Online Continual Learning on Hierarchical Label Expansion
Supplementary Material

Byung Hyun Lee[1,*], Okchul Jung[1,*], Jonghyun Choi[3,†] and Se Young Chun[1,2,†]
[1]Dept. of ECE, [2]INMC & IPAI, Seoul National University, Republic of Korea,
[3]Yonsei University, Republic of Korea
{ldlqudgus756,luckyjung96}@snu.ac.kr jc@yonsei.ac.kr sychun@snu.ac.kr

S1. Algorithms of Pseudo-Labeling based Flexible Memory Sampling (PL-FMS)

In this section, we offer a detailed explanation of the Pseudo-Labeling based Flexible Memory Sampling. The details for pseudo-labeling based memory management and flexible memory sampling can be found in Alg. 1 and Alg. 2 respectively. It is important to note that both algorithms are executed for every iteration, ensuring that the sampling process is constantly optimized. In Alg 2, when $\rho_t(y)$ is not equal to 1, then the sample from the stream data samples $\mathcal{S}_t$ is rejected and a sample is randomly selected from the set difference of the memory and the already selected memory samples. Overall, our Pseudo-Labeling based Flexible Memory Sampling approach is designed to improve the efficiency and effectiveness of the sampling process.

S2. Details on Dataset Configuration

S2.1. Single-Depth Datasets

ImageNet-Hier100 ImageNet-Hier100 is a subset of the ImageNet dataset [2] that is organized based on the taxonomy of WordNet [5]. The dataset is structured to represent 100 fine-grained classes by grouping them into 10 coarse-grained classes that capture the overall semantic structure. Each coarse-grained class consists of 10 corresponding fine-grained classes, which are subcategories that belong to the larger, upper-level group of the coarse-grained class. These 10 coarse-grained classes are referred to as superclasses and include carnivore, bird, arthropod, fruit, fish, ungulate, vehicle, clothing, furniture, and structure. The fine-grained classes, referred as subclasses, that correspond to each superclass are depicted in Figure S1.

S2.2. Multiple-Depth Datasets

CIFAR100 The CIFAR100 dataset is a widely used benchmark for image classification tasks, consisting of 60,000 32x32 color images organized into 100 fine-grained classes, with 600 images per class. The dataset is split into 50,000 training images and 10,000 testing images, making it ideal for evaluating different machine learning models. The images depict real-world objects such as animals, vehicles, and household items and are used for various computer vision tasks, including object recognition and image classification.

For the CIFAR100 dataset, we follow the hierarchical taxonomy as described in [3]. The dataset has five lev-

Algorithm 1 Pseudo-Labeling based Memory Management

```
Input levels of hierarchy $H$, feature extractor $\mathcal{F}$, classifiers $\{\mathcal{G}^k\}_{k=1}^H$, softmax function $\sigma$, memory $\mathcal{M}$, memory size $m$, label sets for hierarchy levels $\{\mathbb{Y}^k\}_{k=1}^H$, new sample $(x_{new}, y_{new})$, hierarchy level of the new sample $h$, per-sample importance measures for samples $\mathcal{H}$
if $|\mathcal{M}| < m$ then
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(x_{new}, y_{new})\}$
else
    $\bar{y} = \arg\max_y |\{(x_n, y_n) \in \mathcal{M} | y_n = y\}|$
    $\mathcal{M}_{\bar{y}} = \{(x_n, y_n) \in \mathcal{M} | y_n = \bar{y}\}$
    $\mathcal{I}_{\bar{y}} = \{j | (x_j, y_j) \in \mathcal{M}_{\bar{y}}\}$
    for $k = 1, 2, \cdots, H$ do
        if $k \neq h$ then
            $s = (0, 0, \cdots, 0) \in \mathbb{R}^{|\mathbb{Y}^k|}$
            for $(x, y) \in \mathcal{M}_{\bar{y}}$ do
                $\hat{i} = \arg\max_i \ \sigma(\mathcal{G}^k(\mathcal{F}(x)))_i$
                $s_{\hat{i}} \leftarrow s_{\hat{i}} + 1$
            end for
            $\hat{y}^k = \arg\max_{i \in \mathbb{Y}^k} \ s_i$
            $\mathcal{I}_{\bar{y}} \leftarrow \mathcal{I}_{\bar{y}} \cup \{j | (x_j, y_j) \in \mathcal{M}, y_j = \hat{y}^k\}$
        end if
    end for
    $\hat{j} = \arg\min_{j \in \mathcal{I}_{\bar{y}}} \mathcal{H}_j$
    $\mathcal{M} \leftarrow (\mathcal{M} \backslash \{(x_{\hat{j}}, y_{\hat{j}})\}) \cup \{(x_{new}, y_{new})\}$
end if
Output $\mathcal{M}$
```

[*] Equal contribution, † Corresponding authors.

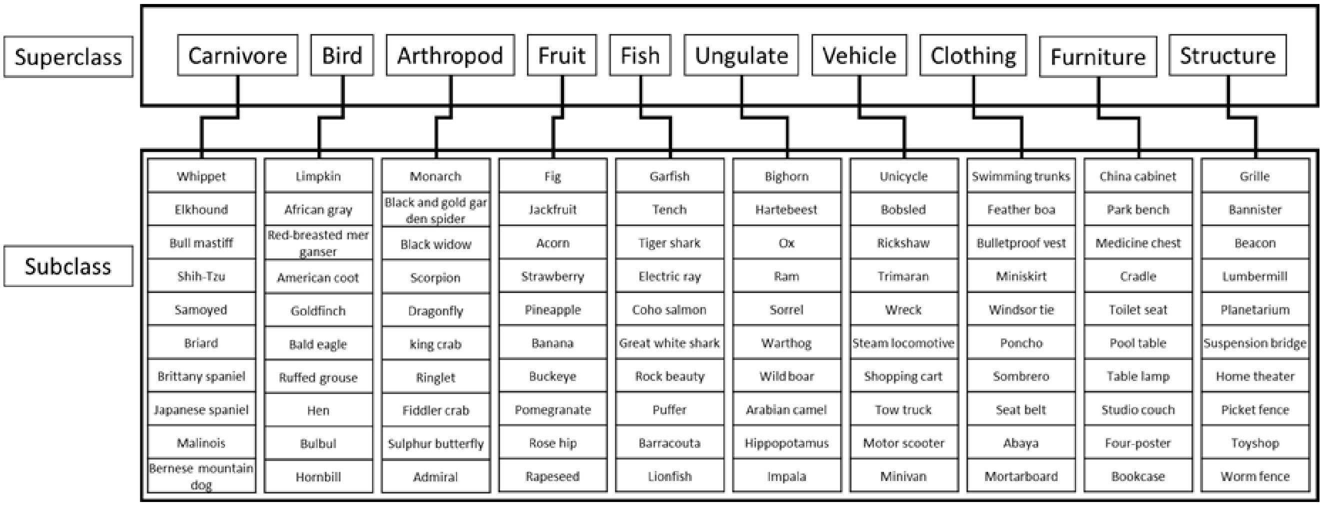

Figure S1: An overview of the ImageNet-Hier100 dataset, which is represented as a hierarchical structure. The dataset includes 10 broad categories or ”superclasses”, which are further divided into 10 more specific categories or ”subclasses”. The subcategories are visually depicted as branches stemming from the main categories, resulting in a total of 100 subclasses in the dataset.

Algorithm 2 Flexible Memory Sampling

```
Input memory $\mathcal{M}$, training iteration $t$, iterations encountering class $y$ at the first time $T_y$, normalizing factor $T$, stream data samples $\mathcal{S}_t$, memory data samples $\mathcal{N}_t$
$\mathcal{B}_t = \mathcal{N}_t$
for $(x, y) \in \mathcal{S}_t$ do
    $\rho_t(y) \sim \text{Bern}\left(\min\left(\frac{t-T_y}{T}, 1\right)\right)$
    if $\rho_t(y)$ is not 1 then
        $(x', y') \sim \mathcal{U}_{\mathcal{M}\setminus\mathcal{B}_t}$
        $\backslash\backslash$ $\mathcal{U}_A$ : uniform random sampler over a set $A$
        $\mathcal{B}_t \leftarrow \mathcal{B}_t \cup \{(x', y')\}$
    else
        $\mathcal{B}_t \leftarrow \mathcal{B}_t \cup \{(x, y)\}$
    end if
end for
Output $\mathcal{B}_t$
```

els of hierarchy with (2,4,8,20,100) classes, excluding the root node. The dataset has an Imbalance Ratio (IR) of 1, indicating a balanced distribution of classes. It has a total of 134 nodes and 100 leaves, denoting the total number of nodes and leaf nodes in the tree-shaped hierarchy, respectively. The Average Branching Factor (ABF) of the dataset is 3.8, representing the average number of children (subclasses) for each superclass. The average pairwise distance is 7.0, reflecting the average distance between each pair of classes in the hierarchy.

iNaturalist-19 The iNaturalist dataset is a large-scale image classification dataset of organisms, containing over 800,000 images from more than 8,000 different species. The iNaturalist-19 dataset, a subset of the larger iNaturalist dataset, was introduced for the 2019 CVPR Fine-Grained Visual Categorization Workshop. It includes metadata with hierarchical relationships between species, making it useful for evaluating methods for fine-grained visual categorization. The iNaturalist-19 dataset comprises 265,213 color images, organized into 1010 fine-grained classes.

However, the test set labels for the iNaturalist-19 dataset are not publicly available. To address this issue, we randomly selected and resampled three splits from the original training and validation data to create a new training, validation, and test set, as suggested in [1]. These sets were created using probabilities of 0.7, 0.15, and 0.15, respectively.

The iNaturalist-19 dataset has a hierarchical taxonomy with seven levels and (3, 4, 9, 34, 57, 72, 1010) classes, excluding the root node. It also has an Imbalance Ratio (IR) of 31, indicating a significant imbalance in the distribution of classes. The dataset has a total of 1,189 nodes and 1,010 leaves, denoting the total number of nodes and leaf nodes in the hierarchy, respectively. Its Average Branching Factor (ABF) is 6.6, representing the average number of children (subclasses) for each superclass. The average pairwise distance is 11, reflecting the average distance between each pair of classes in the hierarchy.

S3. Details on Implementation of Baseline Methods on HLE setup

ER and EWC++. ER and EWC++ uses reservoir sampling strategy for memory management by randomly removing samples in the memory to replace samples. We implemented the reservoir sampling in the hierarchical label

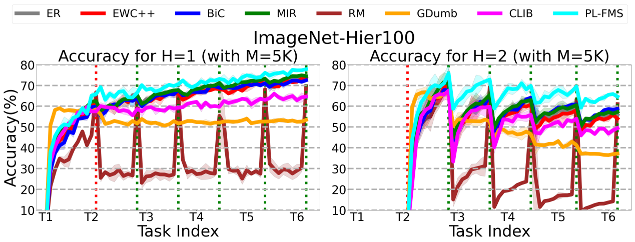

Figure S2: Any-time inference results on ImageNet-Hier100 for single-label scenario with single-depth hierarchy. H=1 is parent classes and H=2 child classes. Task index 1 receives parent class labeled data and subsequent indexes receive child class labeled data. Each data point shows average accuracy over three runs (± std. deviation).

expansion (HLE) setup so that it not only ignores the class information but also the hierarchical information when it randomly selects samples to remove from the memory.

BiC. BiC was originally proposed on the offline CL setup with herding selection [6]. However, herding selection is not applicable since entire task data is required for computing class mean, which is impossible on online CL setup. Therefore, we applied the reservoir sampling for BiC as used in [4]. BiC empirically demonstrates that the classifier is biased towards new classes and proposes a bias correction layer attached at the end of the classifier, which is trained with the separate validation set as a small part of the memory, to correct the classifier. Since there are multiple classifiers for each hierarchy in the HLE setup, we also used multiple bias correction layers for each corresponding classifier. In contrast, the validation set stores the samples for all encountered classes regardless of their hierarchy.

MIR. MIR enhances memory utilization by first drawing a subset of the memory whose cardinality is larger than that of the training batch, and then selecting samples from the subset that would experience the highest loss increase if trained with streamed data to update the model. To apply MIR in the HLE setup, we extracted samples independently for each hierarchy level and ensured that the ratio of the number of samples in the subset and the training batch matched for each level.

RM, GDumb, and CLIB RM, GDumb, and CLIB are originally managed to maintain the balance of the number of samples for each class in memory. Following their memory management schemes, we balance it regardless of hierarchy level in the HLE setup.

S4. Details on Evaluation Metrics

Any-time inference While average accuracy (A_{avg}) is a widely used measure in continual learning evaluation, it only provides a limited evaluation of a model's performance. A_{avg} measures performance only at task transitions, which typically occur only a few times during the learning process. Therefore, it may not provide a comprehensive evaluation of a model's ability to adapt to new tasks without forgetting previously learned ones.

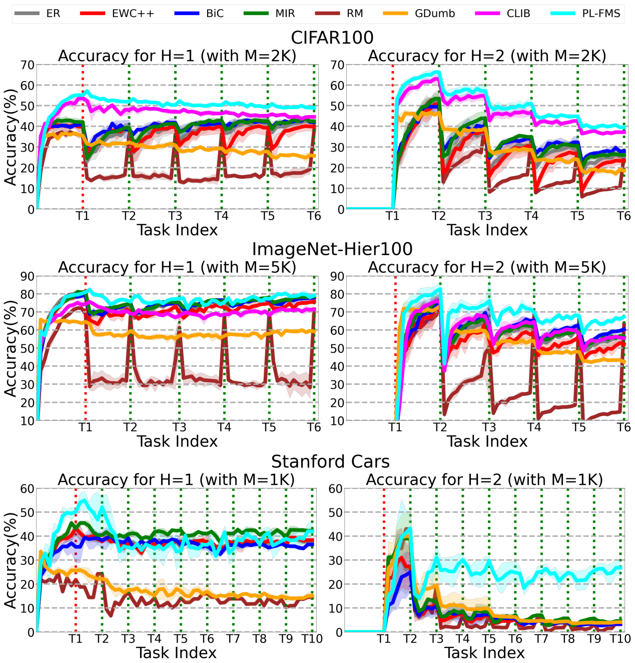

Figure S3: Any-time inference results on CIFAR100, ImageNet-Hier100, and Stanford Cars dataset for dual-label scenario with single-depth hierarchy. H=1 is parent classes and H=2 child classes. Task index 1 receives parent class labeled data and subsequent indexes receive child class labeled data. Each data point shows average accuracy over three runs (± std. deviation).

In contrast to average accuracy, any-time inference is a more appropriate and useful metric for evaluating continual learning models. Any-time inference measures a model's ability to make accurate predictions at any point during the learning process, without relying on explicit task boundaries. To measure any-time inference, we evaluate the model's accuracy after observing every Δn samples, instead of only at discrete task transitions by referring to [4]. This approach provides a more continuous and fine-grained evaluation of a model's performance, reflecting real-world scenarios where new tasks and data can arrive at any time, and the model needs to adapt quickly without sacrificing performance on previously learned tasks. Therefore, any-time inference is a more suitable metric for evaluating continual learning models, as it aligns with the practical requirements of real-world applications where machine learning models must continuously learn and adapt to new data over time.

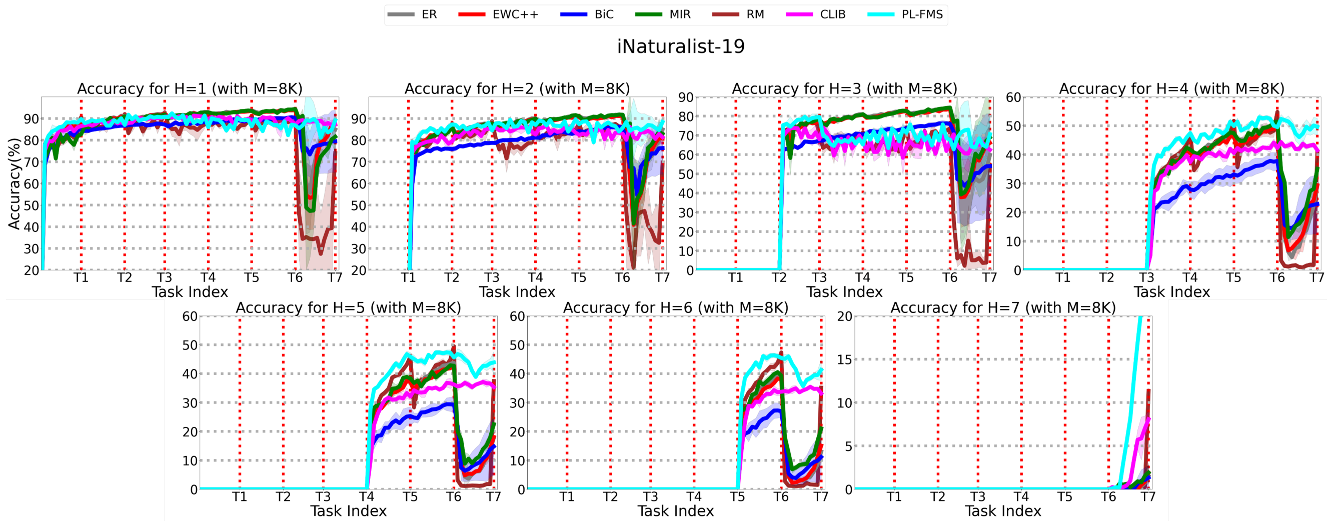

Figure S4: Any-time inference results on iNaturalist-19 dataset for multiple-depth hierarchy. H=1 represents the coarsest level and H=7 represents the finest level of class hierarchy. The dotted line represents the point at which the model is fully given the task data for the corresponding task index. The reported data points represent the average accuracy over three runs (± std. deviation)

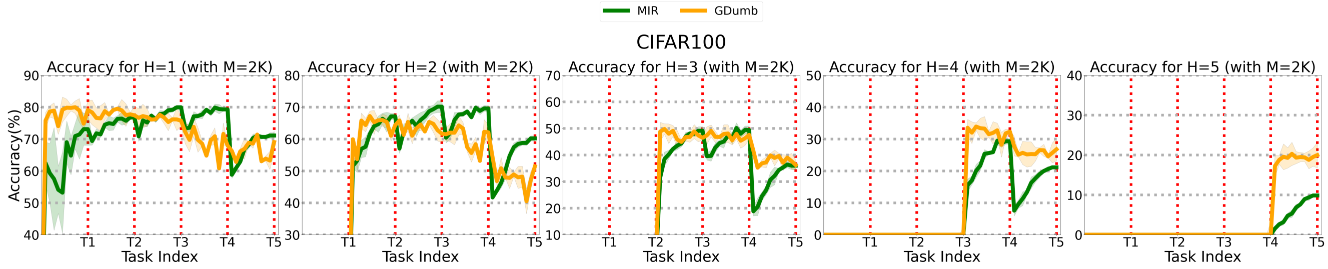

Figure S5: Any-time inference results on CIFAR100 dataset for multiple-depth hierarchy. H=1 represents the coarsest level and H=5 represents the finest level of class hierarchy. The dotted line represents the point at which the model is fully given the task data for the corresponding task index. The reported data points represent the average accuracy over three runs (± std. deviation)

S5. Anytime Inference on ImageNet-Hier100 for Single-Label Scenario

In Figure S2, we report the any-time inference result for ImageNet-Hier100 dataset for single-label scenario with single-depth hierarchy. The trend is similar to the result on CIFAR100 dataset for the single-label scenario in the main paper. It's worth noting that the performance of CLIB was relatively inferior to the methods that employ reservoir sampling, such as ER, EWC++, BiC, and MIR. This could be due to the fact that ImageNet-Hier100 has a longer interval of iterations for each task compared to CIFAR100, and CLIB's memory-only training is limited in its ability to adapt to newly encountered classes.

S6. Anytime Inference on iNaturalist-19 for Multi-Depth Scenario

Figure S4 shows the any-time inference results for multi-depth scenario on iNaturalist-19 dataset. As we observed from the any-time inference results for CIFAR100, the performance of the baseline methods except CLIB for the hierarchy levels from 1 to 6 deteriorate seriously at the end of task 6, where the number of class increases explosively from 179 to 1189 by label expansion to the most fine-grained classes. On the other hand, the performance of PL-FMS and CLIB demonstrates their mild forgetting at the end of the task 6 while the any-time inference results of CLIB for the hierarchy levels larger than 4 shows relatively lower performance compared to PL-FMS and RM. Until the end of task 6, EWC++ shows the highest performance for hierarchy level 1,2, and 3, but it exhibit severe catastrophic forgetting after the task 6. In overall, PL-FMS shows the best performance for all hierarchy levels except the hierarchy level 1 at the end of the training and consistently high performance in terms of any-time inference for all hierarchy levels. We chose not to conduct the GDumb method for the multiple-depth scenario due to its consistently low performance on both single-depth and multiple-depth datasets, and because it required a significant amount of training time. However, we did perform an additional experiment for the MIR baseline method to clarify the performance of all baseline methods except for GDumb. The MIR method demonstrated comparable performance against other baseline methods, which was the motivation for conducting this experiment.

S7. Anytime Inference for Dual-Label Scenario

In Figure S3, we report the any-time inference results for dual-label scenario with single-depth hierarchy on CIFAR100, ImageNet-Hier100, and Stanford Cars dataset. Since the model is trained with more samples and the labels from both hierarchy level 1 and 2 are assigned to same samples, the dual-label scenario showed higher performance compared to the results for the single-label scenario, except some baseline methods. Note that the performance for hierarchy level 1 in dual-label scenario can be more easily saturated in the first task due to the larger number of samples for each task. Because of this, PL-FMS showed the forgetting on CIFAR100 dataset in hierarchy level 1 through the subsequent tasks, while we didn't observe it in the single-label scenario. Furthermore, PL-FMS showed significantly higher performance on Stanford Cars dataset in hierarchy level 2 whereas baseline methods didn't show such dramatic improvement.

S8. Anytime Inference of MIR and GDumb on Mutli-Depth Scenarios

Figure S5 is the results of the any-time inference of MIR and GDumb for multi-depth scenario on CIFAR100 dataset. We can find that MIR shows similar performance to ER and EWC++. Also, GDumb exhibits similar trend that we found from the single-depth scenario, where the performance is maintained during the task since it is trained from the scratch whenever the model is tested. Because of that, as can be seen from the result of RM at the end of the last task, GDumb shows relatively higher performance for the highest hierarchy level compared to other baselines. This is due to the fact that both RM and GDumb train the model with samples in memory for multiple epochs, which is not realistic for task-free online continual learning.

References

[1] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *CVPR*, pages 12506–12515, 2020. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 1

[3] Vivien Sainte Fare Garnot and Loic Landrieu. Leveraging class hierarchies with metric-guided prototype learning. *arXiv preprint arXiv:2007.03047*, 2020. 1

[4] Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. In *ICLR*, 2022. 3

[5] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 1

[6] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 3