Continual Learning Techniques for Image Classification

Student Name: Finlay Boyle
Supervisor Name: Dr. Donald Sturgeon
Submitted as part of the degree of MSci Mathematics and Computer Science to the Board of Examiners in the Department of Computer Sciences, Durham University

Abstract—Updating a neural network in response to new data requires retraining the model from scratch. This is an expensive and time-consuming process, but it is necessary to avoid the complete erase of existing knowledge due to a phenomenon known as catastrophic forgetting. Continual learning focuses on alternative training methods to update the knowledge in a neural network using unseen data without needing to retrain the neural network completely. I present a novel continual learning technique that quantifies the uncertainty of incoming data to carefully construct and update a memory buffer of previously seen samples which are used to update the neural network in combination with a pre-trained vision transformer to alleviate catastrophic forgetting. I compare a variety of existing state-of-the-art techniques with the proposed approach in a challenging experimental setup where samples may only be seen once, known as the class-incremental online setting, using a variety of datasets and maximum memory buffer sizes. The novel technique achieves strong results and outperforms recent state-of-the-art techniques in many aspects while successfully reducing the effects of catastrophic forgetting.

Index Terms—Computer vision, machine learning, performance evaluation

1 Introduction

The standard method to train a neural network on a particular dataset of images is to divide the dataset into batches, and repeatedly optimise the parameters of the model via backpropagation by computing the value of a loss function on each batch of data. This process is repeated many times over a series of epochs to improve the parameters. At each epoch, the model is exposed to all of the data again. Training a model in such a way is known as offline training.

Offline training is capable of producing high quality results assuming that the training process has sufficient resources, such as time, computing power, and a large, representative dataset to train the model on. This is illustrated in Figure 1, the chart shows the results of training a ResNet-18 [1] model using the standard offline training approach for 100 epochs on the CIFAR-10 dataset [2]. A random sample of images from the dataset can be seen in Figure 2. The model achieves convincing results on each of the classes individually and it attains an overall accuracy of 90%.

Despite achieving high quality results, offline training does have some limitations such as difficulties explaining how weights and biases occurred giving the perception of a black box [3], susceptibility to data biases [4], and the inability to learn continuously [5]. This final limitation is significant due to the massive amounts of data generated daily. One example of this is the $3.5 billion worth of sales on Amazon's Prime Day in 2020 by third-party sellers alone [6], this equates to an enormous amount of data to analyse. Another example is the amount of data uploaded to YouTube, in 2020 it was estimated that 500 hours of video were uploaded every second [7]. This suggests that the data distributions of real life are dynamic rather than stationary providing clear use cases for machine learning to reduce immense amounts of data into human digestible formats. However, to do this models need to be able to adapt to these distributions but offline training is not capable of facilitating this without retraining the models from scratch.

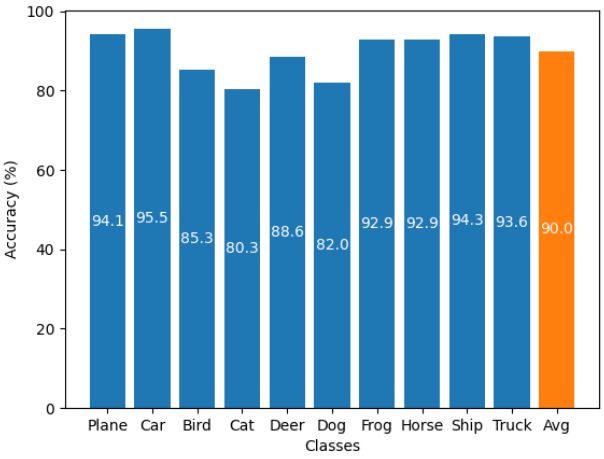


Fig. 1: A chart showing the classification accuracy per class, and the overall accuracy, of a ResNet-18 model trained on CIFAR-10 for 100 epochs.

1.1 Catastrophic Forgetting

Attempting to use offline training to update a previously trained model can have drastic consequences for the knowledge already in the network. Updating the model with unseen data causes the model to forget the previous knowl-

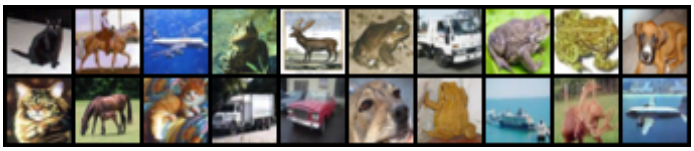


Fig. 2: A random sample of 20 images from the CIFAR-10 dataset. Each individual image is 32x32 pixels.

edge that it has acquired, this is demonstrated in Figure 3 where a ResNet-18 model has again been trained using CIFAR-10, however this time the dataset has been split into 5 distinct tasks with 2 classes each (without overlap). The model is then trained on these tasks sequentially without any exposure to the previous tasks. As each new task is introduced, the model forgets the prior knowledge that is stored in the model by overwriting it. This phenomenon is known as catastrophic forgetting [5].

Training on unseen data is the primary cause of catastrophic forgetting. This is because during offline training the parameters of the model are optimised using gradient descent to update the parameters in the direction that minimises a loss function. Since the previous task's data is no longer available, the model is being trained to minimise the loss on the current task's data only without respect for the previously seen data. The lack of context that the weights and biases of a model represent existing knowledge is a core issue.

1.2 Continual Learning

Neural networks draw inspiration from the brains of animals, typically humans, to model the network based on how neurons interact in the brain [8]. However, implementations of neural networks, and their training procedures, often ignore that humans are capable of learning continually throughout their lifetime.

The study of techniques to alleviate the effects of catastrophic forgetting and enable models to learn continuously is called continual learning [9], also known as incremental learning. Primarily, continual learning has focused on alternative methods to offline training, as such it is typical to fix the model's architecture, however there has been a recent increase in research investigating how the architecture of a model impacts continual learning [10].

The trade-off between retaining knowledge and acquiring new knowledge is rooted in neuroscience. The Stability-Plasticity dilemma describes the contention between these processes in the brain [11]. Stability refers to the ability of the brain to retain existing knowledge over a long period of time whereas plasticity refers to the ability of the brain to learn new information throughout its' lifetime. For humans, the brain initially heavily favours plasticity as we learn new knowledge while growing up. Later in life, the brain shifts towards stability but the ability to learn new knowledge is still present. In artificial neural networks, the learning rate of optimisers partially reflects this trade-off, although this affects the whole network rather than the individual neurons making it difficult to retain knowledge while also maintaining the ability to learn new knowledge. Reducing the learning rate of an optimiser will consolidate existing knowledge but it will simultaneously limit the ability of the model to learn new knowledge, as described in the stability-plasticity dilemma, and overwriting will still occur except at a slower rate.

1.3 Motivation

Continual learning has the potential to offer real-world benefits. Training neural networks is expensive in terms of computational resources, energy usage, and time. As a result, retraining a model from scratch to update the data that it has been on trained on compounds these costs. This is the case if offline training is used. If continual learning is able to achieve competitive performance when compared with offline training then these costs could be greatly reduced by applying these alternative training techniques to update a model rather than training a brand-new model to replace the existing one.

In domains such as image classification, it would be preferable if a model was capable of adapting to changing input data distributions in order to classify new classes of images correctly and identify when it encounters images that are not similar to those that it has been trained to classify previously. This is difficult using offline training, if images from unknown classes are presented then the accuracy of the network would decrease due to misclassification and the false positive rate would increase instead of the model having the potential to adapt. Continual learning has the potential to support this type of behaviour by continuing to train the network on incoming data.

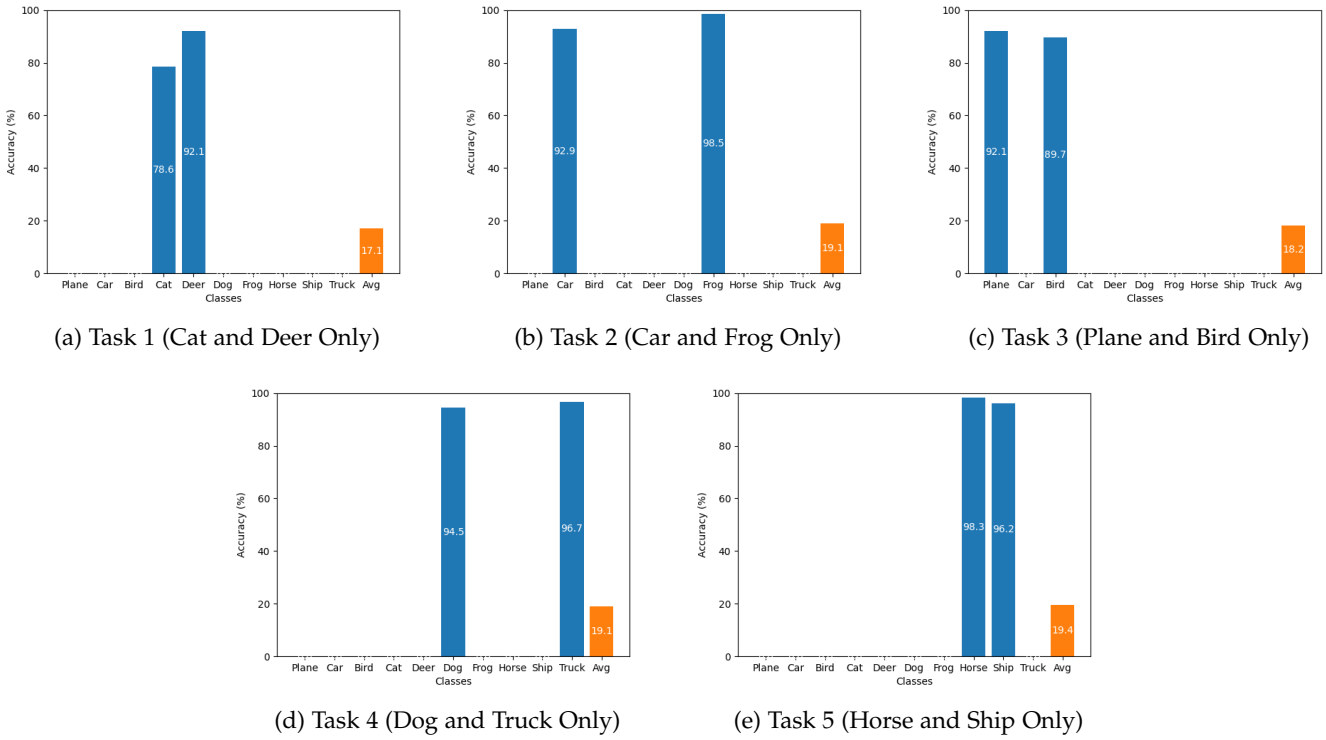
For example, high-quality models such as Microsoft and Nvidia's Megatron-Turing Natural Language Generator [12], cost millions of dollars to train and significant amounts of computer hardware and time [13]. Hence, if high-quality models are to retain their success when new data becomes available or the distribution of the data changes over time, this change in the underlying data needs to be incorporated into the network. Using offline training methods, this would require a complete retraining of the network despite it having already acquired a significant portion of the knowledge. If a full-network retrain can be avoided then this can substantially save on the computational and time costs.

Real-time applications could also greatly benefit from continual learning. Models that are capable of updating on-the-fly for real-time use have the potential to give adopters of them an advantage as well as realising cost savings. This is especially applicable where the real-time data fluctuates periodically such as shopping habits as mentioned previously or other scenarios where the incoming data is unpredictable.

It is also important to note that image classification, and computer vision in general, are not the only domains where continual learning techniques could have an impact. It has also been applied in domains such as Natural Language Processing [14] and Speech Recognition [15]. However, image classification is the primary domain that is used for continual learning research.

1.4 Desirable Properties

The continual learning problem is broad and there are many potentially desirable properties for any solution to have



(a) Task 1 (Cat and Deer Only)

(b) Task 2 (Car and Frog Only)

(c) Task 3 (Plane and Bird Only)

(d) Task 4 (Dog and Truck Only)

(e) Task 5 (Horse and Ship Only)

Fig. 3: Left-to-right, starting on the top row: Each chart shows the classification accuracy of the same ResNet-18 model trained on 5 tasks containing 2 distinct classes each (without overlap) from CIFAR-10 sequentially. They are evaluated on all tasks including those that are unseen. The model suffers from complete catastrophic forgetting.

depending on the use case. The choice of properties can greatly impact the effectiveness of a solution and its applicability to the continual learning problem in real scenarios [16]. In all continual learning setups, it is assumed that the whole dataset is unavailable at training time as this is the fundamental feature of continual learning compared to offline training. The other critical properties that must be considered are:

Task Formulation: A common assumption is to divide the dataset into tasks, this limits the amount of data rather than receiving a continuous stream of data containing potentially unlimited classes. Once the task has been seen, the dataset related to the task is discarded and it is never seen again by the model. For example, this is the scenario used in Figure 3. This can limit the application of techniques, but it is often a useful way to prove that a technique is capable of retaining knowledge even once all samples of a class have been exhausted. The alternative is to have a singular task containing the whole dataset.

Disjoint vs Blurry Tasks: If the setup is following a task-based formulation, there are two possibilities as to how the dataset is divided. One option is complete disjointedness such that the classes in each task are only seen in that particular task and then never again by the model during training. However, this is unlikely to occur naturally. An alternative formulation is to blur the task boundaries such that a class that has already been seen may still appear again later.

Resource Constraints: One of the key benefits of continual learning is reducing the need to retrain models from scratch providing benefits by reducing the long-term costs of training. As such, it is important to constrain techniques such that they are not using unlimited resources. Primarily, this applies to the memory usage of a technique. It is common to limit the usage of memory to either a fixed allocation that is used for the entire training process such that the amount of memory per class is variable (as more classes are introduced, the memory allocation per class decreases) or to assign a fixed amount of memory per class such that the overall memory grows over time. Fixing the total allocation is most realistic as it provides a constant upper bound on the memory usage.

Online vs Offline: A crucial property that severely affects the difficulty of the problem is whether data samples can be seen multiple times during training. In the online case, each sample is only seen once during the training process unless it is stored in a memory buffer. As such, most samples will only be able to contribute to optimising the parameters of the network once. In the offline case, each sample can be used an unlimited number of times without penalty until the task as a whole is discarded. This, in effect, applies offline training to the continual learning problem and it is a severely constrictive property that limits real-world applicability.

Task-Incremental vs Class-Incremental [17]: In the Task-Incremental formulation (Task-IL), data is segmented into predefined tasks with disjoint, limited and known class labels. However, the key distinction from the task formula-

tion property discussed previously is that at inference time the model is informed which task the sample to classify was part of and thus this limits the potential label space which greatly reduces the difficulty. Task-IL is also known as the multi-headed approach. In comparison, the preferred approach is Class-Incremental (Class-IL), at inference time the model is not given any information about which task a sample is from is provided. This is more realistic, and significantly harder, than the Task-IL approach.

The most real-world applicable setup of the continual learning problem is an online non-task-based class-incremental setup with a fixed memory budget for the whole problem. However, this is often the most challenging scenario and it is typical for a task-based class-incremental setup with a fixed memory budget to be used instead.

2 Related Work

2.1 Algorithm Paradigms

Continual learning literature has been heavily focused on the algorithms used to train models rather than the architecture of the models themselves. Within the literature there are three primary paradigms: regularisation, memory-based, and meta-learning. There also exists some approaches that do not fit concretely into these categories as they draw inspiration from outside of the continual learning field such as utilising reinforcement learning techniques [18] and modifying the structure of the network dynamically [19].

2.1.1 Regularisation

Principally, regularisation techniques penalise changes to weights according to some importance factor that is at a per parameter level. The stability-plasticity dilemma [11], as outlined previously, directly relates to regularisation techniques as the penalisation of weight changes reflects the trade-off between preventing knowledge loss (i.e. stability) and changing weights to store new information (i.e. plasticity).

Elastic Weight Consolidation (EWC) [20] proposed using the Fisher information matrix - a measure of the information in a parameter about some unobserved variable [21] - to measure the importance of each parameter with the purpose of reducing changes to weights that are determined to be significant to previous tasks. This importance value limits the plasticity of essential parts of the network to retain acquired knowledge by penalising changes with a quadratic constraint. Likewise, memory aware synapses [22] also uses a quadratic constraint but with the importance being determined by measuring the gradient at each node with respect to specific data points to calculate the value of the nodes for information retention. This technique was influenced by the idea that 'cells that fire together, wire together' from Hebbian Learning [23].

EWC set the initial groundwork for the continual learning field and remains a prominent baseline comparison for proposed solutions. EWC++ [24] proposed a method to compute the Fisher information matrix as a running average instead of a costly computation at the end of the task. This also removed one of the limitations of EWC as the moving average could be computed in an online setting instead of requiring access to the whole dataset to calculate it.

Synaptic Intelligence (SI) [25] and Learning without Forgetting (LwF) [26] consider the effect of the training process on the model's parameters. SI considers the trajectory of the gradient descent and estimates the impact that the change of each specific parameter has on the loss of the model. It then uses this importance to regularise the parameter changes by weighting the mean-squared error between the current parameters and the next parameters. LwF takes a different approach and isolates parts of the network to prevent future tasks impacting the loss of the model with respect to the previous tasks. It does this by maintaining a set of shared parameters that are used by all tasks and introduces distinct parameters for each individual task after the shared parameters.

2.1.2 Memory-based

Memory-based approaches, also referred to as rehearsal-based or episodic approaches in the literature [9], have been the focus of the majority of recent research in the continual learning field. Due to the constraints on the continual learning problem, it is infeasible to store the entirety of the incoming data. However, depending on the setup of the scenario, the sample may potentially never be seen again once it has been discarded, thus determining whether to store a sample can have significant implications. There are a variety of different approaches within the paradigm but the majority of techniques can be characterised by storing samples directly or indirectly such as via feature embeddings or using generative models.

2.1.2.1 Storing Samples: Gradient Episodic Memory (GEM) [27] and Averaged GEM (A-GEM) [28] - a computationally efficient improvement of GEM with a minor accuracy trade-off - are early memory-based techniques that propose optimising the model such that the loss between the model's predictions on the memory samples and the ground truth is monotonically decreasing. This can be viewed as a hybrid approach between retaining samples in memory and regularisation as the constraint of forcing the loss to stay within some bound acts a proxy for regularising the weights of the network and preventing significant changes. A-GEM improves the efficiency by requiring the average loss to be monotonically decreasing instead of for each sample individually. Gradient Sample Selection [29] draws inspiration from GEM. Like A-GEM, it argues that GEM is computationally expensive and reposes the sample selection as a constrained optimisation problem using integer quadratic programming instead to reduce the cost of selecting samples. Furthermore, it also proposes a greedy, less accurate, version that reduces overhead to further extend its applicability.

Riemannian Walk (RWalk) [24] and End-to-End Incremental Learning (E2E) [30] are also in the intersection between regularisation and memory-based approaches. RWalk builds upon EWC++, in terms of regularisation it utilises KL-divergence and the sensitivity of the loss to parameter updates to constrain the weight changes in the network. It proposes combining this approach with a small memory buffer containing samples from previous tasks as it was observed that the regularisation part was ineffective at preventing catastrophic forgetting sufficiently on its own. E2E proposes combining multiple distillation losses [31] and

separate classification heads for each task. These distillation losses constrain the parameters of the network as they are calculated using the previous classification heads that encode information about the previous tasks.

GDumb [16] is a distinguished technique in the continual learning field. The authors of the technique presented a critical view of the direction of continual learning research arguing that much of the literature exaggerated their success by experimenting on simplistic continual learning scenarios - such as the Task-IL setup - and used basic datasets that failed to translate to realistic use cases. To support this, they proposed a technique consisting of a greedy sampler that simply stores the most recent samples with an equal balance from each class seen and then at inference time it trains a classifier on the memory samples only. The technique is capable of outperforming many continual learning specific algorithms in the difficult online Class-IL setup. As a result, it has become an authoritative baseline for serious techniques and it has improved the rigour of techniques in consequence.

ASER [32] and Rainbow [33] focus on improving the diversity of the samples stored in the memory buffer. To achieve this, they compute scores relating to the significance of a sample in protecting the structure of the classes. ASER is grounded in cooperative game theory and is based upon the Shapley Value which can be thought of as measuring the contribution of a sample to the 'game' of retaining existing knowledge. Rainbow differs in its approach to scoring samples, instead it calculates an uncertainty score for each sample by applying a series of augmentations to the image to calculate how uncertain the model is about the class of the image. It then draws samples diversely by ordering them by their uncertainty and drawing uniformly from the set of samples.

Dark Experience Replay (DER) [34] and Mutual Information Maximisation (MIM) [35] both also use image augmentations but unlike Rainbow they use them to diversify the current batch of samples that is being used to train the model rather than using them to identify samples to store in memory. Instead of storing just the samples and the labels, DER also stores the associated logits from the model at the time that the sample was selected. It uses these logits to regularise the model by applying mean-squared error loss to the current model logits compared to the stored logits. An improved approach DER++ [34] also introduces a second regularisation term that applies a cross entropy loss to a second batch of memory samples to constrain the current model's predictions. MIM applies the concept of mutual information, this is designed to quantify the dependence between samples. The authors highlight that this is impractical to calculate and instead utilise InfoNCE loss [36] as a proxy for the mutual information. To do this, they augment the batch of samples and introduce pseudo-classes to represent the combination of augmentations applied to each sample. These are then used to estimate the mutual information via the InfoNCE loss that used to optimise the model to encourage sharing of parameters for mutually dependent samples.

An alternative to directly storing the raw image samples is to train a Generative Adversarial Network (GAN) [37] to act as the memory. This approach was trialled by [38] but due to the notorious instability and difficulty of training GANs using traditional offline training methods, attempting to use GANs in continual learning remains intractable. They require long training times, vast computational resources, and suffer issues with vanishing gradients and mode collapse [39]. As a result, their application in continual learning so far has not exceeded carefully controlled basic examples and it is likely they will remain infeasible for the foreseeable future.

Rather than diversifying the samples stored in memory, Maximally Interfered Retrieval (MIR) [40] proposes drawing samples from the current memory buffer by selecting those that will be most negatively impacted by the next parameter update. These samples are said to be the most interfered with by the model's update and the interference is measured by computing the difference between the loss on the future model (using a sampled batch of data from the data stream) and the loss using the current model. The memory buffer itself is updated randomly. The authors also propose a generative approach, but as discussed previously it is only applicable to carefully controlled experiments on simplistic datasets such as in [38].

As a substitute for storing real samples, feature embeddings can be stored and replayed to the model instead. These can be obtained via pre-trained secondary networks or extracted directly from the classification model by accessing layers from within the network itself. These representations intrinsically encode information about the samples that could be useful in the future.

iCaRL [41], similarly to previously seen real sample methods, has a focus on how to select samples but in iCaRL it stores representations of these samples instead. To select these feature representations, it selects samples that best represent the mean feature representation of the class. To classify samples, it computes the mean feature representation vector for each class using the samples in memory, then it compares the feature representation of the individual sample with each mean vector and the closest one defines the class that it will be classified as, this is known as nearest-class-mean classification [42].

FearNet [43] proposes a triple network structure consisting of two generative networks representing long-term memory and short-term memory, and another network that is used to predict which network contains information about a specific sample or its corresponding class. Inspired by natural behaviour, it includes sleep phases that consolidate samples from short-term to long-term memory with the intention of simulating a human-like brain. To consolidate the data, FearNet uses a generative model since it does not store samples but rather representations of them indirectly in the short and long-term networks.

There are some intermediary approaches between storing features and storing real samples such as REMIND [44] and ACAE-REMIND [45], they save samples using autoencoders that compress them into their feature representations. The difference between the two methods is that REMIND has to freeze the feature extractor after the first task whereas ACAE-REMIND uses an intermediary layer reducing the amount of frozen parameters. The idea is that by reducing the dimensionality of the samples, the memory consumption of the training algorithm can also be reduced

without loss of information. This approach has also been seen in other methods that focus on using generative feature replays such as in [46] which aims to reduce the complexity of generating samples compared to using GANs as was proposed by [38].

2.1.2.2 Alternative Approaches: There are also some techniques that diverge from the norm of storing samples or features but still utilise additional memory as an important part of their training process. Some recent approaches in the literature have combined ideas from outside of the continual learning field.

Learning to Prompt (L2P) [47] draws inspiration from prompting in natural language processing [48]. Prompting is the process of learning small parameters that are appended to feature embeddings to provide task information to a model without requiring explicit task information at test time. L2P proposes learning prompts that are used in conjunction with a pre-trained vision transformer [49] to influence the classification of a sample. This only requires storing a few optimisable prompts instead of many samples in memory but comes with the cost of storing a pre-trained vision transformer.

Contrastive Continual Learning (Co^2L) [50] and Supervised Contrastive Replay (SCR) [51] utilise contrastive representation learning from self-supervised machine learning. Contrastive representation learning attempts to learn a feature space such that similar samples are close to each other and unrelated samples are distant from each other [52]. Both techniques duplicate and augment the data in the batch to diversify the samples available (such as in DER [34] and MIM [35]) and consolidate the structure of the feature space by including perturbed samples while keeping the sample label the same as the original and using them with the Supervised Contrastive Loss objective [53]. Co^2L combines this loss with a secondary regularisation loss between the current embedding space and the previous embedding spaces using instance-wise similarity. SCR only uses the contrastive loss objective but instead of using a traditional softmax classification head it uses nearest-class-mean classification [42] by averaging the embeddings of the memory samples per class as seen previously with iCaRL [41].

2.1.3 Meta-learning

Meta-learning techniques use a dual training procedure consisting of an inner and outer training loop to optimise the learning procedure itself. The inner training loop trains the model to classify data while the outer meta-training loop optimises the training procedure of the inner loop in an alternating manner [54]. To achieve this, these techniques often take a hybrid approach drawing on general themes from the regularisation and memory-based paradigms.

Online aware Meta-Learning (OML) [55] aims to learn sparse representations of the data to generalise the model. It achieves this using two networks: the Representation Learning Network (RLN) is updated in the outer meta-training phase to improve the image representations in the feature embeddings, and the Prediction Learning Network (PLN), that aims to classify samples, is trained in the inner loop. To classify samples, they are passed through the RLN to give a representation of the sample using the learnt model, this representation is then passed into the PLN which predicts the class label.

Similarly, [56] presents a technique that also meta-learns an embedding space for the samples but instead of using a traditional discriminative classifier it proposes to use a generative classifier. Discriminative classifiers predict the class label given the input data whereas generative classifiers attempt to learn how to model the input data given the class label and use this to classify samples by applying Bayesian statistics.

Meta-Experience Replay (MER) [57] and Look-ahead Model-Agnostic Meta-Learning (La-MAML) [58] combine the meta-learning procedure with a memory-based approach by storing samples in a replay buffer. MER aims to disentangle the interference of the network's parameters by computing the dot product between the gradients of different samples to quantify their similarity. It then optimises the model using these dot product values to discourage parameters encoding information about significantly different samples. La-MAML builds upon MER and OML and proposes an online capable meta-learning technique by optimising the same objective as OML but in an online manner. It also combines ideas from regularisation with per-parameter learning rates that are reset on each iteration of the outer meta-learning loop.

Mnemonics [59], Hindsight Anchoring [60], and Dataset Distillation using Neural Feature Regression [61] also combine meta-learning with the memory-based paradigm by treating the samples stored in memory as optimisable parameters, as such the initial sampling strategy is less relevant for these approaches. These techniques are intended for the offline setting in order to support optimising the memory samples in the outer loop to act as synthetic data points encoding more information about the class than any individual sample. The motivation behind treating samples as optimisable parameters is to increase the value of the samples that are replayed to the model during training while minimising the required storage for the samples. Ideally, these optimised samples will act as anchors that reflect the structure of the class and its boundary to reduce misclassification.

MERLIN [62] takes a different perspective by proposing that the parameters of a network can be sampled from a meta-distribution that is learnt during the meta-learning process. During the outer training loop it optimises the parameter meta-distribution using a variational autoencoder [63]. Parameters are then drawn from this distribution multiple times to create an ensemble of models to reduce the inherent uncertainty caused by sampling from a randomised distribution.

Another drastically different meta-learning approach is an alternative to back-propagation itself by instead using feedback and local plasticity (FLP) [64]. The proposed technique predicts the class of a sample and then propagates the error in the prediction to earlier layers of the network via feedback connections and intrinsically encodes parameter plasticity using Oja's learning rule [65], again drawing on ideas from Hebbian Learning [23]. It utilises meta-learning to learn the initialisations of the weights and biases as well as the plasticity of the parameters. The concept of an alternative to back-propagation to mitigate the catastrophic

forgetting problem has also been suggested as a potential avenue for future exploration outside of a meta-learning context [66].

2.1.4 Miscellaneous

Beyond the paradigms outlined so far there exists some unorthodox techniques that present different angles to approach the continual learning process from.

Laplace operator based node-importance dynamic architecture (LNIDA) [19] and Reinforced Continual Learning (RCL) [18] aim to adapt the structure of the network during training time. LNIDA uses an approach that reflects regularisation to evaluate the importance of the nodes in a network. It quantifies this by applying the Laplacian to the loss computed at each specific node. After a fixed number of epochs, this importance value is used to remove interfering connections and reinitialise insignificant nodes. RCL uses reinforcement learning to adapt the architecture of the network each time a new task is introduced. It proposes to use three separate networks: one to generate the policies, another to estimate network value, and a task network that is to be used for classification. The task network is optimised via an actor-critic strategy with the accuracy and complexity of the network determining the reward.

LUCIR [67] consists of three main components: cosine normalisation for the probabilities in the final layer of classification (instead of the traditional softmax layer), the 'less-forget constraint' which is a regularisation constraint that considers the position of previously acquired knowledge relative to the new data in an embedding space, and inter-class separation which is a ranking loss that compares previous samples, used as anchoring points, against the new samples to attempt to separate out classes.

Drift compensation is a technique that utilises embedding networks. These are networks that are able to map data into a lower dimension where simple metrics such as L2-norm can be used to compute similarities between embedding representations [68]. Instead of preventing the drift of classes, where they move around in the embedding space, this method aims to compensate for the drift by estimating how much each class has drifted and accounting for it during classification.

2.1.5 Summary

Ultimately, there has been significant progress in the development of techniques to overcome and mitigate the impact of catastrophic forgetting. Regularisation approaches are the least effective [69] and have become obsolete when they are used on their own due to issues with scaling and their lack of applicability to realistic setups such as online Class-IL continual learning. As such, they are rarely seen in the recent literature unless they are used in conjunction with a different paradigm or acting as a baseline minimum result.

The memory-based paradigm is varied and includes some of the most promising continual learning techniques, especially for the online setting [70]. This paradigm continues to be the primary focus of research in the field and it has been suggested that memory-based techniques may be essential to overcoming the catastrophic forgetting problem [71]. These techniques do come with additional considerations such as their memory usage to ensure that they conform with the goals of continual learning which may limit their applicability in some circumstances however.

Meta-learning techniques offer greater potential in the offline setup of the continual learning problem to fully utilise their dual-phase training setup. Nonetheless, this class of techniques has shown promise and commonly combines aspects from across the other continual learning paradigms and the wider body of meta-learning literature across machine learning in general.

Finally, techniques outside of the main paradigms may offer insight into future approaches for continual learning and offer an opportunity to experiment with potentially unique solutions to the catastrophic forgetting problem.

2.2 Architecture Considerations

Continual learning literature has primarily focused on the algorithms used to train a model to prevent catastrophic forgetting and the architecture of the models being trained is sidelined. Typically, ResNet [1], convolutional neural networks, or vision transformers [49] are used as the underlying model for these training algorithms.

A comprehensive study of how the width and depth of a neural network affects catastrophic forgetting was recently published [72]. It found that two networks based on the same architecture but one with the parameters providing width to the model significantly outperformed the same architecture with the parameters providing depth to the model in terms of both increasing accuracy and decreasing forgetting. This raises interesting questions about the effect of the width of a neural network and provides a potential avenue for future exploration.

A further study was also conducted by the same authors that measured the impact of different architecture types and the impact of specific layers within a network used for continual learning [10]. It was found that ResNet had greater capability to learn new tasks whereas convolutional neural networks and vision transformers were better at retaining information. They also found that the impact of batch normalisation layers was dependent on the data distribution, if it was relatively stationary then they were beneficial, otherwise they were detrimental. In addition to this, it was clear that global pooling layers negatively impacted the performance as they narrow the network which further confirmed the results of [72] whereas max pooling layers improved the performance because they did not narrow the network. Importantly, the authors noted that using a high quality architecture can be as impactful as a high quality training algorithm but the best performance can be achieved by using both.

2.3 Benchmarking

It is important to be able to compare methods described in the literature in order to evaluate their success in the continual learning domain. Early literature was plagued with issues regarding metrics. Many techniques were not compared against the state of the art or created their own metrics in order to quantify the performance of their algorithms [9].

Gradually, key metrics have been identified to compare different algorithms fairly. For example, the average forgetting rate, which is unique to online training methods, measures the drop in performance of the network caused by learning new tasks. Forgetting is measured as the difference between the maximum knowledge about a specific task seen so far and the current knowledge of the model on the same task [24]. Intransigence is another important metric that was also introduced in [24] that quantifies the inability of a model to learn by comparing it against an offline trained version.

Another important metric is the overall accuracy, this is typically measured on a per class or per task basis as well as at the end of the whole training process which is standard with respect to training neural networks. In continual learning it is common to measure the accuracy throughout the training process although this is only relevant in the disjoint task formulation to evaluate the impact of the new distinct tasks on the accuracy.

It is also crucial to consider the decay of knowledge as more tasks are learnt. To measure this, [27] proposes forwards and backwards transfer. Backwards transfer measures the influence of the current task on the model's knowledge of previous tasks by measuring the mean difference between the knowledge known prior to training on the current task compared to the model's knowledge of the previous tasks after training on the current task. Forwards transfer is similar except it measures how much influence the current task has on future tasks in a similar way.

In addition to the accuracy and forgetting metrics, for comparison of continual learning algorithms there are other important considerations. As many of the previously covered methods have a focus on storing samples, or representations thereof, it is important to consider other factors in the practicality of these approaches. Other important metrics in the literature are computational efficiency, memory consumption, and wall-clock time (how long it actually takes to run the training and inference procedures) [16]. These must be considered when comparing methods because a method may have exceptional classification performance but require an unrealistic amount of memory or computation time to achieve these results. This risks nullifying the benefits of continual learning and thus it requires significant consideration.

2.4 Datasets

Datasets are important to assess how well the continual learning algorithms perform on increasingly complex tasks as well as ensuring consistency during the comparison. Early literature typically focused on using simple datasets such as CIFAR-10 [2] and MNIST [73] which are useful as basic examples but do not reflect the complexity of data from real scenarios and thus are not true indicators of how these algorithms would perform in real environments [16]. As the literature has progressed, it is clear that the trend is to use more complicated datasets to better represent and compare algorithms.

CIFAR-100 is a prominent dataset in general image classification literature, it is an expanded version of CIFAR-10 and contains 100 classes (which are classified into 20 higher-level superclasses) of images each containing 600 images [2]. It commonly features in the continual learning literature as it is sufficiently complex to identify any issues with the scaling of approaches as well as providing many classes to identify if techniques breakdown as they are exposed to increasingly many classes.

ImageNet (and Mini-ImageNet), which has been used for visual recognition challenges, is a dataset containing 1000 classes with over 1.3 million images [74] making it the most challenging of the datasets but also the most realistic as it has been used in offline training to create some of the best classification models such as ResNet [1] and it has been used to train vision transformers [49].

A popular evaluation dataset in the literature is Permuted MNIST [75]. This is a variation of the MNIST dataset where a fixed, random permutation is applied to all of the images in the dataset to create a different dataset of similar difficulty. It is designed to test the ability of continual learning models to identify items of the same class from different datasets. However, a review of its use found that it creates an unrealistically perfect scenario for continual learning [76].

2.5 Summary

Overall, there is substantial recent literature in the continual learning domain and active research continues to be carried out to identify potential avenues of exploration both from an algorithm and architectural point of view. Further to this, the literature is beginning to mature but it remains difficult to compare and evaluate across different methods due to the inconsistencies in datasets and metrics used.

3 Methodology

As highlighted previously, the comparison of existing techniques is often inconsistent across the literature [16] due to the variety of different continual learning setups [17]. In order to be able to compare techniques, it is important that the testing procedure is fair and consistent, where possible, across techniques.

3.1 Experiment Setup

To evaluate the techniques, I will primarily use the CIFAR-100 dataset [2]. This dataset is sufficiently complex to test if the techniques are feasible in real-world scenarios and it is one of the most commonly used datasets in the continual learning literature [70]. It consists of 100 classes each with 500 training images and 100 testing images. To adapt this to the continual learning setup, I will split the dataset into 5 tasks consisting of 20 classes each, this will use the entire dataset and expose the model to all 100 classes. Only one task will be available at each iteration and every sample will only be seen once for the non-baseline techniques (unless they are saved to memory).

Each technique will be run 5 times to improve the reliability of the results. To ensure a fair test, it is important that the order of the classes and the images is the same across experiments so that none of the techniques are disadvantaged. As such, I will choose 5 fixed integers that will be used to seed the random number generators at the start of each of these runs. Furthermore, all techniques will be run

using Python 3.10.7 on an i7-9700k CPU, a GTX 1080 with 8GB of VRAM, and 32GB of RAM Windows 10 machine.

Further to this, I will also evaluate the techniques on CIFAR-10. This is a less complex dataset consisting of the same images as CIFAR-100 but with only 10 possible labels instead. This simplifies the classification problem and the results of this evaluation are unlikely to reflect the real-world performance of continual learning approaches. However, it will provide a useful comparison to evaluate how the approaches that perform well on simpler datasets perform on more complex problems and it will provide additional insight into the relative performance drop off that may occur between techniques once the difficulty of the problem is increased.

Where applicable, I will use the recommended hyperparameters from the original papers as these will have been tuned to get the best performance out of the specific algorithm. For memory-based methods, I will report in-depth results on CIFAR-100 with a maximum memory buffer size of 5000 and CIFAR-10 with a maximum memory buffer size of 500 with a balanced buffer (such that at the end of training each class has 50 samples). The number of samples per class dynamically scales with the number of classes seen while maintaining an absolute upper limit. Furthermore, I will experiment with varying the maximum memory buffer size and analyse the final accuracy at the end of training all tasks. The variety of buffer sizes, ranging from 200 to 5000, is intended to highlight weaknesses and strengths in different approaches based on the scenario. All techniques, except offline training and Elastic Weight Consolidation, will use a single pass over the task dataset to conform with the online continual learning setup [70]. Importantly, task identifiers will not be available at testing time for any of the techniques [17] ensuring that all approaches conform to the Class-IL formulation of the continual learning problem.

All techniques, except those that use a pre-trained vision transformer [49] (Learning to Prompt and the novel technique), will use a ResNet-32 [1] and an image size of 32x32 pixels. For the pre-trained vision transformer techniques, the images are upscaled to 224x224 pixels.

3.2 Metrics

For each run, I will record:

- **Wall-Clock Time**: the total duration of the training and inference of the technique
- **Peak VRAM Usage**: the maximum space occupied by PyTorch tensors during the training and inference of the technique
- **Peak RAM Usage**: the maximum space occupied for storing temporary data, the memory buffer where applicable, and any model parameters
- **Per Class Classification Results**: the number of true positives per class which can be used to compute the accuracy and forgetting rates

This data will then be processed and analysed across all 5 runs for each technique to evaluate their performance and compare them with the other techniques available. Per task, I will measure the average accuracy calculated across the entire unseen dataset to evaluate the overall quality of the technique, and average forgetting [24] which measures the decrease in accuracy as the model learns more tasks. Let $a_{m,n}$ be the accuracy on task n after learning task m, then the average forgetting at task k is given by F_k:

$$f_{i,j} = \max_{t \in 1,\dots,i-1} (a_{t,j} - a_{i,j}) \tag{1}$$

$$F_k = \frac{1}{k-1} \sum_{i=1}^{k-1} f_{k,i} \tag{2}$$

Further to this, I will measure the top-5 and bottom-5 class accuracy to highlight the strengths and weaknesses of the different approaches, this is important as techniques may have skewed distributions of classification accuracy that cannot be observed via the average accuracy or forgetting rates.

4 Existing Techniques

I select a variety of different techniques ranging from historical regularisation approaches through to the latest state-of-the-art approaches to give a comprehensive comparison of the different paradigms in the online continual learning setup.

4.1 Baselines

Baseline techniques represent the minimum and maximum bounds for continual learning techniques. They are included to compare the relative performance and progress of the techniques.

Finetuning is the lower bound for continual learning, each sample is processed individually and then discarded. This causes severe catastrophic forgetting as the existing parameters are quickly overwritten by the gradient updates and they are never reinforced as the data is only seen once. As a result, it is expected that this will produce poor results and it is the absolute lower bound as any continual learning technique should be able to outperform this [70].

Offline training is the traditional method used to train a neural network. The entire dataset is available and used multiple times to train the network. This is in direct contrast to the aims of continual learning, as such it is expected that offline training should be superior in performance to continual learning methods that do not use pre-trained models, as such it presents an upper bound.

Commonly, Elastic Weight Consolidation (EWC) [20] is included as a baseline in the continual learning domain. It is important that all recent techniques should be able to outperform EWC since it is a regularisation method that was one of the first to approach the continual learning problem. EWC quantifies the importance of each parameter in a model by computing the Fisher Information Matrix. This is a measure of the unseen information about a parameter in a probabilistic distribution, the matrix is useful in a machine learning context because the gradient of the direction of the loss is equivalent to the Fisher Information near a minimum [77]. The importance is then used to regularise the network via a quadratic loss to penalise changes to the weights in proportion to their value to the network.

4.2 Memory Replay

In a typical online memory based setup, each time a new sample is seen by the training process the training algorithm must decide whether to retain the sample for future use or use it once and discard the sample at the end of the mini-batch [70]. The purpose of retaining samples is to reinforce weights in the network, especially outside of the current task as none of the new incoming samples will be of the previously seen tasks in this disjoint setup.

Ideally, samples that offer the most value to the model's training process will be retained for the future and others will be discarded. Quantifying the importance of samples in relation to their value to future states of the model is a significant challenge.

4.2.1 GDumb

During the traditional training phase, GDumb does not actually perform any gradient updates [16]. Instead, as each new image is received GDumb greedily samples and stores these images in a balanced buffer. As such, it always maintains a buffer consisting of the last seen samples of each class without considering the effectiveness of each sample when replayed to the model.

Then, during the inference phase, GDumb trains a model from scratch in the same way as a traditional offline training technique but instead of using the raw dataset it uses the samples that are stored in memory. Because of this naive sampling approach, GDumb provides a baseline specifically for memory-based methods as careful sampling approaches should outperform greedy sampling.

4.2.2 Dark Experience Replay

Dark Experience Replay (DER) and its improved version DER++ present an alternative approach to the storage of samples [34]. Instead of storing just the images, DER also stores the logits from the model at the time that the sample was saved. The purpose of doing this is to use them in the loss function to constrain model updates to limit the change compared to the stored logits using mean-squared error loss. This snapshot of the logits acts as a proxy for the model at the time.

DER++ further builds on this by using a similar approach by applying cross-entropy loss to an additional batch of images sampled from the buffer memory with their respective labels. This further constrains the updates of the model to limit the changes to the model predictions rather than just the logits.

4.2.3 Rainbow Memory

Rainbow Memory presents an approach to evaluate the expected usefulness of samples prior to adding them to the memory buffer [33]. To do so, it presents a technique to quantify the model's classification uncertainty about each particular sample. Let $\mathcal{A}$ denote a set of augmentations, $\mathcal{C}$ denote the set of classes, and $f_{\boldsymbol{\theta}}$ denote the model. Then, for an individual sample $\boldsymbol{x}$, the uncertainty is computed as $u(\boldsymbol{x})$ where $\delta_{i,j}$ is the Kronecker delta:

$$u(\boldsymbol{x}) = 1 - \frac{1}{|\mathcal{A}|} \max_{c \in \mathcal{C}} \sum_{a \in \mathcal{A}} \delta_{c, f_{\boldsymbol{\theta}}(a(\boldsymbol{x}))} \tag{3}$$

Following this, the samples for each class are ordered by their uncertainty. Instead of simply selecting the most certain samples, Rainbow draws from the ordered list using a uniform step to select a diverse range of samples ranging from most certain to least certain. The intention behind this is to use these samples to define the class boundaries and consolidate the interior of the class. Figure 4 shows the most certain and the least certain samples using Rainbow's uncertainty quantification approach for the class 'deer' from CIFAR-10.

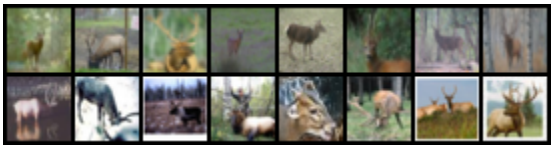


Fig. 4: The top row shows the 8 most certain samples from the 'deer' class, the bottom row shows the 8 least certain samples for the Rainbow technique. The most certain samples have similar backgrounds and colours whereas the least certain samples have unique features.

4.2.4 Supervised Contrastive Replay

Typically in classification tasks, the final output of a model are the logits relating to each possible class. The class corresponding to the index of the maximal logit determines the class of the sample. For some model $f_{\boldsymbol{\theta}}$, a set of classes $\mathcal{C}$ and a sample $\boldsymbol{x}$, the sample is classified using:

$$f(\boldsymbol{x}) = \operatorname*{argmax}_{c \in \mathcal{C}} \delta_{c, f_{\boldsymbol{\theta}}(\boldsymbol{x})} \tag{4}$$

Supervised Contrastive Replay (SCR) [51] uses Nearest Class Mean classification (NCM) [42] in contrast to this typical setup. NCM is directly applicable when dealing with feature representations. Let $\mathcal{B}$ be the memory buffer indexed by the classes $\mathcal{C}$ then a sample $\boldsymbol{x}$ is classified as $f(\boldsymbol{x})$:

$$\boldsymbol{\mu}_c = \frac{1}{|\mathcal{B}_c|} \sum_{\boldsymbol{b} \in \mathcal{B}_c} \boldsymbol{b} \qquad \forall c \in \mathcal{C} \tag{5}$$

$$f(\boldsymbol{x}) = \operatorname*{argmin}_{c \in \mathcal{C}} \|\boldsymbol{\mu}_c - f_{\boldsymbol{\theta}}(\boldsymbol{x})\| \tag{6}$$

NCM classification has seen a small amount of usage in the continual learning literature having first been introduced in iCaRL [41]. However maximisation of logits has appeared more prominently instead before NCM was revived in SCR [51]. It has typically appeared when feature representations are used and it is not exclusive to the continual learning domain [78].

SCR is able to effectively use NCM over softmax classification because it uses Supervised Contrastive Learning (SCL) [79] loss which encourages similar samples, and thus those in the same class, to cluster within the feature space. As such, NCM can be used for classification as SCL loss and NCM classification compliment each other. NCM classification's effectiveness will be highly dependent on the structure of the feature space. If class means are close together relative to the size of the space then it is likely that samples will be incorrectly classified. SCL loss is designed to separate distinct classes out in the feature space while

keeping samples from the same class close together in the feature space. As such, NCM classification is more effective as a result.

4.3 Learning to Prompt

Learning to Prompt (L2P) [47] introduces pre-trained vision transformers (ViT) to the continual learning domain. These are large models that have been trained using massive amounts of data such as the ImageNet-21K dataset using significant amounts of computational resources. While this may seem to contradict the purpose of continual learning, it should instead be treated as a starting point to improve the direction of algorithms within the continual learning domain because these are widely, and freely, available models that would be applied in real-world scenarios.

L2P differs from previous methods as it does not directly store samples, rather it maintains 10 prompts that are each indexed by a key. These 10 prompts and keys are themselves tensors that match the shape of a single dimensional feature that would be outputted by the ViT's feature extractor and they are treated as optimisable parameters. The idea of prompting is influenced by Natural Language Processing [48] where the use of prompts can be used to guide the output of a model. As each sample is processed by the model, a specific feature from the ViT's feature extractor (known as the class feature) is extracted and compared to each key. The top-5 most similar keys are selected and the corresponding prompts are concatenated with the features of the sample. This is then used to classify the sample using a classification head after having been further processed by the ViT. Finally, the classification head, the keys, and the prompts are all optimised.

5 Novel Implementation

I present a series of experiments building up to a novel memory-based approach to the continual learning problem. Drawing upon recent success in the literature, I combine aspects of L2P, Rainbow, SCR, and GDumb as well as introducing two sample uncertainty quantification methods that are applied in a continuous feature space for use with NCM classification to enhance the quality of memory samples.

5.1 Pre-trained Vision Transformer

Learning to Prompt [47] introduced the idea of utilising a pre-trained vision transformer (ViT) [49] as the backbone to enhance the performance of continual learning due to the large amounts of data and computation resources used to train it.

The motivation for using a pre-trained vision transformer is that they are freely available and if continual learning techniques are to be applied to real-world applications it is necessary to make use of all available resources such as pre-trained models. It is essential that the pre-trained model is not trained using the same data that will be used for the continual learning task as otherwise this defeats the purpose of continual learning and transforms the problem into an offline training problem instead. My method utilises the ViT-B-16 model that is trained using ImageNet-21K which does not share image data with the evaluation datasets [80].

5.2 Comparing Loss Functions

Cross-Entropy (CE) loss is the typical loss function that is used in multi-class classification tasks. It provides a measurement between the difference of two probability distributions in the form of log loss [81]. The entropy increases as the predicted label differs from the expected label of the sample

Supervised Contrastive Learning (SCL) loss is designed to separate the feature space into disjoint regions according to the class samples. It contrasts different *views* against the original image sample, the views are augmented versions of the sample to guide the feature encoder to label the augmented views with the same label as the original sample [79].

In order to determine the effectiveness of SCL loss compared to CE loss, I conducted an experiment using two different MLP setups and measured the final accuracy after training the MLP. The two MLP setups are fully connected networks consisting of either 2 layers or 3 layers. Each training process was conducted using the same 5000 randomly selected samples from the CIFAR-100 dataset. Table 1 shows the outcome of these experiments. All of the experiments used the NCM classification as discussed previously.

MLP	CE Loss	SCL Loss
2-layer	**70.33%**	69.46%
3-layer	**66.57%**	62.48%

TABLE 1: A comparison between the final accuracy attained when using each loss function applied to two different MLP heads with varying depth.

Ultimately, in the context of CIFAR-100 there is a small difference between the final accuracy for the two loss functions and this is further amplified when using the 3-layer MLP head. As such, this indicates that a 2-layer MLP is most appropriate and that CE loss and SCL loss perform roughly equally in this case but CE loss does outperform SCL loss in this context. However, it is important to note that in different domains these differences may vary and further experimentation could be necessary to explore this.

5.3 Sample Classification

The use of SCL loss naturally suggests that NCM classification will be effective because of the previously detailed feature clustering. However, a downside to using NCM is the additional overhead introduced at inference time. In contrast to maximisation, NCM requires the pre-computation of the means each time the buffer set is updated and each sample must be compared to all means instead of a single maximisation operation. However, the additional computation provides valuable information about the structure of the feature space which can be exploited to quantify the uncertainty of samples.

5.4 Memory Sample Selection

Memory-based approaches remain at the forefront of the continual learning literature as discussed previously. There have been a variety of approaches to improve the quality and diversity of the samples stored in memory with the

purpose of refining the boundaries between classes to enable the model to be exposed to a diverse set of samples to prevent catastrophic forgetting.

Rainbow [33] introduced an approach to evaluate the uncertainty of a sample using a series of independent augmentations applied to each sample and classifying these samples and counting how many were classified under the same class label. This approach was effective but it requires classification by maximisation of the logits. A similar approach could be applied to NCM classification however this sacrifices a major benefit of the NCM classification approach. With NCM, we have continuous values representing the distance to the means instead of discrete class labels. If we applied the approach from Rainbow, we would be losing substantial amounts of information about the distances.

I propose an alternative method to compute the uncertainty of samples when using NCM classification that accounts for the distance between a sample and the class means.

Let $\mathcal{A}$ denote a set of augmentations, $\mathcal{C}$ denote the set of class means, and $f_{\boldsymbol{\theta}}$ denote the pre-trained model. Then, for an individual sample $\boldsymbol{x}$, the unnormalised uncertainty is computed as $u(\boldsymbol{x})$:

$$d(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{7}$$

$$u(\boldsymbol{x}) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \min_{c \in \mathcal{C}} d(c, f_{\boldsymbol{\theta}}(a(\boldsymbol{x}))) \tag{8}$$

It is important to note that the feature space is unconstrained and unbounded. As such, the uncertainty values in their unnormalised forms do not necessarily convey accurate information about the uncertainty of a sample. This is because the positions of the class means in the feature space do not necessarily follow any pattern, as such the distance between class means themselves does not convey any information about the samples but rather the construction of the feature space through model training.

To alleviate these concerns, consider a batch B of samples drawn from an individual task. For a sufficiently large batch size we would expect the batch to be representative of the task's dataset. By processing the unnormalised uncertainty of the entire batch, we can leverage this representation and normalise the uncertainty values across these values to reduce the uncertainties into the set $[0, 1]$ instead of $[0, \infty)$. I refer to this approach as batch-normalised uncertainty.

5.5 Alternative Uncertainty Approach

A potential issue with the previously proposed uncertainty quantification approach is the structure of the space. It is difficult to explain the positioning of the mean feature vectors in the space due to the black-box nature of the deep learning approach. For example, imagine there are two adjacent mean feature vectors, one with samples sparsely within its radius of classification and the other with samples densely in the radius of classification. Despite these both potentially having relatively the same ratio between distances to samples the sparsely populated class would have significantly worse uncertainty classification results.

To overcome this in the previously proposed approach, I suggest normalising the uncertainties across the batch of samples. This relies on the assumption that the batch is sufficiently large enough to justify this normalisation approach. However, this may breakdown, especially in the continual learning domain when considering that a task may not contain samples from all classes and thus the task does not reflect the true distribution of the data that the model is set to be trained on.

As an alternative, I propose an additional uncertainty quantification approach utilising the relative distance between the two closest mean features. Let d_1 and d_2 be the Euclidean distance to the closest and second closest mean class feature vectors for a sample $\boldsymbol{x}$ such that $d_1 \leq d_2$. Then the uncertainty $u(\boldsymbol{x})$ is given by:

$$u(\boldsymbol{x}) = \frac{d_1}{d_2} \tag{9}$$

By construction, this uncertainty is inherently normalised to the range $[0, 1]$. For an uncertain sample the distance to the two means will satisfy $d_1 \approx d_2$ thus the ratio will be close to 1 and for a certain sample, the distances will satisfy $d_1 \ll d_2$ so the ratio will instead be close to 0. I refer to this approach as relative distance uncertainty.

To determine if the proposed memory sampling techniques are effective, I conducted an experiment to compare the average accuracy over the tasks when using NCM classification with the pre-trained ViT when using random sampling to populate the memory buffer compared to using the proposed uncertainty sampling methods to populate the memory buffer. The samples in the memory buffer were then used to compute the class means that were used for classification. This experiment was a test of the effect of the memory sampling strategies on the model's classification ability.

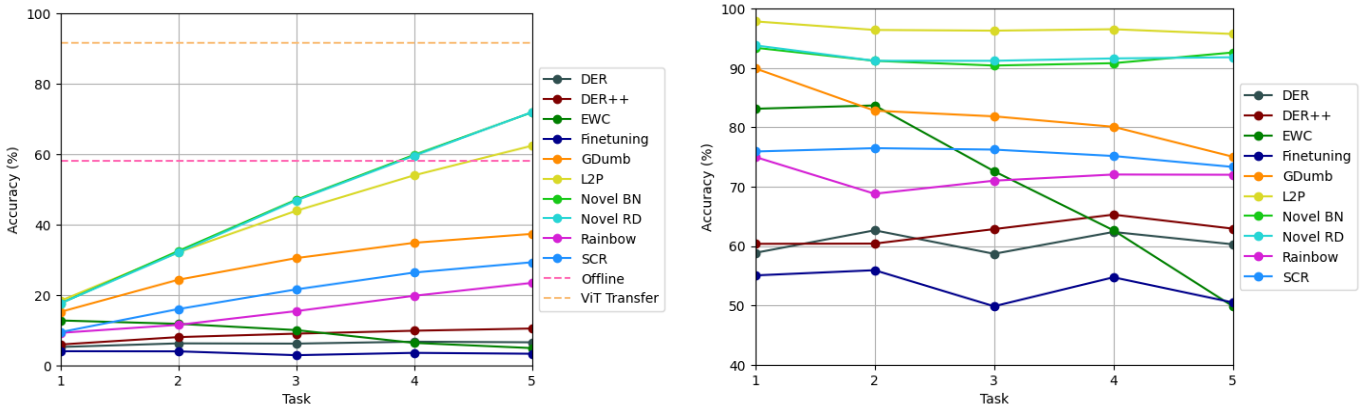
Sampling	Task 1	Task 2	Task 3	Task 4	Task 5
Random	17.34%	32.07%	45.55%	58.74%	70.31%
Batch Norm.	**17.59%**	**32.48%**	**47.05%**	**59.78%**	71.89%
Relative Dist.	17.58%	32.06%	46.83%	59.52%	**71.94%**

TABLE 2: Average accuracy comparison between both uncertainty sampling techniques and random sampling. Bolded values are the greatest value in the column.

Table 2 shows a comparison between the uncertainty techniques and random sampling. As more tasks are introduced, both uncertainty sampling approaches outperform random sampling suggesting that samples of varying certainty help to improve the performance of the model leading to an approximate 2% increase over random sampling, this diverges from random sampling as more tasks are introduced. This suggests that uncertainty sampling is effective.

5.6 Proposed Algorithm

Following these experiments, I present an algorithm that can be applied to the online continual learning setup. The key components that I have experimented with are the loss function, sample classification method, and sample uncertainty quantification. The algorithm is modular in the sense that the uncertainty quantification method can be substituted



(a) Average accuracy over tasks

(b) Top-5 accuracy over tasks

Fig. 5: Graphs showing the overall accuracy and top-5 accuracy after task n has been used to train the model in the CIFAR-100 5000 sample setup. The model is evaluated using classes that it has been exposed to in the previous tasks as well as unseen classes that appear in later tasks.

Technique	Accuracy (%)	Top-5 (%)	Bottom-5 (%)	F_5	Peak RAM (MB)	Peak Tensor (MB)	Total Time (s)
Finetuning	3.28 ± 0.00	50.48 ± 0.05	0.00 ± 0.00	17.95 ± 0.01	2369.19 ± 0.29	243.71 ± 0.00	51.56 ± 0.44
Offline	58.17 ± 0.00	86.40 ± 0.01	26.96 ± 0.01	N/A	2373.20 ± 0.29	243.71 ± 0.00	7534.70 ± 11.10
ViT Transfer [49]	91.67	N/A	N/A	N/A	N/A	N/A	N/A
DER [34]	6.53 ± 0.00	60.28 ± 0.02	0.00 ± 0.00	26.86 ± 0.01	2606.41 ± 0.59	193.50 ± 0.00	$\mathbf{186.76 \pm 0.22}$
DER++ [34]	10.45 ± 0.00	62.92 ± 0.02	0.00 ± 0.00	22.09 ± 0.01	2606.06 ± 0.40	193.50 ± 0.00	188.15 ± 1.30
EWC* [20]	4.89 ± 0.00	49.84 ± 0.00	0.00 ± 0.00	15.74 ± 0.01	$\mathbf{2362.45 \pm 5.37}$	243.71 ± 0.00	2066.33 ± 14.80
GDumb [16]	37.33 ± 0.00	75.04 ± 0.01	10.28 ± 0.01	20.48 ± 0.01	2471.72 ± 15.90	132.70 ± 0.00	5978.18 ± 9.02
Rainbow [33]	23.42 ± 0.00	72.00 ± 0.00	0.44 ± 0.00	7.20 ± 0.01	2451.08 ± 0.71	$\mathbf{130.82 \pm 0.00}$	5041.65 ± 23.39
SCR [51]	29.28 ± 0.01	73.32 ± 0.00	2.40 ± 0.01	8.34 ± 0.01	2583.68 ± 0.59	720.50 ± 0.03	1789.53 ± 3.37
L2P [47]	62.40 ± 0.00	$\mathbf{95.72 \pm 0.00}$	2.48 ± 0.01	32.45 ± 0.01	2323.67 ± 0.36	1954.40 ± 0.22	4429.80 ± 43.35
Novel BN	$71.89 \pm$ TBC	$92.60 \pm$ TBC	$46.80 \pm$ TBC	$7.31 \pm$ TBC	$2865.12 \pm$ TBC	$1753.62 \pm$ TBC	$35958.51 \pm$ TBC
Novel RD	$\mathbf{71.94 \pm}$ TBC	$91.80 \pm$ TBC	$\mathbf{48.00 \pm}$ TBC	$\mathbf{7.05 \pm}$ TBC	$2860.89 \pm$ TBC	$1753.89 \pm$ TBC	$35968.08 \pm$ TBC

TABLE 3: Final results after all 5 tasks have been processed in the CIFAR-100 5000 sample (where applicable) online setup (*except for EWC). Mean values and standard errors are reported. Peak Tensor refers to the peak VRAM allocated to PyTorch tensors as measured by PyTorch. Peak RAM refers to the peak RAM usage solely and does not include any allocated VRAM. F_5 is the average forgetting (relative to the accuracy) at the end of training. Best results for non-baseline techniques are in bold.

out without affecting the rest of the algorithm, this enables both batch-normalised and relative distance uncertainty to be used independently of each other.

The general outline of the algorithm is that for each task, the algorithm iterates over all of the samples in the task dataset once only and uses the uncertainty quantification method to select the samples to save to memory. Then, once these samples have been selected, the algorithm will train a 2-layer MLP head with the pre-trained vision transformer using the samples in the memory buffer only, for a fixed number of epochs per task. Finally, to classify samples the mean of the features for each class will be computed using the memory samples in the buffer. Then, NCM classification will be used to classify the samples. This will be repeated for each new task in the problem.

To optimise the MLP head, I will use SCL loss. Despite it performing slightly worse in the experimental setup, in theory I would expect the loss function to better refine the feature space in comparison to CE loss which should enable the two uncertainty sampling methods to perform better as the feature space possesses a more rigorous structure.

The two uncertainty approaches will be used independently of each other. To differentiate these, for the batch-normalised approach I refer to the algorithm as **Novel-BN** and for the relative distance approach I refer to the algorithm as **Novel-RD** in the comparisons against the existing literature techniques.

6 Results

In line with the methodology, I conduct a series of experiments to compare the performance of the selected continual learning techniques from the literature and the proposed novel techniques. I utilise both the CIFAR-10 and CIFAR-100 datasets and present in-depth results for CIFAR-100 with a maximum buffer size of 5000 samples, as shown in Table 3, and CIFAR-10 with a maximum buffer size of 500 samples, as shown in Table 4. These buffer sizes have been deliberately chosen such that at the end of training each class will have a total of 50 stored samples in memory where applicable. I also visualise the impact of forgetting on the

accuracy of the first task as more tasks are introduced and used for training in both of these setups as shown in Figure 7.

Additionally in Figure 5, I present the average overall accuracy and top-5 accuracy as the number of tasks increases in the CIFAR-100 5000 samples setup. As well as this, I also show the average overall accuracy as the number of tasks increases for CIFAR-10 with both a 500 sample and a 5000 sample memory buffer for additional comparison in Figure 6. Further to this, I compare the techniques over a wide range of maximum buffer sizes to evaluate the impact of the buffer size on the final accuracy, these are 200, 500, 1000, 2000, and 5000 samples on both CIFAR-10 and CIFAR-100 as shown in Figure 8.

6.1 Baselines

The baseline techniques, Finetuning, Offline, and ViT Transfer [49], form the expected bounds for the techniques. As anticipated, Finetuning performs the worst overall across both of the CIFAR-10 setups and the CIFAR-100 5000 samples setup. It maintains a stable performance but ultimately fails to reach any significant classification accuracy. Furthermore, even when the continual learning techniques are given minimal samples, such as the 200 sample CIFAR-100 setup, Finetuning is still outperformed suggesting that the specialised continual learning techniques are beneficial to the classification performance.

Critically, two upper bounds - Offline and ViT Transfer - for the problem are presented due to the design of the different techniques compared. Offline training performs as expected and outperforms all non pre-trained ViT-based techniques suggesting that there is still a significant gap separating continual learning techniques and offline training approaches.

It is important to note that technqiues that use the pre-trained ViT as a backbone are expected to outperform those that do not because of the vast quantity of training data that has been used to create the backbone network. L2P, Novel-BN, and Novel-RD all rely on the pre-trained ViT in their respective approaches. The upper bound for these techniques is ViT Transfer. This is where the pre-trained model is secondarily optimised on the specific dataset to tailor the feature representations. The results for this are directly from the original ViT implementation due to computation restrictions as the ViT is trained using substantial computational resources [49].

6.2 ResNet-based Approaches

DER [34] is the worst performing memory-based approach in terms of classification performance. In the CIFAR-100 setup, the average accuracy immediately stagnates at around 6% and in the CIFAR-10 setup only minor increases in the accuracy are achieved as more tasks are introduced. While this technique is able to retain some information about previously seen tasks as shown in Figure 7, it suffers substantial drop-off in first task accuracy which is especially evident in the CIFAR-100 setup where this falls over 800% between task one and task two.

DER++ is the improved version of DER [34], while it was able to outperform DER in all experimental setups, it still achieves generally low-quality results compared to other memory-based techniques, especially when the maximum number of memory samples increases. In the CIFAR-100 setup, it outperforms DER in overall accuracy, top-5 accuracy, and average forgetting but still maintains a bottom-5 accuracy of 0% suggesting it is unable to retain information about all of the tasks. It experiences a very minor increase in accuracy as more tasks are introduced but it does retain information about the first task significantly better than DER does. In fact, in the CIFAR-10 500 samples setup, DER++ surprisingly outperforms GDumb, Rainbow and SCR in terms of first task accuracy retention and outperforms Rainbow in both final accuracy and top-1 accuracy but is substantially poorer in terms of bottom-1 accuracy.

Both DER and DER++ suffer due to their hybrid regularisation approach. This is highlighted in Figure 8 where, counter-intuitively, DER and DER++ are the only two techniques to suffer performance decreases as the amount of available data increases. However, when considering this in the context of the plasticity-stability dilemma [11] and noting that regularisation approaches can be detrimental [69] by limiting the plasticity of the training process actively harming the model's classification performance, the decrease in performance is expected. Ultimately, DER and DER++ struggle across both datasets and over all sample sizes tested.

EWC is the only offline continual learning [20] technique used in this comparison. It further emphasises that regularisation approaches struggle to compete against memory-based approaches despite having the inherent advantage of being able to revisit samples. It is the worst performing technique on CIFAR-10 with an absolute accuracy increase of approximately 0.7% in overall accuracy compared with Finetuning. This is further evidenced in the CIFAR-100 setup where EWC suffers worse overall accuracy than DER and only slightly better final accuracy than Finetuning again.

In addition to this, EWC is the only technique to have a sustained decrease in top-5 accuracy and average accuracy as the number of tasks increases in the CIFAR-100 setup as evidenced in Figure 5. The top-5 accuracy drops from approximately 83% to under 50% at the end of the final task, this is worse than the performance of Finetuning. It also suffers high forgetting relative to its final accuracy in both the CIFAR-10 and CIFAR-100 setups suggesting that it is unable to retain information about the previously seen tasks further emphasising the difficulties faced by regularisation approaches.

GDumb [16] performs well in most of the setups that were used for experimentation. As the number of samples increases, it is expected that GDumb's performance will converge towards the Offline baseline because GDumb uses identical training methods to offline training but with a smaller dataset consisting solely of the memory buffer. The experimental results align with this expectation and it is particularly clear in Figure 8 as the maximum size of the buffer increases.

In the CIFAR-100 5000 samples setup, GDumb is the best performing ResNet-based continual learning approach attaining overall accuracy of just over 37%. Interestingly, this is roughly 64% of the accuracy of offline training but

using only 10% of the dataset. However, when the number of samples that is available is low, the naivety of the GDumb approach is uncovered. For example, in Figure 8, SCR outperforms GDumb up to 2000 samples on both datasets. Regardless, GDumb does continue to outperform many of the more specialised continual learning approaches.

Further to this, GDumb has the best bottom-5 accuracy in the CIFAR-100 setup out of the ResNet-based continual learning approaches by a significant margin. It also has the best top-5 accuracy but this is less substantial with Rainbow and SCR achieving similar performance in this metric. However, this does imply that GDumb has a tighter spread of accuracy over the tasks contributing to its superior overall performance. In addition, GDumb is capable of retaining information but it does suffer mild forgetting as evidenced by the sustained downwards trend in first task accuracy on the CIFAR-10 500 samples setup in Figure 7, but ultimately it is not substantial relative to the other techniques.

Nonetheless, GDumb is outperformed by both Novel-BN and Novel-RD in all setups, this is expected due to the benefits of using the pre-trained ViT backbone. Similarly, L2P outperforms GDumb in almost all scenarios but impressively GDumb is able to outperform it on the CIFAR-10 5000 samples setup unexpectedly. This is likely due to the saturation of the L2P prompts causing forgetting highlighting a weakness that is discussed later on.

Rainbow [33] is typically the third best technique in this set in terms of overall accuracy on the CIFAR-100 5000 samples setup and maintains a steady top-5 accuracy with very close performance to both GDumb and SCR which outperform it in other metrics. Further to this, it is able to retain information about almost all classes in this setup but the range of accuracy is sizeable with a bottom-5 accuracy of just over 0.4% which is very poor and a top-5 accuracy of roughly 72%. However, Rainbow does experience relatively low average forgetting and while it does have a general downwards trend in first-task accuracy, it is not monotone suggesting that by carefully selecting samples, some of these forgetting effects can actually be reversed or compensated for in later tasks, this is exhibited between task 2 and task 3 in Figure 7a.

Unlike DER, Rainbow follows a stable upwards trend in terms of overall accuracy on the CIFAR-100 5000 samples setup further suggesting that it is retaining information successfully throughout the training process. Over the course of 5 tasks, its accuracy increases from approximately 9% to 23%, however this gain is relatively small in comparison to its performance on the first task.

This technique is especially sensitive to the maximum number of samples in the memory buffer, it has one of the steepest upwards trends behind GDumb as the number of samples increases, especially when considering the jump in overall accuracy as the number of samples increases from 2000 to 5000 compared to SCR plateauing - this is seen across both datasets examined in Figure 8. In addition to this, when the number of samples is low, Rainbow struggles. For example, DER++ outperforms Rainbow with 500 samples in the CIFAR-100 scenario before being eclipsed as the number of samples available increases. This is likely due to the diverse sampling of Rainbow since it selects samples proportionally from the spectrum of sample uncertainty and thus for a low number of samples it may not have sufficient data points to correctly differentiate between class boundaries.

SCR [51] performs well and outperforms GDumb in some scenarios and decisively outperforms Rainbow across all experimental setups. In the CIFAR-100 5000 samples setup, SCR has similar performance to Rainbow on the first task but quickly surpasses the accuracy of Rainbow as the number of tasks increases. This behaviour is further replicated in the CIFAR-10 5000 samples setup and SCR also dominates on the far lower 500 samples setup suggesting that SCR is able to perform well even when the number of samples is low. However, as seen in Figure 8, SCR's performance plateaus once the maximum number of samples reaches 2000 with only very minor gains in performance as the maximum memory buffer size increases from 2000 samples to 5000 samples whereas both GDumb and Rainbow utilise these additional samples more effectively. Nonetheless, SCR still continues to outperform Rainbow at 5000 samples on both datasets.

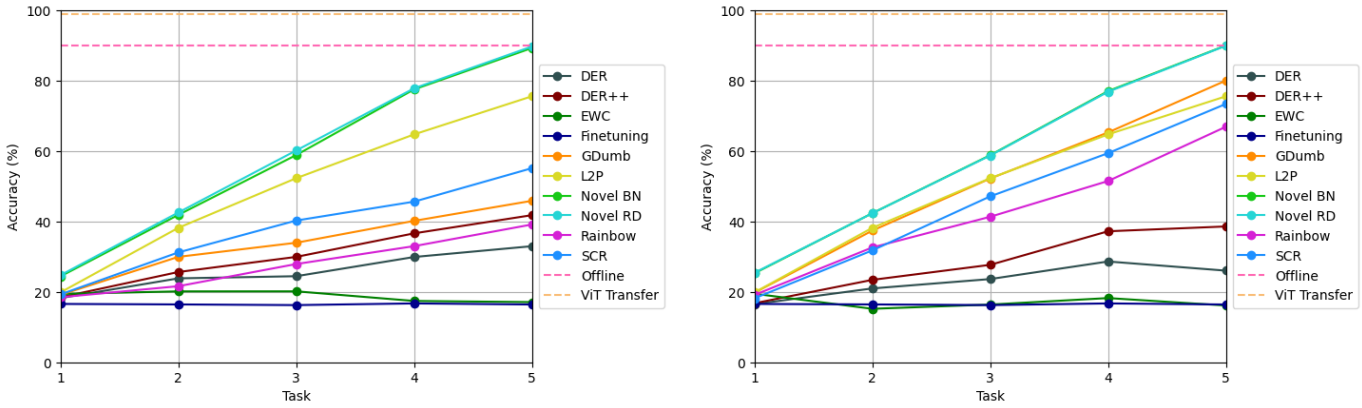
In terms of forgetting, SCR has a sustained upwards trend in overall accuracy on all experimental setups indicating that it is capable of retaining information while also learning about the newly introduced classes. This is further evidenced by the low average forgetting relative to the accuracy on both the CIFAR-100 and CIFAR-10 setups. On the CIFAR-10 setup, SCR has the best top-1, bottom-1, and overall accuracy by a significant margin (except on bottom-1 accuracy where GDumb is less than 2% behind). This indicates that SCR has a tight spread of accuracy across the different tasks and retains information well with a bottom-1 accuracy of over 35%. In addition to this, the decay of the first-task accuracy is slow further suggesting that the stability of the learned information is well balanced against the plasticity of the model enabling it to learn information about future tasks.

6.3 Pre-trained ViT Approaches

In the CIFAR-100 5000 samples setup, L2P [47], Novel-BN, and Novel-RD achieve similar average accuracy after only task 1 and task 2 have been used where they all achieve approximately 32% classification accuracy. However, as more tasks are introduced, their average accuracies start to diverge with Novel-BN and Novel-RD gaining in performance compared to L2P as evidenced in Figure 5a. Regardless, L2P performs well and outperforms offline training a ResNet directly.

Novel-BN and Novel-RD have very similar accuracy on the CIFAR-100 5000 samples setup where they both perform significantly better than the other continual learning techniques. Despite L2P also using a ViT, L2P is unable to compete as the number of tasks increases. Despite the superior overall accuracy, in terms of top-5 accuracy, L2P consistently outperforms both Novel-BN and Novel-RD across all tasks that are introduced. However, in terms of bottom-5 accuracy L2P performs far poorer with approximately 2.5% bottom-5 accuracy compared with Novel-BN and Novel-RD maintaining a bottom-5 accuracy of 46-48%.

On the CIFAR-10 setups, Novel-BN and Novel-RD again have very similar performance but in the low sample setup, where the buffer is limited to 500 samples, Novel-RD performs better on the intermediate tasks and at the end of

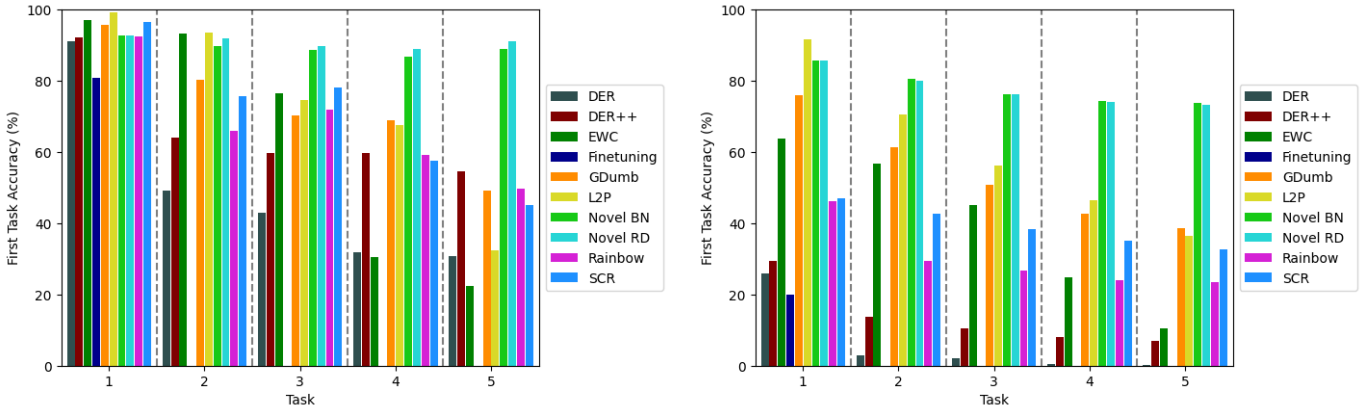


(a) CIFAR-10 500 Samples (b) CIFAR-10 5000 Samples

Fig. 6: Graphs showing the progression of overall accuracy on all tasks (including classes that have not been used to train the model yet) using CIFAR-10.

Technique	Accuracy (%)	Top-1 (%)	Bottom-1 (%)	F_5	Peak RAM (MB)	Peak Tensor (MB)	Total Time (s)
Finetuning	16.42 ± 0.01	86.54 ± 0.02	0.00 ± 0.00	81.95 ± 0.03	2358.13 ± 0.29	1764.49 ± 0.00	72.13 ± 0.56
Offline	$89.95 \pm$ TBC	$95.50 \pm$ TBC	$80.30 \pm$ TBC	N/A	$2385.85 \pm$ TBC	$1764.49 \pm$ TBC	$11640.82 \pm$ TBC
ViT Transfer [49]	98.95	N/A	N/A	N/A	N/A	N/A	N/A
DER [34]	$32.99 \pm$ TBC	$92.30 \pm$ TBC	$1.60 \pm$ TBC	$59.80 \pm$ TBC	$2519.60 \pm$ TBC	$1694.26 \pm$ TBC	**$230.92 \pm$ TBC**
DER++ [34]	$41.81 \pm$ TBC	$87.40 \pm$ TBC	$7.70 \pm$ TBC	$43.43 \pm$ TBC	$2519.58 \pm$ TBC	$1694.26 \pm$ TBC	$231.84 \pm$ TBC
EWC* [20]	$17.10 \pm$ TBC	$83.70 \pm$ TBC	$0.00 \pm$ TBC	$36.01 \pm$ TBC	$2365.35 \pm$ TBC	$1764.49 \pm$ TBC	$2761.40 \pm$ TBC
GDumb [16]	$45.89 \pm$ TBC	$65.20 \pm$ TBC	$34.00 \pm$ TBC	$25.00 \pm$ TBC	$2481.73 \pm$ TBC	$1493.05 \pm$ TBC	$1290.63 \pm$ TBC
Rainbow [33]	$39.16 \pm$ TBC	$56.30 \pm$ TBC	$26.10 \pm$ TBC	$11.14 \pm$ TBC	$2403.55 \pm$ TBC	$1448.12 \pm$ TBC	$2403.50 \pm$ TBC
SCR [51]	$55.16 \pm$ TBC	$90.20 \pm$ TBC	$35.40 \pm$ TBC	$36.10 \pm$ TBC	$2516.21 \pm$ TBC	**$722.74 \pm$ TBC**	$1463.65 \pm$ TBC
L2P [47]	$75.60 \pm$ TBC	**$96.70 \pm$ TBC**	$15.80 \pm$ TBC	$27.81 \pm$ TBC	**$2317.43 \pm$ TBC**	$1994.28 \pm$ TBC	$4473.51 \pm$ TBC
Novel BN	$89.31 \pm$ TBC	$93.00 \pm$ TBC	$77.40 \pm$ TBC	$2.20 \pm$ TBC	$2852.72 \pm$ TBC	$1722.67 \pm$ TBC	$9125.55 \pm$ TBC
Novel RD	**$89.75 \pm$ TBC**	$94.60 \pm$ TBC	**$83.90 \pm$ TBC**	**$2.09 \pm$ TBC**	$2853.14 \pm$ TBC	$1722.67 \pm$ TBC	$9094.40 \pm$ TBC

TABLE 4: Final results after all 5 tasks have been processed in the CIFAR-10 500 sample (where applicable) online setup (*except for EWC). Mean values and standard errors are reported. Peak Tensor refers to the peak VRAM allocated to PyTorch tensors as measured by PyTorch. Peak RAM refers to the peak RAM usage solely and does not include any allocated VRAM. F_5 is the average forgetting (relative to the accuracy) at the end of training. Best results for non-baseline techniques are in bold.



(a) CIFAR-10 500 Samples (b) CIFAR-100 5000 Samples

Fig. 7: Graphs showing the change in the accuracy of the first task as more tasks are trained. Higher is better.

training Novel-RD has minorly better overall accuracy and top-1 accuracy but in terms of bottom-1 accuracy, Novel-RD outperforms Novel-BN by over 6%. This aligns with the reasoning for introducing the Novel-RD approach as it is able to better select samples without relying on the individual batch for normalisation of the uncertainty.

The number of samples does not appear to affect Novel-BN, this is likely due to the pre-trained ViT and the CIFAR-10 dataset being too simple for there to be a substantial effect. However the maximum number of samples does have a significant impact in the CIFAR-100 setup as evidenced in Figure 8b. Up to 2000 samples, there is substantial classification performance gains but, like SCR, this plateaus from 2000 to 5000 samples.

Both Novel-BN and Novel-RD are resilient to forgetting, this is evident in Figure 7 as they both experience very similar first-task forgetting. They both suffer a minor downwards trend on the CIFAR-100 5000 samples setup but maintain a consistently high retention rate of information gained in the first task - especially in contrast to L2P. However, on the CIFAR-10 500 samples setup, Novel-RD outperforms Novel-BN further suggesting that it has a slightly superior approach despite them having nearly identical performance after task 1 is completed.

L2P attains high-quality performance in all experimental setups where it is competitive and outperforms all ResNet-based approaches except on the CIFAR-10 5000 samples setup where GDumb outperforms it on the final task. Despite this generally good classification performance, L2P suffers severe forgetting as evidenced by the drop off of the first task accuracy where it falls by over 50% from the end of training on the first task. Further to this, it has moderately high relative average forgetting and low bottom-1 and bottom-5 accuracy on both of the CIFAR-10 and CIFAR-100 setups indicating that it has difficulties with retaining information.

7 Evaluation

Following the analysis of the results, I explore and evaluate the effectiveness of different techniques as well as the trade-off in classification performance and resource usage that would need to be considered for the deployment of continual learning techniques.

7.1 Pre-trained Vision Transformers

Despite Novel-BN and Novel-RD using the same pre-trained ViT backbone approach as L2P, there is a significant difference between the classification accuracy of these two types of approach. L2P is outperformed in overall accuracy by the other two approaches. This highlights two issues with the L2P approach, namely prompt saturation and key selection frequency. As more samples are used to update the prompts, they are unable to fully adapt to the shifting data distribution due to catastrophic forgetting as they are optimised like network parameters, this is evidenced by the relatively high average forgetting at the end of training. Furthermore, as the keys and prompts are optimised, those keys that are selected initially by chance will move closer to the class tokens from the ViT in turn causing them to be closer to unseen samples and thus they will be selected more frequently. Thus, some prompts become more saturated and, in a self-reinforcing process, their keys become more likely to be selected. This is mitigated by scaling the distances to the keys by their frequency but it does not fully resolve the issue as noted in the results.

The overall usage of pre-trained ViT models indicates that they are capable of high-quality classification results. However, another important factor to consider is the amount of time required to train the model. Novel-BN and Novel-RD take a substantial amount of time at approximately 10 hours for the duration of training on the CIFAR-100 5000 samples setup. In absolute terms, this is not unreasonable especially when considering that the techniques have not been trained in a machine learning specific setup. However, relative to L2P, which takes approximately 73 minutes using the same setup, this would likely be a consideration depending on the use case for the continual learning model. This is further reflected in the CIFAR-10 500 samples setup with L2P being approximately twice as fast as Novel-BN and Novel-RD.

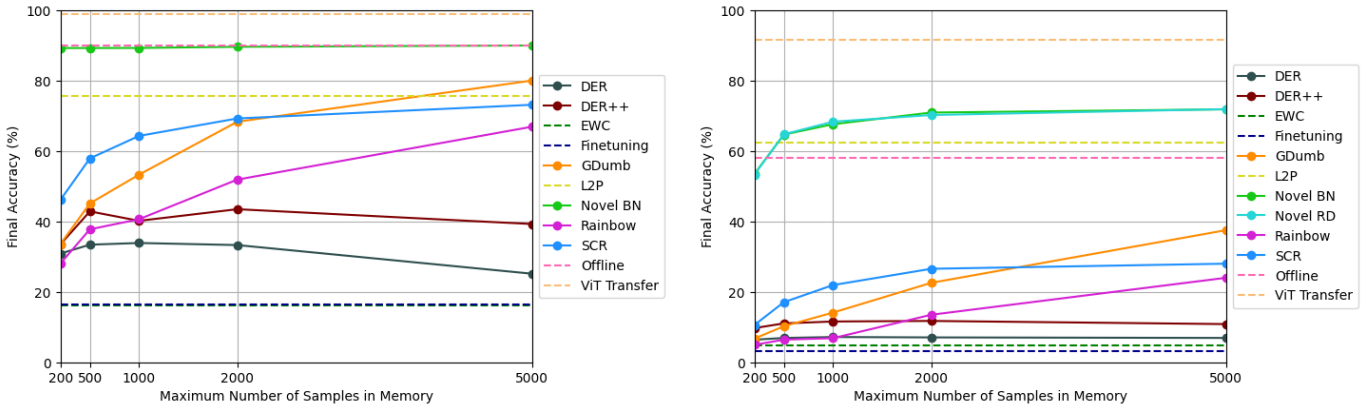
Further to this, memory usage is another important factor for the three techniques that use a pre-trained ViT. This introduces additional overhead costs if these techniques were to be applied in real-world scenarios which would need to be considered. The reason for this increased memory usage is the size of the pre-trained model (occupying nearly 400 MB) and the requirement to upscale images to 224 x 224 pixels instead of 32 x 32 pixels for the ResNet-based approaches which increases the space occupied by tensors in both the buffer memory and on the GPU. However, this is less significant in the case of L2P because it does not store samples directly, rather it stores single-channel feature representations which use substantially less memory.

7.2 ResNet Approaches

ResNet-based approaches do not utilise any pre-training to improve the performance of the model, but in exchange they can be faster and use less computational resources than pre-trained ViT approaches. This set of approaches is more sensitive to the maximum number of samples that can be stored as they are used to alleviate catastrophic forgetting and define the feature space for the model.

EWC is an offline technique making it less applicable in real scenarios, especially for handling real-time data. Furthermore, the performance of EWC is substantially worse than all other techniques across all of the different experimental setups. This makes EWC unsuitable for use as it has been superseded by better techniques.

The fastest algorithms in this class of approaches are DER and DER++ taking approximately 3 to 4 minutes to run in the setups described previously. With a low number of samples, DER and DER++ are competitive - especially in the 200 and 500 setup. However, as the number of samples increases their performance quickly stagnates before dropping off suggesting that they perform poorly when larger quantities of data are available and the trade-off between run time and accuracy becomes too extreme. It is clear that the performance of DER++ is substantially better than DER, as such it is unlikely that DER would be utilised in any scenario. Of particular interest is DER++ outperforming Rainbow up until approximately the 1000 sample mark at which point Rainbow is able to dominate DER and DER++. Additionally, their memory usage is slightly higher than GDumb and Rainbow suggesting that they are not the best choice.



(a) CIFAR-10 (b) CIFAR-100

Fig. 8: Graphs showing the overall accuracy at the end of training on all tasks with varying maximum memory buffer sizes.

As the number of samples increases, GDumb's performance continues to increase. GDumb attains 90% of the accuracy of Offline while using 63% of the the time it required and a tenth of the data. The downside to GDumb in comparison to the other techniques is the total time taken, it requires slightly more time than Rainbow does, taking approximately 100 minutes to complete training which is significantly more than SCR considering that SCR is able to compete, and outperform, GDumb at the lower end of the maximum memory buffer sizes.

Another technique that benefits from increasing the maximum number of samples is Rainbow. It performs poorly at low samples but after 1000 samples it begins to experience a sustained increase with similar performance gains to GDumb. Nevertheless, it is outperformed by SCR at both 2000 samples and 5000 samples. In fact, SCR outperforms Rainbow's 5000 sample accuracy when using only 2000 samples and a third of the time suggesting that SCR has excellent performance comparatively. SCR also outperforms GDumb for lower samples with SCR performing significantly better at 2000 samples and below on both CIFAR-10 and CIFAR-100 while using approximately a fifth of the time required to train GDumb indicating that it could be applied to different use cases such as near real-time scenarios. Additionally, SCR does not require the model to be trained from scratch each time a new sample is introduced like GDumb does. After 2000 samples however, SCR begins to plateau heavily and only a small performance increase is realised from 2000 to 5000 samples.

7.3 Summary

Overall, techniques using a pre-trained ViT backbone show the most promise to offering acceptable performance in the experimental setups presented here. While they typically utilise a greater amount of resources and require more time to run, the benefit of improved accuracy is substantial, especially on more complex datasets. In the case that speed is essential, SCR, GDumb with low samples, or L2P are the most promising, but when maximum classification performance is required, Novel-RD and Novel-BN are the best performers.

8 Conclusion

In conclusion, I propose a novel approach to the continual learning problem that builds upon recent advances in the state-of-the-art by leveraging a pre-trained ViT as proposed by L2P [47] and clustering features using an alternative loss function which was used in SCR [51]. Further to this, I present two similar techniques to compute sample uncertainty in a continuous feature space (rather than with discrete labels as proposed by Rainbow [33]) by exploiting the feature clustering aspect of my algorithm and utilising image augmentations to improve the quality of the memory buffer.

I implemented, compared, and evaluated a wide variety of continual learning techniques ranging from regularisation techniques, such as EWC [20], through to the latest state-of-the-art approaches, such as L2P, alongside my proposed novel approach using both CIFAR-10 and CIFAR-100 with a variety of different maximum buffer sizes to evaluate the effect that this hyperparameter has on the performance of the relevant approaches. Additional care has been taken to ensure that the experimental setups used are both challenging and realistic, this was achieved by using the online class-incremental setup instead of the easier task-incremental approach which has been criticised previously within the literature [16]. Correspondingly, I show that my novel approach is competitive and consistently outperforms the state-of-the-art techniques in terms of classification performance.

I expect future work will continue to explore the use of pre-trained ViTs due to their availability and power, in contrast to training a ResNet directly from scratch. Furthermore, the use of class-balanced memory buffers has been common throughout the literature. I anticipate that future work may utilise sample uncertainty quantification to create imbalanced buffers and provide uncertain classes (from the perspective of the model) with a greater proportion of the sample memory space while providing more certain classes with a reduced amount in order to use the available memory more effectively. This would need be analysed in further detail and careful consideration would need to be given to discarding samples from the memory buffer as they would be lost permanently. Finally, I think it is likely that future work may put greater focus on to the image augmentations

that are used and carefully tailor these to improve the performance of sample uncertainty quantification.

Continual learning will continue to be of importance in the future as the quality of deep learning models improves across a multitude of domains. For example, the recent release of ChatGPT has showcased advances in natural language processing. However, one of its major limitations is that the data used to train it is limited to 2021 and before due to the dataset that it has been trained on [82]. The capability to update impressive large-scale models will be vital into the future as models become deeper and more complex, thus requiring greater resources to train. This is further emphasised by the expected race to utilise large-scale language models to answer search engine queries which would require up-to-date information [83]. Continual learning offers opportunities to achieve this goal in the future.

Ultimately, I present a successful, novel approach to the continual learning problem that is capable of alleviating the effects of catastrophic forgetting and I carry out a variety of experiments to evaluate its performance in a range of challenging scenarios.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[2] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Citeseer*, 2009.

[3] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[4] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[5] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[6] "Amazon prime day 2020 marked the two biggest days ever for small and medium businesses in amazon's stores worldwide," Oct 2020. [Online]. Available: https://press.aboutamazon.com/2020/10/amazon-prime-day-2020-marked-the-two-biggest-days-ever-for-small-medium-businesses-in-amazons-stores-worldwide

[7] L. Ceci and A. 4, "Youtube: Hours of video uploaded every minute 2020," Apr 2022. [Online]. Available: https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/

[8] R. Hecht-Nielsen, "Neurocomputing: picking the human brain," *IEEE spectrum*, vol. 25, no. 3, pp. 36–41, 1988.

[9] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[10] S. I. Mirzadeh, A. Chaudhry, D. Yin, T. Nguyen, R. Pascanu, D. Gorur, and M. Farajtabar, "Architecture matters in continual learning," *arXiv preprint arXiv:2202.00275*, 2022.

[11] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," p. 504, 2013.

[12] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti *et al.*, "Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model," *arXiv preprint arXiv:2201.11990*, 2022.

[13] K. Wiggers, "Ai weekly: Ai model training costs on the rise, highlighting need for new solutions," Oct 2021. [Online]. Available: https://venturebeat.com/2021/10/15/ai-weekly-ai-model-training-costs-on-the-rise-highlighting-need-for-new-solutions/

[14] M. Biesialska, K. Biesialska, and M. R. Costa-Jussa, "Continual lifelong learning in natural language processing: A survey," *arXiv preprint arXiv:2012.09823*, 2020.

[15] S. Sadhu and H. Hermansky, "Continual learning in automatic speech recognition." in *Interspeech*, 2020, pp. 1246–1250.

[16] A. Prabhu, P. H. Torr, and P. K. Dokania, "Gdumb: A simple approach that questions our progress in continual learning," in *European conference on computer vision*. Springer, 2020, pp. 524–540.

[17] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[18] J. Xu and Z. Zhu, "Reinforced continual learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[19] Z. Li, M. Meng, Y. He, and Y. Liao, "Continual learning with laplace operator based node-importance dynamic architecture neural network," in *International Conference on Neural Information Processing*. Springer, 2021, pp. 52–63.

[20] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[21] S. Das, J. C. Spall, and R. Ghanem, "Efficient monte carlo computation of fisher information matrix using prior information," in *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*, 2007, pp. 242–249.

[22] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.

[23] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[24] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 532–547.

[25] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.

[26] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[27] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.

[28] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *arXiv preprint arXiv:1812.00420*, 2018.

[29] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[30] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 233–248.

[31] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[32] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial shapley value," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9630–9638.

[33] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, "Rainbow memory: Continual learning with a memory of diverse samples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8218–8227.

[34] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," *Advances in neural information processing systems*, vol. 33, pp. 15 920–15 930, 2020.

[35] Y. Guo, B. Liu, and D. Zhao, "Online continual learning through mutual information maximization," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 8109–8126. [Online]. Available: https://proceedings.mlr.press/v162/guo22g.html

[36] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014.

[38] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[39] C. G. Willcocks, "Lectures on deep learning and reinforcement learning," URL: https://cwkx.github.io/teaching.html, 2021.

[40] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," *Advances in neural information processing systems*, vol. 32, 2019.

[41] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[42] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2624–2637, 2013.

[43] R. Kemker and C. Kanan, "Fearnet: Brain-inspired model for incremental learning," *arXiv preprint arXiv:1711.10563*, 2017.

[44] T. L. Hayes, K. Kafle, R. Shrestha, M. Acharya, and C. Kanan, "Remind your neural network to prevent catastrophic forgetting," in *European Conference on Computer Vision*. Springer, 2020, pp. 466–483.

[45] K. Wang, J. van de Weijer, and L. Herranz, "Acae-remind for online continual learning with compressed feature replay," *Pattern Recognition Letters*, vol. 150, pp. 122–129, 2021.

[46] X. Liu, C. Wu, M. Menta, L. Herranz, B. Raducanu, A. D. Bagdanov, S. Jui, and J. v. de Weijer, "Generative feature replay for class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 226–227.

[47] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.

[48] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *arXiv preprint arXiv:2107.13586*, 2021.

[49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[50] H. Cha, J. Lee, and J. Shin, "Co2l: Contrastive continual learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9516–9525.

[51] Z. Mai, R. Li, H. Kim, and S. Sanner, "Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3589–3599.

[52] L. Weng, "Contrastive representation learning," *lilianweng.github.io*, May 2021. [Online]. Available: https://lilianweng.github.io/posts/2021-05-31-contrastive/

[53] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[54] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing change: Continual learning in deep neural networks," *Trends in cognitive sciences*, vol. 24, no. 12, pp. 1028–1040, 2020.

[55] K. Javed and M. White, "Meta-learning representations for continual learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[56] M. Banayeeanzade, R. Mirzaiezadeh, H. Hasani, and M. Soleymani, "Generative vs. discriminative: Rethinking the meta-continual learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 592–21 604, 2021.

[57] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," *arXiv preprint arXiv:1810.11910*, 2018.

[58] G. Gupta, K. Yadav, and L. Paull, "La-maml: Look-ahead meta learning for continual learning. 2020," *URL https://arxiv. org/abs*, 2020.

[59] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, and Q. Sun, "Mnemonics training: Multi-class incremental learning without forgetting," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 245–12 254.

[60] A. Chaudhry, A. Gordo, P. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6993–7001.

[61] Y. Zhou, E. Nezhadarya, and J. Ba, "Dataset distillation using neural feature regression," *arXiv preprint arXiv:2206.00719*, 2022.

[62] J. KJ and V. N Balasubramanian, "Meta-consolidation for continual learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 374–14 386, 2020.

[63] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[64] J. Lindsey and A. Litwin-Kumar, "Learning to learn with feedback and local plasticity," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 213–21 223, 2020.

[65] E. Oja, "Simplified neuron model as a principal component analyzer," *Journal of mathematical biology*, vol. 15, no. 3, pp. 267–273, 1982.

[66] A. Chaudhry, "Continual learning for efficient machine learning," Ph.D. dissertation, University of Oxford, 2020.

[67] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 831–839.
[68] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. v. d. Weijer, "Semantic drift compensation for class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6982–6991.
[69] T. Lesort, A. Stoian, and D. Filliat, "Regularization shortcomings for continual learning," *arXiv preprint arXiv:1912.03049*, 2019.
[70] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.
[71] J. Knoblauch, H. Husain, and T. Diethe, "Optimal continual learning has perfect memory and is np-hard," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5327–5337.
[72] S. I. Mirzadeh, A. Chaudhry, D. Yin, H. Hu, R. Pascanu, D. Gorur, and M. Farajtabar, "Wide neural networks forget less catastrophically," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 699–15 717.
[73] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
[74] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
[75] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.
[76] S. Farquhar and Y. Gal, "Towards robust evaluations of continual learning," *arXiv preprint arXiv:1805.09733*, 2018.
[77] R. Pascanu and Y. Bengio, "Revisiting natural gradient for deep networks," *arXiv preprint arXiv:1301.3584*, 2013.
[78] S. Guerriero, B. Caputo, and T. Mensink, "Deepncm: Deep nearest class mean classifiers," *openreview*, 2018.
[79] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, 2020.
[80] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," *arXiv preprint arXiv:2104.10972*, 2021.
[81] G. Cybenko, D. P. O'Leary, and J. Rissanen, *The Mathematics of Information Coding, Extraction and Distribution*. Springer Science & Business Media, 1998, vol. 107.
[82] [Online]. Available: https://help.openai.com/en/articles/6783457-chatgpt-general-faq
[83] C. Stokel-Walker, "Ai chatbots are coming to search engines – can you trust the results?" Feb 2023. [Online]. Available: https://www.nature.com/articles/d41586-023-00423-4