Learning to Learn: How to Continuously Teach Humans and Machines

Parantak Singh[1], Li You[1,2], Ankur Sikarwar[1], Weixian Lei[3], Difei Gao[3],
Morgan Bruce Talbot[4, 5], Ying Sun[1], Mike Zheng Shou[3], Gabriel Kreiman[6], Mengmi Zhang[1]

[1] CFAR and I2R, Agency for Science, Technology and Research, Singapore, [2] University of Wisconsin-Madison, USA,
[3] Show Lab, National University of Singapore, Singapore, [4] Boston Children's Hospital, Harvard Medical School, USA,
[5] Harvard-MIT Health Sciences and Technology, Massachusetts Institute of Technology, USA, [6] Harvard University, USA,
Address correspondence to mengmi@i2r.a-star.edu.sg

Abstract

Our education system comprises a series of curricula. For example, when we learn mathematics at school, we learn in order from addition, to multiplication, and later to integration. Delineating a curriculum for teaching either a human or a machine shares the underlying goal of maximizing the positive knowledge transfer from early to later tasks and minimizing forgetting of the early tasks. Here, we exhaustively surveyed the effect of curricula on existing continual learning algorithms in the class-incremental setting, where algorithms must learn classes one at a time from a continuous stream of data. We observed that across a breadth of possible class orders (curricula), curricula influence the retention of information and that this effect is not just a product of stochasticity. Further, as a primary effort toward automated curriculum design, we proposed a method capable of designing and ranking effective curricula based on inter-class feature similarities. We compared the predicted curricula against empirically determined effectual curricula and observed significant overlaps between the two. To support the study of a curriculum designer, we conducted a series of human psychophysics experiments and contributed a new Continual Learning benchmark in object recognition. We assessed the degree of agreement in effective curricula between humans and machines. Surprisingly, our curriculum designer successfully predicts an optimal set of curricula that is effective for human learning. There are many considerations in curriculum design, such as timely student feedback and learning with multiple modalities. Our study is the first attempt to set a standard framework for the community to tackle the problem of teaching humans and machines to learn to learn continuously.

1. Introduction

When learning mathematics, students advance through a curriculum that guides them to first learn addition, then multiplication, algebra, and integration such that each new concept builds upon existing knowledge (**Fig 1**). Careful design of curricula for humans and machines can enable an incremental learning process that allows for maximum positive knowledge transfer to new tasks and minimal forgetting of learned tasks.

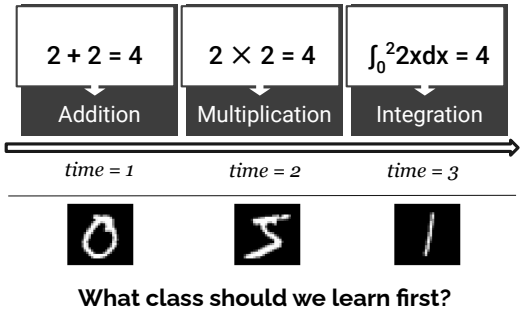


Figure 1. **Curriculum in a school setting**. A natural math curriculum prescribes learning about addition before multiplication, and learning both of these before learning about integrals. Similarly, in order to teach machines to learn to recognize numbers, what would be the best teaching sequence?

The curriculum learning literature in machine learning mainly focuses on weakly-supervised [20, 43, 47], unsupervised [39, 45, 53], and reinforcement learning [16, 27, 36] settings. Existing works have demonstrated the benefits of curricula in improving generalization ability and convergence speed, but only by measuring intra-class difficulty and scheduling examples within a *single* task. Little is known about the effect of curricula across sequences of *multiple* tasks.

To close this research gap, we investigated the effects of curriculum in Continual Learning (CL) settings. Inspired by the fact that humans can learn entirely from a stream of visual inputs as we move about the world, we specifically focused on the incremental class setting in stream learning. Here, machines learn to recognize objects incrementally with a single pass through a continuous stream of image inputs. Our empirical results suggest that curriculum

choices greatly influence retention and forgetting in CL algorithms. We also quantitatively assessed the effects of hyperparameter adjustments in various CL algorithms. We observed that the most empirically effective curricula are highly correlated across multiple different CL algorithms.

Building upon these results, we proposed an automatic curriculum designer (CD) capable of designing and ranking curricula. In a nutshell, our CD enables object classes with the closest feature spaces to be taught to neural networks or humans at distant time steps. Unlike pre-defined curriculum learning algorithms [34, 44, 47, 49], our CD does not require prior knowledge from domain experts, or any human intervention. We stress-tested competitive CL algorithms against the curricula ranked by our CD. The results suggest that our CD improves retention and minimizes forgetting on these algorithms. Among the $92nd$ percentile of the effective curricula empirically found, our CD consistently outperforms chance in terms of the absolute effective curricula counts.

To assess the degree of agreement between the most effective curricula for humans and machines, we conducted a series of human psychophysics experiments and contributed a new CL benchmark for machines and humans involving recognition of novel objects. Results on human-tested curricula and our CD-ranked curricula show a higher agreement between the two compared to random sets of curricula. We summarize our main contributions as follows:

- We established a methodology to study class-incremental curricula in stream learning.
- We introduced a new novel-object recognition dataset to benchmark the effectiveness of class-incremental curricula for humans and CL algorithms.
- We developed an automated curriculum designer that is capable of designing and ranking effective curricula.
- We provided insights into the commonalities among empirically effective curricula for various CL algorithms and humans.

2. Related Works

2.1. Continual Learning

CL strategies can be grouped into the following categories: (1) weight regularization, (2) replay, and (3) architecture expansion. Regularization methods cache weights trained on previous tasks while new tasks are trained with constraints on weight updates [9, 22, 26, 30, 31, 54]. For instance, Elastic Weight Consolidation (EWC) [26] estimates the importance of parameters for old tasks and penalizes weight changes during new tasks accordingly, and Learning without Forgetting (LwF) [31] stores logits from past versions of the model and performs knowledge distillation [23] with these logits in subsequent tasks. Replay-based strategies involve storing a subset of samples from previous tasks and interspersing them with training data from newly encountered tasks to mitigate forgetting [2, 5, 9, 33, 35, 38, 51]. Architecture adaptation methods involve expanding or restructuring neural networks to assimilate new tasks [1,15,18,22,26,30,31,37,41,42,54].

These methods are predominantly evaluated in a class-incremental setting where many passes over data from each task are permitted. In contrast, humans can learn from a non-repeating stream of experience while preserving prior knowledge and continually transferring knowledge to new tasks [21]. Thus, we focused on the stream learning paradigm in the class-incremental setting.

A standard evaluation procedure in CL is to report the average performance over at least three runs with *random* class orders. However, the effects of specific class orders for training these algorithms in class-incremental stream learning settings are less explored. To address this research gap, we exhaustively studied the class presentation order during training.

2.2. Curriculum Learning

Curriculum learning was formalized as the paradigm of learning with a meaningful training order, traditionally progressing from "easier" to "harder" data [3, 7]. Previous works in Curriculum Learning can be categorized into Predefined Curriculum Learning [7, 11, 12, 44] and Automatic Curriculum Learning [14,19,25,48]. 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 [14, 19, 25]. These methods are often deployed in weakly-supervised [20, 43, 47], unsupervised [39, 45, 53] and reinforcement learning [16, 27, 36] settings. In computer vision, curriculum learning approaches are almost exclusively directed toward measuring intra-class example difficulty. Existing methods specifically focus on a *single* multi-class object recognition task [20, 40, 46, 50] in which all examples from each class can be trained on *multiple* times. The efficacy of these proposed curricula is often evaluated in terms of generalization to the test data and the convergence speed during training.

However, one recent study highlighted how the most widely-used curriculum design strategy (increasing difficulty) may not always be optimal, and how anti-curriculum ("harder" to "easier") or random ordering

yield comparable results in multi-class object recognition settings [50]. The study reported that curriculum is only beneficial with limited training iterations or when the training set is small. Aligned with these training constraints, we investigated the effect of curriculum on CL algorithms under stringent conditions where training is limited to a single pass through the data. Across all of our experiments, we consistently observed that effective curricula improve performance in the class-incremental stream learning setting. The benefits of curricula are also observed in human learning during psychophysical experiments with our newly contributed CL benchmark dataset.

3. Experiments

We introduced a methodology to study class-incremental curricula in stream learning. 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 $\mathcal{A}$ 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 conducted our experiments on three datasets: MNIST [29], FashionMNIST [52], and CIFAR10 [28]. Each dataset consists of 10 object classes. Ideally, each curriculum is a permutation of 10 object classes, i.e. 10! (approximately 3 million) curricula. Thus, with limited computational resources, running all permutations is very difficult. To mitigate this issue, we introduced two paradigms: First, we used a subset of the dataset comprising 5 classes with 1 class per incremental step. Second, we made 5 fixed tasks of 2 set classes, and then varied the ordering of the 5 tasks instead of all 10 classes separately. In both cases, we have a total of $5! = 120$ curricula. In the main text, without loss of generality, we only present and discuss results for the first paradigm. See **Supp.** for details of class grouping and results for the second paradigm. In general, the conclusions drawn in the first paradigm also hold true in the second. Next, we list the selected 5 classes from each dataset for the first paradigm:

MNIST ($60,000$ training images, $10,000$ test images). Selected: '0', '1', '2', '3', '4'.

FashionMNIST ($60,000$ training images, $10,000$ test images). Selected: 'coat', 'dress', 'pullover', 'top', 'trouser'.

CIFAR10 ($50,000$ training images, $10,000$ test images). Selected: 'airplane', 'automobile', 'bird', 'cat', 'deer'.

Because to our knowledge we are the first to study curriculum learning in continual stream learning, we used a random curriculum designer (chance) as our baseline. For each dataset, the chance model randomly ranks the 120 curricula. We conducted 100 runs for the chance model.

3.2. Continual Learning Algorithms

Among the CL algorithms surveyed in **Sec 2**, we chose three weight-regularization methods: Vanilla (fine-tuning), Elastic Weight Consolidation (EWC) [26], and Learning without Forgetting (LwF) [31]. Vanilla is a fine-tuned feed-forward neural network model without any measures to prevent catastrophic forgetting. EWC estimates the importance of all weights after each incremental ste, and penalizes weight updates in proportion to their prior importance in the loss function. LwF utilizes the Knowledge Distillation loss [23] to regularize the current loss with soft targets acquired from a preceding version of the model.

The objective of this paper is *not* to exhaustively compare which CL algorithm performs the best, but to study how curricula would benefit the learning mechanism of each algorithm. Replay-based CL algorithms involve joint training on old and new samples. The replay sequence of old data interferes with the fixed class order in a given curriculum. To systematically study curricula for each dataset, replay-based algorithms are excluded from our experiments.

For fair comparison, we used SqueezeNet [24] as the backbone for all three CL algorithms. Rather than using a SqueezeNet model pre-trained on ImageNet [13], we trained a randomly initialized SqueezeNet on a subset of 100 classes from ImageNet, where these 100 classes do not overlap with any of the classes selected for our experiments (**Sec 3.1**). Before continual training, the network parameters are randomly initialized using a uniform distribution. The same set of initial network parameters are used for each of the chosen CL algorithms. To ensure that the observed curriculum effect is not due to random network parameter initialization, we conducted 3 independent runs with different seeds. Results in **Sec 5** are reported based on the performance of the three selected CL algorithms over all 3 runs.

We used the standard implementations of each CL algorithm from the Avalanche CL library [32]. All three CL algorithms are trained using the Adam optimizer with a learning rate of $1e^{-3}$. Our code and data are publicly available here.

3.3. Evaluation Metrics

We introduced three 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

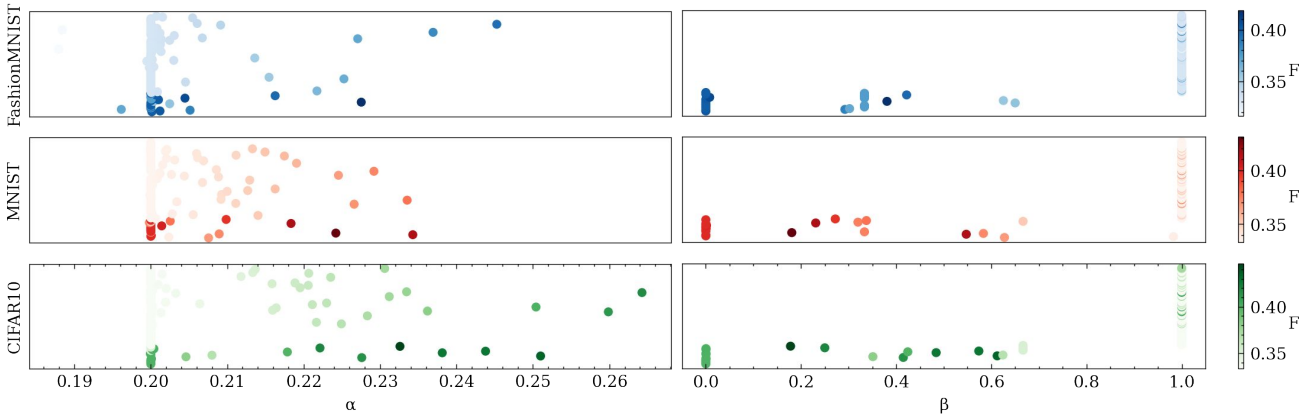


Figure 2. **Curricula influence the learning efficacy of the Vanilla CL algorithm (Sec 3.2) across three datasets: MNIST, FashionMNIST, and CIFAR10 (Sec 3.1)**. We trained the vanilla CL algorithm on all curricula from each dataset. Each dot represents one curriculum. We reported the average accuracy α over all the seen classes, highlighting how the algorithm adapts to learn new tasks **(left panel, Sec 5.1)**. The accuracy difference β reflects the forgetfulness of the algorithm across tasks **(right panel, Sec 5.1)**. We introduced $\mathcal{F}$ as the measure of the learning efficacy of a given curriculum **(Sec 3.3)** taking both α and β into account. The color gradient denotes the magnitude of $\mathcal{F}$, with darker dots representing curricula with higher $\mathcal{F}$. See **Supp.** for similar scatter plots for CL algorithms EWC [26] and LwF [31].

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. $\mathcal{F}$ rewards even contributions from α and β, while punishing extremities. We formulate $\mathcal{F}$ as

$$\mathcal{F} = \frac{2}{\beta + \frac{1}{\alpha}}$$

We reported the distribution of $\mathcal{F}$ for all curricula over three datasets in **Fig 2**. We see that a curriculum with high $\mathcal{F}$ (darker dots) has a 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.

Overlap Counts

To evaluate the ranking accuracy of our CD, we introduced the metric "overlap counts." It is the number of overlapping curricula between the top-30 curricula predicted by our CD and the top-10 empirically determined curricula based on $\mathcal{F}$ after running a CL algorithm exhaustively on all curricula of a dataset. We also varied the top-x criteria predicted by our CD, where x can be 5, 10, and 20. See the **Supp.** for the overlap counts under the top-x criteria.

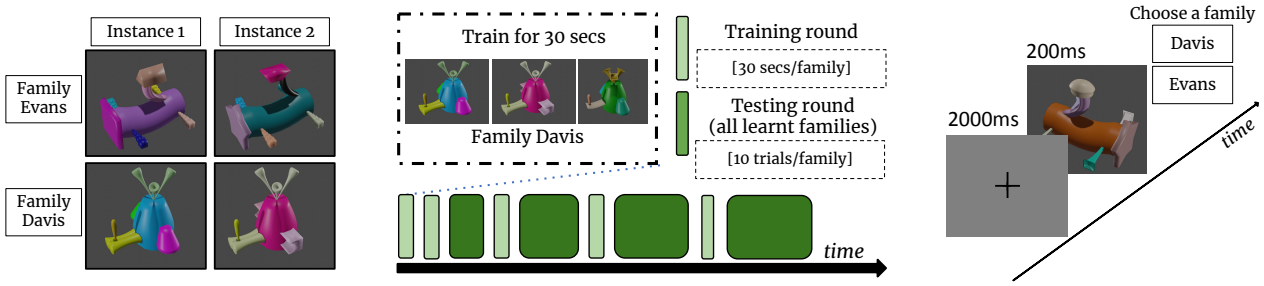
Curricula Consistency

To assess the agreement between two sets of ranked curricula, we introduced the curricula agreement measure ($\mathcal{H}$). The rank could either be determined by our CD, or empirically determined based on $\mathcal{F}$ after running a CL algorithm on all curricula of a dataset.

We sort all curricula based on their $\mathcal{F}$ scores in ascending order, divide the range of $\mathcal{F}$ into 5 uniform-sized bins, and then bin the curricula into 5 tiers with the first tier having the lowest $\mathcal{F}$ range. Since studying the characteristics of the top curricula is critically important for the benefits of human and machine learning, we focused on analyzing the curricula agreement $\mathcal{H}$ from tier 5 in the rest of the text.

Inspired by the gene sequence comparison method [8], we assign each object class to unique alphabets and then convert a curriculum to a string. As an example, 5 object classes in a dataset can be represented with letters A, B, C, D, 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 curricula set from tier 5, we can horizontally (task-wise) 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 Hamming distance to measure its consistency and denote this distance as $\mathcal{H}$. Ideally, if the two ranked curricula are in the same order, the metric $\mathcal{H}$ equals 0. The lower $\mathcal{H}$, the higher the consistency.

In **Fig 2**, we observed a skewed distribution of $\mathcal{F}$ where there are a few curricula with very high $\mathcal{F}$ but many curricula with equally low $\mathcal{F}$. Thus, the number of curricula within each tier could vary. For a pair of ranked curricula sets in tier 5 with different magnitudes, we choose the number of curricula in one set as reference, and compare with the other curricula set containing an equal number of curricula. We do this twice and report the mean.



(a) Example objects from novel families (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 two families each in the Novel Object Dataset (NOD, **Sec 3.4**). (b) Entire experiment schematic. Subjects went through 4 phases, each with a training and testing round. During training, subjects were given 30 seconds to watch three object instances per family rotating continuously around the azimuth, with the goal of being able to recognize the objects presented in the test round in (c). In the first training round, 2 families were introduced. One family per phase was then introduced for subsequent training rounds. During testing, subjects were tested on 10 trials from each learnt family. Trial order is randomly shuffled. (c) In each test trial, subjects were presented with a fixation cross (2000ms) followed by the stimulus shown for 200ms. After the image offset, subjects were presented with the options of all the families that they were introduced to and had to choose which family the image corresponds to.

3.4. Human Benchmark

Novel Object Dataset (NOD)

We introduced the Novel Object Dataset (NOD) containing 3D novel objects with a categorical structure. This dataset is a subset of a larger dataset known as "Fribbles" [6]. The dataset consists of 5 object families, where each differs by either the location of their sub-parts or the main body structure. In each family, there are 5 object instances that differ by the structure of the sub-parts attached. We randomly colored every object instance and their sub-parts. We also replaced the names of the novel object families with commonly used last names in the human experiments so as to aid the subjects in remembering these families. Samples of the object families and instances are illustrated in **Fig 3a**. We used Blender, an open-source 3-D computer graphics software, to load the novel object mesh, then rotated every object instance around the azimuth. For every 10 degrees around the azimuth, we elevated the object instance, and for every 10 degrees of elevation, we rendered a 1920×1080 sized image of the object in a gray background, thereby generating a dataset that contains 32,400 images.

Psychophysics Experiments

We evaluated human performance on NOD using Amazon Mechanical Turk (MTurk). All the psychophysics experiments were conducted with the subjects' informed consent and according to the protocols approved by our Institutional Review Board. Each participant was compensated. Each experiment took approximately 20 minutes.

We divided the experiment into 4 tasks, such that the first task had 2 object families and each subsequent task had 1 object family each; this makes a total of 60 possible curricula. Each subject is randomly assigned a curriculum. We recruited 242 subjects, yielding 34,848 trials. The schematic of the experiment is illustrated in **Fig 3b**. During the training rounds, the subjects were presented with 3 object instances per family (each family comprises 5 instances) that were rotating continuously around the azimuth. During the testing rounds, the subjects were presented a 640×480 sized GIF for each trial from the remaining 2 object instances per family according to the schematic in **Fig 3c**. Train and test instances differ.

To ensure that the subjects always paid attention to the experiments, and to control the data quality, we took the following precautions. (a) Subjects had to click on randomly presented triangles during the training rounds and their reaction times were recorded for attention checks. (b) Subjects had to recognize simple geometric shapes, such as 3D cubes, in randomly dispersed dummy trials during the testing rounds. (c) In each test trial, the "submit" button was disabled before the stimulus was shown to ensure that subjects had ample time to be exposed to the stimulus. We used the reaction time in (a) and the recognition accuracy on geometric shapes in (b) as the participant selection criteria, resulting in 2-4 participants per curriculum. See **Supp.** for details on data quality controls.

Computational Experiments

We evaluated a CL algorithm on NOD with the same experimental paradigm as the psychophysics experiments, i.e. 4 tasks, where the first task had 2 object families and each subsequent task had 1 object family each, making a total of 60 possible curricula.

Algorithm 1: Python-style pseudocode for CD

```
# N, N: number of classes, number of
 incremental steps
# M (N × N): M[i][j] is distance between
 class i and j; M[i][i] = None
# Var():  function to compute variance
# initialize curriculum (N × 1)
C = []
# initialize ranking score
s = 0
# choose C for the 1st incremental step
C[1] = argmin(Var(M, dim=1))
s = -min(Var(M, dim=1))
# choose C for the other incremental steps
for t in ceil(N / 2):
    # select object class with least
     distance for incremental step, T-t+1
    C[T-t+1] = argmin(M[i])
    s -= M[i][C[T-t+1]]
    # select object class with most distance
     for incremental step, t+1
    C[t+1] = argmax(M[i])
    s += M[i][C[t+1]]
# Code can be adapted to just score a given
 curriculum
```

4. Curriculum Designer

We proposed a proof-of-concept model, Curriculum Designer (CD), for the class-incremental stream learning setting (**Alg 1**). Given a curriculum, our CD assigns a ranking score based on inter-class feature similarity. In a dataset, our CD scores all curricula, resulting in a ranked set of curricula. Given the high agreement on the ranked curricula of different $\mathcal{A}$s (see the results in **Sec 5.4**), our CD is independent of the learning mechanisms of any $\mathcal{A}$. The objective of our CD is to propose a universal curriculum that benefits any given $\mathcal{A}$.

4.1. Feature Distance Confusion Matrix

We introduced an inter-class distance confusion matrix M of size $N \times N$, where any tuple (i, j) represents a distance measure between two class prototypes, c_i and c_j. Specifically, we used a teacher network to extract features on all images from the same class, and then determined the prototype by taking the average of all the features from the same class, and then calculated the feature distance between any given pair of class prototypes.

We use a 2D-CNN teacher network, SqueezeNet [24] up to layer 12 as the feature extractor. Drawing on the analogy that a human teacher has full knowledge of the subject they teach, the teacher network is pretrained on ImageNet [13]. To be consistent with $\mathcal{A}$ (**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.

To calculate the feature distance for a given pair of class prototypes, we used cosine distance. We conducted ablation experiments to study other choices of distance metrics (**Sec 5.3**). In practice, for a dataset, exhaustively going through all K images of a class to extract features and then computing the class prototype is computationally costly. Thus, we randomly sample 500 images per class to compute the prototypes.

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.

Drawing on the idea of metric learning [10] and the nearest neighbors algorithm in the meta-learning literature, 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 distance variance implies shorter distances between the selected class prototype and the other classes' prototypes. Starting to learn from the class comprising features shared with most other classes enables fast 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 $1 - Var(\{M_{(i,j)}\}_{j=1}^{N})$, where c_i is the first selected class of a given curriculum 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 [2, 5, 9, 33, 35, 38, 51] 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 $1 - M(c_{t=1}, c_{t=N})$.

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 [7]. 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 $v_{t=2} = M(c_{t=1}, c_{t=2})$

We complete the ranking process of a given curriculum by iteratively performing the advantage evaluation over all

subsequent incremental steps until we have examined all the classes. The final accumulative ranking score s is the sum of advantages over all incremental steps: $\sum_{t=1}^{N} v_t$.

For every curriculum from a dataset, we can compute its corresponding ranking score s. 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 the ablation results (**Sec 5.3**) to assess the effects of design variations in our CD.

5. Results

5.1. Curriculum Strongly Impacts Performance

Fig 2 highlights the effect of curricula on the vanilla CL algorithm (**Sec 3.2**) over all three datasets (**Sec 3.1**). We observed a large variance in α, it fell between 19% to 26% depending on the curriculum. This implies that curricula strongly influences the overall performance over all tasks for the vanilla CL algorithm (**Sec 3.3**). With an optimal curriculum, the average accuracy for the vanilla CL algorithm boost up by 7% over all tasks.

There also exists large variance in β for different curricula. β reflects the forgetfullness of the first task over tasks (**Sec 3.3**). The variance in β indicates that the curricula play a significant role in preventing the vanilla CL algorithm from forgetting. For example, with an optimal curriculum, the accuracy on the images of the first task remains consistent over tasks, leading to relatively smaller difference in accuracy drops; thus, smaller β.

We introduced the learning effectiveness score $\mathcal{F}$, which incorporates both α and β as inputs (**Sec 3.3**). Darker dots indicate higher $\mathcal{F}$. These dots are generally indicative of larger α and smaller β. This implies that our introduced $\mathcal{F}$ is an effective measure of the learning effectiveness of a CL algorithm. We present the results of two other CL algorithms EWC [26] and LwF [31] in **Supp**. The discussion here is also applicable for these two algorithms.

5.2. Our CD Predicts Optimal Curricula

To evaluate whether our predicted optimal curricula is indeed effective for CL algorithms, we introduced the metric "overlap counts". It refers to the number of overlapping curricula between our predicted curricula and the empirically determined optimal curricula (**Sec 3.3**). We also introduced a random curriculum designer (chance model, **Sec 3.2**) for comparison to our CD. We reported the results in **Table 1**. On average, across three CL algorithms and three datasets, there were 3.45 overlaps (**Table 1**) using our CD, which is higher than the chance model (2.6). This implies that our CD is capable of predicting optimal curricula.

overlap Counts		Vanilla	EWC	LwF
FashionMNIST	our CD	**3**	2	**3**
	chance	2.6	**2.6**	2.6
MNIST	our CD	**4**	**5**	**4**
	chance	2.6	2.6	2.6
CIFAR10	our CD	1	**4**	**5**
	chance	**2.6**	2.6	2.6

Table 1. **Our Curriculum Designer (CD) predicts optimal curricula.** Overlap counts (**Sec 5.2**) between the top-30 curricula by our CD and the empirically determined top-10 curricula across three CL algorithms (**Sec 3.2**) over all three datasets (**Sec 3.1**). The best is in bold.

We also analyzed the overlap counts across datasets. With FashionMNIST, MNIST and CIFAR10, our CD scores an average of 2.7, 4.3 and 3.3 overlaps respectively across three CL algorithms. This observation reveals that our CD outperforms chance on each dataset. We also provided the visualization of the top-5 curricula predicted by our CD and determined by CL algorithms for all the datasets (see **Supp.**).

Although our proof-of-concept CD is effective in a majority of the cases, we noticed a few failure cases as well. For example, our CD had a lower number of overlaps than the chance model for the vanilla CL algorithm on CIFAR10. This implies that there is still a large gap in predicting optimal curricula. More complex curriculum designs are necessary to cater for different CL algorithms.

5.3. Analysis on Design Choices of Our CD

To assess the individual design choices in our CD, we introduced variations of our CD. We discuss below the implications on the Vanilla settings for MNIST First, we conducted experiments using layer 11 and layer 6 of the feature extractor to compute the distance confusion matrix M. We found that the overlap count drops by 2 and 2 respectively compared to our CD (**Fig 4**). This implies that the layer choice is important and the higher layers of the network produce more class-representative features that are useful for curricula ranking.

Next, as opposed to the cosine distance metric used in our CD, we changed the distance metric to Euclidean and Optimal Transport Dataset Distance (OTDD) [4]. We noted a drop of 1 and 3 in overlap counts with Euclidean and OTDD respectively (**Fig 4**). This implies that the choice of measure for the inter-class distance is important. Compared to Euclidean and OTDD, cosine distance is a more effective measure of inter-class difficulty. See **Supp.** for more detailed discussions on the effect of design choices in our CD across other CL algorithms and datasets.

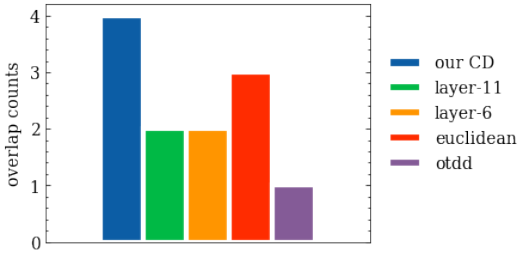


Figure 4. **Ablation results on our CD.** We show the overlaps count (**Sec 5.3**) between the top-30 curricula predicted by a curriculum designer and the top-10 empirical curricula determined by the Vanilla setting (**Sec 3.2**) for MNIST (**Sec. 3.1**). Curriculum designers from the left to the right along the x-axis refer to our CD, our CD with the distance confusion matrix (**Sec 4.1**) computed based on the features extracted at layer 11 and 6 of the teacher network, and our CD with the distance confusion matrices computed with different distance metrics: Euclidean and Optimal Transport Dataset Distance (OTDD) [4].

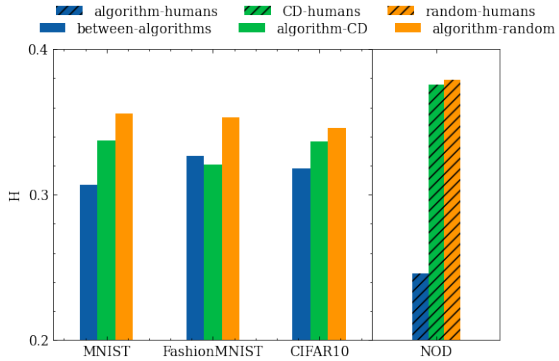


Figure 5. **There exists high agreement on optimal curricula determined by between-algorithms, algorithm-CD, algorithm-humans, and CD-humans.** Left panel: curricula agreement $\mathcal{H}$ (**Sec 3.3**) is reported between pairs of CL algorithms $\mathcal{A}$s (between-algorithm, blue), between $\mathcal{A}$s and our CD (algorithm-CD, green), between $\mathcal{A}$ and the random designer (algorithm-random, orange) across all three datasets MNIST, FashionMNIST, and CIFAR10 (**Sec 5.5**). Right panel: Agreement $\mathcal{H}$ 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**).

5.4. Analysis on Curricula Agreement

The left panel in **Fig 5** presents the agreement $\mathcal{H}$ between optimal curricula determined by CL algorithms $\mathcal{A}$, our CD, and the random curriculum designer on MNIST, FashionMNIST, CIFAR10 (**Sec 3.1**). A decrease in $\mathcal{H}$ indicates an increase in the agreement (**Sec 3.3**).

First, we tested on a simple dataset, MNIST and observed an increase of 0.05 in the agreement from algorithm-random $\mathcal{A}$s to between-algorithms $\mathcal{A}$s. We further tested on a more complex dataset, FashionMNIST and noted an increase of 0.03 in the agreement from algorithm-random to between-algorithms $\mathcal{A}$s. To solidify our findings even further, we tested on the dataset of natural images, CIFAR10 and observed an increase of 0.03 in the agreement from algorithm-random to between-algorithms. Based on these findings, we concluded that CL algorithms $\mathcal{A}$s share a comparable set of top-ranked curricula across three datasets with varying levels of difficulty.

Second, we assessed the curricula agreement between our CD and CL algorithms $\mathcal{A}$s. Consistent across all the three datasets, we observed an increase of $0.02, 0.03, 0.01$ in the agreement from algorithm-random to algorithm-CD on MNIST, FashionMNIST, and CIFAR10 respectively. It implies that our CD can predict optimal curricula well aligned with the empirically determined optimal curricula from CL algorithms $\mathcal{A}$s. However, $\mathcal{H}$ in between-algorithms is still higher than algorithm-CD. This implies that the optimal curricula between CL algorithms is slightly more consistent than the optimal curricula ranked by our CD. See **Supp.** for a quantitative assessment of hyperparameter ablations on various CL algorithms.

5.5. Our Model Benefits Human Learning

The right panel in **Fig 5** presents the agreement on optimal curricula determined by CL algorithms $\mathcal{A}$, our CD, humans, and the random curriculum designer on the Novel Object Dataset (NOD) (**Sec 3.4**).

We discerned that there is an increase of 0.13 in the curricula agreement from random-human to algorithm-human (**Fig 5**). This illustrates the existence of a notable agreement between optimal curricula for humans and CL algorithms. We further noted that there is an increase of 0.001 in the curricula agreement from random-human to CD-human. These observations served as a proof of concept that our CD can predict optimal curricula that can help supplement human learning.

6. Discussion

Our education system encompasses a sequence of curricula. Designing an effective curriculum to teach humans and machines is imperative, with the aim of maximizing knowledge transfer over tasks while, minimizing catastrophic forgetting on the previous tasks. In practice, there are numerous curriculum design considerations that need to be deliberated upon, such as example orders within a class, supercategory and subcategory learning, the learning characteristics of every student, and multi-modal learning. Here, we did not exhaustively study all possible combinations; instead, our study established a methodology for the community to evaluate and benchmark our education systems for both humans and AI. Closely resembling human learning, we formulated the study of curriculum learning in the class-incremental stream learning setting. We surveyed

the 5-class and 10-class incremental settings on 3 CL algorithms over 3 datasets. Moreover, we introduced a proof-of-concept curriculum designer, capable of designing and ranking curricula. To benchmark the curricula efficacy on humans, we also contributed a new novel-object dataset and conducted human behavioral experiments. The insights obtained from our work should pave a way for the community to benchmark AI-assistive education systems for humans and AI.

List of Supplementary Sections

List of Supplementary Figures

List of Tables

S1. Human Benchmark

Psychophysics Experiments

In **Fig. S1**, we show a screenshot of the Amazon Mechanical Turk (MTurk) interface during the training round of human psychophysics experiments. We also show a screenshot of the MTurk interface during the testing round in **Fig. S2**.

To collect high-quality data, we collected responses from *master* workers with at least 1,000 approved hits and a 95% approval rate. We collected responses from 242 participants in total. We filter out participants based on data quality controls like reaction time in attention checks, and recognition accuracy on simple geometric shapes. Finally, we end up with 169 participants with 2-4 participants per curriculum. In **Fig. S3**, we show the distribution of reaction time for all participants on the attention checks. Moreover, we show the accuracy of participants on attention check trials where the participants were required to correctly recognize simple geometric shapes in **Fig. S4**.

We show the average accuracy of humans over all tasks and a α vs β (**Sec 3.3**) distribution for NOD in **Fig S5**. Moreover, we show the effect of curricula for NOD using the vanilla continual learning algorithm, EWC, and LwF (**Sec 3.2**) in **Fig S6**, **Fig S7** and **Fig S8** respectively.

S2. Analysis on training regimes

Drawing on the analogy in pedagogy where the same curricula might effect different students, in computational settings, we verify whether the training regimes for all continual learning algorithms would have an effect on curricula learning. We vary the experimental settings in number of epochs, learning rates, and parameter initialization.

In each experimental setting, we report the $\mathcal{H}$ between all pairs of continual learning algorithms $\mathcal{A}$s (between-algorithm) and between $\mathcal{A}$s and the random curriculum designer (algorithm-random) on FashionMNIST (**Fig S9**). For experiment controls, we only vary one experimental factor at a time.

First, we performed the analysis of curricula agreement after we vary the number of training epochs over 1, 10, and 20 per incremental step for all $\mathcal{A}$s. We observed an average increase of 0.16 in the agreement from algorithm-random to algorithm-algorithm over all three algorithms (**Fig S9, (a)**). It suggests that the curricula agreement persists among continual learning algorithms trained with either single or multiple epochs. The effect of class orders in curricula learning is largely independent of number of epochs.

Next, we vary the learning rates of all continual learning algorithms over $0.5e^{-3}$, $1e^{-3}$, $2e^{-3}$ respectively. We observed an average increase of 0.02 in the agreement from algorithm-random to algorithm-algorithm over all three algorithms (**Fig S9, (b)**). However, when learning rate is too large, we noticed that there exists a large inconsistency (almost comparable to algorithm-random) among empirically determined curricula by each continual learning algorithm. This suggests that large learning rates leads to inconsistent curricula effect among continual learning algorithms.

Lastly, we changed the initialization of the trainable parameters of linear classifiers for all continual learning algorithms over Gaussian, Uniform, and Xavier [17]. We observed an average increase of 0.03 in the curricula agreement from algorithm-random to algorithm-algorithm respectively (**Fig S9, (c)**). However, the agreement becomes much lower in Xavier than the other two initialization. This observation indicates that curricula agreement is robust to varying initialization but the degree of agreement depends on choice of parameter initialization.

S3. Our CD Predicts Optimal Curricula

We analyzed the overlap counts (**Sec 3.3**) using our Curriculum Designer (CD, **Sec 4**) for the top-x criterion, where x can be (5, 10, 20, 30). We conducted our analyses across MNIST, FashionMNIST and CIFAR10 (**Sec 3.1**).

We evaluated the overlap counts on the top-30 criterion for paradigm-I (5 classes, **Sec 3.1**) in **Sec 5.2**. Further, we extended this to paradigms-II (10 classes, **Sec 3.1**). On average, across three datasets and three continual learning algorithms (**Sec 3.2**), there were 2.7 overlaps using our CD for paradigm-II which is higher than the chance model (2.6). Our CD averaged 1.66, 0.33 and 0 overlaps for paradigm-II on FashionMNIST, MNIST, and CIFAR10 respectively.

We evaluated the overlap counts on the top-5 criterion (**Table S1**, **Table S4**). On average, across three datasets and three continual learning algorithms (**Sec 3.2**), there were 2.45 and 0.67 overlaps using our CD for paradigm-I (5 classes, **Sec 3.1**) and paradigm-II (10 classes, **Sec 3.1**) respectively; they were both higher than the chance model (0.2). Individually, our CD averaged 2.33, 3 and 2 overlaps for paradigm-I on FashionMNIST, MNIST, and CIFAR10. Additionally, our CD averaged 1.66, 0.33 and 0 overlaps for paradigm-II on FashionMNIST, MNIST, and CIFAR10 respectively.

We evaluated the overlap counts on the top-10 criterion (**Table S2**, **Table S5**). On average, across three datasets and three continual learning algorithms (**Sec 3.2**), there were 3.44 and 1.44 overlaps using our CD for paradigm-I (5 classes, **Sec 3.1**) and paradigm-II (10 classes, **Sec 3.1**) respectively; they were both higher than the chance model (0.6). Individually, our

CD averaged 2.66, 4.33 and 3.33 overlaps for paradigm-I on FashionMNIST, MNIST, and CIFAR10. Additionally, our CD averaged 2.33, 1.33 and 0.66 overlaps for paradigm-II on FashionMNIST, MNIST, and CIFAR10 respectively.

We evaluated the overlap counts on the top-20 criterion (**Table S3**, **Table S6**). On average, across three datasets and three continual learning algorithms (**Sec 3.2**), there were 3.44 and 2.11 overlaps using our CD for paradigm-I (5 classes, **Sec 3.1**) and paradigm-II (10 classes, **Sec 3.1**) respectively; they were both higher than the chance model (1.3). Individually, our CD averaged 2.66, 4.33 and 3.33 overlaps for paradigm-I on FashionMNIST, MNIST, and CIFAR10. Additionally, our CD averaged 3.33, 2.33 and 0.66 overlaps for paradigm-II on FashionMNIST, MNIST, and CIFAR10 respectively.

We further visualised the top-5 curricula across three datasets (**Sec 3.1**) and three continual learning algorithms (**Sec 3.2**) for paradigm-I (5 classes, **Sec 3.1**). **Fig S10**, **Fig S11** and **Fig S12** highlight the top-5 curricula for FashionMNIST using the vanilla continual learning algorithm, EWC and LwF (**Sec 3.2**) respectively. **Fig S13**, **Fig S14** and **Fig S15** highlight the top-5 curricula for MNIST using the vanilla continual learning algorithm, EWC and LwF (**Sec 3.2**) respectively. **Fig S16**, **Fig S17** and **Fig S18** highlight the top-5 curricula for CIFAR10 using the vanilla continual learning algorithm, EWC and LwF (**Sec 3.2**) respectively.

We further visualised the top-5 curricula across three datasets (**Sec 3.1**) and three continual learning algorithms (**Sec 3.2**) for paradigm-II (10 classes, **Sec 3.1**). **Fig S19**, **Fig S20** and **Fig S21** highlight the top-5 curricula for FashionMNIST using the vanilla continual learning algorithm, EWC and LwF (**Sec 3.2**) respectively. **Fig S22**, **Fig S23** and **Fig S24** highlight the top-5 curricula for MNIST using the vanilla continual learning algorithm, EWC and LwF (**Sec 3.2**) respectively. **Fig S25**, **Fig S26** and **Fig S27** highlight the top-5 curricula for CIFAR10 using the vanilla continual learning algorithm, EWC and LwF (**Sec 3.2**) respectively.

S4. Curriculum Strongly Impacts Performance

Sec 5.1 highlighted the effect of curricula on the vanilla continual setting algorithm (**Sec 3.2**) over all three datasets (**Sec 3.1**). To further these insights, we also visualized the average accuracy over all tasks and a α vs β (**Sec 3.3**) distribution for FashionMNIST, MNIST and CIFAR10 using the vanilla continual learning algorithm (**Sec 3.2**) in **Fig S28**, **Fig S29** and **Fig S30** respectively.

Here we discuss the impact of curricula over all three datasets and all three continual learning algorithms for paradigm-I (5 classes, **Sec 3.1**) and paradigm-II (10 classes, **Sec 3.1**). α (**Sec 3.3**) highlights the average accuracy over all tasks. β (**Sec 3.3**) reflects the forgetfullness of the first task over tasks. $\mathcal{F}$ (**Sec 3.3**), which incorporates both α and β as inputs, reflects the overall effectiveness of a curriculum.

Fig S31 and **Fig S35** highlight the effect of curricula on EWC [26] and LwF [31] respectively (**Sec 3.2**) over all three datasets (**Sec 3.1**) for paradigm-I (5 classes, **Sec 3.1**). We observed a large variance in α (**Sec 3.3**), it fell between 20% to 24% depending on the curriculum. With an optimal curriculum, the average accuracy for EWC or LwF boosts up by 4% over all tasks. There also exists large variance in β for different curricula, which indicates that the curricula play a significant role in preventing EWC and LwF from forgetting. To further these insights, we also visualized the average accuracy over all tasks and a α vs β (**Sec 3.3**) distribution for FashionMNIST, MNIST and CIFAR10 using EWC in **Fig S32**, **Fig S33** and **Fig S34** respectively. We also visualized the average accuracy over all tasks and a α vs β (**Sec 3.3**) distribution for FashionMNIST, MNIST and CIFAR10 using LwF in **Fig S36**, **Fig S37** and **Fig S38** respectively.

Fig S39, **Fig S43** and **Fig S47** highlight the effect of curricula on the vanilla continual learning algorithm, EWC [26] and LwF [31] respectively (**Sec 3.2**) over all three datasets (**Sec 3.1**) for paradigm-II (10 classes, **Sec 3.1**). We observed a large variance in α (**Sec 3.3**), it fell between 10% to 24% depending on the curriculum. With an optimal curriculum, the average accuracy for the vanilla continual learning algorithm, EWC or LwF boosts up by 14% over all tasks. There also exists large variance in β for different curricula, which indicates that the curricula play a significant role in preventing the vanilla continual learning algorithm, EWC and LwF from forgetting. To further these insights, we also visualized the average accuracy over all tasks and a α vs β (**Sec 3.3**) distribution for FashionMNIST, MNIST and CIFAR10 using the vanilla continual learning algorithm in **Fig S40**, **Fig S41** and **Fig S42** respectively. We also visualized the average accuracy over all tasks and a α vs β (**Sec 3.3**) distribution for FashionMNIST, MNIST and CIFAR10 using EWC in **Fig S44**, **Fig S45** and **Fig S46** respectively. We also visualized the average accuracy over all tasks and a α vs β (**Sec 3.3**) distribution for FashionMNIST, MNIST and CIFAR10 using LwF in **Fig S48**, **Fig S49** and **Fig S50** respectively.

S5. Analysis on Curricula Agreement

Fig S51 presents the agreement $\mathcal{H}$ between optimal curricula determined by continual learning algorithms $\mathcal{A}$, our CD, and the random curriculum designer on MNIST, FashionMNIST, CIFAR10 (**Sec 3.1**) for paradigm-II (10 classes, **Sec 3.1**).

A decrease in $\mathcal{H}$ indicates an increase in the agreement (**Sec 3.3**).

First, we tested on a simple dataset, MNIST and observed an increase of 0.01 in the agreement from algorithm-random $\mathcal{A}$s to between-algorithms $\mathcal{A}$s. We further tested on a more complex dataset, FashionMNIST and noted an increase of 0.001 in the agreement from algorithm-random to between-algorithms $\mathcal{A}$s. To solidify our findings even further, we tested on the dataset of natural images, CIFAR10 and observed an increase of 0.014 in the agreement from algorithm-random to between-algorithms. Based on these findings, we concluded that continual learning algorithms $\mathcal{A}$s share a comparable set of top-ranked curricula across three datasets with varying levels of difficulty.

Second, we assessed the curricula agreement between our CD and continual learning algorithms $\mathcal{A}$s. We observed an increase of 0.01 in the agreement from algorithm-random to algorithm-CD only on CIFAR10. Our model fails for MNIST and FashionMNIST. This implies that there is still a large gap in predicting optimal curricula. More complex curriculum designs are necessary to cater for different continual learning algorithms.

S6. Analysis on Design Choices of our CD

To assess the individual design choices in our CD, we introduced variations of our CD. We discussed the implications on the Vanilla settings for MNIST in **Sec 5.3**. We conducted experiments using layer 11 and layer 6 of the feature extractor to compute the distance confusion matrix M. Next, as opposed to the cosine distance metric used in our CD, we changed the distance metric to Euclidean and Optimal Transport Dataset Distance (OTDD) [4]. We calculated the overlap counts across three datasets (**Sec 3.1**) and three continual learning algorithms (**Sec 3.2**) for paradigm-I (5 classes, **Sec 3.1**) over all ablations.

We noted in **Table S8** that the default setting for our CD recorded higher or equal overlap counts as opposed to its ablation in seven out of nine cases. Using the euclidean metric results in greater overlap counts for LwF with CIFAR10 and using the otdd metric results in greater overlap counts for EWC with FashionMNIST.

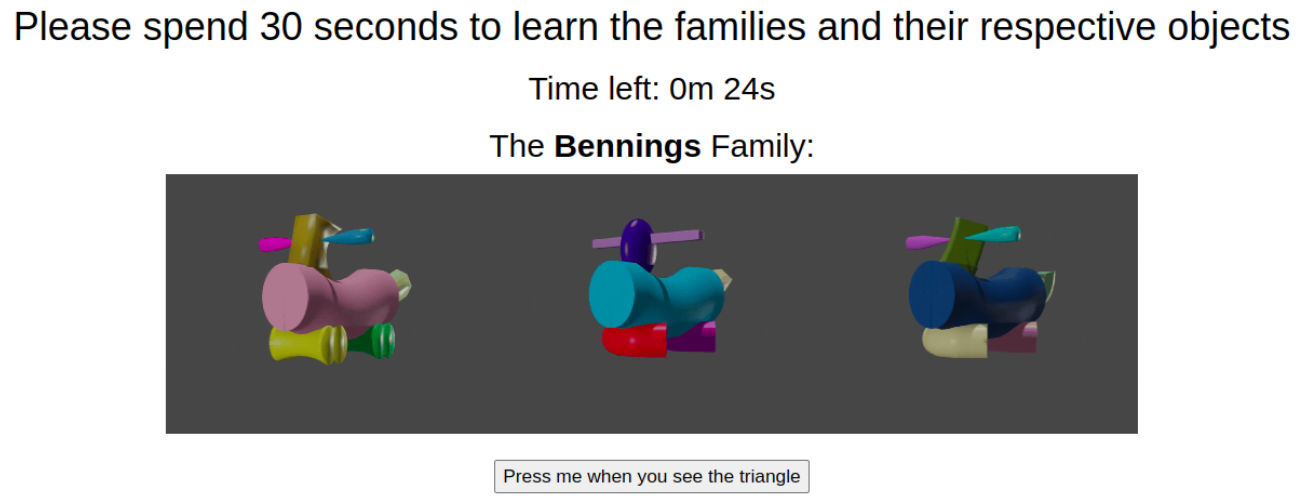


Figure S1. **A screenshot of the AMT interface during the training phase.**

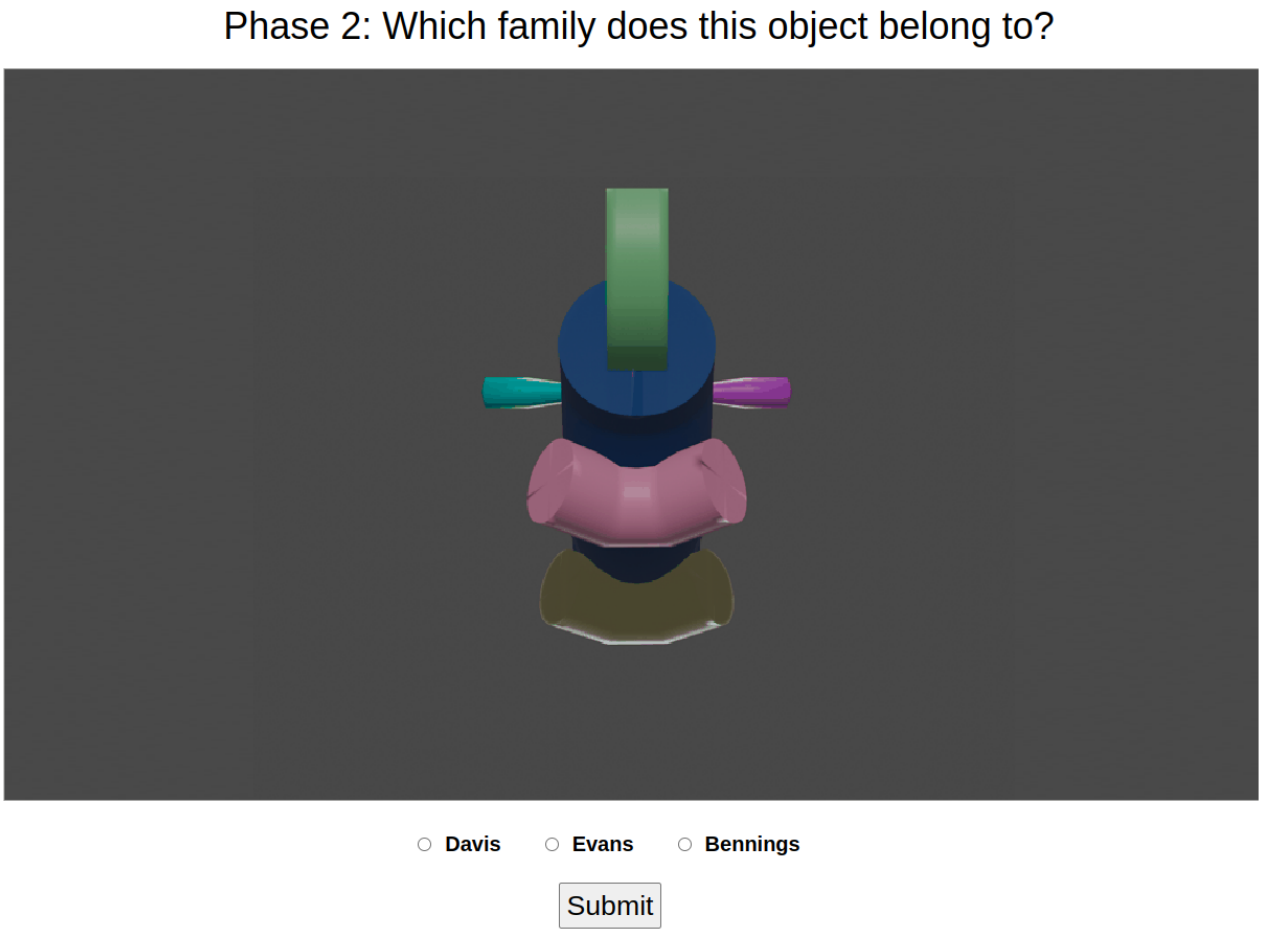


Figure S2. **A screenshot of the AMT interface during the testing round.**

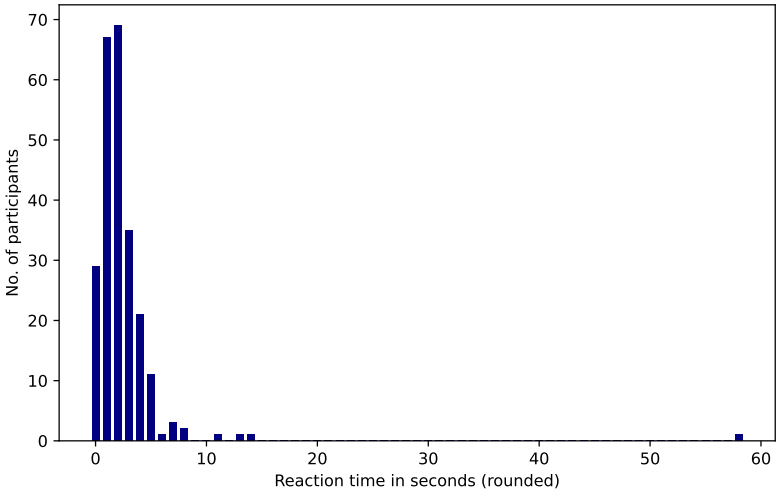


Figure S3. **Reaction time distribution for all participants in attention checks during training rounds.** Participants were required to click on randomly presented triangles during the training rounds and their reaction time was recorded. On the x-axis, we show the reaction time in seconds (rounded).

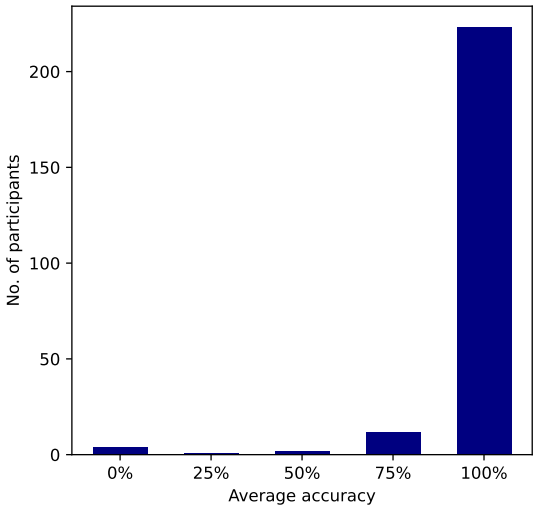


Figure S4. **Average accuracy of all participants in attention checks during testing rounds.** During the testing rounds, each participant was given four attention check trials in which they had to recognize simple geometric shapes. We only selected those participants who answered all four questions correctly for the result analysis.

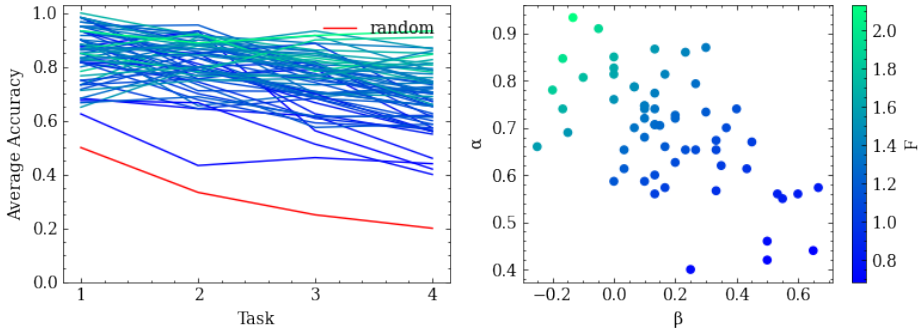


Figure S5. **Curricula impacts performance on NOD for humans (5 classes, Sec 3.4).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

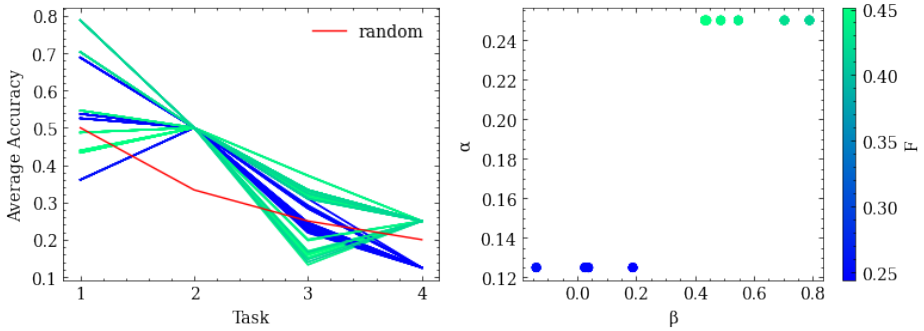


Figure S6. **Curricula impacts performance on NOD using the vanilla continual learning algorithm (Sec 3.2) for NOD paradigm (5 classes, Sec 3.4).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

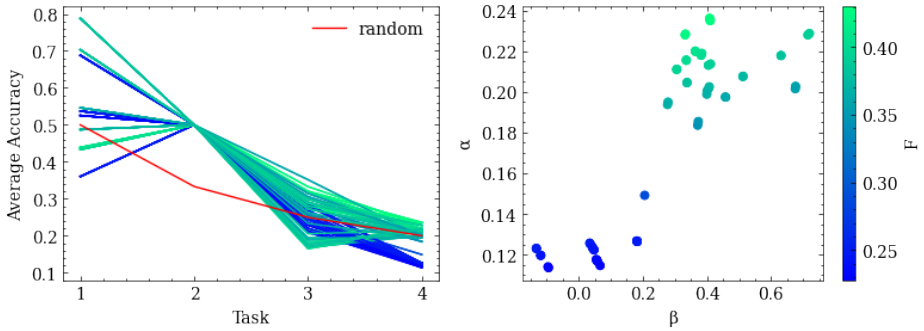


Figure S7. **Curricula impacts performance on NOD using EWC (Sec 3.2) for NOD paradigm (5 classes, Sec 3.4).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

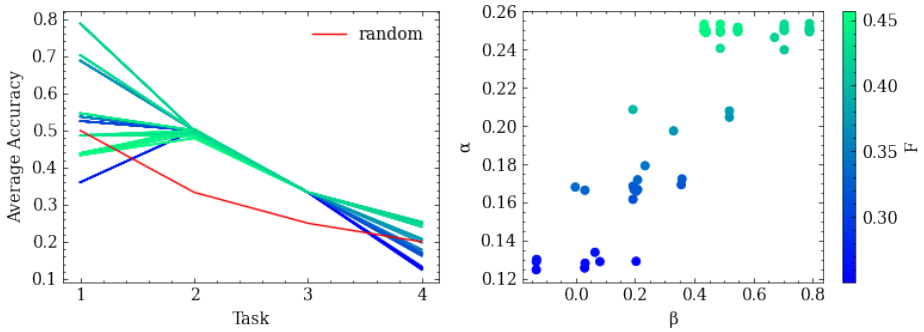


Figure S8. **Curricula impacts performance on NOD using LwF (Sec 3.2) for NOD paradigm (5 classes, Sec 3.4).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

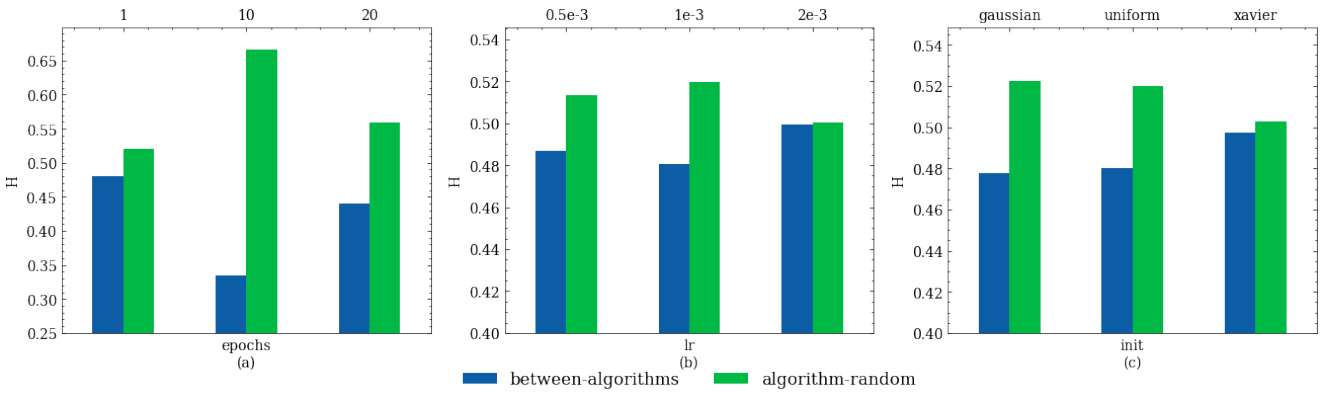


Figure S9. **The curricula agreement persists between continual learning algorithms across different experimental settings on FashionMNIST.** The agreement between two sets of ranked curricula is measured in $\mathcal{H}$ (**Sec 3.3**). The smaller the better. Within each group of bars, the agreement between pairs of ranked curricula from continual learning algorithms $\mathcal{A}$ (between-algorithm) is presented on the left (blue), and between $\mathcal{A}$ and the randomly ranked curricula (algorithm-random) is on the right (green). We vary the number of epochs (a), the learning rates (lr) (b), and the different network parameter initialisation of $\mathcal{A}$ (c). For visualization purposes, within each group of bars, we normalize the $\mathcal{H}$ over between-algorithm and algorithm-random so that the sum of these two agreements (green + blue) equals 1. The relative difference between the green and the blue matters. The normalization does not alter the conclusions.

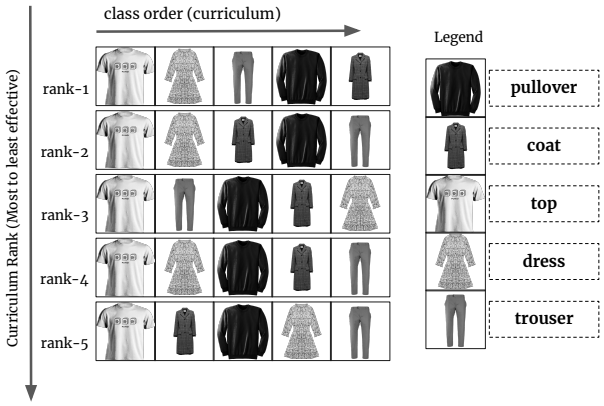


Figure S10. **Top 5 empirically determined curricula on FashionMNIST using the vanilla continual learning algorithm (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

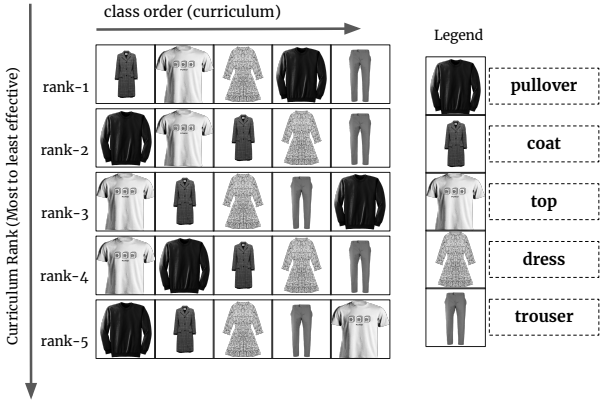


Figure S11. **Top 5 empirically determined curricula on FashionMNIST using EWC (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

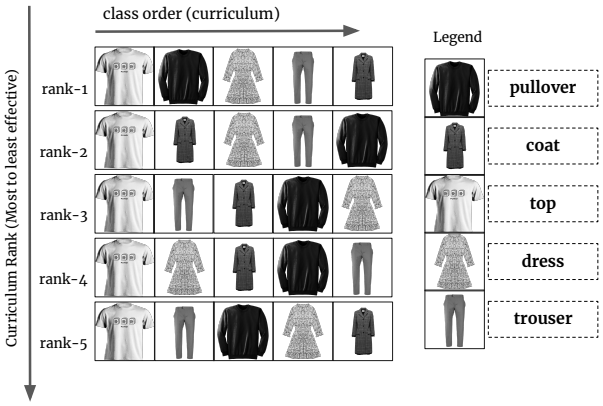


Figure S12. **Top 5 empirically determined curricula on FashionMNIST using LwF (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

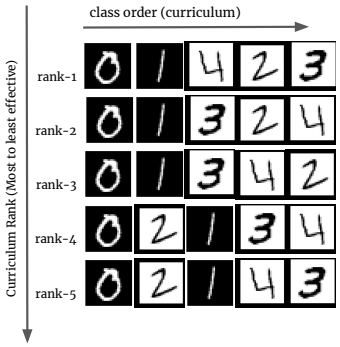


Figure S13. **Top 5 empirically determined curricula on MNIST using the vanilla continual learning algorithm (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

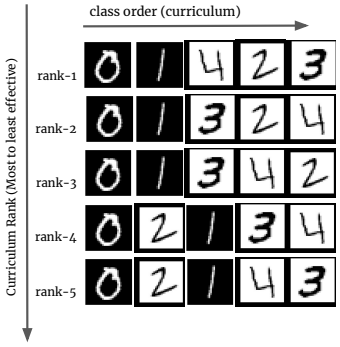


Figure S14. **Top 5 empirically determined curricula on MNIST using EWC (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

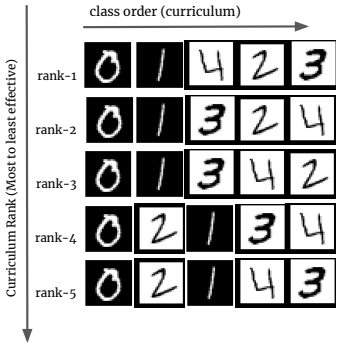


Figure S15. **Top 5 empirically determined curricula on MNIST using LwF (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

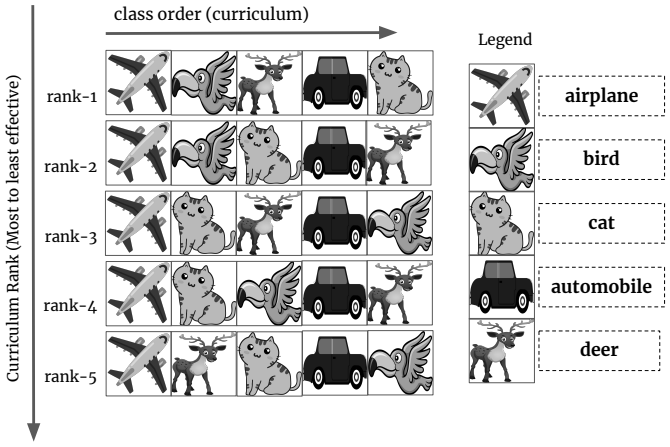


Figure S16. **Top 5 empirically determined curricula on CIFAR10 using the vanilla continual learning algorithm (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner. Examples shown are for illustrative purposes only, actual images differ.

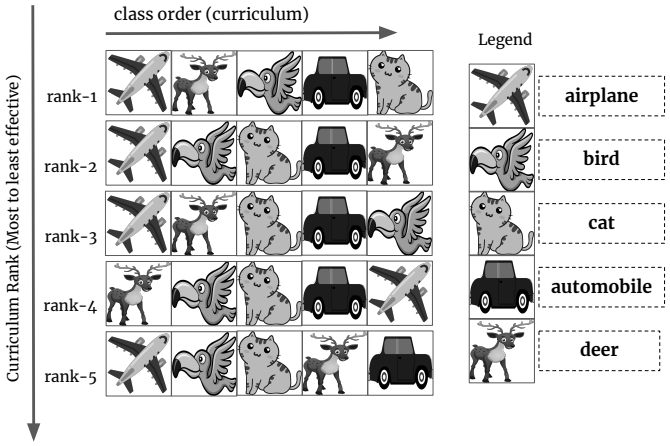


Figure S17. **Top 5 empirically determined curricula on CIFAR10 using EWC (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner. Examples shown are for illustrative purposes only, actual images differ.

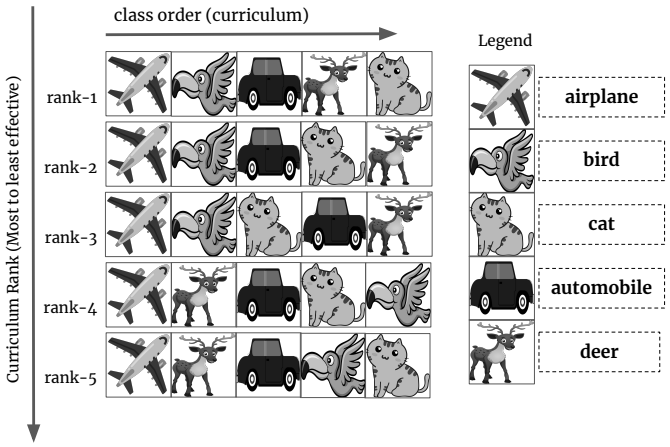


Figure S18. **Top 5 empirically determined curricula on CIFAR10 using LwF (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner. Examples shown are for illustrative purposes only, actual images differ.

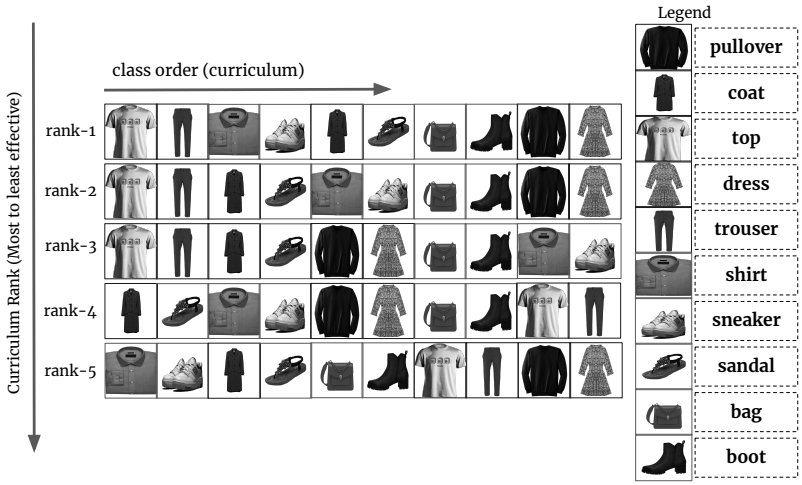


Figure S19. **Top 5 empirically determined curricula on FashionMNIST using the vanilla continual learning algorithm (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

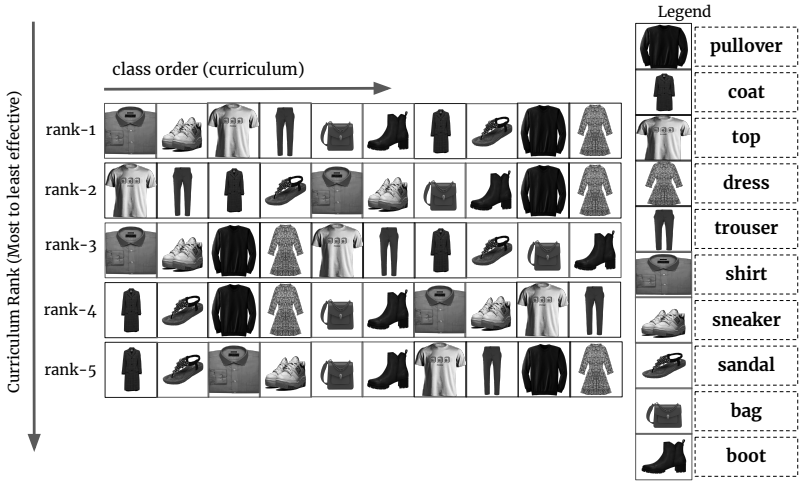


Figure S20. **Top 5 empirically determined curricula on FashionMNIST using EWC (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

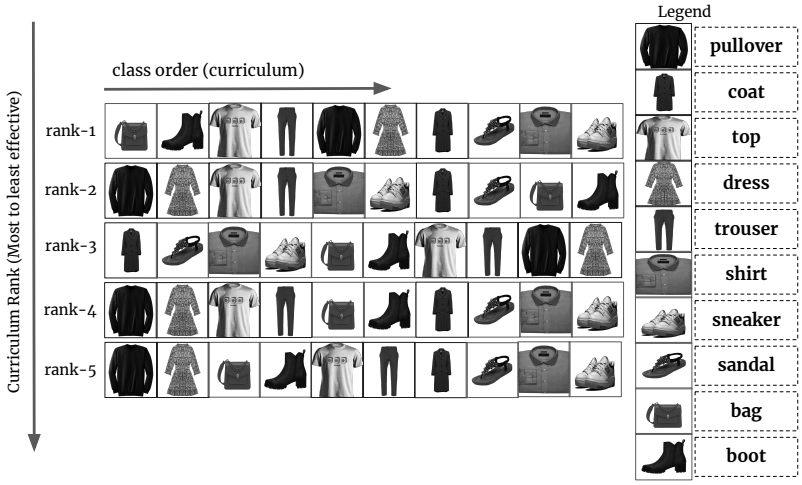


Figure S21. **Top 5 empirically determined curricula on FashionMNIST using LwF (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

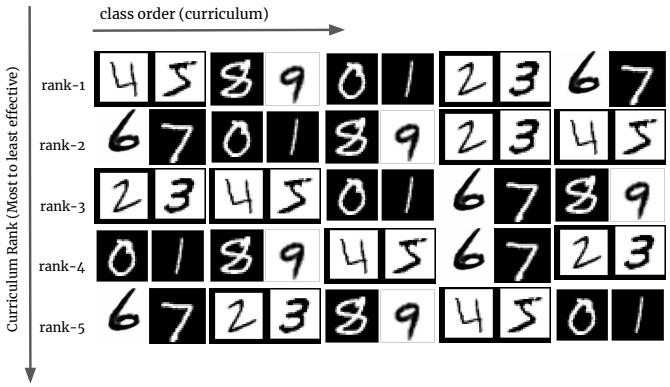


Figure S22. **Top 5 empirically determined curricula on MNIST using the vanilla continual learning algorithm (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

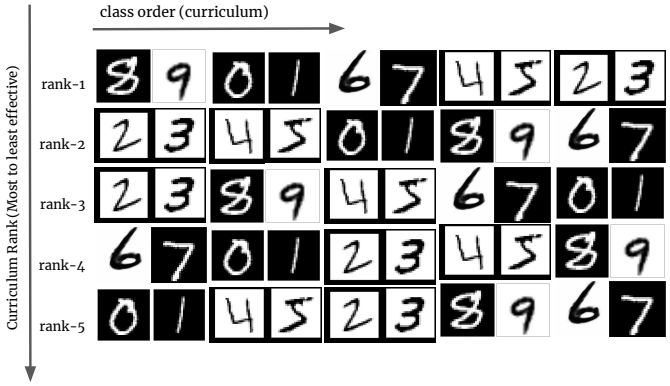


Figure S23. **Top 5 empirically determined curricula on MNIST using EWC (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

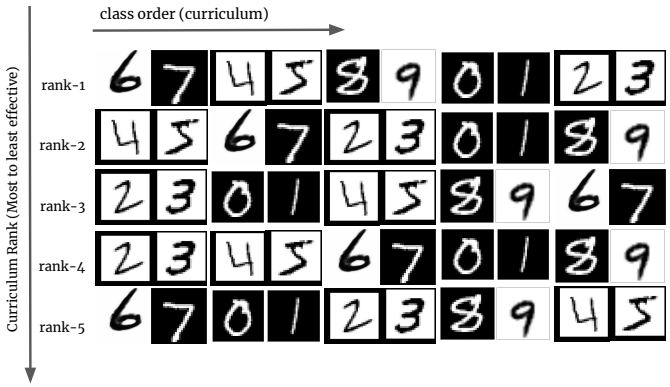


Figure S24. **Top 5 empirically determined curricula on MNIST using LwF (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner.

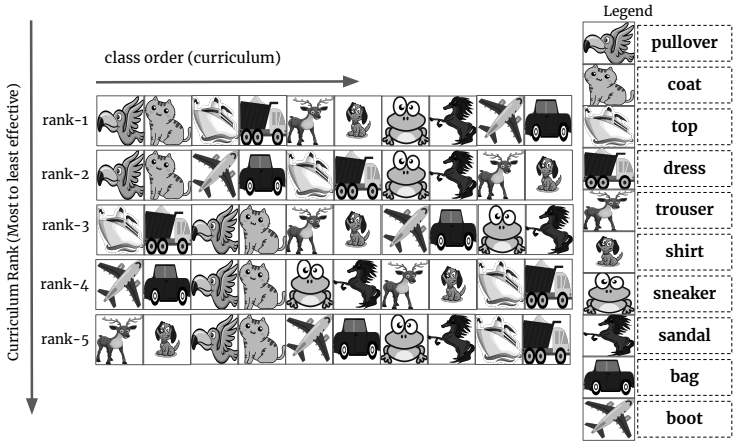


Figure S25. **Top 5 empirically determined curricula on CIFAR10 using the vanilla continual learning algorithm (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner. Examples shown are for illustrative purposes only, actual images differ.

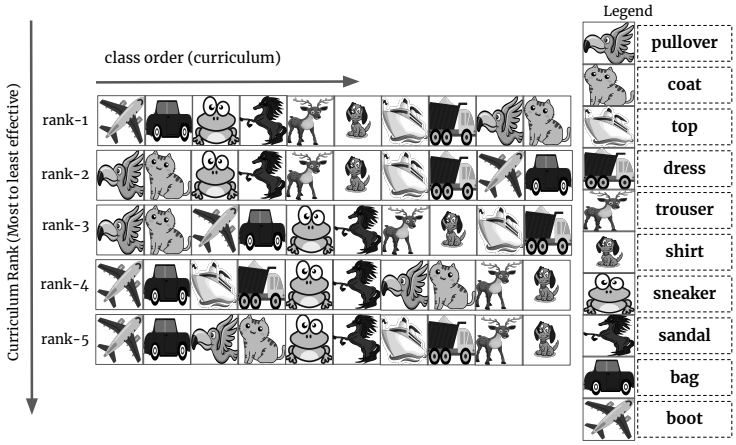


Figure S26. **Top 5 empirically determined curricula on CIFAR10 using EWC (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner. Examples shown are for illustrative purposes only, actual images differ.

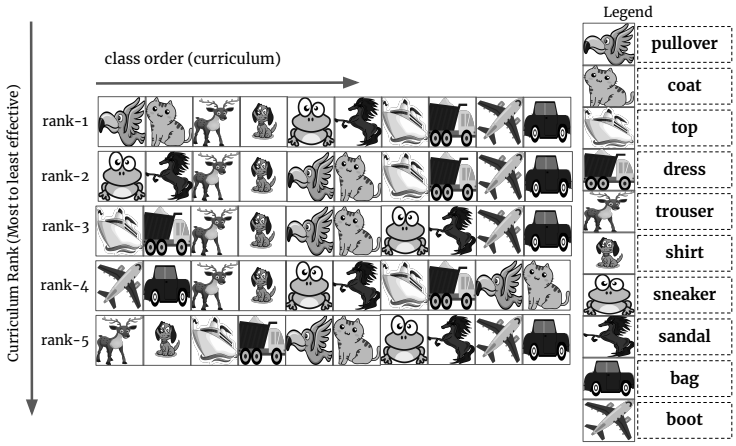


Figure S27. **Top 5 empirically determined curricula on CIFAR10 using LwF (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** Each row in the figure is a curriculum. The curricula are arranged best to worst in a top down manner. Examples shown are for illustrative purposes only, actual images differ.

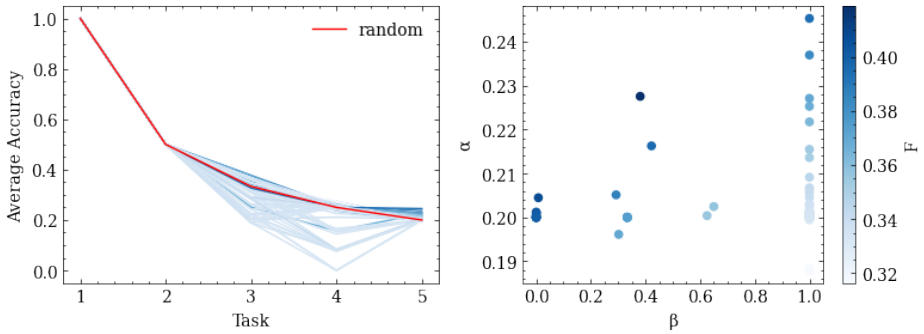


Figure S28. **Curricula impacts performance on FashionMNIST using the vanilla continual learning algorithm (Sec 3.2) for paradigm-I (5 classes, Sec 3.1)**. We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

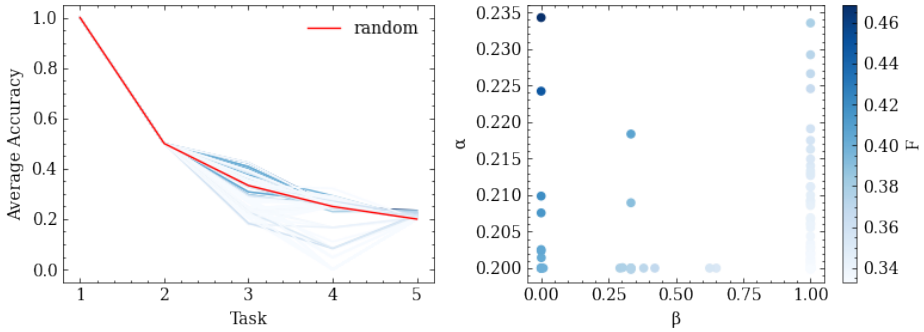


Figure S29. **Curricula impacts performance on MNIST using the vanilla continual learning algorithm (Sec 3.2) for paradigm-I (5 classes, Sec 3.1)**. We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

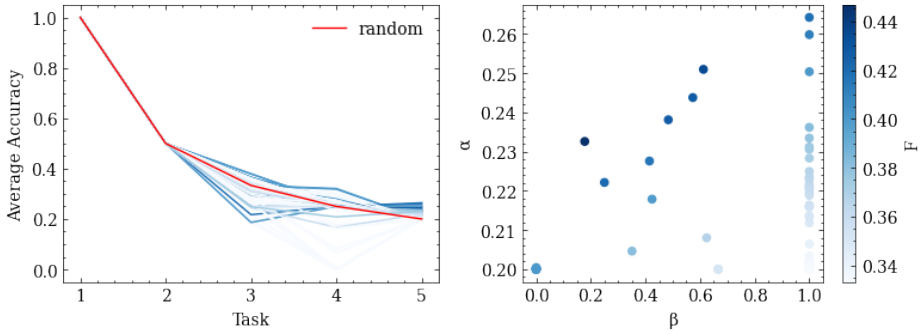


Figure S30. **Curricula impacts performance on CIFAR10 using the vanilla continual learning algorithm (Sec 3.2) for paradigm-I (5 classes, Sec 3.1)**. We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

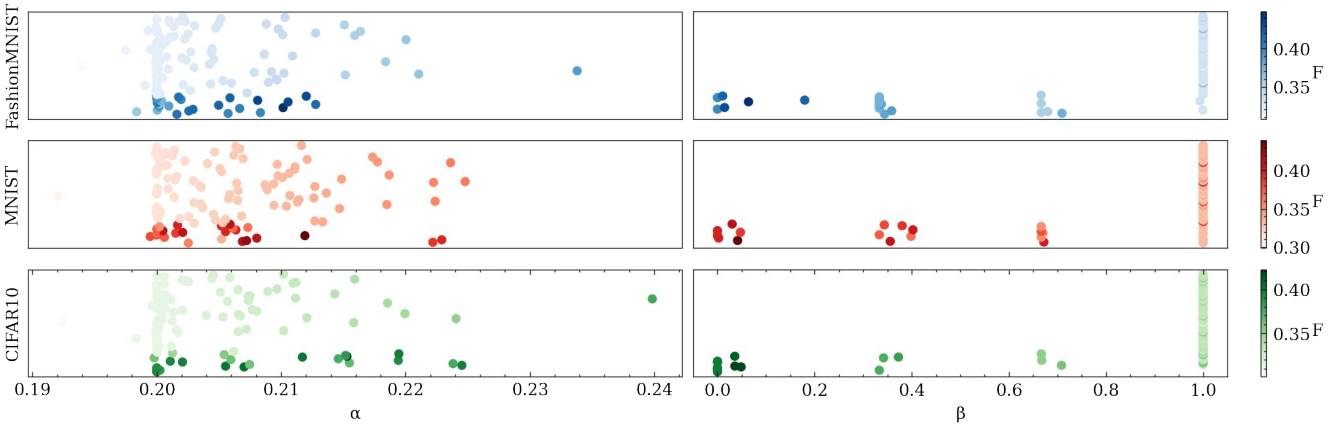


Figure S31. **Curricula influence the learning efficacy of the EWC [26] algorithm (Sec 3.2) across three datasets: MNIST, FashionMNIST, and CIFAR10 (Sec 3.1) for paradigm-I (5 classes, Sec 3.1).** We trained the EWC algorithm on all curricula from each dataset. Each dot represents one curriculum. We reported the average accuracy α over all the seen classes, highlighting how the algorithm adapts to learn new tasks **(left panel, Sec 5.1)**. The accuracy difference β reflects the forgetfulness of the algorithm across tasks **(right panel, Sec 5.1)**. We introduced $\mathcal{F}$ as the measure of the learning efficacy of a given curriculum **(Sec 3.3)** taking both α and β into account.

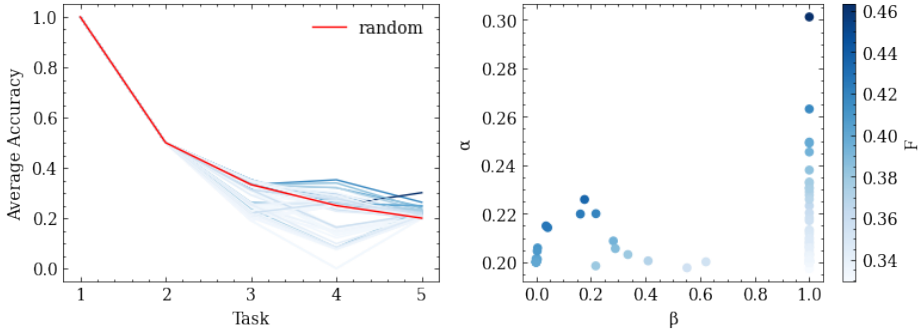


Figure S32. **Curricula impacts performance on FashionMNIST using EWC (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

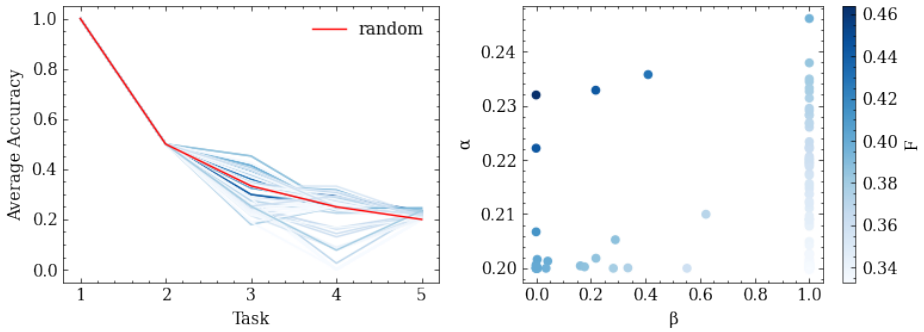


Figure S33. **Curricula impacts performance on MNIST using EWC (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

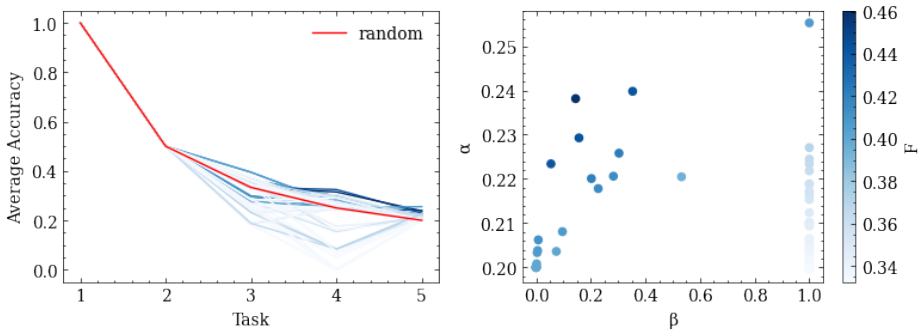


Figure S34. **Curricula impacts performance on CIFAR10 using EWC (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

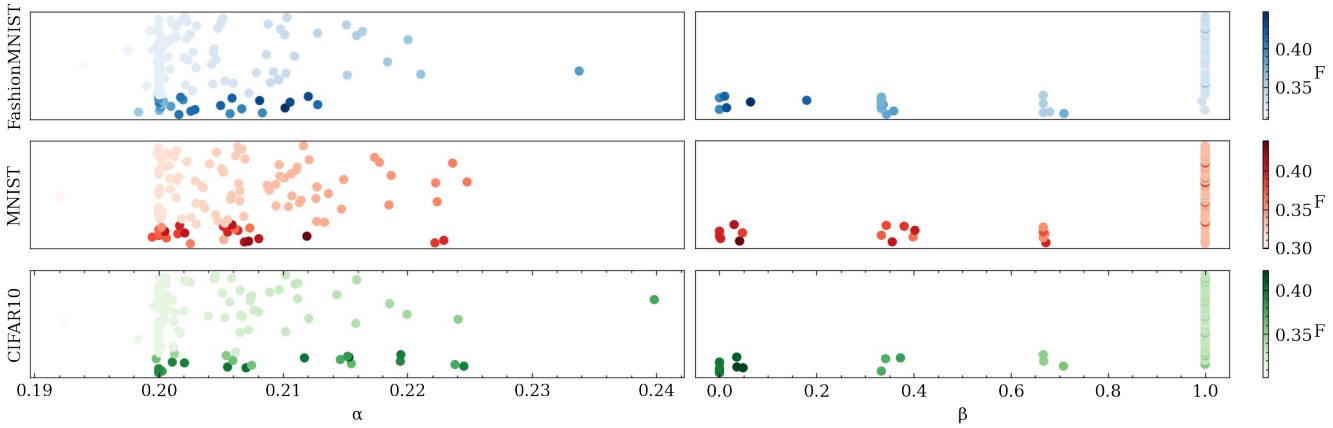


Figure S35. **Curricula influence the learning efficacy of the LwF [31] algorithm (Sec 3.2) across three datasets: MNIST, FashionMNIST, and CIFAR10 (Sec 3.1) for paradigm-I (5 classes, Sec 3.1).** We trained the LwF algorithm on all curricula from each dataset. Each dot represents one curriculum. We reported the average accuracy α over all the seen classes, highlighting how the algorithm adapts to learn new tasks **(left panel, Sec 5.1)**. The accuracy difference β reflects the forgetfulness of the algorithm across tasks **(right panel, Sec 5.1)**. We introduced $\mathcal{F}$ as the measure of the learning efficacy of a given curriculum (**Sec 3.3**) taking both α and β into account.

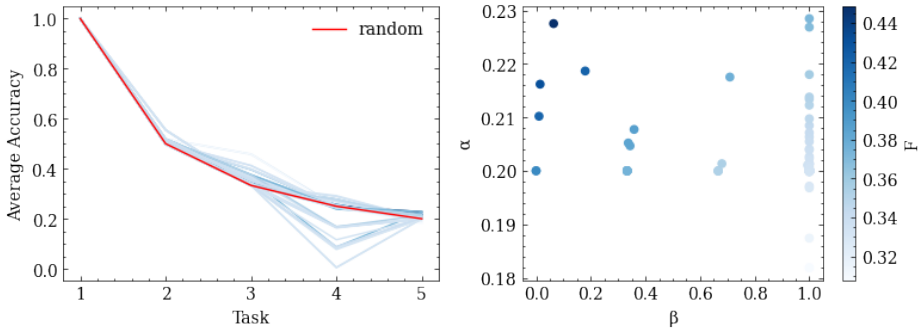


Figure S36. **Curricula impacts performance on FashionMNIST using LwF (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

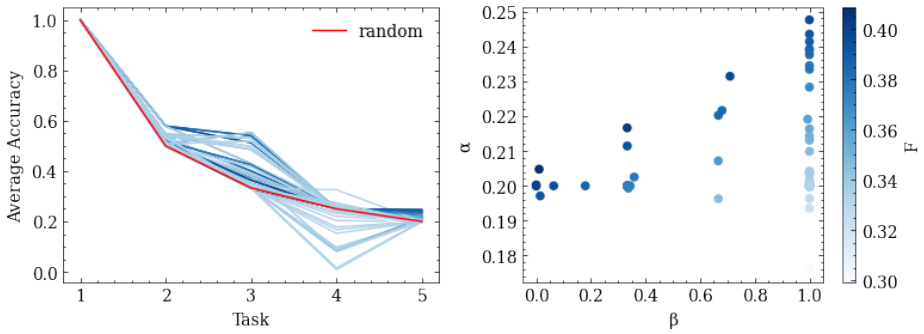


Figure S37. **Curricula impacts performance on MNIST using LwF (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

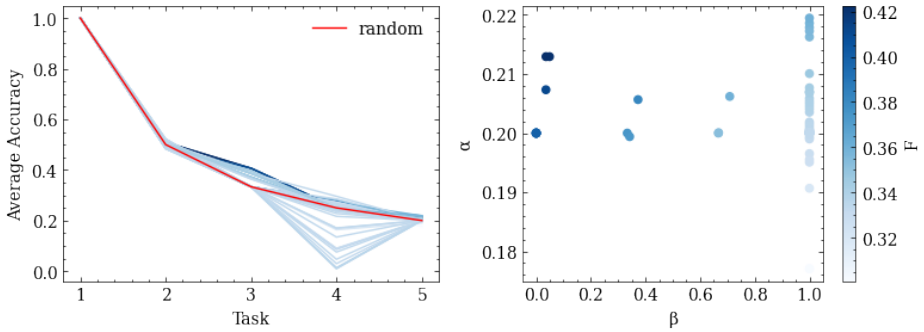


Figure S38. **Curricula impacts performance on CIFAR10 using LwF (Sec 3.2) for paradigm-I (5 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

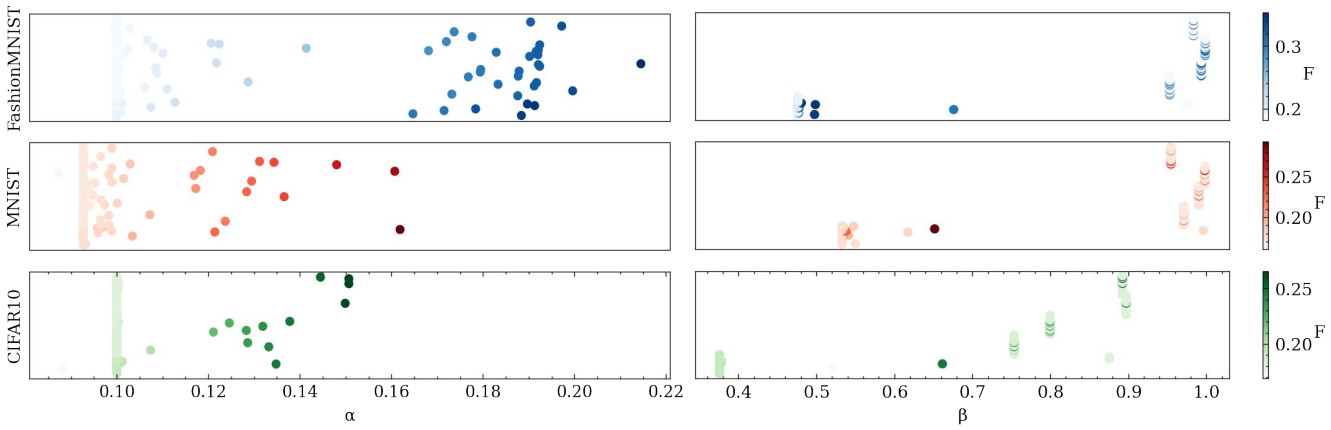


Figure S39. **Curricula influence the learning efficacy of the Vanilla continual learning algorithm (Sec 3.2) across three datasets: MNIST, FashionMNIST, and CIFAR10 (Sec 3.1) for paradigm-II (10 classes, Sec 3.1).** We trained the Vanilla continual learning algorithm on all curricula from each dataset. Each dot represents one curriculum. We reported the average accuracy α over all the seen classes, highlighting how the algorithm adapts to learn new tasks **(left panel, Sec 5.1)**. The accuracy difference β reflects the forgetfulness of the algorithm across tasks **(right panel, Sec 5.1)**. We introduced $\mathcal{F}$ as the measure of the learning efficacy of a given curriculum (**Sec 3.3**) taking both α and β into account.

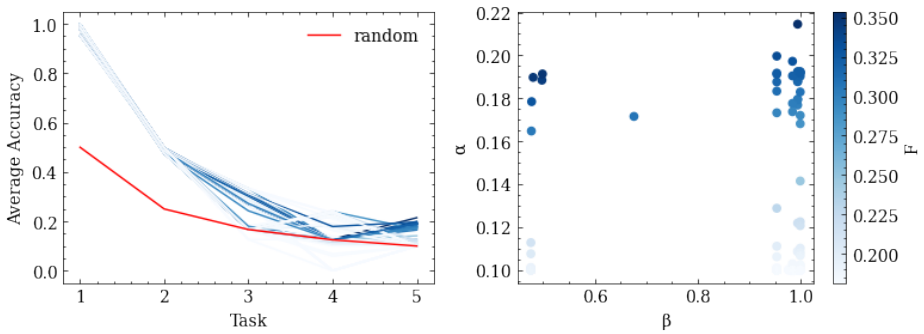


Figure S40. **Curricula impacts performance on FashionMNIST using the vanilla continual learning algorithm (Sec 3.2) for paradigm-II (10 classes, Sec 3.1)**. We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

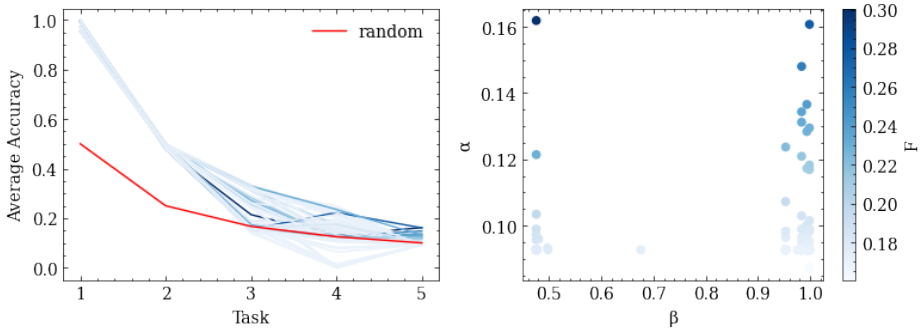


Figure S41. **Curricula impacts performance on MNIST using the vanilla continual learning algorithm (Sec 3.2) for paradigm-II (10 classes, Sec 3.1)**. We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

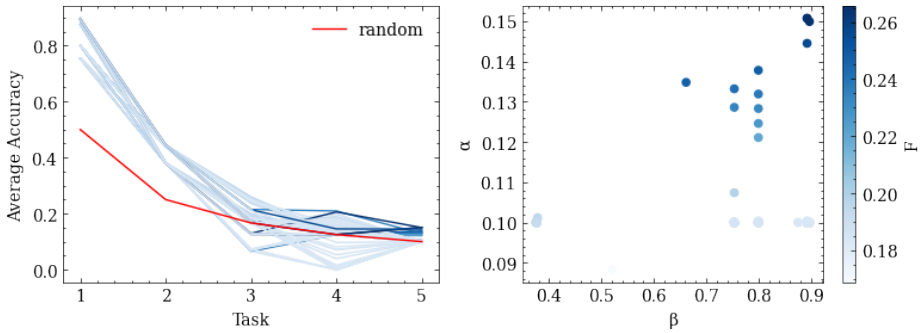


Figure S42. **Curricula impacts performance on CIFAR10 using the vanilla continual learning algorithm (Sec 3.2) for paradigm-II (10 classes, Sec 3.1)**. We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

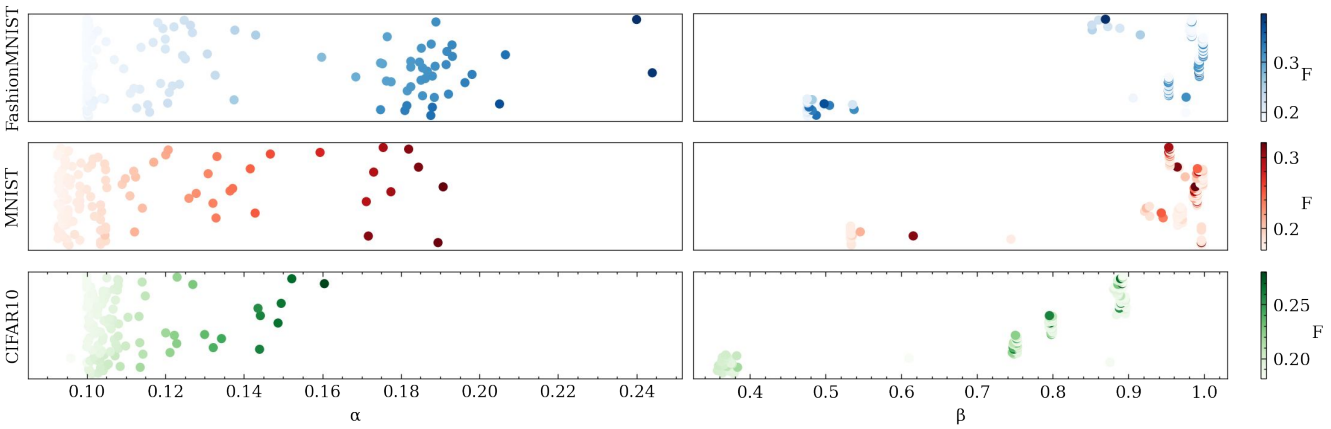


Figure S43. **Curricula influence the learning efficacy of the EWC [26] algorithm (Sec 3.2) across three datasets: MNIST, FashionMNIST, and CIFAR10 (Sec 3.1) for paradigm-II (10 classes, Sec 3.1).** We trained the EWC algorithm on all curricula from each dataset. Each dot represents one curriculum. We reported the average accuracy α over all the seen classes, highlighting how the algorithm adapts to learn new tasks **(left panel, Sec 5.1)**. The accuracy difference β reflects the forgetfulness of the algorithm across tasks **(right panel, Sec 5.1)**. We introduced $\mathcal{F}$ as the measure of the learning efficacy of a given curriculum (**Sec 3.3**) taking both α and β into account.

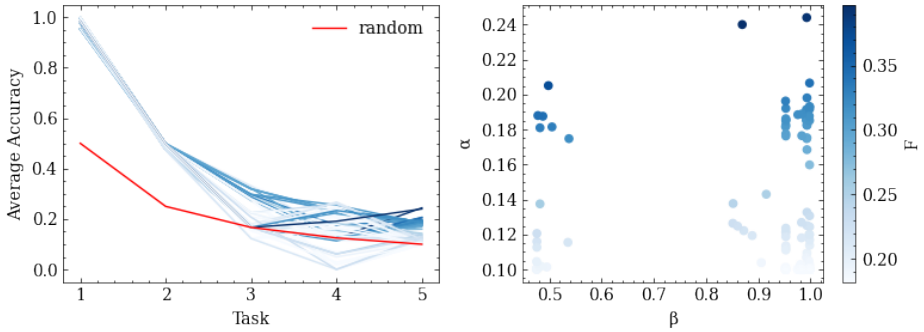


Figure S44. **Curricula impacts performance on FashionMNIST using EWC (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

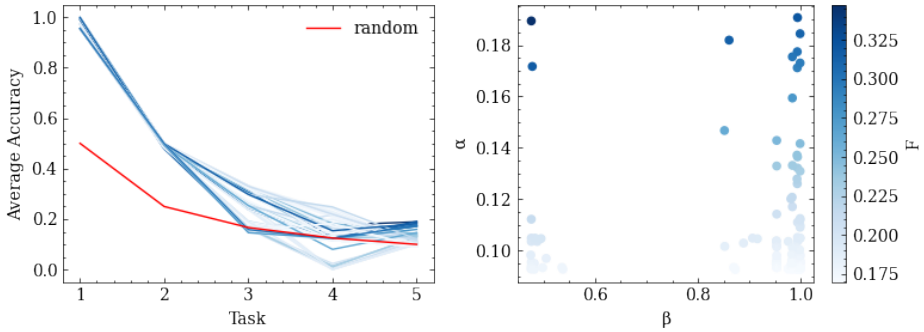


Figure S45. **Curricula impacts performance on MNIST using EWC (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

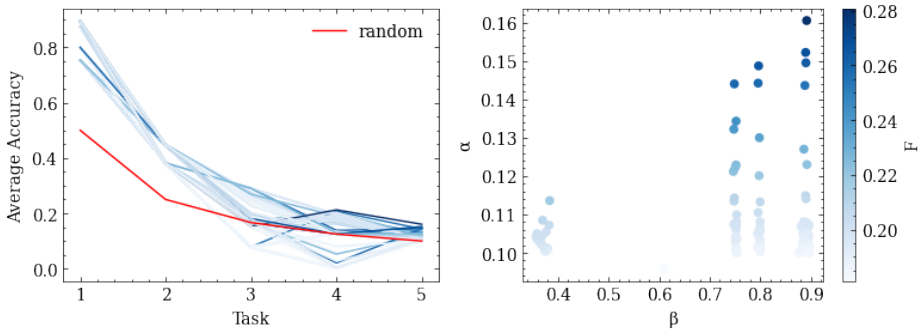


Figure S46. **Curricula impacts performance on CIFAR10 using EWC (Sec 3.2) for paradigm-II (10 classes, Sec 3.1)**. We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

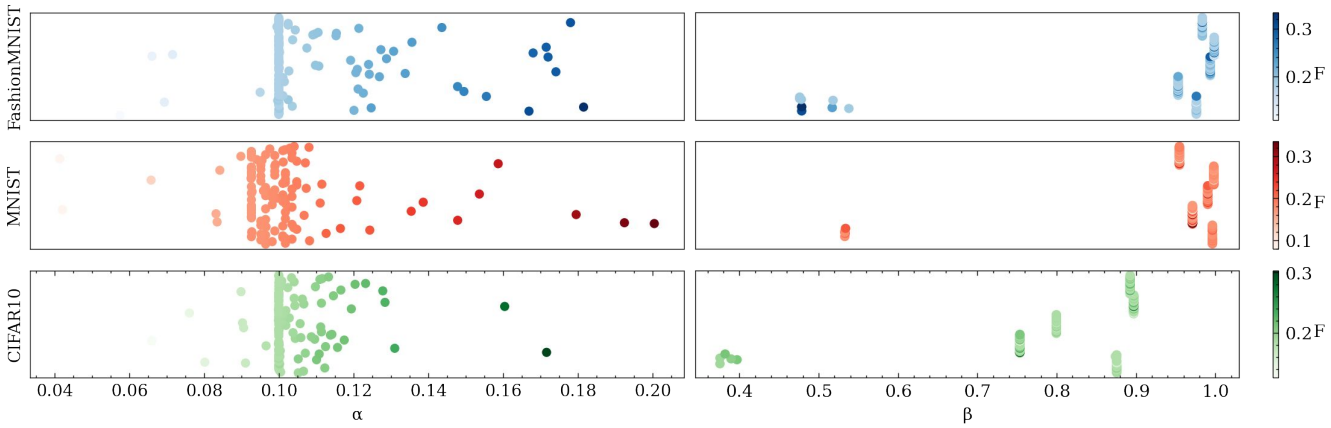


Figure S47. **Curricula influence the learning efficacy of the LwF [26] algorithm (Sec 3.2) across three datasets: MNIST, FashionMNIST, and CIFAR10 (Sec 3.1) for paradigm-II (10 classes, Sec 3.1)**. We trained the EWC algorithm on all curricula from each dataset. Each dot represents one curriculum. We reported the average accuracy α over all the seen classes, highlighting how the algorithm adapts to learn new tasks **(left panel, Sec 5.1)**. The accuracy difference β reflects the forgetfulness of the algorithm across tasks **(right panel, Sec 5.1)**. We introduced $\mathcal{F}$ as the measure of the learning efficacy of a given curriculum (**Sec 3.3**) taking both α and β into account.

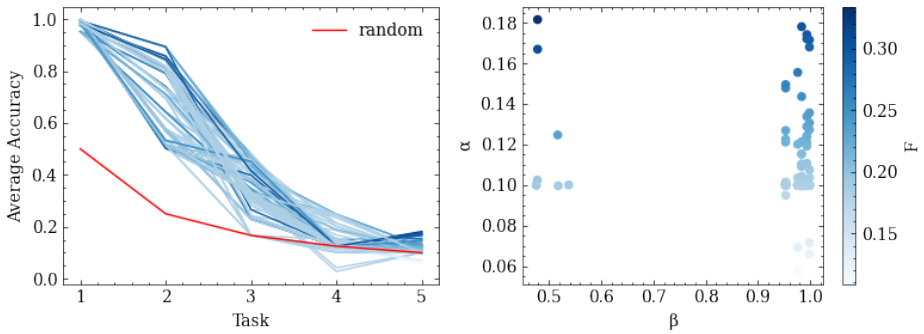


Figure S48. **Curricula impacts performance on FashionMNIST using LwF (Sec 3.2) for paradigm-II (10 classes, Sec 3.1)**. We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

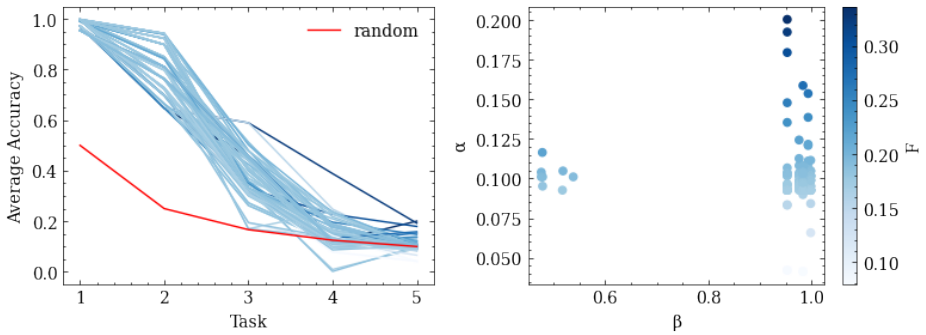


Figure S49. **Curricula impacts performance on MNIST using LwF (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

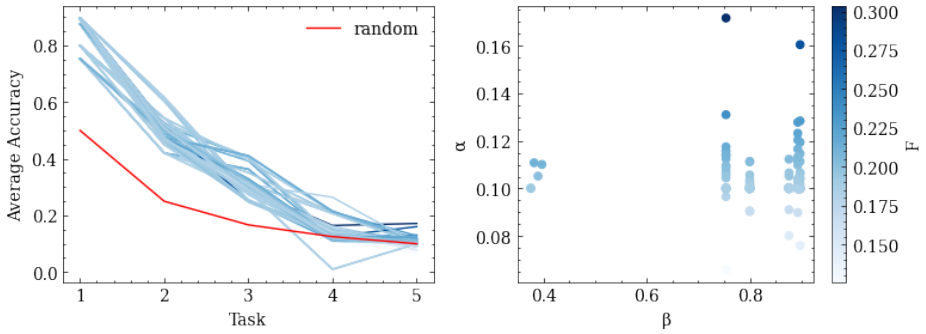


Figure S50. **Curricula impacts performance on CIFAR10 using LwF (Sec 3.2) for paradigm-II (10 classes, Sec 3.1).** We reported the average accuracy over tasks (**left**). We also plotted α vs β (**Sec 3.3**). The effectiveness measure, $\mathcal{F}$ (**Sec 3.3**) incorporates both α and β.

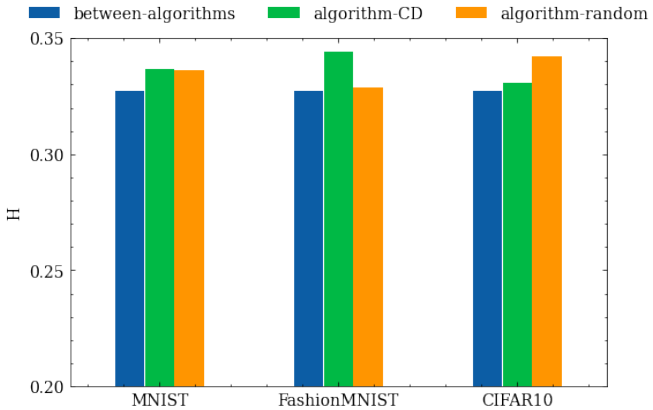


Figure S51. **There exists high agreement on optimal curricula determined by between-algorithms and algorithm-CD.** Curricula agreement $\mathcal{H}$ (**Sec 3.3**) is reported between pairs of continual learning algorithms $\mathcal{A}$s (between-algorithm, blue), between $\mathcal{A}$s and our CD (algorithm-CD, green), between $\mathcal{A}$ and the random designer (algorithm-random, orange) across all three datasets MNIST, FashionMNIST, and CIFAR10 (**Sec S5**) for Paradigm-II (10 classes, **Sec 3.1**).

overlap Counts top-5 criteria		Vanilla	EWC	LwF
FashionMNIST	designer	**3**	**1**	**3**
	chance	0.2	0.2	0.2
MNIST	designer	**3**	**2**	**4**
	chance	0.2	0.2	0.2
CIFAR10	designer	0	**2**	**4**
	chance	**0.2**	0.2	0.2

Table S1. **Our Curriculum Designer (CD) predicts optimal curricula.** Overlap counts (**Sec S3**) between the top-5 curricula by our CD and the empirically determined top-10 curricula across three continual learning algorithms (**Sec 3.2**) over all three datasets (**Sec 3.1**) for Paradigm-I (5 classes, **Sec 3.1**). The best is in bold.

overlap Counts top-10 criteria		Vanilla	EWC	LwF
FashionMNIST	designer	**3**	**2**	**3**
	chance	0.6	0.6	0.6
MNIST	designer	**4**	**5**	**4**
	chance	0.6	0.6	0.6
CIFAR10	designer	**1**	**4**	**5**
	chance	0.6	0.6	0.6

Table S2. **Our Curriculum Designer (CD) predicts optimal curricula.** Overlap counts (**Sec S3**) between the top-10 curricula by our CD and the empirically determined top-10 curricula across three continual learning algorithms (**Sec 3.2**) over all three datasets (**Sec 3.1**) for Paradigm-I (5 classes, **Sec 3.1**). The best is in bold.

overlap Counts top-20 criteria		Vanilla	EWC	LwF
FashionMNIST	designer	**3**	**2**	**3**
	chance	1.3	1.3	1.3
MNIST	designer	**4**	**5**	**4**
	chance	1.3	1.3	1.3
CIFAR10	designer	1	**4**	**5**
	chance	**1.3**	1.3	1.3

Table S3. **Our Curriculum Designer (CD) predicts optimal curricula.** Overlap counts (**Sec S3**) between the top-20 curricula by our CD and the empirically determined top-10 curricula across three continual learning algorithms (**Sec 3.2**) over all three datasets (**Sec 3.1**) for Paradigm-I (5 classes, **Sec 3.1**). The best is in bold.

overlap Counts top-5 criteria		Vanilla	EWC	LwF
FashionMNIST	designer	**2**	**3**	0
	chance	0.2	0.2	**0.2**
MNIST	designer	0	**1**	0
	chance	**0.2**	0.2	**0.2**
CIFAR10	designer	0	0	0
	chance	**0.2**	**0.2**	**0.2**

Table S4. **Our Curriculum Designer (CD) predicts optimal curricula.** Overlap counts (**Sec** S3) between the top-5 curricula by our CD and the empirically determined top-10 curricula across three continual learning algorithms (**Sec** 3.2) over all three datasets (**Sec** 3.1) for Paradigm-II (10 classes, **Sec** 3.1). The best is in bold.

overlap Counts top-10 criteria		Vanilla	EWC	LwF
FashionMNIST	designer	**3**	**3**	**1**
	chance	0.6	0.6	0.6
MNIST	designer	**1**	**1**	**2**
	chance	0.6	0.6	0.6
CIFAR10	designer	**1**	0	**1**
	chance	0.6	**0.6**	0.6

Table S5. **Our Curriculum Designer (CD) predicts optimal curricula.** Overlap counts (**Sec** S3) between the top-10 curricula by our CD and the empirically determined top-10 curricula across three continual learning algorithms (**Sec** 3.2) over all three datasets (**Sec** 3.1) for Paradigm-II (10 classes, **Sec** 3.1). The best is in bold.

overlap Counts top-20 criteria		Vanilla	EWC	LwF
FashionMNIST	designer	**5**	**4**	1
	chance	1.3	1.3	**1.3**
MNIST	designer	**2**	**2**	**3**
	chance	1.3	1.3	1.3
CIFAR10	designer	1	0	1
	chance	**1.3**	**1.3**	**1.3**

Table S6. **Our Curriculum Designer (CD) predicts optimal curricula.** Overlap counts (**Sec** S3) between the top-20 curricula by our CD and the empirically determined top-10 curricula across three continual learning algorithms (**Sec** 3.2) over all three datasets (**Sec** 3.1) for Paradigm-II (10 classes, **Sec** 3.1). The best is in bold.

overlap Counts top-30 criteria		Vanilla	EWC	LwF
FashionMNIST	designer	**5**	**4**	2
	chance	2.6	2.6	**2.6**
MNIST	designer	**4**	2	**3**
	chance	2.6	**2.6**	2.6
CIFAR10	designer	1	0	**3**
	chance	**2.6**	**2.6**	2.6

Table S7. **Our Curriculum Designer (CD) predicts optimal curricula.** Overlap counts (**Sec** S3) between the top-30 curricula by our CD and the empirically determined top-10 curricula across three continual learning algorithms (**Sec** 3.2) over all three datasets (**Sec** 3.1) for Paradigm-II (10 classes, **Sec** 3.1). The best is in bold.

overlap Counts		our CD	layer-11	layer-6	otdd	euclidean
FashionMNIST	Vanilla	**3**	**3**	**3**	2	**3**
	EWC	2	2	2	**3**	2
	LwF	**3**	2	2	2	**3**
MNIST	Vanilla	**4**	2	2	1	3
	EWC	**5**	2	2	2	**5**
	LwF	**4**	2	2	1	3
CIFAR10	Vanilla	**1**	**1**	**1**	**1**	**1**
	EWC	**4**	3	3	3	4
	LwF	5	4	4	4	**6**

Table S8. **Ablation results on our CD.** We show the overlaps count (**Sec 5.3**) between the top-30 curricula predicted by a curriculum designer and the top-10 empirical curricula determined across three continual learning algorithms (**Sec 3.2**) and three datasets (**Sec. 3.1**). Curriculum designers from the left to the right in the table along each row refer to our CD, our CD with the distance confusion matrix (**Sec 4.1**) computed based on the features extracted at layer 11 and 6 of the teacher network, and our CD with the distance confusion matrices computed with different distance metrics: Optimal Transport Dataset Distance (OTDD) [4] and Euclidean. The best in each row is bolded.

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. 7, 8, 10, 15, 18, 38

[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] 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, 6

[8] 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. 4

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

[10] 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

[11] 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

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

[13] 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

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

[15] 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

[16] 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. 1, 2

[17] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 16

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

[19] 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

[20] 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. 1, 2

[21] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017. 2

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

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3

[24] 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

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

[26] 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, 4, 7, 10, 12, 13, 14, 17, 29, 33, 34

[27] Pascal Klink, Hany Abdulsamad, Boris Belousov, and Jan Peters. Self-paced contextual reinforcement learning. In *Conference on Robot Learning*, pages 513–529. PMLR, 2020. 1, 2

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

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

[30] 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

[31] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2, 3, 4, 7, 10, 13, 17, 30

[32] 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. 3

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

[34] 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

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

[36] 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. 1, 2

[37] 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

[38] 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

[39] 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. 1, 2

[40] 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. 2

[41] 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

[42] 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

[43] 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. 1, 2

[44] 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

[45] 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. 1, 2

[46] 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. 2

[47] 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. 1, 2

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

[49] 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

[50] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? In *International Conference on Learning Representations*, 2021. 2, 3

[51] 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

[52] 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

[53] 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. 1, 2

[54] 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