Learning to Learn: How to Continuously Teach Humans and Machines

Parantak Singh[1, 2], You Li[2,3], Ankur Sikarwar[1, 2], Weixian Lei[4], Difei Gao[4],
Morgan B. Talbot[5, 6], Ying Sun[2], Mike Zheng Shou[4], Gabriel Kreiman[5], Mengmi Zhang[1, 2]
[1] Nanyang Technological University (NTU), Singapore [2] CFAR and I2R, Agency for Science, Technology and Research, Singapore,
[3] University of Wisconsin-Madison, USA, [4] Show Lab, National University of Singapore, Singapore,
[5] Boston Children's Hospital, Harvard Medical School, USA, [6] Harvard-MIT Health Sciences and Technology, MIT,
Address correspondence to mengmi@i2r.a-star.edu.sg

Abstract

Curriculum design is a fundamental component of education. For example, when we learn mathematics at school, we build upon our knowledge of addition to learn multiplication. These and other concepts must be mastered before our first algebra lesson, which also reinforces our addition and multiplication skills. Designing a curriculum for teaching either a human or a machine shares the underlying goal of maximizing knowledge transfer from earlier to later tasks, while also minimizing forgetting of learned tasks. Prior research on curriculum design for image classification focuses on the ordering of training examples during a single offline task. Here, we investigate the effect of the order in which multiple distinct tasks are learned in a sequence. We focus on the online class-incremental continual learning setting, where algorithms or humans must learn image classes one at a time during a single pass through a dataset. We find that curriculum consistently influences learning outcomes for humans and for multiple continual machine learning algorithms across several benchmark datasets. We introduce a novel-object recognition dataset for human curriculum learning experiments and observe that curricula that are effective for humans are highly correlated with those that are effective for machines. As an initial step towards automated curriculum design for online class-incremental learning, we propose a novel algorithm, dubbed Curriculum Designer (CD), that designs and ranks curricula based on inter-class feature similarities. We find significant overlap between curricula that are empirically highly effective and those that are highly ranked by our CD. Our study establishes a framework for further research on teaching humans and machines to learn continuously using optimized curricula. Our code and data are available through this link.

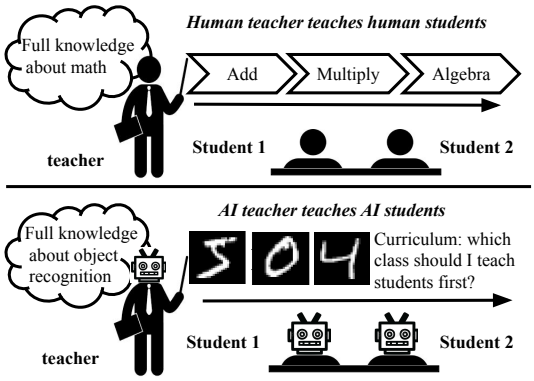


Figure 1: **Curricula in classroom and machine learning settings**. In human education, a natural curriculum designed by a knowledgeable math teacher prescribes teaching, in order, addition, multiplication, and algebra. Student 1 and Student 2 learn these concepts in a continuous fashion. Similarly, in an image classification task, what is the optimal curriculum for an AI teacher to continuously teach AI students to recognize images?

1. Introduction

When learning mathematics, students continuously advance through a curriculum that guides them to first learn addition, then multiplication, and later algebra such that each new concept both builds upon and reinforces existing knowledge (**Fig 1**). Studies on curriculum development in education show that careful design of curricula for human students can enable an incremental learning process, facilitating positive knowledge transfer to new tasks and minimizing forgetting of learned tasks [55]. Drawing on this inspiration, our goal is to develop a knowledgeable artificially intelligent (AI) teacher (a "curriculum designer") that produces optimized curricula that enhance learning outcomes of both human students and machine learning algorithms ("AI students").

A growing body of literature in the field of "curriculum

learning" investigates the order in which training examples are presented to machine learning (ML) algorithms. The effects of curriculum on ML outcomes have been explored in supervised [57, 71, 60, 66, 7], weakly-supervised [59, 53, 22], unsupervised [68, 57, 48], and reinforcement learning (RL) [31, 19, 45] settings. Existing work in supervised learning [57, 71, 60, 66, 7] has demonstrated improved generalization ability and convergence speed through the design of more effective curricula, but only by estimating intra-class example difficulty and scheduling examples within a *single* task. Unlike supervised classification algorithms that require multiple passes over large, shuffled training datasets to learn many classes in parallel, humans learn a variety of tasks incrementally through a continuous stream of non-repeating experience. This process is more closely emulated in continual learning (CL) settings, where ML algorithms learn a series of tasks one at a time, and particularly in online CL settings where each training example is shown only once [42]. Although the presentation order of separate tasks is a central focus in designing curricula for humans, the influence of task order on offline and online CL outcomes remains largely unexplored.

To address this question, we investigated the effects of class presentation order ("curriculum") during online class-incremental CL by machines and humans. An ideal learning algorithm in this setting would leverage its knowledge of early tasks to more effectively learn later tasks (forward transfer) while also avoiding forgetting early tasks. The challenging problem of "catastrophic forgetting" in artificial neural networks has been addressed with a variety of CL-specific algorithms [61]. Since each CL algorithm modulates the learning process using a different strategy, we conceptualize different CL algorithms as distinct AI students that may or may not maximally benefit from the same curricula. Our empirical ML results suggest that curriculum design choices greatly influence knowledge transfer and forgetting across CL algorithms and hyperparameter settings of each. We demonstrate a strong correlation among different CL algorithms in the relative effectiveness of different curricula. We also found curriculum effects that are correlated among CL algorithms in a continual visual question answering setting [36].

Building upon these findings, we propose an automatic curriculum designer (CD), an algorithm that efficiently designs and ranks curricula. In a nutshell, our CD enables pairs of object classes that are nearer to each other in feature space to be separated farther from each other in time during the training processes of neural networks and humans. Unlike pre-defined curriculum learning algorithms [59, 41, 63, 56], our CD does not require prior knowledge from domain experts, nor any human intervention. Our results demonstrate that curricula ranked highly by our CD improve learning performance across multiple CL algorithms.

To probe further whether the optimal curricula for continual machine learning are also beneficial for human learning, we conducted a series of human psychophysics experiments and contributed a new novel-object recognition CL benchmark. From the experiments, we observed a high degree of agreement between the most effective curricula for CL algorithms and humans.

Our main contributions to this work are as follows:

- We establish a methodology to study curriculum effects in online class-incremental learning.
- We introduce a new novel-object recognition dataset to benchmark the effectiveness of class-incremental curricula for humans and CL algorithms.
- We quantify commonalities among empirically optimal curricula for CL algorithms and humans.
- We propose an automated curriculum designer that can design the optimal curricula and rank (score) the existing curricula by their effectiveness.

2. Related Works

2.1. Continual Learning (CL)

CL strategies can be grouped into three categories: weight regularization, replay, and architecture expansion. Regularization methods constrain or regularize weight updates during training on new tasks using information from previous tasks [37, 10, 24, 30, 70, 35]. Replay-based strategies involve storing a subset of examples from previous tasks and interspersing them with training data from newly encountered tasks to mitigate forgetting [65, 47, 2, 10, 44, 40, 5]. Architecture adaptation methods involve expanding or restructuring neural networks to assimilate new tasks [37, 24, 30, 70, 35, 20, 50, 17, 46, 52, 1]. CL methods are predominantly evaluated in *offline* class-incremental settings where many passes over data within each task are permitted. Researchers report average performance over multiple runs with *random* class orders. Here, we exhaustively study the effect of class presentation order during *online* class-incremental learning, where only one pass over the data within each task is allowed.

2.2. Curriculum Learning

Curriculum learning refers to learning with a meaningful ordering of training examples, commonly from "easier" to "harder" data [8, 3]. The efficacy of proposed curricula is evaluated in terms of generalization to test data and convergence speed during training. Previous works in curriculum learning can be categorized into predefined curriculum learning [8, 56, 12, 13] and automatic curriculum learning [62, 29, 16, 21]. Predefined curriculum learning entails designing a data scheduler or a difficulty

measure with human priors. These algorithms work well when designed for specific tasks, but generalize poorly to out-of-domain tasks. In contrast, we propose an automatic curriculum designer that can design and rank curricula based on inter-class feature differences.

In automatic curriculum learning, most works adopt data-driven approaches [29, 16, 21] and RL-based approaches incorporating student feedback [54, 25, 15, 43, 51]. These methods are often deployed in teaching both machines [59, 53, 22, 68, 57, 48, 31, 19, 45] and humans [54, 25, 15, 43, 51]. In image classification settings, curriculum learning approaches are almost exclusively oriented toward measuring intra-class example difficulty. Existing methods specifically focus on a single multi-class object recognition task [64, 58, 49, 22] in which all examples from each class can be trained on multiple times. We deviate from previous studies in examining the order in which classes or tasks are presented to the network, rather than the ordering of training examples within one task.

One recent study highlighted how the most widely-used curriculum design strategy (increasing difficulty) may not always be optimal, and how anti-curricula ("harder" to "easier") or random orderings yield comparable results in multi-class image classification settings [64]. The study reported that curriculum effects become stronger when the number of training iterations is limited. Aligned with this constraint, we investigated the effect of curriculum on CL algorithms under stringent online conditions where training is limited to a single pass through the data.

3. Experiments

We conducted our experiments in the online class-incremental learning setting. An image dataset D comprises N object classes $\{c_1, c_2 \cdots c_N\}$ with K training images each. The objective is to propose a temporal order of class presentation T from $t_1, t_2 \cdots t_N$ (a "curriculum") such that a given CL algorithm $\mathcal{A}$ (a "student") yields the optimal learning outcome. That is, $\mathcal{A}$ learns to adapt to new classes with minimal forgetting of previously learned classes while progressing through T.

3.1. Datasets and Baselines

We used three datasets for our experiments: MNIST ($60,000$ training images, $10,000$ test images) [33], FashionMNIST ($60,000$ training and $10,000$ test images) [67], and CIFAR10 ($50,000$ training and $10,000$ test images) [32]. Each dataset consists of 10 object classes. Ideally, each curriculum is a permutation of 10 object classes, resulting in a total of 10! (more than $3e^6$) possible curricula per dataset. Thus, running all permutations is infeasible due to limited computational resources. To mitigate this issue, we introduced two paradigms: in "paradigm-I", we chose a subset of the dataset comprising 5 classes with 1 class per task, and in "paradigm-2", we made 5 tasks with 2 classes each. In both paradigms, the order of the exemplars from the classes within a task is fixed and only the task sequence is permuted, resulting in a total of $5! = 120$ curricula. Without loss of generality, we only present and discuss results for paradigm-I. See **Sec S2** for details of class grouping, and see **Sec S7-S9**, and **Fig S11-S13**, **S18-S22**, **S24**, **S27**, **S28** for results in paradigm-I. In general, the conclusions drawn in the first paradigm also hold true in the second. In paradigm-I, we used classes '0,' '1,' '2,' '3,' and '4' from MNIST, classes 'coat,' 'dress,' 'pullover,' 'top,' and 'trouser' from FashionMNIST, and classes 'airplane,' 'automobile,' 'bird,' 'cat,' and 'deer' from CIFAR10.

As we are the first to study curriculum learning in online class-incremental learning, we used a random curriculum designer as our baseline. The random designer randomly ranks the 120 curricula for each dataset. We repeated the random designers over 100 times with different random seeds, resulting in 100 sets of 120 randomly ranked curricula per dataset.

3.2. Continual Learning Algorithms

Among the CL algorithms surveyed in **Sec 2.1**, we chose two weight regularization methods: Elastic Weight Consolidation (EWC) [30] and Learning without Forgetting (LwF) [37]. EWC estimates the importance of all weights after each task and penalizes weight updates in proportion to their prior importance in the loss function. LwF uses the knowledge distillation loss [26] to regularize the current loss with soft targets acquired from a preceding version of the model. Replay-based CL algorithms involve joint training on old and new samples and often yield superior performance. We thus also include one replay method, where the images from previous tasks are randomly selected for the memory buffer and intermixed with the training data in the current task for replays. We fix the memory buffer size constant over all the tasks, which approximately equals the size of storing 2% of the entire training set in each dataset. See curriculum analysis of the replay method in **Sec S10** and **Fig S25**. However, these results should be interpreted with caution since the replay sequence of replay data interferes with the fixed class order in a given curriculum. We evaluate EWC, LwF, and naive replay alongside a "vanilla" fine-tuned method without any measures to prevent catastrophic forgetting.

The objective of this paper is not to exhaustively compare the performance of CL algorithms, but to study how curriculum affects the learning mechanism of each algorithm. For fair comparisons, we used a frozen SqueezeNet [27] pre-trained on a subset of 100 classes from ImageNet [14] (ImageNet100) as the feature extractor for all three CL algorithms. We ensured that the 100

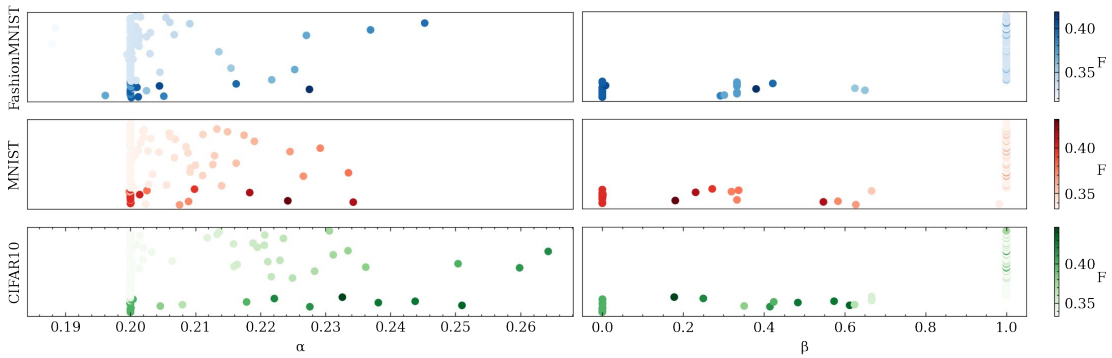


Figure 2: **Curricula influence the learning efficacy of the Vanilla CL algorithm (Sec 3.2) across MNIST, FashionMNIST, and CIFAR10 datasets (Sec 3.1).** We trained the vanilla CL algorithm on all curricula from each dataset. Each dot represents one curriculum. We report the distribution of average accuracy α over all the seen classes **(left panel, Sec 3.3)** and the distribution of forgetfulness β at the last task **(right panel, Sec 3.3)**. We introduced $\mathcal{F}$ as the measure of the learning efficacy of a given curriculum (**Sec 3.3**). See the colorbar on the right for different $\mathcal{F}$ values. Note that the y-axis does not carry any meaning. All the dots are randomly spread along the y-axis for easy visualization of the α and β distributions.

classes used for pre-training do not overlap with any of the classes selected for our CL experiments (**Sec 3.1**). The fine-tunable classification layers for all CL algorithms were initialized with the same set of random weights prior to continual training. Results in **Sec 5** are reported based on the performance of the three selected CL algorithms over 3 independent runs with different random seeds.

We used the standard public implementations of each CL algorithm from [39]. Note that the online CL results reported in our paper deviate from the original CL results in [39], because each training example can be seen only once in the online setting. All three CL algorithms are trained using the Adam optimizer with a learning rate of $1e^{-3}$. We performed hyperparameter searches for all CL algorithms. See **Sec 5.4** for results and discussions about hyper-parameter variations. However, we emphasize that each CL algorithm with a different set of hyper-parameters is conceptualized as a different "student." Though the same curriculum can be applied to all CL algorithms, the learning outcomes for different students might vary.

3.3. Evaluation Metrics

Learning Effectiveness $\mathcal{F}$. An effective CL algorithm quickly adapts to new classes with minimal forgetting of previously learned classes. To evaluate the learning efficacy of a CL algorithm for a given curriculum, we introduced the effectiveness score $\mathcal{F}$. The metric $\mathcal{F}$ accounts for two aspects: (1) the average accuracy α over all seen classes should be as high as possible, and (2) the accuracy difference β on the test images from the first task between the first task and the last task should be as small as possible. We formulate $\mathcal{F}$ as $\frac{2}{\beta+\frac{1}{\alpha}}$. $\mathcal{F}$ considers contributions from both α and β, while penalizing extreme values.

We report the distribution of $\mathcal{F}$ for all curricula over three datasets in **Fig 2 and Sec 5.1**. We see that a curriculum with high $\mathcal{F}$ (darker dots) has high α (**Fig 2**, left panel) and low β (**Fig 2**, right panel), highlighting how $\mathcal{F}$ reflects the overall learning effectiveness of a CL algorithm. We also reported $\mathcal{F}$ as a function of number of tasks (**Sec S5** and **Fig S29**) and found that the curriculum effect becomes more prominent with longer task sequences.

Recall@K. We used Recall@K to assess the teaching effectiveness of our curriculum designer (CD, **Sec 4**). Recall@K calculates the proportion of overlap between the top-K recommended curricula by our CD among the union set of all the top-K empirically ranked curricula by all $\mathcal{A}$s. We used the empirical curriculum rankings of EWC, LwF, and Vanilla for these calculations. Recall@K ranges from 0 to 1, where a higher value indicates better CD performance. Note that Recall@K also depends on the similarity of the curriculum effect among different CL algorithms.

Recall@K quantifies our CD's ability to identify the top-k empirically ranked curricula, but is not influenced at all by the rankings of less effective curricula. We argue that the CD's rank order among the most effective curricula is of special importance, particularly for applications where the goal is simply for the CD to find the most effective possible curriculum. We nonetheless include supplementary results for Spearman's rank correlation coefficient, which assesses the degree of agreement in rankings across all curricula (see **Sec S6**). One disadvantage of both Recall@K and rank correlation coefficients is that they do not account for the similarities between the curricula themselves. In the next section, we introduce the discrepancy measure $\mathcal{H}$ as a complementary measure that addresses this issue.

Curriculum Discrepancy $\mathcal{H}$. To assess the consistency between two sets of ranked curricula, we propose the curriculum discrepancy measure ($\mathcal{H}$), inspired by

gene sequence comparison methods [9]. $\mathcal{H}$ quantifies the dissimilarity between two sets of ranked curricula. Curriculum rankings are either determined by a CD or empirically determined based on $\mathcal{F}$ after exhaustively running $\mathcal{A}$ on all curricula of a given dataset.

We sort curricula using $\mathcal{F}$ in ascending order, and divide the range of $\mathcal{F}$ into 5 uniformly-sized bins or "tiers." Since studying the characteristics of the most effective curricula is critically important for the benefits of human and machine learning, in this work we focus on analyzing the curriculum discrepancy $\mathcal{H}$ from the top tier with the highest $\mathcal{F}$.

To calculate $\mathcal{H}$, we first assign each object class to a unique letter identifier and convert each curriculum to a string. As an example, 5 object classes in a dataset can be represented with letters A, B, C, D, and E. Any curriculum can then be represented as a combination of these 5 letters, such as $ABCED$ for curriculum 1 and $DECBA$ for curriculum 2. For a ranked curriculum set in the top tier, we can concatenate all the curricula into one string. In the example above, we have $ADBECCEBDA$. Given a pair of strings (two sets of ranked curricula), we use the Hamming distance to measure their curriculum discrepancy $\mathcal{H}$. The lower the $\mathcal{H}$ value, the higher the consistency: if the two ranked curricula are in exactly the same order, $\mathcal{H} = 0$. Note that Recall@K and ranking metrics like NDCG [28] and rank correlations [69] focus solely on comparing the order in which curricula are ranked, without reference to similarities among class orderings within curricula. We are unaware of any existing metrics that address rank similarities both within and between curricula.

In **Fig 2**, we observe a skewed distribution of $\mathcal{F}$ where there are a few curricula with very high $\mathcal{F}$ but many curricula with similarly low $\mathcal{F}$s. Thus, different tiers have different numbers of curricula. For a pair of ranked curricula sets in tier 5 where each set may have a different number of curricula, we choose the number of curricula in one set as a reference and compare it with the other curricula set containing an equal number of curricula. We do this once with each of the sets as the reference. The mean is then reported as the $\mathcal{H}$ for this pair of ranked curricula sets.

We conducted statistical tests for all experiments involving the above evaluation metrics, and report the results in **Sec S13**.

3.4. Human Benchmark

Novel Object Dataset (NOD)

We introduce the Novel Object Dataset (NOD) containing novel 3D objects with a categorical structure to test the continual learning abilities of humans and continual learning algorithms. NOD is a subset of the larger "Fribbles" dataset [6]. The dataset comprises 5 object families with 5 object instances per family. The instances and families differ in their main body structure and in the locations and shapes of various appendages (**Fig 3a**). We used Blender [18] to load the 3D object meshes, and rendered a 1920 $\times$ 1080 sized image of each object for every 10 degrees of azimuth and every 10 degrees of elevation, resulting in a total of 32,400 images (36^2 images per instance). We rendered the objects against a grey background to avoid confounding factors such as background biases. We randomly colored every object instance's body and appendages separately. To make the families easier for subjects to remember, we assigned a commonly used surname to each family.

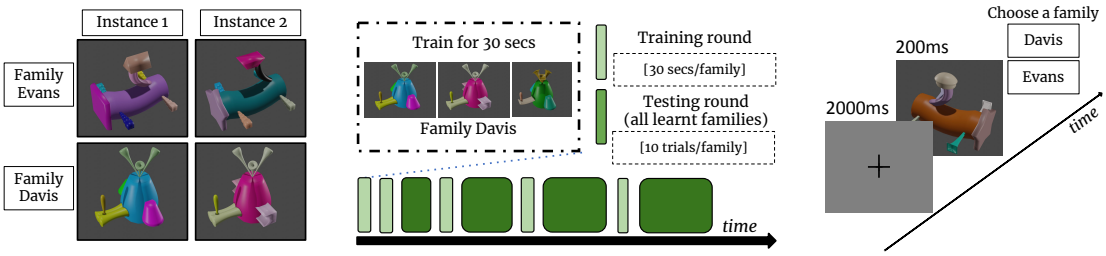
Psychophysics Experiments

Following standard protocols approved by our Institutional Review Board, we evaluated human performance on NOD using Amazon Mechanical Turk (MTurk) with the subjects' informed consent. The experiment duration on average was 20 minutes. Each participant was compensated. For quality control purposes, we also conducted in-lab experiments. We report the results from MTurk here and provide the details and results of the in-lab experiments in **Sec S1** and **Fig S2-S4**, **S6**, **S7**. The in-lab results support the conclusions drawn from the MTurk experiments.

We divided the experiment into 4 tasks, such that the first task had 2 object families and each subsequent task had 1 object family; this makes a total of $\binom{5}{2} \times 3! = 60$ possible curricula. Each subject is randomly assigned a curriculum. We recruited 242 subjects for a total of 34,848 test trials, with an average of 4 subjects tested on each curriculum. A schematic of the experiment is illustrated in **Fig 3b**. During the training rounds, the subjects were presented with 3 object instances per family that were shown rotating continuously along the azimuth. During the testing rounds, the subjects were shown a 640 $\times$ 480 sized GIF for each trial from the remaining 2 object instances per family (**Fig 3c**). Train and test instances differ. We took several precautions to ensure data quality and that subjects paid attention to the experiments (see **Sec S1**). Despite our simple stimulus design, we found that the majority of the participants ranked the experiments as difficult with an average difficulty score of 6.8/10 (10 = max. difficulty).

4. Curriculum Designer

We propose a proof-of-concept model, a Curriculum Designer (CD) for online class-incremental learning. Given a curriculum, our CD assigns a ranking score based on inter-class feature similarity. Our CD scores all possible curricula to produce a ranked set of curricula for each dataset. The low discrepancy in the ranked curricula of different continual learning algorithms (see the results in **Sec 5.4**) suggests that our CD does not necessarily need to depend on the feedback of a specific learning algorithm $\mathcal{A}$.



(a) Example objects (b) Class incremental learning setting (c) Test trial schematics

Figure 3: **Overview of human behavioral experiments in a class incremental setting**. (a) Two example object instances from each of two families in the Novel Object Dataset (NOD, **Sec 3.4**). (b) Experiment schematic. Subjects progressed through 4 tasks, each with a training and testing round. During training, subjects were presented with three rotating object instances per family for 30 seconds, with the goal of being able to recognize the objects presented in the testing round. In the first training round, 2 families were introduced. In subsequent training rounds, one additional family was introduced per task, without showing instances from previously learned families. During testing, subjects were tested on 10 trials from each learned family. The trial order was randomly shuffled during testing. (c) In each test trial, subjects were presented with a fixation cross (2000ms) followed by the stimulus (200ms). After the image offset, subjects were asked to choose the family of the presented object among all previously encountered families.

The objective of our CD is to propose a universal curriculum that improves learning outcomes of any given $\mathcal{A}$ relative to the average of randomly chosen curricula.

4.1. Feature Distance Confusion Matrix

Given an curriculum defined as $c_{t=1}, c_{t=2}, ..., c_{t=N}$, our CD uses an inter-class distance confusion matrix M of size $N \times N$, where any element $M_{(i,j)}$ represents a distance measure between two class prototypes, $c_{t=i}$ and $c_{t=j}$. To calculate a class prototype vector for each class, we used a teacher network to extract features from all images of the given class and took the vector mean. The feature distance $M_{(i,j)}$ between each pair of class prototypes $c_{t=i}$ and $c_{t=j}$ is calculated with the cosine distance. We conducted ablation experiments on distance metrics (**Sec 5.3**). In practice, extracting features from all images in a large dataset is computationally costly. Thus, we randomly sampled 500 images per class to compute the prototypes.

We used layers 1-12 of 2D-CNN SqueezeNet as our teacher network for computing class prototypes [27]. Drawing on the analogy that a human teacher has full knowledge of the subject they teach, the teacher network is pre-trained on ImageNet [14]. For consistency with the learning algorithms themselves (**Sec 3.2**), we fine-tuned the teacher network on the same set of 100 classes from ImageNet. The extracted feature vector of an input image is of size 1000. Prior knowledge of either the teacher or the student influences learning outcomes. We investigated the effect of prior knowledge in **Sec 5.3**.

4.2. Ranking Curricula

Given the inter-class distance confusion matrix M, we introduce a ranking score s that keeps track of the accumulative advantage v_t of choosing class c_t at incremental step t up to the final incremental step N: $s = \sum_{t=1}^{t=N} v_t$. Among all the curricula, the curriculum with the highest s is selected as the optimal. Next, we introduce the design of the advantage v_t for c_t and its motivations.

Drawing on the idea of metric learning [11] as well as the theoretical and practical foundations behind the impact of task ordering [38, 34], we choose the class $c_{t=1}$ at the first incremental step with the following criteria: the variance of the distances between the selected class prototype and the other classes' prototypes should be as small as possible. Intuitively, lower class distance variance implies relatively similar distances to other classes: the first class is near the center of the multivariate class feature distribution. Starting to learn from the class comprising features shared with most other classes facilitates positive knowledge transfer when learning other classes at later steps. Thus, to encourage our CD to prioritize selecting the first class with the smallest distance variance, we define the advantage $v_{t=1}$ at the first incremental step as $1 - Var(\{M_{(1,j)}\}_{j=2}^{N})$, where j is the corresponding class c_j at incremental step $t = j$ and $Var(\cdot)$ is a function computing the variance from a set of distances.

Subsequently, to eliminate catastrophic forgetting over incremental steps, we draw ideas from replay mechanism in CL [65, 47, 2, 10, 44, 40, 5] and select the last class $c_{t=N}$ based on the following criteria: the prototype of the selected class should have the smallest distance to $c_{t=1}$. The design motivation is to ensure that $c_{t=N}$ is the most similar to $c_{t=1}$ in terms of features. While $\mathcal{A}$ learns to classify $c_{t=N}$, these common features are functionally analogous to a feature replay of $c_{t=1}$, which regularizes the parameters of $\mathcal{A}$ to prevent forgetting. Correspondingly, to encourage CD to prioritize replay-like class selection at the last incremental

step, we define the advantage $v_{t=N}$ as $1 - M_{(N,1)}$.

Conversely, for the selection of the second class to learn at step $t = 2$, we encourage CD to select the class whose prototype is the farthest away from its previous class $c_{t=1}$. This is in accordance with the classical notion in the curriculum learning literature that a curriculum should always be arranged in order, from easiest to the hardest [8]. The farther away the distance between two class prototypes, the easier it is for the algorithm $\mathcal{A}$ to learn the classification boundary between these two visually distinct classes. In this case, we define the advantage $v_{t=2}$ as $M_{(2,1)}$

We complete the ranking process of a given curriculum by iteratively performing the advantage evaluation back and forth over all subsequent incremental steps until we have examined all the classes. We summarize the piece-wise advantage function below:

$$v_t = \begin{cases} 1 - \mathrm{Var}(\{M_{(1,j)}\}_{j=2}^{N}) & , t = 1 \\ M_{t,t-1} & , 1 < t \leq \lfloor \frac{N}{2} \rfloor \\ 1 - M_{t,N-t+1} & , \lfloor \frac{N}{2} \rfloor < t \leq N \end{cases}$$

For every curriculum from a dataset, we compute its corresponding ranking score s by summing the advantage for each class in a curriculum. Although it is daunting to perform heuristic searches for optimal curricula by exhaustively going through all possible curricula for a dataset, it is still computationally efficient for our CD given that it only scores curricula based on a 2D distance confusion matrix M. See **Algorithm 1** (**Supp.**) for the pseudo-code of CD implementation.

5. Results

5.1. Curriculum Strongly Impacts Performance

Fig 2 highlights the effect of curricula on the vanilla $\mathcal{A}$ (**Sec 3.2**) over all three datasets (**Sec 3.1**). We observed a large variance in average accuracy α, which ranged from 19% to 26% depending on the curriculum. This implies that curriculum strongly influences the overall performance over all tasks for the vanilla $\mathcal{A}$ (**Sec 3.3**). β reflects the degree of forgetting of the first task while learning later tasks (**Sec 3.3**). The large variance in β indicates that curriculum plays a significant role in preventing the vanilla $\mathcal{A}$ from forgetting the first class. The empirically optimal curriculum results in a more gradual decline in the accuracy on images from the initial task as subsequent tasks are introduced, which leads to a smaller β.

We introduced the learning effectiveness score $\mathcal{F}$, which incorporates both α and β (**Sec 3.3**). Darker dots in **Fig 2** indicate higher $\mathcal{F}$, generally implying larger α and smaller β. For example, for a model which learns the 1st task perfectly well and achieves 100% accuracy but fails to adapt to any new tasks (0% for the other four classes), we can calculate its effectiveness scores as: $\alpha = (100\% + 4 \times 0\%)/5 = 20\%$, $\beta = 100\% - 100\% = 0\%$ and $\mathcal{F} = 2/(0 + 5) = 0.4$. Another instance would be $\alpha = 0.25$ but higher β, where the CL model learns a bit of each task and tends to forget previous tasks. The $\mathcal{F}$ differs by 0.09, 0.07 and 0.07 from the best to the worst curriculum for MNIST, FashionMNIST and CIFAR10. These results from regularization-based CL algorithms $\mathcal{A}$s (**Sec 3.2**) are constrained by the online class-incremental setting. Their $\mathcal{F}$ scores are in contrast to those of the highly effective replay method (**Sec 3.2**) with an average $\mathcal{F} = 0.99, 0.87, 0.69$ on MNIST, FashionMNIST and CIFAR10, which often serve as upper bounds of continual learning performances. We present the distributions of α, β, and $\mathcal{F}$ for EWC [30] and LwF [37] in **Sec S4** and **Fig S14-S17**. The curricula trends observed in the discussion here are also applicable to these two algorithms.

5.2. Our CD Predicts Optimal Curricula

To evaluate the effectiveness of the predicted curricula by our CD for CL algorithms $\mathcal{A}$s, we report results in terms of Recall@K (**Sec 3.3**) in **Fig 4**. We used a random curriculum designer as a baseline for comparison to our CD. Across all three datasets, our CD (blue) outperformed the random model (green), particularly at small k values. Our CD achieves peaks in Recall@K of 0.5, 0.2, and 1 at K=2, K=5 and K=10 for MNIST, FashionMNIST and CIFAR10 respectively.

Our results suggest that the CD performance does not depend on data complexity, as CD performs well on both MNIST and CIFAR10 despite CIFAR10 having more complex image features. Our curriculum designer exhibits remarkable performance on CIFAR-10. A plausible conjecture could be that these results are attributed to the striking resemblance between CIFAR-10 and ImageNet. The latter was employed for pre-training and served as the fundamental feature extractor for our curriculum designer. We provide visualizations of the top-5 empirically-determined and CD-predicted curricula for all datasets in **Fig S8-S13**. The top curricula seem to align with the intuitions behind our CD design (**Sec 4**). Although our CD is effective in most cases, there is considerable room for improvement. We note that our CD has relatively weak performance on FashionMNIST, with Recall@K below the random CD for $K < 4$ and only slightly above random for $K \geq 4$.

5.3. Analysis of CD Design Decisions

To evaluate the impact of individual design choices in our CD, we conducted experiments with variations of our CD on MNIST and presented the Recall@K results for K=5, 10, and 20 in **Fig 5**. First, instead of the cosine distance metric used in our CD, we changed the

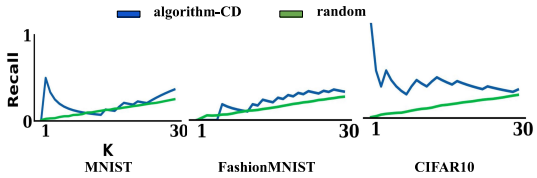


Figure 4: **Our Curriculum Designer (CD) predicts optimal curricula better than a random CD.** Recall@K (**Sec 3.3**) of our CD (blue, **Sec 4**) and a random curricula designer (green) are reported as a function of K ranging from 1 to 30 across all three datasets (**Sec 3.1**), where K is the number of top curricula included in the metric.

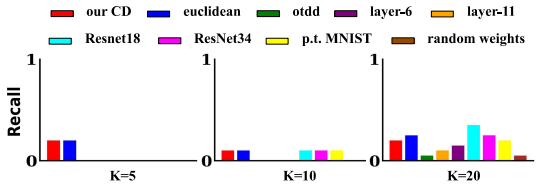


Figure 5: **Ablation results on our CD**. Recall@K bar plots for k=5, 10, and 20 with our CD and its ablations compared against the empirical curricula ranking determined by all continual learning algorithms $\mathcal{A}$s (**Sec 3.2**) on MNIST (**Sec 3.1**) for paradigm-I (5 classes, **Sec 3.1**). See **Sec 5.3** for the description of ablated CDs.

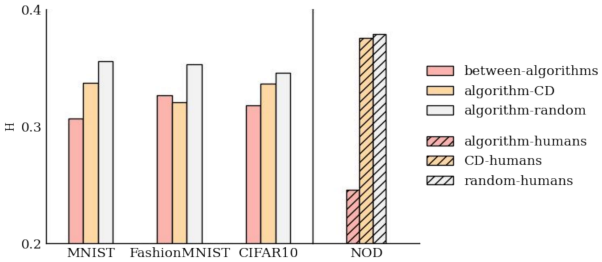


Figure 6: **There exists low discrepancy on optimal curricula determined by between-algorithms, algorithm-CD, algorithm-humans, and CD-humans.** Left panel: curricula discrepancy $\mathcal{H}$ (**Sec 3.3**) is reported between pairs of CL algorithms $\mathcal{A}$s (between-algorithm, blue), between $\mathcal{A}$s and our CD (algorithm-CD, green), between $\mathcal{A}$ and the random designer (algorithm-random, orange). Right panel: $\mathcal{H}$ is reported on NOD dataset between $\mathcal{A}$s and humans (algorithm-human, blue hashed), between CD and humans (CD-humans, green hashed), and between the random designer and humans (random-humans, orange hashed) (**Sec 5.4**).

distance metric to Euclidean and Optimal Transport Dataset Distance (OTDD) [4] (euclidean and otdd). The ablated model with Euclidean outperforms OTDD and performs competitively well as our CD with cosine distance. This implies that the choice of measure for the inter-class distance is essential for curriculum designs. Next, we evaluated the effect of changing the layers used in the feature extractor to compute the distance confusion matrix M by using layers 6 and 11 (layer-6 and layer-11). We observed that using layer-11 or layer-6, on average, leads to a performance decrement in recall at earlier Ks. This implies that the higher layers of the network produce more class-representative features that are useful for curricula ranking. Furthermore, we replaced our default feature extractor SqueezeNet with ResNet34 and ResNet18 [23]. Though the recall of these ablated models is not as high as our CD at K=5, they achieve a high recall at K=10. This implies that a change in architecture does not lead to dramatic performance deterioration in continual learning.

To study the effect of prior knowledge of our CD as the teacher, we introduce two variations. First, we pre-trained the feature extractor of our CD on MNIST (p.t. MNIST). Compared with our original CD pre-trained on 100 classes of ImageNet (**Sec 4.1**), we did not observe any increase in recall at K=5; but we observed the high recall at K=10. It is possible that the 100 classes from ImageNet share similar features with the classes from MNIST. Drawing on an example in pedagogy that a teacher with general math knowledge can teach arithmetic as efficiently as a teacher with only arithmetic-specific expertise, this experiment indicates that a teacher with broad knowledge in the field is as good as a teacher with area-specific knowledge. Next, we evaluate our CD with the weights of its feature extractor randomly initialized (random-teacher). With the observation of the drastic drop in recall even at K=20, we conclude that prior knowledge of a teacher is indeed important for designing efficient curricula.

5.4. Analysis on Curriculum Agreement

We set out to study the extent of agreement among curricula empirically optimized for individual students. For example, do the most effective curricula for EWC share commonalities with the most effective curricula for LwF? To address this question, we report the discrepancy $\mathcal{H}$ between any sets of ranked curricula determined empirically by CL algorithms $\mathcal{A}$, by our CD, and by the random curriculum designer on three image datasets of varying complexity (**Fig 6**). A decrease in $\mathcal{H}$ indicates an increase in the agreement (**Sec 3.3**). As a lower bound ("between-algorithms"), we first calculated the averaged discrepancy $\mathcal{H}$ over all pairs of $\mathcal{A}$s chosen among Vanilla, EWC, and LwF (**Sec 3.2**). We consistently observe a large $\mathcal{H}$ decrease in "between-algorithms" relative to "algorithm-random" (average discrepancy $\mathcal{H}$ between sets of empirically ranked curricula and set of randomly ranked curricula). This implies that continual learning algorithms $\mathcal{A}$s agree with each other in empirically ranking the most effective curricula, more so than with random curricula. In other words, curricula that work well for one $\mathcal{A}$ tend to work well for another. We also examined the effect of $\mathcal{A}$'s hyperparameters on curriculum agreement (see **Sec S3** and **Fig S5**), and found that the relative efficacy of curricula is

consistent even with variations in the number of epochs, the learning rate and the network initialization.

We also assessed the discrepancy $\mathcal{H}$ between our CD's curriculum rankings and empirical curriculum rankings from $\mathcal{A}$. Across the three datasets (MNIST, FashionMNIST and CIFAR10, **Sec 3.1**), there is an average decrease of 0.02 in $\mathcal{H}$ from algorithm-random to algorithm-CD. It implies that our CD can predict optimal curricula well aligned with the curricula determined by $\mathcal{A}$s. However, $\mathcal{H}$ in between-algorithms is still higher than in algorithm-CD, indicating that the curricula ranked empirically by different $\mathcal{A}$s are more consistent with one another than with those ranked by our CD.

The right panel in **Fig 6** shows the agreement in algorithm-humans, CD-humans, and random-humans on the Novel Object Dataset (NOD, **Sec 3.4**). There is an $\mathcal{H}$ decrease of 0.13 from random-humans to algorithm-humans. This indicates a notable degree of agreement between optimal curricula for humans and $\mathcal{A}$s. We further observe that there is a slight decrease in $\mathcal{H}$ from random-humans to CD-humans, indicating a minimal degree of alignment between humans and our CD. However, we notice that there still exists a huge gap in $\mathcal{H}$ from algorithm-humans to CD-humans.

6. Discussion

Curriculum design is an important problem in both machine learning and human education. Key goals for both humans and machines include maximizing forward knowledge transfer across tasks while minimizing forgetting of previous tasks. In practice, there are numerous potential curriculum design considerations, such as the ordering of training examples within and between classes and tasks, hierarchical learning across super-categories and sub-categories, learning characteristics of students, and feedback from students. Here, we introduce an initial proof-of-concept curriculum designer, which designs effective curricula for multiple CL algorithms by optimizing the ordering of a sequence of continuously learned tasks.

While curriculum design proves effective for enhancing CL algorithms, its direct translation to human learning still encounters challenges. To benchmark curriculum efficacy in humans, we introduced the Novel Object Dataset (NOD) and conducted human behavioral experiments. We observed a high discrepancy between optimal curricula ranked by our AI teacher and effective for human learning. There could be multiple reasons for this. First, the visual diets for humans and our AI teacher are different. Humans learn from temporally correlated video streams, which our AI teacher does not take into account. Second, there remains a gap between the background knowledge of humans and our AI teacher. Humans accumulate rich experiences through interactions with the real world involving multiple sensory modalities, but our AI teacher has been limited to knowledge from static naturalistic images in vision. Third, human individuals have large variability in learning due to individual cognitive capabilities and knowledge backgrounds. Our AI teacher lacks specialized curriculum designs for learning in individual humans.

To resemble a human learning process, we took initial efforts and formulated our study of curriculum learning in the online class-incremental learning setting. Given computational resource constraints, we only exhaustively and empirically surveyed the 5-class and 10-class incremental settings on 3 CL algorithms across 3 datasets (**Sec 3.1**). Additional studies could explore a wider range of problem settings, such as task-incremental learning and long-range CL with many classes. As a preliminary follow-up, we explored the effect of curriculum on the problem of visual question answering in function incremental settings (**Sec S12**, **Fig S26**). We also investigated offline class-incremental learning, allowing the CL models to make multiple passes over the data within each task (**Sec S11**, **Fig S23**). Moreover, we extended our online learning tests to replay-based CL approaches (**Sec S10**, **Fig S25**). Throughout all of these experiments, we observe curriculum effects that persist across variations in problem settings, datasets, and continual learning algorithms.

AI for education and education for AI remain open challenges. Our study establishes a methodology for the community to evaluate and benchmark curriculum design approaches for both humans and AI. The insights obtained from our work open doors to many research opportunities, such as AI-assisted learning and education systems for both AI and human students.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-025), its NRFF award NRF-NRFF15-2023-0001, the National Science Foundation under grant number NSF CCF 1231216, the National Institutes of Health under grant number NIH R01EY026025, and the National Institute of General Medical Sciences under award number T32GM144273. We also acknowledge Mengmi Zhang's Startup Grant from Agency for Science, Technology, and Research (A*STAR), and Early Career Investigatorship from Center for Frontier AI Research (CFAR), A*STAR. The authors declare that they have no competing interests. The funders had no role in study design, data collection and analysis, the decision to publish, or the preparation of the manuscript.

References

[1] Tameem Adel, Han Zhao, and Richard E Turner. Continual learning with adaptive weights (claw). *arXiv preprint arXiv:1911.09514*, 2019. 2

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *arXiv preprint arXiv:1903.08671*, 2019. 2, 6

[3] Eugene L Allgower and Kurt Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 2012. 2

[4] David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020. 8

[5] 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*, pages 8218–8227, 2021. 2, 6

[6] Tom J Barry, James W Griffith, Stephanie De Rossi, and Dirk Hermans. Meet the fribbles: novel stimuli for use within behavioural research. *Frontiers in Psychology*, 5:103, 2014. 5

[7] Samuel J Bell and Neil D Lawrence. The effect of task ordering in continual learning. *arXiv preprint arXiv:2205.13323*, 2022. 2

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 2, 7

[9] Oliver Bonham-Carter, Joe Steele, and Dhundy Bastola. Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis. *Briefings in bioinformatics*, 15(6):890–905, 2014. 5

[10] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 2, 6

[11] Haoxing Chen, Huaxiong Li, Yaohui Li, and Chunlin Chen. Multi-level metric learning for few-shot image recognition. In *International Conference on Artificial Neural Networks*, pages 243–254. Springer, 2022. 6

[12] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1431–1439, 2015. 2

[13] Jaehoon Choi, Minki Jeong, Taekyung Kim, and Changick Kim. Pseudo-labeling curriculum for unsupervised domain adaptation. *arXiv preprint arXiv:1908.00262*, 2019. 2

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3, 6

[15] Shayan Doroudi, Vincent Aleven, and Emma Brunskill. Where's the reward? a review of reinforcement learning for instructional sequencing. *International Journal of Artificial Intelligence in Education*, 29:568–620, 2019. 3

[16] Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Learning to teach. *arXiv preprint arXiv:1805.03643*, 2018. 2, 3

[17] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017. 2

[18] Sergei Valer'evich Filippov. Blender software platform as an environment for modeling objects and processes of science disciplines. *Keldysh Institute Preprints*, (230):1–42, 2018. 5

[19] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR, 2017. 2, 3

[20] Siavash Golkar, Michael Kagan, and Kyunghyun Cho. Continual learning via neural pruning. *arXiv preprint arXiv:1903.04476*, 2019. 2

[21] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. PMLR, 2017. 2, 3

[22] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 135–150, 2018. 2, 3

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[24] Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. 2018. 2

[25] Joy He-Yueya and Adish Singla. Quizzing policy using reinforcement learning for inferring the student knowledge state. *International Educational Data Mining Society*, 2021. 3

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

[27] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 3, 6

[28] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. 5

[29] Tae-Hoon Kim and Jonghyun Choi. Screenernet: Learning self-paced curriculum for deep neural networks. *arXiv preprint arXiv:1801.00904*, 2018. 2, 3

[30] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2, 3, 7

[31] Pascal Klink, Hany Abdulsamad, Boris Belousov, and Jan Peters. Self-paced contextual reinforcement learning. In *Conference on Robot Learning*, pages 513–529. PMLR, 2020. 2, 3

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3

[33] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 3

[34] Sebastian Lee, Sebastian Goldt, and Andrew Saxe. Continual learning in the teacher-student setup: Impact of task similarity. In *International Conference on Machine Learning*, pages 6109–6119. PMLR, 2021. 6

[35] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in neural information processing systems*, pages 4652–4662, 2017. 2

[36] Stan Weixian Lei, Difei Gao, Jay Zhangjie Wu, Yuxuan Wang, Wei Liu, Mengmi Zhang, and Mike Zheng Shou. Symbolic replay: Scene graph as prompt for continual learning on vqa task. *arXiv preprint arXiv:2208.12037*, 2022. 2

[37] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2, 3, 7

[38] Sen Lin, Peizhong Ju, Yingbin Liang, and Ness Shroff. Theory on forgetting and generalization of continual learning. *arXiv preprint arXiv:2302.05836*, 2023. 6

[39] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas Tolias, Simone Scardapane, Luca Antiga, Subutai Amhad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: an end-to-end library for continual learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2nd Continual Learning in Computer Vision Workshop, 2021. 4

[40] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 2, 6

[41] Reza Lotfian and Carlos Busso. Curriculum learning for speech emotion recognition from crowdsourced labels. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(4):815–826, 2019. 2

[42] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022. 2

[43] Tong Mu, Shuhan Wang, Erik Andersen, and Emma Brunskill. Automatic adaptive sequencing in a webgame. In *Intelligent Tutoring Systems: 17th International Conference, ITS 2021, Virtual Event, June 7–11, 2021, Proceedings 17*, pages 430–438. Springer, 2021. 3

[44] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017. 2, 6

[45] Meng Qu, Jian Tang, and Jiawei Han. Curriculum learning for heterogeneous star network embedding via deep reinforcement learning. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 468–476, 2018. 2, 3

[46] Jathushan Rajasegaran, Munawar Hayat, Salman H Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

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

[48] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7374–7383, 2019. 2, 3

[49] Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. Data parameters: A new family of parameters for learning a differentiable curriculum. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[50] Jonathan Schwarz, Jelena Luketina, Wojciech M Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. *arXiv preprint arXiv:1805.06370*, 2018. 2

[51] Ayon Sen, Purav Patel, Martina A Rau, Blake Mason, Robert Nowak, Timothy T Rogers, and Jerry Zhu. For teaching perceptual fluency, machines beat human experts. In *CogSci*, 2018. 3

[52] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018. 2

[53] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Transferable curriculum for weakly-supervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4951–4958, 2019. 2, 3

[54] Adish Singla, Anna N Rafferty, Goran Radanovic, and Neil T Heffernan. Reinforcement learning for education: Opportunities and challenges. *arXiv preprint arXiv:2107.08828*, 2021. 3

[55] Iram Siraj-Blatchford, Stella Muttock, Kathy Sylva, Rose Gilden, and Danny Bell. Researching effective pedagogy in the early years. 2002. 1

[56] Petru Soviany, Claudiu Ardei, Radu Tudor Ionescu, and Marius Leordeanu. Image difficulty curriculum for generative adversarial networks (cugan). In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3463–3472, 2020. 2

[57] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum self-paced learning for

cross-domain object detection. *Computer Vision and Image Understanding*, 204:103166, 2021. 2, 3

[58] Ye Tang, Yu-Bin Yang, and Yang Gao. Self-paced dictionary learning for image classification. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 833–836, 2012. 3

[59] Radu Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P Papadopoulos, and Vittorio Ferrari. How hard can it be? estimating the difficulty of visual search in an image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2157–2166, 2016. 2, 3

[60] Kaiping Wang, Yan Wang, Bo Zhan, Yujie Yang, Chen Zu, Xi Wu, Jiliu Zhou, Dong Nie, and Luping Zhou. An efficient semi-supervised framework with multi-task and curriculum learning for medical image segmentation. *International journal of neural systems*, 32(09):2250043, 2022. 2

[61] 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. 2

[62] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[63] Jerry Wei, Arief Suriawinata, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Louis Vaickus, Charles Brown, Michael Baker, Mustafa Nasir-Moin, Naofumi Tomita, et al. Learn like a pathologist: curriculum learning by annotator agreement for histopathology image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2473–2483, 2021. 2

[64] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? In *International Conference on Learning Representations*, 2021. 3

[65] 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*, pages 374–382, 2019. 2, 6

[66] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 247–263. Springer, 2020. 2

[67] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 3

[68] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. In *European Conference on Computer Vision*, pages 608–624. Springer, 2020. 2, 3

[69] Jerrold H Zar. Spearman rank correlation. *Encyclopedia of biostatistics*, 7, 2005. 5

[70] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org, 2017. 2

[71] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: from clean label detection to noisy label self-correction. In *International Conference on Learning Representations*, 2021. 2