GRASP: A REHEARSAL POLICY FOR EFFICIENT ONLINE CONTINUAL LEARNING

Md Yousuf Harun
Rochester Institute of Technology
United States of America
mh1023@rit.edu

Jhair Gallardo
Rochester Institute of Technology
United States of America
gg4099@rit.edu

Junyu Chen
University of Rochester
United States of America
jchen175@ur.rochester.edu

Christopher Kanan
University of Rochester
United States of America
ckanan@cs.rochester.edu

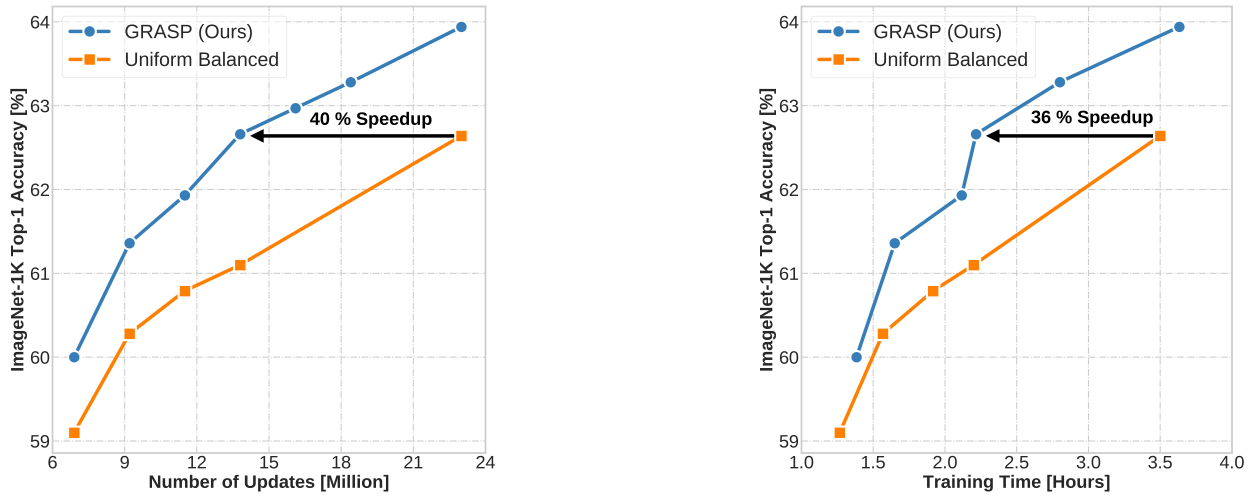
ABSTRACT

Continual learning (CL) in deep neural networks (DNNs) involves incrementally accumulating knowledge in a DNN from a growing data stream. A major challenge in CL is that non-stationary data streams cause catastrophic forgetting of previously learned abilities. A popular solution is rehearsal: storing past observations in a buffer and then sampling the buffer to update the DNN. Uniform sampling in a class-balanced manner is highly effective, and better sample selection policies have been elusive. Here, we propose a new sample selection policy called GRASP that selects the most prototypical (easy) samples first and then gradually selects less prototypical (harder) examples. GRASP has little additional compute or memory overhead compared to uniform selection, enabling it to scale to large datasets. Compared to 17 other rehearsal policies, GRASP achieves higher accuracy in CL experiments on ImageNet. Compared to uniform balanced sampling, GRASP achieves the same performance with 40% fewer updates. We also show that GRASP is effective for CL on five text classification datasets. Source code for GRASP is available at https://yousuf907.github.io/graspsite.

1 INTRODUCTION

In deep continual learning (CL), a deep neural network (DNN) is sequentially updated from a growing data stream, where the distribution of the data stream is unknown. When the stream is non-stationary, catastrophic forgetting of previously learned abilities occurs if the DNN is progressively fine-tuned with only new samples. CL methods aim to overcome this obstacle. One of the best CL methods is rehearsal (e.g., experience replay) (van de Ven et al., 2022; Zhou et al., 2023). Rehearsal is highly effective across CL scenarios, and it is especially effective for class incremental learning (CIL), where classes must be learned sequentially (Rebuffi et al., 2017). Rehearsal methods store a subset of old examples, or representations of those examples, in a buffer and then update the DNN with a chosen mixture of new and old data. A *rehearsal policy* governs which samples are selected. The most common approach is uniform sampling, but better policies can potentially reduce the time required for rehearsal. While some have been shown to reduce the total number of updates needed on small-scale CL problems (Aljundi et al., 2019a;b; Bang et al., 2021; Yoon et al., 2022; Koh et al., 2021; Shim et al., 2021), sampling is expensive, resulting in no improvement in the total amount of time required for training. One could simply use more updates with uniform selection.

For large-scale problems, uniform selection has been shown to outperform more sophisticated policies (Harun et al., 2023b; Prabhu et al., 2023a). It is also a highly effective strategy for active learning and dataset pruning (Sorscher et al., 2022; Evans et al., 2023). Uniform selection is particularly effective in CL settings where the buffer is highly constrained, unlike compute. This is because, over a large number of training steps, uniform selection will eventually achieve optimal accuracy. In such cases, an ideal selection policy may not provide added value. However, when compute is limited and the buffer is large, only a subset of examples can be chosen due to computational constraints. In this scenario, a sample selection strategy plays a crucial role in achieving optimal accuracy over a small number of training steps. In this paper, we aim to identify a computationally efficient rehearsal policy that maximizes DNN accuracy with a fixed number of updates and a relatively large buffer. Our setting also aligns with the industry, where computational costs significantly exceed storage costs (Prabhu et al., 2023a).



(a) Compute Comparison

(b) Time Comparison

Figure 1: GRASP achieves the best accuracy of the popular uniform balanced policy while requiring 40% fewer gradient descent updates and 36% less training time for CIL with SIESTA on ImageNet-1K.

Our method, GRASP, is inspired by dataset pruning, where the goal is to identify a sample subset that when trained on has the same performance as the entire dataset. Sorscher et al. (2022) showed that the optimal pruning strategy varied based on the size of the dataset. Based on similarity to class prototypes, they found retaining the easiest samples was the best strategy for small datasets, whereas for large datasets keeping the hardest samples was best. These observations were made based on having a fixed dataset, where the size of the dataset is known beforehand. In contrast, in CL, the size of the dataset and the number of samples for each class change over time. This suggests that an effective rehearsal policy should adapt to the amount of currently available data for a class.

This paper makes the following key contributions:

1. We propose GRASP, a dynamic rehearsal policy that initially selects easy samples and gradually selects harder ones. GRASP is compute and memory efficient. We integrated GRASP into three rehearsal-based CL systems, SIESTA (Harun et al., 2023b), DERpp (Buzzega et al., 2020), and GDumb (Prabhu et al., 2020).
2. In CIL experiments on ImageNet, GRASP outperforms 17 other rehearsal policies, including class-balanced uniform selection.
3. We show that GRASP is effective across CL distributions, including independent and identically distributed (IID) and long-tailed distributions.
4. We demonstrate that GRASP is effective for natural language processing (NLP) on 5 CL benchmarks.

2 Related Work

2.1 Rehearsal-Based Continual Learning

Rehearsal is inspired by neuroscience, where recent memories are stored in the hippocampus and then reactivated for long-term consolidation (O'Neill et al., 2010; Hayes et al., 2021). Most CL methods use a rehearsal buffer where the maximum size is constrained (Hayes et al., 2021; Chaudhry et al., 2019b; Abraham & Robins, 2005; Belouadah & Popescu, 2019; Castro et al., 2018; Chaudhry et al., 2018b;a; Hayes et al., 2019; 2020; Harun et al., 2023b; Rebuffi et al., 2017; Tao et al., 2020; Wu et al., 2019; Aljundi et al., 2019b). These constraints are often arbitrary and require having a policy for maintaining the size constraint (see Sec. 2.2). However, some early CL methods used *cumulative rehearsal* where rehearsal buffers store all previously observed samples (Gepperth & Karaoguz, 2016; Hayes et al., 2019; Kemker et al., 2018). Recently, there has been a resurgence of interest in this setting because it allows one to focus on how to utilize the buffer best, e.g., to minimize training time while maximizing accuracy (Al Kader Hammoud et al., 2023; Prabhu et al., 2023a;b; Harun & Kanan, 2023; Verwimp et al., 2024). We study both settings in this paper.

Rehearsal methods fall into three categories: veridical, latent, and generative. In **veridical rehearsal**, raw input data is stored in a memory buffer for later rehearsal (Rebuffi et al., 2017; Chaudhry et al., 2019b; Castro et al., 2018; Lopez-Paz & Ranzato, 2017; Bang et al., 2021; Chaudhry et al., 2018a; Wu et al., 2019; Aljundi et al., 2019b). Instead of

Table 1: **Advanced Rehearsal Policies.** Current state-of-the-art methods except MIR and ASER mainly proposed buffer maintenance policy and used uniform random as rehearsal policy.

Method	Buffer Policy	Rehearsal Policy	Metric	Expensive	Scalable
MIR (Aljundi et al., 2019a)	✗	✔	MIR Loss	✔	✗
Rainbow Memory (Bang et al., 2021)	✔	✗	Uncertainty	✔	✗
OCS (Yoon et al., 2022)	✔	✗	OCS	✔	✗
GSS (Aljundi et al., 2019b)	✔	✗	Grad Variance	✔	✗
Grad Matching (Campbell & Broderick, 2019)	✔	✗	Gradient	✔	✗
Bi-level (Borsos et al., 2020)	✔	✗	Bi-level Opt	✔	✗
CLIB (Koh et al., 2021)	✔	✗	Max Loss	✔	✗
ASER (Shim et al., 2021)	✔	✔	Shapley Value	✔	✗
GRASP (Ours)	✗	✔	Cosine Distance	✗	✔

storing raw images, **latent rehearsal** methods store features from hidden layers (Hayes et al., 2020; Iscen et al., 2020; Caccia et al., 2020; Pellegrini et al., 2020; Zhao et al., 2021; Harun et al., 2023b), allowing them to store far more samples than veridical under the same memory budget. If storing data is prohibited, **generative rehearsal** methods produce synthetic images or features to retrieve old knowledge (Shin et al., 2017; He et al., 2018; Hu et al., 2018; Liu et al., 2020; Kemker & Kanan, 2018; Ostapenko et al., 2019; Xiang et al., 2019) by incorporating a generator into the DNN; however, these methods increase compute and often struggle with feature drift. While most work on rehearsal has focused on images, these three types have also been used for CL in NLP (Ke & Liu, 2022; Biesialska et al., 2020). In our work, we conduct experiments in both the veridical and latent rehearsal settings.

2.2 Rehearsal & Buffer Maintenance Policies

Rehearsal algorithms alternate between a sample acquisition phase, where samples are added to a buffer, and a rehearsal phase, where the DNN is updated using the buffer and newly acquired samples. All rehearsal algorithms must then define a *Rehearsal Policy* for what should be sampled from the buffer. For memory-constrained rehearsal, this is often entangled with what we call the *Buffer Maintenance Policy*, which defines what should be kept within the buffer. In many memory-constrained methods, the entire buffer is used to update the network, where the rehearsal policy is then implicitly governed by the criteria used to determine what is kept within the buffer. This work aims to disentangle these two aspects by converting buffer maintenance policies into rehearsal policies to compare approaches. We summarize the properties of recent methods in Table 1.

Rehearsal Policies. While buffer maintenance policies have been heavily explored in CIL, much less work has been done to identify which stored samples should be selected for rehearsal (see Table 1). We briefly describe the rehearsal policies that have been studied. Prior CL work (Hayes & Kanan, 2021; Chaudhry et al., 2018a; Harun et al., 2023b; Prabhu et al., 2023a) studied a variety of sampling policies e.g., uniform balanced, min rehearsal, max loss, min margin, min logit-distance, and min confidence; however, these policies showed limited efficacy for large-scale datasets. More details about these policies are given in Sec 4.3. More advanced rehearsal policies for CL include MIR and ASER. In MIR, when a new batch of data arrives, virtual updates are made to the DNN using the new batch to find the maximally interfered old samples (Aljundi et al., 2019a). Next, it updates DNN using the new batch mixed with interfered old data. Virtual updates are computationally expensive, and MIR disproportionately prioritizes redundant samples in the most interfered category (Shim et al., 2021). ASER uses Shapley values to prioritize representative samples for storage and interfered samples for rehearsal (Shim et al., 2021). ASER uses uniform random sampling to construct evaluation and candidate sets to reduce computational overhead, and it is difficult to scale since it computes the Euclidean distance of each candidate sample from each evaluation sample at every training iteration. Recently, Prabhu et al. (2023a) compared many rehearsal policies and found that balanced uniform outperforms others for large-scale datasets.

Buffer Maintenance Policies. The most commonly used method is reservoir sampling, where a new sample overwrites a randomly selected sample from the buffer once the buffer is full, and then samples are chosen uniformly from the buffer for rehearsal. This strategy is used in both vision (Wang et al., 2023; Riemer et al., 2019; Chaudhry et al., 2019a) and NLP (Ke & Liu, 2022; Biesialska et al., 2020). More advanced strategies exploit the statistics of the stored samples e.g., GDumb (Prabhu et al., 2020), ExStream (Hayes et al., 2019), ring buffer (Lopez-Paz & Ranzato, 2017), herding-based (Rebuffi et al., 2017), k-Means (Chaudhry et al., 2019a), MoF (Chaudhry et al., 2019b), and FIFO (Lopez-Paz & Ranzato, 2017). An alternative to data-driven methods are model-based methods that determine what to retain in the buffer by analyzing the DNN's behavior on the stored samples (Chaudhry et al., 2018a; Koh et al., 2021; Borsos et al., 2020; Campbell & Broderick, 2019; Aljundi et al., 2019b; Yoon et al., 2022). For example, rainbow memory (Bang et al., 2021) stores diverse samples based on classification uncertainty and image augmentation.

However, model-based methods, especially gradient-based methods, are computationally expensive and intractable for large-scale datasets e.g., ImageNet-1K. For instance, OCS (Yoon et al., 2022) is computationally demanding since at every training iteration it scores data based on mini-batch gradient similarity and cross-batch gradient diversity. See Wang et al. (2023) for a review. We repurposed buffer maintenance policies for rehearsal policies to compare them with GRASP. More details are given in Sec 4.3.

3 The GRASP Rehearsal Policy

We propose a new rehearsal policy named GRASP (**GRA**dually **S**elect less **P**rototypical). GRASP is simple, scalable, hyperparameter-free, and has little additional compute or memory overhead compared to uniform selection. Our goal is to identify a rehearsal policy that minimizes the number of DNN updates to gain computational efficiency.

GRASP is based on the hypothesis that choosing only easy or hard samples are both suboptimal and that the DNN would benefit from a curriculum that combines both. GRASP first selects the most prototypical (easy) samples from the buffer and then gradually selects harder samples, where easy samples are closest to the class mean and hard samples are farthest. While prior work has explored policies that select samples close to prototypes for buffer maintenance (Rebuffi et al., 2017; Chaudhry et al., 2019b), their policies are biased toward easy samples near class prototypes instead of progressively choosing hard samples as well.

Algorithm 1 GRASP Rehearsal Policy

Require: $\mathcal{M} = \{(\mathcal{X}, \mathcal{D})_k\}_{k=1}^{K}$ ▷ Memory buffer
  $\mathcal{R} \leftarrow \{\}$ ▷ Will contain $\mathcal{U}$ selected samples
  $c \leftarrow 0$ ▷ Initialize counter
  $\mathcal{U} = n \times b$ ▷ Compute Budget
  while $c < \mathcal{U}$ **do**
    for k = 1 to K **do**
      $\mathcal{X}_k, \mathcal{D}_k \leftarrow \mathcal{M}$ ▷ Obtain data for class k
      $P_k = \text{normalize}\left(\mathcal{D}_k^{-1}\right)$ ▷ Compute probabilities
      $m \overset{1}{\sim} \text{sample}(\mathcal{X}_k, P_k)$
      $\mathcal{D}_k[m] \leftarrow \mathcal{D}_k[m] + \max(\mathcal{D}_k)$ ▷ Virtual update
      $(\mathbf{x}, k) \leftarrow \mathcal{X}_k[m]$ ▷ Obtain sample
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\mathbf{x}, k)\}$ ▷ Add sample
      $c \leftarrow c + 1$ ▷ Update counter
      if $c \geq \mathcal{U}$ **then**
        break
      end if
    end for
  end while
  for t = 1 to b **do**
    $B_t \leftarrow \mathcal{R}[(t-1)n : tn]$ ▷ Select a mini-batch of size n
    $\theta \leftarrow \text{SGD}(\theta, B_t)$ ▷ Update model on n samples
  end for ▷ Rehearsal cycle ends

GRASP can be integrated into both online and offline rehearsal-based CL methods, and it can be used with either veridical or latent rehearsal. In all cases, we assume that the CL algorithm maintains a buffer containing both old and new samples. In offline CL memory constraints on the buffer only apply to old samples, where all new samples are added to it before rehearsal, and then after rehearsal, the buffer is compressed to the given memory budget. Severe catastrophic forgetting in minority classes can occur in rehearsal due to class-imbalance (Wu et al., 2019), which GRASP overcomes by selecting samples in a class-balanced way, where minority classes can be oversampled.

Sample Acquisition Phase. For the CL algorithms we study, a sequence of labeled samples are provided to the learner over time, where after a sufficiently large number of samples arrive the learner uses rehearsal to update the DNN. To simplify notation, we assume at time t the learner receives a single sample X_t with label k_t, where X_t is the raw input in veridical rehearsal or an embedding from the middle of the network in latent rehearsal. The sample is then added to the buffer, where a sample from the largest class is removed from the buffer if it is full.

After buffer maintenance, we compute the distances between stored samples and class prototypes for GRASP. Let $\mathbf{z} \in \mathbb{R}^d$ be the embedding computed by the DNN from the penultimate layer. For each class k, a class prototype vector $\mathbf{q}_k$ is computed by averaging the penultimate embedding vectors of corresponding samples $\mathcal{X}_k$, i.e., $\mathbf{q}_k = \frac{1}{J}\sum_{j=1}^{J} \mathbf{z}_{k,j}$. Next for each sample, the cosine distance d between its penultimate embedding and the class prototype is calculated as $d = 1 - (\mathbf{z} \cdot \mathbf{q}_k)/(\|\mathbf{z}\|_2 \|\mathbf{q}_k\|_2)$. The distance d is used during rehearsal to select samples based on how far they are from the prototype and is stored in the buffer. Therefore, the buffer consists of $\mathcal{M} = \{(\mathcal{X}, \mathcal{D})_k\}_{k=1}^{K}$, where K is the total number of classes that have been seen, $\mathcal{X}_k$ is the set of stored inputs from class k for veridical rehearsal or stored embeddings for latent rehearsal, and $\mathcal{D}_k$ has the cached distances to the class prototypes.

Rehearsal Phase. Because the focus of this work is computational efficiency, the rehearsal phase is given a compute budget of $\mathcal{U} = nb$ gradient updates to the DNN, where b is the total number of minibatches and n is the minibatch size. GRASP iteratively selects $\mathcal{U}$ samples from the buffer. For class k, GRASP assigns a selection probability P_k inversely proportional to $\mathcal{D}_k$ for all the data points in class k. The selected sample's distance is then temporarily

updated by adding $\max \mathcal{D}_k$ to it. This makes it unlikely for that sample to be chosen again in the rehearsal session or task because it will have the lowest probability; therefore, within a rehearsal session, GRASP chooses progressively less prototypical samples for rehearsal. Approximately the same number of samples are chosen from each class, where oversampling can occur for classes with fewer samples. After the rehearsal session ends and a new session begins, the distances among samples and class prototypes, $\mathcal{D}_k$ are recomputed. Pseudocode for GRASP is given in Algorithm 1.

4 Experimental Setup

In experiments, we study rehearsal policies in both the unbounded memory and the conventional memory-bounded settings. The unbounded setting enables us to focus on the efficacy of rehearsal policies regardless of the amount of storage permitted or the choice of buffer maintenance policy. Moreover, for industrial applications, the cost of deep learning largely depends on computing rather than storage. The unlimited memory setting has been studied in recent CL papers that have argued it is better aligned with industry needs (Harun et al., 2023a; Harun & Kanan, 2023; Prabhu et al., 2023a;b; Bornschein et al., 2022; Al Kader Hammoud et al., 2023; Verwimp et al., 2024).

4.1 CL Datasets and DNN Architectures

To show scalability to large-scale datasets, our main results use **ImageNet** (Russakovsky et al., 2015). We conduct CIL experiments (Non-IID ordering) with the 1000 class version of the dataset (ImageNet-1K) and two variants with 150 and 300 classes, referred to as ImageNet-150 and ImageNet-300 respectively. Long-tailed datasets are challenging in CIL, and to assess rehearsal policies in this setting we use **Places-LT** (Liu et al., 2019). Additionally, we study IID orderings on ImageNet-1K to assess the robustness of rehearsal policies to non-adversarial data streams. More dataset details are included in Appendix C. Continual text classification performs task incremental learning. Following Huang et al. (2021), we use five large-scale benchmark text datasets: **AG News** (news classification), **Yelp** (sentiment analysis), **DBPedia** (Wikipedia article classification), **Amazon** (sentiment analysis), and **Yahoo! Answer** (Q&A classification). See Huang et al. (2021) for details. Methods are evaluated on 6 task sequences.

For our main image classification experiments, we use **MobileNetV3-Large** (Howard et al., 2019) since it outperforms widely used ResNet-18 (Harun et al., 2023b) and has lower latency. Following Harun et al. (2023b), we pre-trained MobileNetV3-Large on the first 100 classes from ImageNet-1K using SwAV (Caron et al., 2020) for all experiments. In Appendix F, we also conduct experiments on a vision transformer **MobileViT-Small** (Mehta & Rastegari, 2021). We use **BERT** (Kenton & Toutanova, 2019) in continual text classification. Implementation details are in Appendix B.

4.2 Rehearsal-Based CL Algorithms

We study the rehearsal policies in three algorithms for large-scale CL in vision and one for CL in NLP:

1. **SIESTA** is a state-of-the-art latent rehearsal CL algorithm that alternates between online and offline phases (Harun et al., 2023b). SIESTA is initialized on the first 100 classes of ImageNet-1K using self-supervised learning (Gallardo et al., 2021). During its online phase, it only updates its output layer and stores quantized tensor embeddings of the input images. During its offline phase (every 100 ImageNet classes), it uses latent rehearsal to update its non-frozen layers with a fixed number of updates. We also do experiments with a variant that uses veridical rehearsal. We use SIESTA for our main experiments because it requires only 2 hours to finish training on ImageNet-1K on a single GPU and because it matches an offline model's accuracy on ImageNet-1K in the augmentation-free setup.
2. **Dark Experience Replay (DERpp)** combines rehearsal with knowledge distillation and regularization (Buzzega et al., 2020). It uses a distillation loss on logits of old samples for consistency. It uses reservoir sampling to maintain a constrained memory buffer.
3. **GDumb** uniformly removes a sample from the largest class upon arrival of a new sample when memory buffer is full (Prabhu et al., 2020).
4. **IDBR** is a CL system for text classification (Huang et al., 2021). Using rehearsal, it continually updates a BERT text encoder and a linear classification layer. In the original IDBR system, it also updated regularization sub-networks for retaining task-generic information and for adapting to task-specific information for regularization. We exclude these sub-networks to focus on rehearsal.

In our experiments, all methods use a fixed number of rehearsal updates. Experiments omit image augmentation to focus on comparing rehearsal policies.

Our main results use SIESTA (Harun et al., 2023b). SIESTA is pre-trained on the first 100 ImageNet classes. Subsequently, it learns 50/ 200/ 900 additional classes continually. For SIESTA with MobileNetV3-Large, the first 8 layers are frozen, and the remaining layers (97.81% of parameters) are updated during latent rehearsal. For SIESTA, DERpp, and GDumb, we use versions with latent and veridical rehearsal. We created latent rehearsal versions of DERpp and GDumb using SIESTA's setup.

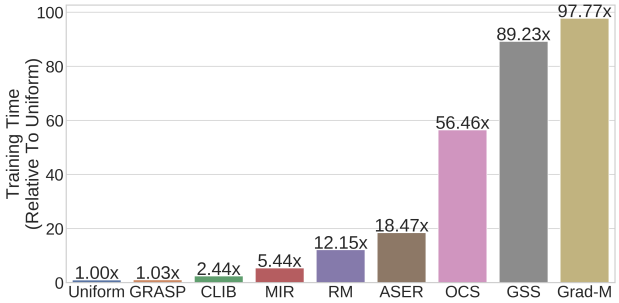


Figure 2: Unlike GRASP, existing state-of-the-art methods require significantly longer training time than uniform.

Memory Constraints. We study CL with both unbounded memory (access to the entire dataset) and bounded memory settings. Following Hayes et al. (2020), all methods are limited to 1.5 GB of storage for ImageNet-1K. This corresponds to 10000 old images for veridical rehearsal, which excludes the newly acquired batch of 120000 images. For ImageNet-150/300 subset, latent rehearsal methods use up to 0.2 GB for the buffer. For IDBR, we randomly select 50% of seen examples to store in the memory buffer. Samples are removed from the largest class/task to maintain a bounded rehearsal buffer.

Compute Constraints. Rehearsal policies use a fixed computational budget ($\mathcal{U} = nb$) that indicates the total number of samples used for backpropagation during rehearsal. Following SIESTA (Harun et al., 2023b), in our main results on ImageNet-300 we set the number of iterations per rehearsal session to 1251 (n) with a mini-batch size 512 (b) for all methods. In ImageNet-150 experiments, we use 500 iterations per rehearsal session with a mini-batch size 64. In ImageNet-1K experiments, we use 2502 iterations per rehearsal session with a mini-batch size 512. Details for other settings are given in Appendix B. For continual text classification with IDBR, we bound compute by using rehearsal sessions of a fixed length, whereas the original IDBR increased the amount of compute as new tasks were learned.

4.3 Compared Rehearsal Policies and Evaluation Criteria

Here, we briefly describe all the rehearsal policies that we consider as baselines. Note that we convert buffer maintenance policies into rehearsal policies by using them to score stored samples for selection during rehearsal. All baseline methods use the same buffer maintenance policy as described in Sec. 3 where a randomly chosen old sample from the largest class is removed to accommodate a new sample when the buffer is full.

1. **Uniform:** samples are selected uniformly at random (Vitter, 1985).
2. **Uniform Balanced:** samples are selected uniformly at random with an equal number of samples per category (Prabhu et al., 2020). This is a strong baseline for large-scale CL (Prabhu et al., 2023a).
3. **Min Rehearsal:** samples with least rehearsal count(s) are most likely to be selected (Hayes & Kanan, 2021).
4. **Max Loss:** samples with higher (lower) cross-entropy loss are defined as hard (easy) samples and prioritized for selection (Kawaguchi & Lu, 2020).
5. **Min Margin:** easy and hard examples are defined with margins, where the margin is the difference between the predicted and correct class probabilities. Hard samples are more likely to be chosen (Scheffer et al., 2001).
6. **Min Logit-Distance:** samples closer to the decision boundary are selected (Chaudhry et al., 2018a).
7. **Min Confidence:** samples with lower DNN confidence (softmax scores) are prioritized (Gal et al., 2017).
8. **k-Means:** features from the penultimate layer are clustered into k centroids with k-Means. Samples closer to centroids are more likely to be sampled (Chaudhry et al., 2019b; Prabhu et al., 2023a).
9. **MoF / Easy Biased:** prioritizes samples near the class mean (Chaudhry et al., 2019b).
10. **Hard Biased:** prioritizes samples far from class mean.
11. **Rainbow Memory:** keeps class diverse examples based on uncertainty and augmentation (Bang et al., 2021).
12. **MIR:** prioritizes old samples from buffer that maximally interfere with new samples (Aljundi et al., 2019a).
13. **CLIB:** prioritizes samples with maximum loss decrease (Koh et al., 2021).
14. **ASER:** selects samples based on adversarial Shapley values (SVs) (Shim et al., 2021). Positive SVs w.r.t. memory samples indicate representative samples whereas negative SVs w.r.t. new samples indicate interfered samples.
15. **OCS:** uses mini-batch gradient similarity and cross-batch gradient diversity for sample selection (Yoon et al., 2022).
16. **GSS:** frames the sample selection as a constraint selection problem to maximize the variance of gradient direction (Aljundi et al., 2019b).
17. **Grad Matching:** selects samples using Hilbert coreset (Campbell & Broderick, 2019).

Evaluation Criteria. For evaluation, we use the average accuracy μ over all rehearsal sessions T, where $\mu = \frac{1}{T}\sum_{t=1}^{T}\alpha_t$, with α_t denoting the accuracy at rehearsal session t. We use μ_N, μ_O, and μ_A to denote average accuracy

Table 2: **GRASP vs. Various Rehearsal Methods.** This uses latent rehearsal for CIL with SIESTA on **ImageNet-300**. μ_A denotes accuracy (%) averaged over rehearsals, and α is the final accuracy (%). Training time T is in hours.

Method	Unbounded Memory		Bounded Memory		Time
	$\mu_A \uparrow$	$\alpha \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$	$T \downarrow$
Uniform	77.46	72.26	75.36	68.17	0.30
Min Confidence (Gal et al., 2017)	77.69	72.42	75.11	67.79	0.41
Min Margin (Scheffer et al., 2001)	77.51	72.59	75.20	67.61	0.41
Max Loss (Kawaguchi & Lu, 2020)	75.40	68.89	72.96	64.55	0.41
Min Logit Dist (Chaudhry et al., 2018a)	77.36	72.07	75.19	67.85	0.41
k-Means (Chaudhry et al., 2019b)	77.63	72.56	75.50	68.33	0.56
Min Rehearsal (Hayes & Kanan, 2021)	75.76	69.76	74.87	67.37	0.41
MoF / Easy Biased (Chaudhry et al., 2019b)	77.43	71.99	75.54	68.60	0.33
Hard Biased	76.90	71.89	74.75	67.59	0.33
Uniform Balanced	77.52	72.62	75.39	68.11	0.32
Rainbow Memory (Bang et al., 2021)	74.93	68.43	72.73	64.67	3.89
MIR (Aljundi et al., 2019a)	77.33	71.35	75.30	67.93	1.74
CLIB (Koh et al., 2021)	77.44	72.21	75.22	67.89	0.78
ASER (Shim et al., 2021)	75.16	69.09	73.79	66.09	5.91
GRASP (Ours)	**78.39**	**73.65**	**76.12**	**69.06**	0.33

Table 3: **GRASP vs. Gradient-Based Methods.** This uses latent rehearsal for CIL with SIESTA on **ImageNet-150**. μ_A denotes accuracy (%) averaged over rehearsals, and α is the final accuracy (%). Training time T is in hours.

Method	Unbounded Memory		Bounded Memory		Time
	$\mu_A \uparrow$	$\alpha \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$	$T \downarrow$
OCS (Yoon et al., 2022)	74.93	70.36	74.13	69.01	7.34
GSS (Aljundi et al., 2019b)	75.04	70.25	75.38	70.47	11.60
Grad Matching (Campbell & Broderick, 2019)	76.48	72.20	76.71	72.36	12.71
GRASP (Ours)	**77.75**	**73.96**	**77.88**	**73.71**	0.13

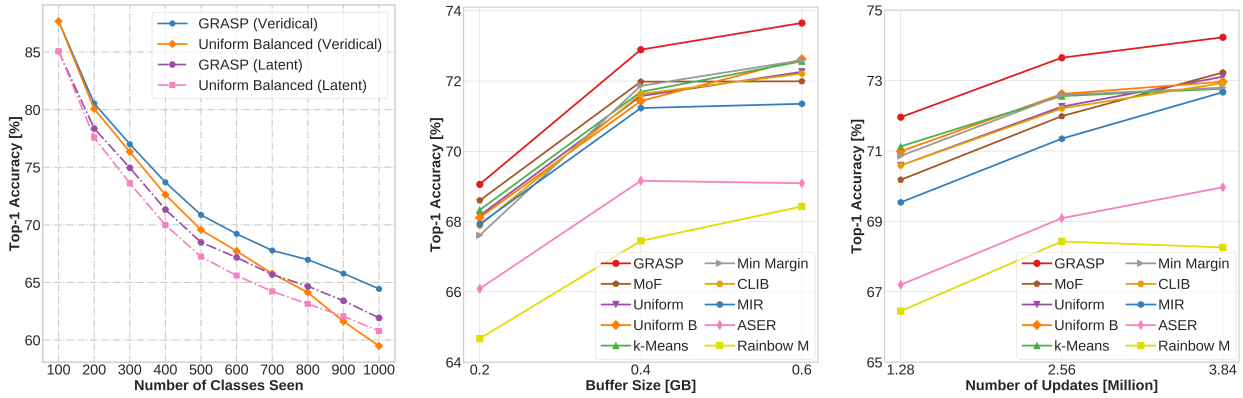
μ on new, old, and all classes respectively. Final accuracy on all classes is represented by α. All metrics use top-1 accuracy (%). For continual text classification, following IDBR (Huang et al., 2021) our evaluation takes place after training models on all tasks and the average across on all test sets is reported.

5 Results

We compare GRASP with a variety of 14 rehearsal methods on ImageNet-300 in Sec. 5.1. We also compare GRASP with 3 gradient-based methods on ImageNet-150 in Sec. 5.2. Next, we evaluate GRASP on full ImageNet-1K in various settings in Sec. 5.3. Finally, we summarize additional supporting results in Sec. 5.4.

5.1 GRASP vs. Various Rehearsal Methods

First, we analyze the performance of the 14 rehearsal methods including data-driven and model-based methods on ImageNet-300 using SIESTA with latent rehearsal during CIL. After pre-training MobileNet on the first 100 classes, the remaining 200 classes are learned in 4 rehearsal sessions (50 classes per rehearsal session). As shown in Table 2, in both unbounded and bounded memory settings, GRASP achieves the highest final and average accuracy on all classes. In all criteria, GRASP outperforms uniform balanced rehearsal, which earlier works found was the most effective policy for large-scale datasets (Prabhu et al., 2023a). Training time comparison is given in Fig. 2. To validate that GRASP performs effectively under varied memory and compute constraints, we compare GRASP with competitive rehearsal policies i.e., MoF, uniform, uniform balanced, min margin, k-Means, CLIB, MIR, Rainbow Memory, and ASER under varied memory and compute constraints. As shown in Fig. 3b and 3c, GRASP consistently surpasses compared methods in all cases. In Appendix H, we also qualitatively compare GRASP with these competitive methods.



(a) ImageNet-1K Learning Curves (b) Varied Memory Constraints (c) Varied Compute Constraints

Figure 3: (a) The ImageNet-1K learning curves of GRASP and uniform balanced policies in CIL using SIESTA. (b and c) The final accuracy of various rehearsal policies in CIL on ImageNet-300 using SIESTA and latent rehearsal.

Table 4: **Latent Rehearsal Results for CIL with SIESTA on ImageNet-1K (3 runs).** μ_N, μ_O, and μ_A denote accuracy on new, old, and all classes respectively averaged over rehearsals, and α is final ImageNet-1K accuracy.

Method	Unbounded Memory				Bounded Memory			
	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$
Uniform B	66.98 $\pm$0.14	69.35 $\pm$0.01	68.92 $\pm$0.03	60.76 $\pm$0.05	67.07 $\pm$0.09	69.27 $\pm$0.05	68.88 $\pm$0.06	60.35 $\pm$0.12
GRASP	**68.13** $\pm$0.13	**70.40** $\pm$0.05	**70.02** $\pm$0.07	**62.05** $\pm$0.09	**68.22** $\pm$0.05	**70.37** $\pm$0.05	**70.00** $\pm$0.02	**61.81** $\pm$0.14

Table 5: **Veridical Rehearsal Results for CIL with SIESTA on ImageNet-1K (3 runs).** μ_N, μ_O, and μ_A denote accuracy on new, old, and all classes respectively averaged over rehearsals, and α is final ImageNet-1K accuracy.

Method	Unbounded Memory				Bounded Memory			
	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$
Uniform B	68.60 $\pm$0.42	68.62 $\pm$1.55	68.48 $\pm$1.43	56.20 $\pm$2.37	63.58 $\pm$0.07	62.04 $\pm$0.58	62.17 $\pm$0.49	46.79 $\pm$0.85
GRASP	**69.85** $\pm$0.34	**72.69** $\pm$0.12	**72.24** $\pm$0.12	**63.87** $\pm$0.41	**64.07** $\pm$0.08	**63.36** $\pm$0.03	**63.38** $\pm$0.04	**49.27** $\pm$0.11

5.2 GRASP vs. Gradient-Based Methods

We also compare GRASP with SoTA gradient-based methods e.g., OCS, GSS, and Grad Matching. It is computationally prohibitive to scale these methods (see Fig. 2), for instance, Grad Matching requires 97$\times$ more training time than GRASP to learn 50 ImageNet classes. Therefore we had to keep this comparison small scale with ImageNet-150 subset. After pre-training 100 classes, the next 50 classes are learned in 5 rehearsal sessions or tasks (10 classes per rehearsal). As shown in Table 3, GRASP outperforms compared methods with significantly less training time.

5.3 ImageNet-1K Experiments

Having shown that under the same computational budget, GRASP achieves SoTA accuracy with little computational overhead compared to uniform when combined with SIESTA, we next turn to assessing GRASP's abilities on ImageNet-1K under a variety of scenarios: latent rehearsal, veridical rehearsal, IID CL, and generalization to other algorithms beyond SIESTA. As a baseline, we use balanced uniform in these experiments. In Appendix H, we also analyze the performance improvements of GRASP over uniform balanced in various ImageNet-1K experiments.

Latent Rehearsal. ImageNet-1K results for CIL with SIESTA using latent rehearsal are given in Table 4. GRASP consistently outperforms uniform balanced across criteria in both unbounded and bounded memory settings. Learning curves for GRASP and uniform balanced latent rehearsal are given in Fig. 3a. GRASP (latent) achieves higher accuracy than uniform balanced (latent) in all rehearsal sessions (100 ImageNet classes per rehearsal session). GRASP provides 40% and 36% speedups in terms of compute and training time respectively (see Fig. 1). Additionally, in Table 7 we compare GRASP with uniform balanced on ImageNet-1K under varied compute and memory constraints. Under all circumstances, GRASP consistently exceeds uniform balanced. Using McNemar's test, we compare the predictive accuracy of GRASP and uniform balanced and find that they are significantly different ($P < 0.001$) in all cases.

Table 6: **DERpp & GDumb (3 runs).** Comparison among rehearsal policies when combined with DERpp and GDumb in offline CIL on **ImageNet-1K**. † and ‡ denote variants that use uniform balanced and GRASP respectively.

Method	Latent Rehearsal				Veridical Rehearsal			
	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$
DERpp†	76.31 ±0.07	64.40 ±0.15	66.08 ±0.13	53.89 ±0.12	**57.14** ±4.99	62.12 ±0.71	61.27 ±1.11	45.08 ±1.93
DERpp‡	**77.54** ±0.08	**65.04** ±0.02	**66.82** ±0.02	**54.47** ±0.14	54.75 ±3.48	**63.42** ±0.35	**61.91** ±0.80	**48.57** ±0.86
GDumb†	67.72 ±0.02	69.80 ±0.09	69.41 ±0.07	61.05 ±0.07	62.40 ±0.50	62.84 ±0.39	62.69 ±0.40	47.45 ±0.94
GDumb‡	**69.04** ±0.04	**70.84** ±0.11	**70.51** ±0.09	**62.73** ±0.07	**63.68** ±0.10	**64.14** ±0.02	**64.00** ±0.03	**50.03** ±0.10

Table 7: **Compute and Memory Constraints Analysis for CIL with SIESTA on ImageNet-1K.** Buffer size and number of updates are reported in GB and million respectively. Reported is the final accuracy (%) on 1000 classes.

Method	Updates (M)			Buffer (GB)		
	0.76	1.02	1.53	0.75	1.51	2.01
Uniform Bal	59.10	60.28	61.10	57.13	60.27	60.79
GRASP	**60.00**	**61.36**	**62.66**	**58.44**	**62.00**	**61.93**

Veridical Rehearsal. To assess if GRASP is effective for veridical rehearsal instead, we compared GRASP to uniform balanced rehearsal with a variant of SIESTA that stores raw images. CIL results on ImageNet-1K are given in Table 5, GRASP persistently outperforms uniform balanced baseline in both unbounded and bounded memory settings. In terms of final ImageNet-1K accuracy, GRASP exceeds uniform balanced by absolute 7.67% (unbounded memory) and 2.48% (bounded memory). Fig. 3a shows learning curves. GRASP (veridical) obtains higher accuracy than uniform balanced (veridical) in all rehearsal sessions (100 ImageNet classes per rehearsal session). Additionally, we show ImageNet-1K curves for old and new tasks in Fig. 6. We compare the final predictions of GRASP and uniform balanced using McNemar's test and find that they are significantly different ($P < 0.001$) in all experiments.

Continual IID Learning. An ideal rehearsal policy should excel regardless of distribution. CIL is an extreme adversarial setting where catastrophic forgetting is severe. Here we consider the other extreme, IID CL, where catastrophic forgetting is minimal (Hayes et al., 2018). In IID CL settings, we conduct bounded-memory experiments (3 runs) on ImageNet-1K using SIESTA with latent rehearsal where each task contains 128K samples from randomly sampled classes. Other details adhere to the CIL's ImageNet-1K bounded memory setting. GRASP achieves higher accuracy ($\mu_A = 61.49 \pm 0.02$ and $\alpha = 63.22 \pm 0.09$) than uniform balanced ($\mu_A = 60.32 \pm 0.06$ and $\alpha = 61.52 \pm 0.03$). McNemar's test shows a significant difference ($P < 0.001$) between GRASP and uniform balanced policies.

Experiments with GDumb and DERpp. To validate that GRASP shows effectiveness for other rehearsal-based CL methods beyond SIESTA, we combined GRASP with two commonly used offline CL methods that use rehearsal: GDumb and DERpp for both the latent and veridical rehearsal settings. These experiments were done with the same MobileNetV3-L architecture used by SIESTA, which was pre-trained on the first 100 ImageNet classes. Compute and memory constraints are imposed. On CIL experiments with ImageNet-1K, GRASP outperformed the uniform balanced, as shown in Table 6. McNemar's test comparing final predictions of GRASP and uniform balanced reveals a significant difference ($P < 0.001$) between them in all conditions.

5.4 Additional Experiments

Why is GRASP More Effective? An efficient rehearsal policy should not perturb previously learned representations otherwise training overhead increases with an increase in representation drift. While learning new classes, the representations of old classes abruptly change and drift over time (Caccia et al., 2021). This abrupt change in old representations causes catastrophic forgetting of old knowledge and is difficult to correct without longer training. As shown in Fig. 4, existing methods exhibit higher representation drift. These methods mostly prioritize difficult samples, for instance, MIR and ASER select samples with maximum interference. Consequently, the old representations are excessively perturbed especially in the early stage of rehearsal in-

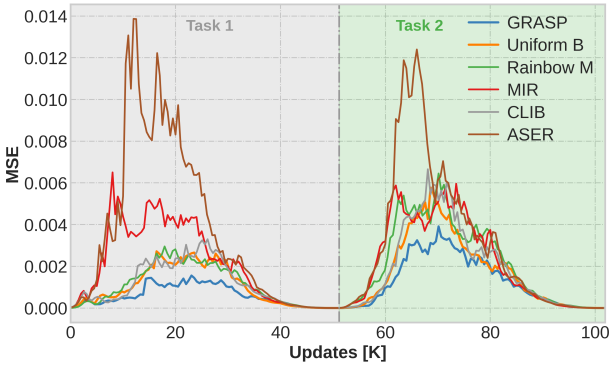


Figure 4: The representation drift of old classes while learning new classes in a stream of two tasks denoted by background colors. GRASP reduces representation drift.

Table 8: **Representation drift in two tasks setup for CIL with SIESTA.** We compare GRASP with other state-of-the-art policies in terms of representation drift on old tasks. Here, β and ϕ denote AUC drift and average drift over training iterations respectively.

Method	Task 1		Task 2	
	$\beta \downarrow$	$\phi \downarrow$	$\beta \downarrow$	$\phi \downarrow$
Rainbow Memory (Bang et al., 2021)	0.1100	0.0010	0.2148	0.0021
MIR (Aljundi et al., 2019a)	0.2192	0.0022	0.2273	0.0022
CLIB (Koh et al., 2021)	0.1136	0.0011	0.1920	0.0019
ASER (Shim et al., 2021)	0.3781	0.0038	0.3156	0.0030
Uniform Balanced	0.1039	0.0010	0.1706	0.0017
GRASP	**0.0600**	**0.0006**	**0.1275**	**0.0013**

dicated by the sharp rise in MSE. On the contrary, GRASP reduces representation drift by learning from subsets of increasing difficulty levels. A quantitative comparison is also given in Table 8. We see that compared to other competitive methods, GRASP achieves the lowest drift in all criteria. Following Caccia et al. (2021), we measure the representation drift of an old sample X at each training iteration t as $\left\| f_{\theta_t}(X) - f_{\theta_{t+1}}(X) \right\|$ where f_θ denotes model parameters. See Appendix B.4 for additional implementation details.

Continual Text Classification. Using IDBR, we evaluate GRASP and uniform rehearsal policies in continual text classification. They perform task incremental learning with various task sequences based on 5 datasets (AG News, Yelp, DBPedia, Amazon, and Yahoo! Answer). Compute and memory constraints are imposed. Performance is averaged over 3 runs. As shown in Table 11 in Appendix G, GRASP outperforms uniform in 5 out of 6 task sequences. Performance gains by GRASP align with the ones of IDBR (Huang et al., 2021) in the same benchmark datasets.

Long-Tailed Recognition. Besides balanced data streams, a rehearsal policy should also work for long-tailed data streams since real-world data distributions are often imbalanced and long-tailed. In both unbounded and bounded memory settings, GRASP exceeds uniform balanced in CIL on Places-LT-365 (see Appendix E).

Vision Transformer Results. To examine GRASP's efficacy in a ViT architecture, we conduct CL experiments using MobileViT-small with SIESTA. GRASP outperforms uniform balanced in CIL on ImageNet-300 (see Appendix F).

6 Conclusion

We showed that GRASP is a highly effective rehearsal policy compared to others on both large-scale image and NLP datasets. GRASP is effective for both latent and veridical rehearsal, and it works for multiple data distributions. GRASP is the first method to outperform balanced uniform for CIL on ImageNet-1K. We found that GRASP is more effective than other policies under a range of compute and memory constraints.

We focused on rehearsal policies, however, in future work, it would be interesting to examine the use of GRASP for buffer maintenance. We primarily focused on classification tasks to compare GRASP with the majority of existing sample selection policies that were originally designed for classification tasks. Besides classification tasks, GRASP can be explored in other computer vision and NLP tasks, including continual object detection (Acharya et al., 2020). However, a suitable hardness score would have to be designed for other tasks since the distance to the class prototype is only appropriate for classification. We studied GRASP with a fixed compute budget i.e., pre-defined fixed training steps. Future work could explore dynamically adapting the amount of training during rehearsal where the DNN stops early after achieving maximum performance.

While periodic retraining is currently the industry standard for updating DNNs, we believe GRASP is an important step toward supplanting this extremely computationally expensive process with much more efficient CL methods, and therefore reducing the carbon footprint from training models (Wu et al., 2022). Likewise, GRASP can be used to make on-device CL more efficient, where both compute and memory are heavily constrained (Hayes & Kanan, 2022).

Acknowledgments

We thank Tyler Hayes for comments on an early version of the manuscript. This work was supported in part by NSF awards #1909696, #2326491, and #2125362. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements of any sponsor.

REFERENCES

Wickliffe C Abraham and Anthony Robins. Memory retention–the synaptic stability versus plasticity dilemma. *Trends in neurosciences*, 28(2):73–78, 2005. 2

Manoj Acharya, Tyler L Hayes, and Christopher Kanan. Rodeo: Replay for online object detection. In *BMVC*, 2020. 10

Hasan Abed Al Kader Hammoud, Ameya Prabhu, Ser-Nam Lim, Philip HS Torr, Adel Bibi, and Bernard Ghanem. Rapid adaptation in online continual learning: Are we evaluating it right? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18852–18861, 2023. 2, 5

Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. In *NeurIPS*, 2019a. 1, 3, 6, 7, 10, 16

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019b. 1, 2, 3, 6, 7, 16

Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8218–8227, 2021. 1, 2, 3, 6, 7, 10, 16

Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 583–592, 2019. 2

Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-jussà. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6523–6541, 2020. 3

Jorg Bornschein, Alexandre Galashov, Ross Hemsley, Amal Rannen-Triki, Yutian Chen, Arslan Chaudhry, Xu Owen He, Arthur Douillard, Massimo Caccia, Qixuang Feng, et al. Nevis'22: A stream of 100 tasks sampled from 30 years of computer vision research. *arXiv preprint arXiv:2211.11747*, 2022. 5

Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems*, 33:14879–14890, 2020. 3, 16

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 2, 5

Lucas Caccia, Eugene Belilovsky, Massimo Caccia, and Joelle Pineau. Online learned continual compression with adaptive quantization modules. In *International conference on machine learning*, pp. 1240–1250. PMLR, 2020. 3

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2021. 9, 10, 16

Trevor Campbell and Tamara Broderick. Automated scalable bayesian inference via hilbert coresets. *The Journal of Machine Learning Research*, 20(1):551–588, 2019. 3, 6, 7, 16

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 5

Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 233–248, 2018. 2

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018a. 2, 3, 6, 7

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018b. 2

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019a. 3

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486, 2019*, 2019b. 2, 3, 4, 6, 7

Talfan Evans, Shreya Pathak, Hamza Merzic, Jonathan Schwarz, Ryutaro Tanno, and Olivier J Henaff. Bad students make great teachers: Active learning accelerates large-scale visual understanding. *arXiv preprint arXiv:2312.05328*, 2023. 1

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pp. 1183–1192. PMLR, 2017. 6, 7

Jhair Gallardo, Tyler L Hayes, and Christopher Kanan. Self-supervised training enhances online continual learning. *In British Machine Vision Conference (BMVC)*, 2021. 5

Alexander Gepperth and Cem Karaoguz. A bio-inspired incremental learning architecture for applied perceptual problems. *Cognitive Computation*, 8(5):924–934, 2016. 2

Md Yousuf Harun and Christopher Kanan. Overcoming the stability gap in continual learning. *arXiv preprint arXiv:2306.01904*, 2023. 2, 5

Md Yousuf Harun, Jhair Gallardo, Tyler L. Hayes, and Christopher Kanan. How efficient are today's continual learning algorithms? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2430–2435, June 2023a. 5

Md Yousuf Harun, Jhair Gallardo, Tyler L. Hayes, Ronald Kemker, and Christopher Kanan. SIESTA: Efficient online continual learning with sleep. *Transactions on Machine Learning Research*, 2023b. ISSN 2835-8856. URL https://openreview.net/forum?id=MqDVlBWRRV. 1, 2, 3, 5, 6, 15, 16

Tyler L Hayes and Christopher Kanan. Selective replay enhances learning in online continual analogical reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3502–3512, 2021. 3, 6, 7

Tyler L Hayes and Christopher Kanan. Online continual learning for embedded devices. In *CoLLAs*, 2022. 10

Tyler L Hayes, Ronald Kemker, Nathan D Cahill, and Christopher Kanan. New metrics and experimental paradigms for continual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 2031–2034, 2018. 9

Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9769–9776. IEEE, 2019. 2, 3

Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pp. 466–483. Springer, 2020. 2, 3, 6

Tyler L Hayes, Giri P Krishnan, Maxim Bazhenov, Hava T Siegelmann, Terrence J Sejnowski, and Christopher Kanan. Replay in deep learning: Current approaches and missing biological elements. *Neural Computation*, 33(11):2908–2950, 2021. 2

Chen He, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exemplar-supported generative reproduction for class incremental learning. In *BMVC*, pp. 98, 2018. 3

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019. 5

Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *International conference on learning representations*, 2018. 3

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. Continual learning for text classification with information disentanglement based regularization. *arXiv preprint arXiv:2104.05489*, 2021. 5, 7, 10, 16, 17

Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pp. 699–715. Springer, 2020. 3

Kenji Kawaguchi and Haihao Lu. Ordered sgd: A new stochastic optimization framework for empirical risk minimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 669–679. PMLR, 2020. 6, 7

Zixuan Ke and Bing Liu. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*, 2022. 3

Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018. 3

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 2

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019. 5

Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. In *International Conference on Learning Representations*, 2021. 1, 3, 6, 7, 10, 16

Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 226–227, 2020. 3

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2537–2546, 2019. 5, 17

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 2, 3

Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2021. 5

Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11321–11329, 2019. 3

Joseph O'Neill, Barty Pleydell-Bouverie, David Dupret, and Jozsef Csicsvari. Play it again: reactivation of waking experience and memory. *Trends in neurosciences*, 33(5):220–229, 2010. 2

Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent replay for real-time continual learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10203–10209. IEEE, 2020. 3

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 524–540. Springer, 2020. 2, 3, 5, 6

Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet K Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3698–3707, 2023a. 1, 2, 3, 5, 6, 7

Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, and Ozan Sener. Online continual learning without the storage constraint. *arXiv preprint arXiv:2305.09253*, 2023b. 2, 5

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017. 1, 2, 3, 4

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*. International Conference on Learning Representations, ICLR, 2019. 3

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5, 17

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International symposium on intelligent data analysis*, pp. 309–318. Springer, 2001. 6, 7

Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9630–9638, 2021. 1, 3, 6, 7, 10, 16

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 3

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. arxiv. *arXiv preprint arXiv:1708.07120*, 2017. 15

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022. 1, 2

Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *European Conference on Computer Vision*, pp. 254–270. Springer, 2020. 2

Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, pp. 1–13, 2022. 1

Eli Verwimp, Rahaf Aljundi, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L. Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H. Lampert, Martin Mundt, Razvan Pascanu, Adrian Popescu, Andreas S. Tolias, Joost van de Weijer, Bing Liu, Vincenzo Lomonaco, Tinne Tuytelaars, and Gido M van de Ven. Continual learning: Applications and the road forward. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=axBIMcGZn9. 2, 5

Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1): 37–57, 1985. 6

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023. 3, 4

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813, 2022. 10

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019. 2, 4

Ye Xiang, Ying Fu, Pan Ji, and Hua Huang. Incremental learning using conditional adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6619–6628, 2019. 3

Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. In *International Conference on Learning Representations*, 2022. 1, 3, 4, 6, 7, 16

Hanbin Zhao, Hui Wang, Yongjian Fu, Fei Wu, and Xi Li. Memory-efficient class-incremental learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5966–5977, 2021. 3

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 17

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey. *arXiv preprint arXiv:2302.03648*, 2023. 1

Appendix

A Overview of GRASP

We illustrate how GRASP works compared to uniform random policy in Fig. 5. We see that GRASP initially selects the most prototypical (representative) samples near the class mean and progressively selects less prototypical samples far from the class mean. Thus GRASP varies difficulty level to facilitate faster convergence compared to uniform random policy.

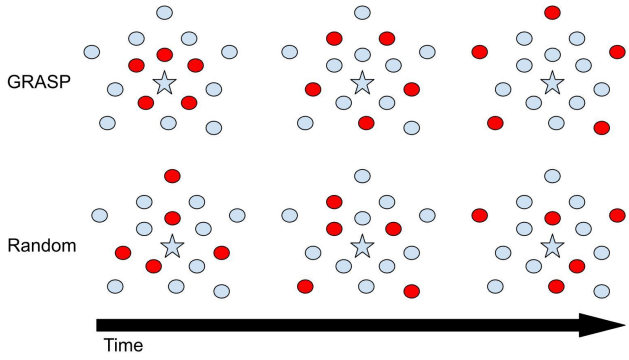


Figure 5: Overview of GRASP and Random Rehearsal Policies. Class mean is denoted by star. Selected samples are indicated by red circle.

B Implementation Details

B.1 MobileNet/ImageNet-1K Experiments

For all MobileNet/ImageNet-1K experiments in Sec. 5.3, we use SGD optimizer and OneCycle learning rate (LR) scheduler (Smith & Topin, 2017). We set weight decay and momentum to 10^{-5} and 0.9 respectively. For latent rehearsal experiments, we use an initial LR of 1.6 in the last layer with layer-wise LR reduction by a factor of 0.99 for earlier layers and mini-batch size 512. For veridical rehearsal experiments, we set initial LR to 0.4 for mini-batch size 256 and use similar layer-wise LR reduction as before.

For MobileNet's base-initialization (pre-training) on 100 ImageNet classes, we adopt pre-trained weights from SIESTA. For additional details about pre-training, we refer readers to SIESTA paper (Harun et al., 2023b). We also use the same ImageNet-1K data ordering as used in SIESTA. We configure OPQ to use 8 codebooks of size 256. OPQ is also trained on the same 100 ImageNet classes as used for pre-training MobileNet and kept fixed during the CL phase. OPQ is only used for latent rehearsal methods.

After base-initialization on 100 ImageNet classes, models continually learn the remaining 900 ImageNet classes in 9 rehearsal sessions (100 classes per rehearsal session). We set the number of iterations per rehearsal session to 2502 for mini-batch size 512 in latent rehearsal experiments. Whereas, in veridical rehearsal experiments, we set the number of iterations per rehearsal session to 5004 for a mini-batch size 256.

All algorithms e.g., SIESTA, GDumb, and DERpp use the same settings e.g., the same hyperparameters and the same pre-trained MobileNet architecture. For latent variants of GDumb and DERpp, we use the same configurations as SIESTA such as identical frozen earlier layers, identical plastic layers, the same pre-trained MobileNet architecture, and the same pre-trained OPQ model.

In all cases including both latent and veridical rehearsals, all methods use the same pre-trained MobileNet architecture and the same base initialization phase. We do not apply image augmentation in any experiment to solely focus on rehearsal policy without the influence of other variables.

B.2 MobileNet/ImageNet-300 Experiments

For MobileNet/ImageNet-300 experiments in Sec. 5.1, we use the same settings as aforementioned ImageNet-1K experiments e.g., hyperparameters, OPQ, optimizer, LR scheduler, and pre-trained MobileNet architecture. After base-

initialization on 100 ImageNet classes, models continually learn the remaining 200 ImageNet classes in 4 rehearsal sessions (50 classes per rehearsal session). We set the number of iterations per rehearsal session to 1251 for mini-batch size 512 in latent rehearsal experiments. Whereas in veridical rehearsal experiments, we set the number of iterations per rehearsal session to 2502 for mini-batch size 256. We implement Rainbow memory (Bang et al., 2021), MIR (Aljundi et al., 2019a), CLIB (Koh et al., 2021), and ASER (Shim et al., 2021) following the corresponding papers and codes.

B.3 MobileNet/ImageNet-150 Experiments

Here we specify settings used to compare gradient-based methods in Sec. 5.2. After base-initialization on 100 ImageNet classes, models continually learn the remaining 50 ImageNet classes in 5 rehearsal sessions (10 classes per rehearsal session). We set LR to 0.2 and the number of iterations per rehearsal session to 500 for mini-batch size 64. Other details adhere to the aforementioned ImageNet-1K experiments e.g., hyperparameters, OPQ, optimizer, LR scheduler, and pre-trained MobileNet architecture. We implement OCS (Yoon et al., 2022) and GSS (Aljundi et al., 2019b) following the corresponding papers and codes. We implement Grad matching (Campbell & Broderick, 2019) following the implementation from Borsos et al. (2020).

B.4 Representation Drift Experiments

We describe settings used in Sec. 5.4 for the representation drift experiments. After base-initialization on 100 ImageNet classes, models continually learn 2 tasks each of which consists of 10 ImageNet classes. The number of iterations per task is 100 for a mini-batch size of 512. Other details adhere to the aforementioned ImageNet-1K experiments e.g., hyperparameters, OPQ, optimizer, LR scheduler, and pre-trained MobileNet architecture.

Following Caccia et al. (2021), we measure the representation drift of an old sample X at each training iteration t as $\left\| f_{\theta_t}(X) - f_{\theta_{t+1}}(X) \right\|$ where f_θ denotes model parameters excluding the final layer. For this, we use a validation set of unseen old samples and their penultimate embedding vectors. To compute Area Under the Curve (AUC), we use Scikit-learn's *sklearn.metrics.auc* function.

B.5 Text Classification Experiments.

Here we describe settings used in continual text classification experiments in Sec. 5.4. We use AdamW optimizer with LR of 3×10^{-5} and weight decay of 0.01. We use batch size 8 and a maximum sequence length of 256. Other settings follow the replay baseline from IDBR paper (Huang et al., 2021). We study a total of 6 task sequences. They are: order 1 (ag $\rightarrow$ yelp $\rightarrow$ yahoo), order 2 (yelp $\rightarrow$ yahoo $\rightarrow$ ag), order 3 (yahoo $\rightarrow$ ag $\rightarrow$ yelp), order 4 (ag $\rightarrow$ yelp $\rightarrow$ amazon $\rightarrow$ yahoo $\rightarrow$ dbpedia), order 5 (yelp $\rightarrow$ yahoo $\rightarrow$ amazon $\rightarrow$ dbpedia $\rightarrow$ ag), and order 6 (dbpedia $\rightarrow$ yahoo $\rightarrow$ ag $\rightarrow$ amazon $\rightarrow$ yelp).

B.6 MobileNet/Places-LT-365 Experiments

These implementation details correspond to experiments in Sec. E. Since Places-LT is a small dataset, for base initialization, we adopt MobileNet backbone pre-trained on 100 ImageNet classes from MobileNet/ImageNet-1K experiment. We also adopt OPQ model pre-trained on same 100 ImageNet classes from MobileNet/ImageNet-1K experiment. After base initialization, CL phase begins where models learn 365 Places-LT classes in 5 rehearsal sessions (73 classes per rehearsal session) using SIESTA and latent rehearsal. We use SGD optimizer and OneCycle LR scheduler with initial LR of 0.1 and mini-batch size 32. We set the number of iterations per rehearsal session to 1200. Under memory constraints, memory is bounded by 20K samples. In an unconstrained memory setting, the entire dataset (62500 samples) is stored in a memory buffer. Other settings follow MobileNet/ImageNet-1K experiments. During the evaluation, we only use Places-LT-365 test set and do not use the 100 ImageNet classes subset used for base initialization.

B.7 MobileViT/ImageNet-300 Experiments.

Here we describe the settings used in Sec. F. Following SIESTA (Harun et al., 2023b), we use cosine cross entropy loss and replace batch norm with group norm and weight standardization in MobileViT-S architecture. For universal feature extraction, we freeze the first 8 blocks including stem, 6 MobielNetV2 blocks, and 1 MobileViT block. We keep the remaining blocks (1 MobileNetV2 block and 2 MobileViT blocks) and layers (1 CNN layer and 1 linear layer) plastic during the continual learning phase. Product quantization (OPQ) settings follow MobileNet/ ImageNet-1K experiments.

We use AdamW optimizer with initial LR of 4×10^{-4} and weight decay of 0.01. We use OneCycle LR scheduler. During base initialization, we train MobileViT on 100 ImageNet classes using supervised pre-training for 300 epochs. For this, we use the same settings described above. After base-initialization, models continually learn the remaining 200 ImageNet classes in 4 rehearsal sessions (50 classes per rehearsal session). We set the number of iterations per rehearsal session to 10K for mini-batch size 64. Under memory constraints, memory is bounded by 130K samples. In an unconstrained memory setting, all 383708 samples are stored in a memory buffer. MobileViT experiments are based on SIESTA and latent rehearsal.

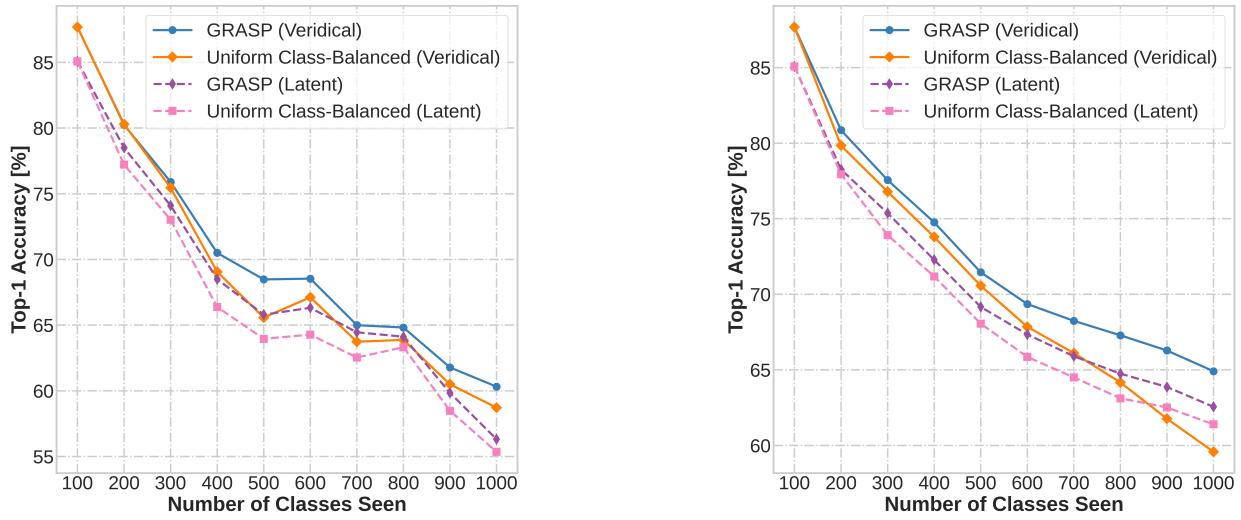
Compute. For compute (GPU) reasons, we vary the mini-batch size and number of iterations accordingly but compute constraints (iterations $\times$ mini-batch size) remain constant across experiments. We use a single GPU (NVIDIA RTX A5000) for all experiments.

C Dataset Details

We conduct vision experiments on ImageNet-1K and Places-LT. **ImageNet-1K** (Russakovsky et al., 2015) has 1000 categories, each with $732 - 1300$ training images and 50 for validation. In total, it contains 1.28 million training images and 50000 test images. **Places-LT** (Liu et al., 2019) is a long-tailed version of Places-2 (Zhou et al., 2017). Places-LT has 62500 training images spanning 365 classes with 5 to 4980 images each. For evaluation, we use the Places-LT validation set, which has 20 images per category (7300 total). For NLP dataset details, we refer readers to IDBR paper (Huang et al., 2021).

D ImageNet-1K Curves for Old and New Tasks

In the main text, we showed ImageNet-1K learning curves for all seen tasks (Fig. 3a). Here we show ImageNet-1K curves for old and new tasks in Fig. 6. These experiments use our default MobileNet/ImageNet-1K setup with SIESTA. We find that GRASP shows better performance than uniform balanced in all rehearsal sessions (100 ImageNet classes per rehearsal session) for both old and new tasks. This demonstrates that GRASP maintains a good balance between stability (old task) and plasticity (new task).



(a) Accuracy on new classes (plasticity)

(b) Accuracy on old classes (stability)

Figure 6: Performance of latent rehearsal policies on old and new classes in online CIL on ImageNet-1K. All methods use SIESTA and the same pre-trained MobileNet architecture.

E Places-LT-365 Results

Here, we use our default MobileNet/Places-LT setup with SIESTA and latent rehearsal. We run each compared method 6 times using 6 data orderings and report the average results in Table 9. In long-tailed recognition, GRASP consistently

Table 9: **Long-Tailed Recognition (Places-LT-365)**. Comparison between GRASP and uniform balanced in CIL on Places-LT with SIESTA and latent rehearsal. Here μ_N, μ_O, and μ_A denote accuracy (%) on new, old, and all classes respectively averaged over rehearsal sessions. And, α_H, α_T, and α stand for final accuracy (%) on head ($>$ 100 examples), tail ($\leq$ 100 examples), and all classes respectively. Reported results are averaged over 6 runs. Uniform balanced rehearsal is referred as Uniform†.

Method	Unbounded Memory						Bounded Memory					
	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha_H \uparrow$	$\alpha_T \uparrow$	$\alpha \uparrow$	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha_H \uparrow$	$\alpha_T \uparrow$	$\alpha \uparrow$
Uniform†	35.09	33.19	33.72	13.41	14.53	25.70	35.13	32.63	33.37	12.71	14.44	24.91
GRASP	**35.51**	**33.37**	**34.02**	**13.63**	**14.67**	**25.99**	**35.48**	**33.41**	**34.03**	**12.83**	**14.86**	**25.54**

Table 10: **MobileViT Results (ImageNet-300).** Comparison between GRASP and uniform balanced in CIL on ImageNet-300 with SIESTA and latent rehearsal. Here μ_N, μ_O, and μ_A denote accuracy (%) on new, old, and all classes respectively averaged over rehearsal sessions. And α is the final accuracy (%) on all classes.

Method	Unbounded Memory				Bounded Memory			
	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$	$\mu_N \uparrow$	$\mu_O \uparrow$	$\mu_A \uparrow$	$\alpha \uparrow$
Uniform Bal	67.60	71.43	70.53	64.19	65.82	68.47	67.75	59.01
GRASP	**68.18**	**72.05**	**71.15**	**65.37**	**66.54**	**68.86**	**68.20**	**59.26**

outperforms uniform balanced in all evaluation criteria for both unbounded and bounded memory settings. Therefore GRASP demonstrates robustness to long-tailed data distributions.

F MobileViT Results

For this analysis, we use our default MobileViT/ImageNet-300 setup with SIESTA and latent rehearsal. Previously, we evaluated GRASP using CNN, now we evaluate GRASP using ViT. Table 10 summarizes the results. We observe that GRASP achieves higher accuracy than uniform balanced in all metrics for both unbounded and bounded memory settings. This indicates that GRASP generalizes to ViT architecture besides CNN.

Table 11: **Continual Text Classification (3 runs).** GRASP versus the uniform balanced in continual text classification.

Method	Length-3 Task Sequences				Length-5 Task Sequences			
Order	**1**	**2**	**3**	**Avg.**	**4**	**5**	**6**	**Avg.**
Uniform Bal	73.06 $\pm$0.45	**73.33** $\pm$0.15	72.72 $\pm$0.18	73.04	74.33 $\pm$0.05	74.15 $\pm$0.41	74.15 $\pm$0.25	74.21
GRASP	**73.59** $\pm$0.12	72.96 $\pm$0.07	**73.26** $\pm$0.36	**73.27**	**74.67** $\pm$0.21	**74.22** $\pm$0.06	**74.36** $\pm$0.49	**74.42**

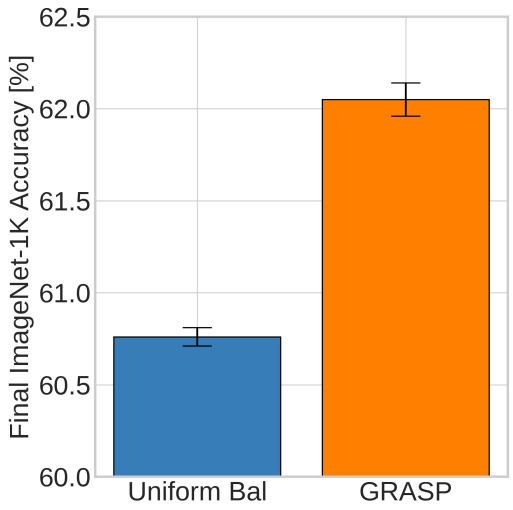
G Continual Text Classification Results

Due to space limitations in the main paper, we include the continual text classification results in this section. As shown in Table 11, GRASP surpasses uniform balanced rehearsal in 5 out of 6 task sequences.

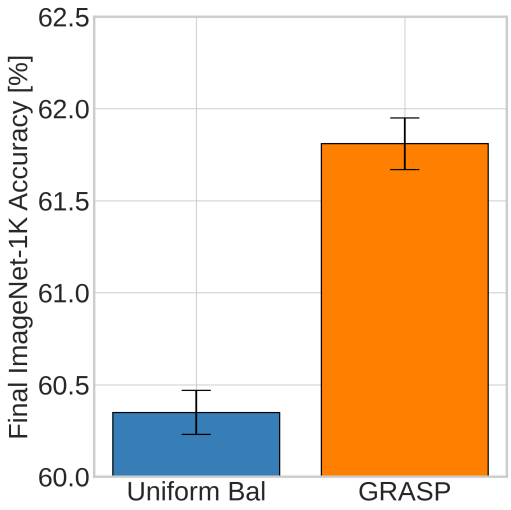
H Qualitative Analysis

In Sec. 5.3, we summarized all the ImageNet-1K results where GRASP outperformed uniform balanced rehearsal policy. In this section, we present bar plots to analyze the performance improvements of GRASP over uniform balanced rehearsal policy in various ImageNet-1K experiments. As shown in Fig. 7, Fig. 8, and Fig. 9, GRASP outperforms uniform balanced by nontrivial margins in all ImageNet-1K experiments.

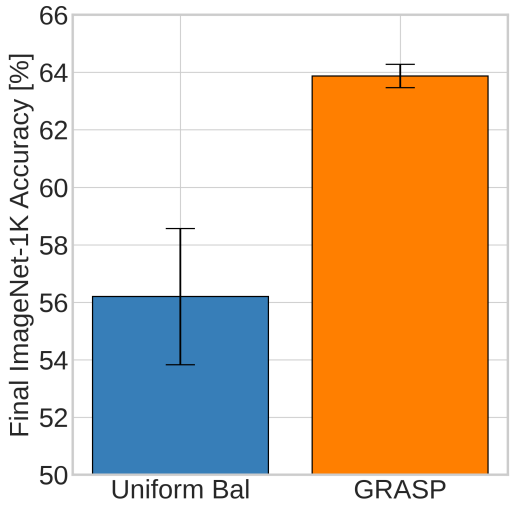
In Sec. 5.1, we summarized all the ImageNet-300 results where GRASP outperforms various rehearsal policies. Here, we qualitatively compare GRASP with the performant policies. As illustrated in Fig. 10, GRASP outperforms other competitive policies in CIL experiments on ImageNet-300.



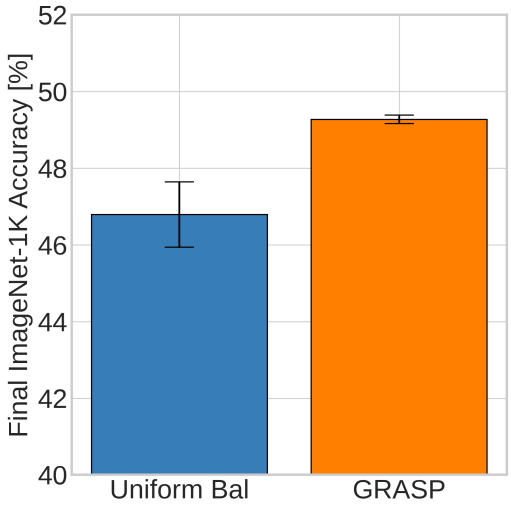
(a) **Latent Rehearsal (Unbounded Memory)**



(b) **Latent Rehearsal (Bounded Memory)**

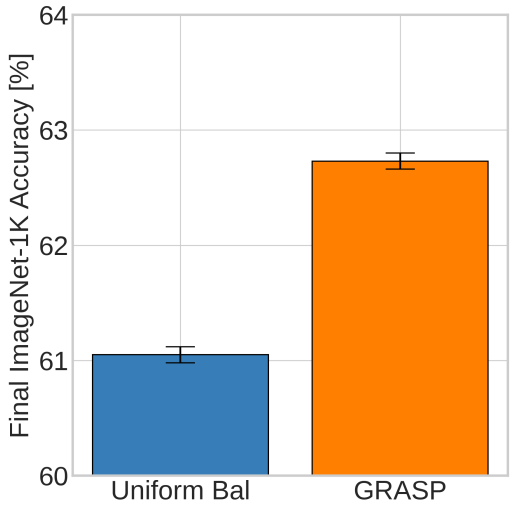


(c) **Veridical Rehearsal (Unbounded Memory)**

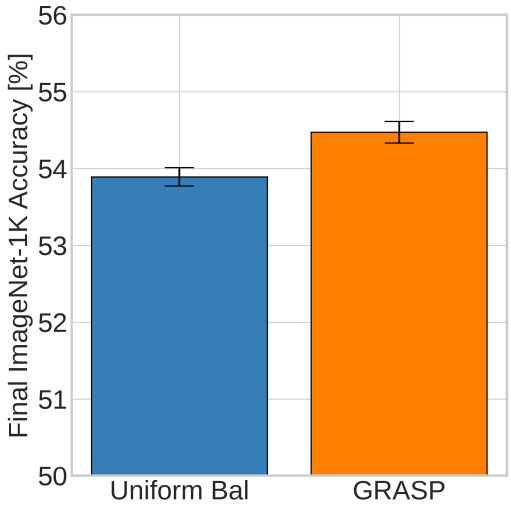


(d) **Veridical Rehearsal (Bounded Memory)**

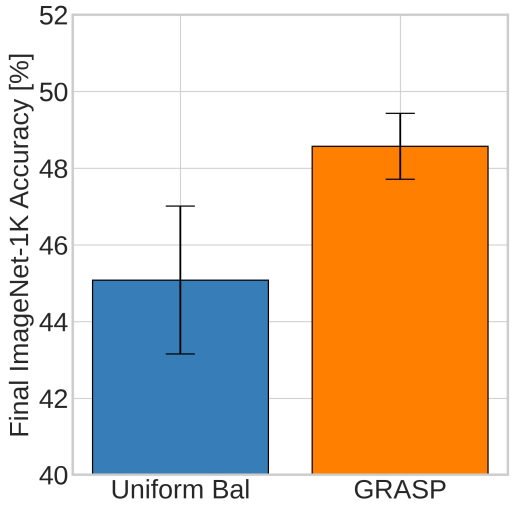
Figure 7: Qualitative comparison between GRASP and uniform balanced in CIL on ImageNet-1K with SIESTA. Each plot shows final accuracy (%) on ImageNet-1K averaged over 3 runs while indicating standard deviation ($\pm$).



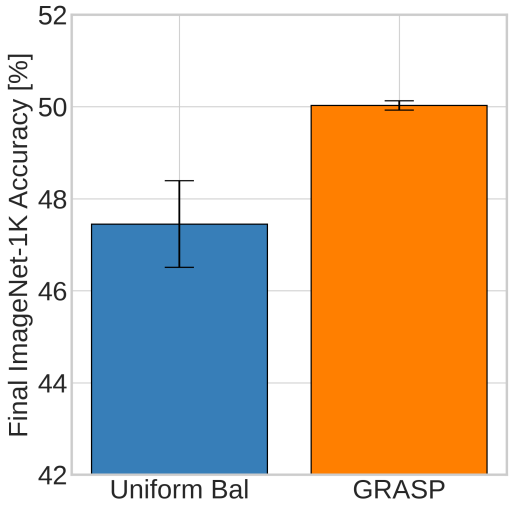
(a) **Latent Rehearsal (GDumb)**



(b) **Latent Rehearsal (DERpp)**



(c) **Veridical Rehearsal (GDumb)**



(d) **Veridical Rehearsal (DERpp)**

Figure 8: Qualitative comparison between GRASP and uniform balanced in memory-bounded CIL on ImageNet-1K with GDumb and DERpp. Each plot shows final accuracy (%) on ImageNet-1K averaged over 3 runs while indicating standard deviation ($\pm$).

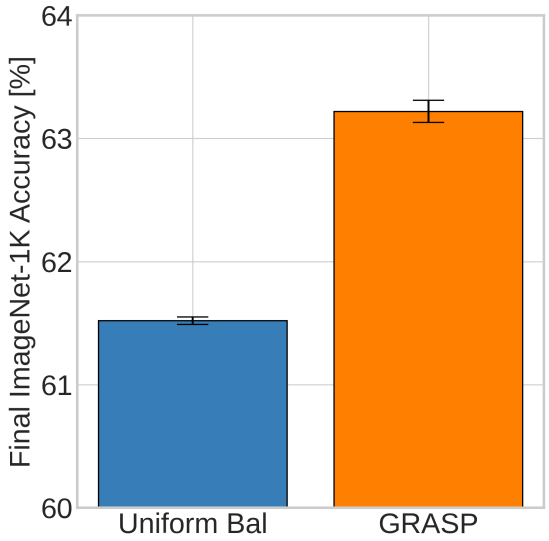


Figure 9: Qualitative comparison between GRASP and uniform balanced in memory-bounded IID CL experiments on ImageNet-1K using SIESTA with latent rehearsal. Each plot shows final accuracy (%) on ImageNet-1K averaged over 3 runs while indicating standard deviation (±).

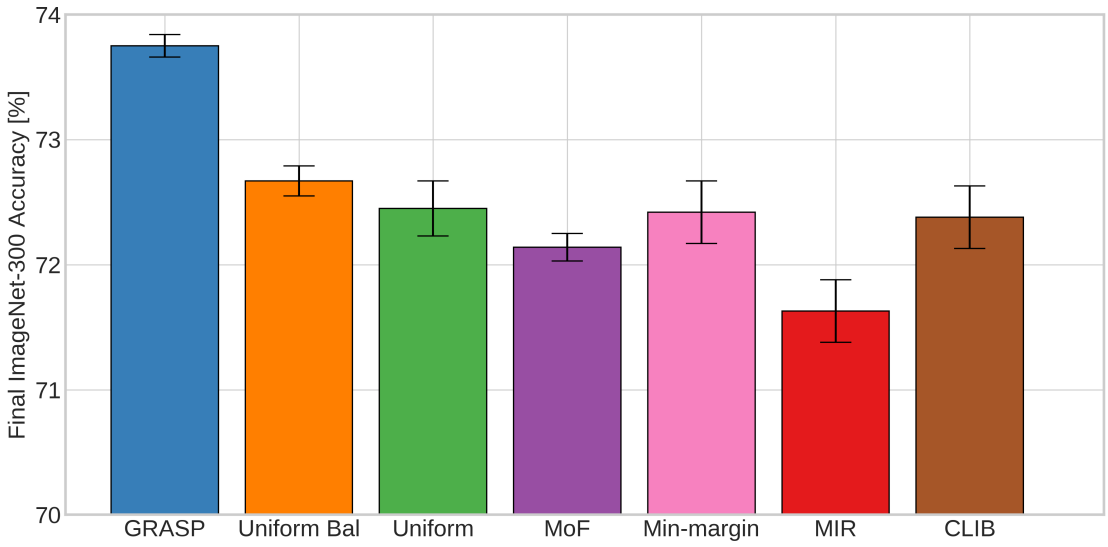


Figure 10: Qualitative comparison between GRASP and other competitive policies in memory-unbounded CIL experiments on ImageNet-300 using SIESTA with latent rehearsal. Each plot shows final accuracy (%) on ImageNet-300 averaged over 3 runs while indicating standard deviation (±).