

How green is continual learning, really? Analyzing the energy consumption in continual training of vision foundation models

Tomaso Trinci[✉], Simone Magistri[✉], Roberto Verdecchia[✉], and Andrew D.
Bagdanov[✉]

Department of Information Engineering, University of Florence, Italy
name.surname@unifi.it

Abstract. With the ever-growing adoption of AI, its impact on the environment is no longer negligible. Despite the potential that continual learning could have towards Green AI, its environmental sustainability remains relatively uncharted. In this work we aim to gain a systematic understanding of the energy efficiency of continual learning algorithms. To that end, we conducted an extensive set of empirical experiments comparing the energy consumption of recent representation-, prompt-, and exemplar-based continual learning algorithms and two standard baseline (fine tuning and joint training) when used to continually adapt a pre-trained ViT-B/16 foundation model. We performed our experiments on three standard datasets: CIFAR-100, ImageNet-R, and DomainNet. Additionally, we propose a novel metric, the *Energy NetScore*, which we use measure the algorithm efficiency in terms of energy-accuracy trade-off. Through numerous evaluations varying the number and size of the incremental learning steps, our experiments demonstrate that different types of continual learning algorithms have very different impacts on energy consumption during both training and inference. Although often overlooked in the continual learning literature, we found that the energy consumed during the inference phase is crucial for evaluating the environmental sustainability of continual learning models.

Keywords: Green AI · Continual Learning · Foundation Models

1 Introduction

The widespread adoption of Artificial Intelligence for real-world applications has been driven by the ability of deep learning to solve increasingly complex problems in various fields, such as computer vision and natural language processing [29]. The growing demand for higher-performing models has led to the development of large models pre-trained on massive datasets, such as Llama [46] and CLIP [12], whose training requires intensive hardware resources. Such models, based on Language and Vision Transformer architectures [12, 47] and commonly referred to as *foundation models* [1], can reach hundreds of billions of parameters and

perform trillions of operations. This scale has prompted *Green AI* researchers to address the environmental concerns associated with them [32].

Current best practices involve adapting foundation models through transfer learning, or *fine-tuning*, to solve specific tasks. However, the standard learning paradigm of these approaches is largely *static*. When models must be updated to learn new tasks or to improve their performance with additional data, the preferred solution is *joint-training* on both new and old data. This may offer optimal performance, but demands increasing computational resources as the number of tasks or the volume of data grows. Conversely, sequentially fine-tuning on novel tasks requires fewer training resources but can lead to a rapid decrease in performance on previous tasks, a phenomenon known as *catastrophic forgetting* [10,37]

Continual Learning (CL) aims at enabling deep learning models to continuously learn from new data while mitigating catastrophic forgetting [51]. This enables model update even when old data disappear due to *privacy concerns* and requires *fewer training computational resources*, reducing environmental impact and resource costs [30,31]. These benefits make CL appealing for applications where privacy and efficiency are major concerns [24,33,34,42]. The development of efficient training strategies has been the primary motivation for CL approaches aimed at mitigating forgetting in large, hardware-intensive pre-trained foundation models [38,53,57]. Surprisingly, despite the efficiency promises of CL, and the non-negligible carbon footprint of foundation model training, a systematic empirical evaluation of the sustainability of CL appears to be missing in the literature. Specifically, there is a lack of analysis on the additional training complexities introduced to address catastrophic forgetting and how current CL methods compare to joint training in terms of energy usage. Additionally, the complexity these CL methods add to the models inference phase has been largely overlooked in the CL research community.

In this paper we present an extensive empirical investigation in the energy consumption of CL with pre-trained models, focusing on vision foundation models due to its prominence in CL literature. Our study aims to determine whether CL methods are energy efficient during training and inference phases, and hence whether (and to what degree) they render CL sustainable in real-world applications. Specifically, we measure the energy costs associated with CL techniques applied to pre-trained vision models and compare them to the energy required to completely retrain a model for each new task (see Figure 1). Moreover, we propose a new metric called the *Energy NetScore* that measures overall algorithm efficiency in terms of energy-accuracy trade-off. By exploring the intersection of vision foundation models, continual learning, and Green AI, we aim to address a gap in current research and to evaluate how effective CL is at saving energy.

This research empirically answers key research questions regarding the energy consumption of CL applied to pre-trained vision foundation models:

- *How do CL approaches compare in training energy consumption?*
- *How do training energy costs scale with data and model updates?*
- *How do CL approaches compare in Energy NetScore?*
- *How does the CL strategy impact inference energy consumption?*

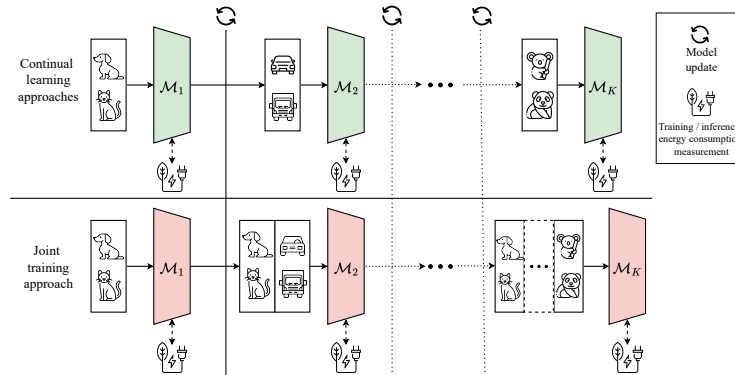


Fig. 1: In continual learning a model \mathcal{M}_k (in green) learns and adapts using only current training data without forgetting previous information. Conversely, the joint incremental training strategy (in red) uses both previous and current data, leading to comprehensive learning but higher computational and storage costs. In this work we aim to understand the impact on the energy consumption of model \mathcal{M}_k when trained following different CL approaches and how they compare to joint incremental training.

2 Related Work

In recent years research into the environmental impact of AI systems has gained increasing momentum [50]. Among the many works on this topic, a considerable portion of Green AI studies propose solutions to optimize the environmental impact of AI through different strategies. Green AI solutions often consider hyperparameter tuning [7], deployment [44], data-centric [49], and trade-off between precision and energy consumption strategies [55].

A smaller set of Green AI works instead present observational studies, i.e. inquiries aimed to solely assess, rather than optimize, the environmental sustainability of AI. Examples of observational studies include the comparison of different deep learning frameworks in terms of energy efficiency [14], understanding the impact of hyperparameter tuning on power consumption [7], and monitoring carbon intensity of AI algorithms in cloud environments [11]. Our study falls in the Green AI observational study category, as it presents an empirical evaluation of the energy efficiency of continual training of vision foundation models.

In terms of architectures considered in Green AI studies, we note that neural networks are by far most investigated [50]. Studies falling under this category cover a vast and heterogeneous spectrum of topics, ranging from the impact of precision quantization on energy efficiency [19], power capping techniques [28] and deployment strategies in multi-GPU environments [5] to energy consumption prediction approaches [41]. In such a variegated set of arguments connected to the environmental sustainability of neural networks, our study positions itself by exploring a niche that has to date remained largely unexplored, namely Green AI in Continual Learning, as further discussed below.

To the best of our knowledge, only a single study takes into account the environmental sustainability of CL. In their work, Chavan et al. [6] present an observational study assessing the energy efficiency of four CL algorithms in an industrial setting. They argue that in such scenarios it preferable to alternate CL techniques with complete retraining when certain thresholds of acceptability are reached. In contrast, we are interested in comparing different continual learning methods and understanding how various solutions to mitigate forgetting impact energy consumption, both during training and inference phases when adapting vision foundation models to a sequence of incremental tasks.

We believe that our study presents the most extensive evaluation to date of continual learning algorithms in terms of environmental sustainability. In the next section we describe the specific scenario we consider and the CL approaches of vision foundation models we consider in our evaluation.

3 Continual Learning of Vision Foundation Models

In this section we introduce the key concepts of CL applied to vision foundation models, which are at the basis of our empirical evaluation.

3.1 Preliminaries

CL addresses the challenge of training a model on a sequence of non-stationary data without the need to retrain on all previously used data, a process known as joint incremental training, while also preventing forgetting. Traditionally, CL has been evaluated using Convolutional Neural Network (CNN) models [2, 13, 16, 35, 60]. However, the impressive performance of large vision foundation models pre-trained on extensive datasets, such as the Vision Transformer (ViT) [12], has led recent incremental learning studies to shift their focus toward CL with these pre-trained models as opposed to training from scratch [8, 15, 43, 53].

ViT models are vision models based on the Transformer architecture [47]. In ViT, an image is divided into patches, treated as a sequence of tokens, and a learnable class token is added. The input sequence is processed by alternating layers of multi-head attention and multi-layer perceptrons. The final embedding of the input image x , represented by the class token after the last layer as $z_{\text{cls}}(x; \Theta)$ with Θ model’s parameters, is used by a classifier to generate predictions.

In the context of CL, a pre-trained ViT model \mathcal{M}_t is sequentially trained on K tasks, typically image classification tasks, each characterized by a disjoint set of classes. We can define the sequence of task used to train the model as $\mathcal{D}_K = \{\mathcal{X}_t, \mathcal{Y}_t\}_{t=1}^K$, where \mathcal{X}_t and \mathcal{Y}_t are the sets of images and classification labels for task t , respectively. The output of the model after task t for an input x is the composition of its class token $z_{\text{cls}}(x; \Theta_t)$, depending on parameters Θ_t , and a classifier with parameters W_t , which are sequentially updated on new tasks. The simplest classifier is a single linear layer (or classification head) followed by a softmax activation function, resulting in the model output:

$$\mathcal{M}_t(x; \Theta_t, W_t) \equiv p(y | x; \Theta_t, W_t) = \text{softmax}(W_t^\top z_{\text{cls}}(x; \Theta_t)). \quad (1)$$

Training \mathcal{M}_t only on the current task data t (i.e., fine-tuning) is more efficient than re-training on all previous task data. However, this approach can lead to what is known as *catastrophic forgetting*: when a model forgets the initial task after learning one or more new tasks [37]. The causes of catastrophic forgetting can be found in: (i) *weight/activation drift*, where training on a new tasks modifies crucial weights for previous tasks; (ii) *task-recency bias*, where the linear classifier output becomes biased towards new task classes, leading to uncalibrated predictions for previous classes; and (iii) *inter-task confusion*, where the final classifier, trained only on the last task’s classes, has suboptimal decision boundaries for the previous tasks’ classes [36].

CL can be viewed as a collection of training techniques aimed at mitigating catastrophic forgetting. In the following section, we give insights into specific methodologies that we utilized in our analysis.

3.2 Continual Learning Approaches

Continual learning approaches can be broadly categorized into Class-Incremental Learning (class-IL), where the model must distinguish among an increasing number of classes; Task-Incremental Learning (task-IL), where the model learns distinct tasks in sequence; and Domain-Incremental Learning (domain-IL), where the model is trained on a fixed task but with varying input domains [48].

Continual learning methods can be further distinguished as either *exemplar-based* or *exemplar-free*. The former stores and replays a subset \mathcal{E}_{t-1} of previous task data, known as *exemplars*, when learning new tasks [40], while the latter relies only on the current task data [53].

In this paper, we focus on the class-IL setting, the most explored in CL with pre-trained vision models, highlighting the key features and energy impacts of both exemplar-free and exemplar-based methods.

Exemplar-based. iCaRL [40] is an exemplar-based method, originally designed for incremental learning with CNNs, but recently adapted for pre-trained ViT architecture [53, 57]. iCaRL fine-tunes the entire ViT with both exemplars and current task data making use of a cross entropy loss and knowledge distillation as regularizer [22]. In particular, the regularizer aligns the output of the current model \mathcal{M}_t with those of the previous model \mathcal{M}_{t-1} to mitigate activation drift. At the end of each task, iCaRL mitigates task-recency bias by computing the mean of class tokens and using nearest mean classification for predictions, eliminating the need for the linear classifier W_t .

MEMO [58] is an exemplar-based method, which dynamically expand the network as the number of tasks increases. In MEMO, the initial layers of the ViT are frozen, while the final layers are duplicated and trained separately for each task. Like iCaRL, MEMO uses exemplars to mitigate the activation drift.

Exemplar-free. Recent exemplar-free approaches for pre-trained models include the *prompt-based* methods. These methods keep the pre-trained feature extractor frozen and inject a few learnable parameters, called *prompts*, before the multi-head self-attention layers. These prompts guide the pre-trained model

in learning new tasks by adapting the feature representation. The output of an incremental ViT model \mathcal{M}_t using prompts is:

$$\mathcal{M}_t(x; \Theta_0, W_t, P_t^1, \dots, P_t^M) \equiv \text{softmax}(W_t^\top z_{\text{cls}}^L(x; \Theta_0, P_t^1, \dots, P_t^M)), \quad (2)$$

where Θ_0 are the pre-trained frozen weights and P_t^1, \dots, P_t^M are the learnable prompts updated across tasks. At training and test time, these prompts are selected from a pool \mathcal{P} using a query function calculated on the input x . A common approach is to use the class token $z_{\text{cls}}^L(x; \Theta_0)$ as query function [43, 53].

Prompt-based algorithms vary in the prompt selection processes, in the number of prompts M used, and in the injection points into the model. Learning to Prompt (L2P) [53] appends learnable prompts to the embedding input sequence. DualPrompt [52] injects *general* prompts, shared across tasks, and *expert* prompts, which are task-specific, in different points along the backbone. CODA-Prompt [43] modifies DualPrompt by splitting the prompts into *prompt components* and it uses a weighted summation with learnable weights to determine which components to utilize during inference.

Another category of exemplar-free methods is the *representation-based* methods, which aim to maximize the model’s ability to generate high-quality features from the pre-trained model, even for data not specifically trained on [56].

EASE [57] freezes the pre-trained feature extractor and adapts it using task-specific learnable adapters, then uses a prototype-based classifier for predictions and estimates representation drift of old prototypes as new tasks are added. At inference time, images are passed through multiple adapters. SimpleCIL [59], similarly to [25], computes the class means of the feature representations without any training, and uses these representations as a weight matrix for a final linear classifier. RanPAC [38], like some recent CL approaches [15, 39], employs a *first session adaptation* of the backbone, where the ViT is adapted during the first task by updating a few learnable weights through parameter-efficient fine-tuning. Additionally, it enhances feature space via random projection to improve class separability. The final classification is based on a metric derived from the Gram Matrix of features and class means. For subsequent tasks, the ViT is frozen, and only the Gram Matrix and class means are updated to account for new classes.

Efficiency Considerations. The previous sections hint at the strengths and weaknesses of the various approaches in terms of efficiency, which we aim to quantify. Exemplar-free methods are more training-efficient than exemplar-based methods (i.e., iCaRL and MEMO) as they avoid sample replay during the current task training [35]. However, these methods come with their own complexities. For example, prompt-based methods require double inferences: one for prompt selection and another for the forward pass. EASE increases architecture size and necessitates multiple forward passes at inference time. RanPAC significantly expands the feature space dimensionality, resulting in a large final classifier. In the next section, we document how we empirically analyze the energy consumption of the different CL approaches, which is often taken for granted by CL literature [53, 57]. Additionally, we introduce a novel metric to evaluate the trade-off between performance and efficiency in these approaches.

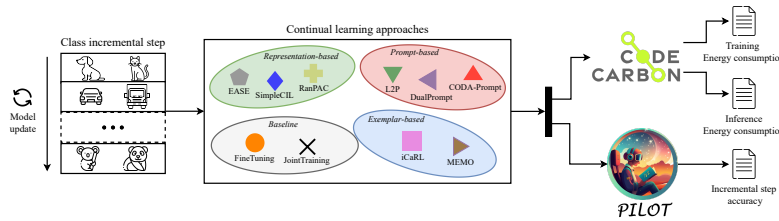


Fig. 2: Overview of our experimental methodology. PILOT [56] is the framework that implements the CL approaches measuring the accuracy over incremental training steps, while CodeCarbon [9] evaluates energy consumption during training and inference.

4 Methodology

The previous section outlines the inherent complexity entailed by the multi-faceted and heterogeneous characteristics of CL approaches. To tackle the challenges of experimenting with CL, our research methodology is based on directly measuring energy consumption of approaches, both during training and inference, followed by a thorough analysis of accuracy and efficiency trade-offs. Specifically, we aim to quantitatively understand:

- The impact of replaying old data in terms of energy cost relative to performance gains in exemplar-based approaches, identifying the incremental scenarios (if they exist) in which is most beneficial.
- The energy savings achievable by utilizing a small number of trainable parameters in prompt-based methods, balanced against the additional cost of the double inference required for prompt selection.
- The benefits of heavily relying on a pre-trained backbone without adaptation, or with a first task adaptation, weighted against the challenges posed by significant domain shifts or potential increases in inference costs.

To achieve our research goal, we use two Python libraries: the CL framework PILOT [56] and CodeCarbon [9], a package for measuring energy consumption. Figure 2 provides a high-level overview of our experimental methodology, detailing both the methods and the empirical output data collected.

In the following sections we explain in detail the procedure we leverage for measuring the energy consumption, supported by the introduction of a novel metric to evaluate models by considering both its performance and energy usage.

4.1 Measuring Energy Consumption

To monitor the energy footprint of the CL approaches, we used CodeCarbon [9], a Python package that estimates the energy usage of the key hardware components, namely memory, CPU, and GPU. Memory consumption (E_{RAM}) is estimated using a model that depends on the amount of allocated memory, while the CPU energy usage (E_{CPU}), CodeCarbon utilizes *Intel's RAPL* (Running Average

Power Limit) interface. For GPU (E_{GPU}), it uses the *pynvml* library, which is specific to NVIDIA GPUs. The energy consumption reported by RAPL and *pynvml* reflects the consumption of the respective devices and not just the observed process, so experiments must be conducted in a controlled environment [3].

Throughout this paper, the reported energy values will always refer to the *total energy consumed* during an experiment in kWh, i.e., $E = E_{\text{RAM}} + E_{\text{CPU}} + E_{\text{GPU}}$. This is not a limitation of our analysis, as we found that the distribution of energy consumption among the components remains consistent regardless of the dataset and method analyzed. Specifically, GPUs are responsible for approximately 79% of the total energy consumption during training, while CPUs and RAM account for 15% and 6%, respectively. These numbers are in line with those reported in [23]. In practical terms, for each approach, we systematically isolate the functions responsible for the training loop and evaluation and collect the respective consumption data for each incremental step.

4.2 The Energy NetScore

Many different metrics can evaluate neural networks based on accuracy and architectural and computational complexity. Canziani et al. propose the *information density* representing the accuracy-to-parameter ratio of a network [4]. Building on this metric, Wong introduces the *NetScore*, which keep in consideration the number of multiply-accumulate (MAC) operations needed for inference into this ratio [54]. The NetScore was further elaborated and adapted for CL to include other measurement such as training time, memory occupancy and the number of backward operations [18, 20].

In this paper we propose representing architectural and computational complexity by considering the energy consumed by the model during an entire experiment. This, combined with accuracy, defines what we call the *Energy NetScore* of the model \mathcal{M} , denoted as $\Omega(\mathcal{M})$. It measures the trade-off between performance and efficiency and will be used to rank the analyzed approaches. Specifically, following a similar approach to [54], we define $\Omega(\mathcal{M})$ as

$$\Omega(\mathcal{M}) := 20 \log \left(\frac{A(\mathcal{M})^\alpha}{E(\mathcal{M})^\beta} \right), \quad (3)$$

where α and β controlling the influence of accuracy $A(\mathcal{M})$ and energy consumed $E(\mathcal{M})$ of the model on the score, respectively. Although in this paper we applied this evaluation in the context of CL, we emphasize that the definition of the Energy NetScore is quite general and can be used to evaluate the trade-off between performance and consumption in any scenario.

5 Experimental Setup

Here we document the benchmarks, metrics, and experimental settings we used (see supplementary material for further details).

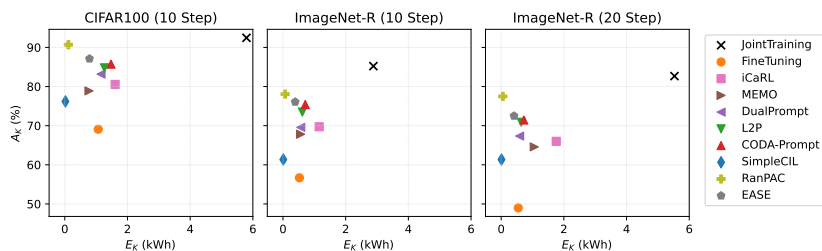


Fig. 3: Comparison in terms of training energy consumption (x -axis) and accuracy after the final incremental step (y -axis) across benchmarks and task sequence lengths.

Benchmarks. Following [38, 53, 57], we evaluated the methods on CIFAR-100 [27], ImageNet-R [21], and DN4IL [17], a balanced subset of the DomainNet dataset. For CIFAR-100, which consists of 100 classes and 60,000 images, we used a 10-step scenario where the classes are equally split into 10 tasks. For ImageNet-R, which has 200 classes and 30,000 images, we considered both 10-step and 20-step scenarios with 3,000 and 1,500 examples per task, respectively. DN4IL consists of six domains, each with 100 classes, and a total of 85,000 images. For this benchmark, each domain (i.e. quickdraw, infograph, etc) is treated as a different task, resulting in a 6-step scenario and it is used to evaluate the performance when large domain shifts occur.

Metrics. CL approaches are evaluated by measuring their accuracy after each task. We refer to this quantity as *per-step accuracy* (A_k). We denote E_k the total energy consumption upon task k , measured in kWh. For the Energy NetScore Ω_k at task k , defined in Eq. (3), following [18], we set $\alpha = 2$ and $\beta = 0.125$.

Training Details. All approaches start from a ViT-B/16 [12] pretrained on ImageNet-21K and fine-tuned on ImageNet-1K as the backbone. For hyperparameters we followed those suggested in PILOT. To ensure a fair comparison between approaches, we maintained a fixed batch size of 64 and set the number of epochs to 20 for each task. JointTraining is not implemented in PILOT and we follow the settings suggested in [38]. For iCaRL and MEMO, we use 20 exemplars per class as memory scenarios, as is common in the literature [26].

Hardware. The experiments were conducted on a Linux server with an Intel[®] Core[™] i5-10600KF CPU, an NVIDIA RTX[™] 3060 GPU, and 64GB of DDR3 RAM. We ensure CodeCarbon recognizes both CPU and GPU, enabling real-time energy consumption data collection without approximations.

6 Experimental Results

In this section, we present and discuss the results of our empirical experiment, with a primary focus on addressing our research questions (see Section 1).

How do CL approaches compare in training energy consumption? Figure 3 gives a high-level overview of the relationship between training energy

consumption (on the x -axis) and accuracy (on the y -axis) at the end of class-incremental training. JointTraining clearly achieves the highest accuracy, but also that it consumes significantly more energy than all other methods.

Prompt-based methods like L2P, DualPrompt, and CODA-prompt achieve higher accuracy than FineTuning, while maintaining comparable energy costs despite having two orders of magnitude fewer trainable parameters, as shown in Figure 4. This challenges the common belief that the number of trainable parameters determines model efficiency, especially when starting from pre-trained models. The high energy consumption of prompt-based methods is due to the two forward passes for each backward pass: one for selecting prompts from the prompt pool, and another for making predictions based on the selected prompts.

Representation-based methods achieve the best balance between efficiency and accuracy (see Figure 3). As expected, SimpleCIL demonstrates very good energy efficiency and, thanks to the strong backbone, and a decent accuracy, outperforming FineTuning despite having no trainable parameters. RanPAC achieves optimal performance with minimal energy overhead by using a first task adaptation without further training on subsequent tasks. These observations are consistent across all tested scenarios, indicating that the trade-off between consumption and accuracy does not change when varying benchmarks.

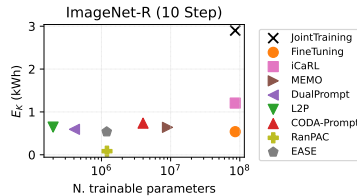


Fig. 4: Trainable parameters versus Energy Consumption. SimpleCIL has zero parameters, while for RanPAC we select the number of trainable parameters for the first task.

How do training energy costs scale with data and model updates?

Figure 3 provides only a partial perspective, as it does not explore how energy consumption scales with an increasing number of tasks. Figure 5 completes the picture by showing on the left plot that all exemplar-free methods exhibit *linear* growth in the cumulative energy consumption, in contrast to Joint Training, iCaRL and MEMO, which show *quadratic* growth with the number of tasks, though in different manners. This observation is crucial, indicating that for a very large (potentially infinite) sequence of tasks, exemplar-based methods and in particular JointTraining may incur an unsustainable cost. Moreover, the bar plot on the right of Figure 5 illustrates the effect of splitting the *same* dataset into more tasks, i.e., halving the task sizes. The final energy consumption remains unchanged for all exemplar-free methods, but increases for the others.

How do CL approaches compare in Energy NetScore? The results of the experiments on CIFAR-100 and ImageNet-R are summarized in Table 1, which includes the Energy NetScore Ω_K , defined in Equation 3 at the end of training. According to the Energy NetScore metric, the representation-based methods (i.e., SimpleCIL, EASE and RanPAC) significantly outperform the competitors. For relatively small benchmarks and short task sequences, exemplar-based methods, especially JointTraining, perform as well as or better than prompt-based

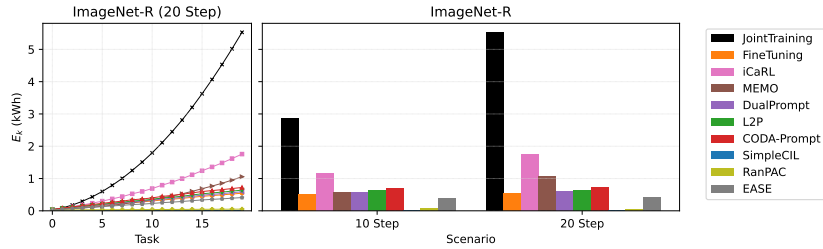


Fig. 5: **Left:** Cumulative training energy consumption increases linearly for exemplar-free methods as the number of tasks grows, while exemplar-based methods show quadratic growth. **Right:** Total training energy consumption for each CL strategy on ImageNet-R split into 10 and 20 tasks, thus indicating fewer or more samples per task. Exemplar-based methods consume more energy with smaller tasks, whereas exemplar-free methods’ energy consumption remains independent of task size.

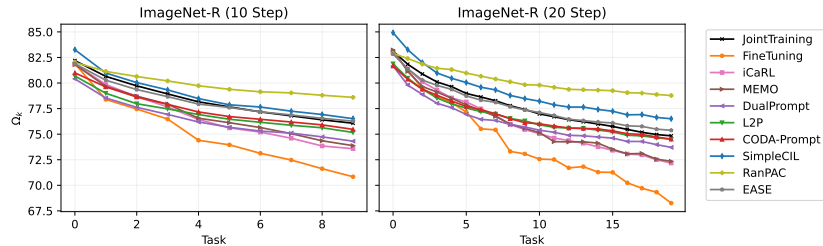


Fig. 6: The Energy NetScore Ω_k across incremental steps on ImageNet-R. RanPAC consistently outperforms all others in both scenarios and on each task. Exemplar-based models perform well initially but become sub-optimal with increasing updates.

methods. However, increasing the demand for model update on the same dataset, i.e., halving the task sizes in our experiments, the prompt-based methods close the gap with the exemplar-based methods. This trend is clearly shown in Figure 6, where iCaRL initially scores competitively with the prompt-based methods, but as the number of exemplars during the training grows, its increased energy consumption lowers its score in the final tasks. A similar pattern can be observed when comparing EASE and JointTraining on ImageNet-R - 20 Step, where their positions in the ranking switch after 10 steps.

Domain and class-IL experiments. We conducted an additional experiment in a challenging scenario, where each task involves changes in classes and in the domain. This setting better reflects real-world applications, where different objects may be observed under varying external conditions or using different instruments. Figure 7 shows that representation-based methods, particularly SimpleCIL and RanPAC, continue to offer the best trade-off between accuracy and energy consumption. However, compared to previous scenarios, the accuracy gap with exemplar-based methods narrows, while it widens with JointTraining (see

Table 1: Results on CIFAR100 and ImageNet-R. A_K , E_K and Ω_K represent accuracy(%), energy consumed(kWh), and Energy NetScore at the end of the task sequence, respectively. The memory column reports the number of exemplars employed.

Method	Memory	CIFAR100 (10 Step)			ImageNet-R (10 Step)			ImageNet-R (20 Step)		
		$A_K(\uparrow)$	$E_K(\downarrow)$	$\Omega_K(\uparrow)$	$A_K(\uparrow)$	$E_K(\downarrow)$	$\Omega_K(\uparrow)$	$A_K(\uparrow)$	$E_K(\downarrow)$	$\Omega_K(\uparrow)$
Joint Training	∞	92.45	5.80	76.73	85.22	2.88	76.07	82.67	5.53	74.84
Fine Tuning	\times	69.09	1.07	73.50	56.68	0.52	70.85	48.98	0.55	68.25
iCaRL [40]	20/class	80.55	1.61	75.73	69.72	1.15	73.58	66.00	1.76	72.17
MEMO [58]	20/class	78.90	0.77	76.17	67.82	0.56	73.88	64.58	1.06	72.34
L2P [53]	\times	84.78	1.26	76.88	73.47	0.62	75.16	70.87	0.64	74.50
DualPrompt [52]	\times	83.18	1.16	76.64	69.58	0.57	74.31	67.35	0.59	73.71
CODA-Prompt [43]	\times	85.73	1.48	76.90	75.42	0.71	75.47	71.45	0.72	74.52
SimpleCIL [59]	\times	76.12	0.02	<u>79.53</u>	61.35	0.01	<u>76.51</u>	61.35	0.01	<u>76.51</u>
EASE [57]	\times	<u>87.11</u>	0.79	77.86	<u>76.08</u>	0.39	76.27	<u>72.50</u>	0.41	75.38
RanPAC [38]	\times	90.69	<u>0.12</u>	80.60	78.07	<u>0.07</u>	78.59	77.48	<u>0.05</u>	78.82

Table 2). For instance, comparing with the 10-step ImageNet-R experiment, the performance gap between RanPAC and iCaRL narrows from 9% to 1.5% points, while the gap with JointTraining widens from 5% to 10% points. This highlights the growing criticism of methods that either do not adapt the backbone or only use first-session adaptation, as they fail to fully address the challenges of CL [45]. Such methods may be inadequate in scenarios where the context differs significantly from pre-training or where the domain gap between tasks shifts abruptly. Finally, Prompt-based methods and EASE seem to suffer in this setting.

Method	Memory	DN4IL - 6 Step		
		$A_K(\uparrow)$	$E_K(\downarrow)$	$\Omega_K(\uparrow)$
Joint Training	∞	76.04	4.78	73.54
Fine Tuning	\times	30.68	1.43	59.09
iCaRL [40]	20/class	<u>66.25</u>	2.64	71.79
MEMO [58]	20/class	62.37	1.16	71.64
L2P [53]	\times	40.37	1.68	63.62
DualPrompt [52]	\times	37.78	1.55	62.61
CODA-Prompt [43]	\times	41.02	1.96	63.79
SimpleCIL [59]	\times	57.99	0.03	<u>74.34</u>
EASE [57]	\times	50.74	1.05	68.16
RanPAC [38]	\times	67.85	<u>0.20</u>	75.01

Table 2: Results DN4IL. A_K , E_K and Ω_K represent accuracy, energy consumed, and Energy NetScore at the end of the task sequence, respectively.

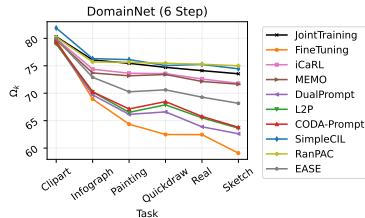


Fig. 7: The Energy NetScore Ω_k as a function of the incremental step. Each step represent a different domain, as specified on the x -axis.

How does the CL strategy impact inference consumption? Once trained, machine learning models are used for inference, which can consume substantial energy due to high request volumes [32]. Thus, assessing a model’s environmental impact must consider efficiency during both training and inference. Figure 8 summarizes our findings regarding the inference energy consumption on ImageNet-R, with similar conclusions applicable to other benchmarks.

The bar plot on the left shows the energy consumption for a single inference (i.e., batch size of 1) for each method. EASE has high inference consumption

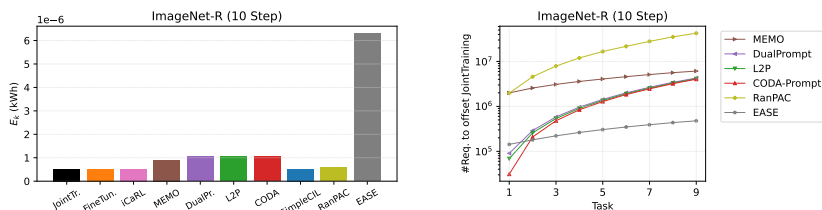


Fig. 8: **Left:** Energy consumption for a single inference for each approach. **Right:** The number of requests required for approaches consuming more energy per inference, compared to JointTraining, to offset the energy savings gained during training.

despite low training costs (see Table 1) as it repeats the inference along the backbone for each trained adapter, leading to linear scaling of the inference energy consumption with the number of tasks. Other methods, except MEMO that is a dynamic expandable architecture method, maintain constant energy consumption regardless of task sequence length. However, we observe that Prompt-based methods, despite minimal parameter overhead, consume roughly double the inference energy of other methods due to requiring a two forward pass for each prediction. In contrast, RanPAC, which only requires a single forward pass, effectively controls energy consumption, adding only a small overhead from the random projection in a larger feature space. SimpleCIL, FineTuning, iCaRL, and JointTraining incur in no additional inference costs beyond a standard forward pass. Specifically, the energy impact of adding exemplars is limited to the training phase, and regularizers like Knowledge Distillation (KD), despite needing two forward passes during training, do not increase inference costs.

Observing that a subset of CL approaches, aimed at reducing training costs, consume more energy during inference compared to JointTraining, raises a natural question: After how many requests do CL methods with inference overhead consume as much energy as retraining the model each time on the entire dataset? In Figure 8 (right), we show the results of this analysis. Specifically, the y -axis represents the number of model requests after which the accumulated training savings up to task k are offset by the inference overhead. We observe that RanPAC, which adds minimal consumption per inference, proves to be the most efficient among all approaches considered, reaching the break-even point with JointTraining after more than 10^7 requests. Prompt-based methods reach the break-even point with JointTraining at roughly the same time, as they have similar inference consumptions. EASE, due to its linear increase in inference energy consumption, reaches the break-even point with JointTraining after significantly fewer requests compared to all other methods.

Note that these numbers depend significantly on the size of the training dataset. Higher energy usage during training increases the number of requests required for methods with additional inference cost to reach the break-even point with JointTraining. Nevertheless, we believe that a truly effective CL algorithm should not have additional overhead during inference compared to JointTraining.

Comprehensive evaluation. Here, we test a realistic scenario where training and inference alternate during the incremental step. For this analysis, we considered the energy consumed for both training and answering (after each training step) to 10,000 requests, a number comparable to the size of the training dataset. These consumption values were used to compute the Energy NetScore Ω_k , shown in Figure 9. The solid line represents the Ω_k score with zero inference cost, while the dashed line includes inference costs. SimpleCIL loses 2.5 points due to the impact of inference compared to its low training costs, while EASE loses about one point due to high inference costs. Despite a slight inference overhead, RanPAC remains the best solution in terms of Ω_k . Other methods with higher training costs and relatively low inference costs, show minimal impact on their Energy NetScore from this number of requests.

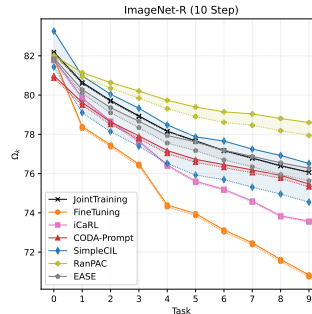


Fig. 9: Ω_k as a function of the steps on ImageNet-R. Each step considers the consumption from both training and 10,000 inferences when computing Ω_k .

7 Conclusion

How green is continual learning, really? In this paper we presented the first systematic analysis of the energy consumption of CL approaches with pre-trained backbones, and our results clearly indicate that *it depends*. Our study highlights the complexities in selecting suitable CL algorithms, as these choices are highly context-dependent and influenced by factors such as the frequency of model updates, the magnitude of domain changes between steps, and the demands of inference. Nonetheless, our findings allow us to draw several general conclusions:

- Representation-based approaches, especially RanPAC, are the most energy-efficient during training while still maintaining performance close to JointTraining, as shown by the Ω_k values in Table 1. However, they still suffer when tested in large-domain shift scenarios (see Table 2).
- Exemplar-based methods are effective when the domain gap between steps is significant, performing comparably to representation-based approaches in Ω_k despite their quadratic growth in training energy consumption with an increasing number of tasks (see Table 2).
- The minimal overhead of RanPAC for inference makes it particularly appealing for real-world applications (see Figures 8 and 9). In contrast, methods like EASE and MEMO, whose parameters and inference costs increase with the number of tasks, or prompt-based models, which require two forward passes per prediction, are less suitable for handling high volumes of requests.
- CL algorithms, to be competitive with JointTraining regardless of their potential usage, should not incur any additional computational overhead during inference.

Acknowledgements

This work was supported by funding by the European Commission Horizon 2020 grant #951911 (AI4Media).

References

1. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
2. Boschini, M., Bonicelli, L., Buzzega, P., Porrello, A., Calderara, S.: Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(5), 5497–5512 (2023). <https://doi.org/10.1109/TPAMI.2022.3206549>
3. Bouza, L., Bugeau, A., Lannelongue, L.: How to estimate carbon footprint when training deep learning models? a guide and review. *Environmental Research Communications* **5**(11), 115014 (2023)
4. Canziani, A., Paszke, A., Culurciello, E.: An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678 (2016)
5. Castro, F.M., Guil, N., Marín-Jiménez, M.J., Pérez-Serrano, J., Ujaldón, M.: Energy-based tuning of convolutional neural networks on multi-gpus. *Concurrency and Computation: Practice and Experience* **31**(21), e4786 (2019)
6. Chavan, V., Koch, P., Schlüter, M., Briese, C.: Towards realistic evaluation of industrial continual learning scenarios with an emphasis on energy consumption and computational footprint. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 11506–11518 (2023)
7. de Chavannes, L.H.P., Kongsbak, M.G.K., Rantzau, T., Derczynski, L.: Hyperparameter power impact in transformer language model training. In: *Proceedings of the second workshop on simple and efficient natural language processing*. pp. 96–118 (2021)
8. Cossu, A., Carta, A., Passaro, L., Lomonaco, V., Tuytelaars, T., Bacciu, D.: Continual pre-training mitigates forgetting in language and vision. *Neural Networks* p. 106492 (2024). <https://doi.org/https://doi.org/10.1016/j.neunet.2024.106492>, <https://www.sciencedirect.com/science/article/pii/S0893608024004167>
9. Courty, B., Schmidt, V., Luccioni, S., Goyal-Kamal, MarionCoutarel, Feld, B., Lecourt, J., LiamConnell, Saboni, A., Inimaz, supatomic, Léval, M., Blanche, L., Cruveiller, A., ouminasara, Zhao, F., Joshi, A., Bogroff, A., de Lavoreille, H., Laskaris, N., Abati, E., Blank, D., Wang, Z., Catovic, A., Alencon, M., Stęchły, M., Bauer, C., Lucas-Otavio, JPW, MinervaBooks: mlco2/codecarbon: v2.4.1 (May 2024). <https://doi.org/10.5281/zenodo.11171501>, <https://doi.org/10.5281/zenodo.11171501>
10. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(7), 3366–3385 (2022). <https://doi.org/10.1109/TPAMI.2021.3057446>
11. Dodge, J., Prewitt, T., Tachet des Combes, R., Odmark, E., Schwartz, R., Strubell, E., Luccioni, A.S., Smith, N.A., DeCario, N., Buchanan, W.: Measuring the carbon intensity of AI in cloud instances. In: *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*. pp. 1877–1894 (2022)

12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
13. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 86–102. Springer International Publishing, Cham (2020)
14. Georgiou, S., Kechagia, M., Sharma, T., Sarro, F., Zou, Y.: Green AI: Do deep learning frameworks have different costs? In: Proceedings of the 44th International Conference on Software Engineering. pp. 1082–1094 (2022)
15. Goswami, D., Liu, Y., Twardowski, B., van de Weijer, J.: Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems* **36** (2024)
16. Goswami, D., Soutif-Cormerais, A., Liu, Y., Kamath, S., Twardowski, B., van de Weijer, J.: Resurrecting old classes with new data for exemplar-free continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 28525–28534 (June 2024)
17. Gowda, S., Zonooz, B., Arani, E.: Dual cognitive architecture: Incorporating biases and multi-memory systems for lifelong learning. *Transactions on Machine Learning Research*
18. Harun, M.Y., Gallardo, J., Hayes, T.L., Kanan, C.: How efficient are today’s continual learning algorithms? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2430–2435 (2023)
19. Hashemi, S., Anthony, N., Tann, H., Bahar, R.I., Reda, S.: Understanding the impact of precision quantization on the accuracy and energy of neural networks. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017. pp. 1474–1479. IEEE (2017)
20. Hayes, T.L., Kanan, C.: Online continual learning for embedded devices. In: Conference on Lifelong Learning Agents (CoLLAs) (August 2022)
21. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV* (2021)
22. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
23. Hodak, M., Gorkovenko, M., Dholakia, A.: Towards power efficiency in deep learning on data center hardware. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 1814–1820. IEEE (2019)
24. Hurtado, J., Salvati, D., Semola, R., Bosio, M., Lomonaco, V.: Continual learning for predictive maintenance: Overview and challenges. *Intelligent Systems with Applications* **19**, 200251 (2023). <https://doi.org/https://doi.org/10.1016/j.iswa.2023.200251>, <https://www.sciencedirect.com/science/article/pii/S2667305323000765>
25. Janson, P., Zhang, W., Aljundi, R., Elhoseiny, M.: A simple baseline that questions the use of pretrained-models in continual learning. In: NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications (2022), <https://openreview.net/forum?id=dnVNYctP3S>
26. Kang, M., Park, J., Han, B.: Class-Incremental Learning by Knowledge Distillation with Adaptive Feature Consolidation. In: CVPR (2022)

27. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical report (2009)
28. Krzywaniak, A., Czarnul, P., Proficz, J.: Gpu power capping for energy-performance trade-offs in training of deep convolutional neural networks for image recognition. In: International conference on computational science. pp. 667–681. Springer (2022)
29. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (May 2015). <https://doi.org/10.1038/nature14539>, <https://doi.org/10.1038/nature14539>
30. Li, Z., Hoiem, D.: Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(12), 2935–2947 (2018). <https://doi.org/10.1109/TPAMI.2017.2773081>
31. Lomonaco, V., Maltoni, D.: Core50: a new dataset and benchmark for continuous object recognition. In: Levine, S., Vanhoucke, V., Goldberg, K. (eds.) Proceedings of the 1st Annual Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 78, pp. 17–26. PMLR (13–15 Nov 2017), <https://proceedings.mlr.press/v78/lomonaco17a.html>
32. Luccioni, A.S., Viguier, S., Ligozat, A.L.: Estimating the carbon footprint of bloom, a 176b parameter language model. *Journal of Machine Learning Research* **24**(253), 1–15 (2023), <http://jmlr.org/papers/v24/23-0069.html>
33. Magistri, S., Baracchi, D., Shullani, D., Bagdanov, A.D., Piva, A.: Towards continual social network identification. In: 2023 11th International Workshop on Biometrics and Forensics (IWBF). pp. 1–6 (2023). <https://doi.org/10.1109/IWBF57495.2023.10157835>
34. Magistri, S., Baracchi, D., Shullani, D., Bagdanov, A.D., Piva, A.: Continual learning for adaptive social network identification. *Pattern Recognition Letters* **180**, 82–89 (2024). <https://doi.org/https://doi.org/10.1016/j.patrec.2024.02.020>, <https://www.sciencedirect.com/science/article/pii/S0167865524000540>
35. Magistri, S., Trinci, T., Soutif, A., van de Weijer, J., Bagdanov, A.D.: Elastic feature consolidation for cold start exemplar-free incremental learning. In: The Twelfth International Conference on Learning Representations (2024), <https://openreview.net/forum?id=7D9X2cFnt1>
36. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(5), 5513–5533 (2023). <https://doi.org/10.1109/TPAMI.2022.3213473>
37. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, vol. 24, pp. 109–165. Academic Press (1989). [https://doi.org/https://doi.org/10.1016/S0079-7421\(08\)60536-8](https://doi.org/https://doi.org/10.1016/S0079-7421(08)60536-8), <https://www.sciencedirect.com/science/article/pii/S0079742108605368>
38. McDonnell, M.D., Gong, D., Parvaneh, A., Abbasnejad, E., van den Hengel, A.: Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems* **36** (2024)
39. Panos, A., Kobe, Y., Reino, D.O., Aljundi, R., Turner, R.E.: First session adaptation: A strong replay-free baseline for class-incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18820–18830 (2023)
40. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)

41. Rodrigues, C.F., Riley, G., Luján, M.: Synergy: An energy measurement and prediction framework for convolutional neural networks on jetson tx1. In: Proceedings of the international conference on parallel and distributed processing techniques and applications (PDP'TA). pp. 375–382. The Steering Committee of The World Congress in Computer Science, Computer ... (2018)
42. Shaheen, K., Hanif, M.A., Hasan, O., Shafique, M.: Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. *Journal of Intelligent & Robotic Systems* **105**(1), 9 (Apr 2022). <https://doi.org/10.1007/s10846-022-01603-6>, <https://doi.org/10.1007/s10846-022-01603-6>
43. Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: 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. pp. 11909–11919 (2023)
44. Tao, Y., Ma, R., Shyu, M.L., Chen, S.C.: Challenges in energy-efficient deep neural network training with fpga. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 400–401 (2020)
45. Thede, L., Roth, K., Hénaff, O.J., Bethge, M., Akata, Z.: Reflecting on the state of rehearsal-free continual learning with pretrained models. arXiv preprint arXiv:2406.09384 (2024)
46. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 30. Curran Associates, Inc. (2017)
48. Van de Ven, G.M., Tuytelaars, T., Tolias, A.S.: Three types of incremental learning. *Nature Machine Intelligence* **4**(12), 1185–1197 (2022)
49. Verdecchia, R., Cruz, L., Sallou, J., Lin, M., Wickenden, J., Hotellier, E.: Data-centric Green AI: An exploratory empirical study. In: 2022 international conference on ICT for sustainability (ICT4S). pp. 35–45. IEEE (2022)
50. Verdecchia, R., Sallou, J., Cruz, L.: A systematic review of green ai. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **13**(4), e1507 (2023)
51. Verwimp, E., Aljundi, R., Ben-David, S., Bethge, M., Cossu, A., Gepperth, A., Hayes, T.L., Hüllermeier, E., Kanan, C., Kudithipudi, D., Lampert, C.H., Mundt, M., Pascanu, R., Popescu, A., Tolias, A.S., van de Weijer, J., Liu, B., Lomonaco, V., Tuytelaars, T., van de Ven, G.M.: Continual learning: Applications and the road forward. *Transactions on Machine Learning Research* (2024), <https://openreview.net/forum?id=axBIMcGZn9>
52. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: European Conference on Computer Vision. pp. 631–648. Springer (2022)
53. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 139–149 (2022)
54. Wong, A.: Netscore: towards universal metrics for large-scale performance analysis of deep neural networks for practical on-device edge usage. In: International Conference on Image Analysis and Recognition. pp. 15–26. Springer (2019)

55. Zhang, B., Davoodi, A., Hu, Y.H.: Exploring energy and accuracy tradeoff in structure simplification of trained deep neural networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **8**(4), 836–848 (2018)
56. Zhou, D.W., Sun, H.L., Ning, J., Ye, H.J., Zhan, D.C.: Continual learning with pre-trained models: A survey. In: *IJCAI*. pp. 8363–8371 (2024)
57. Zhou, D.W., Sun, H.L., Ye, H.J., Zhan, D.C.: Expandable subspace ensemble for pre-trained model-based class-incremental learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 23554–23564 (2024)
58. Zhou, D.W., Wang, Q.W., Ye, H.J., Zhan, D.C.: A model or 603 exemplars: Towards memory-efficient class-incremental learning. In: *The Eleventh International Conference on Learning Representations* (2023)
59. Zhou, D.W., Ye, H.J., Zhan, D.C., Liu, Z.: Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv preprint arXiv:2303.07338* (2023)
60. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 5871–5880 (June 2021)