Selective Replay Enhances Learning in Online Continual Analogical Reasoning

Tyler L. Hayes[1] and Christopher Kanan[1,2,3]
Rochester Institute of Technology[1], Paige[2], Cornell Tech[3]
{tlh6792, kanan}@rit.edu

Abstract

In continual learning, a system learns from non-stationary data streams or batches without catastrophic forgetting. While this problem has been heavily studied in supervised image classification and reinforcement learning, continual learning in neural networks designed for abstract reasoning has not yet been studied. Here, we study continual learning of analogical reasoning. Analogical reasoning tests such as Raven's Progressive Matrices (RPMs) are commonly used to measure non-verbal abstract reasoning in humans, and recently offline neural networks for the RPM problem have been proposed. In this paper, we establish experimental baselines, protocols, and forward and backward transfer metrics to evaluate continual learners on RPMs. We employ experience replay to mitigate catastrophic forgetting. Prior work using replay for image classification tasks has found that selectively choosing the samples to replay offers little, if any, benefit over random selection. In contrast, we find that selective replay can significantly outperform random selection for the RPM task[1].

1. Introduction

Deep neural networks excel at pattern recognition tasks, and they now rival or surpass humans at tasks such as image classification. However, the human capability for image classification is not unique in the animal kingdom. Multiple primate species are also capable of image classification at levels that rival humans [22, 63, 64]. One of the characteristics of human intelligence that distinguishes us over other animals is our ability to perform analogical reasoning [27, 30, 55]. Specifically, analogical encoding, i.e., the comparison of two situations which are partially understood, has been shown to facilitate forward knowledge transfer to new problems, as well as backwards transfer for memory retrieval [28]. This transfer is facilitated through the formation of higher level abstract schemas over time that are derived from comparisons between situations. More generally, abstraction capabilities enable forward knowledge transfer in humans, where previously learned knowledge is used to improve future learning [29, 40]. Recently, multiple deep learning models for analogical reasoning have been proposed [70, 77, 80, 88]; however, these systems assume that all training data is available at once and they will not be updated again, so backward and forward transfer in these systems cannot be studied. In this paper, we pioneer the development of the first systems for online continual learning of analogical reasoning tasks over time enabling forward and backward transfer in these tasks to be studied in models (see Fig. 1).

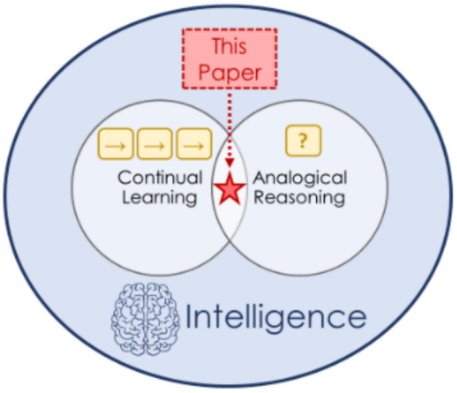

Figure 1. Humans are capable of continual learning, analogical reasoning, and performing both simultaneously. This facilitates much of what we consider intelligence. However, conventional neural networks suffer from catastrophic forgetting when updated on non-stationary data distributions and struggle to reason abstractly. Here, we study a network's ability to continually learn analogical reasoning tasks.

While humans have strong abstract reasoning abilities, neural networks struggle with these problems [37, 38, 70, 72, 86]. A popular cognitive test for analogical reasoning in humans is the Raven's Progressive Matrices (RPMs) problem [65], which provides a user with a 3×3 grid of images where the last image in the grid is missing (see Fig. 2). The rows and/or columns of images in the grid follow a specific rule and the user must compare a set of 8 choice images

[1] https://github.com/tyler-hayes/Continual-Analogical-Reasoning

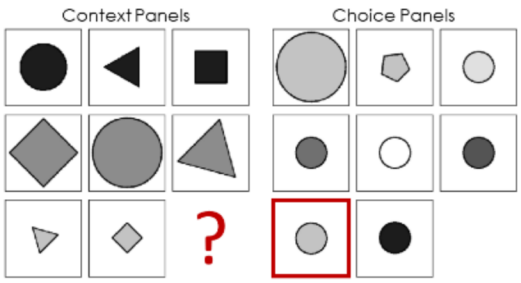

Figure 2. Example of an RPM problem from RAVEN [86]. Each RPM problem consists of 8 context panels (images) and 8 choice panels (images). Given these 16 panels as inputs, an agent must reason about the relationships among the 8 context panels and choose the best choice panel to complete the context matrix. For this example, we highlighted the answer panel with a red box.

to select the one that best fits in the final location. RPMs are ideal for measuring analogical reasoning in neural networks because they isolate the reasoning task directly, and several RPM datasets for neural networks have been created [70, 86].

Network architectures have been proposed for the RPM problem, with the most successful using a form of relation network [70, 77, 80, 88]. These models perform well in an offline setting, where they are trained on all available data and then evaluated. However, offline learning means that forward and backward transfer in these systems cannot be studied, and this is one of the critical capabilities that abstract reasoning is thought to facilitate in humans. When conventional neural networks are trained incrementally from non-stationary data streams, mirroring human learning, they suffer from catastrophic forgetting [58]. The field of continual learning endeavors to overcome this limitation [10, 17, 47, 61]. **This paper makes the following contributions:**

- We pioneer continual learning for analogical reasoning problems. We establish protocols and metrics for this problem using the Relational and Analogical Visual rEasoNing (RAVEN) dataset [86].
- We integrate both regularization and replay continual learning mechanisms into neural networks for analogical reasoning to establish baseline results.
- We study replay selection policies and find improved performance over uniform random selection for the RPM problem. This is interesting as selective replay has shown little benefit over uniform random selection in standard supervised classification settings [15, 34].

2. Problem Formulation

There are two paradigms for training a continual learner: the incremental batch paradigm and the online streaming paradigm. In incremental batch learning of RPMs, a dataset is divided into T batches, i.e., $\mathcal{D} = \bigcup_{t=1}^{T} B_t$, and at each time-step, t, the learner receives a new batch of data consisting of N_t samples, i.e., $B_t = \{(\mathbf{S}_i, y_i)\}_{i=1}^{N_t}$, where $\mathbf{S}_i$ is a single RPM sample consisting of 8 context panels (images), 8 choice panels (images), and a label y_i indicating which choice panel is the correct answer. Given this batch of samples, the agent is allowed to loop over the batch until it has been learned and is subsequently evaluated. The sequences of batches are ordered by task, which would induce catastrophic forgetting in a conventional network.

The incremental batch paradigm is unrealistic for real-time agents that must learn and evaluate on new data immediately and it does not mirror human learning. Online streaming learning addresses these drawbacks and requires an agent to learn new samples one at a time ($N_t = 1$) with only a single epoch through the entire dataset. Further, it requires models to operate under severe memory and time constraints, making them more ideal for deployment. Our streaming protocol is depicted in Fig. 3.

3. Related Work

3.1. Neural Networks for RPMs

One of the earliest works on neural networks for RPMs [12] identified a discrete set of rules required to solve them. This set of rules was used in [82] to develop a technique for automatically generating valid RPM problems, which was used to create the Procedurally Generated Matrices (PGM) dataset [70]. PGM was the first large-scale RPM-based dataset containing enough problems to successfully train deep neural networks. The Wild Relation Network (WReN) for PGM was proposed in [70], which uses a relation network [71] for reasoning. WReN outperformed other baselines on PGM by over 14%, demonstrating the strength of relation networks for analogical reasoning. Performance has been further improved by augmenting WReN with an unsupervised disentangled variational autoencoder [78] and Transformer attention mechanisms [32].

More recently, the synthetically generated RAVEN dataset was released [86]. Unlike PGM, each image in RAVEN contains objects in a structured pattern, requiring models to perform both structural and analogical reasoning. While WReN works well on PGM, it performs much worse on RAVEN, which is attributed to its lack of compositional reasoning abilities. To improve performance on RAVEN, the Contrastive Perceptual Inference Network uses a contrastive loss and contrast module to jointly learn rich features for perception [87]. In [88], a reinforcement learning policy was used to select informative samples for training a Logic Embedding Network to reason about panels. In [81], a graph neural network was used to extract object representations from images and reason about them.

Today, the Rel-Base and Rel-AIR architectures have the

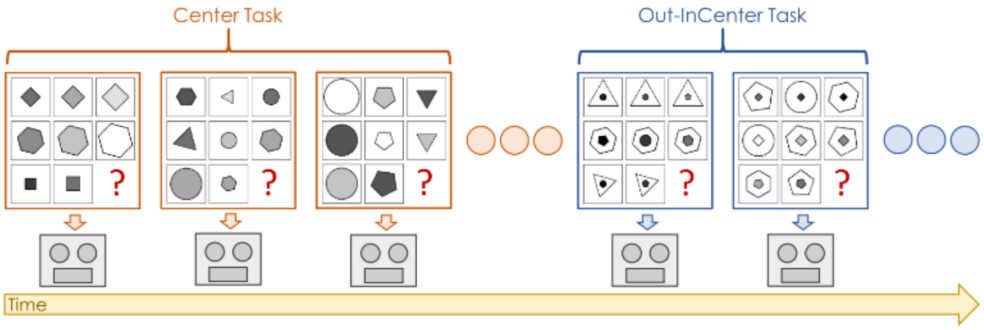

Figure 3. A demonstration of our training protocol on RAVEN [86]. At each time-step, a streaming learner receives a single new example from a particular task. It learns the new example and then can be subsequently evaluated.

best results on RAVEN and they perform competitively on PGM [77]. Rel-Base uses an object encoder network to process image panels individually. These encodings are passed to a sequence encoder to extract relationships before being scored. Rel-AIR first trains the Attend-Infer-Repeat (AIR) module [21] to extract objects from images. These objects are encoded and paired with additional position and scale information before being processed by a sequence encoder. Due to Rel-Base's simplicity and strong results, we extend it to the continual learning setting here.

3.2. Continual Learning in Neural Networks

One of the hallmarks of human intelligence is our ability to continually learn new information throughout our lifetimes without catastrophically forgetting previous knowledge. However, neural networks struggle to learn from non-stationary data distributions over time. When naively updated on new information, conventional neural networks catastrophically forget prior information [26, 58], which results from the stability-plasticity dilemma [1, 58]. The field of continual learning seeks to overcome these challenges.

There are three primary mechanisms used to mitigate forgetting in neural networks [10, 17, 47, 61]: 1) regularizing the plasticity of weights [4, 15, 16, 18, 48, 51, 53, 67, 74, 85], 2) growing the network architecture to accommodate new data [35, 41, 60, 69, 84], and 3) maintaining a subset of previous data in a memory buffer or generating previous data to replay when new data becomes available [9, 13, 16, 20, 25, 33, 34, 42, 46, 66, 79, 83]. Specifically, several regularization strategies seek to directly preserve important network parameters over time [4, 15, 48, 85], while others use variants of distillation to preserve outputs at various locations in the network [18, 20, 51]. However, for image classification, replay (or rehearsal) methods are currently the state-of-the-art approach, especially for large-scale datasets [13, 20, 34, 42, 66, 79, 83]. Standard replay approaches store explicit images or features in a memory buffer, while generative replay (or pseudo-rehearsal) methods train a generative model to generate previous samples [25, 36, 46, 60, 75]. When new data becomes available, either all or a subset of these previous samples are mixed with new samples to fine-tune the network.

Continual learning has also been studied for object detection [2, 76], semantic segmentation [14, 59], and robotics [24, 50]. Especially relevant to this paper are continual learners for visual question answering [31, 34], which requires agents to answer questions about images. However, visual question answering requires models to process natural language inputs, overcome severe dataset biases, and does not provide an isolated test of analogical reasoning. Using RPMs enables us to isolate continual analogical reasoning before moving on to tasks requiring more abilities.

4. Continual Learning Models & Baselines

We train continual models on RPM tasks in the following way. First, we perform a base initialization phase where Rel-Base is trained offline on the first task. After base initialization, each continual learner starts from these initialization weights and learns each of the remaining tasks one at a time, while being evaluated on all test data after every task. While batch models process new tasks in batches that they loop over several times, streaming models process samples one at a time, with only a single epoch through the dataset. For all models, we fix the task and sample order.

We study three methods to enable continual learning in Rel-Base: Distillation, EWC, and Partial Replay. We also study three baselines. These are described below:

- **Fine-Tune** – Rel-Base without any mechanisms to enable continual learning. It serves as a lower bound on performance. We run fine-tune in the streaming and batch paradigms.
- **Distillation** – Given a new batch of data, Distillation [39] optimizes both a classification and distillation loss, where the soft targets of the distillation loss are the scores of the model from the previous time-step. Distillation has been effective in mitigating forgetting on image classification [18, 20, 51, 66], object detection [76], and semantic segmentation [14] tasks.

- **EWC** – The Elastic Weight Consolidation (EWC) batch model uses a quadratic regularization term to encourage weights to remain close to their previous values [48]. Given a new batch of data, EWC optimizes both a classification loss and a quadratic penalty loss weighted by each parameter's importance, determined by the Fisher Information Matrix.
- **Partial Replay** – Partial Replay continually fine-tunes a model with all new data and a *subset* of previous data. It achieves strong results on classification tasks [13, 42, 66, 83]. We use it in the streaming setting and provide more details in Sec. 4.1.
- **Cumulative Replay** – Cumulative Replay continually fine-tunes a model with *all* new and previous data. It has been shown to mitigate forgetting [33], but is resource intensive. We train Cumulative Replay in the batch setting due to compute constraints.
- **Offline** – This is an offline model trained from scratch on all data until the current time-step. It serves as an upper bound on continual learning performance.

Distillation and EWC operate on batches since their loss constraints are computed from a model at the previous time-step, which would be less beneficial in the streaming setting. For Partial Replay, we study multiple policies for choosing which samples to replay, which we describe next.

4.1. Selective Replay Policies

Studying selective replay is important because it has the potential to enable better network generalization, allow networks to use fewer computational resources, and more closely aligns with biology [49, 54, 62]. Although replay selection policies have been explored for supervised classification [5, 6, 15, 57, 83], they have not yielded significant improvements, especially on large-scale datasets [34, 83]. For example, [83] found that selective replay performed 0.46% better on average over random sampling on CIFAR-100, but it is unclear if this improvement is statistically significant. In reinforcement learning, selective replay has provided more benefit [7, 73]. Further, selectively choosing training samples has yielded improved performance for offline RPM models [38, 88], but its effectiveness in the continual setting has not been explored, so we study it here.

Formally, we train a Partial Replay model in two stages: a base initialization phase and a streaming phase. During base initialization, we train the network offline using standard mini-batches and optimization updates. We then subsequently store all base initialization data in a replay buffer, $\mathcal{B}$. Then, labeled RPM sample pairs $(\mathbf{S}_i, y_i)$ are streamed into the model one at a time, where S_i is an RPM sample consisting of 8 context panels and 8 choice panels and y_i is the associated label for one of the $K = 8$ choice panels. The model mixes the current example with r labeled samples, which are selected from the replay buffer based on a selection probability p_i defined by:

$$p_i = \frac{v_i}{\sum_{\mathbf{v}_j \in \mathcal{B}} v_j} \quad , \tag{1}$$

where v_i is the value associated with choosing sample $\mathbf{S}_i$ from buffer $\mathcal{B}$ for replay. Using an unlimited buffer, we study the seven selective replay policies described below.

Uniform Random: Randomly select examples with uniform probability from the memory buffer. This is the simplest sampling approach and has demonstrated success for image classification [13, 15, 34, 83].

Minimum Logit Distance: Samples are scored according to their distance to a decision boundary [15]:

$$s_i = \sum_{j=1}^{K} |\phi(\mathbf{S}_i)_j \, \mathbf{y}_j| \quad , \tag{2}$$

where $\phi(\mathbf{S}_i) \in \mathbb{R}^K$ is a vector of network scores for the current example and $\mathbf{y} \in \mathbb{R}^K$ is a one-hot encoding of the label, y_i. Since neural networks are more uncertain about samples close to decision boundaries, this method prioritizes replaying these difficult samples.

Minimum Confidence: Samples are selected based on network confidence (i.e., softmax output):

$$s_i = \sum_{j=1}^{K} \text{softmax}(\phi(\mathbf{S}_i))_j \, \mathbf{y}_j = P(C = y_i | \mathbf{S}_i) \quad , \tag{3}$$

where softmax $(\cdot)$ returns the network's predicted probabilities and $\mathbf{y} \in C$ is a one-hot encoding of the label y_i. Intuitively, updating the model on samples that it is uncertain about could improve performance.

Minimum Margin: Samples are selected by:

$$s_i = P(C = y_i | \mathbf{S}_i) - \max_{y', y' \neq y_i} P(C = y' | \mathbf{S}_i) \quad , \tag{4}$$

where the first term represents the probability of the classifier choosing the correct label and the second term represents the probability of the network choosing the most probable label from the remaining classes. Smaller margin values indicate more uncertainty.

Maximum Loss: Scores are assigned based on cross-entropy loss:

$$s_i = -\sum_{j=1}^{K} \mathbf{y}_j \log P(j = y_i | \mathbf{S}_i) \quad . \tag{5}$$

Since networks seek to minimize classification loss, choosing to replay samples with the largest loss values should improve network performance.

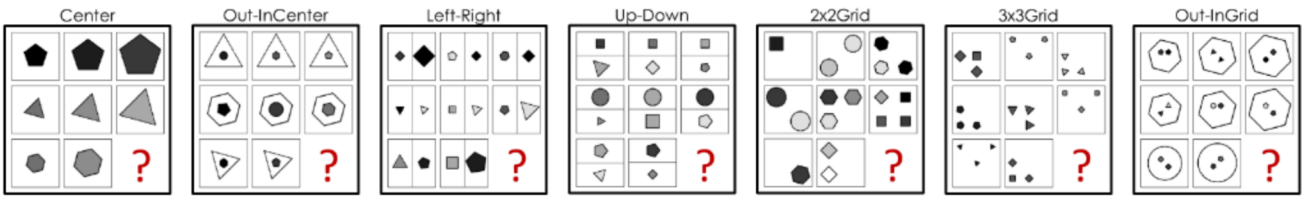

Figure 4. Example problems from each of the seven RAVEN tasks [86].

Maximum Time Since Last Replay: Samples are selected based on the last time they were seen by the network. Samples that have not been replayed in a long time could be forgotten and are prioritized for replay.

Minimum Replays: Samples are selected based on the number of times they have been replayed to the network. Intuitively, samples with the fewest number of replays might not have been well-learned by the network and should be replayed. We initialize all replay counts to the number of base initialization epochs.

To put all s_i values into a similar range, we shift all values by the minimum value in the buffer such that the smallest value is 1, i.e., $s_i \leftarrow s_i + (1 - \min_j s_j)$. For the Random and Max Loss techniques, v_i is equal to s_i. For all other techniques, we invert the s_i values such that the most probable samples have the largest v_i values, i.e., $v_i = (s_i + \epsilon)^{-1}$, where $\epsilon = 10^{-7}$ ensures the denominator is non-zero.

After the r samples are chosen, the network updates on this batch of $r + 1$ samples for a single iteration and the associated s_i values of the $r + 1$ samples are subsequently updated. All s_i values are appropriately initialized after the base initialization phase by pushing the base initialization data through the network. Further, during stream training, we only update the s_i values for samples that were replayed to save on compute time.

For each selection policy, we evaluate two ways of choosing samples: unbalanced and balanced such that we oversample a task if it is underrepresented. While the unbalanced strategy replays samples strictly based on their selection probabilities, the balanced strategy ensures that replay samples are not prioritized from only a few classes.

4.2. Implementation Details

We use the hyperparameters and pre-processing steps for Rel-Base from [77]. This includes resizing images to 80×80, normalizing pixel values in $[0, 1]$, and inverting images to increase signal. The hyperparameters are: Adam optimizer with learning rate=3e-4, β_1=0.9, β_2=0.999, ϵ=1e-8, batch size=32, and epochs=50 per task for batch and base initialization models. For offline models, we run at least 50 epochs, at most 250 epochs, early stop if validation loss does not improve for 10 epochs, and choose the checkpoint with the highest validation accuracy. We use replay mini-batches of size 32 for our main selective replay experiments and compare additional batch sizes in Sec. 6.2.1. For selective replay experiments, we allow models to use an unlimited buffer to focus on the selection methods directly. All models use a single output head, where task labels are unknown during test time. For regularization models, we grid searched for the regularization loss weight and found 1 and 10 to work best for Distillation and EWC, respectively. All timing experiments were run on the same machine with an NVIDIA TITAN RTX GPU, 48 GB of RAM, and an NVME SSD for consistency.

5. Experimental Setup

5.1. Dataset & Protocol

We conduct experiments on the RAVEN dataset [86], which has naturally defined tasks unlike the PGM dataset [70], making RAVEN more suitable for continual learning. RAVEN contains 1,120,000 images with 70,000 associated questions. These questions are distributed equally among seven unique figure configurations, depicted in Fig. 4, where each configuration requires different reasoning capabilities. We use these configurations to define our continual learning tasks, i.e., each task consists of one unique configuration. Since the order in which tasks appear can impact performance, we evaluate models under several fixed task permutations and fix the sample order across models for all experiments. We run each experiment with the following three permutations of the task ordering and report the mean performance: {Center, Out-InCenter, Left-Right, Up-Down, 2×2Grid, 3×3Grid, Out-InGrid}, {Up-Down, Center, Out-InCenter, Out-InGrid, 3×3Grid, 2×2Grid, Left-Right}, and {2×2Grid, Left-Right, Out-InGrid, Up-Down, 3×3Grid, Center, Out-InCenter}. An example of our streaming protocol is in Fig. 3.

Statistical biases have been identified in RAVEN's answer set [43, 77], and [77] suggested using models that process image frames independently to prevent bias exploitation. Since the Rel-Base model processes images independently, this bias is not a concern in our experimental results.

5.2. Metrics

To compute our metrics, we define a matrix $R \in \mathbb{R}^{T \times T}$, where each entry $R_{i,j}$ denotes the continual learner's test accuracy on task t_j after learning t_i and there are T total tasks. Following [33, 47], we measure a continual learner's performance with respect to an offline baseline:

$$\Omega = \frac{1}{T} \sum_{i=1}^{T} \frac{\gamma_i}{\gamma_{\text{offline},i}} \quad , \tag{6}$$

where $\gamma_i = \frac{1}{T} \sum_{j=1}^{T} R_{i,j}$ is the accuracy of the continual model at time i and $\gamma_{\text{offline},i}$ denotes the offline learner's accuracy at time i, computed using its associated R_{offline} matrix defined similarly to R. By normalizing the continual learner's performance to an offline learner, it is easier to compare results across task permutations. Higher values of Ω are better and an Ω of 1 indicates that the continual learner performed as well as the offline learner.

We also adopt three metrics from [19] to evaluate average accuracy and backward/forward transfer. Using R, these metrics are defined as:

$$A = \frac{2}{T\,(T+1)} \sum_{i \geq j}^{T} R_{i,j} \tag{7}$$

$$BWT = \frac{2}{T\,(T-1)} \sum_{i=2}^{T} \sum_{j=1}^{i-1} \left(R_{i,j} - R_{j,j} \right) \tag{8}$$

$$FWT = \frac{2}{T\,(T-1)} \sum_{i<j}^{T} R_{i,j} \quad , \tag{9}$$

where A is average model accuracy, BWT is backward transfer, and FWT is forward transfer. For BWT and FWT, a larger value indicates that learning new tasks improved performance on previously seen tasks and unseen tasks, respectively. Note that BWT can be negative, indicating that a model catastrophically forgot previous knowledge. We also report memory and compute requirements for each model since ideal learners require fewer resources.

6. Results

6.1. Main Results

Our main results are in Table 1 and the associated learning curves are in Fig. 5. Chance performance is ⅛ (12.5%), which is equal to an Ω of 0.237. The final accuracy that we use to normalize Ω is 91.7%, as reported in [77]. We include the top performing Partial Replay model (using the unbalanced Min Replays selection strategy) for comparison.

In terms of performance measures, Cumulative Replay consistently performed the best, with the Partial Replay model performing second best and requiring less time.

Table 1. Continual analogical reasoning performance on RAVEN. Each result is the average over three permutations. Additional memory requirements beyond the neural network (MB) and overall compute time (MIN) for each model are also reported. We report the best Partial Replay model (32 samples, Min Replays).

MODEL	Ω	A	BWT	FWT	MIN	MB
Stream Learners						
Fine-Tune	0.256	0.121	-0.238	0.091	8	0
Partial Replay	0.924	0.811	0.006	0.229	65	4301
Batch Learners						
Fine-Tune	0.581	0.417	-0.456	0.183	77	0
Distillation	0.513	0.357	-0.288	0.167	94	5
EWC	0.615	0.459	-0.347	0.178	85	5
Cumul. Replay	**0.990**	**0.893**	**0.028**	**0.232**	325	4301

Since the Partial Replay model uses an unlimited replay buffer, its memory usage is tied for worst with the Cumulative Replay learner. In the future, different buffer management strategies could be evaluated and paired with the best selective replay strategies to improve performance and reduce memory resources further. While the streaming Fine-Tune method required the fewest computational resources, it had the worst overall performance in terms of Ω, A, and FWT. This is unsurprising as Fine-Tune (Stream) does not have any mechanisms to mitigate catastrophic forgetting and sees each training example only once. The batch variant of Fine-Tune performed much better than the streaming variant, but had the worst overall BWT. We hypothesize this is because Fine-Tune (Batch) does not have any mechanisms to mitigate forgetting and batch training on new tasks causes overfitting, leading to worse BWT.

EWC outperformed both Fine-Tune variants in terms of Ω, A, and BWT, but had slightly worse FWT than Fine-Tune (Batch), likely due to its regularization loss. Distillation performed worse than Fine-Tune (Batch). While Distillation has demonstrated success in standard supervised classification scenarios, the distillation penalty does not prove useful for continual learning on RPMs. This could be because the output space for RPMs consists of a multiple choice problem of selecting one of eight images, which doesn't have as much semantic meaning as explicit object categories in standard classification settings. Note that both regularization models and Fine-Tune models suffered from forgetting, as indicated by their negative BWT scores.

6.2. Selective Replay Results

We endeavor to study the impact of selective replay policies on Partial Replay performance using the strategies outlined in Sec. 4.1. Performance for each of these methods in the unbalanced and balanced settings are in Table 2. Overall, the top performing method was Min Replays and the worst performing method was Random. We ran multiple comparisons tests against both of these selection methods

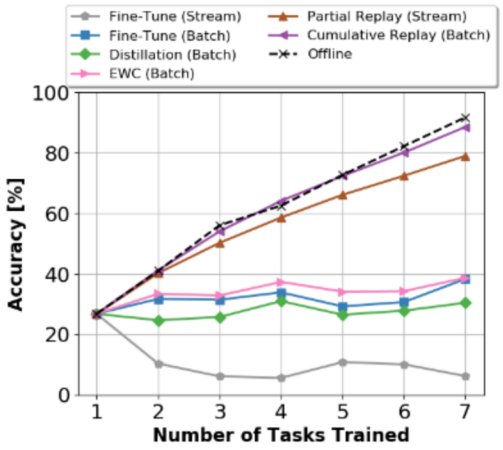

Figure 5. Learning curves for each baseline model.

(Random and Min Replays) in the unbalanced and balanced settings to determine if the other selection methods were statistically significantly different. To perform the tests, we first sampled 300 non-overlapping subsets of test instances randomly and computed the average final accuracy of the subsets across the three runs. We then ran a paired Welch's t-test (unequal variances) on the sets of subset accuracies. Unless otherwise noted, we corrected for multiple hypothesis testing using Holm-Bonferroni correction and used a significance level of 0.01.

In the unbalanced case, all selective replay methods had statistically significant performance differences compared to the Random selection policy, and we found that all selective replay strategies had statistically significant performance differences compared to the Min Replays policy (for all comparisons, $P < 0.001$). In the balanced case, only the Min Margin and Min Replays methods had statistically significant performance differences compared to the Random baseline (for both, $P < 0.001$), and only Random was statistically significant from Min Replays ($P < 0.001$).

We also performed significance tests of each unbalanced selection method with its associated balanced counterpart and found that only the unbalanced Max Loss and Min Replays strategies were statistically significant from their balanced variants (without Holm-Bonferroni correction). We also ran a paired t-test of an average of the final results from all selection methods in the unbalanced versus balanced settings and failed to reject the null hypothesis of equal means without Holm-Bonferroni correction ($P = 0.08$). Thus, on average there is no statistical significance between the unbalanced and balanced sampling strategies.

Fig. 6 shows histograms of the number of training samples with their associated number of replays after the completion of stream training for each unbalanced selection method. Qualitatively, the histograms for the Random and

Table 2. Comparison of selective replay strategies with unbalanced (unbal) and balanced (bal) sampling, averaged over three runs. Final accuracies (FA) used for significance testing are also reported.

	UNBAL			BAL		
METHOD	Ω	A	FA	Ω	A	FA
Random	0.882	0.769	0.752	0.897	0.785	0.758
Min Logit Dist	0.905	0.793	0.772	0.895	0.785	0.764
Min Confidence	0.895	0.783	0.764	0.902	0.790	0.767
Min Margin	0.906	0.795	0.773	0.900	0.790	**0.773**
Max Time	0.887	0.774	0.764	0.897	0.787	0.762
Max Loss	0.909	0.800	0.776	0.900	0.789	0.763
Min Replays	**0.924**	**0.811**	**0.790**	**0.907**	**0.795**	0.771

Max Time strategies, which performed the worst, look similar. Both histograms have the most samples with the fewest replays across all histograms and also have the most replays of the first task. These results suggest the poor performance of these methods was due to overplaying a small set of examples, while underplaying many other examples. Visually, the Min Confidence, Min Margin, Min Logit Dist, and Max Loss replay distributions look similar and performed similarly. The most unique distribution is from the top-performing Min Replays strategy, which has the fewest samples with the fewest replays and a more uniform replay count across all tasks compared to the other histograms.

6.2.1 Influence of Number of Replay Samples

Our main Partial Replay experiments used replay mini-batches of size 32. However, we were also interested in how each selection method's performance changed as a function of mini-batch size. Overall Ω results for each method using replay batches of size 8, 16, 32, and 64 are shown in Fig. 7. All curves are monotonically increasing with the Min Replays and Max Loss strategies yielding the top two performances across all batch sizes. Similarly, Random and Max Time produced the worst results across all batch sizes. Although there was a slight average performance increase (2.6% Ω) across all methods from a batch size of 32 to 64, running experiments with 64 samples required 1.9× more compute time, making it less ideal for streaming learning.

7. Discussion

Our main results indicate that replay-based learners perform the best for continually solving RPM puzzles. Although Cumulative Replay completely mitigates forgetting, it is computationally intensive and not ideal for streaming learning. To overcome this compute bottleneck, researchers often use Partial Replay to replay only a subset of the dataset at each time-step. Our experimental results indicate that Partial Replay performs well and its performance can be further improved by strategically selecting samples based on some criteria. While all sample selection methods

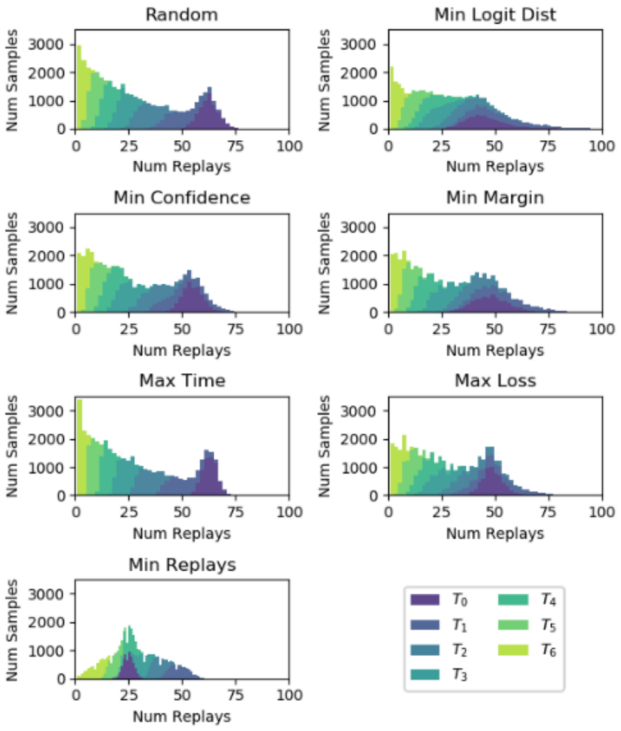

Figure 6. Histograms for each unbalanced selective replay method showing the number of samples with their associated replay counts after streaming learning. Each color denotes samples from the i-th task (T_i). Each plot is the average over three order permutations.

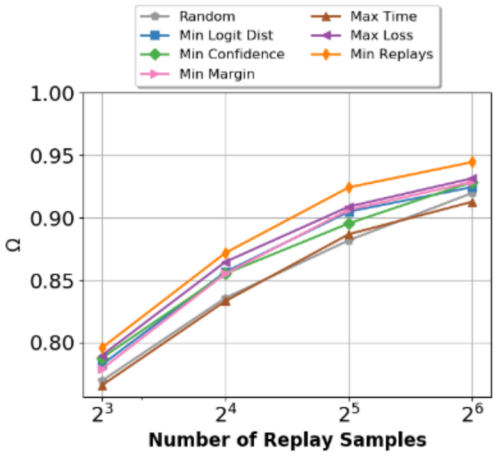

Figure 7. Ω performance as a function of replay samples for each unbalanced selective replay method averaged over three runs.

we tested performed better than uniform random selection, replaying samples based on the Min Replays and Max Loss strategies yielded the best overall results. In the future, it would be interesting to explore sample selection methods that optimize for samples directly [52], optimize for selection directly [5], or train a teacher network to choose samples for the learner [23]. In an offline setting, [88] paired a reinforcement learning policy with a teacher to intelligently compose a training curriculum for an RPM-based student, which improved performance over standard mini-batch training. This is promising evidence to explore additional selective replay strategies for continual RPM learning. Additionally, future work should include testing additional architectures designed for RPMs in the continual setting, which was beyond the scope of this study. This would inform future continual learning model designs.

Beyond RPMs, it would also be interesting to explore additional problems for continual analogical reasoning, including those introduced in [38]. Moreover, general abstract reasoning requires additional skills such as numerical reasoning, inductive reasoning, logical reasoning, etc. Future studies could explore continual learning in the context of these other reasoning skills using baselines introduced in [37, 72]. While most of these tasks require only reasoning, agents operating in the real-world should also be capable of processing additional inputs such as natural language questions, or identifying and mitigating biases to abstract general knowledge. Some problems requiring these additional components include visual question answering [8, 44, 56], referring expression recognition [45, 68], visual query detection [3], and image captioning [11]. While models have been developed for continual visual question answering [31, 34], the abstraction capabilities of these models have not been evaluated directly. More studies should be conducted to evaluate models on additional reasoning tasks.

8. Conclusion

While humans continually acquire new information and strengthen their reasoning capabilities over their lifetimes, deep neural networks struggle with these problems. In this paper, we introduced protocols, baseline methods, and metrics for evaluating networks on continual analogical reasoning tasks using the RPM-based RAVEN dataset. We found that replay methods had the best global performance and backward/forward knowledge transfer. We further studied several replay selection policies and found statistically significant performance improvements by all methods over a uniform random policy. Designing and testing more sophisticated architectures and continual learning strategies for RPMs remains an area of future work.

Acknowledgements. This work was supported in part by the DARPA/SRI Lifelong Learning Machines program [HR0011-18-C-0051], AFOSR grant [FA9550-18-1-0121], and NSF award #1909696. 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. We thank Robik Shrestha and Jhair Gallardo for their comments and useful discussions.

References

[1] Wickliffe C Abraham and Anthony Robins. Memory retention–the synaptic stability versus plasticity dilemma. *Trends in Neurosciences*, 2005. 3

[2] Manoj Acharya, Tyler L Hayes, and Christopher Kanan. Rodeo: Replay for online object detection. In *BMVC*, 2020. 3

[3] Manoj Acharya, Karan Jariwala, and Christopher Kanan. Vqd: Visual query detection in natural scenes. In *NAACL*, 2019. 8

[4] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018. 3

[5] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *NeurIPS*, pages 11849–11860, 2019. 4, 8

[6] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, pages 11816–11825, 2019. 4

[7] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NeurIPS*, pages 5048–5058, 2017. 4

[8] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *ICCV*, 2015. 8

[9] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *ICCV*, pages 583–592, 2019. 3

[10] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 2020. 2, 3

[11] Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 55:409–442, 2016. 8

[12] Patricia A Carpenter, Marcel A Just, and Peter Shell. What one intelligence test measures: a theoretical account of the processing in the raven progressive matrices test. *Psychological review*, 97(3):404, 1990. 2

[13] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018. 3, 4

[14] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *CVPR*, pages 9233–9242, 2020. 3

[15] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, pages 532–547, 2018. 2, 3, 4

[16] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *ICLR*, 2019. 3

[17] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2(6), 2019. 2, 3

[18] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, pages 5138–5146, 2019. 3

[19] Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: new metrics for continual learning. In *NeurIPSW*, 2018. 6

[20] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning: Supplementary material. In *ECCV*, 2020. 3

[21] SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *NeurIPS*, pages 3225–3233, 2016. 3

[22] Joël Fagot and Robert G Cook. Evidence for large long-term memory capacities in baboons and pigeons and its implications for learning and the evolution of cognition. *PNAS*, 103(46):17564–17567, 2006. 1

[23] Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Learning to teach. In *ICLR*, 2018. 8

[24] Fan Feng, Rosa HM Chan, Xuesong Shi, Yimin Zhang, and Qi She. Challenges in task incremental learning for assistive robotics. *IEEE Access*, 2019. 3

[25] Robert M French. Pseudo-recurrent connectionist networks: An approach to the 'sensitivity-stability' dilemma. *Connection Science*, 9(4):353–380, 1997. 3

[26] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. 3

[27] Dedre Gentner, Keith J Holyoak, and Boicho N Kokinov. *The analogical mind: Perspectives from cognitive science*. MIT press, 2001. 1

[28] Dedre Gentner, Jeffrey Loewenstein, and Leigh Thompson. Analogical encoding: Facilitating knowledge transfer and integration. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26, 2004. 1

[29] Dedre Gentner, Jeffrey Loewenstein, Leigh Thompson, and Kenneth D Forbus. Reviving inert knowledge: Analogical abstraction supports relational retrieval of past events. *Cognitive science*, 33(8):1343–1382, 2009. 1

[30] Dedre Gentner and Arthur B Markman. Structure mapping in analogy and similarity. *American psychologist*, 52(1):45, 1997. 1

[31] Claudio Greco, Barbara Plank, Raquel Fernández, and Raffaella Bernardi. Psycholinguistics meets continual learning: Measuring catastrophic forgetting in visual question answering. In *Annual Meeting of the Association for Computational Linguistics*, pages 3601–3605, 2019. 3, 8

[32] Lukas Hahne, Timo Lüddecke, Florentin Wörgötter, and David Kappel. Attention on abstract visual reasoning. *arXiv preprint arXiv:1911.05990*, 2019. 2

[33] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *ICRA*, pages 9769–9776, 2019. 3, 4, 6

[34] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *ECCV*, 2020. 2, 3, 4, 8

[35] Tyler L Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *CVPRW*, 2020. 3

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

[37] José Hernández-Orallo, Fernando Martínez-Plumed, Ute Schmid, Michael Siebers, and David L Dowe. Computer models solving intelligence test problems: Progress and implications. *Artificial Intelligence*, 230:74–107, 2016. 1, 8

[38] Felix Hill, Adam Santoro, David Barrett, Ari Morcos, and Timothy Lillicrap. Learning to make analogies by contrasting abstract relational structure. In *ICLR*, 2019. 1, 4, 8

[39] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[40] Mark K Ho. The value of abstraction. *Current opinion in behavioral sciences*, 29, 2019. 1

[41] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Lifelong learning via progressive distillation and retrospection. In *ECCV*, pages 437–452, 2018. 3

[42] Saihui Hou, Xinyu Pan, Zilei Wang, Chen Change Loy, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 3, 4

[43] Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Stratified rule-aware network for abstract visual reasoning. In *AAAI*, 2021. 5

[44] Kushal Kafle and Christopher Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 2017. 8

[45] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014. 8

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

[47] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *AAAI*, pages 3390–3398, 2018. 2, 3, 6

[48] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017. 3, 4

[49] Albert K Lee and Matthew A Wilson. Memory of Sequential Experience in the Hippocampus during Slow Wave Sleep. *Neuron*, 36(6):1183–1194, Dec. 2002. 4

[50] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, 2020. 3

[51] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *ECCV*, pages 614–629. Springer, 2016. 3

[52] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, pages 12245–12254, 2020. 8

[53] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, pages 6467–6476, 2017. 3

[54] K Louie and M A Wilson. Temporally Structured Replay of Awake Hippocampal Ensemble Activity during Rapid Eye Movement Sleep. *Neuron*, 29(1):145–156, Jan. 2001. 4

[55] Andrew Lovett and Kenneth Forbus. Modeling visual problem solving as analogical reasoning. *Psychological review*, 124(1):60, 2017. 1

[56] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NeurIPS*, 2014. 8

[57] James L McClelland, Bruce L McNaughton, and Andrew K Lampinen. Integration of new information in memory: new insights from a complementary learning systems perspective. *Philosophical Transactions of the Royal Society B*, 375(1799):20190637, 2020. 4

[58] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989. 2, 3

[59] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCVW*, pages 0–0, 2019. 3

[60] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jähnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *CVPR*, 2019. 3

[61] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019. 2, 3

[62] Adrien Peyrache, Mehdi Khamassi, Karim Benchenane, Sidney I Wiener, and Francesco P Battaglia. Replay of Rule-Learning Related Neural Patterns in the Prefrontal Cortex during Sleep. *Nature neuroscience*, 12(7):919–926, July 2009. 4

[63] Rishi Rajalingham, Elias B Issa, Pouya Bashivan, Kohitij Kar, Kailyn Schmidt, and James J DiCarlo. Large-scale, high-resolution comparison of the core visual object recognition behavior of humans, monkeys, and state-of-the-art deep artificial neural networks. *Journal of Neuroscience*, 38(33):7255–7269, 2018. 1

[64] Rishi Rajalingham, Kailyn Schmidt, and James J DiCarlo. Comparison of object recognition behavior in human and monkey. *Journal of Neuroscience*, 35(35):12127–12136, 2015. 1

[65] John C Raven. *Raven's Progressive Matrices (1938): Sets A, B, C, D, E.* Australian Council for Educational Research, 1938. 1

[66] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 3, 4

[67] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. In *NeurIPS*, pages 3738–3748, 2018. 3

[68] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, pages 817–834. Springer, 2016. 8

[69] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv:1606.04671*, 2016. 3

[70] Adam Santoro, Felix Hill, David Barrett, Ari Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In *ICML*, pages 4477–4486, 2018. 1, 2, 5

[71] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *NeurIPS*, pages 4967–4976, 2017. 2

[72] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *ICLR*, 2019. 1, 8

[73] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *ICLR*, 2016. 4

[74] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, pages 4555–4564, 2018. 3

[75] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, pages 2990–2999, 2017. 3

[76] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, pages 3400–3409, 2017. 3

[77] Steven Spratley, Krista Ehinger, and Tim Miller. A closer look at generalisation in raven. In *ECCV*, 2020. 1, 2, 3, 5, 6

[78] Xander Steenbrugge, Sam Leroux, Tim Verbelen, and Bart Dhoedt. Improving generalization for abstract reasoning tasks using disentangled feature representations. *arXiv preprint arXiv:1811.04784*, 2018. 2

[79] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *ECCV*, 2020. 3

[80] Damien Teney, Peng Wang, Jiewei Cao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. V-prom: A benchmark for visual reasoning using visual progressive matrices. In *AAAI*, pages 12071–12078, 2020. 1, 2

[81] Duo Wang, Mateja Jamnik, and Pietro Lio. Abstract diagrammatic reasoning with multiplex graph networks. In *ICLR*, 2020. 2

[82] Ke Wang and Zhendong Su. Automatic generation of raven's progressive matrices. In *IJCAI*, 2015. 2

[83] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 3, 4

[84] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *ICLR*, 2018. 3

[85] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995, 2017. 3

[86] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *CVPR*, pages 5317–5327, 2019. 1, 2, 3, 5

[87] Chi Zhang, Baoxiong Jia, Feng Gao, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. Learning perceptual inference by contrasting. In *NeurIPS*, pages 1075–1087, 2019. 2

[88] Kecheng Zheng, Zheng-Jun Zha, and Wei Wei. Abstract reasoning with distracting features. In *NeurIPS*, pages 5842–5853, 2019. 1, 2, 4, 8