Towards Realistic Evaluation of Industrial Continual Learning Scenarios with an Emphasis on Energy Consumption and Computational Footprint

Vivek Chavan* Paul Koch Marian Schlüter Clemens Briese

Fraunhofer IPK, Berlin, Germany

Abstract

Incremental Learning (IL) aims to develop Machine Learning (ML) models that can learn from continuous streams of data and mitigate catastrophic forgetting. We analyse the current state-of-the-art Class-IL implementations and demonstrate why the current body of research tends to be one-dimensional, with an excessive focus on accuracy metrics. A realistic evaluation of Continual Learning methods should also emphasise energy consumption and overall computational load for a comprehensive understanding. This paper addresses research gaps between current IL research and industrial project environments, including varying incremental tasks and the introduction of Joint Training in tandem with IL. We introduce InVar-100 (Industrial Objects in Varied Contexts), a novel dataset meant to simulate the visual environments in industrial setups and perform various experiments for IL. Additionally, we incorporate explainability (using class activations) to interpret the model predictions. Our approach, RECIL (Real-World Scenarios and Energy Efficiency Considerations for Class Incremental Learning) provides meaningful insights about the applicability of IL approaches in practical use cases. The overarching aim is to bring the Incremental Learning and Green AI fields together and encourage the application of CIL methods in real-world scenarios. Code and dataset are available.

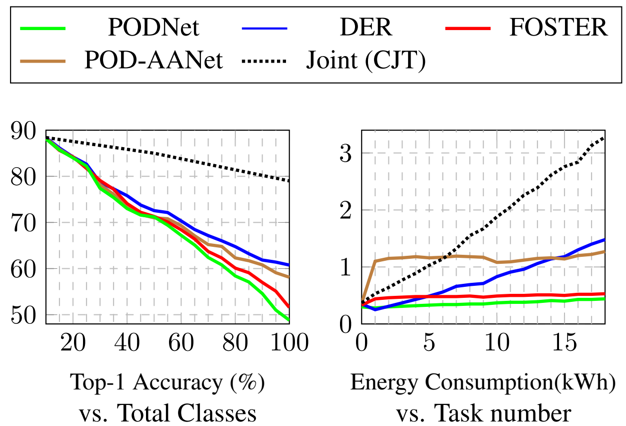


Figure 1: Top-1 Accuracy and Task-wise Energy Consumption for ImageNet-Subset for different CIL approaches. Task 0 introduces 10 classes, and all subsequent tasks add 5 classes. The total energy consumption of an approach is given by the area under the curve. Comparing the methods using only accuracy provides an incomplete understanding; computational footprint consideration is also important.

1 Introduction

Advances in Machine Learning (ML) and Computer Vision have demonstrated the capabilities of deep neural network-based (NN) models to learn from diverse data and perform a multitude of tasks with high accuracy [15, 30, 41]. The utilisation of ML in industrial applications is expected to increase substantially [6, 48, 52, 54]. A gradual ramp-up of raw materials, components, and related data occurs in industrial projects with long timelines (e.g. manufacturing [17, 29], reverse logistics [2, 57]). This necessitates the retraining of the ML model by sequentially learning from new data streams (tasks). It is established that NN models tend to forget the information learned from older data as they are retrained on new information; this phenomenon is known as *Catastrophic forgetting* [21, 27]. Iteratively retraining the model from scratch on the entire appended dataset is not a viable long-term solution, since it would result in a compounding of training times and computational load. Such industrial applications present an opportunity for widespread adoption and implementation of Continual Learning.

*Correspondence: vivek.chavan@ipk.fraunhofer.de
Dataset: http://dx.doi.org/10.24406/fordatis/266.2
DOI: 10.24406/fordatis/266.2
Code: https://github.com/Vivek9Chavan/RECIL

Metric	POD	DER	FOS	P-AA	CJT
#Param(M) ↓	11.7	213.1	22.14	13.1	11.2
#PFLOPs ↓	0.55	5.39	1.06	1.97	31.81
Time (h) ↓	54.9	119.5	73.5	179.5	261.3
E (kWh) ↓	6.86	15.08	6.12	21.25	33.29
Acc_{avg} (%) ↑	68.6	72.4	69.8	71.1	83.7

Table 1: Incremental Learning Results on ImageNet Subset with energy consumption and computational footprint. ↑ indicates higher value is better, ↓ indicates lower value is better.

Developing ML systems that can continuously learn and adapt to new data has been a broad topic of research in Artificial Intelligence (AI) for numerous applications [10, 25, 26]. In recent years, several implementations have been proposed for combating catastrophic forgetting and incrementally training ML models on new tasks. Van de ven and Tolias [60] identified three scenarios for Incremental Learning, viz. Task-, Domain- and Class-Incremental Learning (CIL), the last being the most challenging of the three.

CIL can be generalised as an ML problem with a continuously growing dataset D, where new classes are introduced sequentially over training tasks *0, 1,...T*, each containing new classes C_0, C_1...C_T. The model must be able to classify the test images from all available classes $\sum_{i=0}^{t} C_i$ at a given phase t of the project life cycle. CIL implementations generally focus heavily on the Top-1 and Top-5 accuracy in standardised benchmarking settings [5, 47, 60, 73]. While this allows direct comparison between different implementations, it leads to an overemphasis on the established benchmarks, while neglecting diverse scenarios and other metrics. After a production ML model is trained (Task 0), the subsequent incoming data is likely to contain a significant variation in the number of object classes, amount of data and feature complexities from one task to the next [1, 2, 9, 57].

Additionally, original research works and review papers on IL generally do not expound on the training times, energy consumption or computational complexity. *Reduction in training time* and *lower energy consumption* are the key reasons for adopting a continual learning-based framework in practice. Theoretically, if the accuracy of prediction were the only metric of significance, then retraining the model on the whole dataset (Cumulative Joint Training: CJT) would be preferred over incremental learning implementations [5, 47, 72]. Thus, the current body of research in this field is lacking and one-dimensional. As shown in Figure 1 and Table 1, comparing IL approaches only using accuracy metrics is not sufficient.

This study focuses on the gap between current AI research and its practical implementation in industry projects. We advocate for IL research to be extended to practical scenarios with an emphasis on energy consumption and computational footprint. We look at performance metrics and considerations for comparing IL methods comprehensively. Industrial ML projects also require maintaining performance above a certain threshold, which depends on the requirements, complexity of the problem and available data [6, 8, 54]. In a continual learning framework, this means introducing periodic Joint Training updates (JT_{update}) in tandem with incremental training. We study the impact of such updates on different state-of-the-art CIL approaches. We also introduce a novel dataset of industrial objects in varied contexts, spanning different levels of intra-class visual complexities w.r.t. classification. We study Class Activation Maps (CAMs) [50, 70] to interpret and understand the prediction patterns for the approaches. Our overarching aim is to bring the domains of Green AI and Incremental Learning together and provide the AI community with useful tools for the same. This work also aims to propose methodologies for researchers and AI adopters to assess the suitability of continual learning frameworks for their own use cases.

2 Related work

Incremental Learning. Tackling catastrophic forgetting and the plasticity-rigidity dilemma in a continual learning framework is an active area of research [10, 45, 66]. Approaches such as packNet [46] and EWC [36] perform well on task and domain-IL problems respectively, but suffer severe catastrophic forgetting in CIL settings with a growing number of object classes. Several approaches have been proposed to address the CIL-specific challenges, including iCARL [55], UCLR [33], IL2M [3], Weight Aligning (WA) [69], PODNet [19], CCIL [49], Few-Shot Learning [12, 59, 71], DER [64], AANet [42], FOSTER [62], among others. Recently, Transformer-based [18, 61] based IL approaches have also been suggested [16, 20]. Generally, CIL implementations involve regularisation-based intervention, model augmentation, and rehearsal memory among other techniques to mitigate catastrophic forgetting and maintain model plasticity [5, 47]. A majority of approaches employ Herding [55] for selecting memory exemplars. Alternative solutions have also been proposed, including kNN search [34], Mnemonics [44] and RMM [43]. Other works propose CIL implementations without rehearsal memory storage [4, 51, 74].

Green AI. In the context of broader AI research, an emphasis on *efficiency* and *energy consumption* is still lacking [39, 40, 58]. Improvement in state-of-the-art often corresponds with an increase in model size, training data size and computational complexity [7, 13, 14], and the development of compute-optimal models is infrequent [32, 67]. W.r.t. IL research, we observe a similar trend. Comparison of IL approaches w.r.t. model size and computational complexity is not widely reported [20, 42, 64]. Based on our estimation, approx. 120 papers were published at top-tier ML Conferences within the last year on the theme of *Continual Learning*, the majority of which focus on accuracy as the sole metric for comparison with other works of research, with no consideration of the computational load.

Research Gaps. While, IL research aims to resolve several practical issues, such as catastrophic forgetting, data privacy [11, 28, 51, 75] and data imbalance [3, 33, 63], the lack of focus and transparency w.r.t. computational footprint is one of the general research gaps in IL. We also address research questions specific to industrial implementations. This includes analysing the performance of CIL implementations on tasks of varying class sizes, varying rehearsal memory limits, and their applicability for projects with long timelines with JT_{update}. In that regard, a change of perspective is seen for practical implementation. Instead of the question: *Which IL implementation yields the highest incremental accuracy on the dataset?*, the focus is likely to be on: *What is the optimal configuration of tasks T that can be performed using IL, in tandem with periodic* JT_{update} *that yields acceptable performance based on accuracy and computational requirements for the application?*

Most IL implementations use CIFAR-100 [38] and ImageNet [15] datasets, which do not reflect the controlled visual environments in industrial setups. Data collected in nascent industrial production environments tend to be uncurated and heterogeneous, and issues such as cropping, blur, occlusion, and clutter may be present. Additionally, the classification between objects may be fine-grained. We study the effects of such visual intricacies on IL-based frameworks using data collected by our team.

3 Methods

Setup. We use FACIL [47] and PyCIL [72] toolboxes along with the open-source implementations from the original works for our research. We chose PODNet (POD) [19], DER [64], FOSTER (FOS) [62] and POD+AANet with RMM (P-AA) [43] for an in-depth analysis. ResNet-18 [30] with He initialisation is used as the base network across all implementations. A dedicated, low-performance workstation (16GB System Memory, 8 Cores with 1 GPU-NVIDIA GeForce GTX 1070) is used during the investigation to allow for an impartial comparison between the implementations. From our observations, the impact of excessive heating on energy consumption results was more pronounced in the case of larger workstations. However, this had a comparatively minor effect on the chosen system.

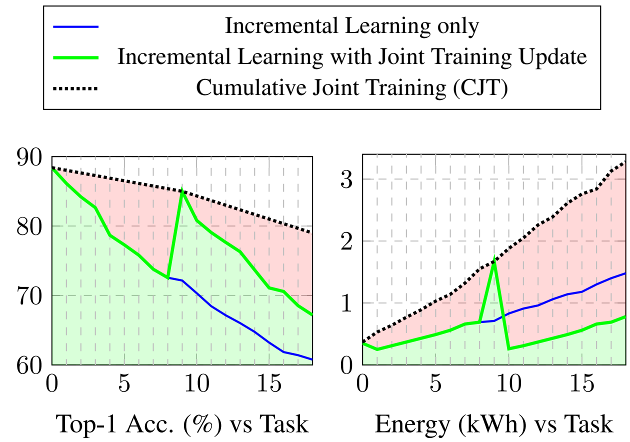


Figure 2: Accuracy and Energy Curves for practical incremental learning use case with the DER Implementation

3.1 Computational Footprint Considerations

Previous works discuss several approaches for tracking energy consumption and carbon footprint for ML training, including electricity usage, elapsed time and model parameter size [22, 23, 58, 65]. In particular, measuring the computational complexity in terms of Floating Point Operations (FLOPs) or Multiply–Accumulate Operations (MACs) is a hardware-agnostic approach. It should be noted, however, that these values do not directly correlate with the actual energy consumption and run time [58]. This is especially significant when comparing IL implementations due to the supplementary processes, such as feature-boosting [62], finetuning [19, 43] and exemplar selection. We use the Shelly smart power plug [56] to monitor and log the real-time task-wise energy consumption for the experiments. This allows a comparison of the overall energy consumption as well as the general trend over the project duration.

The ImageNet-Subset (comprising 100 randomly chosen classes from the larger dataset [15]) is used as a starting point for the study. 10 classes are introduced during the initial training, and 5 new classes are introduced during each new increment (18 incremental tasks, initial training is Task 0). The pre-established rehearsal memory of 2000 exemplars was used for all tasks. Figure 1 and Table 1 give the results of this setup for the selected CIL methods and the equivalent joint training. Cumulative Joint Training (CJT) represents a naive approach in which the model is retrained from scratch during each new task on the entire dataset. This approach represents the upper bound in terms of accuracy. The cumulative energy consumption of the implementation up until a given task (T) is given by the area under the curve or sum of the task-wise consumption values $E_{CIL} = \sum_i^T E_i$.

We measure the cumulative FLOP count for the IL project *#FLOPs*, by adding the computational complexity of each individual task *i*, which, in turn, is given by the complexity due to n_i total training samples (exemplars and new data) in that task with an input size of s = (3,224,224). In this paper, we report results in Petaflops (*#PFLOPs*).

$$\#FLOPs = \sum_i FLOP_i = \sum_i \sum_{n_i} FLOP_s \quad (1)$$

Figure 2 shows the modified results for the DER implementation, where JT_{update} is introduced midway through incremental training. In such cases, we introduce a weighing factor W_i, which is proportional to the deployment period of an incrementally trained model prior to the next incremental training. We introduce ***Area Under the Curve ratio for accuracy (AUC$_{acc}$)***, that can visually be represented by the ratio of two areas under the curve in Figure 2.

$$AUC_{acc} = \frac{\sum_{i=0}^{T} acc_i \times w_i}{\sum_{i=0}^{T} acc_i^{joint}} = \frac{AUC_{acc(CIL)}}{AUC_{acc(CJT)}} \quad (2)$$

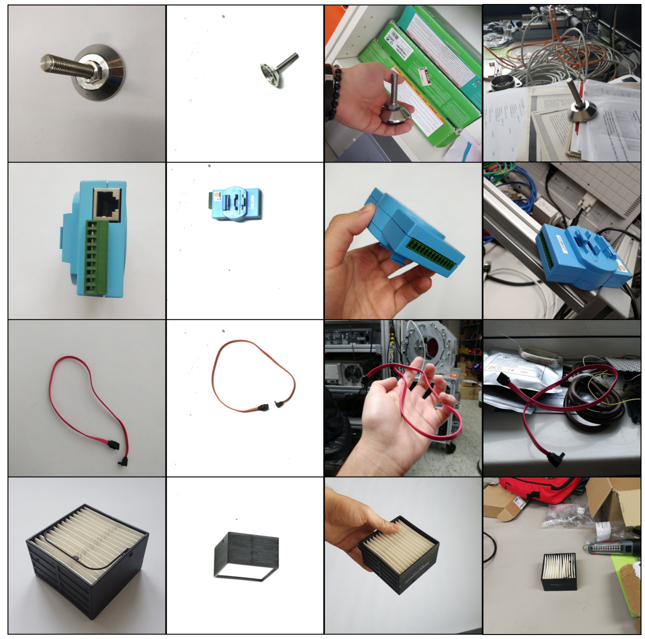
Similarly, we propose ***AUC energy ratio (AUC$_e$)***, which gives the energy consumption of CIL to that of CJT.

$$AUC_e = \frac{E_{CIL}}{E_{CJT}} = \frac{AUC_{e(CIL)}}{AUC_{e(CJT)}} \quad (3)$$

These metrics have the added benefit of applicability in broad-term continual learning frameworks, where the model is periodically *reset* (w.r.t. model size and architecture) via JT_{update}. In the case of incremental training at constant intervals, $w_i = 1$ and AUC_{acc} equals the average accuracy. It is evident from Figure 2 that implementations such as DER would perform better during long sequences on large datasets when a joint training update is made, which reduces the model size and complexity.

3.2 InVar-100 Dataset

The Industrial Objects in Varied Contexts (InVar) Dataset was internally produced by our team and contains 100 objects in 20800 total images (208 images per class). The objects consist of common automotive, machine and robotics lab parts. Each class contains 4 sub-categories (52 images each) with different attributes and visual complexities. **White background (D_{wh}):** The object is against a clean white background and the object is clear, centred and in focus. **Stationary Setup (D_{st}):** These images are also taken against a clean background using a stationary camera setup, with uncentered objects at a constant distance. The images have lower DPI resolution with occasional cropping. **Handheld (D_{ha}):** These images are taken with the user holding the objects, with occasional occluding. **Cluttered background (D_{cl}):** These images are taken with the object placed along with other objects from the lab in the background and no occlusion. Table 2 gives details on the subcategories and their visual attributes.



(a) White (b) Stationary (c) Handheld (d) Cluttered

Figure 3: Example of images from the InVar-100 dataset with the subcategories. Further details on the objects, the visual contexts, related metadata (weight, length, breadth, and height of objects, along with the superclass, material, shape, colour and additional descriptors) and a datasheet [24] are available on the online repository.

There are other larger datasets on industrial objects, such as the ABC dataset [37], MECCANO [53] and the MCB project [35]. While other datasets contain a higher number of classes and images, the four subcategories in our dataset simulate the different visual contexts in which industrial objects are generally digitised during inference time. The context of the images changes, but the underlying features of the target object remain constant, making it ideal for our investigation. Datasets such as NICO [31] and NICO++ [68] also present object classes in different visual contexts. However, the industrial objects in our dataset are unlikely to be present in general large pretraining datasets such as ImageNet [15]. The dataset can, thus, serve as a useful downstream dataset for research investigations. Figure 3 shows sample images for the four subcategories.

3.3 Proposed Approach

We address the research gaps (discussed in §2) by performing analyses on the InVar-100 dataset using the RECIL

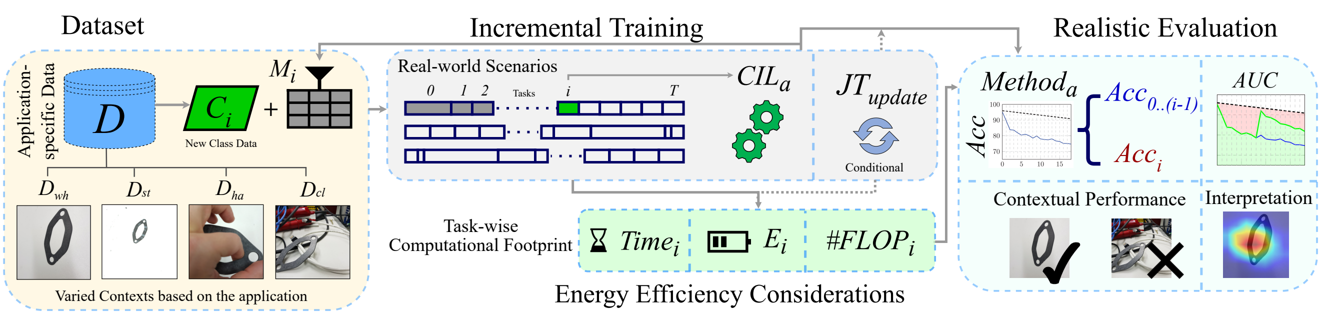


Figure 4: A summary of the RECIL approach. The application-specific data (InVar-100) is used to assess the CIL implementations for different incremental learning scenarios. Task-wise model energy consumption (E_i) and model performance are reported (New classes: Acc_i, Old classes: $Acc_{0..(i-1)}$) for each task i. For long project timelines, AUC metrics are reported. CAMs are studied to understand model plasticity, rigidity and contextual performance.

Attribute	D_{wh}	D_{st}	D_{ha}	D_{cl}
Object is centered	✓	✓*	✗	✗
Object in focus	✓	✓	✗	✗
High Resolution	✓	✗	✓	✓
Cropping	✓*	✓*	✗	✗
Occlusion	✗	✗	✓*	✗
Clutter	✗	✗	✓*	✓
Blur	✗	✗	✓*	✓*

Table 2: Details on the subcategories of the InVar-100 dataset(* means only a fraction of images have the attribute).

(Real-World Scenarios and Energy Efficiency Considerations for Class Incremental Learning) framework, as shown in Figure 4. In order to comprehensively understand a given CIL method (CIL_a), it is tested with varying task increment sizes and sequences ($0..T$). The subsets of data are tested individually for each scenario with differing rehearsal memory buffer (M_i) restrictions. The energy consumption (E_i), training times ($Time_i$), and computational complexity ($FLOP_i$) of the models are monitored for each task i, along with accuracies of old and new classes. Depending on the use case, retraining (JT_{update}) may be introduced after a pre-planned duration or may be triggered when the model performance falls below an established threshold. The context-wise performance of the IL implementations is studied and CAMs are used to interpret the incorrect predictions.

4 Experiments

All the following experiments are conducted on the InVar-100 dataset with different increment sequences and rehearsal memory budgets.

Experiment 1: Constant Increments, 18 IL Tasks, M_i = 2000. We implement the same conditions as those introduced (Figure 1) for ImageNet-Subset (10 classes at Task 0, 5 new classes during each task, 2000 total exemplars as memory M_i for each task) on the InVar-100 dataset. Figure 5 shows the Top-1 accuracy curves as well as average incremental accuracies against the computational complexity and model size. A similar trend to Figure 1 and Table 1 is seen. Figure 6 show a comparison of performance on old and new class data for POD and DER implementations.

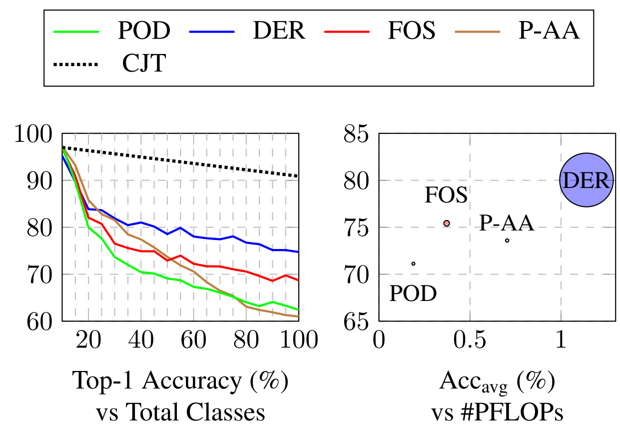


Figure 5: Accuracy and Computational Footprint for Experiment 1 (Constant Increments). The size of the circle is proportional to the model size at the end of the training. The results follow a similar trend as those for ImageNet-Subset shown in Figure 1.

Experiment 2: Comparison of performance for different task sequences, 12 IL Tasks, M_i = 5 per class. We assess the performance of the implementations on the individual subcategories of the InVar-100 dataset on two different randomised task sequences (12 tasks). The overall computational footprint of the sequences remains the same. Each

Method	#Params ↓	#PFLOPs ↓	Sequence 1				Sequence 2			
			D_{wh}	D_{st}	D_{ha}	D_{cl}	D_{wh}	D_{st}	D_{ha}	D_{cl}
Finetuning	11.2M	0.070	22.1%	23.1%	22.9%	21.6%	29.7%	23.2%	25.2%	22.5%
iCARL [55]	11.2M	0.071	49.9%	76.1%	55.9%	50.9%	66.2%	68.1%	54.0%	43.7%
WA [69]	11.2M	0.072	72.1%	51.8%	58.8%	54.4%	67.6%	66.2%	56.4%	47.7%
PODNet [19]	11.7M	0.072	84.9%	70.6%	57.1%	49.7%	**90.4%**	67.6%	61.4%	44.8%
DER [64]	145.7M	0.49	**90.3%**	**81.4%**	**67.5%**	**58.6%**	83.9%	**77.8%**	**61.7%**	**61.5%**
FOSTER [62]	22.6M	0.14	75.2%	54.2%	45.8%	39.7%	74.1%	51.8%	41.3%	35.4%
POD-AANet [43]	13.1M	0.26	82.8%	56.7%	61.1%	52.9%	86.0%	51.8%	35.4%	52.9%
CJT	11.2M	2.41	**98.6%**	**93.1%**	**89.4%**	**88.1%**	**98.6%**	**93.1%**	**89.4%**	**88.1%**

Table 3: Average Incremental Accuracy Results for Experiment 2 comparing the different task sequences. A clear difference in the performance between the two sequences can be observed, even though they have the same number of total tasks and class shuffling order.

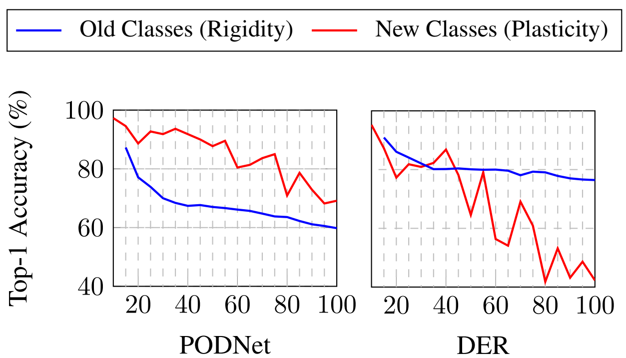


Figure 6: Old vs new class performance with PODNet and DER for Experiment 1 with constant task sizes. This provides a clearer understanding of the plasticity and rigidity of the method.

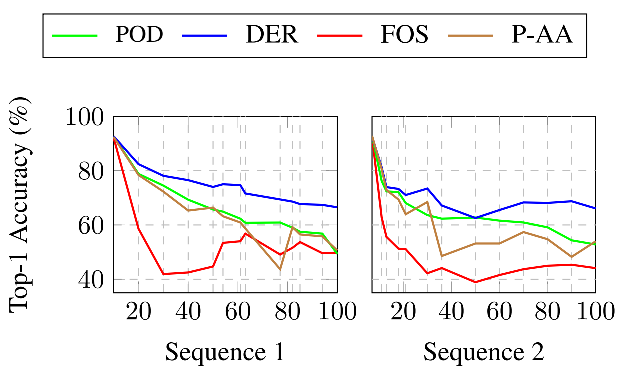


Figure 7: Results for the two task sequences in Experiment 2, averaged across the four subcategories for each method. Sudden drop and rise in the accuracy can be seen, which was not encountered during Experiment 1.

subcategory is trained individually with a rehearsal memory limit of 5 exemplars per class for each class. The results are summarised in Table 3 for different CIL implementations. The averaged results across the four subcategories are shown in Figure 7. A performance shift can be seen for FOS and P-AA, which was not previously observed with constant task sizes and higher per-class rehearsal memory for the earlier tasks. Moreover, a clear striation can be seen w.r.t. the accuracies on the different subcategories.

Metric	POD	DER	FOS	P-AA	CJT
#PFLOPs ↓	0.39	2.81	0.76	1.44	2.87
Time (h) ↓	96.9	110.1	111.4	188.6	138.2
E (kWh) ↓	11.7	15.0	13.4	23.2	17.5
Acc_{last} (%) ↑	82.89	80.29	84.04	81.37	90.87
Acc_{avg} (%) ↑	88.65	86.65	90.45	87.06	93.94

Table 4: Results for Experiment 3 (varying task sequence and increased rehearsal memory) w.r.t. accuracy, energy consumption and computational footprint. ↑ indicates higher value is better, ↓ indicates lower value is better.

Experiment 3: Comparison of different subcategories with varying task sequences, 13 IL Tasks, M_i = 2000 per subcategory for all tasks. A different variable increment sequence (20 classes at Task 0 and 13 new tasks) is introduced. The rehearsal memory limit is increased to 2000 per subcategory (8000 for the entire dataset), implying that the initial increments (shaded regions in Figures 8, 9 and 10) have access to all old data as exemplars. The performance is significantly better and the results on D_{wh} and D_{st} are consistent through the increments (close to acc_{joint}). However, the increased memory limit per task results in greater energy consumption and complexity, as summarised in Table 4. Figure 10 shows the performance for DER and POD on old and new classes for the D_{wh} and D_{cl} data. It can be seen that the performance of POD on old classes can be improved using a higher exemplar memory. However, the performance on new classes for D_{cl} worsens during incremental training. This drop in performance on new data is much worse for DER.

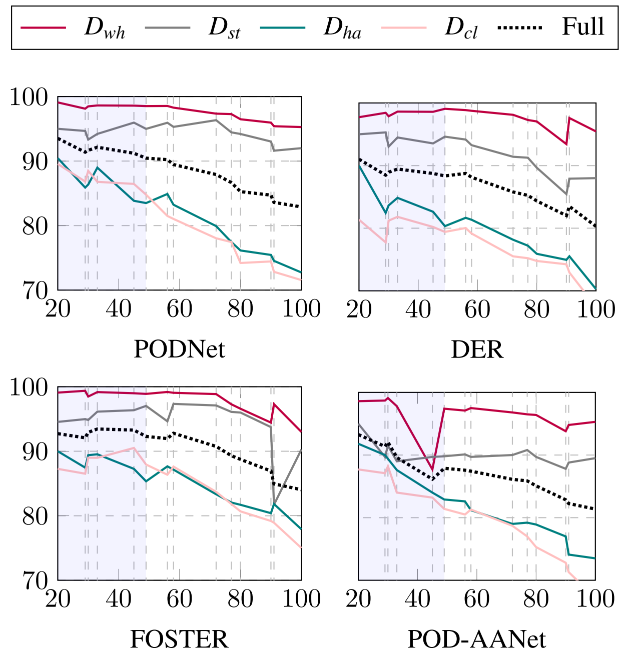


Figure 8: Top-1 Accuracy vs Total Classes for Experiment 3 (Table 4) with increased rehearsal memory budget. Earlier increments (shaded region) have access to all old data as exemplars. The *Handheld* and *Cluttered* subcategories experience a greater drop in accuracy as more tasks are introduced.

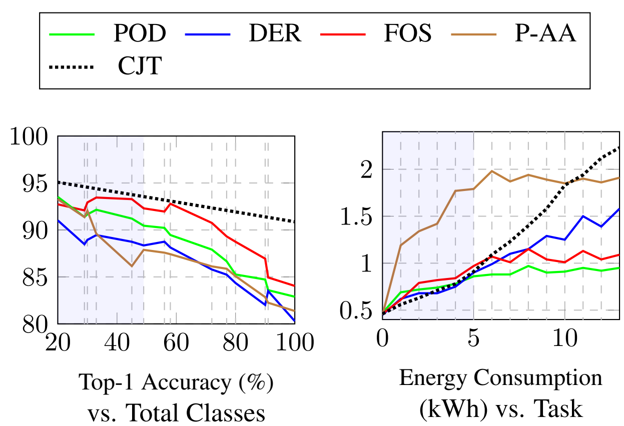


Figure 9: Accuracy and Energy Consumption for Experiment 3. Earlier increments (shaded region) have access to all old data as exemplars. As a result, the energy consumption increases more rapidly.

Experiment 4: Study of long increment sequences and Joint Training Update, M_i = 10 per class. We conduct a 6-month (26-week) long study to analyse the performance of the CIL implementations over a long timeline. At the 3-month mark, we introduce JT_{update}. Table 5 summarises the results, and an increase in AUC_{acc} can be seen with the update for each implementation. Figure 11 visualises these results and puts the average incremental accuracy in relation to the computational footprint for the study.

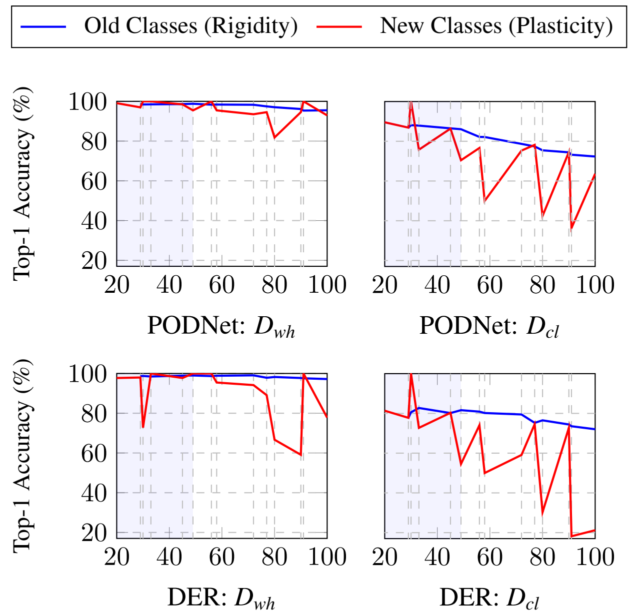


Figure 10: Top-1 Accuracy on *old* and *new* classes for Experiment 3 for *white background* and *cluttered* subcategories for POD and DER. The trends are generally similar to those seen in Figure 6, with increased fluctuation for new class.

5 Discussion

Incremental Accuracy. The Experiments provide several insights that are not gleaned from standard benchmark tests. Based on the results, we roughly classify the CIL implementations as *plastic* and *rigid*. *Plastic* implementations perform better on newer classes compared to the older classes and require a larger memory buffer to mitigate forgetting of old data. In contrast, *rigid* methods are better at maintaining learning from old data, but struggle to learn from new data. They also need lesser exemplars to maintain adequate performance on old data. PODNet and POD-AANet are generally more plastic, and DER and FOSTER are more rigid. The performance of each implementation worsens as more tasks are introduced, especially for data with clutter and occlusion. The performance of *plastic* methods on old data can be boosted using a higher rehearsal memory limit, however, the performance of *rigid* implementations on new data cannot be improved and JT_{update} would be necessary. One exception to this classification is FOSTER, which performs well on Experiments 1, 3 and 4.

Metric	PODNet [19]		DER [64]		FOSTER [62]		POD-AANet [43]		CJT
	IL	w/JT	IL	w/JT	IL	w/JT	IL	w/JT	
#Params ↓	11.7M	11.7M	291.9M	157.2M	22.6M	22.6M	13.1M	13.1M	11.2M
#PFLOPs ↓	0.11E15	0.14E15	1.61E15	1.22E15	0.21E15	0.27E15	0.39E15	0.51E15	8.11E15
$Acc_{avg\text{-}Top:5}$ ↑	77.24%	83.15%	90.22%	94.33%	84.45%	88.12%	80.5%	90.7%	99.7%
AUC_{acc} ↑	0.58	0.67	0.72	0.80	0.62	0.69	0.64	0.78	1

Table 5: Results for Experiment 4 (six-month study) with and without JT_{update}. The performance of the approaches improves after retraining, while the overall increase in the computational footprint is significantly lower than CJT.

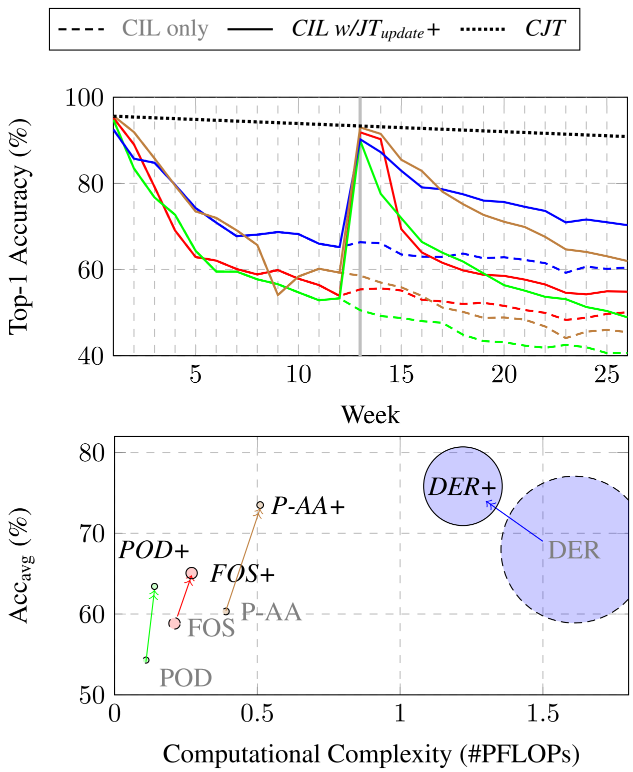


Figure 11: Results for Experiment 4 (six-month study). The radius corresponds to the model size at the end of training. The effect of JT_{update} on the accuracy and computational footprint for the different methods is shown.

However, it underperforms in Experiment 2, especially on D_{ha} and D_{cl}. We hypothesise this is due to fewer classes in the initial increments.

Computational Footprint. Analysis of the computational load of the implementation is necessary to compare IL approaches. For instance, Experiments 3 and 4 introduce different approaches to improve model accuracy (increasing the rehearsal memory limit and introducing JT_{update}, respectively). Based on the computational complexity, the approach taken in Experiment 4 is optimal. With JT_{update}, the DER implementation significantly improves w.r.t. accuracy and computational complexity. PODNet, FOSTER AND P-AA achieve a higher AUC_{acc} score with an increase in the computational load. PODNet w/JT_{update} has higher AUC_{acc} and lower complexity than FOSTER and P-AA with only IL. Tracking the task-wise energy consumption is more accurate and provides a detailed understanding of the computational footprint. From Tables 1 and 4, we observe that the total time consumption for an implementation correlates with the energy consumption, given that all the other variables are controlled. In the absence of such a setup, however, it is recommended to report the model sizes and #FLOPs alongside accuracy results, as shown in Figures 5 and 11. This can be readily done with no added planning or setup.

Interpretation. We study the CAMs to interpret the incorrect model predictions for the CIL methods. Figure 12 shows the effects of *plastic* CIL implementations. The model predicts newer classes with greater confidence. Figure 13 demonstrates the opposite scenarios for the relatively *rigid* DER and FOSTER implementations. We observe that older classes are predicted with higher confidence. In the case of clutter, occlusion or blur, the issue worsens, as models falsely tend to overfit the features of background objects that may be similar to new class data. The InVar-100 subcategories help highlight these issues which a carefully curated clean industrial dataset would not. W.r.t. visual context, clutter has the most impact on IL performance due to false class activations. Occlusion by hand and cropping tends to be less impactful in industrial contexts, provided that the target object features are sufficiently captured.

Project Applicability. Addressing the research questions in §2, we see that the maximum possible IL-Tasks T that can be performed in a continual project depends on the requirements, the quality of data and its availability. For instance, a *plastic* implementation may be a better option for projects where little data is available at the beginning of the project. Implementation of regular JT_{update} similar to Experiment 4 is efficacious w.r.t. performance and computational footprint trade-off. This trade-off generally depends on the dataset, IL methods, and setup. The rehearsal memory re-

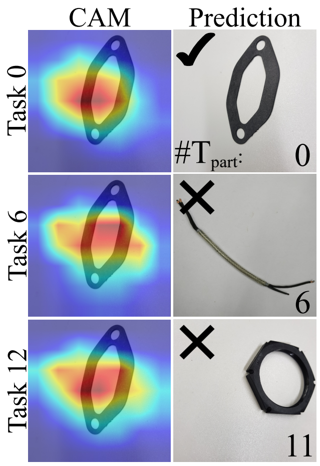
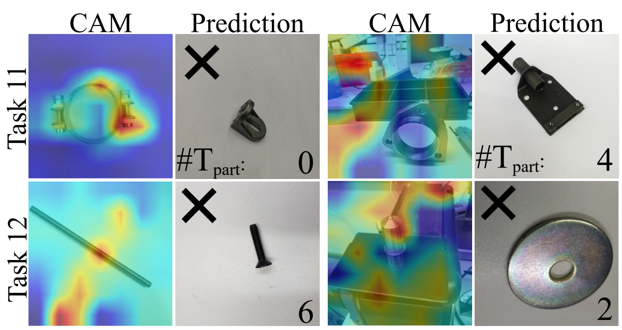


Figure 12: CAMs and Predictions for incremental training with PODNet [19]. The original object is introduced during Task 0. $\#T_{part}$ gives the Task during which the predicted object was introduced. Class activation region remains relatively constant, however, newer classes with similar features are predicted with a higher probability.



(a) DER (b) FOSTER

Figure 13: Example objects, corresponding CAMs and Predictions for incremental training with (a) DER [64] and (b) FOSTER [62]. $\#T_{part}$ gives the Task during which the predicted object was introduced. CAM activations show that learnt features from the new classes overfit the older classes. This is further exacerbated for data with occlusion and clutter.

quirements and optimal retrain period would be subject to the IL approach and the visual contexts in the data. Thus, accuracy gain from higher memory limits is also context and data-dependent (Table 3, Figure 8). However, the Experiments show that CIL approaches can be implemented for practical use cases and offer a significant reduction in overall training time and computational footprint compared to retraining the model from scratch (CJT).

Limitations. Introducing real-world scenarios for continual learning makes it challenging to directly compare and benchmark different approaches. Energy Consumption for ML training can be difficult to study in cloud and shared computing environments. The Experiments (§4) primarily focus on the four CIL implementations. We considered including additional methods for all experiments and found them redundant for the core focus of this paper. The selected IL approaches and the chosen set of experiments cover diverse scenarios (w.r.t. computational load, energy-accuracy trade-off) and setups (memory budgets, increment sequences, retraining), but are not exhaustive. Our paper introduces a generalised approach, which can be expanded and applied to more use cases.

6 Conclusion

We introduced RECIL, a more realistic approach for evaluating Incremental Learning methods, especially for industrial use cases. The InVar-100 dataset is one of the core contributions of this paper, which can also be used for other general Computer Vision research. The experiments demonstrate that the computational footprint is a crucial metric for assessing and comparing different implementations. Putting the incremental accuracy in relation to the energy consumption, training times or computational complexity provides a fair and comprehensive comparison between IL approaches and can be done with no added planning (#FLOPs) or minimal setup (measuring the energy consumption with an energy metering device). Additionally, we identify and address research gaps between current IL research and its practical implementation. We note that performance on standardised benchmarks on well-curated data does not transfer to practical use cases. An emphasis on Green AI is essential for a sustainable, broad-scale adoption of IL research in real-world applications with long timelines. We encourage the IL community to adopt these practices to increase trust and understandability in their work.

Acknowledgments

This work is funded by the German Federal Ministry of Education and Research (BMBF) within the EIBA project 033R226 in the ReziProK program over the FONA platform for sustainable research. We thank Jan Lehr for the data collection and curation. We are grateful to Prof. Dr.-Ing. Jörg Krüger and Dipl.-Ing. Johannes Hügle for their guidance and leadership. We also want to acknowledge the invaluable contributions of the Computer Vision community. The shared resources, tools, and frameworks played an instrumental role in advancing our work and fostering a culture of knowledge exchange.

References

[1] Ann-Louise Andersen, Kjeld Nielsen, and Thomas Ditlev Brunoe. Prerequisites and barriers for the development of reconfigurable manufacturing systems for high speed ramp-up. *Procedia Cirp*, 51:7–12, 2016. 2

[2] Chunguang Bai and Joseph Sarkis. Flexibility in reverse logistics: a framework and evaluation approach. *Journal of Cleaner Production*, 47:306–318, 2013. 1, 2

[3] Eden Belouadah and Adrian Popescu. IL2M: class incremental learning with dual memory. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 583–592. IEEE, 2019. 2, 3

[4] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. Initial classifier weights replay for memoryless class incremental learning. In *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020. 2

[5] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021. 2

[6] Massimo Bertolini, Davide Mezzogori, Mattia Neroni, and Francesco Zammori. Machine learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications*, 175:114820, 2021. 1, 2

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 2

[8] Andrés Bustillo, Roberto Reis, Alisson Machado, and Danil Pimenov. Improving the accuracy of machine-learning models with data from machine test repetitions. *Journal of Intelligent Manufacturing*, Article in press:1–19, 01 2022. 2

[9] Toly Chen, Yi-Chi Wang, and Horng-Ren Tsai. Lot cycle time prediction in a ramping-up semiconductor manufacturing factory with a som–fbpn-ensemble approach with multiple buckets and partial normalization. *International Journal of Advanced Manufacturing Technology*, 42, 2009. 2

[10] Zhiyuan Chen, Bing Liu, Ronald Brachman, Peter Stone, and Francesca Rossi. *Lifelong Machine Learning*. Morgan & Claypool Publishers, 2nd edition, 2018. 2

[11] Gopinath Chennupati, Milind Rao, Gurpreet Chadha, Aaron Eakin, Anirudh Raju, Gautam Tiwari, Anit Kumar Sahu, Ariya Rastrow, Jasha Droppo, Andy Oberlin, Buddha Nandanoor, Prahalad Venkataramanan, Zheng Wu, and Pankaj Sitpure. Ilasr: privacy-preserving incremental learning for automatic speech recognition at production scale. In *KDD 2022*, 2022. 3

[12] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2534–2543, 2021. 2

[13] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *arxiv:2204.02311*, 2022. 2

[14] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *ICML*, 2023. 2

[15] 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, 2009. 1, 3, 4

[16] Jieren Deng, Jianhua Hu, Haojian Zhang, and Yunkuan Wang. Incremental prototype tuning for class incremental learning, 2022. 2

[17] Uwe Dombrowski, Jonas Wullbrandt, and Philipp Krenkel. ”industrie 4.0 in production ramp-up management”. *Procedia Manufacturing*, 17:1015–1022, 2018. 28th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2018), June 11-14, 2018, Columbus, OH, USAGlobal Integration of Intelligent Manufacturing and Smart Industry for Good of Humanity. 1

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 2

[19] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Confer-*

ence, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX, volume 12365 of *Lecture Notes in Computer Science*, pages 86–102. Springer, 2020. 2, 3, 6, 8, 9

[20] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[21] Robert French. Catastrophic interference in connectionist networks: Can it be predicted, can it be prevented? In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1993. 1

[22] Eva García-Martín, Niklas Lavesson, Håkan Grahn, Emiliano Casalicchio, and Veselka Boeva. How to measure energy consumption in machine learning algorithms. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*, pages 243–255. Springer, 2019. 3

[23] Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134:75–88, 2019. 3

[24] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Commun. ACM*, 64(12):86–92, nov 2021. 4

[25] Alexander Gepperth and Barbara Hammer. Incremental learning algorithms and applications. In *European symposium on artificial neural networks (ESANN)*, 2016. 2

[26] Christophe Giraud-Carrier. A note on the utility of incremental learning. *Ai Communications*, 13(4):215–223, 2000. 2

[27] Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 1

[28] Bertha Guijarro-Berdiñas, Santiago Fernandez-Lorenzo, Noelia Sánchez-Maroño, and Oscar Fontenla-Romero. A privacy-preserving distributed and incremental learning method for intrusion detection. In *ICANN (1)*, pages 415–421, 2010. 3

[29] Martin Haller, Andreas Peikert, and Josef Thoma. Cycle time management during production ramp-up. *Robotics and Computer-Integrated Manufacturing*, 19(1-2):183–188, 2003. 1

[30] 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. 1, 3

[31] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110:107383, 2021. 4

[32] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2

[33] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3

[34] Ahmet Iscen, Thomas Bird, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. A memory transformer network for incremental learning. In *British Machine Vision Conference*, 2022. 2

[35] Sangpil Kim, Hyung-gun Chi, Xiao Hu, Qixing Huang, and Karthik Ramani. A large-scale annotated mechanical components benchmark for classification and retrieval tasks with deep neural networks. In *Proceedings of 16th European Conference on Computer Vision (ECCV)*, 2020. 4

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

[37] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[38] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 3

[39] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *CoRR*, abs/1910.09700, 2019. 2

[40] Loïc Lannelongue, Jason Grealey, and Michael Inouye. Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8, 2020. 2

[41] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[42] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2544–2553, 2020. 2, 3

[43] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Rmm: Reinforced memory management for class-incremental learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3478–3490. Curran Associates, Inc., 2021. 2, 3, 6, 8

[44] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental

learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020. 2

[45] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[46] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7765–7773. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[47] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 3

[48] Daniele Mazzei and Reshawn Ramjattan. Machine learning for industry 4.0: A systematic review using deep learning-based topic modelling. *Sensors*, 22(22), 2022. 1

[49] Sudhanshu Mittal, Silvio Galesso, and Thomas Brox. Essentials for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3513–3522, 2021. 2

[50] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[51] 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 (WACV)*, pages 3911–3920, January 2023. 2, 3

[52] Gheorghe H Popescu, Sonia Petreanu, Bogdan Alexandru, and Horea Corpodean. Internet of things-based real-time production logistics, cyber-physical process monitoring systems, and industrial artificial intelligence in sustainable smart manufacturing. *Journal of Self-Governance & Management Economics*, 9(2), 2021. 1

[53] Francesco Ragusa, Antonino Furnari, Salvatore Livatino, and Giovanni Maria Farinella. The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1569–1578, January 2021. 4

[54] Rahul Rai, Manoj Kumar Tiwari, Dmitry Ivanov, and Alexandre Dolgui. Machine learning in manufacturing and industry 4.0 applications, 2021. 1, 2

[55] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2, 6

[56] Allterco Robotics. Shelly plug. `https://shelly-api-docs.shelly.cloud/`, accessed 2022. 3

[57] Marian Schlüter, Hannah Lickert, Katharina Schweitzer, Pinar Bilge, Clemens Briese, Franz Dietrich, and Jörg Krüger. Ai-enhanced identification, inspection and sorting for reverse logistics in remanufacturing. *Procedia CIRP*, 98:300–305, 2021. The 28th CIRP Conference on Life Cycle Engineering, March 10 – 12, 2021, Jaipur, India. 1, 2

[58] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Commun. ACM*, 63(12):54–63, nov 2020. 2, 3

[59] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192, 2020. 2

[60] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4:1185–1197, 2022. 2

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[62] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pages 398–414. Springer, 2022. 2, 3, 6, 8, 9

[63] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[64] S. Yan, J. Xie, and X. He. Der: Dynamically expandable representation for class incremental learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3013–3022, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society. 2, 3, 6, 8, 9

[65] Tan Yigitcanlar, Rashid Mehmood, and Juan M Corchado. Green artificial intelligence: Towards an efficient, sustainable and equitable technology for smart cities and futures. *Sustainability*, 13(16):8952, 2021. 3

[66] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR, 06–11 Aug 2017. 2

[67] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, An-

jali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. 2

[68] Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16036–16047, June 2023. 4

[69] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 6

[70] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2

[71] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *CVPR*, 2022. 2

[72] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, and De-Chuan Zhan. Pycil: a python toolbox for class-incremental learning. *SCIENCE CHINA Information Sciences*, 66(9):197101–, 2023. 2, 3

[73] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey, 2023. 2

[74] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022. 2

[75] Huiping Zhuang, Zhenyu Weng, Hongxin Wei, Renchunzi Xie, Kar-Ann Toh, and Zhiping Lin. ACIL: Analytic class-incremental learning with absolute memorization and privacy protection. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 3