prototype augmentation to generate pseudo features for the previous tasks. Extensive experiments validate the effectiveness of our method, which maintains interpretability throughout the continual learning process.

Limitation and Future Work. Our approach builds upon Concept Bottleneck Models, making the model's interpretability inherently dependent on the quality of the selected concepts. We have observed that the concepts generated by LLMs sometimes include non-visual descriptions, despite the use of appearance-related prompts. Future work will focus on improving concept selection quality and enhancing training efficiency.

7. Acknowledgement

We acknowledge the support from the National Natural Science Foundation of China under Grants 62036012, 62476199, 62202331, U23A20387, U21B2044 and 62376268.

References

[1] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10044–10054, 2020. 2

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018. 2

[3] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018. 2

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014. 5

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. 2

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*. 2

[7] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 2

[8] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021. 1

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019. 2

[11] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10265–10275, 2022. 2

[12] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[13] Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11483–11493, 2023. 2, 6

[14] Alex Gomez-Villa, Bartlomiej Twardowski, Lu Yu, Andrew D. Bagdanov, and Joost van de Weijer. Continually learning self-supervised representations with projected functional regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3867–3877, 2022. 2

[15] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019. 2

[16] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 831–839, 2019. 2

[17] Linlan Huang, Xusheng Cao, Haori Lu, and Xialei Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *European Conference on Computer Vision*, pages 214–231. Springer, 2024. 2

[18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 2

[19] Sangwon Kim, Jaeyeal Nam, and Byoung Chul Ko. Vit-net: Interpretable vision transformers with neural tree decoder. In *International conference on machine learning*, pages 11162–11172. PMLR, 2022. 2

[20] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2

[22] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 1, 2, 3

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[25] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 2

[27] Zhuowei Li, Long Zhao, Zizhao Zhang, Han Zhang, Di Liu, Ting Liu, and Dimitris N Metaxas. Steering prototypes with prompt-tuning for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2523–2533, 2024. 2, 6

[28] RuiQi Liu, Boyu Diao, Libo Huang, Zijia An, Zhulin An, and Yongjun Xu. Continual learning in the frequency domain. In *Advances in Neural Information Processing Systems*, pages 85389–85411, 2024. 2

[29] Xialei Liu, Jiang-Tian Zhai, Andrew D. Bagdanov, Ke Li, and Ming-Ming Cheng. Task-adaptive saliency guidance for exemplar-free class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23954–23963, 2024. 2

[30] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 2

[31] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022. 1

[32] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 1

[33] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020. 2

[34] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14933–14943, 2021. 2

[35] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 5

[36] Tuomas Oikarinen, Subhro Das, Lam Nguyen, and Lily Weng. Label-free concept bottleneck models. In *International Conference on Learning Representations*, 2023. 2, 3, 4, 7, 1

[37] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3911–3920, 2023. 5

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[39] Jathushan Rajasegaran, Munawar Hayat, Salman H Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. *Advances in neural information processing systems*, 32, 2019. 1

[40] Sukrut Sridhar Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *18th European Conference on Computer Vision*. Springer, 2024. 7

[41] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2

[42] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019. 1, 2

[43] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021. 2

[44] Dawid Rymarczyk, Joost van de Weijer, Bartosz Zieliński, and Bartlomiej Twardowski. Icicle: Interpretable class incremental continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1887–1898, 2023. 1, 2, 6

[45] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2

[46] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2591–2600, 2019. 2

[47] Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental residual concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11030–11040, 2024. 2, 3

[48] Ivaxi Sheth and Samira Ebrahimi Kahou. Auxiliary losses for learning generalizable concept-based models. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[49] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023. 2, 6

[50] Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Pilot: A pre-trained model-based continual learning toolbox. *arXiv preprint arXiv:2309.07117*, 2023. 5

[51] Zhe Tao, Lu Yu, Hantao Yao, Shucheng Huang, and Changsheng Xu. Class incremental learning for light-weighted networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 2

[52] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022. 2, 6

[53] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[54] Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 10962–10971, 2023. 1, 3

[55] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648, 2022. 2, 6

[56] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022. 1, 2, 5, 6

[57] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In *International Conference on Machine Learning*, pages 11205–11216. PMLR, 2021. 4, 5

[58] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3090–3100, 2023. 2, 3, 7, 1

[59] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021. 2

[60] Yang Yang, Zhiying Cui, Junjie Xu, Changhong Zhong, Wei-Shi Zheng, and Ruixuan Wang. Continual learning with bayesian model based on a fixed pre-trained feature extractor. *Visual Intelligence*, 1(1):5, 2023. 2

[61] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2023. 2, 3, 1

[62] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*, 33: 20554–20565, 2020. 2

[63] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. 2

[64] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6982–6991, 2020. 1, 5

[65] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022. 2

[66] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 2

[67] Jiang-Tian Zhai, Xialei Liu, Andrew D. Bagdanov, Ke Li, and Ming-Ming Cheng. Masked autoencoders are efficient class incremental learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19104–19113, 2023.

[68] Jiang-Tian Zhai, Xialei Liu, Lu Yu, and Ming-Ming Cheng. Fine-grained knowledge selection and restoration for non-exemplar class incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6971–6978, 2024. 2

[69] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19148–19158, 2023. 2, 5, 6

[70] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23554–23564, 2024. 2, 5, 6

[71] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[72] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021. 2, 5