This ICCV 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.

ICICLE: Interpretable Class Incremental Continual Learning

Dawid Rymarczyk[1,2,3,*] Joost van de Weijer[4,5] Bartosz Zieliński[1,3,6]
Bartłomiej Twardowski[4,5,6]
[1] Faculty of Mathematics and Computer Science, Jagiellonian University
[2] Doctoral School of Exact and Life Sciences, Jagiellonian University [3] Ardigen SA
[4] Autonomous University of Barcelona [5] Computer Vision Center [6] IDEAS NCBR
*dawid.rymarczyk@doctoral.uj.edu.pl

Abstract

Continual learning enables incremental learning of new tasks without forgetting those previously learned, resulting in positive knowledge transfer that can enhance performance on both new and old tasks. However, continual learning poses new challenges for interpretability, as the rationale behind model predictions may change over time, leading to interpretability concept drift. We address this problem by proposing Interpretable Class-InCremental LEarning (ICICLE), an exemplar-free approach that adopts a prototypical part-based approach. It consists of three crucial novelties: interpretability regularization that distills previously learned concepts while preserving user-friendly positive reasoning; proximity-based prototype initialization strategy dedicated to the fine-grained setting; and task-recency bias compensation devoted to prototypical parts. Our experimental results demonstrate that ICICLE reduces the interpretability concept drift and outperforms the existing exemplar-free methods of common class-incremental learning when applied to concept-based models.

1. Introduction

With the growing use of deep learning models in diverse domains, including robotics [10], medical imaging [17], and autonomous driving [43], there is a pressing need to develop models that can adapt to ever-changing conditions and learn new tasks from non-stationary data. However, a significant challenge with neural networks is their tendency to suffer from *catastrophic forgetting* [26, 30, 44], where performance on previous tasks deteriorates rapidly as new ones are acquired. Continual Learning (CL) [19] has emerged as a promising technique to address this challenge by enabling models to learn new tasks without forgetting those learned before.

While existing CL approaches significantly reduce catastrophic forgetting, they are often difficult for humans to understand. It is especially problematic because deep networks often predict the right answer for the wrong reason (the "Clever Hans" phenomenon), leading to excellent performance in training but poor performance in practice [63]. This results in serious societal problems that deeply affect health, freedom, racial bias, and safety [11]. As a result, some initial steps were taken in the literature to introduce explainable posthoc methods into the CL setup [32, 52, 64]. However, explaining black boxes, rather than replacing them with interpretable (self-explainable) models, can escalate the problem by providing misleading or false characterizations [69] or adding unnecessary authority to the black box [12]. Therefore, there is a clear need for innovative machine-learning models that are inherently interpretable [11]. To the best of our knowledge, no interpretable CL approach has been proposed so far.

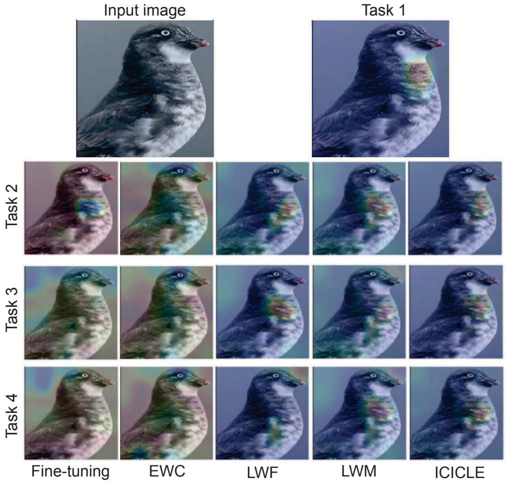

Figure 1: We process the input image (top left) through the network and visualize how its specific areas are similar to one of the prototypes. The interpretability concept drift occurs when such a similarity map differs between tasks. ICICLE performs best, preserving similarity maps better than the other continual learning methods.

In this work, we introduce Interpretable Class-

Incremental Learning (ICICLE), an interpretable approach to class-incremental learning based on prototypical parts methodology. Similarly to *This looks like that* reasoning [16], ICICLE learns a set of prototypical parts representing reference concepts derived from the training data and makes predictions by comparing the input image parts to the learned prototypes. However, the knowledge transfer between tasks in continual learning poses new challenges for interpretability. Mainly because the rationale behind model predictions may change over time, leading to *interpretability concept drift* and making explanations inconsistent (see Figure 1 and Table 1). Therefore, ICICLE contains multiple mechanisms to prevent this drift and, at the same time, obtain satisfactory results. First, we propose an interpretability regularization suited for prototypical part-based models to retain previously gained knowledge while maintaining model plasticity. It ensures that previously learned prototypical parts are similarly activated within the current task data, which makes explanations consistent over time. Moreover, considering the fine-grained nature of considered datasets, we introduce proximity-based prototype initialization for a new task. It searches for representative concepts within the new task data close to previously learned concepts, allowing the model to recognize high-level features of the new task and focusing on tuning details. Thirdly, to overcome task-recency bias in class-incremental learning scenarios, we propose a simple yet effective method that balances the logits of all tasks based on the last task data. Finally, we reduce multi-stage training while preserving user-friendly positive reasoning.

We evaluate ICICLE on two datasets, namely CUB-200-2011 [83] and Stanford Cars [46], and conduct exhaustive ablations to demonstrate the effectiveness of our approach. We show that this problem is challenging but opens up a promising new area of research that can further advance our understanding of CL methods. Our contributions can be summarized as follows:

- We are the first to introduce interpretable class-incremental learning and propose a new method ICICLE, based on prototypical part methodology.
- We propose interpretability regularization that prevents interpretability concept drift without using exemplars.
- We define a dedicated prototype initialization strategy and a method compensating for task-recency bias.

2. Related Works

Continual Learning and Class Incremental Learning Existing continual learning methods can be broadly categorized into three types: replay-based, architecture-based, and regularization-based methods [19, 54]. Replay-based methods either save a small amount of data from previously seen tasks [5, 15] or generate synthetic data with a generative model [85, 92]. The replay data can be used during training together with the current data, such as in iCaRL [67] and LUCIR [37], or to constrain the gradient direction while training, such as in AGEM [14]. Architecture-based methods activate different subsets of network parameters for different tasks by allowing model parameters to grow linearly with the number of tasks. Previous works following this strategy include DER [88], Piggyback [50], PackNet [51]. Regularization-based methods add an additional regularization term derived from knowledge of previous tasks to the training loss. This can be done by either regularizing the weight space, which constrains important parameters [78, 81], or the functional space, which constrains predictions or intermediate features [23, 38]. EWC [44], MAS [3], REWC [49], SI [91], and RWalk [13] constrain the importance of network parameters to prevent forgetting. Methods such as LWF [48], LWM [21], and BiC [87] leverage knowledge distillation to regularize features or predictions. Additionally, more challenging setups are considered in the field such as open-set interpretable continual learning [57]. When it comes to interpretable CL, the generative replay approaches [58] provide a certain degree of latent clarity. However, they require a decoder (for visualization) and may fail to produce realistic prototype images [16]. Class-incremental learning (class-IL) is the most challenging scenario where the classifier learns new classes sequentialy. The model needs to maintain good performance on all classes seen so far [82]. Two types of evaluation methods are defined [54]: task-agnostic (no access to task-ID during inference, e.g., BiC [87]) and task-aware (task-ID is given during inference, e.g., HAT [77]).

Method	IoU			
	task 1	task 2	task 3	mean
Finetuning	0.115	0.149	0.260	0.151
EWC	0.192	0.481	0.467	0.334
LWF	0.221	0.193	0.077	0.188
LWM	0.332	0.312	0.322	0.325
ICICLE	**0.705**	**0.753**	**0.742**	**0.728**

Table 1: Quantitative results for interpretability concept drift presented in Figure 1. We compute IoU between similarities obtained after each task and after incremental tasks. E.g. in column "task 1", we calculate IoU between similarity maps of task one prototypes after each learning episode.

Explainable Artificial Intelligence In the field of deep learning explanations, two types of models have been explored: post hoc and self-explainable models [69]. Post hoc models explain the reasoning process of black-box methods, including saliency maps [53, 66, 75, 76, 79], concept activation vectors [18, 29, 40, 45, 89], counterfactual examples [1, 31, 56, 62, 86], and image perturbation analysis [7, 24, 25, 68]. Self-explainable models, on the other hand, aim to make the decision process more transparent and have attracted significant attention [4, 9]. Recently, researchers have focused on enhancing the concept

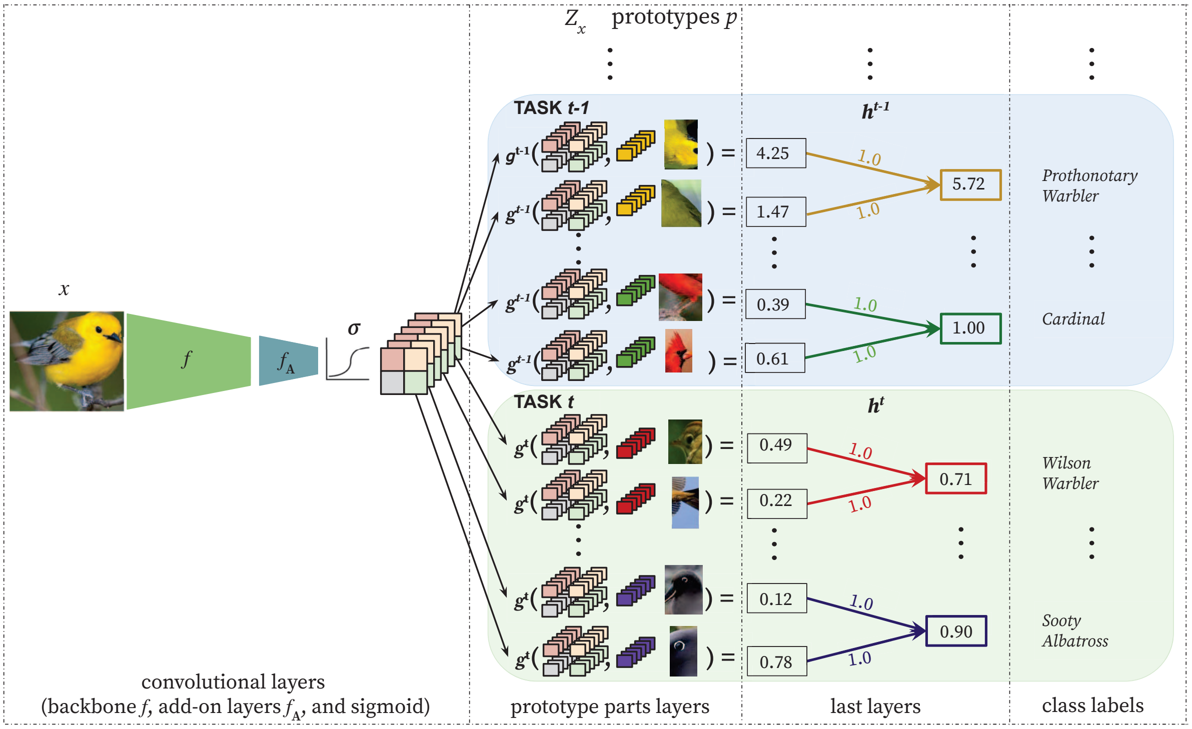

Figure 2: The architecture of our ICICLE with separate prototypical part layers g^t for each task t. In this example, prototypes of classes *Prothonotary Warbler* and *Cardinal* belong to task $t-1$, while prototypes of *Wilson Warbler* and *Sooty Albatross* to task t. Layers g^t are preceded by shared backbone f, add-on f_A, and sigmoid. Moreover, they are followed by the last layers h^t with weight $h^t_{ci} = 1$ if prototype p_i is assigned to class c and equals 0 otherwise.

of prototypical parts introduced in ProtoPNet [16] to represent the activation patterns of networks. Several extensions have been proposed, including TesNet [84] and Deformable ProtoPNet [22], which exploit orthogonality in prototype construction. ProtoPShare [73], ProtoTree [59], and ProtoPool [72] reduce the number of prototypes used in classification. Other methods consider hierarchical classification with prototypes [33], prototypical part transformation [47], and knowledge distillation techniques from prototypes [39]. Prototype-based solutions have been widely adopted in various applications such as medical imaging [2, 6, 41, 71, 80], time-series analysis [28], graph classification [70, 93], sequence learning [55], and semantic segmentation [74]. In this work, we adapt the prototype mechanism to class incremental learning.

3. Methods

The aim of our approach is to increase the interpretability in the class-incremental scenario. For this purpose, we adapt prototypical parts [16], which directly participate in the model computation, making explanations faithful to the classification decision. To make this work self-contained, we first recall the prototypical parts methodology, and then we describe how we adapt them to the class-incremental scenario. We aim to compensate for interpretability concept drift, which we define at the end of this section. As we aim to compensate for interpretability concept drift, we define it at the end of this section.

3.1. Prototypical parts methodology

Architecture. The original implementation of prototypical parts [16] introduces an additional prototypical part layer g proceeded by a backbone convolutional network f with an add-on f_A and followed by the fully connected layer h. The f_A add-on consists of two 1×1 convolutional layers and a sigmoid activation at the end, translating the convolutional output to a prototypical part space. The prototypical part layer g consists of K prototypes $p_i \in \mathbb{R}^D$ per class, and their assignment is handled by the fully connected layer h. If prototype p_i is assigned to class c, then $h_{ci} = 1$, otherwise, it is set to -0.5.

Inference. Given an input image x, the backbone f generates its representation $f(x)$ of shape $H \times W \times D$, where H and W are the height and width of the representation obtained at the last convolutional layer, and D is the number of channels. This representation is translated by f_A to a prototypical part space, again of size $H \times W \times D$. Then, each prototypical part p_i is compared to each of $H \times W$ representation vectors to calculate the maximum similarity (i.e. the maximal activation of this prototype on the analyzed image) $\max_{j \in \{1..HW\}} sim(p_i, z_j)$, where

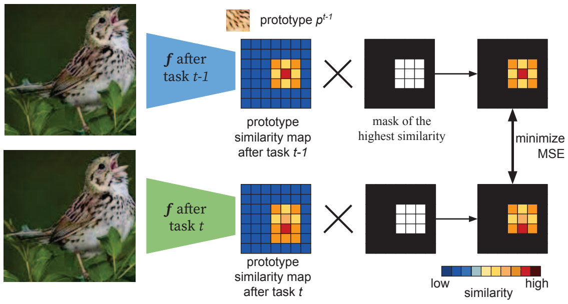
prototype p^{t-1}
f after task *t-1*
prototype similarity map after task *t-1*
mask of the highest similarity
minimize MSE
f after task *t*
prototype similarity map after task *t*
low
similarity
high

Figure 3: Our interpretability regularization aims to minimize the changes in the prototype similarities. It takes a prototype p^{t-1} of previous tasks and an image from task t, selects the image area with the highest similarity to this prototype (binary mask S), and punishes the model for any changes in this area caused by training task t.

$sim(p_i, z_j) = \log \frac{|z_j - p_i|_2 + 1}{|z_j - p_i|_2 + \eta}$ and $\eta \ll 1$. To obtain the final predictions, we push those values through the fully connected (and appropriately initialized) layer h.

Training. Training is divided into three optimization phases: warm-up, joint learning, and convex optimization of the last layer. The first phase trains add-on f_A and the prototypical part layer g. The second phase learns f_A, g, and the backbone network f. The last phase fine-tunes the fully-connected layer h. Training is conducted with the cross-entropy loss supported by two regularizations, cluster and separation costs [16]. Cluster encourages each training image to have a latent patch close to at least one prototype of its class. In contrast, the separation cost encourages every latent patch of a training image to stay away from the prototypes of the remaining classes.

3.2. ICICLE

Significant modifications of architecture and training are required to employ prototypical parts methodology to class-incremental learning (the inference is identical). Mostly because incremental learning has considerably different conjectures. It assumes T tasks $(C^1, X^1), (C^2, X^2), \ldots, (C^T, X^T)$, where each task t contains classes C^t and training set X^t. Moreover, during task t, only X_t training data are available, as we consider the exemplar-free setup, where it is prohibited to save any data from previous tasks (no replay buffer is allowed).

Architecture. As in the baseline model, ICICLE comprises backbone f and add-on f_A. However, it does not use one fixed prototypical part layer g and one fully-connected layer h. Instead, it introduces a prototypical part layer g^t and a fully-connected layer h^t for each successive task. Layers g^t consist of M^t prototypical parts, where $M^t = K \cdot C^t$ and K is the number of prototypes per class. On the other hand, layer h^t has weight $h^t_{ci} = 1$ if prototype p_i is assigned to class c and it is set to 0 otherwise. We eliminated negative weights (-0.5) from the last layer because multi-stage training is not beneficial for a class-incremental scenario (see Figure 6).

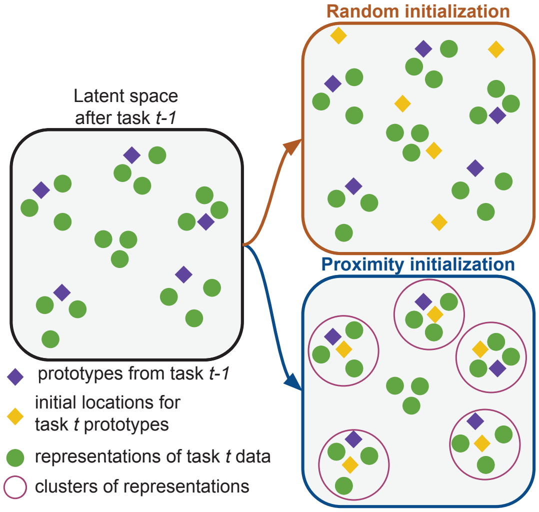
Random initialization
Latent space after task *t-1*
Proximity initialization
prototypes from task *t-1*
initial locations for task *t* prototypes
representations of task *t* data
clusters of representations

Figure 4: We introduce a new proximity-based prototype initialization. It starts by passing training samples of task t through the network (green dots) and choosing representations closest to existing prototypes (violet diamonds). This results in many points, which we cluster (purple circles) to obtain the initial locations of task t prototypical parts (yellow diamonds). Such initialization (bottom right) is preferred over random initialization (top right), where new prototypes can be created far from the old ones, even though they are only slightly different.

Training. To prevent catastrophic forgetting, ICICLE modifies the loss function of the baseline solution. Additionally, it introduces three mechanisms: interpretability regularization, proximity-based prototype initialization, and task-recency bias compensation. Regarding the baseline loss function, the cross-entropy is calculated on the full output of the model, including logits from classes learned in previous tasks. However, the cluster and separation costs are only calculated within the g^t head.

Interpretability regularization. Knowledge distillation [35] is one of the strong regularization methods applied to prevent forgetting [48]. However, the results obtained by its straightforward application are not satisfactory and lead to significant interpretation drift (see Figure 1 and Section 5). Therefore, we introduce an additional regularization cost L_{IR} (see Figure 3), inspired by [39], that minimizes the changes in the similarities for the prototypical parts of the previous tasks. It is defined as:

$$L_{IR} = \sum_{i=0}^{H} \sum_{j=0}^{W} |sim(p^{t-1}, z^t_{i,j}) - sim(p^t, z^t_{i,j})| \cdot S_{i,j} \quad (1)$$

where $sim(p^{t-1}, z^t_{i,j})$ is computed for the model stored before training task t, and S is a binary mask of size $H \times W$, indicating the representation pixels with the highest similarity (γ quantile of those pixels). Such similarity distillation gives higher plasticity when learning a new task but, at the same time, reduces the interpretability drift.

Proximity-based prototype initialization. Random initialization of prototypes, proposed in [16], fails in the incremental learning (see Table 5). Most probably because new prototypes can be created far from the old ones, which are only slightly different (e.g. wing prototypes of various bird species). Therefore, we introduce initialization that sets new prototypes close to the old ones (see Figure 4). We start by passing training samples of task t through the network and determining which patches are the most similar to existing prototypes (we choose patches from the α quantile). More specifically, we compute set $\{z_j^t : \max_i sim(p_i^{t-1}, z_j^t) \in \alpha \text{ quantile}\}$. This results in many candidates for new task prototypical parts. To obtain $K \cdot C^t$ prototypes, we perform KMeans++ clustering, and the resulting centers of clusters are used to initialize the prototypical parts in g^t.

Task-recency bias compensation. When the model learns task t, the similarities to the prototypes of previous tasks drop, mostly due to the changes in the backbone (see Figure 5). That is why, after training the final task, we compensate for this using $T-1$ constants obtained using the last tasks data. More precisely, for each of the previous tasks $t < T$, we take logits $y^t = h^t \circ g^t \circ f(x)$ obtained for all $x \in X^T$ and calculate bias c^t so that $|\{x \in X^T : \max(y^t + c^t) > \max y^T\}| = u|X^T|$. Intuitively, we adjust c^t so the model changes $u\%$ of its prediction from task T to task t. We determined experimentally that $u = 10\%$ is optimal.

3.3. Interpretability Concept Drift

As noted in the caption of Figure 1, the interpretability concept drift occurs when a similarity map differs between tasks. Therefore, it can be formally defined as:

$$ICD = \mathbb{E}_{i,j=1}^{H,W} \left| sim(p^{t-1}, z_{i,j}^t) - sim(p^t, z_{i,j}^t) \right|, \quad (2)$$

where $(z_{i,j})_{i,j=1}^{H,W}$ corresponds to input image representation, p^{t-1} and p^t correspond to prototypical part p before and after task t, and sim is similarity defined in Section 3.1 of the paper. Thus, the greater ICD, the greater the interpretability concept drift.

4. Experimental Setup

We evaluate our approach on the CUB-200-2011 [83] and Stanford Cars [46] datasets to classify 200 bird species and 196 car models, respectively. We consider 4, 10, and 20 task learning scenarios for birds and 4, 7, and 14 options for cars. As the backbone f, we take ResNet-34 [34] without the last layer and pre-trained on ImageNet [20]. We set the number of prototypes per class to 10. Moreover, we use prototypical parts of size $1 \times 1 \times 256$ and $1 \times 1 \times 128$ for birds and cars, respectively. The weights of CE, cluster, separation, and distillation costs in the loss function equal 1.0, 0.8, -0.08, and 0.01. In distillation, we take $\lambda = 1/49$ representation pixels with the highest similarity. For proximity-based initialization, we use $\alpha = 0.5$. For task-recency bias compensation, we take c_t, which changes the predictions of the last validation set by less than 10%. As the implementation framework, we use FACIL [54] based on the PyTorch library[1]. Details on the experimental setup are provided in the Supplementary Materials[2].

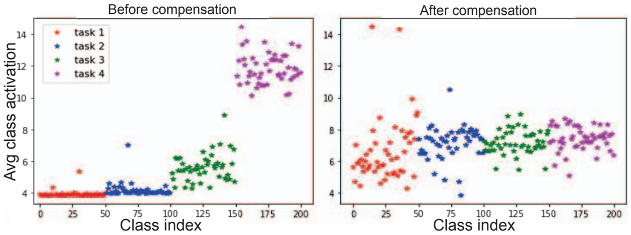

Figure 5: When the model learns task t, the similarities to the prototypes of previous tasks drop and are significantly lower than those of new tasks (upper plot). That is why, after training the final task, we compensate it with $T-1$ calculated constants. As a result, the similarities obtained by prototypes of all tasks are roughly equalized.

5. Results

Performance. We evaluated the effectiveness of ICICLE by comparing it with commonly used exemplar-free baseline methods in class-incremental learning, including LWF [48], LWM [21], and EWC[44][3]. Additionally, Fine-tuning, and Freezing of the feature extractor (not trained at all) are provided. We also report multitask learning as an upper-bound where the various tasks are learned jointly in a multitask manner. To do so, we analyzed task-aware and task-agnostic accuracy for each task after the last one (Table 3) and the aggregated incremental average accuracies after learning the last task in scenarios involving 4, 10, and 20 tasks for CUB (Table 2) and 4, 7, and 14 tasks for Stanford Cars (Supplementary Materials). All methods use the same feature extractor network architectures and ProtoPNet for prototypical part-based learning. Our method outperformed the baseline methods in all cases, indicating its superior performance for prototypical part-based learning in a continual manner. ICICLE retains knowledge from previous tasks better, which results in a more balanced accuracy between tasks and higher accuracy for the first task compared to all other approaches. However, despite the significant improvement, our approach still has room for improvement compared to the upper-bound of multi-task training. With a larger number of tasks, the forgetting of the model is

[1] https://pytorch.org

[2] Code available at: https://github.com/gmum/ICICLE

[3] Extending interpretable models with more complicated exemplar-free methods is not straightforward, and we, therefore, excluded methods such as SDC [90] (which requires learning with a metric loss) and PASS [94] (which requires a combination with self-supervised learning).

Method	Avg. inc. task-aware accuracy			Avg. inc. task-agnostic accuracy		
	4 tasks	10 tasks	20 tasks	4 tasks	10 tasks	20 tasks
Freezing	0.560 ± 0.027	0.531 ± 0.042	0.452 ± 0.055	0.309 ± 0.024	0.115 ± 0.028	0.078 ± 0.004
Finetuning	0.229 ± 0.005	0.129 ± 0.017	0.147 ± 0.021	0.177 ± 0.006	0.072 ± 0.008	0.044 ± 0.006
EWC	0.445 ± 0.012	0.288 ± 0.034	0.188 ± 0.031	0.213 ± 0.008	0.095 ± 0.007	0.046 ± 0.011
LWM	0.452 ± 0.023	0.294 ± 0.032	0.226 ± 0.025	0.180 ± 0.028	0.090 ± 0.011	0.044 ± 0.008
LWF	0.301 ± 0.048	0.175 ± 0.028	0.129 ± 0.023	0.219 ± 0.019	0.078 ± 0.008	0.072 ± 0.008
ICICLE	**0.654 ± 0.011**	**0.602 ± 0.035**	**0.497 ± 0.099**	**0.350 ± 0.053**	**0.185 ± 0.005**	**0.099 ± 0.003**
Multi-task	0.858 ± 0.005	0.905 ± 0.012	0.935 ± 0.019	0.499 ± 0.009	0.196 ± 0.017	0.148 ± 0.009
FeTrIL [65]	0.750 ± 0.008	0.607 ± 0.018	0.407 ± 0.051	0.375 ± 0.006	0.199 ± 0.003	0.127 ± 0.011
PASS [94]	0.775 ± 0.006	0.647 ± 0.003	0.518 ± 0.012	0.395 ± 0.001	0.233 ± 0.009	0.139 ± 0.017

Table 2: Average incremental accuracy comparison for different numbers of tasks on CUB-200-2011, demonstrating the negative impact of the high number of tasks to be learned on models' performance. Despite this trend, ICICLE outperforms the baseline methods across all task numbers. Additionally, we show the gap between interpretable and black-box models by comparing ICICLE to FeTrIL and PASS.

Method	Task-aware Accuracy				Task-agnostic Accuracy			
	Task 1	Task 2	Task 3	Task 4	Task 1	Task 2	Task 3	Task 4
Freezing	**0.806 ± 0.024**	0.462 ± 0.037	0.517 ± 0.041	0.455 ± 0.027	**0.570 ± 0.031**	0.195 ± 0.017	0.258 ± 0.019	0.213 ± 0.020
Finetuning	0.007 ± 0.004	0.016 ± 0.008	0.032 ± 0.009	0.759 ± 0.019	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.759 ± 0.019
EWC	0.244 ± 0.024	0.378 ± 0.072	0.539 ± 0.043	0.602 ± 0.054	0.001 ± 0.001	0.059 ± 0.004	0.267 ± 0.031	0.527 ± 0.051
LWF	0.169 ± 0.046	0.119 ± 0.008	0.235 ± 0.017	0.743 ± 0.061	0.158 ± 0.035	0.003 ± 0.002	0.018 ± 0.003	0.537 ± 0.142
LWM	0.195 ± 0.012	0.412 ± 0.014	0.430 ± 0.028	**0.772 ± 0.011**	0.027 ± 0.024	0.023 ± 0.020	0.085 ± 0.005	**0.772 ± 0.037**
ICICLE	0.523 ± 0.020	**0.663 ± 0.053**	**0.709 ± 0.038**	0.723 ± 0.002	0.233 ± 0.014	**0.365 ± 0.021**	**0.314 ± 0.011**	0.486 ± 0.021

Table 3: Comparison of task accuracies for modified ProtoPNet architecture in a class-incremental learning scenario after 4 tasks train on CUB-200-2011 dataset, averaged over 3 runs with standard error of the mean. Our ICICLE outperforms baseline methods and achieves the best results for all previous incremental tasks, demonstrating its ability to maintain prior knowledge while learning new tasks. Freezing due to the weight fixation cannot properly learn new tasks.

higher, resulting in poorer results, which may be attributed to the number of details that prototypes need to capture to classify a task correctly. Furthermore, we have noticed that freezing is a robust baseline for a task-aware scenario because of the model's fixed nature and pretrained backbone.

Interpretability To evaluate if the prototype's graphical representation of the concept has changed and how much we use the IoU metric [74]. IoU measures what is the overlap of the prototype visual representation (like in Figure 1) from the task in which it was learned, through all the following tasks. Freezing is superior in preserving the prototypical information because all weights from previous tasks are fixed. In terms of methods allowing changes in backbone and previously learned prototypes, ICICLE is superior over all baselines, as shown in Table 1. ICICLE keeps interpretable prototypes consistent with interpretability regularization distilling already learned concepts.

5.1. Ablation study and analysis

Why changes in ProtoPNet architecture and training are needed? ProtoPNet in the last training stage (last layer convex optimizations) aims to finetune positive connections and regularize the negative to be 0. As a result, the converged model returns interpretations in the form of positive reasoning, desired by the end users [11]. In the CL setting, the last step of training changes the negative connections in a different manner (see Figure 6). On the other hand, in an exemplar-free continual learning scenario, conducting the last-layer learning phase is unfeasible at the end of the training. That is why, we modified the ProtoPNet's last layer and retained only positive connections initialized to 1, eliminating the need for the convex optimization step.

Regularization	Initialization	Compensation	TAw acc.	TAg acc.
			0.216	0.182
✓			0.559	0.280
✓	✓		**0.654**	0.335
✓	✓	✓	**0.654**	**0.350**

Table 4: Influence of different novel components on the average incremental accuracy in four tasks learning scenario. Combination of all components results in the best-performing model.

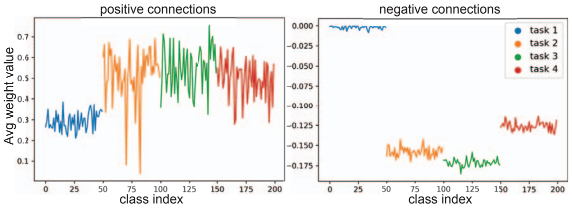

Figure 6: Average weight of positive and negative connections per class in 4 task learning scenario. Unbalanced and strong negative connections between tasks result in undesired properties in terms of the model's interpretability.

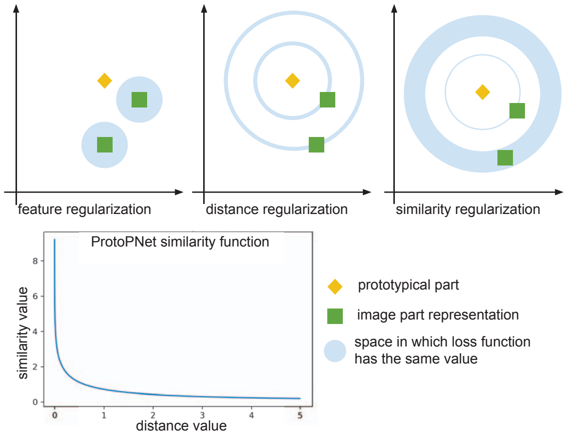
feature regularization
distance regularization
similarity regularization
ProtoPNet similarity function
similarity value
distance value
0
1
2
3
4
5
0
2
4
6
8
prototypical part
image part representation
space in which loss function has the same value

Figure 7: Visualization of three possible approaches to interpretability similarity and their influence on the model plasticity. Only similarity-based regularization takes into account how a given image part corresponds to a prototypical part. If it is close then the similarity value is high and small changes in the distance results in a great decrease in similarity. While latent vectors that are distant from prototypical parts can more freely be changed by the model to better represents the current task data. Other approaches are limiting the models' plasticity treating each latent representation of the image part as equally important.

What is the influence of each of the introduced components? Table 4 presents the influence of different components of our approach on the final model's average incremental accuracy in the CUB-200-2011 dataset with four tasks split scenario. Combining all the components resulted in the best-performing model. Our results show that compensation of task-recency bias helps in task-agnostic evaluation and gives additional improvement of 4.5%. However, most of the accuracy improvements were attributed to interpretability regularization and proximity initialization. Notably, task-recency bias compensation significantly improved the performance of task one classes compared to an approach without it, from 0.028 to 0.255 in a task-agnostic scenario, as detailed in the Supplementary Materials.

Where should we perform interpretability regularization? The ProtoPNet model's prototypical layer can be regularized in three different ways: feature regularization on add-on layer representations, regularization of distances between prototypical parts and latent data vectors, and similarity-based regularization. The strictest approach is feature regularization, which does not allow the model to change how it represents data from a new task, resulting in significantly reduced model plasticity. When distances are regularized, the model can change its representation to maintain the same distance from the prototype on the surface of the sphere. On the other hand, similarity-based regularization allows the model to retain key knowledge from previous tasks by preserving only the information related to specific features that are close to the prototypical parts in the latent space, allowing for greater flexibility in exchange for forgetting irrelevant features. Therefore, we stick to interpretability regularization in ICICLE, which is based on similarities and maintains the essential knowledge from previous tasks while retaining high plasticity to learn new ones. Figure 7 illustrates these three approaches and their comparison in terms of average incremental accuracy for ProtoPNet only with regularization (without changing initialization): 0.507, 0.535, and 0.559 in task-aware and 0.261, 0.230, and 0.280 in task-agnostic scenarios for feature, distance, and similarity-based regularization, respectively, on the CUB-200-2011 dataset with four tasks scenarios.

Initialization type	Random	Distant	All	Proximity
Task aware acc.	0.559	0.592	0.626	**0.654**
Task agnostic acc.	0.280	0.290	0.297	**0.335**

Table 5: Comparison of different initialization strategies for prototypical parts. Our proximity initialization of new task prototypes is superior.

What is the influence of hyperparameters in interpretability regularization? In Figure 8 and Figure 9 the influence of λ and mask percentile threshold in the interpretability regularization on average incremental accuracies are presented. We use CUB-200-2011 datasets with four tasks split setting. For this dataset, the results reveal that the regularization of only the maximum prototypical similarity is the most effective (Figure 9). Regarding λ_{IR}, a value that is too small leads to high network plasticity, increased forgetting, and poor results, while a value that is too large reduces model plasticity and may not represent new knowledge well.

Which way is the best to initialize new prototypical parts? In this ablation part, we investigate the optimal strategy for initializing prototypical parts at the beginning of a new task in the ProtoPNet model. We evaluate our initialization method, which initializes the parts in close proximity to existing prototypes, against three other approaches: random initialization, clustering of all image part representations, and clustering of only distant latent vectors. Results are presented in Table 5. The proximity initialization method outperforms the distant strategy, as the latter tends to assign prototypical parts to latent vectors that correspond to the background of the images, resulting in learning irrelevant concepts that can easily activate on other task data, as shown in the Supplementary Materials.

Does ICICLE generalize to other architectures? Lastly, we show that ICICLE generalizes to other concept-based architecture. We demonstrate that using a TesNet model [84], and provide results in Table 6, where ICICLE obtains the best results. The average incremental accuracy of ICICLE with TesNet is even better than ProtoPNet for both task-aware and task-agnostic evaluation.

	Freezing	Finetuning	EWC	LWM	LWF	ICICLE
TAw Acc.	0.637	0.355	0.592	0.648	0.581	**0.746**
TAg Acc.	0.222	0.183	0.272	0.252	0.205	**0.362**

Table 6: Results for four task learning scenario on CUB-200-2011 dataset with TesNet [84] as a concept-based architecture. The table shows the versatility of the ICICLE approach for interpretable models.

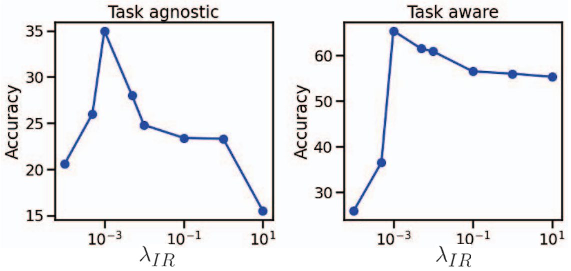

Figure 8: Influence of the λ_{IR} in the interpretability regularization.

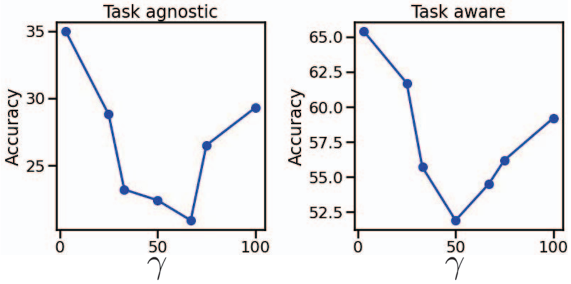

Figure 9: Influence of γ in the interpretability regularization. Notice that regularizing only in the place of maximum similarity is the most beneficial for ICICLE for the four task learning scenario in CUB-200-2011.

6. Conclusions and future work

This work proposes a novel approach called ICICLE for interpretable class incremental learning. ICICLE is based on prototypical parts and incorporates interpretability regularization, proximity initialization, and compensation for task-recency bias. The proposed method outperforms classical class-incremental learning methods applied for prototypical part-based networks in terms of task-aware and task-agnostic accuracies while maintaining prototype interpretability. We also conducted ablation studies and multiple analyses to justify our choices and highlight the challenges associated with combining interpretable concepts with CL. This work is expected to inspire research on XAI and CL.

Moving forward, we plan to explore methods suitable for single-class incremental learning with interpretable models. We also intend to investigate how other interpretable architectures, such as B-COS [8], can be adapted to the class incremental learning scenario.

Limitations. Our work is limited to prototypical part methods, that are suited for fine-grained image recognition and inherits their drawbacks previously discussed in [27, 36, 42, 60, 72].However, recently there are first works generalizing them to standard datasets (not fine-grained) [61]. Additionally, as we consider only an exemplar-free scenario and closed-set recognition, we do not analyze how having a replay buffer would influence the method's performance and how this method would fit in the open-set settings.

Impact. ICICLE highlights that traditional exemplar-free approaches for continual learning are not well suited for gray-box models that utilize concepts for predictions. This finding has implications for the development of continual learning methods, as they must balance the need for generality with the need to be adapted to specific architectures. Furthermore, it has an impact on the field of concept-based models and explainable AI, demonstrating the need for further research on CL methods for XAI. In some cases, practitioners who know that their system will need to learn new tasks continuously may choose to use black-box models and explainers rather than interpretable models, sacrificing the fidelity of explanations for improved model performance.

Acknowledgements

Joost van de Weijer and Bartłomiej Twardowski acknowledge the support from the Spanish Government funding for projects PID2019-104174GB-I00, TED2021-132513B-I00, and grant RYC2021-032765-I, the European Commission under the Horizon 2020 Programme, funded by MCIN/AEI/10.13039/501100011033 and by European Union NextGenerationEU/PRTR. The work of Dawid Rymarczyk was supported by the National Science Centre (Poland), grant no. 2022/45/N/ST6/04147. He also received an incentive scholarship from the program Excellence Initiative - Research University funds at the Jagiellonian University in Kraków. The work of Bartosz Zieliński was funded by the National Science Centre (Poland) grant no. 2022/47/B/ST6/03397 and the research project "Bio-inspired artificial neural network" (grant no. POIR.04.04.00-00-14DE/18-00) within the Team-Net program of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund. Finally, some experiments were performed on servers purchased with funds from a Priority Research Area (Artificial Intelligence Computing Center Core Facility) grant under the Strategic Programme Excellence Initiative at Jagiellonian University.

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 Con-*

ference on Computer Vision and Pattern Recognition, pages 10044–10054, 2020. 2

[2] Michael Anis Mihdi Afnan, Yanhe Liu, Vincent Conitzer, Cynthia Rudin, Abhishek Mishra, Julian Savulescu, and Masoud Afnan. Interpretable, not black-box, artificial intelligence should be used for embryo selection. *Human Reproduction Open*, 2021. 3

[3] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision*, 2018. 2

[4] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[5] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227, 2021. 2

[6] Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y Lo, and Cynthia Rudin. A case-based interpretable deep learning model for classification of mass lesions in digital mammography. *Nature Machine Intelligence*, 3(12):1061–1070, 2021. 3

[7] Dominika Basaj, Witold Oleszkiewicz, Igor Sieradzki, Michał Górszczak, B Rychalska, T Trzcinski, and B Zielinski. Explaining self-supervised image representations with visual probing. In *International Joint Conference on Artificial Intelligence*, 2021. 2

[8] Moritz Böhle, Mario Fritz, and Bernt Schiele. B-cos networks: alignment is all we need for interpretability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10329–10338, 2022. 8

[9] Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019. 2

[10] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022. 1

[11] Rudin C. et al. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022. 1, 6

[12] Rudin C. and Radin J. Why are we using black box models in ai when we don't need to? a lesson from an explainable ai competition. *Harvard Data Science Review*, 1(2):10–1162, 2019. 1

[13] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: understanding forgetting and intransigence. In *European Conference on Computer Vision*, 2018. 2

[14] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2019. 2

[15] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 2

[16] 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, 3, 4, 5

[17] Xuxin Chen, Ximin Wang, Ke Zhang, Kar-Ming Fung, Theresa C Thai, Kathleen Moore, Robert S Mannel, Hong Liu, Bin Zheng, and Yuchen Qiu. Recent advances and clinical applications of deep learning in medical image analysis. *Medical Image Analysis*, page 102444, 2022. 1

[18] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. 2

[19] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 2

[20] 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

[21] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5

[22] 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. 3

[23] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020. 2

[24] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2950–2958, 2019. 2

[25] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017. 2

[26] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in cog. scie.*, 1999. 1

[27] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like

that: Enhancing self-explaining models by prototypical relevance propagation. *Pattern Recognition*, 136:109172, 2023. 8

[28] Alan H Gee, Diego Garcia-Olano, Joydeep Ghosh, and David Paydarfar. Explaining deep classification of time-series data with learned prototypes. In *CEUR workshop proceedings*, volume 2429, page 15. NIH Public Access, 2019. 3

[29] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[30] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *International Conference on Learning Representations*, 2014. 1

[31] 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

[32] Filip Guzy, Michał Woźniak, and Bartosz Krawczyk. Evaluating and explaining generative adversarial networks for continual learning under concept drift. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 295–303. IEEE, 2021. 1

[33] Peter Hase, Chaofan Chen, Oscar Li, and Cynthia Rudin. Interpretable image recognition with hierarchical prototypes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 32–40, 2019. 3

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[35] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*, 2014. 4

[36] Adrian Hoffmann, Claudio Fanconi, Rahul Rade, and Jonas Kohler. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks, 2021. 8

[37] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *International Conference on Computer Vision*, 2019. 2

[38] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3957–3966, 2021. 2

[39] Monish Keswani, Sriranjani Ramakrishnan, Nishant Reddy, and Vineeth N Balasubramanian. Proto2proto: Can you recognize the car, the way i do? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10233–10243, 2022. 3, 4

[40] 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

[41] Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. Xprotonet: Diagnosis in chest radiography with global and local explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15719–15728, 2021. 3

[42] Sunnie SY Kim, Nicole Meister, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: evaluating the human interpretability of visual explanations. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 280–298. Springer, 2022. 8

[43] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021. 1

[44] 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. *National Academy of Sciences*, 2017. 1, 2, 5

[45] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020. 2

[46] 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. 2, 5

[47] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3

[48] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2, 4, 5

[49] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *International Conference on Pattern Recognition*, 2018. 2

[50] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *European Conference on Computer Vision*, 2018. 2

[51] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[52] Emanuele Marconato, Gianpaolo Bontempo, Stefano Teso, Elisa Ficarra, Simone Calderara, and Andrea Passerini. Catastrophic forgetting in continual concept bottleneck models. In *Image Analysis and Processing. ICIAP 2022 Workshops: ICIAP International Workshops, Lecce, Italy, May 23–27, 2022, Revised Selected Papers, Part II*, pages 539–547. Springer, 2022. 1

[53] Diego Marcos, Sylvain Lobry, and Devis Tuia. Semantically interpretable activation maps: what-where-how explanations within cnns. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4207–4215. IEEE, 2019. 2

[54] Marc Masana, Xialei Liu, Bartlomiej 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*, pages 1–20, 2022. 2, 5

[55] Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 903–913, 2019. 3

[56] 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

[57] Martin Mundt, Yongwon Hong, Iuliia Pliushch, and Visvanathan Ramesh. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. *Neural Networks*, 160:306–336, 2023. 2

[58] Martin Mundt, Iuliia Pliushch, Sagnik Majumder, Yongwon Hong, and Visvanathan Ramesh. Unified probabilistic deep continual learning through generative replay and open set recognition. *Journal of Imaging*, 8(4):93, 2022. 2

[59] Meike Nauta et al. 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. 3

[60] Meike Nauta, Annemarie Jutte, Jesper Provoost, and Christin Seifert. This looks like that, because... explaining prototypes for interpretable image recognition. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2021, Virtual Event, September 13-17, 2021, Proceedings, Part I*, pages 441–456. Springer, 2022. 8

[61] Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2744–2753, 2023. 8

[62] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12700–12710, 2021. 2

[63] Schramowski P. et al. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020. 1

[64] Arijit Patra and J Alison Noble. Incremental learning of fetal heart anatomies using interpretable saliency maps. In *Medical Image Understanding and Analysis: 23rd Conference, MIUA 2019, Liverpool, UK, July 24–26, 2019, Proceedings 23*, pages 129–141. Springer, 2020. 1

[65] 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. 6

[66] Sylvestre-Alvise Rebuffi, Ruth Fong, Xu Ji, and Andrea Vedaldi. There and back again: Revisiting backpropagation saliency methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8848, 2020. 2

[67] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2

[68] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 2

[69] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 1, 2

[70] Dawid Rymarczyk, Daniel Dobrowolski, and Tomasz Danel. Progrest: Prototypical graph regression soft trees for molecular property prediction. *SIAM International Conference on Data Mining*, 2023. 3

[71] Dawid Rymarczyk, Aneta Kaczyńska, Jarosław Kraus, Adam Pardyl, and Bartosz Zieliński. Protomil: Multiple instance learning with prototypical parts for fine-grained interpretability. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2022. 3

[72] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3, 8

[73] 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 International Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021. 3

[74] Mikołaj Sacha, Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protoseg: Interpretable semantic segmentation with prototypical parts. In *Winter Conference on Applications of Computer Vision (WACV)*, 2023. 3, 6

[75] 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

[76] 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

[77] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, 2018. 2

[78] Yujun Shi, Li Yuan, Yunpeng Chen, and Jiashi Feng. Continual learning via bit-level information preserving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16674–16683, 2021. 2

[79] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014. 2

[80] Gurmail Singh and Kin-Choong Yow. These do not look like those: An interpretable deep learning model for image recognition. *IEEE Access*, 9:41482–41493, 2021. 3

[81] Shixiang Tang, Dapeng Chen, Jinguo Zhu, Shijie Yu, and Wanli Ouyang. Layerwise optimization by gradient decomposition for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9634–9643, 2021. 2

[82] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. In *NeurIPS Continual Learning Workshop*, 2018. 2

[83] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 5

[84] Jiaqi Wang et al. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 895–904, 2021. 3, 7, 8

[85] Liyuan Wang, Kuo Yang, Chongxuan Li, Lanqing Hong, Zhenguo Li, and Jun Zhu. Ordisco: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5383–5392, 2021. 2

[86] Pei Wang and Nuno Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8981–8990, 2020. 2

[87] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[88] 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

[89] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20554–20565. Curran Associates, Inc., 2020. 2

[90] 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. 5

[91] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017. 2

[92] Mengyao Zhai, Lei Chen, and Greg Mori. Hyper-lifelonggan: Scalable lifelong learning for image conditioned generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2246–2255, 2021. 2

[93] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. Protgnn: Towards self-explaining graph neural networks. 2022. 3

[94] 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. 5, 6