CLDyB: Towards Dynamic Benchmarking for Continual Learning with Pre-trained Models

Anonymous authors
Paper under double-blind review

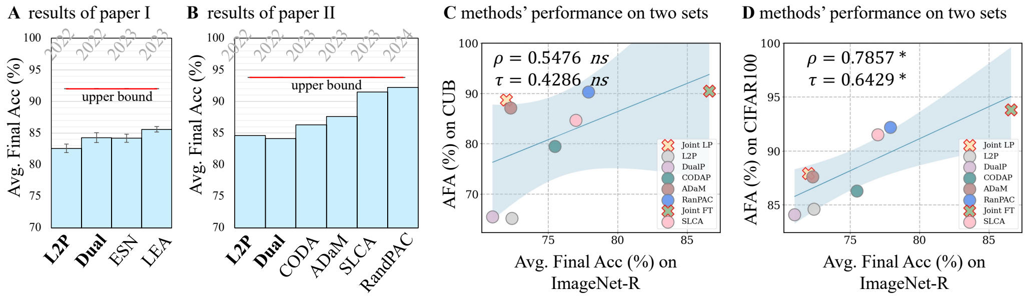

Figure 1: **Limitations of existing CL benchmarks.** We compiled comparison results of various CL methods from recent years on existing benchmarks in the literature to highlight several key issues with current CL benchmarks. **A** and **B** present results of multiple methods reported in two papers, (I) (Gao et al., 2023a) and (II) (McDonnell et al., 2024), on the Split-CIFAR-100 benchmark. We can observe limitations in terms of *benchmark upper bounds* and *the insufficient sensitivity to performance differences*. **C** and **D** future show the inconsistencies in method evaluation across datasets, highlighting *limited reliability of evaluation results* and *possible data contamination in pre-training stages*. For each method, we show its average final accuracy on two datasets in the 2-D correlation sub-graph and present Spearman's ρ and Kendall's τ to assess the correlation between two datasets for evaluations. '*ns*' and '$*$' means that the correlation coefficient is not statistically significant (p-value > 0.05) and statistically significant (p-value < 0.05), respectively.

Abstract

The emergence of the foundation model era has sparked immense research interest in utilizing pre-trained representations for continual learning (CL), yielding a series of strong CL methods with outstanding performance on standard evaluation benchmarks. Nonetheless, there are growing concerns regarding potential data contamination within the massive pre-training datasets. Furthermore, the static nature of standard evaluation benchmarks tends to oversimplify the complexities encountered in real-world CL scenarios, putting CL methods at risk of overfitting to these benchmarks while still lacking robustness needed for more demanding real-world applications. To solve these problems, this paper proposes a general framework to evaluate methods for **C**ontinual **L**earning on **Dy**namic **B**enchmarks (CLDyB). CLDyB continuously identifies inherently challenging tasks for the specified CL methods and evolving backbones, and dynamically determines the sequential order of tasks at each time step in CL using a tree-search algorithm, guided by an overarching goal to generate highly challenging task sequences for evaluation. To highlight the significance of dynamic evaluation on the CLDyB, we first simultaneously evaluate multiple state-of-the-art CL methods under CLDyB, resulting in a set of commonly challenging task sequences where existing CL methods tend to underperform. We intend to publicly release these task sequences for the CL community to facilitate the training and evaluation of more robust CL algorithms. Additionally, we perform individual evaluations of the CL methods under CLDyB, yielding informative evaluation results that reveal the specific strengths and weaknesses of each method.

1 INTRODUCTION

AI is undergoing a paradigm shift with the development of foundation models such as VIT (Dosovitskiy, 2020), CLIP (Radford et al., 2021), BERT (Devlin, 2018), which are trained on large-scale datasets and can be effectively adapted well to a wide range of downstream tasks. Continual learning (CL), as a crucial approach for model adaptation, faces both new opportunities and challenges amidst this paradigm shift. **Opportunities** arise from the complementary nature of CL and foundation models. Specifically, the foundation models will significantly benefit from continual learning to incrementally acquire new knowledge, while continual learning approaches can use the strong capability of foundation models as an advantageous starting point. **Challenges** primarily arise from the limitations of existing commonly used CL benchmarks in providing comprehensive and indicative evaluations of CL methods that start from stronger pre-trained foundation models.

Fig. 1 illustrates two fundamental limitations associated with current CL benchmarks, hinder their ability to deliver informative and reliable evaluations of CL methods, especially for those employing pre-trained models. Firstly, **data contamination**: the exponential growth in pre-training data volume for foundational models heightens the risk of overlap with downstream CL tasks. The high similarity in data distribution has led to performance saturation in recent years, making it tedious to compare CL methods based on marginal performance improvements. More critically, this raises concerns about whether recent progress in CL is largely attributable to exploiting more robust pre-trained models rather than genuine algorithmic innovation (Janson et al., 2022; Galashov et al., 2023), thereby hindering substantial progress within the CL community. Second, **limited reliability of evaluation results**. The static nature of conventional evaluation benchmarks often simplifies the complexities inherent in real-world continual learning scenarios. For instance, traditional benchmarks often assume that tasks are randomly sampled from classes within a single static dataset, and are presented sequentially in an unstructured, random manner. This is only marginally representative of continual learning in real-world scenarios, where task sequences can be diverse and ever-changing. Consequently, continual learning methods may overfit these simplified benchmarks while still lacking the robustness needed for effective performance in more demanding applications.

In light of the urgent need for a challenging and robust continual learning benchmark, we present CLDyB, a pioneering dynamic benchmarking asset specifically crafted to advance algorithmic development in continual learning with pre-trained models. At the core of CLDyB lies a versatile CLDyB-pipeline, which can be applied to any selected set of CL methods to generate task sequences dynamically during CL training for evaluation purposes. These evaluation sequences are characterized by tasks and their sequential order being both dynamically determined based on the current states of the CL models, following a two-step procedure at every time step in CL: (A) **Sampling for difficult tasks**. We propose a greedy task sampling algorithm which identifies continual learning tasks from a class data pool that are intrinsically and individually challenging to all evolving CL models at each step, including the pre-trained model. Thus, mitigating data contamination associated with using strong pre-trained backbones for CL. (B) **Searching for difficult sequences**. By formulating the search for challenging task sequential order as an online sequential decision making problem (Puterman, 1994), we employ the Monte Carlo tree search (MCTS) algorithm (Coulom, 2006) to dynamically plan and select the optimal next task for incremental learning and evaluation that leads to overall maximally challenging task sequences for the CL methods being evaluated. Consequently, the resultant task sequences temper the CL methods under realistic challenging scenarios, thereby enhancing the likelihood of algorithmic developments made with our benchmark translating into strong real-world performance.

We instantiate the CLDyB for class-incremental continual learning for visual classification (Zhou et al., 2024b) - one of the most popular CL problems nowadays. In our experiments, we first employ the CLDyB-pipeline on a group of representative CL methods, creating a set of commonly challenging task sequences appropriate for assessing state-of-the-art CL methods, yielding evaluation results that are generalizable. Additionally, we demonstrate CLDyB-pipeline as a general framework for assessing the individual robustness of CL methods, producing indicative evaluation results revealing the potential failure cases for the CL methods separately. Finally, we evaluate and analyse diverse CL methods on the CLDyB across different dimensions, such as robustness, memory efficiency and accuracy (see Fig. 4), which reveal the unique characteristics and weaknesses of different methods.

In summary, our major contributions are (1) exploring the potential of dynamic benchmarking to provide robust, consistent, and comprehensive evaluations for continual learning by designing the

CLDyB framework, which dynamically searches for challenging tasks, (2) providing a commonly challenging CL benchmark (task sequences) created by the CLDyB-pipeline, (3) showing the ability of the CLDyB-pipeline to search task sequences specifically challenging a given continual learning method, and (4) evaluating and comparing diverse CL methods from different perspectives on the CLDyB, which provides some insight to better understanding of current CL methods.

2 PRELIMINARY: CONTINUAL LEARNING

Class-incremental continual learning (CiCL) strives to build a universal classifier that can handle all seen classes by incrementally incorporating new knowledge while maintaining performance on previously learned tasks (Zhou et al., 2024b). More formally, in CiCL, a CL algorithm $\mathcal{A}$ iteratively trains a parameterized model f on a sequence of N classification tasks, $\mathbb{T}^{<N+1} := \{\mathcal{T}^1, \mathcal{T}^2, ..., \mathcal{T}^N\}$, introduced one at a time as $f^t = \mathcal{A}(f^{t-1}, \mathcal{T}^t)$. Each task $\mathcal{T}^t$ contains n_t^* (image, label) pairs $\{\boldsymbol{x}_j^t, y_j^t\}_{j=1}^{n_t^*}$, divided into training/validation/testing splits, where $*$ denotes the corresponding split. Note that all tasks have disjoint class label spaces. A key restriction in CL is that the algorithm cannot access data from past or future tasks while learning the current task at any time step t. Consequently, the central challenge in CL lies in training f to recognize new classes incrementally without suffering from catastrophic forgetting, where the model loses knowledge of previously learned classes as new ones are introduced. Additionally, the model must maintain high plasticity to adapt to upcoming data and tasks efficiently. We quantify forgetting and plasticity of a single CL model f^t at any specific time step t through the standard metrics **A**verage **F**orgetting **M**easure (Chaudhry et al., 2018) and **A**verage **L**earning **A**ccuracy (Riemer et al., 2018) defined as

$$\texttt{AFM}(\mathbb{T}^{<t+1}, f^t) = \frac{1}{t-1}\sum_{t'=1}^{t-1} \texttt{Acc}(\mathcal{T}^{t'}, f^{t'}) - \texttt{Acc}(\mathcal{T}^{t'}, f^{t}), \tag{1}$$

$$\texttt{ALA}(\mathbb{T}^{<t+1}, f^t) = \frac{1}{t}\sum_{t'=1}^{t} \texttt{Acc}(\mathcal{T}^{t'}, f^{t'}). \tag{2}$$

Both metrics are evaluated on all tasks appeared in the sequence $\mathbb{T}^{<t+1}$, and $\texttt{Acc}(\mathcal{T}, f)$ represents the empirical classification accuracy of f on the testing split of task $\mathcal{T}$. For notation consistency, the superscript ^{t} denotes a particular time step and $^{<t}$ refers to all time steps preceding t. Subscripts $_{i,j,k}$ are used to index elements in a set. Without loss of generality, we assume that there are $M \geq 1$ CL algorithms, each associated with its respective parameterized model for evaluation, i.e., $\mathbb{A} := \{\mathcal{A}_m\}_{m=1}^M$ and $\mathbb{F}^t := \{f_m^t\}_{m=1}^M$. We denote the average AFM and ALA over these M models as $\texttt{AFM}(\mathbb{T}^{<t+1}, \mathbb{F}^t) = \frac{1}{M}\sum_{m=1}^M \texttt{AFM}(\mathbb{T}^{<t+1}, f_m^t)$ and $\texttt{ALA}(\mathbb{T}^{<t+1}, \mathbb{F}^t) = \frac{1}{M}\sum_{m=1}^M \texttt{ALA}(\mathbb{T}^{<t+1}, f_m^t)$, respectively.

3 DYNAMIC BENCHMARKING FOR CONTINUAL LEARNING

Our objective is primarily to facilitate rapid algorithm development within the continual learning community while also providing a framework that enhances the likelihood of algorithmic developments made with our benchmark translating into strong real-world performance. In this section, we describe the two stages of our benchmarking framework, referred to as the **CLDyB-pipeline**, as shown in Fig. 2, i.e., sampling difficult tasks and searching for difficult task sequences, each corresponding to the two key challenges outlined in the Introduction.

3.1 TOWARDS INTRA-TASK DIFFICULTY – SAMPLING DIFFICULT TASKS

When the tasks used for continual learning of a pre-trained model largely overlap with the pre-training data, the resulting high performance is unsurprising and fails to fairly evaluate the effectiveness of CL algorithms (Galashov et al., 2023), as data contamination is the *dominant contributor* to high performance. To avoid such data contamination while simultaneously challenging the continually learned model, our first goal is to *construct continual learning tasks that are intrinsically and individually difficult to the evolving CL model f_m^t at each step t, including for the pre-trained model f_m^0*. Provided with a pool of classes, one straightforward recipe to screening difficult classification tasks involves: (1) randomly sampling K classes with equal probability from the pool to form a task,

(2) repeating step (1) to enumerate as many tasks as possible, and (3) selecting tasks that exceed a predefined difficulty threshold. Unfortunately, due to the vast size of the data pool which we will detail in Appendix A and the frequency of task screening (performed at every step t), this approach turns prohibitively expensive.

Greedy task sampling To significantly reduce the complexity, we turn to directly sample $|\mathbb{T}^t_{(g)}|$ most probably difficult tasks $\mathbb{T}^t_{(g)}$, where the sampling probability of a task $p^t(\mathcal{T})$ is naturally a joint distribution over K classes. By formulating this joint distribution with Markov Random Field (Sherrington & Kirkpatrick, 1975) and leveraging the observation that the difficulty of a multi-class classification task closely relates to the separability between pairs of classes (He et al., 2020), we define the sampling probability as proportional to the product of class pairwise potential functions, i.e.,

$$p^t(\mathcal{T}) \propto \prod_{p,q\in\{1,\cdots,K\}} \Psi(\mathcal{C}_p, \mathcal{C}_q), \quad \Psi(\mathcal{C}_p, \mathcal{C}_q) = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{z}(\cos(\boldsymbol{\mu}^{t-1}_{m,p}, \boldsymbol{\mu}^{t-1}_{m,q})). \tag{3}$$

The operator $\boldsymbol{z}(\cdot)$ denotes the min-max normalization $\frac{(\cdot)-\min(\cdot)}{\max(\cdot)-\min(\cdot)} \mapsto [0,1]$, and $\cos(\cdot,\cdot)$ represents cosine similarity. Here, $\boldsymbol{\mu}^t_{m,p}$ represents the prototype for class p according to the features extracted via f^{t-1}_m. This sampling probability thus targets the identification of the most difficult tasks that challenge the current feature space's capacity – a task involving more challenging pairwise class discrimination, reflected in the higher value of the pairwise potential product, is assigned greater sampling probability for its corresponding K classes.

Inspired by pairwise random field based image segmentation (Kohli et al., 2013), we propose to sample K classes that maximize $p^t(\mathcal{T})$ through a greedy algorithm, where classes from each task are selected sequentially. Initially, the first class is uniformly sampled from the entire set of available classes in our data pool $\mathbb{D}$, ensuring that the selected tasks in $\mathbb{T}^t_{(g)}$ exhibit sufficient diversity. Sub-

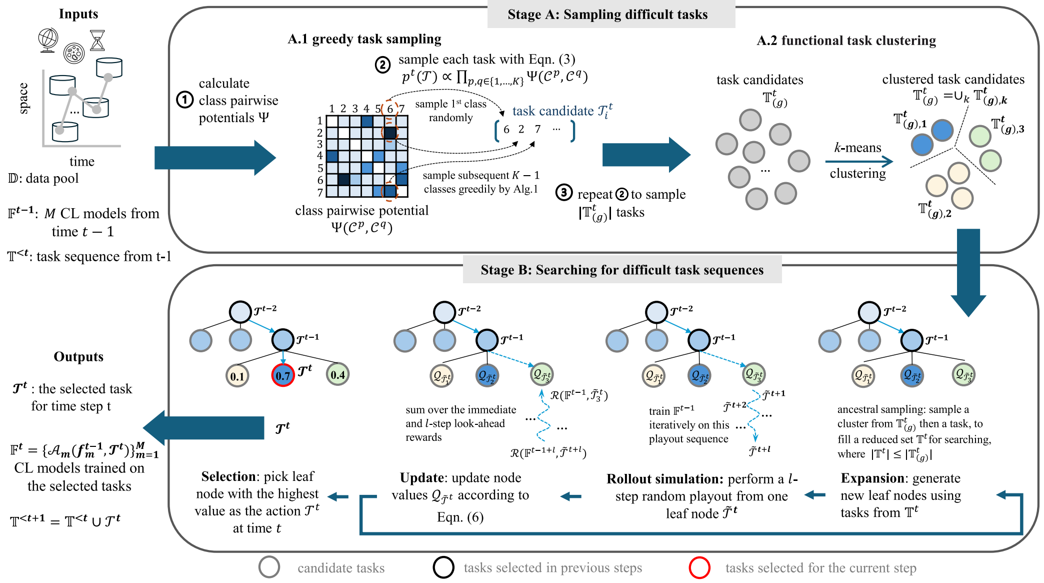
Inputs
space
time
$\mathbb{D}$: data pool
$\mathbb{F}^{t-1}$: M CL models from time $t-1$
$\mathbb{T}^{<t}$: task sequence from t-1
Stage A: Sampling difficult tasks
A.1 greedy task sampling
① calculate class pairwise potentials Ψ
② sample each task with Eqn. (3) $p^t(\mathcal{T}) \propto \prod_{p,q\in\{1,\dots,K\}} \Psi(\mathcal{C}^p, \mathcal{C}^q)$
1 2 3 4 5 6 7
sample 1st class randomly
task candidate $\mathcal{T}^t_i$
6 2 7 ···
sample subsequent $K-1$ classes greedily by Alg.1
class pairwise potential $\Psi(\mathcal{C}^p, \mathcal{C}^q)$
③ repeat ② to sample $|\mathbb{T}^t_{(g)}|$ tasks
A.2 functional task clustering
task candidates $\mathbb{T}^t_{(g)}$
clustered task candidates $\mathbb{T}^t_{(g)} = \cup_k \mathbb{T}^t_{(g),k}$
k-means clustering
$\mathbb{T}^t_{(g),1}$
$\mathbb{T}^t_{(g),2}$
$\mathbb{T}^t_{(g),3}$
Stage B: Searching for difficult task sequences
$\mathcal{T}^{t-2}$
$\mathcal{T}^{t-1}$
$\mathcal{T}^t$
0.1
0.7
0.4
$Q_{\tilde{\mathcal{T}}^t_1}$
$Q_{\tilde{\mathcal{T}}^t_2}$
$Q_{\tilde{\mathcal{T}}^t_3}$
$\mathcal{R}(\mathbb{F}^{t-1}, \tilde{\mathcal{T}}^t_3)$
$\mathcal{R}(\mathbb{F}^{t-1+l}, \tilde{\mathcal{T}}^{t+l})$
sum over the immediate and l-step look-ahead rewards
train $\mathbb{F}^{t-1}$ iteratively on this playout sequence
$\tilde{\mathcal{T}}^{t+1}$
$\tilde{\mathcal{T}}^{t+2}$
$\tilde{\mathcal{T}}^{t+l}$
ancestral sampling: sample a cluster from $\mathbb{T}^t_{(g)}$ then a task, to fill a reduced set $\mathbb{T}^t$ for searching, where $|\mathbb{T}^t| \leq |\mathbb{T}^t_{(g)}|$
Selection: pick leaf node with the highest value as the action $\mathcal{T}^t$ at time t
Update: update node values $Q_{\tilde{\mathcal{T}}^t}$ according to Eqn. (6)
Rollout simulation: perform a l-step random playout from one leaf node $\tilde{\mathcal{T}}^t$
Expansion: generate new leaf nodes using tasks from $\mathbb{T}^t$
Outputs
$\mathcal{T}^t$: the selected task for time step t
$\mathbb{F}^t = \{\mathcal{A}_m(f^{t-1}_m, \mathcal{T}^t)\}^M_{m=1}$ CL models trained on the selected tasks
$\mathbb{T}^{<t+1} = \mathbb{T}^{<t} \cup \mathcal{T}^t$
○ candidate tasks
○ tasks selected in previous steps
○ tasks selected for the current step

Figure 2: An overview of the proposed two-stage **CLDyB-pipeline** for dynamically constructing challenging task sequences to the M CL models $\mathbb{F}^{t-1}$, at each time step t. **Stage A**: Initially, the CLDyB-pipeline generates $|\mathbb{T}^t_{(g)}|$ intrinsically difficult tasks (Eqn. (3)) from a potentially time-evolving data pool. These candidates are then clustered based on the functional skills required for accurate classification, ensuring a comprehensive range of diverse tasks in the serach space $\mathbb{T}^t$. **Stage B**: Monte Carlo tree search is employed to estimate the values (Eqn. (6)) of the tasks in $|\mathbb{T}^t|$. The task with the highest value is selected as the t-th task for all CL models to learn and evaluate, resulting in the updated task sequence $\mathbb{T}^{<t+1} = \mathbb{T}^{<t} \cup \mathcal{T}^t$ and CL models $\mathbb{F}^t = \{\mathcal{A}_m(f^{t-1}_m, \mathcal{T}^t)\}^M_{m=1}$. Pseudocodes can be found in Alg. 3.

sequently, during each k-th iteration of the greedy process, we choose the class $\hat{p}$ that contributes the maximum increase in the product of potentials, i.e., $\hat{p} = \arg\max_p \prod_{q\in\{1,\cdots,k-1\}} \Psi(\mathcal{C}_p, \mathcal{C}_q)$. The greedy sampling continues until K distinct classes are sampled. We repeat this greedy process $|\mathbb{T}^t_{(g)}|$ times. Detailed procedures are available in Alg. 1.

Functional task clustering The set of sampled tasks $\mathbb{T}^t_{(g)}$ from the previous step is likely to exhibit (1) a biased distribution of tasks across the functional skills necessary for accurate classification, and (2) redundancy. This occurs because the M CL models naturally fall into different clusters based on their functional skills. Consequently, uniform task sampling tends to favor those tasks associated with the models belonging to the dominant functional skill cluster, which contradicts our goal of covering task candidates that challenge M CL models equally. To address this issue, we propose further clustering tasks based on their functional skills required and adopting an ancestral sampling approach to fill a reduced candidate set $\mathcal{T}^t$ with $|\mathcal{T}^t| \le |\mathcal{T}^t_{(g)}|$. We first uniformly sample a cluster and then draw a task from the chosen cluster. To create these clusters, we construct an M-dim functional vector for each task by evaluating the negative log-likelihood (NLL) of each sampled task under M k-NN classifiers derived from the CL models $\mathbb{F}^t_m$. The clusters are then formed using a standard k-means algorithm. Pseudocodes are provided in Alg. 2.

3.2 TOWARDS INTER-TASK DIFFICULTY – SEARCHING FOR DIFFICULT TASK SEQUENCES

The primary challenge in continual learning arises from its non-stationary nature, where tasks encountered in the wild are non-*i.i.d.* and previously unseen (Verwimp et al., 2024). As a result, even algorithms that perform quite well on current static benchmarks remain susceptible to real-world tasks, accompanied with either severe forgetting or negative transfer. For less overfitting to static benchmarks, our second goal is to *dynamically construct continual learning task sequences that pose significant challenges to M CL models across all time steps, i.e., by maximizing forgetting and minimizing plasticity*. Mathematically, for a sequence of N tasks, we formulate and solve the following optimization problem to identify these challenging sequences in a model-based manner,

$$\mathbb{T}^{<N+1} = \underset{\tilde{\mathbb{T}}^{<N+1}\in\boldsymbol{\pi}}{\arg\max}\ \texttt{AFM}(\tilde{\mathbb{T}}^{<N+1}, \mathbb{F}^N) - \texttt{ALA}(\tilde{\mathbb{T}}^{<N+1}, \mathbb{F}^N), \tag{4}$$

where $\boldsymbol{\pi}$ is the set of all possible ordered task sequences. Note that our goal for task sequence construction in Eqn. (4) opposes the general objectives of CL algorithms, which are to minimize forgetting and maximize learning plasticity. Intuitively, the stronger the CL algorithms (and models), the fewer weaknesses they will have, and the higher their performance will be when subjected to the adversarial task sequence constructed from Eqn. (4). Thus, the results of our evaluation provide insights into the robustness of CL algorithms. This helps reveal the shortcomings of state-of-the-art CL algorithms, and yield valuable training and assessment data which the CL community can leverage to develop even stronger algorithms.

Eqn. (4), however, is an offline optimization problem, and the global maximal depends on the sequence length N - which is an unknown variable or potentially infinite in real-world CL applications. To this end, we reformulate Eqn. (4) into an online sequential decision-making problem as shown in Eqn. (5). Concretely, we consider the CL algorithms as a deterministic stationary discrete-time system of the form $\mathbb{F}^t = \{\mathcal{A}_m(f^{t-1}_m, \mathcal{T}^t)\}_{m=1}^M$, which defines a state transition function from observable state $\mathbb{F}^{t-1}$ to $\mathbb{F}^t$ under action $\mathbb{T}^t$ returned by our task selection policy $\mathcal{A}^{\texttt{dycl}}$, i.e., $\mathcal{T}^t = \mathcal{A}^{\texttt{dycl}}(\mathbb{F}^{t-1})$, at time t. Defining the **immediate reward** function after transition from $\mathbb{F}^{t-1}$ to $\mathbb{F}^t$ with action $\mathcal{T}^t$ as $\mathcal{R}(\mathbb{F}^{t-1}, \mathcal{T}^t) = \texttt{AFM}(\tilde{\mathbb{T}}^{<t+1}, \mathbb{F}^t) - \texttt{ALA}(\tilde{\mathbb{T}}^{<t+1}, \mathbb{F}^t)$, and the value function of $\mathcal{A}^{\texttt{dycl}}$ at the initial state as $\mathcal{Q}_{\mathcal{A}^{\texttt{dycl}}}(\mathbb{F}^0)$, we aim to find an optimal policy $\mathcal{A}^{\texttt{dycl}}$, hence sequences of optimal actions under $\mathcal{A}^{\texttt{dycl}}$, that maximizes this value function, that is

$$\mathbb{T}^{<N+1} = \{\mathcal{T}^t = \mathcal{A}^{\texttt{dycl}}(\mathbb{F}^{t-1})\}_{t=1}^N;\ \text{s.t. } \mathcal{A}^{\texttt{dycl}} = \underset{\tilde{\mathcal{A}}^{\texttt{dycl}}}{\arg\max}\ \underbrace{\lim_{N\to\infty} \sum_{t=1}^{N} \alpha^{t-1} \mathcal{R}(\mathbb{F}^{t-1}, \tilde{\mathcal{A}}^{\texttt{dycl}}(\mathbb{F}^{t-1}))}_{\mathcal{Q}_{\tilde{\mathcal{A}}^{\texttt{dycl}}}(\mathbb{F}^0)}, \tag{5}$$

where $\alpha \in (0, 1]$ is a discount factor. Solving the above optimization problem presents two key challenges, including (1) accurately estimating the value function, which can vary significantly by CL algorithms and task properties (e.g., the number of classes in a task) and (2) efficiently navigating the vast and discrete search space, which is compounded by the large number of tasks $|\mathbb{T}^t|$ at each time step and the considerable length of the task sequence.

Tree search with approximation in value Monte Carlo tree search (MCTS) (Coulom, 2006) is one of the widely-adopted solutions for online sequential decision-making problems (Puterman, 1994), particularly well-suited to our CL setup due to its desirable ability in calculating value functions focused on a particular initial state on the fly. MCTS utilizes Monte Carlo simulations to approximate the value $\mathcal{Q}_{\mathcal{A}^{\texttt{dycl}}}(\mathbb{F}^t)$ of each state $\mathbb{F}^t$ in a search tree. In each action round, MCTS alternates between four steps: (1) rollout simulation, where it performs simulations that expand the tree recursively to estimate the current state value until a predefined simulation budget is exhausted; (2) update, where the action values and visit counts of all preceding states associated with the current state are updated; (3) selection, where the action (i.e., task $\mathcal{T}^t$) leading to the state with the highest immediate value is chosen; and (4) expansion, where all CL algorithms learn on the selected task (the chosen action) and transit from the current state $\mathbb{F}^{t-1}$ to the next state $\mathbb{F}^t$.

Unfortunately, in a CL setup, MCTS simulations are resource-intensive due to (a) the necessity for performing an $(N-t)$-step rollout to evaluate the value function $\mathcal{Q}_{\mathcal{A}^{\texttt{dycl}}}(\mathbb{F}^t)$, which involves training models on tasks at all $(N-t)$ steps, and (b) the indeterminate and potentially infinite termination length N in an open-ended CL experiment. To mitigate these issues, we substitute the exact value function with the sum of the immediate reward and an l-step look-ahead reward. The hyperparameter l mediates the trade-off between approximation bias and computational cost, allowing us to solve for a suboptimal action at each t as

$$\mathcal{T}^t = \arg\max_{\tilde{\mathcal{T}}^t \in \mathbb{T}^t} \mathcal{Q}_{\tilde{\mathcal{T}}^t}(\mathbb{F}^{t-1}) \approx \arg\max_{\tilde{\mathcal{T}}^t \in \mathbb{T}^t} \mathcal{R}(\mathbb{F}^{t-1}, \tilde{\mathcal{T}}^t) + \sum_{l'=0}^{l-1} \mathcal{R}(\mathcal{A}(\mathbb{F}^{t-1+l'}, \tilde{\mathcal{T}}^{t+l'}), \tilde{\mathcal{T}}^{t+l'+1}). \tag{6}$$

For simplicity, We set $l = 1$ and defer the use of more advanced techniques for value function approximation, such as learning a value network (Silver et al., 2016), for future work.

4 RELATED WORK

Class-incremental CL methods Recently studies show that pre-trained models (PTMs) inherently resist robustness against forgetting and exhibit strong generalizability to a variety of downstream tasks, making PTM-based CL an increasingly popular topic (Ostapenko et al., 2022; Zhang et al., 2023). Driven by the recent successes in parameter-efficient fine-tuning (PEFT), researchers have combined previous CL methods with PEFT approaches, culminating in numerous compute-efficient CL approaches tailored for PTMs, including orthogonal projection (Liang & Li, 2024; Qiao et al., 2024), model expansion (Zhou et al., 2021; Wang et al., 2022; Smith et al., 2022; Wang et al., 2023a) and ensemble methods (Gao et al., 2023b; Zhou et al., 2024a). On the other hand, representation-based strategies aim to preserve stable PTMs feature representation, typically by freezing the backbone after learning of the first task (Zhou et al., 2023a) or using low learning rates (Zhang et al., 2023). A non-parametric classifier for CL is then progressively constructed using second-order class feature statistics (Zhou et al., 2023a), enhanced with random projections (McDonnell et al., 2024) or intermediate representations (Ahrens et al., 2023). A more detailed discussion of conventional CL methods is deferred to Appendix B.1.

Dynamic benchmarking Evaluation of rapidly advancing PTMs using standard static benchmarks is becoming inadequate for a thorough assessment due to potential data contamination (Shi et al., 2023; Zhou et al., 2023b), bias and low robustness in evaluation results (McIntosh et al., 2024; Kiela et al., 2021). Dynabench (Kiela et al., 2021) and DynaBoard (Ma et al., 2021) address these challenges by using crowd-sourced data collection and creating dynamic benchmarks for LLM evaluation, with evolving test sets. To mitigate the substantial cost linked to manual data collection, recent work proposes to dynamically generate the test set utilizing directed acyclic graphs (Zhu et al., 2023) and a multi-agent framework (Wang et al., 2024). Orthogonal to dynamic benchmarking for reliable evaluation, dynamic programming has been used for test set selection to develop scalable and cost-effective evaluation methods for large-scale, ever-growing datasets (Prabhu et al., 2024).

5 EXPERIMENTS

CLDyB-pool As we have discussed in Introduction, Fig. 1 and Appendix A.3, image classification datasets, such as CIFAR-100 (Krizhevsky et al., 2009), CUB-200 (Wah et al., 2011) and

Stanford Cars (Krause et al., 2013) have been commonly used for the evaluation of CL strategies, but they have their respective limitations. To solve these limitations, we propose an initial strategy that dynamically assesses CL models by searching a series of tasks from a large data pool. It includes a diverse collection of real-world datasets for the primary experiments, as well as several generated datasets for additional explorations. There are a total of 26 datasets, including 2,505,185 images across 2,403 categories. These classes vary in granularity, ranging from coarse-grained to fine-grained, encompassing multi-national and multicultural diversity, and spanning across different time periods. The details are provided in Appendix A.

Two use-cases of the CLDyB-pipeline In alignment with our objectives in Section 1, we conduct our primary experiments in two distinct scenarios: (1) **C**ommonly **C**hallenging (**CC**) **CLDyB-seq**: the CLDyB-pipeline is applied to a group of representative CL methods *simultaneously* to identify common task sequences that are challenging for existing CL methods and pre-trained backbones; (2) **I**ndividually **C**hallenging (**IC**) **CLDyB-seq**: we apply the CLDyB-pipeline to each CL method *independently*, uncovering task sequences that pose challenges for the specific CL method.

CL algorithms We select a total of nine CL methods for evaluation based on two criteria: (1) competitive performance: prioritizing the latest published methods with top performance on standard CL benchmarks, and (2) high representativeness: ensuring the selected algorithms collectively encompass a wide variety of techniques. **Implementation details:** All experiments adhere stringently to the standard CiCL protocols. For CL evaluation, we resort to three standard metrics, including **A**verage **A**ccuracy (**AA**↑), **A**verage **R**etention[1] (**AR**↑) (Chaudhry et al., 2018), and **A**verage **L**earning **A**ccuracy (**ALA**↑) (Riemer et al., 2018). Higher values indicate better performance of the CL methods, yet it also implies that the benchmark poses *less* of a challenge for CL. More details on the selected CL methods and the experiment setups can be found in Appendix B.2 and B.3.

5.1 KEY RESULTS

CLDyB-pipeline finds commonly challenging task sequences In Fig. 3 and Fig. 9, it is evident that the discovered CC CLDyB-seq are consistently challenging for the state-of-the-art CL methods, as all evaluated CL methods struggle more with the CC CLDyB-seq than on the standard CL sequences, which consist of randomly ordered tasks with disjoint class labels. Overall, CL on the CC CLDyB-seq results in an average decrease of 26% in Final AA and 9% in Final AR across the CL methods, as illustrated in Fig 7. Moreover, in Fig. 3, when two separate groups of CL methods are used for searching and evaluating the CLDyB-seq, the CL methods reserved solely for evaluation still perform much worse on the CLDyB-seq compared to on the standard sequences. This cross-validation result underscores that the challenges posed by the CC CLDyB-seq are generalizable across different CL methods, making them a valid benchmark for assessing the performance of a wide range of CL methods.

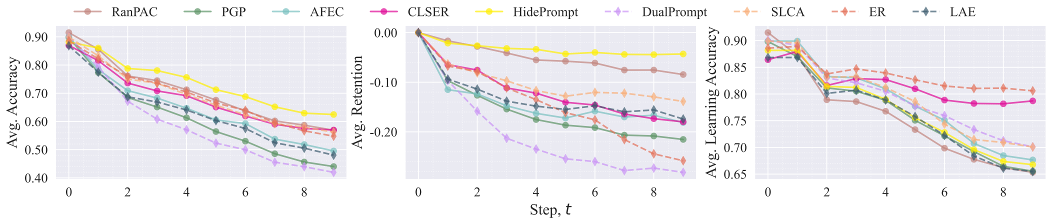

Figure 3: Performance of CL methods evaluated on the CC CLDyB-seq obtained from applying the CLDyB-pipeline to a subset of CL methods (solid). The remaining CL methods (dashed) are unseen during searching for the CLDyB-seq and are reserved for evaluation only.

We present some qualitative visualizations of the identified CC CLDyB-seq in Fig. 15 and Fig. 21, illustrating that the chosen tasks originate from various datasets, forming a sequence where similar tasks are interspersed with dissimilar ones. We believe this scenario is reflective of real-world CL situations, and the diversity observed here might be a reason why CLDyB-seq poses significant challenges for CL methods.

---

[1] Equivalent to negative **A**verage **F**orgetting **M**easure; The negation is applied to ensure that a higher value indicates better performance of the CL method, aligning with the other two evaluation metrics.

Table 1: Final average accuracy (%) of CL methods on various benchmarks. ‡ indicate published results. 'ns' and '∗' respectively indicate statistical non-significance and statistical significance.

Method	**Final Average Accuracy**			
	CIFAR-100‡	**ImageNet-R**‡	**CLDyB-seq (Ours)**	**Heldout**
RanPAC (McDonnell et al., 2024)	92.2	**78.1**	56.9	81.0
HidePrompt (Wang et al., 2023a)	**92.6**	75.1	**62.5**	**84.9**
DualPrompt (Wang et al., 2022)	86.5	68.1	41.9	70.8
PGP (Qiao et al., 2024)	86.9	69.3	44.0	68.7
LAE (Gao et al., 2023b)	85.6	72.7	48.1	71.1
SLCA (Zhang et al., 2023)	91.5	77.0	56.2	80.3
ER (Rolnick et al., 2018)	67.9	55.1	54.8	79.9
Spearman's ρ (per column & Heldout)	0.643 (ns)	0.643 (ns)	**0.964** ∗	N/A
Kendall's τ (per column & Heldout)	0.429 (ns)	0.429 (ns)	**0.905** ∗	N/A

Commonly challenging CLDyB-seq produce reliable evaluation results We show that evaluation results on the CC CLDyB-seq are a better indication of the relative performance of CL methods in an unseen CL scenario compared to those of the standard static benchmarks. In Tab. 1, we list the performance of CL methods on the CLDyB-seq, two standard CL benchmarks, and a Heldout benchmark consists of randomly ordered tasks constructed from multiple datasets that are *absent* in any of the prior benchmarks. The ranking correlations between these benchmarks and the Heldout are assessed using Spearman's and Kendall's correlation coefficients. CLDyB-seq achieves the highest correlation with the Heldout, demonstrating excellent generalization in evaluation results and significant potential for translating algorithmic development into reliable real-world performance.

Comparing CL methods on the CLDyB Applying the CLDyB-pipeline independently to each CL method results in evaluation outcomes on the IC CLDyB-seq, testifying the worst-case performance of the method, as illustrated in Eqn. (5). We thus define **Robustness** (↑) of a method as the average Final AA obtained on multiple independent IC CLDyB-seq discovered by the CLDyB-pipeline for that method, and include it as an evaluation criterion along with Memory Efficiency[2] in MB (↑) and the three standard metrics obtained on the CC CLDyB-seq for comparing the CL methods in Fig. 4.

We note the following key observations: **(a)** Methods adopting variants of replay, e.g., HidePrompt, RanPAC, CLSER and SLCA, generally demonstrate greater resistance to forgetting, resulting in higher final AA. However, these methods are markedly less memory efficient. **(b)** Exemplar-replay methods, specifically ER and CLSER, display significantly higher model plasticity compared to others, whereas PEFT methods like HidePrompt and LAE exhibit limited average learning accuracy (ALA), indicating poorer forward transfer ability - a criterion often overlooked in current continual earning literature. **(c)** More sophisticated methods tend to improve performance over their predecessors but are not necessarily more robust, e.g., PGP vs DualPrompt, CLSER vs ER. And finally **(d)**, we highlight that simple ER - despite not achieving the top performances in final AA and AR, still remains competitive in memory efficiency, final ALA, and robustness, thus ranking as the overall top CL method.

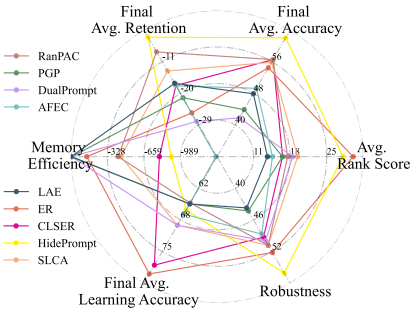

Figure 4: A comparison of CL methods evaluated on the CLDyB across multiple dimensions. Higher metric values indicate better performance. The comparison highlights the strengths and weaknesses of each method.

5.2 ADDITIONAL ANALYSIS

CLDyB-pipeline finds informative and individually challenging task sequences In Fig. 5 *Right*, we visualize the IC CLDyB-seq discovered for each CL method in terms of feature similarity between selected tasks. We notice that the IC CLDyB-seq exhibit meaningful clustering

---

[2] Equivalent to negative memory storage requirement. Detailed calculation can be found in Appendix B.4

in the dendrogram, reflecting commonality in the CL methods. For example, CLSER and ER both adopt exemplar replay, SLCA and HidePrompt both utilize classifier calibration.

A detailed examination of Fig. 5 *Right* reveals that the IC CLDyB-seq offer crucial insights into the limitations of individual CL method evaluated. Specifically, LAE and AFEC both exhibit increased vulnerability to sequences of tasks that present a large distribution shift in the feature space - as the CLDyB-pipeline tends to select dissimilar tasks for both methods in their IC CLDyB-seq. We hypothesize that this vulnerability arises because LAE and AFEC utilize moving average weight ensemble and parameter-wise regularization, respectively, which are both less effective when significant model updates are required for CL. Conversely, CLSER and ER are more susceptible to a sequence of closely similar tasks, which aligns with prior theoretical findings that demonstrate a correlation between task similarity and forgetting (Lee et al., 2021).

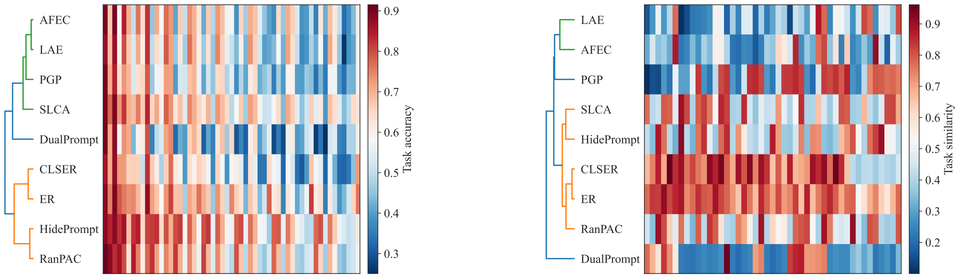

Figure 5: Dendrograms of CL methods based on *Left*: average accuracy trajectories obtained from continual learning on the CC CLDyB-seq, and *Right*: the flattened version of the 2D task-to-task similarity matrix obtained on the IC CLDyB-seq. The dengrograms exhibit noticeable consistency in their hierarchical structures, reflecting commonality in the CL methods.

CLDyB-pool is expandable over time with AI-generated data The CLDyB-pool need not to be static - we demonstrate a simple example of how our CLDyB-pipeline can offer valuable feedback for selecting new datasets to integrate into the CLDyB-pool, thus making the CLDyB-pool dynamically evolving over time and mitigating the issue of saturation in static benchmarks.

We initially employ the CLDyB-pipeline to dynamically evaluate selected CL methods for three tasks. Based on the tree-search history of the CLDyB-pipeline for this sequence, we observe tasks related to animals consistently yield higher rewards compared to other explored task candidates, indicating potential challenges for CL methods with these task categories. Given that, we employ diffusion models as a tool for rapid data expansion and add the generated images of novel animal categories to the CLDyB-pool at this point, simulating a dynamic, expansive CLDyB-pool. In Fig. 6, we compare the performance of the CL methods on upcoming tasks selected by the CLDyB-pipeline from the original and the augmented CLDyB-pools.

As observed in Fig. 19, the AI-generated classes are indeed frequently chosen, while the drop in final AA and AR in Fig. 6 demonstrates that the new sequence from the augmented CLDyB-pool generally presents greater challenges to the CL methods evaluated. Both findings confirm that our CLDyB-pool is highly expandable, thus avoiding saturation. Additionally, the tasks depicted in Fig. 19 transition in image style, such as evolving from real to sketch and painting, indicating that current CL methods may struggle with stylistic changes over time. In particular, in Fig. 6, we notice that rehearsal-free methods like AFEC (Wang et al., 2021) and PGP (Qiao et al., 2024) appear to be more susceptible to task sequences experiencing style shifts, showing a greater decline in performance compared to the others.

Ablation study We first remove each proposed component of the CLDyB-pipeline, generating four distinct ablated variants: **(a)** random task sequence, **(b)** the greedy task sampling is replaced by uniform sampling from the classes within each dataset in the CLDyB-pool to create the task candidates, **(c)** the functional task clustering is omitted, leading to each candidate for tree search being randomly selected from the task candidates produced by greedy task sampling, and **(d)** the tree search for selecting the next task is substituted with consistently choosing the task most similar

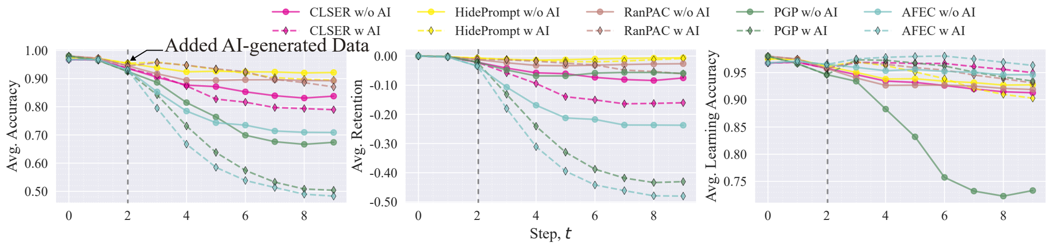

Figure 6: Performance on CC CLDyB-seq discovered from CLDyB-pool with (w AI) and without (w/o AI) additional diffusion-generated class images introduced after the third task.

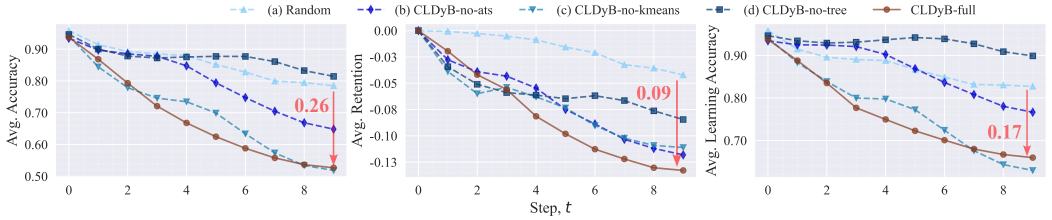

Figure 7: Ablation study comparing the average performance of CL methods on task sequences obtained by the CLDyB-pipeline and ablated variants. Arrows show performance gap between ablated variant (a) standard random sequences, and the CC CLDyB-seq of the full CLDyB-pipeline.

in representation to previously encountered tasks, as investigated in Bell & Lawrence (2022). These variants are compared to the full CLDyB-pipeline to evaluate the effectiveness of each component.

In Fig. 7, it is evident that the full version of the CLDyB-pipeline outperforms all ablated variants in identifying more challenging CLDyB-seq, which leads to lower accuracy and retention, thus validating the efficacy of each proposed component. Furthermore, to validate the effectiveness of the CLDyB-pipeline for long CL task sequences, we present the evaluation results for an extended sequence consisting of 40 tasks in Fig. 10. It is observed that the CL methods, on average, consistently under-perform on the CLDyB-seq compared to standard randomly ordered task sequences. Finally, in Fig. 11, we demonstrate that the CLDyB-pipeline remains effective in dynamically identifying CC CLDyB-seq to combat data contamination and performance saturation for CL methods using alternative pre-trained backbones.

6 CONCLUSION

The use of pre-trained models trained on large-scale data, combined with the lack of timely updates to existing benchmarks, has led to significant issues, such as insufficient and inconsistent evaluation, in evaluating continue learning (CL) method. To address these challenges, this paper proposes a reform of CL evaluation protocols to assess CL methods in a dynamic and realistic manner. Specifically, we introduce **CLDyB**, a framework that dynamically constructs and selects tasks over time using a tree-search algorithm to conduct more rigorous evaluation and effectively challenging CL methods. To validate the effectiveness of CLDyB, we use it to search for a common task sequence for evaluating various CL methods, as well as the specialized sequence tailored to a given CL method. Experimental results show that the common task sequence search by CLDyB-pipeline presents consistent challenges across different CL methods, while the specialized sequences effectively target and stress individual methods. Meanwhile, we design some interesting experiences, such as adding AI generated into CLDyB-pool , to show the generalization ability of CLDyB facing different data distributions. In addition, we provide extensive analysis and discussion of various CL methods on CLDyB-seq, which can potentially enhance the understanding of CL techniques and provide insights into optimizing CL strategies. We will release the CLDyB-pipeline and the common CLDyB-seq to the community, believing that CLDyB will promote the development of continue learning.

REFERENCES

Aditya. One Piece Anime, 2024. URL https://www.kaggle.com/datasets/aditya2803/one-piece-anime. Accessed: 2024-09-23.

Hongjoon Ahn, Donggyu Lee, Sungmin Cha, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. *ArXiv*, abs/1905.11614, 2019.

Kyra Ahrens, Hans Hergen Lehmann, Jae Hee Lee, and Stefan Wermter. Read between the layers: Leveraging intra-layer representations for rehearsal-free continual learning with pre-trained models. *ArXiv*, abs/2312.08888, 2023.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Neural Information Processing Systems*, 2019.

Zach Aluza. CNFood-241: Chinese Food Dataset, 2024. URL https://www.kaggle.com/datasets/zachaluza/cnfood-241. Accessed: 2024-09-23.

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uxxFrDwrE7Y.

Sourav Banerjee. Indian Food Images Dataset, 2023. URL https://www.kaggle.com/datasets/iamsouravbanerjee/indian-food-images-dataset. Accessed: 2024-09-23.

Sourav Banerjee. Animal Image Dataset: 90 Different Animals, 2024. URL https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals. Accessed: 2024-09-23.

Samuel J Bell and Neil D. Lawrence. The effect of task ordering in continual learning. *ArXiv*, abs/2205.13323, 2022. URL https://api.semanticscholar.org/CorpusID:249097496.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *ArXiv*, abs/2004.07211, 2020.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9496–9505, 2021.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *ArXiv*, abs/1812.00420, 2018.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pp. 72–83. Springer, 2006.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Mehrdad Farajtabar, Navid Azizan, Alexander Mott, and Ang Li. Orthogonal gradient descent for continual learning. *ArXiv*, abs/1910.07104, 2019.

Alexandre Galashov, Jovana Mitrovic, Dhruva Tirumala, Yee Whye Teh, Timothy Nguyen, Arslan Chaudhry, and Razvan Pascanu. Continually learning representations at scale. In Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup (eds.), *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pp. 534–547. PMLR, 22–25 Aug 2023. URL https://proceedings.mlr.press/v232/galashov23a.html.

Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11483–11493, 2023a.

Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jing Zhang. A unified continual learning framework with general parameter-efficient tuning. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11449–11459, 2023b.

Andrei Grigorev. Clothing Dataset Full, 2024. URL https://www.kaggle.com/datasets/agrigorev/clothing-dataset-full. Accessed: 2024-09-23.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 204–207, 2018.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8320–8329, 2020.

Muhammad Tanvirul Islam. Jute pest, 2024. URL https://www.kaggle.com/dsv/8332009.

Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. URL https://openreview.net/forum?id=dnVNYctP3S.

Joel Joseph and Alex Felix Gu. La-MAML: Look-ahead meta learning for continual learning, ML reproducibility challenge 2020, 2021. URL https://openreview.net/forum?id=d0svLMnvzWK.

Kaiska. Apparel Dataset, 2024. URL https://www.kaggle.com/datasets/kaiska/apparel-dataset. Accessed: 2024-09-23.

Parneet Kaur, Karan Sikka, Weijun Wang, Serge Belongie, and Ajay Divakaran. Foodx-251: a dataset for fine-grained food classification. *arXiv preprint arXiv:1907.06167*, 2019.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Talat, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in nlp. *ArXiv*, abs/2104.14337, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1611835114`.

Pushmeet Kohli, Anton Osokin, and Stefanie Jegelka. A principled deep random field model for image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1971–1978, 2013.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

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

Vencer Lanz. Sea Animals Image Dataset, 2024. URL `https://www.kaggle.com/datasets/vencerlanz09/sea-animals-image-dataste`. Accessed: 2024-09-23.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Sebastian Lee, Sebastian Goldt, and Andrew M. Saxe. Continual learning in the teacher-student setup: Impact of task similarity. In *International Conference on Machine Learning*, 2021. URL `https://api.semanticscholar.org/CorpusID:235790418`.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947, 2016.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23638–23647, 2024.

Weiduo Liao, Ying Wei, Mingchen Jiang, Qingfu Zhang, and Hisao Ishibuchi. Does continual learning meet compositionality? new benchmarks and an evaluation framework. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 33499–33513. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/6a42b45af2b72e6e5b5e3a6fe695809f-Paper-Datasets_and_Benchmarks.pdf`.

Zhiqiu Lin, Jia Shi, Deepak Pathak, and Deva Ramanan. The clear benchmark: Continual learning on real-world imagery. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, 2017.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6470–6479, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 10351–10367. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/55b1927fdafef39c48e5b73b5d61ea60-Paper.pdf`.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Timothy R. McIntosh, Teo Susnjak, Tong Liu, Paul Watters, and Malka N. Halgamuge. Inadequacies of large language model benchmarks in the era of generative artificial intelligence. *ArXiv*, abs/2402.09880, 2024.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkQqq0gRb.

Oleksiy Ostapenko, Timothée Lesort, Pau Rodr'iguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Continual learning with foundation models: An empirical study of latent replay. In *CoLLAs*, 2022.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pp. 1–8. IEEE, 2007.

Greg Piosenka. Balls Image Classification, 2024a. URL https://www.kaggle.com/datasets/gpiosenka/balls-image-classification. Accessed: 2024-09-23.

Greg Piosenka. Sports Classification, 2024b. URL https://www.kaggle.com/datasets/gpiosenka/sports-classification. Accessed: 2024-09-23.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*.

Ameya Prabhu, Vishaal Udandarao, Philip H. S. Torr, Matthias Bethge, Adel Bibi, and Samuel Albanie. Lifelong benchmarks: Efficient model evaluation in an era of rapid progress. *ArXiv*, abs/2402.19472, 2024.

Martin L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. 1994.

Xiaoman Qi, Panpan Zhu, Yuebin Wang, Liqiang Zhang, Junhuan Peng, Mengfan Wu, Jialong Chen, Xudong Zhao, Ning Zang, and P Takis Mathiopoulos. Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:337–350, 2020.

Jingyang Qiao, zhizhong zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, and Yuan Xie. Prompt gradient projection for continual learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=EH2O3h7sBI.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2016.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *ArXiv*, abs/1810.11910, 2018.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *Neural Information Processing Systems*, 2018.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3AOj0RCNC2.

Jonathan Schwarz, Wojciech M. Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. *ArXiv*, abs/1805.06370, 2018.

Kritik Seth. Fruit and Vegetable Image Recognition, 2024. URL https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition. Accessed: 2024-09-23.

David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35:1792–1796, 1975.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke S. Zettlemoyer. Detecting pretraining data from large language models. *ArXiv*, abs/2310.16789, 2023.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Neural Information Processing Systems*, 2017.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, L. Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

James Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogério Schmidt Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. *CVPR*, pp. 11909–11919, 2022.

Stealth Technologies. Rock Classification, 2024. URL https://www.kaggle.com/datasets/stealthtechnologies/rock-classification. Accessed: 2024-09-23.

Phuc Thai. Butterfly Image Classification, 2024. URL https://www.kaggle.com/datasets/phucthaiv02/butterfly-image-classification. Accessed: 2024-09-23.

Michalis K. Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkxCzeHFDB.

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

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. Afec: Active forgetting of negative transfer in continual learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22379–22391. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bc6dc48b743dc5d013b1abaebd2faed2-Paper.pdf.

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=9XieH21Tlf.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:5362–5383, 2023b.

Siyuan Wang, Zhuohan Long, Zhihao Fan, Zhongyu Wei, and Xuanjing Huang. Benchmark self-evolving: A multi-agent framework for dynamic llm evaluation. *ArXiv*, abs/2402.11443, 2024.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. *ArXiv*, abs/2204.04799, 2022.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. *ArXiv*, abs/2106.01085, 2021.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. *ICCV)*, pp. 19091–19101, 2023.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Da-Wei Zhou, Han-Jia Ye, De chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *ArXiv*, abs/2303.07338, 2023a.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. *CVPR*, pp. 23554–23564, 2024a.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024b.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130:2337 – 2348, 2021.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your llm an evaluation benchmark cheater. *ArXiv*, abs/2311.01964, 2023b.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *International Conference on Learning Representations*, 2023.

APPENDIX

A CLDYB DATA POOL

A.1 CONSTRUCTING THE CLDYB-POOL

In this section, we will introduce the building of the data pool, CLDyB-pool. The details are as follows.

Real-world data. The CLDyB-pool real-world data inludes 2,043 classes from 22 publicly available image classification datasets. Subjects of these datasets have a broad distribution, including animals, plants, scenes, food, etc. The subjects across two major aspects within the CLDyB-pool: 1) Natural and Biological Sciences include classes of fauna, flora and other natural elements and ecosystems. These subjects cover various levels of granularity, from coarse-grained, such as different categories of animals from Animal-90 (Banerjee, 2024), to fine-grained categories, such as various classes of butterflies from Butterfly-70 (Thai, 2024), 2) Human-Made Objects/Scenes contain images of clothing, food, buildings, and entertainment scenes. This group also captures multi-national and multicultural diversity, highlighting historical diversity across different regions and eras. For example, we have food classes from Food-101 (Bossard et al., 2014), CNFOOD-241(Aluza, 2024) (Chinese cuisine), Indian Food Images (Banerjee, 2023) etc., datasets. which includes cuisines of a variety of regions and cultures. By involving data from the two aspects, CLDyB-pool includes a wider range of domains and diverse classes, offering broader distribution coverage, thus offering an ideal test platform for CL methods.

Diffusion mode generated data. The CLDyB-pool will also include the 4 AI-generated datasets used in our experiments in Section 5.2. These datasets are generated by SDXL (Podell et al.). Specifically, 360 classes of three animal and one product datasets are generated by using prompts '*A high-quality image of a kind of animal:* {*animal name*}', '*A high-quality sketch image of a kind of animal:* {*animal name*}', '*A high-quality* {*image style*} *image of a kind of animal:* {*animal name*}' and '*A high-quality image of a kind of product:* {*product name*}'. Incorporating the generated data will expand the CLDyB pool to include additional distributions, leading to more challenging task series (we have discussed in Section 5.2).

We use the CLDyB-pool as a dynamic source to dynamically search for task series. We expect that CLDyB-pool will be valuable for ongoing research in continual learning and beyond. In the future, our goal is to develop CLDyB-pool into a dynamic online project through open-source collaboration with the community.

A.2 STATISTICS AND DISCUSSIONS OF CLDYB-POOL

Statistics. The details of the datasets in the data pool are shown in Table 2. There are two types of datasets: real-world data and AI-generated data created with SDXL.

License. The CLDyB-pool dataset inherits licenses from its respective sources, as long as those licenses are explicitly stated. If a license is not explicitly stated, the dataset in CLDyB-pool is distributed under the CC BY-NC 4.0 license.1 [3], which restricts its use for non-commercial purposes.

A.3 CLASS-INCREMENTAL CONTINUAL LEARNING BENCHMARKS

The common practice for evaluating CiCL methods, as first established in Rebuffi et al. (2016), involves partitioning the classes of a labeled dataset into sequentially ordered tasks with non-overlapping class labels for CiCL training and evaluation. Typical datasets used include MNIST (Deng, 2012), CIFAR-100 (Krizhevsky et al., 2009), CUB-200 (Wah et al., 2011) and ImageNet variants (Hendrycks et al., 2020; Le & Yang, 2015). Other benchmarks, e.g., OmniBenchmark and VTAB proposed in (Zhou et al., 2023a), consist of a series of standard classification datasets, each regarded as an individual task for incremental learning. Beyond remixing readily available datasets, several new datasets have been specifically curated for CiCL benchmarking, including Core50 (Lomonaco & Maltoni, 2017) which includes various objects in diverse settings ,

---

[3] https://creativecommons.org/licenses/by-nc-sa/4.0/

Table 2: **Statistics of datasets in current version CLDyB-pool.** '# class' and '# image' mean the class and image numbers, respectively. For some datasets. their # classes are different from their official version because we selected the classes with a limited number of images(< 45).

Type	Dataset	# class	# image
Real-world data	Fruits-Vegetable (Seth, 2024)	35	3,047
	Jute-Pest (Islam, 2024)	17	7,151
	Animal-90 (Banerjee, 2024)	90	5,400
	Sea-animal (Lanz, 2024)	23	13,711
	Rock (Technologies, 2024)	9	3,687
	Butterfly-75 (Thai, 2024)	75	6,499
	Clothing-20 (Kaiska, 2024)	17	5,325
	Apparel (Grigorev, 2024)	37	16,170
	Food101 (Bossard et al., 2014)	101	101,000
	CNFOOD241 (Aluza, 2024)	240	170,835
	Indian-Food (Banerjee, 2023)	80	4,000
	FoodX251 (Kaur et al., 2019)	250	118,441
	Oxford5k (Philbin et al., 2007)	17	5,063
	Places365 (Zhou et al., 2017)	365	1,803,460
	SUN397 (Xiao et al., 2010)	339	17,355
	EuroSAT-RGB (Helber et al., 2018)	10	27,000
	MLRSNet (Qi et al., 2020)	46	109,161
	RESISC45 (Cheng et al., 2017)	45	5,100
	FGVC-Aircraft (Maji et al., 2013)	100	10,000
	Sports100 (Piosenka, 2024b)	100	13,492
	One-piece-Anime (Aditya, 2024)	18	11,737
	Balls30 (Piosenka, 2024a)	29	3,708
SDXL-generated data	Product-71	71	7,095
	Animal-multi-styles	133	19,962
	Animal-sketch	77	8,494
	Animal	79	8,292
CLDyB-pool	-	2,403	2,505,185

and CLEAR (Lin et al., 2021) which features a natural temporal progression of visual concepts. Furthermore, Liao et al. (2023) have introduced CGQA and COBJ benchmarks for evaluating the compositional generalization abilities of CL methods. Nevertheless, all current CiCL benchmarks remain static, and none are explicitly tailored to accommodate evaluation of CL methods with strong pre-trained models.

B CONTINUAL LEARNING ALGORITHMS

B.1 A BRIEF INTRODUCTION OF TRADITIONAL CL METHODS

Traditional class-incremental continual learning algorithms can be broadly divided into (i) rehearsal-based, (ii) regularization-based, (iii) model-based, (iv) optimization-based and (v) representation-based, see Wang et al. (2023b) for a comprehensive survey.

Rehearsal-based approaches (Rolnick et al., 2018) utilize a memory buffer to store and replay old training samples or task-specific data when learning new tasks. Recent work emphasizes exemplar selection (Aljundi et al., 2019; Yoon et al., 2021), constraint optimization (Chaudhry et al., 2018; Lopez-Paz & Ranzato, 2017) and generative replay (Shin et al., 2017). Regularization-based (Kirkpatrick et al., 2017) methods restrict significant changes in the CL model and thus preserve old knowledge. These include regularization of important parameters, with importance measured by various approximations like the Fisher information matrix (Kirkpatrick et al., 2017), weight uncertainty (Ahn et al., 2019) and variational posterior (Nguyen et al., 2018); as well as functional regularization utilizing Gaussian processes (Titsias et al., 2020), knowledge distillation (Li & Hoiem, 2016; Buzzega et al., 2020), and contrastive learning (Cha et al., 2021). Model-based methods often involve model expansion (Schwarz et al., 2018; Wang et al., 2023a) and parameter isolation (Wang et al., 2022) to handle and safeguard task-specific knowledge. Lastly, optimization-based methods use techniques such as orthogonal gradient projections (Farajtabar et al., 2019; Saha et al., 2021) and meta-learning (Riemer et al., 2018; Joseph & Gu, 2021) to mitigate negative interference between tasks.

B.2 SELECTED CL METHODS FOR EVALUATION

We provide a brief description of the representative CL methods used for evaluation in this work below. The CL techniques used by different CL methods are shown in Tab. 3 for a better comparison between the methods. All CL methods are capable of using a pre-trained vision backbone as a starting point for CL.

SLCA (Zhang et al., 2023) tunes the backbone (feature encoder) with a small learning rate while tuning the classifier with a large learning rate. Although naive, such strategy ensures that the model can extract stable feature, which is important to mitigate catastrophic forgetting. Besides that, to resist the forgetting of the classifier, it also models and replays the class-wise feature distribution to calibrate the classifier.

RanPAC (McDonnell et al., 2024) combines Random Projection (RP) and Class-Prototype (CP) strategies for continual learning. After training on the first task, RanPAC freezes the backbone network and applies a Random Projection layer to the features. It then decorrelates class prototypes using the inverse of the Gram matrix of the projected features. This approach avoids catastrophic forgetting and achieves strong performance without using rehearsal memory.

LAE (Gao et al., 2023b) maintains a Online PET module (regular training) and uses exponential moving average to update an Offline PET module. In inference, LAE uses a simple model ensemble strategy, i.e., choosing the “most confident” of the two models.

PGP (Qiao et al., 2024) find that reaching the orthogonal condition for the gradient of prompts can effectively prevent forgetting. They achieved this goal by conducting Singular Value Decomposition (SVD) on an element-wise sum space between input space and prompt space.

HidePrompt (Wang et al., 2023a) adds several optimizations to the previous Prompt-based method, like prompt-ensemble strategy, constrastive feature constraint, and a joint optimization for three tasks (Within-Task Prediction,Task-Identity Prediction and Task Adaptive Prediction).

DualPrompt (Wang et al., 2022) is a typical prompt based continual learning method. Based on L2P, it further explores the importance of prompt depth by attaching prompts to different layers. It also separate general and task-specific prompts, so that model could better learn both global and local knowledge.

CLSER (Arani et al., 2022) maintains a stable model and a plastic model, standing for long-term and short-term semantic memory, by different EMA strategies. During training, a mse distillation loss would be added between the working model and the most confident EMA model. Therefore, this method provides an adaptive trade-off between stability and malleability.

AFEC (Wang et al., 2021), inspired by biological neural networks, proposed a method to actively forget old knowledge conflicting with new experience. Based on Elastic Weight Consolidation (EWC), another fisher information matrix of the current training model and task-specific fine-tuned model is introduced as regularization term in the loss function. This allows the model to actively forget some of the old knowledge, rather than preferring to retain it all.

ER (Rolnick et al., 2018) is the most simple and fundamental rehearsal based method. It simply stores previously seen data in a fixed memory buffer, and replays them during training to mitigate catastrophic forgetting of previous tasks.

Table 3: Summary of techniques employed by the selected continual learning methods.

Methods		SLCA (Zhang et al., 2023)	RanPAC (McDonnell et al., 2024)	LAE (Gao et al., 2023b)	PGP (Qiao et al., 2024)	HidePrompt Wang et al. (2023a)	DualPrompt (Wang et al., 2022)	CLSER (Arani et al., 2022)	AFEC (Wang et al., 2021)	ER (Rolnick et al., 2018)
Buffer	Exemplar							✓		✓
	Class statistics	✓	✓			✓				
Training	Full	✓	✓					✓	✓	✓
	PEFT			✓	✓	✓	✓			
Regularization	Functional							✓		
	Parameter								✓	
	Orthogonality				✓					
Model	Isolation & expansion				✓	✓	✓			
	Ensemble			✓		✓		✓		

B.3 EXPERIMENT DETAILS

All experiments adhere stringently to the standard CiCL protocols. For CL evaluation, we resort to three standard metrics, including **A**verage **A**ccuracy (**AA**↑), **A**verage **R**etention[4] (**AR**↑) (Chaudhry et al., 2018), and **A**verage **L**earning **A**ccuracy (**ALA**↑) (Riemer et al., 2018). Higher values indicate better performance of the CL methods, yet it also implies that the benchmark poses *less* of a challenge for CL. More details on the selected CL methods and the experiment setups can be found in Appendix B.2 and B.3. For CL training, the hyper-parameters of each CL method are selected on the validation sets of the first three tasks following Chaudhry et al. (2018). Every task in the CLDyB-seq is a 20-category classification problem, and the sequence length N is set to 10. All CL models are initialized with the ViT-Base-Sup21K (Dosovitskiy et al., 2021) pre-trained backbone. We conduct our experiments with multiple different random seeds and present the averaged outcomes. The first task is randomly chosen and consistently fixed for the same seed to ensure a fair starting point for comparison of different task selection strategies.

B.4 MEMORY FOOTPRINT CALCULATION

In the calculation of Memory Footprint, we account for the storage space used by each Learner in addition to the ViT Encoder. This includes model replicas (such as old models and models updated using EMA), PET modules, dataset statistics, and samples from previous tasks. All parameters are calculated as 32-bit floating point values, while stored images are treated as uncompressed 8-bit RGB images with a resolution of 224x224.

---

[4] Equivalent to negative **A**verage **F**orgetting **M**easure

C PSEUDOCODE FOR CLDYB-PIPELINE

We will publicly release our curated datasets, dynamic evaluation benchmark, and baseline implementations at https://PLACEHOLDER/FOR/GITHUB/URL/ upon acceptance of the paper.

Algorithm 1 `GreedyTaskSampling`$(\mathbb{D}, \mathbb{C}^{<t}, \mathbb{F}^{t-1}, K, J)$

```
Require: Data pool $\mathbb{D}$; Classes in previous CL tasks $\mathbb{C}^{<t}$ Continual learning models $\mathbb{F}^{t-1} := \{f_1^{t-1}, f_2^{t-1} ... f_M^{t-1}\}$; Num. task candidates to sample $J$; Num. classes in each task $K$

$\mathbb{C} \leftarrow \mathbb{D} \setminus \mathbb{C}^{<t}$    ▷ Filter out previously sampled classes in $\mathbb{D}$
$\Psi(\mathcal{C}_p, \mathcal{C}_q) = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{z}(\cos(\boldsymbol{\mu}_{m,p}^{t-1}, \boldsymbol{\mu}_{m,q}^{t-1})),\ \forall p, q \in \mathbb{C}$    ▷ Initialize class pair-wise potentials
$\mathbb{T}_{(g)}^t = \emptyset$    ▷ Initialize the candidate set by an empty set

for $j = 1, \ldots, J$ do
    for $k = 1, \ldots, K$ do
        if $k == 1$ then    ▷ Randomly sample the first task to ensure diversity
            $\mathcal{T}^t = \{\mathcal{C}_i\};\ i \sim [|\mathbb{C}|]$
        else    ▷ Greedily add class $i$ in $\mathcal{T}$ that maximizes the class-clique potential, Eqn. (3)
            $\mathcal{T}^t \leftarrow \mathcal{T}^t \cup \mathcal{C}_i;$  s.b. $\hat{i} = \arg\max_i \prod_{q \in \{1, \cdots, k-1\}} \Psi(\mathcal{C}_i, \mathcal{C}_q)$
        end if
    end for
    $\mathbb{T}_{(g)}^t \leftarrow \mathbb{T}_{(g)}^t \cup \mathcal{T}^t$    ▷ Add sampled task to the candidate set
end for

Return: $\mathbb{T}_{(g)}^t$
```

Algorithm 2 `FunctionalTaskClustering`$(\mathbb{T}_{(g)}^t, \mathbb{F}^{t-1}, J)$

```
Require: A sample task set $\mathbb{T}_{(g)}^t$; Continual learning models $\mathbb{F}^{t-1} := \{f_1^{t-1}, f_2^{t-1} ... f_M^{t-1}\}$; Num. of desired task candidates $J$, $J \leq |\mathbb{T}_{(g)}^t|$;

$\mathcal{G} \leftarrow \mathbf{0}^{|\mathbb{T}_{(g)}^t| \times M}$    ▷ Initialize task feature matrix
for $i \in [|\mathbb{T}_{(g)}^t|]$ do
    for $m \in [M]$ do    ▷ Evaluate the average NLL of samples in $\mathcal{T}^t$ under a KNN classifier build from the $m$-th CL model
        $\mathcal{G}[i, m] = -\frac{1}{|\mathcal{T}_i^t|}\sum_{x,y \in \mathcal{T}_i^t} \log p_m^{t-1}(y|x)$
    end for
end for

$\{\mathbb{T}_1^t, \mathbb{T}_2^t, ..., \mathbb{T}_K^t\} \leftarrow \texttt{KMeans}(\mathcal{G})$    ▷ split $\mathbb{T}_{(g)}^t$ into K clusters

$\mathbb{T}^t = \emptyset$
for $j = 1, 2, ..., J$ do    ▷ ancestral sampling without replacement from $\mathbb{T}_{(g)}^t$
    $\mathbb{T} \sim \{\mathbb{T}_1^t, \mathbb{T}_2^t, ..., \mathbb{T}_K^t\}$    ▷ uniformly sample one cluster
    $\mathcal{T} \sim \mathbb{T}$    ▷ uniformly sample one task from the cluster
    $\mathbb{T} \leftarrow \setminus \mathcal{T}, \mathbb{T}^t \leftarrow \mathbb{T}^t \cup \mathcal{T}$    ▷ remove the sampled task from the cluster to the candidate set
end for

Return: $\mathbb{T}^t$
```

Algorithm 3 CLDyB: Train and evaluate M continual learning algorithms on a sequence of N tasks selected dynamically by the proposed CLDyB-pipeline in Section 3 and Fig. 2

```
Require: Data pool $\mathbb{D}$; Pretrained Models $\mathbb{F} := \{f_1, f_2...f_M\}$; CL Algorithms $\mathbb{A} := \{\mathcal{A}_1, \mathcal{A}_2...\mathcal{A}_M\}$; Num. tasks for greedy task sampling $J_{(g)}$; Num. task candidates for MCTS $J$; Desired sequence length $N$

Initialize $f_m^0 = f_m, \forall m \in [M]$
Initialize $\mathbb{T}^{<1} = \emptyset$    ▷ the sampled CLDyB-seq

for $t = 1, 2, ..., N$ do
    $\mathbb{T}^t_{(g)} = $ GreedyTaskSampling($\mathbb{D}, \mathbb{C}^{<t}, \mathbb{F}^{t-1}, K, J_{(g)}$)    ▷ Sample $J_{(g)}$ difficult tasks, Alg. (1)

    $\mathbb{T}^t \leftarrow$ FunctionalTaskClustering($\mathbb{T}^t_{(g)}, \mathbb{F}^{t-1}, J$)    ▷ Prepare the search space for MCTS, Alg. (2)

    $\mathcal{T}^t \leftarrow$ MCTS(curr state $= \mathbb{F}^{t-1}$, action space $= \mathbb{T}^t$, simulation budget $= B$, value function $= \mathcal{Q}_{\mathcal{T}^t}(\mathbb{F}^{t-1})$)    ▷ Search for the optimal action on curr state that leads to the maximal value estimation i.e., $\mathcal{T}^t = \arg\max_{\tilde{\mathcal{T}} \in \mathbb{T}^t} \mathcal{Q}_{\tilde{\mathcal{T}}}(\mathbb{F}^{t-1})$) Eqn. (6)

    $\mathbb{F}^t = \{\mathcal{A}_m(f_m^{t-1}, \mathcal{T}^t)\}_{m=1}^M$    ▷ Train all $M$ CL models on the selected task i.e., standard continual learning
    $\mathbb{T}^{<t+1} = \mathbb{T}^{<t} \cup \mathcal{T}^t$    ▷ Add task to the sampled sequence
end for
Return: $\mathbb{T}^{<N+1}$    ▷ the discovered CLDyB-seq for CL evaluation
```

D ADDITIONAL RESULTS

D.1 CLDYB-SEQ WITH VARIOUS DIFFICULTY LEVELS

In the main text, while we have primarily focused on generating the most challenging CLDyB-seq by greedily selecting the most difficult task from the candidate pool that maximizes the search reward defined in Eqn. (6), it is possible to consider incorporating task sequences of various difficulty levels into the CLDyB-seq.

We propose a possible implementation to incorporate this feature into the CLDyB-pipeline. Instead of greedy selection, at each time step, we adopt a probabilistic sampling-based approach. The sampling probability of choosing a particular task is made proportional to the task's reward, $\mathcal{Q}_{\mathcal{T}^t}(\mathbb{F}^{t-1})$, that is

$$\mathcal{T}^t \overset{p(\tilde{\mathcal{T}}^t=\mathcal{T}^t)}{\sim} \mathbb{T}^t, \quad \text{where } p(\tilde{\mathcal{T}}^t = \mathcal{T}^t) := \frac{\exp(\mathcal{Q}_{\mathcal{T}^t}(\mathbb{F}^{t-1})/\tau)}{\sum_{\tilde{\mathcal{T}}^t \in \mathbb{T}^t} \exp(\mathcal{Q}_{\tilde{\mathcal{T}}^t}(\mathbb{F}^{t-1})/\tau)}. \tag{7}$$

To introduce varying difficulty levels, we employ an additional temperature scaling factor, τ in the sampling distribution, as a hyperparameter, to control the bias towards selecting the most challenging task. A temperature of $\tau = 0$ corresponds to the original greedy selection, while a higher temperature makes all candidate tasks more equally likely to be selected. All other components of the CLDyB-pipeline remain unchanged.

Based on this approach, we created two additional CLDyB-seqs at the 'medium' and 'easy' difficult levels, with the original CLDyB-seq serving as the 'hard' version for evaluation. We evaluate all CL methods on these three variations of the CLDyB-seq, the results are presented in Tab. 4. We observe that: (a) As expected, the performance of all CL methods improved as the difficulty decreased; (b) Performance on the hardest CLDyB-seq serve as a lower bound for each CL method; and (c) although there is some minor local shuffling in the relative rankings, the evaluation results for all three versions of CLDyB-seq still exhibit much higher correlation with the Holdout data compared to the standard benchmarks shown in Tab. 1. These results confirm that evaluation results and algorithmic developments made on our challenging CLDyB-seq are more likely to translate into strong real-world performance.

Table 4: Final average accuracy (%) of CL methods on our CLDyB-seq with difficulty levels at Hard, Medium and Easy. 'ns' and '$*$' respectively indicate statistical non-significance and statistical significance.

Method	**Final Average Accuracy**			
	Hard	**Medium**	**Easy**	**Heldout**
RanPAC (McDonnell et al., 2024)	56.9	62.2	77.0	81.0
HidePrompt (Wang et al., 2023a)	62.5	67.6	80.1	84.9
DualPrompt (Wang et al., 2022)	41.9	53.7	69.0	70.8
PGP (Qiao et al., 2024)	44.0	52.0	69.2	68.7
LAE (Gao et al., 2023b)	48.1	50.9	70.1	71.1
SLCA (Zhang et al., 2023)	56.2	59.7	77.2	80.3
ER (Rolnick et al., 2018)	54.8	60.5	76.3	79.9
CLSER Arani et al. (2022)	57.0	60.7	75.4	81.1
AFEC Wang et al. (2021)	49.5	51.9	72.5	76.3
Spearman's ρ (per column & Heldout)	0.983 $*$	0.833 $*$	0.867 $*$	N/A
Kendall's τ (per column & Heldout)	0.944 $*$	0.667 $*$	0.722 $*$	N/A

D.2 ADDITIONAL FIGURES OF CL EVALUATION RESULTS

Figures 8 to 13 provide additional results of CLDyB evaluations under varying conditions. Specifically:

- Fig. 8 provides evaluation results of CL methods on the standard, randomly ordered CL task sequences.
- Fig. 9 provides a performance comparison of the individual CL learner in Fig. 3 (CLDyB-seq) and Fig. 8 (random).
- Fig. 10 validates the effectiveness of the proposed CLDyB-pipeline in a long sequence CL setup.
- Fig. 11 validates the effectiveness of the proposed CLDyB-pipeline in challenging CL methods augmented with the CLIP-ViT-Base (Radford et al., 2021) which is a more powerful foundation model pre-trained on a much larger web-based dataset, FYCC-100M, compared to the ViT-Base-Sup21K (Dosovitskiy et al., 2021).
- Fig. 12 validates the effectiveness of the proposed CLDyB-pipeline in searching for commonly challenging CLDyB-seq for a subset of CL methods using a mixture of the ViT-Base-Sup21K (Dosovitskiy et al., 2021) and the CLIP-ViT-Base (Radford et al., 2021) pre-trained backbones.
- Fig. 13 provides a performance comparison of the individual CL learner in Fig. 12.
- Fig. 14 provides a comparison between the evaluation results of CL methods, using the CLIP-ViT-Base pre-trained backbone, on the CLDyB-seq discovered from the original and augmented CLDyB-pool with AI-generated data.

D.3 VISUALIZATION OF TASK SEQUENCES

Figures 15, 16, 17 and 18 show the t-SNE of tasks in CC and IC CLDyB, respectively. Meanwhile, Figures 19 and 20 are image examples of the searched task sequences for the experiment: *CLDyB-pool is expandable over time with AI-generated data* . We also provide some visualizations of the commonly challenging sequences discovered in Fig. 21, Fig. 22 and Fig. 23.

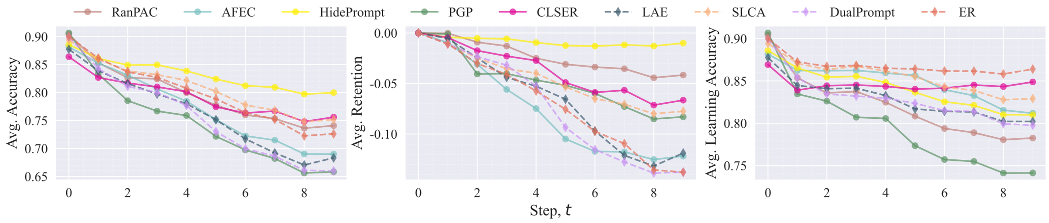

Figure 8: Performance of CL methods on the standard CL sequences constructed from randomly ordered tasks sampled unifromly from the CLDyB-pool.

E DISCUSSION OF LIMITATIONS

As the concept of dynamic benchmark is relatively new in the CL community and has not been widely explored, this paper focuses on class incremental CL for vision recognition, without extending to other machine learning tasks such as domain-incremental CL, natural language processing, and multimodal CL. However, we believe that the idea behind CLDyB can inspire the development of dynamic benchmarks for other machine learning tasks, and we plan to explore these directions in future work. On the other hand, the current CLDyB-pool we used requires manual updates for data pool expansion. We will develop it into an online project for the community, aiming to incorporate new data in a fully automatic fashion, as the community evolves, and hope that the CLDyB pool will also facilitate research on other computer vision tasks.

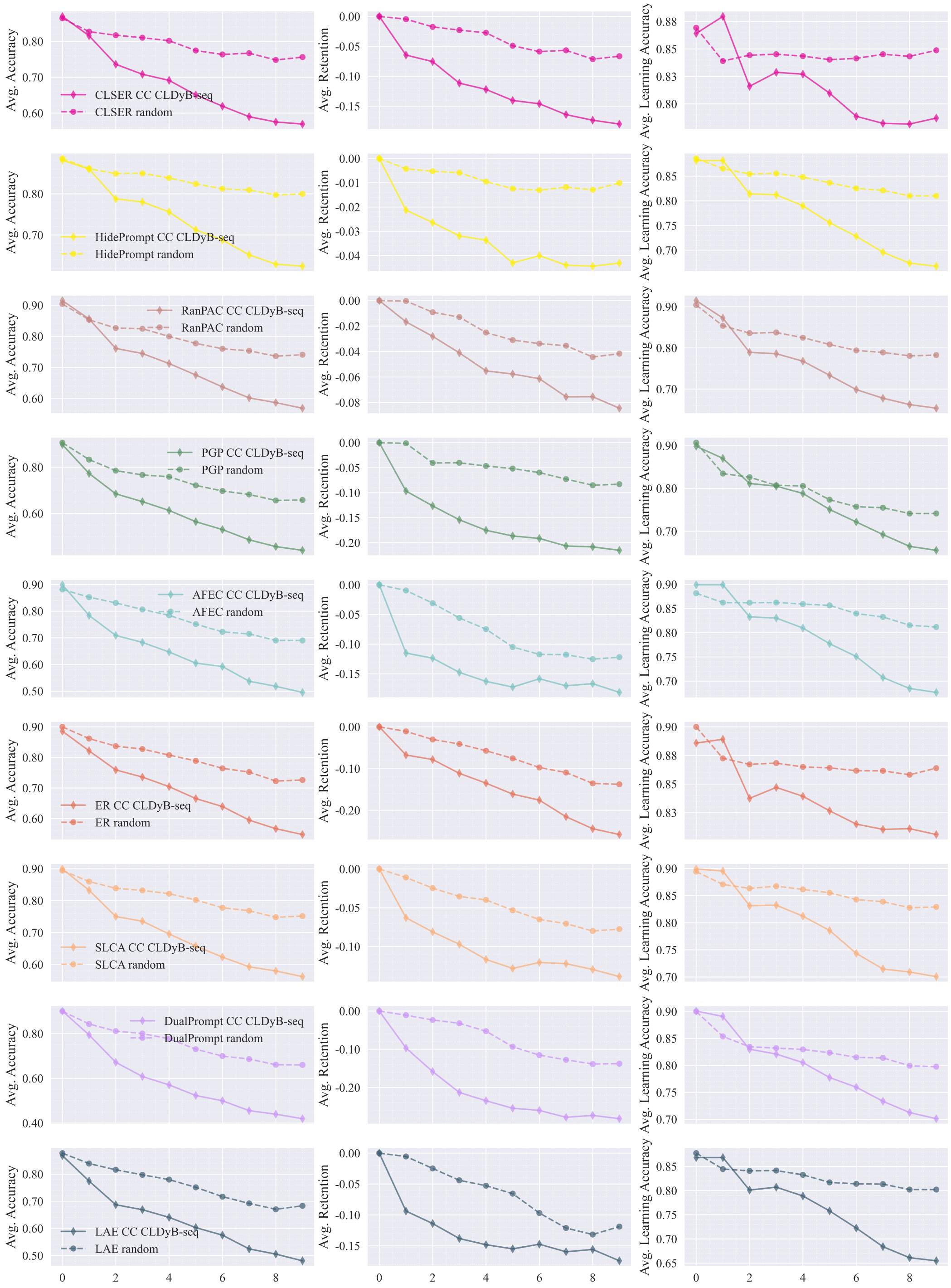

Figure 9: Performance comparison of individual CL methods on the commonly challenging CLDyB-seq obtained by the CLDyB-pipeline and standard sequences constructed from randomly ordered tasks. CLSER, HidePrompt, RanPAC, PGP and AFEC are used during searching while ER, SLCA, LAE and DualPrompt are reserved for evaluation only. All CL methods use the ViT-Base backbone (Dosovitskiy, 2020) supervised pre-trained on the ImageNet-21K.

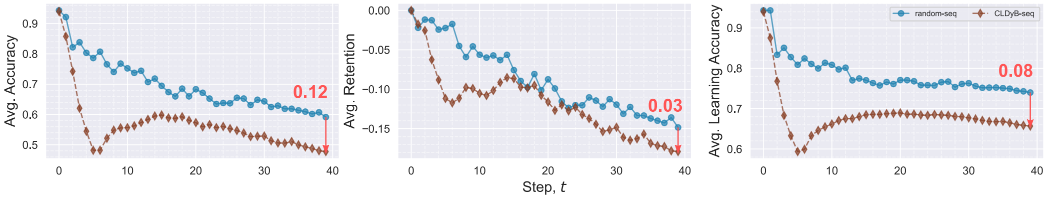

Figure 10: Average performance of CL methods on long task sequences constructed from the CLDyB-pipeline and randomly ordered tasks. Arrows indicate performance gap.

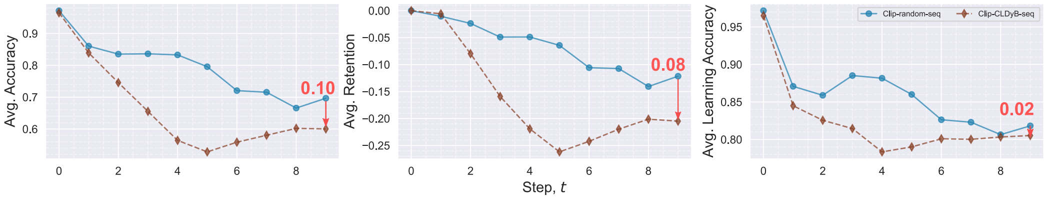

Figure 11: Average performance of the evaluated CL methods on commonly challenging CLDyB-seq obtained by running the CLDyB-pipeline on a subset of CL methods all using the CLIP-ViT-Base (Radford et al., 2021) pre-trained backbone. Arrows indicate CL performance gap to that on the standard sequences constructed from randomly ordered tasks.

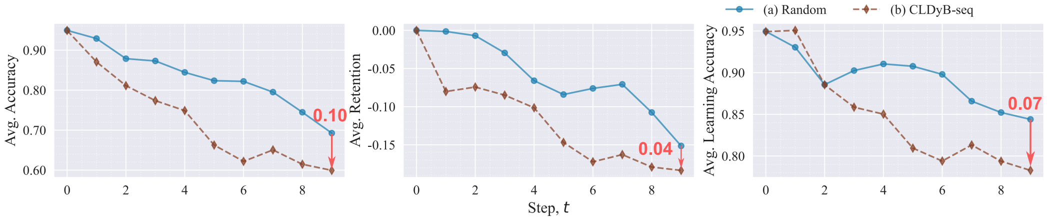

Figure 12: Average performance of the evaluated CL methods on commonly challenging CLDyB-seq discovered by running the CLDyB-pipeline on a subset of CL methods augmented with a mixture of the ViT-Base-Suo21K (Dosovitskiy et al., 2021) and the CLIP-ViT-Base (Radford et al., 2021) pre-trained backbones. Arrows indicate CL performance gap to that on the standard sequences constructed from randomly ordered tasks.

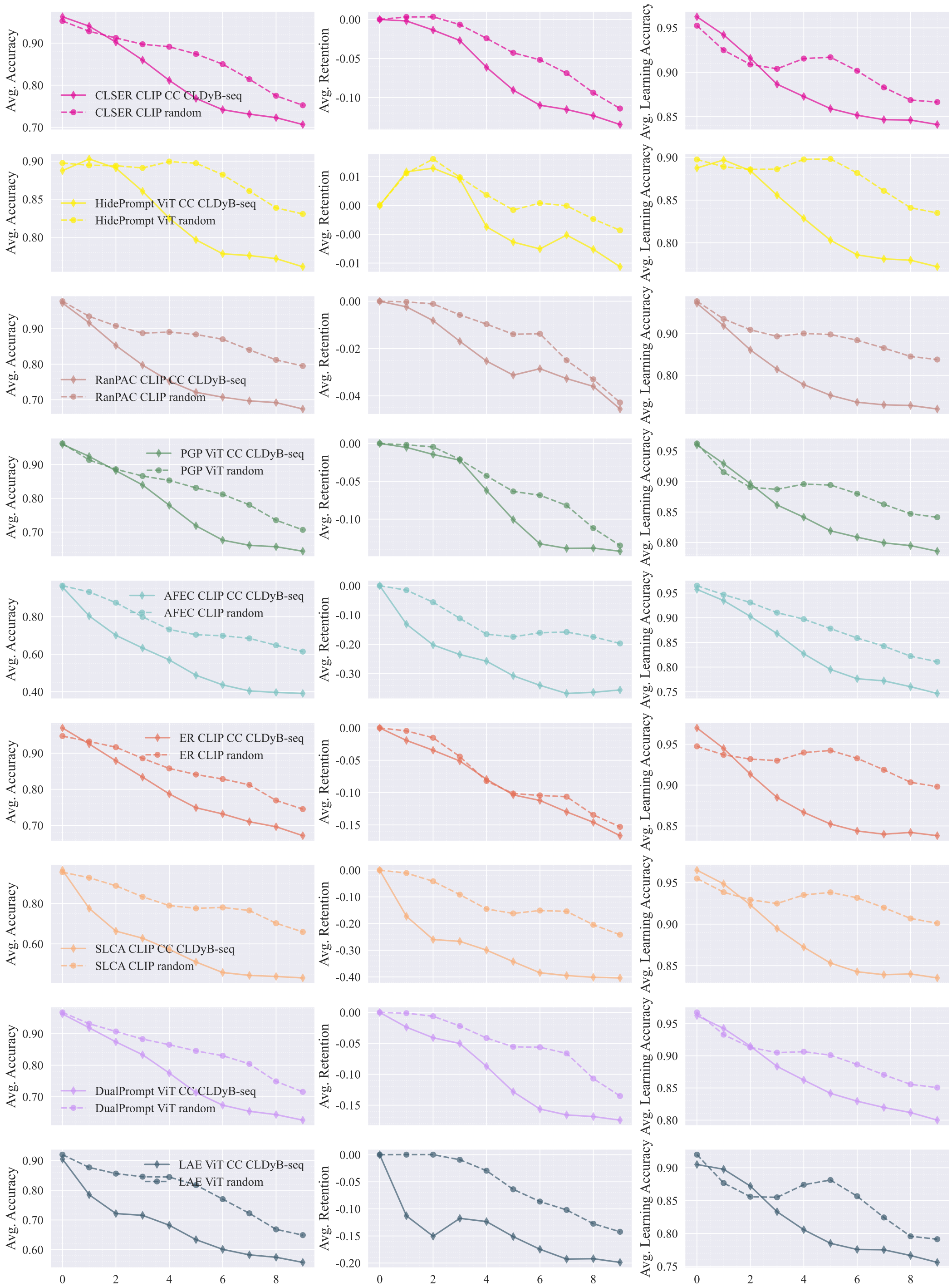

Figure 13: Performance comparison of individual CL methods on the commonly challenging CLDyB-seq obtained by the CLDyB-pipeline and standard sequences constructed from randomly ordered tasks. CLSER, HidePrompt, RanPAC, PGP and AFEC are used during searching while ER, SLCA, LAE and DualPrompt are reserved for evaluation only. Individual CL methods are augmented with either the ViT-Base-Suo21K (Dosovitskiy et al., 2021) or the CLIP-ViT-Base (Radford et al., 2021) pre-trained vision backbones as indicated by plot labels.

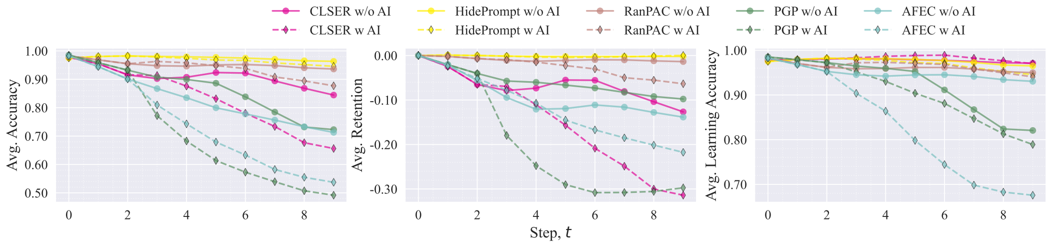

Figure 14: Performance of CL methods, using the CLIP-ViT-Base pre-trained backbone, on CC CLDyB-seq discovered from CLDyB-pool with (w AI) and without (w/o AI) additional diffusion-generated class images introduced after the third task.

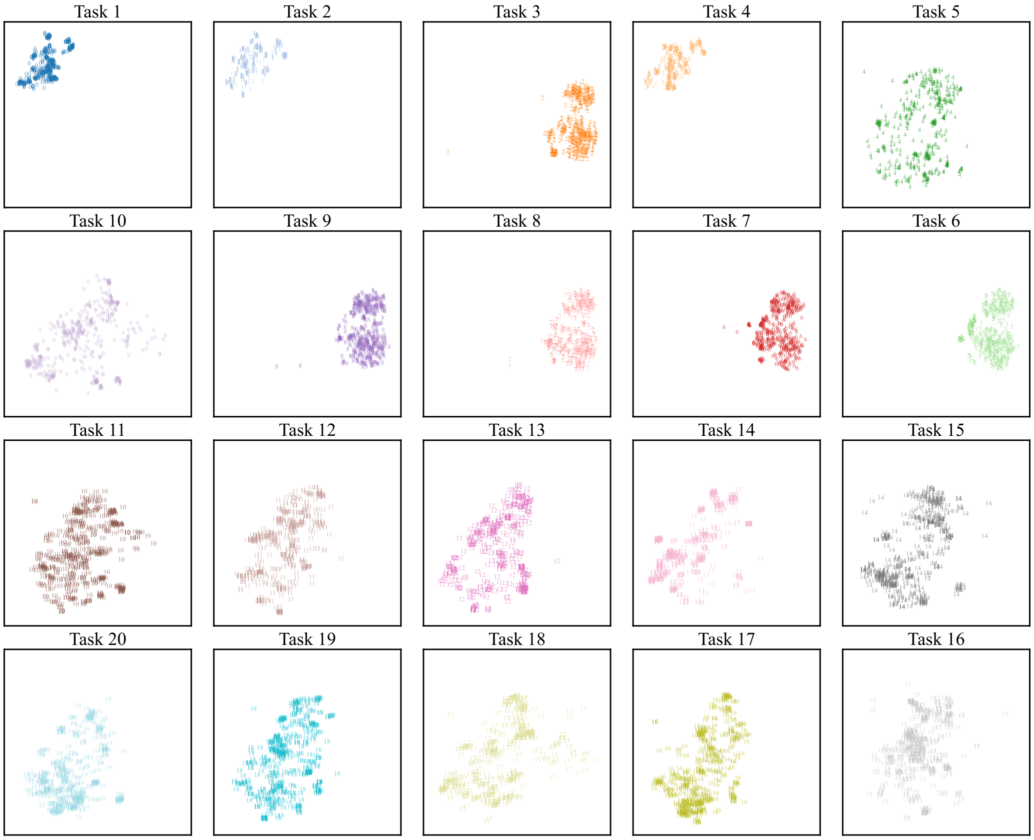

Figure 15: t-SNE of tasks in CC CLDyB-seq (seed 0).

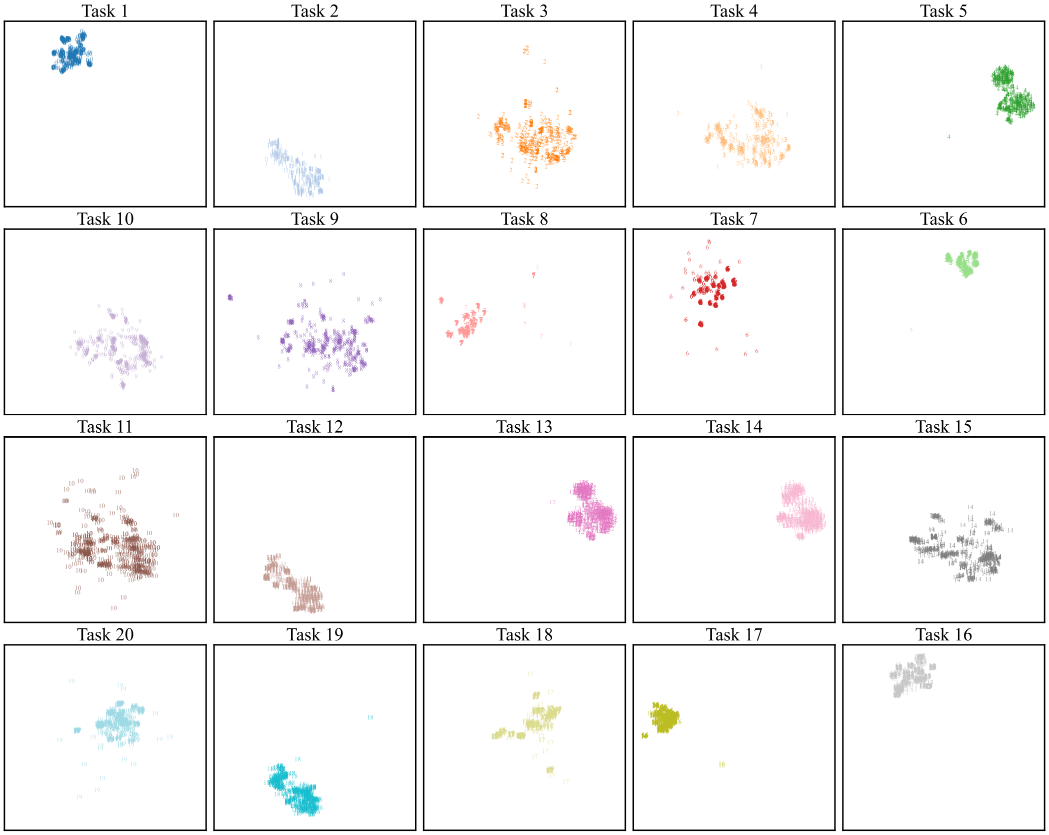

Figure 16: t-SNE of tasks in IC CLDyB-seq for LAE.

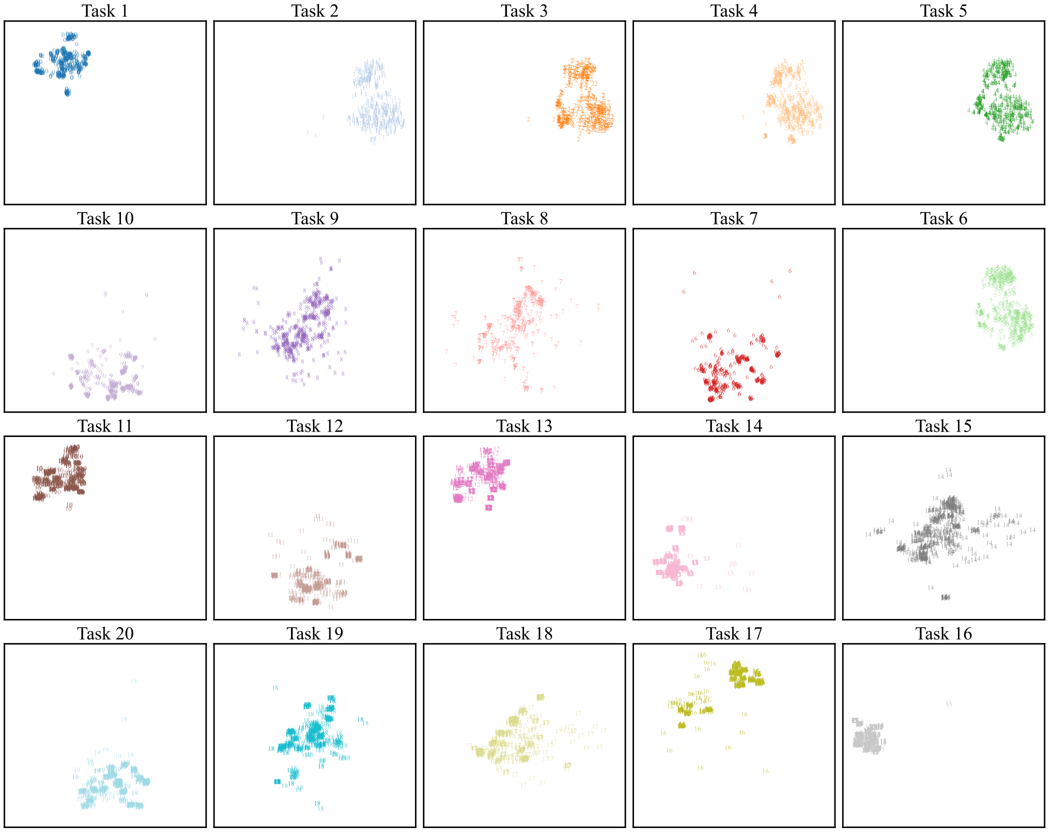

Figure 17: t-SNE of tasks in IC CLDyB-seq for ER.

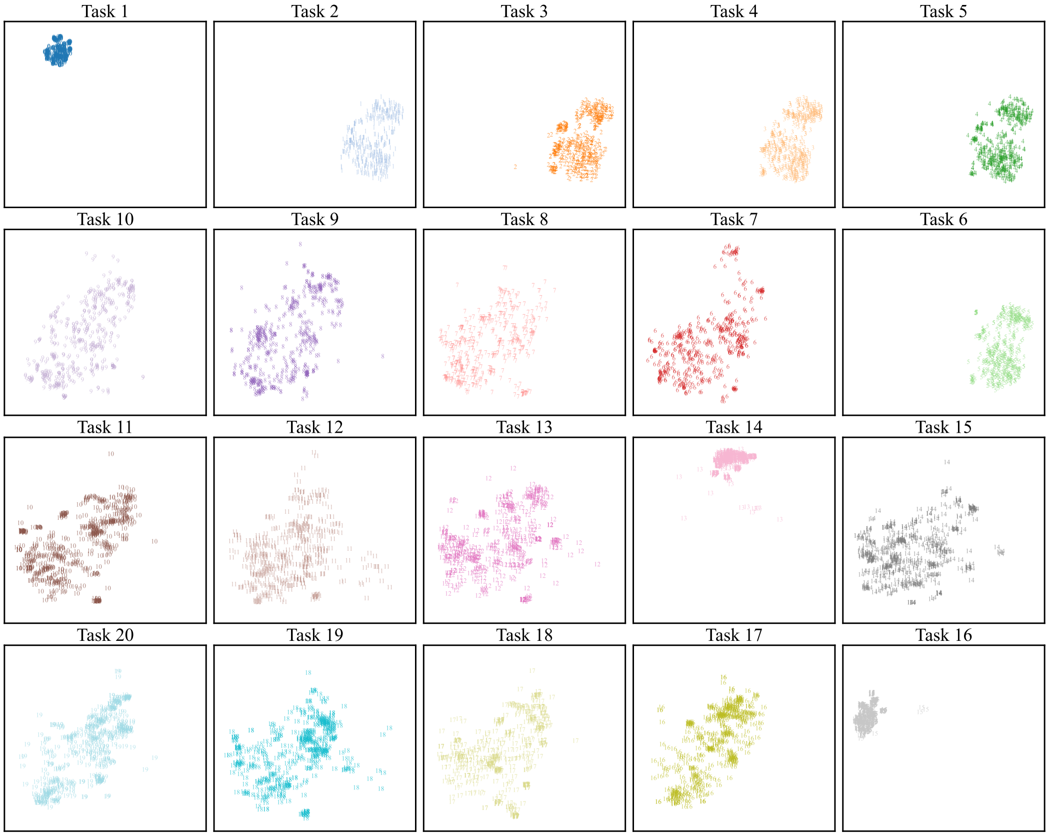

Figure 18: t-SNE of tasks in IC CLDyB-seq for RanPAC.

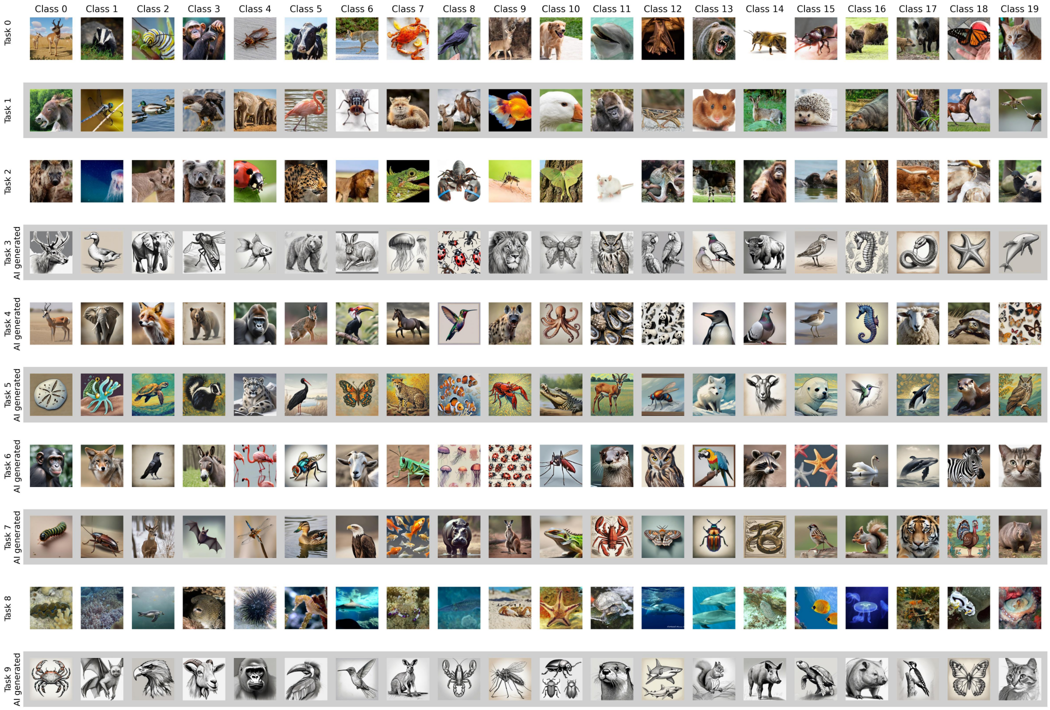

Figure 19: Visualization of task sequence discovered by the CLDyB-pipeline from augmented data pool with ai-generated data after time step $t = 2$. CL method used for task sequence searching: RanPAC, PGP, AFEC, CLSER, HidePrompt. Tasks consisting of AI-generated images are selected at steps $\{3, 4, 5, 6, 7, 9\}$

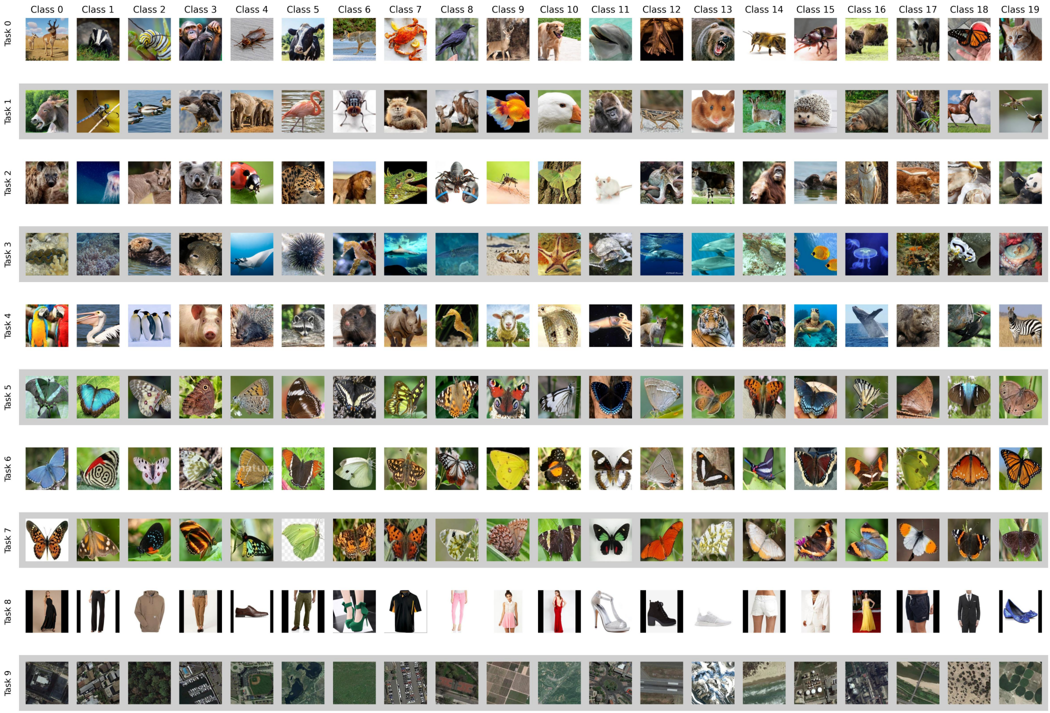

Figure 20: Visualization of task sequence using data pool without ai-generated data. CL method used for task sequence searching: RanPAC, PGP, AFEC, CLSER, HidePrompt.

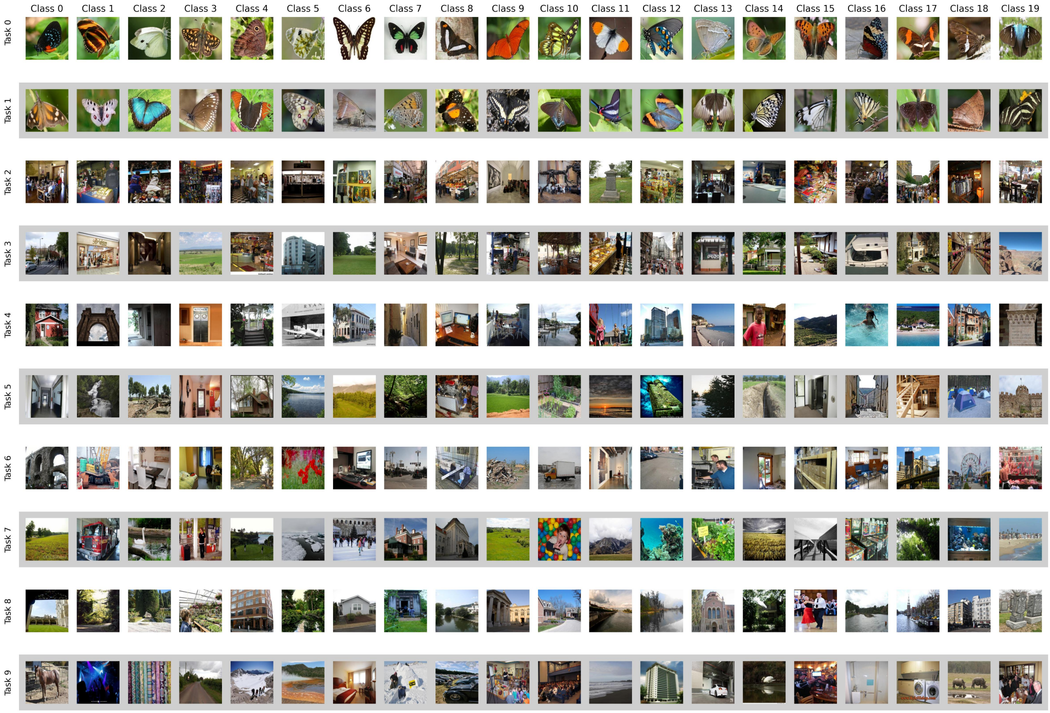

Figure 21: Visualization of a commonly challenging task sequence discovered by the proposed CLDyB-pipeline (seed-0).

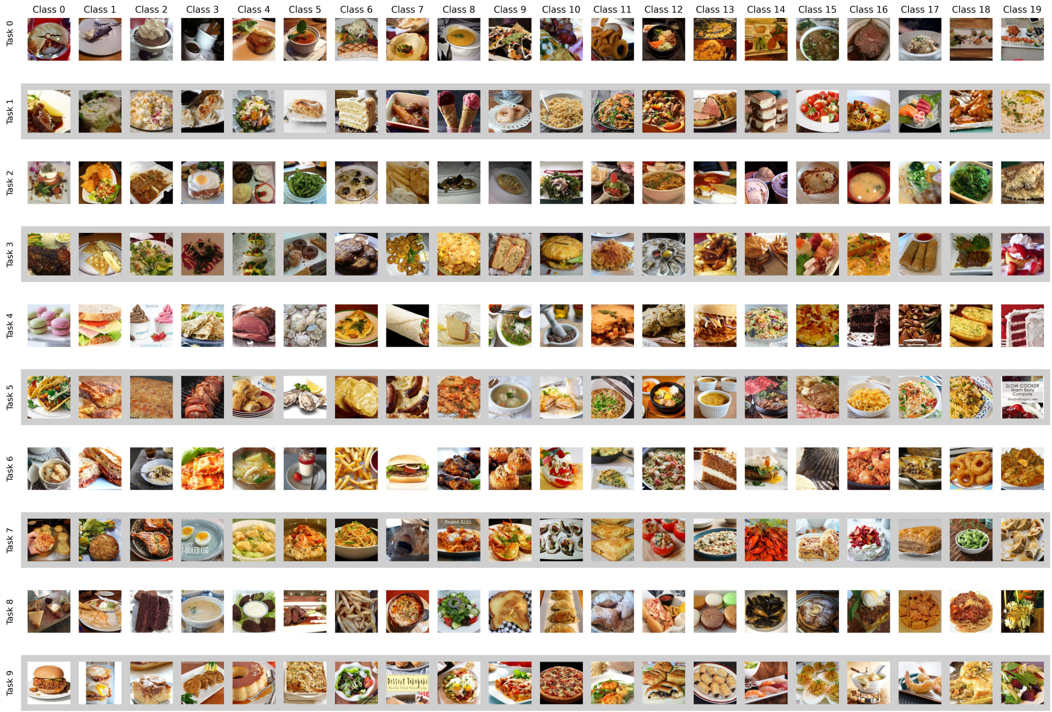

Figure 22: Visualization of a commonly challenging task sequence discovered by the proposed CLDyB-pipeline (seed-1)

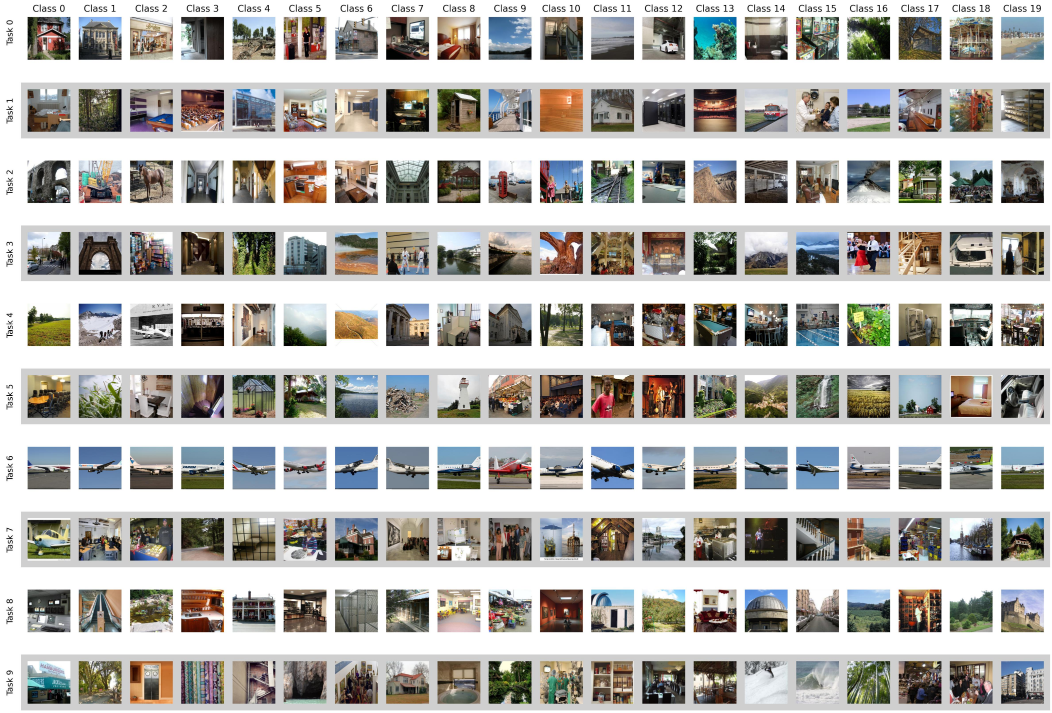

Figure 23: Visualization of a commonly challenging task sequence discovered by the proposed CLDyB-pipeline (seed-2)