Lifelong Content-based Histopathology Image Retrieval via Bilevel Coreset Selection and Distance Consistency Rehearsal

Xinyu Zhu[a,b,c], Zhiguo Jiang[a,b,c], Kun Wu[a,b,c], Jun Shi[d], Yushan Zheng[a,*]

[a]*Beijing Advanced Innovation Center for Biomedical Engineering, School of Engineering Medicine, Beihang University, Beijing, 100191, China.*
[b]*Image Processing Center, School of Astronautics, Beihang University, Beijing, 100191, China.*
[c]*Tianmushan Laboratory, Hangzhou, 311115, China.*
[d]*School of Software, Hefei University of Technology, Hefei, 230601, China.*

Abstract

Content-based histopathological image retrieval (CBHIR) has shown strong performance on static databases by retrieving whole slide images (WSIs) with similar content to query images. However, in clinical settings, the rapid growth of WSI databases challenges current CBHIR methods, which either require costly retraining or suffer performance degradation on new data. To address this, we propose a Lifelong Content-based Histopathology Image Retrieval (LCBHIR) framework that mitigates catastrophic forgetting in continual retrieval, where models lose prior knowledge when updated on expanding databases. The central challenge is balancing stability and plasticity. To enhance plasticity, we design a local memory bank with bilevel coreset sampling, formulating instance selection as a two-level optimization problem. This assigns higher weights to informative or hard-to-learn samples, refining decision boundaries in the feature space. To preserve stability, we introduce a distance consistency rehearsal (DCR) module, which maintains the relative feature distances of old samples, ensuring consistency across retrieval tasks and improving reliability in clinical applications. We validate our method on a large-scale continual WSI dataset from TCGA projects, comprising approximately 7,400 WSIs across 6 primary sites and 19 cancer subtypes. The experimental results have demonstrated the proposed method is effective and is superior

*Corresponding author
Email address: yszheng@buaa.edu.cn (Yushan Zheng)

to the state-of-the-art methods, achieving 5.7∼19.4% higher mAP compared to existing continual learning methods. The code is available at https://github.com/OliverZXY/LCBHIR.
Keywords: Histopathology image analysis, CBIR, Continual learning, Digital pathology

1. Introduction

The past few decades have witnessed the rapid development in digital storage technology of high-resolution whole slide images (WSIs). As the clinical institutions can scan and store more WSIs, they seek to leverage the digital morphologic content of these images [1]. Hence, high-performance content-based histopathological image retrieval systems are emerging [2, 3, 4, 5], enabling the retrieval of WSIs similar in content to query images from a database.

The volume of medical diagnostic data is steadily increasing, posing significant challenges for the development of effective CBHIR systems. The challenge now lies in adapting to continuously expanding datasets, as traditional CBHIR methods trained on static databases cannot be directly applied to address this issue. A critical issue is the system's ability to update without losing prior knowledge. Although fine-tuning is a common approach to handling continuous data streams, it often results in catastrophic forgetting—a phenomenon where previously acquired knowledge deteriorates as new data is introduced [6, 7, 8, 9].

Continual learning (CL) has been introduced as a solution to mitigate catastrophic forgetting when learning from non-stationary data streams [10, 11]. Existing CL methods can be broadly categorized into three main approaches: replay methods, regularization-based methods, and parameter isolation methods [11]. Among these, replay methods have shown considerable promise by storing subsets of the data stream as exemplars for experience replay. In natural scenes, various replay methods have been developed to address catastrophic forgetting in downstream tasks such as classification [12, 13] and semantic segmentation [14, 15]. Replay methods developed for medical retrieval scenarios face heightened challenges, in contrast to those designed for classification and segmentation tasks: 1) Medical retrieval systems not only need high precision, but

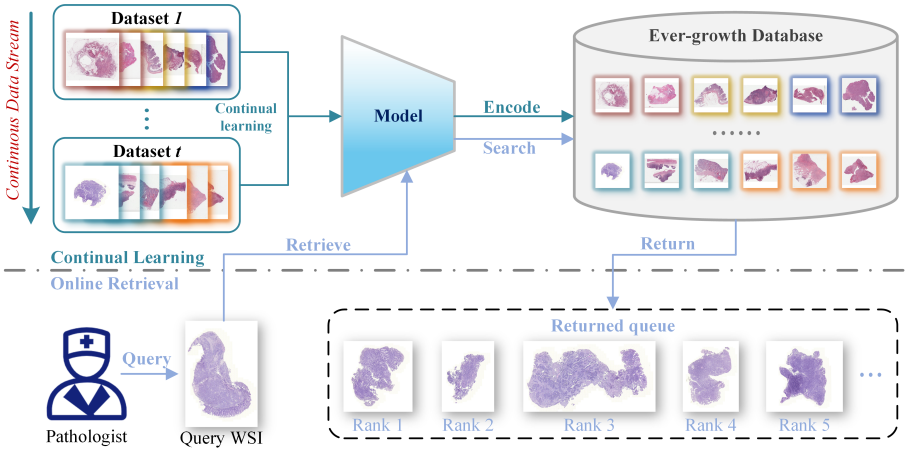


Figure 1: The universal process of continual learning and online retrieval for LCBHIR.

also require maintaining consistency in retrieval outputs for prior tasks after continual learning. This is particularly crucial in medical image retrieval systems, where clinicians may rely on consistent and accurate results for diagnosis. Any variability in retrieved results over time could compromise the system's reliability, potentially leading to diagnostic errors. 2) The design of the memory bank requires equilibrium between each category. It not only need to preserve representative samples of each category but also have to ensure diversity and balance within each class, thereby identifying challenging samples essential for continual learning.

In this paper, we propose a novel continual whole slide retrieval framework named lifelong content-based histopathology image retrieval (LCBHIR). A major dilemma in continual learning is how to achieve trade-off between learning plasticity and memory stability, where an excess of the former interferes with the latter, and vice versa [16]. We define the retrieval queue's consistency for previous tasks as stability in a lifelong CBHIR system and introduce a Distance Consistency Rehearsal (DCR) module, proven effective in enhancing retrieval performance and queue stability after continuous learning. Additionally, a bilevel optimization-based sampling method [17] is employed to maintain feature space diversity of memory bank, ensuring the model balances focus on both current and previous tasks, thereby supporting learning plasticity. By integrating these two strategies, LCBHIR achieves a favorable balance between stability

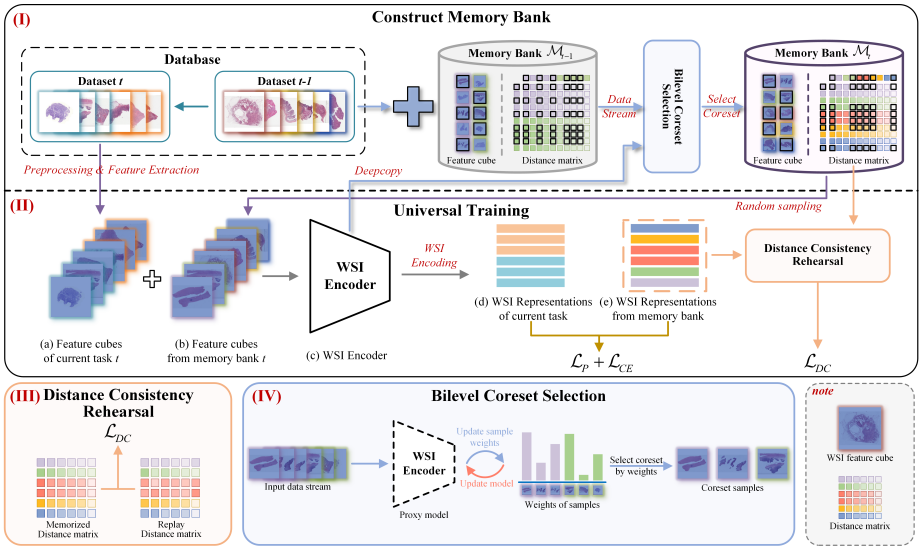


Figure 2: The overview of the proposed lifelong content-based histopathology image retrieval (LCBHIR) framework, where (I) shows the process of constructing memory bank before a new task, (II) describes universal training process of every task, (III) illutrates the proposed distance consistency rehearsal (DCR) module that is detailed in section 4.3 and algorithm 1, and (IV) is the bilevel sampling method for buffer sampling and updating, which is elaborated in section 4.4 and algorithm 2.

and plasticity, leading to superior retrieval performance. The proposed framework was evaluated on a large public TCGA continual dataset, encompassing six primary sites: Brain, Urinary, Gastrointestinal, Pulmonary, Gynecology, and Breast, containing 7347 WSIs in total. Experimental results demonstrate that LCBHIR is highly effective in continual WSI retrieval and outperforms typical classification-oriented continual learning approaches.

The contribution of the paper can be summarized in three aspects.

(1) We propose a novel lifelong content-based histopathology image retrieval (LCBHIR) framework to address the challenge of continual learning in histopathology image retrieval by appropriately balancing learning plasticity and memory stability. To our knowledge, this is the first approach aimed at solving the continual learning problem within the domain of histopathology image retrieval.

(2) A novel distance consistency rehearsal (DCR) module is designed to ensure consistency in the retrieval queues for previous tasks by imposing constraints on the

relative distance matrix. This approach enhances the stability of a lifelong CBHIR system. Visualization results demonstrate that the proposed DCR module not only delivers highly relevant retrieval results but also maintains consistency with the results from previous tasks.

(3) A bilevel optimization-based sampling module is employed to perform sampling based on the importance weights of each instance within the continual learning process. By framing the sampling process as an optimization problem, the module effectively assigns weights to instances, enabling the identification of challenging samples. This approach enhances the plasticity of the lifelong CBHIR system.

2. Related Works

2.1. Content-based Histopathology Image Retrieval

Content-based histopathology image retrieval (CBHIR) has thrived with the advancements in modern computing technology, which is capable of returning morphologically similar results in content to query images from a database [18, 19, 20]. Several studies on histopathological image retrieval have focused on the analysis of patches within WSIs. Ma et al. [21] utilized histogram based feature to describe textural characteristics of patches for effective retrieval. Shi et al. [22] developed a deep learning framework for extracting binary features, utilizing a matrix-based loss function designed to preserve intra-class similarity and maximize inter-class differentiation. Leveraging these binary features, similar patches can be efficiently identified and retrieved using the Hamming distance as the metric. Recently, WSI-level CBHIR methods have emerged, treating the entire WSI as the query for retrieving similar images from a database. A self-supervised-based framework is introduced by Chen et al. [1], which was capable of extracting compact WSI representations and utilized a tree data structure to achieve fast retrieval. The technique employs a ranking strategy based on uncertainty to boost both search efficiency and accuracy. Hu et al. [23] integrated WSIs with corresponding diagnostic reports to jointly enhance the model's ability to learn fine-grained semantics, pioneering the first cross-modal retrieval system in the domain of CBHIR.

However, the aforementioned studies were developed based on static databases and are not well-suited for handling continuously expanding datasets. Consequently, as the volume of data increases, it becomes essential to integrate continual learning techniques into CBHIR to establish a lifelong learning system.

2.2. Continual Learning

The increasing number of data and knowledge poses challenge to today's artificial intelligence (AI) systems. We expect them could continually acquire, update, accumulate and exploit knowledge as humans do, which motivates the study of continual learning [16]. Recently, community proposes several solutions to mitigate catastrophic forgetting, broadly categorized into three groups [11] or five groups [16] if more refined. Three common methods include replay methods, regularization-based methods, and parameter isolation methods.

Replay methods, usually storing and revisiting past examples to prevent forgetting, exhibit powerful potential to alleviate catastrophic forgetting [24, 25, 26, 27]. Caccia et al. [24] integrated experience replay with active classification to alleviate catastrophic forgetting. However, this may face challenges with significant data distribution shifts and introduces computational overhead due to the need for careful sample selection. Another key challenge for replay methods is effectively sampling rehearsal instances for previous tasks. While many approaches use reservoir random sampling [24, 25, 26, 27], not all training instances are equally beneficial, as some may even degrade performance on prior tasks. Recently, Borsos et al. [28] formulated the coreset selection as a cardinality-constrained bilevel optimization problem. Furthermore, Liu et al. [17] tried to learn a probability distribution in a low-dimensional manifold, instead of directly learning the binary masks for each sample, which exhibits promising performance in coreset selection domain.

For replay methods in histopathology continual learning, Huang et al. [9] succeeded exploring continual learning on WSI classification tasks. They proposed Breakup-Reorganize (BuRo) for sampling and data augmentation, along with Cross-Scale Similarity Learning (CSSL) to achieve encouraging outcomes on continual WSI analysis. However, the exploration of continuous retrieval frameworks in CBHIR remains unex-

plored. Besides, current replay methods whether in natural scene or in digital pathology fail to take returned queues' consistency of old tasks after learning new ones into consideration, which we think is the most significant function a lifelong retrieval system should have, especially in clinical applications. This paper tries to apply continual learning in CBHIR scenario. A part of this work has been presented in the conference paper [27].

3. Preliminaries

3.1. Problem Definition

Continual learning for CBHIR can be defined as training a model on non-stationary data when new data added into the WSI database. As shown in Fig. 2(I), we define the database as a sequence of datasets $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$, where $\mathcal{D}_t = \{x_i, y_i\}_{i=1}^{N_t}$ is the dataset of task t, and T is the total number of tasks. Dataset $\mathcal{D}_t$ contains N_t labeled WSIs, and y_i is the class label of WSI x_i. As illustrated in Fig. 1 and 2, after class-incremental learning, the model needs to return relative WSIs sequence of given query WSIs from both old tasks and new tasks.

3.2. Data Preparation

For an original WSI, we first divided it into several non-overlapping patches with size of 256×256 under $20\times$ lens. Then, we utilize pre-trained encoder PLIP [29] to extract patch features, for its excellent performance in understanding image semantic information. Afterwards, a feature cube of a WSI can be represented as $x \in \mathbb{R}^{n_p \times d_f}$, where d_f is the dimension of the feature and n_p is the number of patches in the WSI. To alleviate storage pressure and facilitate retrieval, we save feature cubes in the memory bank instead of the original WSIs.

4. Method

4.1. Overview

The proposed continual WSI retrieval framework is shown in Fig. 2. For each task, excluding the initial one, mini-batches of feature cubes are sampled from both

current dataset and memory bank, then fed into WSI encoder to get representations, as depicted in Fig. 2(a-c). Subsequently, we calculate retrieval related loss and execute distance consistency rehearsal to facilitate continual learning, described as Fig. 2(d-e, III). At the end of current task, bilevel optimization is employed to select coreset from current dataset and current memory bank. The selected coreset is then saved into a new memory bank, as demonstrated in Fig. 2(IV).

Algorithm 1: Distance Consistency Rehearsal.

Input: current encoder f_t, previous encoder f_{t-1}, current dataset $\mathcal{D}_t$, previous dataset $\mathcal{D}_{t-1}$, current memory bank $\mathcal{M}_t$ and previous memory bank $\mathcal{M}_{t-1}$

`// Target distance matrix construction`

$\mathcal{M}_t \leftarrow \mathbf{X}_{t-1}, \mathbf{X}_{t-1}^m \leftarrow$ Bilevel Coreset Selection$(\mathcal{D}_{t-1}, \mathcal{M}_{t-1})$

$\mathbf{F}_{t-1}, \mathbf{F}_{t-1}^m \leftarrow f_{t-1}\left(\mathbf{X}_{t-1}\right), f_{t-1}\left(\mathbf{X}_{t-1}^m\right)$

$\mathbf{C}_{t-1} \leftarrow \text{Concat}\left(\mathbf{F}_{t-1}, \mathbf{F}_{t-1}^m\right)$

for $\mathbf{c}_{t-1}^i, \mathbf{c}_{t-1}^j$ *in* $\mathbf{C}_{t-1}$ **do**

  $\mathbf{D}_{t-1}[i,j] \leftarrow d\left(\mathbf{c}_{t-1}^i, \mathbf{c}_{t-1}^j\right) = \|\mathbf{c}_{t-1}^i - \mathbf{c}_{t-1}^j\|_2$

end

`// Distance consistency rehearsal for current task`

for *each mini-batch* $\mathbf{X}_t$ *in* $\mathcal{D}_t$ **do**

  `//` $\mathbf{d}_{t-1}$ `is the the sub-matrix sampled from` $\mathbf{D}_{t-1}$ `according to the indexes of` $\mathbf{X}_t^m$

  $\mathbf{X}_t^m, \mathbf{d}_{t-1} \leftarrow$ Random Sampling$(\mathcal{M}_t)$

  $\mathbf{F}_t, \mathbf{F}_t^m \leftarrow f_t\left(\mathbf{X}_t\right), f_t\left(\mathbf{X}_t^m\right)$

  for $\mathbf{f}_t^i, \mathbf{f}_t^j$ *in* $\mathbf{F}_t^m$ **do**

    $\mathbf{d}_t[i,j] \leftarrow d\left(\mathbf{f}_t^i, \mathbf{f}_t^j\right) = \|\mathbf{f}_t^i - \mathbf{f}_t^j\|_2$

  end

  $\mathcal{L}_{DC}\left(\mathbf{d}_t, \mathbf{d}_{t-1}\right) = \|\mathbf{d}_t - \mathbf{d}_{t-1}\|_F^2$

end

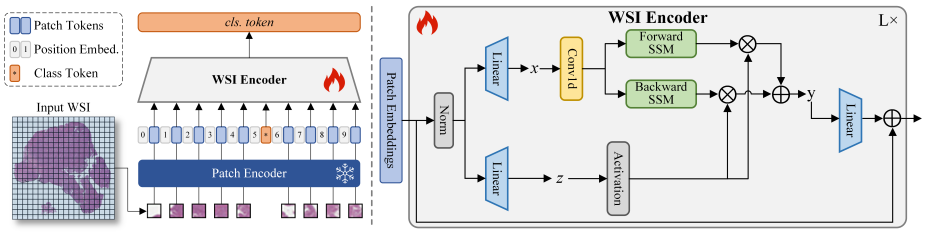


Figure 3: The detailed architecture of WSI Encoder. We employed Vision Mamba (vim) [30] as the slide-level encoder and used the output class token to execute downstream retrieval.

4.2. Model Architecture

We employ the Vision Mamba (Vim) [30] as the slide aggregator, the architecture is elaborate in Fig. 3. For each WSI, we first split it into N non-overlapping patches and input then into the frozen patch encoder to get feature sequence $\{\mathbf{p_0}, \mathbf{p_1}, \mathbf{p_i}, \cdots, \mathbf{p_{N-1}}\} \in \mathbb{R}^{N \times E}$, where E is the embedding dimension of patch features.

Next, we linearly project the p_i to the vector with size D by learnable projection matrix $\mathbf{W} \in \mathbb{R}^{E \times D}$ and add position embeddings $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$, as follows:

$$\mathbf{z}_0 = [\mathbf{p_0 W}; \mathbf{p_1 W}; \cdots; \mathbf{p_{cls}}; \cdots; \mathbf{p_{N-1} W}] + \mathbf{E}_{pos} \tag{1}$$

We then send the token sequence ($\mathbf{z}_{l-1}$) to the l-th layer of the **Vim** encoder, and get the output $\mathbf{z}_l$. Finally, we normalize the output class token $\mathbf{z}_L^0$ to get the final slide-level representation $\mathbf{F}$, as follows:

$$\begin{aligned} \mathbf{z}_l &= \mathbf{Vim}(\mathbf{z}_{l-1}) + \mathbf{z}_{l-1}, \quad l = 1 \ldots L, \\ \mathbf{F} &= \mathbf{Norm}(\mathbf{z}_L^0) \end{aligned} \tag{2}$$

where **Vim** is vision mamba block, L is the number of layers, and **Norm** is the normalization layer. For detailed architecture information about **Vim** block, please refer to Fig. 3 and Appendix A. The source code of this thesis is now available.[1]

4.3. Distance Consistency Rehearsal

As illustrated in section 4.1, we design distance consistency rehearsal (DCR) module to maintain stability of result queues for old tasks, which is detailed in Fig. 2(III) and illustrated in algorithm 1.

In practical scenarios, the stability of the CBHIR system is demonstrated by the consistency of the result queues for old tasks. In terms of feature space, this stability is reflected by maintaining constant distances between instances of old tasks even after learning new tasks. Based on this prior, we propose the distance consistency rehearsal (DCR), which is achieved by minimizing the differential value of distance between

[1] https://github.com/OliverZXY/LCBHIR

representations of current replay samples and distance matrix of corresponding replay samples saved in memory bank. The rehearsal process is elaborated as follows.

First, we construct the target distance matrix of instances saved in memory bank. At the ending of task $t-1$, we execute bilevel coreset selection (BCS) algorithm which is detailed in section 4.4 on dataset $\mathcal{D}_{t-1}$ and memory bank $\mathcal{M}_{t-1}$ to get sampled feature cubes $\mathbf{X}_{t-1} \in \mathbb{R}^{n_{t-1} \times n_p \times d_f}$ and $\mathbf{X}^m_{t-1} \in \mathbb{R}^{n^m_{t-1} \times n_p \times d_f}$ for the memory bank $\mathcal{M}_t$ of task t, where n_{t-1} and n^m_{t-1} means sampled number of $\mathcal{D}_{t-1}$ and $\mathcal{M}_{t-1}$ respectively. Then, sampled feature cubes is fed into task $t-1$ encoder f_{t-1} to get feature representations $\mathbf{F}_{t-1} \in \mathbb{R}^{n_{t-1} \times d_F}$ and $\mathbf{F}^m_{t-1} \in \mathbb{R}^{n^m_{t-1} \times d_F}$, where d_F is the dimension of the whole slide representation. With combination of $\mathbf{F}_{t-1}$ and $\mathbf{F}^m_{t-1}$, denoted as $\mathbf{C}_{t-1} \in \mathbb{R}^{n^d_{t-1} \times d_F}$, we calculate the Euclidean distance between every pair of elements in set $\mathbf{C}_{t-1}$ to obtain the target distance matrix $\mathbf{D}_{t-1}$, which is formulated as

$$\mathbf{D}_{t-1}[i,j] = d\left(\mathbf{c}^i_{t-1}, \mathbf{c}^j_{t-1}\right) = \|\mathbf{c}^i_{t-1} - \mathbf{c}^j_{t-1}\|_2 \tag{3}$$

where $\mathbf{D}_{t-1} \in \mathbb{R}^{n^d_{t-1} \times n_{t-1}}$, $n^d_{t-1} = n_{t-1} + n^m_{t-1}$, denotes the total number of instances sampled at the ending of task $t-1$, and $\mathbf{c}^i_{t-1}, \mathbf{c}^j_{t-1}$ represent the i-th and j-th sampled instance.

Then, distance consistency rehearsal is implemented for current task. In each mini-batch of current dataset $\mathcal{D}_t$, we get representations of current dataset and instances sampled from memory bank, denoted as $\mathbf{F}_t$ and $\mathbf{F}^m_t$. For $\mathbf{F}^m_t$, distance matrix $\mathbf{d}_t$ is obtained by the same way in Eq. (3) and illustrated in algorithm 1. To maintain the distance consistency of rehearsal samples, we minimize the Mean Squared Error (MSE) loss between distance matrix of current task and replay samples by the equations

$$\mathcal{L}_{DC}(\mathbf{d}_t, \mathbf{d}_{t-1}) = \|\mathbf{d}_t - \mathbf{d}_{t-1}\|^2_F \tag{4}$$

where $\mathbf{d}_{t-1}$ is sub-matrix sampled from $\mathbf{D}_{t-1}$ according to the indexes of $\mathbf{X}^m_t$.

4.4. Bilevel Coreset Selection

We model the process of choosing replay instances as a bilevel optimization problem, naming Bilevel Coreset Selection (BCS). The bilevel optimization contains upper-level and lower-level problems. Upper-level problem hereby aims to find the optimal model parameters θ that minimize the weighted training loss for a given set of sample

weights ω. Lower-level problem aims to optimize the sample weights ω to minimize the overall training error. The main idea of this method is to learn a probability distribution over the entire dataset, ensuring that the optimal model parameters found by minimizing loss on a sampled subset are also optimal for the full dataset, and then use this distribution to sample a coreset. Specifically, the objective function of BCS is:

$$\begin{aligned} \min_{\substack{0\le\omega(i)\le 1 \\ \|\omega\|_1=1}} & \left[\phi(\omega) = \sum_{i=1}^{n} \ell_i(\theta^*(\omega)) - \lambda \sum_{i=1}^{C} \mathbb{E}_z(\omega + \delta z)_{[i]} \right] \\ s.t., \theta^*(\omega) = \arg\min_{\theta} & \left[L(\theta, \omega) = \sum_{i=1}^{n} \omega_{(i)} \ell_i(\theta) \right] \end{aligned} \tag{5}$$

where n is the data stream size, θ is the model parameter, ω is weights of each instance in data stream, $\ell_i(\theta)$ is the loss of i-th instance under model parameter θ, $\omega_{(i)}$ is the weight of i-th instance in ω, $\omega_{[i]}$ means the i-th largest component in ω. Following [17], we also add a smoothed top-K regularization item to the optimization objective, for the sake of coreset selection. Note that $\mathcal{R} = -\lambda \sum_{i=1}^{C} \mathbb{E}_z(\omega + \delta z)_{[i]}$ is the added regularization item.

As illustrated in Algorithm 2 and Fig. 2(IV), at the end of every task, we execute BCS on current dataset and current memory bank to get sampled instances, then forming new memory bank for next task. During bilevel sampling, detailed in Algorithm 2, model parameter θ and weight distribution ω are updated alternatively. The sampling process is expected to sample C instances from given data stream $\mathcal{B}$ based on the final sample weights.

4.4.1. Initialization

In order not to modify current encoder, we deepcopy the parameter of current encoder to get θ. The initial weights of each instance is set as $\frac{1}{n}$, which could be regarded as uniform distribution.

4.4.2. Lower-level Problem

The lower problem is to optimize the model parameter θ with current sample weights ω. As depicted in Eq. (5), the total training loss is the summation of single loss of each

Algorithm 2: Bilevel Coreset Selection.

Input: current dataset $\mathcal{D}_{t-1}$ and current memory bank $\mathcal{M}_{t-1}$

Initialize: encoder parameter θ_{t-1}, new memory bank $\mathcal{M}_t = \{\}$;

for *mini-batch* $\mathcal{B}_i$ *in* $\mathcal{D}_{t-1}$ **do**
    Select coreset $\mathcal{S}_i = \mathrm{S}(\theta_{t-1}, \mathcal{B}_i)$
    $\mathcal{M}_t = \mathcal{M}_t \cup \mathcal{S}_i$
end

for *mini-batch* $\mathcal{B}_j$ *in* $\mathcal{M}_{t-1}$ **do**
    Select coreset $\mathcal{S}_j = \mathrm{S}\left(\theta_{t-1}, \mathcal{B}_j\right)$
    $\mathcal{M}_t = \mathcal{M}_t \cup \mathcal{S}_j$
end

`// Definition of coreset selection function`

Function $\mathrm{S}(\theta, \mathcal{B})$
Input: current encoder parameter θ, current data stream $\mathcal{B}$, outer optimization loop number M, inner optimization loop number m and hypergradient estimate number H
Output: a set of C instances sampled from $\mathcal{B}$
Initialize: coreset size C, data stream size $n = |\mathcal{B}|$, and weight vector v_0
Begin
    Set instances' initial values $\omega_0 = \left[\frac{1}{n}, \cdots, \frac{1}{n}\right]$;
    Set $\theta_1^0 = \theta$;
    `// Outer optimization loop (Upper-level problem)`
    for $j \leftarrow 0$ **to** $M - 1$ **do**
        if $j > 0$ **then** Set $\theta_j^0 = \theta_{j-1}^m$
        else Set $v_j^0 = v_0$
        end if
        `// Inner optimization loop (Lower-level problem)`
        for $k \leftarrow 1$ **to** m **do**
            update θ_j^k by Eq. 6
        end for
        if $j > 0$ **then** Set $v_j^0 = v_{j-1}^H$
        else Set $v_j^0 = v_0$
        end if
        Compute estimate v_j^H by Eq. (8)
        Compute hypergradient estimate in Eq. (7)
        Update ω_{j+1} and project it onto simplex by Eq. (9)
    end for
    $\mathcal{S} \leftarrow$a set of C instances sampled from $\mathcal{B}$ according to ω_M
    return $\mathcal{S}$
End
End Function

instance combined with regularization item. After computing loss, we execute m steps of gradient descent to get satisfactory parameters θ, formulated as:

$$\theta_j^k = \theta_j^{k-1} - \alpha \nabla_\theta L\left(\theta_j^{k-1}, \omega_j\right) \tag{6}$$

where θ_j^k means the model parameter at the j-th outer loop and the k-th inner loop.

4.4.3. Upper-level Problem

The upper-level problem intends to find a weight distribution ω that can describe the importance of each instance on minimizing the training loss of current model. In order to achieve this purpose, we utilize a hypergradient estimator to approximate the true gradient $\nabla\phi(\omega)$ of ω:

$$\varphi_j = \frac{1}{|\mathcal{B}|} \sum_{\tilde{z} \in \mathcal{B}} \nabla_\omega \mathcal{R}(\omega, \delta; \tilde{z}) - \nabla_\omega \nabla_\theta L(\theta_j^m, \omega_j) v^* \tag{7}$$

where $\mathcal{R}(\omega, \delta; \tilde{z}) := -\lambda \sum_{i=1}^{C} (\omega + \delta\tilde{z})_{[i]}$ and $\tilde{z} \sim \mathcal{N}(0, 1)$. v^* can be approximated by solving the following quadratic programming problem efficiently by H steps of gradient descent using Eq. (8):

$$\min_v \frac{1}{2} v^T \nabla_\theta^2 L(\theta_j^m, \omega_j) v - v^T \sum_{i=1}^{n} \nabla_\theta \ell_i(\theta_j^m) \tag{8}$$

Then the weight is updated and projected onto simplex by Eq. (9):

$$\omega_{j+1} = \mathcal{P}_{\Delta^n}(\omega_j - \beta\varphi_j) \tag{9}$$

where $\Delta^n := \{\omega \in \mathbb{R}^n : 0 \leq \omega_{(i)} \leq 1, \|\omega\|_1 = 1\}$.

4.5. Objective and Optimization

Above all, we design the distance consistency loss $\mathcal{L}_{DC}$ to maintain the queue stability of a continual CBHIR system. Meanwhile, pair-wise loss and cross-entropy loss have been proven effective in image retrieval task [18, 19]. As a result, the model is optimized in an end-to-end fusion by

$$\mathcal{L}_{all} = \mathcal{L}_P\left(\mathbf{F}_t, \mathbf{F}_t^m, y_t\right) + \mathcal{L}_{CE}\left(y_t, y_t^{pred}\right) + \alpha \mathcal{L}_{DC}\left(\mathbf{d}_t, \mathbf{d}_{t-1}\right) \tag{10}$$

where $\mathcal{L}_P$ is the pair-wise loss, $\mathcal{L}_{CE}$ is the cross-entropy loss, $\mathcal{L}_{DC}$ is the distance consistency loss, y_t and y_t^{pred} are the ground-truth label and output logits of input instances, and α is a balancing factor. Through Eq. (10), we attempt to not only improve retrieval precision of LCBHIR but also maintain queue consistency as much as possible.

5. Experiment And Result

5.1. Experimental Settings

The proposed method was evaluated on a large-scale sequential WSI retrieval dataset from The Cancer Genome Atlas program (TCGA) of National Cancer Institute (NCI), which consists of 6 primary-site (organ) datasets and 19 cancer subtypes, with a total of 7347 WSIs, all listed datasets are publicly available. In each dataset, the WSIs were randomly separated into train, validation and test subsets by the proportion of 7:1:2. The validation subsets were used to perform early stopping and hyper-parameter selection. All methods were evaluated under this setting. The distribution of WSIs is given in Table 1.

Following the WSI pre-processing in [2], we first segmented the foreground tissue. Then each WSI was divided into non-overlapping 256×256 patches under $20\times$ magnification. We utilized PLIP [29] as the pre-trained feature extractor to extract patch-level features from the corresponding images. Vision Mamba (Vim) [30] was utilized as the WSI encoder (shown in Fig. 2(c)) for its effectiveness has been verified on various WSI analysis tasks [31, 32]. For a fair comparison, we adopted the same patch-level features as the model input and Vim as the slide encoder for all methods. Moreover, we conducted experiments under buffer size of 100, 200 and 500 instances For more comprehensive analysis.

The proposed method was implemented in Python 3.10 with Pytorch 2.1 and run on a computer with 2 Intel Xeon 2.90 GHz CPUs and 2 GPUs of Nvidia Geforce RTX 4090. For more detailed experiment implementations, please refer to https://github.com/OliverZXY/LCBHIR.

5.2. Evaluation Metrics

For evaluation of retrieval precision, we reported 14 metrics, containing slide-level mAP, R@3, and P@5 to evaluate the performance of 19-class tumor type retrieval,

Table 1: The WSI distribution of the experimental datasets

Brain	**LGG**			**GBM**									**Cases**
	Train	Val	Test	Train	Val	Test							
	575	82	165	599	86	172							1679
Urinary	**KIRP**			**KIRC**			**KICH**			**BLCA**			**Cases**
	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test	
	207	29	60	361	51	104	84	11	25	319	46	92	1389
Gastrointestinal	**COAD**			**ESCA**			**READ**			**STAD**			**Cases**
	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test	
	308	44	89	109	16	32	110	16	32	280	40	80	1156
Pulmonary	**LUSC**			**LUAD**			**MESO**						**Cases**
	Train	Val	Test	Train	Val	Test	Train	Val	Test				
	347	49	100	349	50	100	60	9	18				1082
Gynecology	**OV**			**UCS**			**CESC**			**UCEC**			**Cases**
	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test	
	74	11	22	63	9	19	195	28	56	396	56	114	1043
Breast	**IDC**			**ILC**									**Cases**
	Train	Val	Test	Train	Val	Test							
	555	80	159	142	21	41							998
Total Cases													7347

and site-mAP, site-R@3, site-P@5 for the 6-anatomic-site retrieval. Furthermore, we adopted two statistical correlation metrics Spearman's rank correlation coefficient (SRC) [33] and Kendall rank correlation coefficient (KRC) [34] to assess the consistency of returned queues for old tasks, which is the stability of a good lifelong CBHIR system should have. SRC and KRC can be defined by equations:

$$SRC = \frac{1}{n-1}\sum_{i=1}^{n-1}\left(\frac{1}{n-i}\sum_{j=i+1}^{n}\rho_{i,j}\right) \tag{11}$$

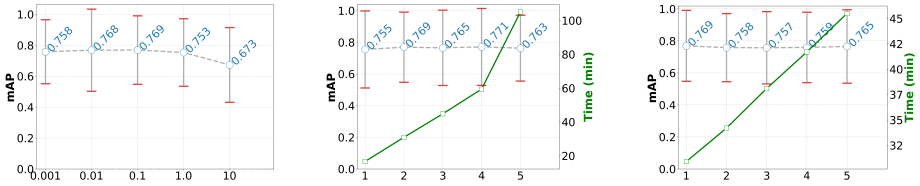
$$\rho_{i,j} = 1 - \frac{6\sum d_k^2}{n(n^2-1)} \tag{12}$$

where d_k is the difference between the ranks of retrieval sequence of i-th task after j tasks' training, and n is the number of instance of each sequence.

$$KRC = \frac{1}{n-1}\sum_{i=1}^{n-1}\left(\frac{1}{n-i}\sum_{j=i+1}^{n}\tau_{i,j}\right) \tag{13}$$

$$\tau_{i,j} = \frac{C - D}{\frac{n(n-1)}{2}} \tag{14}$$

where C is the number of concordant pairs of retrieval sequence of i-th task after j tasks' training, D is the number of discordant pairs, and n is the number of instance of each sequence.



(a) The DC loss balancing factor α (b) The max outer iteration num M (c) The max inner iteration num m

Figure 4: Performance curves of LCBHIR on the validation data as functions of the hyper-parameters, where the red bar indicates the distribution of mAP at the subtype level.

5.3. Hyper-Parameter Verification

We first conducted experiments to verify the key components of LCBHIR. Three important hyper-parameters (α, M, m) decide the capacity of LCBHIR. We executed selection experiments on the validation part of sequential dataset under a buffer size of 500 instances. The mAP and running time costs of these hyper-parameters are displayed in Fig. 4. Note that while one hyper-parameter was being optimized, the remaining hyper-parameters were held constant.

5.3.1. The DC loss Balancing Factor

The DC Loss determines a model's focus on previous tasks. Paying too much attention to previous tasks will degrade the model's generalization ability to adapt current tasks. Meanwhile, too less focus will accelerate model forgetting previous tasks. The curves in Fig. 4a shows that the retrieval performance achieve the best when alpha is set as 0.1 and demonstrates a sharp decline when alpha is 10.0. As a result, we set $\alpha = 0.1$ for its best balance in stability and plasticity.

5.3.2. The Bilevel Optimization Coefficient

M and m decide the number of outer and inner iteration loops. From Figs. 4b and 4c, different combinations of M and m show robust towards retrieval performance. However, the time cost or computational load varies a lot between different M and m. To attain an equilibrium between retrieval performance and computational efficiency, we chose $\{M, m\} = \{2, 1\}$ in the following experiments.

Table 2: Results on the test subset for the ablation study, reporting subtype/primary site-level retrieval precision alongside the mean of subtype and primary site-level retrieval precision, combined with consistency metrics.

Method	**Subtype-level** (%)			**Primary Site-level** (%)			**Consistency**	
	mAP@5 / $\overline{\text{C}}$	R@3 / $\overline{\text{C}}$	P@5 / $\overline{\text{C}}$	mAP@5 / $\overline{\text{C}}$	R@3 / $\overline{\text{C}}$	P@5 / $\overline{\text{C}}$	SRC	KRC
LCBHIR w/o DCL	72.8 / 60.1	**83.1** / **72.1**	63.8 / **49.4**	88.5 / 87.0	92.5 / 91.8	85.0 / 83.0	81.7	68.1
LCBHIR w/o BCS	73.1 / 59.4	82.9 / 71.6	65.4 / 49.3	88.3 / 85.8	92.0 / 90.7	85.1 / 82.1	82.0	68.7
LCBHIR w/o all	72.3 / 58.7	82.2 / 70.8	65.1 / 48.3	88.6 / 86.2	92.1 / 90.6	85.5 / 82.6	81.8	68.3
LCBHIR w/ BuRo	72.8 / 57.4	82.0 / 67.4	64.7 / 47.3	89.4 / 86.5	93.2 / 91.3	85.7 / 82.6	80.8	67.5
LCBHIR w/ iCaRL	57.2 / 42.5	71.5 / 54.9	44.3 / 31.0	75.5 / 74.1	85.6 / 85.2	66.5 / 64.6	38.8	27.3
LCBHIR	**74.0** / **60.8**	82.7 / 70.3	**65.8** / 49.3	**90.0** / **88.0**	**93.7** / **92.9**	**86.0** / **83.3**	**83.3**	**70.8**

5.4. Ablation Study

Table 2 provides a summary of the ablation study results. LCBHIR w/o DCL denotes LCBHIR without distance consistency loss, where the whole model is optimized only by cross-entropy loss and pair-wise loss. It can be observed a decrease both in retrieval performance and previous tasks' consistency when comparing LCBHIR w/o DCL with LCBHIR, where the subtype-level mAP@5, R@3, P@5, site-level mAP@5, R@3, P@5 and SRC, KRC dropped 0.7%~2.7%. Without the constraint of distance consistency loss, the model failed to preserve knowledge of previous tasks, indicating that employing relative distance between slides to construct lifelong learning loss is of vital importance for CBHIR.

Moreover, we also studied the different sampling strategy for updating memory bank. In Table 2, LCBHIR w/o BCS means reservoir random sampling, LCBHIR w/ BuRo refers to Breakup-Reorganize (BuRo) scheme in ConSlide [9] and LCBHIR w/ iCaRL denotes buffer sampling strategy in iCaRL [12]. From Table 2, our

BCS sampling method achieve the best performance. Compared with reservoir random sampling, BCS can detect hard samples and attach higher weights to them when sampling. As the strong continual learning baseline in histopathology image analysis, BuRo from ConSlide achieve impressive results but still weaker than BCS. Though BuRo can be regarded as a mean of feature augmentation by randomly selecting and combining patches from same category WSIs, the artificial samples created by it will inevitably damage the reliability of a continual histopathology retrieval system. Note that LCBHIR w/ iCaRL showed a significant performance gap between other strategies, indicating that feature space generated by iCaRL lack of diversity for retrieval tasks and undermines the model's retrieval performance. Furthermore, this elucidates that for continual learning retrieval tasks, the feature space should be as dispersed as possible and not concentrated around samples of each disease category.

5.5. Comparison with State-Of-The-Art Continual Learning Approaches

We compared our method with 7 methods, including JointTrain, Finetune, LwF [35], EWC [36], ER-ACE [24], A-GEM [25] and DER++ [26]. For all methods, we employed the same WSI encoder and reservoir random sampling to execute model training. The subtype-level and site-level retrieval metrics and consistency metrics are presented in Table 3.

Initially, we trained a model using all datasets under full supervision, referred to as JointTrain in Table 3, which is the upper bound. Then, we conducted sequential individual training on the 6-task datasets, referred to as Finetune, representing the lower boundary. As shown in Table 3, the model subjected to Finetune demonstrates the poorest performance, indicating that catastrophic forgetting occurs as the database size increases. In contrast, our method significantly alleviates catastrophic forgetting, achieving performance nearly on par with the JointTrain approach. Although our method shows a certain performance gap compared to JointTrain, it offers better scalability and is capable of continuously learning new data even in the absence of previous data.

LwF [35] and EWC [36] are two regularization based continual learning method which did not store or utilize previous tasks' samples during a whole training period. However, from Table 3, they failed to effectively alleviate catastrophic forgetting. LwF

Table 3: Performance comparison of state-of-the-art continual learning approaches, reporting subtype/primary site-level retrieval precision alongside the mean of subtype and primary site-level retrieval precision, combined with consistency metrics, and the best values are shown in bold.

Method	**Buffer Size**	**Subtype-level** (%)			**Primary Site-level** (%)			**Consistency**	
		mAP@5 / $\overline{C}$	R@3 / $\overline{C}$	P@5 / $\overline{C}$	mAP@5 / $\overline{C}$	R@3 / $\overline{C}$	P@5 / $\overline{C}$	SRC	KRC
Baselines									
JointTrain	-	**83.7 / 72.9**	**89.4 / 81.0**	**78.6 / 63.9**	**94.0 / 93.0**	**95.9 / 95.4**	**92.3 / 90.8**	-	-
Finetune		46.2 / 35.2	61.3 / 46.2	32.0 / 22.8	64.4 / 61.5	79.5 / 76.7	52.3 / 47.9	57.0	42.4
Regularization Based									
LwF [35]	-	66.9 / 52.8	78.4 / 63.8	55.1 / 39.8	79.8 / 76.5	87.6 / 85.7	71.5 / 66.7	60.9	47.9
EWC [36]		58.7 / 44.7	73.0 / 56.1	46.0 / 32.3	74.5 / 71.0	85.0 / 82.9	64.6 / 59.4	58.0	43.2
Replay Based									
ER-ACE [24]	100	67.5 / 50.2	77.2 / 60.5	57.6 / 39.3	83.4 / 79.1	87.9 / 86.0	78.7 / 73.0	76.3	62.4
A-GEM [25]		63.3 / 48.5	77.0 / 61.4	51.9 / 36.1	79.5 / 75.1	88.9 / 86.8	70.8 / 64.6	63.4	48.9
DER++ [26]	≈ 5 WSIs	68.0 / 52.7	77.5 / 62.6	60.0 / 43.8	84.8 / 81.2	89.2 / 87.6	81.0 / 76.3	72.6	58.4
LCBHIR		**73.7 / 60.1**	**83.4 / 71.3**	**64.8 / 49.4**	**88.8 / 86.6**	**92.7 / 91.8**	**84.7 / 81.9**	**80.5**	**66.6**
ER-ACE [24]	200	67.9 / 51.1	77.2 / 60.9	60.5 / 43.0	84.5 / 78.2	88.8 / 84.6	81.0 / 74.0	77.3	62.5
A-GEM [25]		66.9 / 53.0	78.9 / 64.7	54.1 / 39.3	81.8 / 78.2	90.2 / 89.3	73.6 / 68.4	62.3	47.7
DER++ [26]	≈ 10 WSIs	68.1 / 50.9	78.5 / 61.5	60.0 / 41.2	84.7 / 79.4	88.6 / 85.1	81.6 / 75.0	77.5	63.2
LCBHIR		**74.0 / 60.8**	**82.7 / 70.3**	**65.8 / 49.3**	**90.0 / 88.0**	**93.7 / 92.9**	**86.0 / 83.3**	**83.3**	**70.8**
ER-ACE [24]	500	69.9 / 53.2	78.4 / 62.8	62.9 / 44.5	85.5 / 80.8	89.6 / 86.9	81.9 / 75.6	77.5	63.3
A-GEM [25]		68.8 / 55.8	81.7 / 68.5	57.8 / 43.2	84.6 / 81.7	91.5 / 90.7	77.8 / 73.4	77.6	66.0
DER++ [26]	≈ 25 WSIs	71.1 / 53.5	80.7 / 63.8	63.6 / 45.4	87.3 / 83.3	90.3 / 87.6	84.1 / 79.4	77.0	62.3
LCBHIR		**78.1 / 64.0**	**86.0 / 74.6**	**70.4 / 52.6**	**91.2 / 89.0**	**94.6 / 93.3**	**88.0 / 85.1**	**85.4**	**73.0**

achieves continual learning by using knowledge distillation to preserve the knowledge of old tasks while learning new tasks. As database grows, LwF struggles to balance knowledge retention and new task learning, leading to ineffective knowledge distillation over time. EWC prevents catastrophic forgetting by penalizing changes to important parameters, achieved by using the Fisher Information Matrix. These regularization constraints accumulate across tasks, eventually restricting the model's flexibility to learn new tasks effectively, causing performance degradation.

ER-ACE [24], A-GEM [25] and DER++ [26] are replay based methods. A-GEM prevents forgetting by constraining the gradient updates during new task learning, but gradient projection mechanism might become overly restrictive as the number of tasks increases, limiting the model's ability to learn new tasks effectively. ER-ACE and

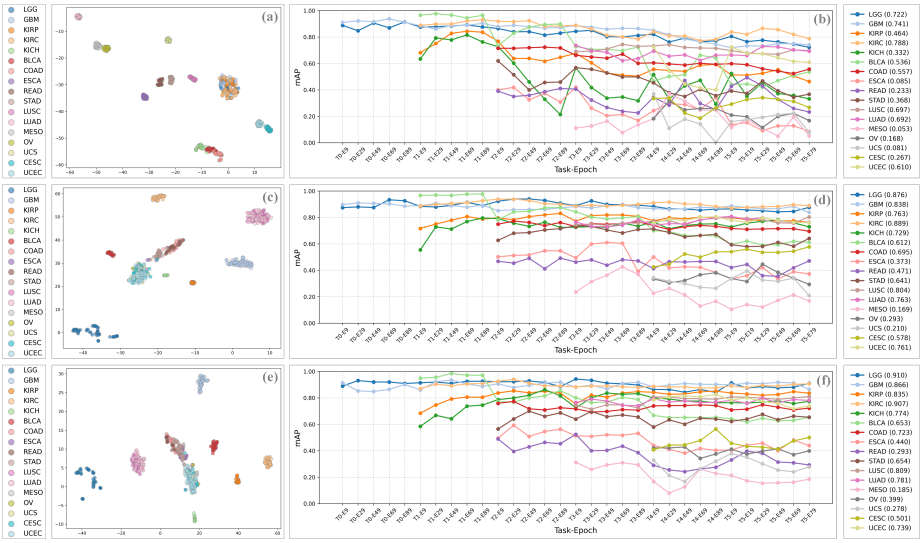


Figure 5: The t-SNE visualization of different sampling strategies and mAP values for each label across tasks and epochs, where (a)-(b) is the corresponding results of iCaRL, (c)-(d) is BuRo and (e)-(f) is BCS.

DER++ have comparative lifelong learning ability, but they still fall short compared to LCBHIR. Through active selection strategies, ER-ACE prioritizes and revisits significant past experiences, allowing the model to retain important knowledge. This active selection strategy might struggle to prioritize the most critical samples as the number of tasks increases, resulting in sub-optimal replay and a gradual forgetting of earlier tasks. DER++ maintains continual learning by storing and replaying both past experiences' logits and their predicted outputs, but the stored logits may no longer accurately reflect the evolving nature of the tasks, causing the replay mechanism to become less effective. In comparison, our LCBHIR utilize the distance consistency loss to execute replaying, which is a nature prior in lifelong CBHIR, leading the model to preserve present knowledge by the limit of relative distance matrix. Besides, the bilevel coreset selection is employed to assure feature space's diversity when sampling.

5.6. Scalability in Two-Stage Retrieval

Retrieval systems commonly adopt a two-stage strategy to improve retrieval accuracy [37]. We further evaluated the scalability of the proposed lifelong framework within this two-stage retrieval setting, elaborate as below:

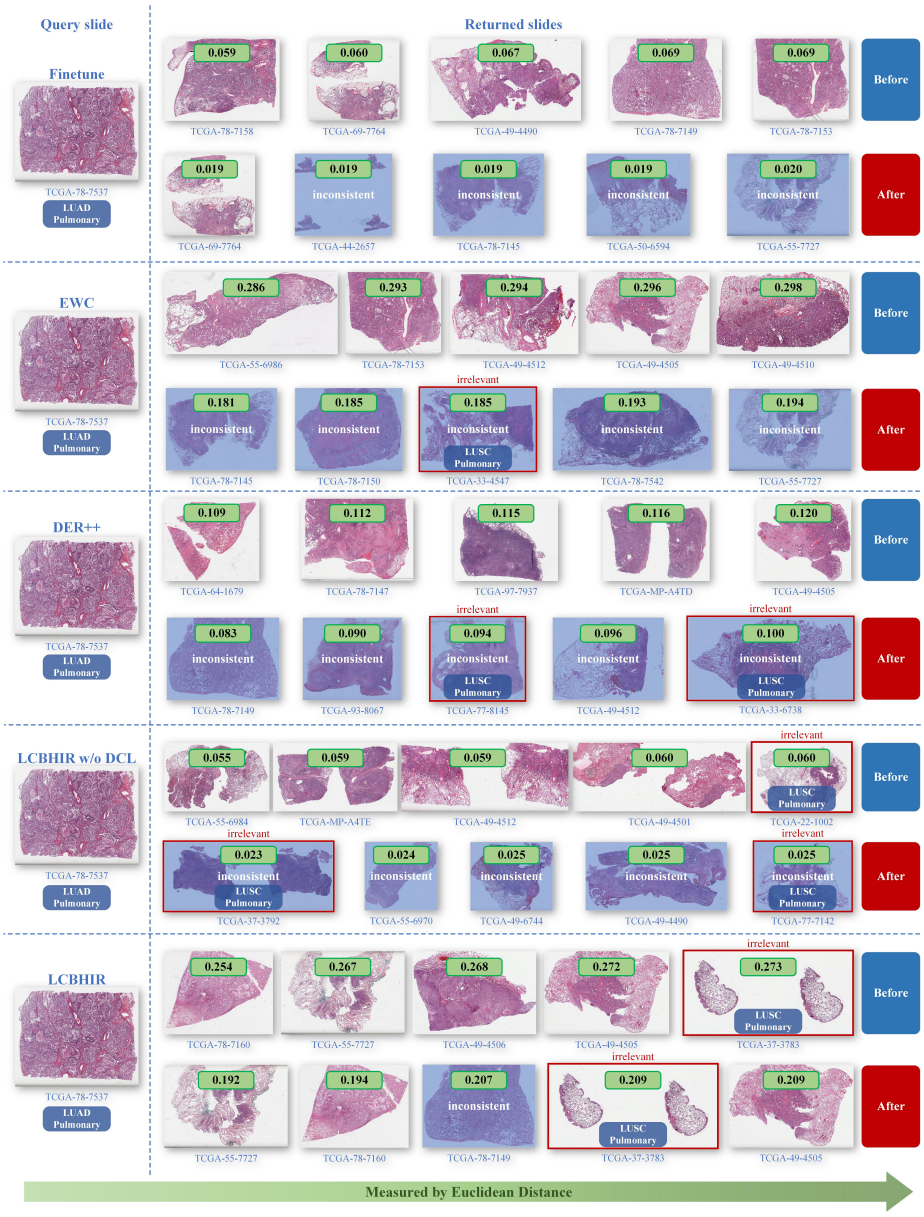


Figure 6: Visualization of returned queues before and after continual learning. Euclidean distance of each returned WSI is labeled in green box, where smaller values indicate higher similarity to the query slide. Inconsistent results are filled with blue, irrelevant results are framed in red, and query and irrelevant images' cancer subtype and primary site are labelled with blue boxes.

Table 4: Performance comparison of single-stage and two-stage approaches, reporting subtype- and primary site-level Precision@5 (P@5), their mean values across individual classes ($\overline{\text{P@5}}$), and retrieval time.

Method	**Two Stage**	**Subtype-level** (%)		**Primary Site-level** (%)		**Retrieval Efficiency**
		P@5	$\overline{\text{P@5}}$	P@5	$\overline{\text{P@5}}$	Speed (s/it)
LCBHIR	✗	70.4	52.6	88.0	85.1	16
	✔	71.9	56.2	89.0	86.6	147
	Δ	+1.5	+3.6	+1.0	+1.5	-131

1) Slide-level coarse-grained retrieval. We first retrieved the top-m most similar slides (with $m = 20$) using the same procedure described above.

2) Patch-level fine-grained retrieval. We followed RetCCL [37] to execute patch-level retrieval. Specifically, within the m retrieved slides, we constructed a mosaic representation for each WSI by applying k-means clustering to extract s centroids and selecting the s nearest patch features, resulting in $WSI = \{P_1, P_2, P_i, \cdots, P_s\}$. Each patch P_i was then used as a query to retrieve top-t similar patches, forming s bags: $Bag = \{B_1, B_2, B_i, \cdots, B_s\}$, where $B_i = \{b_i^1, b_i^2, b_i^i, \cdots, b_i^t\}$ and $D_i = \{d_1, d_2, d_i, \cdots, d_t\}$ denotes the corresponding Euclidean distances between each P_i and its retrieved top-k similar patches. We computed the mean distance of each bag as the ranking criterion, reordered the bags accordingly to obtain Bag', and used majority voting within each B_i to determine the corresponding slide W_i. Finally, we returned the top-k most relevant WSIs. Here, $s = 10$, $t = 100$ and $k = 5$.

As shown in Table 4, the two-stage retrieval setting improves Precision@5 by 1.5% at the subtype level and 1.0% at the primary site level compared to the single-stage retrieval. This demonstrates that the two-stage mechanism helps mitigate the performance degradation caused by the loss of contextual information.

5.7. Computational Complexity

The computational cost of continual learning methods is critical for their large-scale deployment in real-world applications. To provide a clearer comparison of computational efficiency, we have conducted a time and space complexity analysis of all

Table 5: Time and space complexity analysis. **Notation:** T–number of tasks, n–number of samples per task, θ–model parameters, M–replay buffer size, D–dimension of input sample, d–dimension of model output (logits or class token).

Method	Type	Time Complexity	Space Complexity
Finetune	Baseline	$O(T \cdot n \cdot \lvert\theta\rvert)$	$O(\lvert\theta\rvert)$
LwF [35]	Regularization based	$O(T \cdot n \cdot \lvert\theta\rvert)$	$O(2\lvert\theta\rvert)$
EWC [36]		$O(T \cdot n \cdot \lvert\theta\rvert)$	$O(4\lvert\theta\rvert)$
ER-ACE [24]	Replay based	$O(T \cdot (n+M) \cdot \lvert\theta\rvert)$	$O(\lvert\theta\rvert + M \cdot (D+d))$
A-GEM [25]		$O(T \cdot (n+M) \cdot \lvert\theta\rvert)$	$O(2\lvert\theta\rvert + M \cdot (D+d))$
DER++ [26]		$O(T \cdot (n+M) \cdot \lvert\theta\rvert)$	$O(\lvert\theta\rvert + M \cdot (D+d))$
LCBHIR		$O(T \cdot (2n+M) \cdot \lvert\theta\rvert)$	$O(\lvert\theta\rvert + M \cdot (D+d) + M^2)$

compared methods. For all continual learning baselines and methods, the architecture of WSI encoder and retrieval inference procedure are the same. Therefore, the analysis focuses on the additional training cost and memory overhead introduced by the continual learning strategies. We use T denotes number of tasks, n denotes number of samples per task, θ denotes model parameters, M denotes replay buffer size, D denotes dimension of input sample and d denotes dimension of model output (logits or class token). As shown in the Table 5, The time complexity of Finetune corresponds to standard training, and its space complexity equals the model size $|\theta|$, as it introduces no additional operations during sequential learning. Regularization-based methods (*i.e.*, LwF [35] and EWC [36]) have the same time complexity as Finetune. LwF [35] requires storing a copy of the previous model for knowledge distillation, resulting in $O(2|\theta|)$ space complexity. EWC [36] maintains the current model, the optimal parameters from the previous task, the corresponding Fisher diagonal, and an accumulated Fisher vector, yielding a space complexity of $O(4|\theta|)$. Replay-based methods incur additional time due to rehearsal and require memory storage, with typical time and space complexity denoted as $O(T \cdot (n+M) \cdot |\theta|)$ and $O(|\theta| + M \cdot (D+d))$, respectively. A-GEM [25] further stores a previous model for distillation, increasing its space complexity to $O(|\theta| + M \cdot (D+d)) + O(|\theta|)$. In our LCBHIR, bilevel coreset selection is performed at the end of each task, adding $O(T \cdot n \cdot |\theta|)$ to the time complexity. Additionally, dis-

tance consistency rehearsal requires storing a relative distance matrix, increasing the space complexity by $O(M^2)$. However, the overall computation and memory overhead of replay-based methods is not significantly higher than that of Finetune, as $M \ll N$. The total GPU memory consumption of our method during training is 22.39 GB with a batch size of 32 (including cached replay samples), compared to 16.27 GB for Finetune, resulting in an increase of 6.12 GB. This additional cost remains acceptable in practical application scenarios.

5.8. Visualization

To more effectively illustrate the superiority of our approach, we execute visualization to display effect of different sampling strategies and retrieval returned queue of several methods.

5.8.1. Visualization of Sampling Strategy

We believe plasticity in lifelong CBHIR systems is reflected in retaining the memory of previous tasks while acquiring new knowledge. Different sampling strategies affect the model's focus on current and previous tasks, thereby influencing model plasticity. We saved class tokens of instances in the memory bank for three sampling strategies: iCaRL, BuRo and Bilevel Coreset Selection (BCS), then executing t-SNE visualization to observe them on the same scale. Meanwhile, we plotted the mAP values for each label changing across tasks and epochs to observe different sampling methods' capacity towards maintaining precision of previous tasks.

As illustrated in Fig. 5(a), iCaRL's clustering mechanism leads to a noticeable clustering effect within each category. However, in fine-grained retrieval tasks, this clustering effect diminishes feature diversity and would cause feature space collapsing. Depicted in Fig. 5(b), the performance degradation of iCaRL reveals a continuous decline, where it fails to maintain precision for previous tasks after continual learning. In contrast to BuRo, BCS provides a more dispersed feature space, displayed in Figs. 5(c) and 5(e). Moreover, Figs. 5(d) and 5(f) demonstrates that BCS achieves better retrieval precision retention than BuRo in categories such as LGG and STAD, and even more challenging categories like ESCA. The broader and more diverse feature space

generated by BCS supports fine-grained retrieval, reducing the likelihood of generating virtual cases as BuRo would.

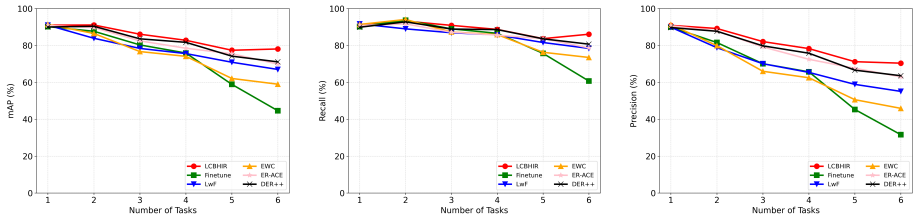
5.8.2. Consistency of Returned Queues

To validate the effectiveness of distance consistency loss in our framework, we compared the returned queues for a query slide after task 3 and after the entire continual training across different methods. As presented in Fig. 6, we compared five methods, with the query slide on the left and the corresponding returned queues on the right. For each method, the first row shows the initial returned queues, and the second row shows the queues after training. Euclidean distance is used as the retrieval metric. Euclidean distance is used as the retrieval metric. In Figure 6, we annotate each returned WSI with its Euclidean distance (in green box), where smaller values indicate higher similarity to the query slide. Irrelevant slides in both queries are outlined in red, while inconsistencies in the second query are highlighted in blue.

In Fig. 6, our method demonstrates the best performance in maintaining the consistency of returned queues. Without the consistency loss constraint, LCBHIR w/o DCL fails to preserve the top returned slides and introduces more irrelevant cases in the second query. In a lifelong histopathology CBHIR system, balancing stability and plasticity is crucial. Stability here refers to the consistency of returned queues throughout continual training. However, other methods, including Finetune, EWC, and DER++, struggle with this consistency. Finetune suffers from catastrophic forgetting due to the continuous data stream, while EWC and DER++, despite incorporating knowledge distillation, can only ensure the relevance of returned cases, not the consistency, which undermines the system's credibility.

5.8.3. Forgetting in Continual Learning for WSI Retrieval

Fig. 7 presents the performance curves (mAP, Recall, and Precision at the subtype level) of the baseline and several continual learning methods as the number of tasks increases. As shown in the Fig. 7a, Finetune suffers from severe catastrophic forgetting, especially with a sharp performance drop from 0.758 to 0.589 during task 4 changing to task 5. Tasks 4, 5, and 6 contain both rare and common cancer subtypes, which fur-



(a) Incremental mAP@5 (b) Incremental Recall@3 (c) Incremental Precision@5

Figure 7: Incremental performance curves on sequential tasks at subtype level.

ther exacerbates the already limited discriminative ability of Finetune, leading to poor overall performance.

Fig. 7c witnesses that DER++ drops from 0.897 to 0.636 and EWC from 0.913 to 0.459 on Precision@5, demonstrating that replay-based methods have better performance retention than regularization-based approaches, possibly because most regularization methods are originally designed for natural images. These regularization losses tend to be less effective when applied to pathological images, due to highly heterogeneous and exhibit significant domain shifts.

On Recall@3 (Fig. 7b), our method achieves excellent stability, with a slight drop from 0.910 to 0.861, while ER-ACE has a sharp drop from 0.908 to 0.787, showing that our method outperforms other replay-based approaches. We attribute this to the proposed distance consistency rehearsal and bilevel coreset selection. These strategies not only help maintain stable retrieval queues, but also reinforce the decision boundaries in the feature space by mining hard samples, thereby ensuring stable retrieval performance over time.

6. Discussion

In this work, we propose LCBHIR, a continual learning method for WSI retrieval. A core challenge in continual learning is achieving a balance between stability and plasticity, which is also essential for model interpretability. In this work, we define interpretability primarily at the global level, focusing on whether the system's behavior during sequential learning can be understood and trusted. This is illustrated directly by

the experimental results shown in Fig. 5–7. Stability is captured in Fig. 6 and Fig. 7. Fig. 6 shows the retrieval queues before and after continual learning, where our method preserves consistent retrieval results for previously seen data. Fig. 7 further provides the performance–forgetting curve, quantifying how the system retains retrieval accuracy across tasks. Together, these visualizations make it interpretable why our system maintains stable behavior over time—avoiding abrupt shifts that could mislead clinicians in real applications. Plasticity is reflected in Fig. 5, which demonstrates how new subtype data are integrated into the feature space. The visualization shows that both common subtypes (e.g., COAD and STAD) and rare ones (e.g., LGG) are effectively incorporated without disrupting prior knowledge. This provides a clear global explanation of the model's adaptability and generalization capacity in diverse clinical contexts.

In the aspect of application, efforts will be made on the practical clinical deployment of continual learning-based retrieval systems. We plan to optimize both the training and inference processes, and further integrate our approach with modern database systems [38]. For example, implement incremental indexing tables could be used to temporarily store newly learned data and support batch merging. Additionally, user feedback could be integrated at the SQL layer to update image matching weights, enabling personalized retrieval.

7. Conclusion

In this work, we proposed a Lifelong Content-based Histopathology Image Retrieval (LCBHIR) framework to address the challenge of catastrophic forgetting in continually expanding whole slide image (WSI) databases. Our method strikes a balance between stability and plasticity, which is crucial for continual learning systems in clinical applications. A bilevel coreset sampling strategy with a local memory bank is introduced to maintain plasticity, which has been proven effective in mining challenging samples and refining decision boundaries. To preserve stability, we designed a distance consistency rehearsal (DCR) module to maintain consistent retrieval queues for previously learned tasks, thereby improving the reliability of retrieval outcomes

over time. In the future, we will concentrate on the clinical application of continual learning-based retrieval systems and explore more practical scenarios, such as data-incremental settings and fine-grained retrieval of rare cancer subtypes. We hope this study could encourage further research on continual learning in histopathology image retrieval and contribute to the advancement of large-scale, lifelong CBHIR systems.

CRediT authorship contribution statement

Xinyu Zhu: Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Zhiguo Jiang:** Funding acquisition, Formal analysis, Supervision, Writing – review & editing. **Kun Wu:** Data curation, Writing – review & editing. **Jun Shi:** Funding acquisition, Supervision, Writing – review & editing. **Yushan Zheng:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing. All authors have read and approved this version of the manuscript, and due care has been taken to ensure the integrity of this work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partly supported by Beijing Natural Science Foundation (Grant No. 7242270), partly supported by the National Natural Science Foundation of China (Grant No. 62171007, 61901018, and 61906058), partly supported by the Fundamental Research Funds for the Central Universities of China (Grant No. YWF-23-Q-1075), and partly supported by the Anhui Provincial Natural Science Foundation (Grant No. 2408085MF162).

Appendix A. Vision Mamba

Appendix A.1. Preliminaries

The Vision Mamba (Vim) [30] is inspired by the continuous system, which maps a 1-D function or sequence $x(t) \in \mathbb{R} \mapsto y(t) \in \mathbb{R}$ through a hidden state $h(t) \in \mathbb{R}^H$. This system uses $\mathbf{A} \in \mathbb{R}^{H\times H}$ as the evolution parameter and $\mathbf{B} \in \mathbb{R}^{H\times 1}$, $\mathbf{C} \in \mathbb{R}^{1\times H}$ as the projection parameters. The continuous system works as follows: $h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t)$ and $y(t) = \mathbf{C}h(t)$. Vim is the discrete versions of the continuous system, which include a timescale parameter $\mathbf{\Delta}$ to transform the continuous parameters $\mathbf{A}$, $\mathbf{B}$ to discrete parameters $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$. The commonly used method for transformation is zero-order hold (ZOH), which is defined as follows:

$$\begin{aligned} \overline{\mathbf{A}} &= \exp(\mathbf{\Delta A}), \\ \overline{\mathbf{B}} &= (\mathbf{\Delta A})^{-1}(\exp(\mathbf{\Delta A}) - \mathbf{I}) \cdot \mathbf{\Delta B}. \end{aligned} \tag{A.1}$$

After the discretization of $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$, the discretized version using a step size $\mathbf{\Delta}$ can be rewritten as:

$$\begin{aligned} h_t &= \overline{\mathbf{A}}h_{t-1} + \overline{\mathbf{B}}x_t, \\ y_t &= \mathbf{C}h_t. \end{aligned} \tag{A.2}$$

Finally, the models compute output by a global convolution.

$$\begin{aligned} \overline{\mathbf{K}} &= (\mathbf{C}\overline{\mathbf{B}}, \mathbf{C}\overline{\mathbf{A}\mathbf{B}}, \ldots, \mathbf{C}\overline{\mathbf{A}}^{N-1}\overline{\mathbf{B}}), \\ \mathbf{y} &= \mathbf{x} * \overline{\mathbf{K}}, \end{aligned} \tag{A.3}$$

where N is the length of the input sequence $\mathbf{x}$, and $\overline{\mathbf{K}} \in \mathbb{R}^N$ is a structured convolutional kernel.

Appendix A.2. Vim Block

The process of **Vim** block is detailed in Algorithm3. The input token sequence $\mathbf{z}_{l-1}$ is first normalized by the normalization layer. Next, we linearly project the normalized sequence to the $\mathbf{x}$ and $\mathbf{k}$ with dimension size S. Then, we process the $\mathbf{x}$ from the forward and backward directions. For each direction, we first apply the 1-D convolution to the $\mathbf{x}$ and get the $\mathbf{x}'_o$. We then linearly project the $\mathbf{x}'_o$ to the $\mathbf{B}_o$, $\mathbf{C}_o$, $\mathbf{\Delta}_o$, respectively. The

$\mathbf{\Delta}_o$ is then used to transform the $\overline{\mathbf{A}}_o$, $\overline{\mathbf{B}}_o$, respectively. Finally, we compute the $\mathbf{y}_{forward}$ and $\mathbf{y}_{backward}$ through the SSM. The $\mathbf{y}_{forward}$ and $\mathbf{y}_{backward}$ are then gated by the $\mathbf{k}$ and added together to get the output token sequence $\mathbf{z}_l$.

In algorithm 3, L denotes the number of blocks, D denotes the hidden state dimension, S denotes the expanded state dimension, and H denotes the SSM dimension.

Algorithm 3: Vim Block

Require: token sequence $\mathbf{z}_{l-1} : (B, N, D)$

Ensure: token sequence $\mathbf{z}_l : (B, N, D)$

// normalize the input sequence $\mathbf{z}'_{l-1}$

$\mathbf{z}'_{l-1} : (B, N, D) \leftarrow \mathbf{Norm}(\mathbf{z}_{l-1})$

$\mathbf{x} : (B, N, S) \leftarrow \mathbf{Linear}^{\mathbf{x}}(\mathbf{z}'_{l-1})$

$\mathbf{k} : (B, N, S) \leftarrow \mathbf{Linear}^{\mathbf{k}}(\mathbf{z}'_{l-1})$

// process with different direction

for o in {forward, backward} **do**

  $\mathbf{x}'_o : (B, N, S) \leftarrow \mathbf{SiLU}(\mathbf{Conv1d}_o(\mathbf{x}))$

  $\mathbf{B}_o : (B, N, H) \leftarrow \mathbf{Linear}^{\mathbf{B}}_o(\mathbf{x}'_o)$

  $\mathbf{C}_o : (B, N, H) \leftarrow \mathbf{Linear}^{\mathbf{C}}_o(\mathbf{x}'_o)$

  // softplus ensures positive $\mathbf{\Delta}_o$

  $\mathbf{\Delta}_o : (B, N, S) \leftarrow \log(1 + \exp(\mathbf{Linear}^{\mathbf{\Delta}}_o(\mathbf{x}'_o) + \mathbf{Parameter}^{\mathbf{\Delta}}_o))$

  // shape of $\mathbf{Parameter}^{\mathbf{A}}_o$ is (S, H)

  $\overline{\mathbf{A}}_o : (B, N, S, H) \leftarrow \mathbf{\Delta}_o \bigotimes \mathbf{Parameter}^{\mathbf{A}}_o$

  $\overline{\mathbf{B}}_o : (B, N, S, H) \leftarrow \mathbf{\Delta}_o \bigotimes \mathbf{B}_o$

  // initialize h_o and $\mathbf{y}_o$ with 0

  $h_o : (B, S, H) \leftarrow \text{zeros } (B, S, H)$

  $\mathbf{y}_o : (B, N, S) \leftarrow \text{zeros } (B, N, S)$

  // SSM recurrent

  for i in {0, ..., N-1} **do**

    $h_o = \overline{\mathbf{A}}_o[:, i, :, :] \bigodot h_o + \overline{\mathbf{B}}_o[:, i, :, :] \bigodot \mathbf{x}'_o[:, i, :, None]$

    $\mathbf{y}_o[:, i, :] = h_o \bigotimes \mathbf{C}_o[:, i, :]$

  end for

end for

// get gated $\mathbf{y}$

$\mathbf{y}'_{forward} : (B, N, S) \leftarrow \mathbf{y}_{forward} \bigodot \mathbf{SiLU}(\mathbf{k})$

$\mathbf{y}'_{backward} : (B, N, S) \leftarrow \mathbf{y}_{backward} \bigodot \mathbf{SiLU}(\mathbf{k})$

// residual connection

$\mathbf{z}_l : (B, N, D) \leftarrow \mathbf{Linear}^{\mathbf{z}}(\mathbf{y}'_{forward} + \mathbf{y}'_{backward}) + \mathbf{z}_{l-1}$

Return: $\mathbf{z}_l$

References

[1] C. Chen, M. Y. Lu, D. F. Williamson, T. Y. Chen, A. J. Schaumberg, F. Mahmood, Fast and scalable search of wholeslide images via self-supervised deep learning, Nature Biomedical Engineering 6 (12) (2022) 1420-1434.

[2] M. Y. Lu, D. F. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, F. Mahmood, Dataefficient and weakly supervised computational pathology on whole-slide images, Nature Biomedical Engineering 5 (6) (2021) 555-570.

[3] R. J. Chen, M. Y. Lu, D. F. Williamson, T. Y. Chen, J. Lipkova, M. Shaban, M. Shady, M. Williams, B. Joo, Z. Noor, et al., Pan-cancer integrative histology-genomic analysis via multimodal deep learning, Cancer Cell (2022).

[4] R. J. Chen, C. Chen, Y. Li, T. Y. Chen, A. D. Trister, R. G. Krishnan, F. Mahmood, Scaling vision transformers to gigapixel images via hierarchical self-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 16144–16155.

[5] R. J. Chen, R. G. Krishnan, Selfsupervised vision transformers learn visual concepts in histopathology, Learning Meaningful Representations of Life, NeurIPS 2021 (2021).

[6] R. M. French, Catastrophic forgetting in connectionist networks, Trends in cognitive sciences 3 (4) (1999) 128-135.

[7] I. J. Goodfellow, M. Mirza, X. Da, A. C. Courville, Y. Bengio, An empirical investigation of catastrophic forgeting in gradient-based neural networks, in: 2nd International Conference on Learning Representations (ICLR), 2014.

[8] A. Chaudhry, P. K. Dokania, T. Ajanthan, P. H. Torr, Riemannian walk for incremental learning: Understanding forgetting and intransigence, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 532-547.

[9] Y. Huang, W. Zhao, S. Wang, Y. Fu, Y. Jiang, L. Yu, Conslide: Asynchronous hierarchical interaction transformer with breakup-reorganize rehearsal for con-

tinual whole slide image analysis, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 21349-21360.

[10] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Advances in neural information processing systems 30 (2017).

[11] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE transactions on pattern analysis and machine intelligence 44 (7) (2021) 3366–3385.

[12] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, icarl: Incremental classifier and representation learning, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 2001–2010.

[13] S. Yan, J. Xie, X. He, Der: Dynamically expandable representation for class incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3014–3023.

[14] A. Douillard, Y. Chen, A. Dapogny, M. Cord, Plop: Learning without forgetting for continual semantic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 4040–4050.

[15] A. Maracani, U. Michieli, M. Toldo, P. Zanuttigh, Recall: Replay-based continual learning in semantic segmentation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 7026–7035.

[16] L. Wang, X. Zhang, H. Su, J. Zhu, A comprehensive survey of continual learning: Theory, method and application, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (8) (2024) 5362–5383.

[17] M. L. Jie Hao, Kaiyi Ji, Bilevel coreset selection in continual learning: A new formulation and algorithm, in: Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[18] Y. Zheng, Z. Jiang, F. Xie, J. Shi, H. Zhang, J. Huai, M. Cao, X. Yang, Diagnostic regions attention network (dra-net) for histopathology wsi recommendation and retrieval, IEEE transactions on medical imaging 40 (3) (2021) 1090–1103.

[19] Y. Zheng, Z. Jiang, H. Zhang, F. Xie, J. Shi, Tracing diagnosis paths on histopathology wsis for diagnostically relevant case recommendation, in: Medical Image Computing and ComputerAssisted Intervention, 2020, pp. 459-469.

[20] Y. Zheng, Z. Jiang, J. Shi, F. Xie, H. Zhang, W. Luo, D. Hu, S. Sun, Z. Jiang, C. Xue, Encoding histopathology whole slide images with locationaware graphs for diagnostically relevant regions retrieval, Medical Image Analysis (2022) 102308.

[21] Y. Ma, Z. Jiang, H. Zhang, F. Xie, Y. Zheng, H. Shi, Y. Zhao, J. Shi, Generating region proposals for histopathological whole slide image retrieval, Computer methods and programs in biomedicine 159 (2018) 1–10.

[22] X. Shi, M. Sapkota, F. Xing, F. Liu, L. Cui, L. Yang, Pairwise based deep ranking hashing for histopathology image classification and retrieval, Pattern Recognition 81 (2018) 14–22.

[23] D. Hu, Z. Jiang, J. Shi, F. Xie, K. Wu, K. Tang, M. Cao, J. Huai, Y. Zheng, Histopathology language-image representation learning for fine-grained digital pathology cross-modal retrieval, Medical Image Analysis 95 (2024) 103163.

[24] L. Caccia, R. Aljundi, N. Asadi, T. Tuytelaars, J. Pineau, E. Belilovsky, New insights on reducing abrupt representation change in online continual learning, in: The Tenth International Conference on Learning Representations (ICLR), 2022.

[25] A. Chaudhry, M. Ranzato, M. Rohrbach, M. Elhoseiny, Efficient lifelong learning with A-GEM, in: 7th International Conference on Learning Representations (ICLR), 2019.

[26] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, Advances in neural information processing systems 33 (2020) 15920–15930.

[27] X. Zhu, Z. Jiang, K. Wu, J. Shi, Y. Zheng, Lifelong histopathology whole slide image retrieval via distance consistency rehearsal, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2024, pp. 274–284.

[28] Z. Borsos, M. Mutny, A. Krause, Coresets via bilevel optimization for continual learning and streaming, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 14879–14890.

[29] Z. Huang, F. Bianchi, M. Yuksekgonul, T. J. Montine, J. Zou, A visual-language foundation model for pathology image analysis using medical twitter, Nature Medicine (2023) 1–10.

[30] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, X. Wang, Vision mamba: efficient visual representation learning with bidirectional state space model, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024.

[31] S. Yang, Y. Wang, H. Chen, Mambamil: Enhancing long sequence modeling with sequence reordering in computational pathology, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2024, pp. 296–306.

[32] Y. Huang, W. Zhao, Y. Fu, L. Zhu, L. Yu, Unleash the power of state space model for whole slide image with local aware scanning and importance resampling, IEEE Transactions on Medical Imaging (2024).

[33] C. Spearman, The proof and measurement of association between two things, The American journal of psychology 100 (3/4) (1987) 441–471.

[34] M. G. Kendall, A new measure of rank correlation, Biometrika 30 (1/2) (1938) 81–93.

[35] Z. Li, D. Hoiem, Learning without forgetting, IEEE transactions on pattern analysis and machine intelligence 40 (12) (2017) 2935–2947.

[36] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proceedings of the national academy of sciences 114 (13) (2017) 3521–3526.

[37] X. Wang, Y. Du, S. Yang, J. Zhang, M. Wang, J. Zhang, W. Yang, J. Huang, X. Han, Retccl: Clustering-guided contrastive learning for whole-slide image retrieval, Medical image analysis 83 (2023) 102645.

[38] N. Angelescu, H. G. Coanda, I. Caciula, C. Dragoi, F. Albu, Sql query optimization in content based image retrieval systems, in: 2016 International Conference on Communications (COMM), IEEE, 2016, pp. 395–398.