Effective Data Selection and Replay for Unsupervised Continual Learning

Hanmo LIU[1], Shimin DI[2], Haoyang LI[2], Shuangyin LI[3], Lei CHEN[1,2], Xiaofang ZHOU[2]
[1]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
[2]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[3]South China Normal University, Guangzhou, China
{hliubm,sdiaa,hlicg}@connect.ust.hk, shuangyinli@scnu.edu.cn, leichen@hkust-gz.edu.cn, zxf@cse.ust.hk

2024 IEEE 40th International Conference on Data Engineering (ICDE) | 979-8-3503-1715-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDE60146.2024.00119

***Abstract*—Recently, continual learning (CL) has attracted much attention due to its widespread applications in the real world. Given a set of data sets sequentially, continual learning aims to achieve good performance on the new data sets while avoiding deterioration in performance on the old data sets. Despite the success, most CL models follow the supervised setting, which limits their potential in data scarcity cases. Thus, some pioneering works study unsupervised CL (UCL) to discuss what CL tricks suit the unsupervised setting. However, their advancements lack in-depth analysis of the characteristics of UCL, especially the lack of attention to the use of old data. We identify that using old data sets is essential for improving the UCL model performance while existing works ignore them. Unfortunately, given a limited data storage budget, it is a non-trivial task to select representative data and effectively replay them without label assistance. To further improve the UCL performance, we present a new method in this paper, named Effective Data Selection and Replay (EDSR) for UCL. Specifically, we analyze that entropy can be an effective data selection metric, where representative data usually exhibit the highest entropy in the representation space. Then, to balance the model stability for old data and the plasticity for new data, we adopt a strategy of replaying those stored representative data with a noise-enhanced knowledge distillation process. The empirical study demonstrates the outstanding performance of EDSR on benchmark computer vision data sets. Especially, EDSR shows strong resistance to forgetting old data knowledge while maintaining high accuracy. The implementation is publicly available at https://github.com/LeeJarvis996/edsr_project/tree/main/EDSR.**

***Index Terms*—continual learning, self-supervised learning, image classification**

I. Introduction

In recent years, continual learning has attracted increasing attention due to its practical applications in many real-world scenarios [1], [2]. Continual learning aims to continuously learn new datasets while maintaining knowledge of old ones, particularly when the old datasets are no longer accessible. However, without access to the old datasets, the model can become biased towards the new data and forget the old knowledge, which is known as the Catastrophic Forgetting problem [3]. For example, an autonomous driving system must adapt to the changing driving environment. However, the past driving environments can still influence the system, and thus it needs continual learning to maintain its effectiveness in both old and new environments. To alleviate this issue, many promising methods have been proposed in recent decades [4]–[20]. Among the different methods, storing and replaying old data has been found to be the most effective [10], [11], [18], [21]. This is because the old data can help the model retain the knowledge of old tasks and alleviate the restrictions on learning new data sets. However, due to the limited storage capacity and practical constraints, there is usually a limit on the number of stored data. Therefore, existing works mainly develop two techniques: 1) *Data selection:* This involves selecting representative data from the old data sets while avoiding outliers or excessively similar data [10], [22], [23]. 2) *Data replay:* This involves repeatedly replaying the stored data to the model while avoiding overfitting to the stored data and allowing flexibility for learning new data. [11], [14], [18].

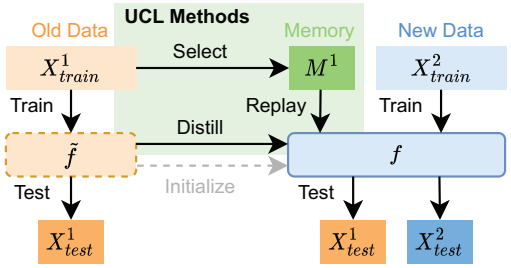


Fig. 1. An illustration of unsupervised continual learning (UCL) and approaches to prevent forgetting. All the data sets are unlabeled.

Despite the success of continual learning, the mainstream methods still follow the supervised setting, which requires a large amount of labeled data that is difficult to obtain in real-life situations. Unfortunately, many potential applications of continual learning receive continuously incoming unlabeled data after deploying the model. For instance, after a visual robot is deployed, it needs to continually update itself with new unlabeled images. To broaden the application scenarios of continual learning methods, unsupervised continual learning (UCL) is an important topic to explore. As illustrated in Fig. 1, UCL models continuously learn from unlabeled new data sets without accessing the old ones, and are expected to perform well on all learned data sets. However, most existing supervised continual learning (SCL) methods are no longer applicable to the unsupervised setting [24], facing difficulties in lacking label feedback and ineffectiveness.

Several unsupervised continual learning (UCL) methods have been proposed to handle the unsupervised setting, which can be broadly categorized into two types: VAE-based UCL and CSSL-based UCL. VAE-based UCL methods such as

2375-026X/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDE60146.2024.00119

VASE [25] and CURL [26] are built on Variational Autoencoders (VAE) [27]. They use VAE to generate old data to prevent forgetting and expand the representation space to accommodate new knowledge. However, due to the simple structure of VAE, these methods show a significant drop in performance on complex data sets [24], such as high-resolution images. In recent years, contrastive self-supervised learning (CSSL) [28]–[32] has emerged as a more powerful unsupervised learning paradigm, showing its ability to learn knowledge comparable to supervised learning. The core idea of CSSL is to make representations of two augmented views of the same input close to each other while keeping representations from different images at a distance. Recent UCL methods have been built on CSSL and have achieved better performance, such as LUMP [24] and CaSSLe [33]. LUMP [24] is the first method to introduce a CSSL loss to UCL. It randomly selects data to store and replays them using the mix-up trick [34], which corresponds to the select-replay path in Fig. 1. CaSSLe [33] proposes a knowledge distillation mechanism [35] to bring old knowledge from the past model to the current one, which is the distillation path in Fig. 1.

Although CSSL-based UCL methods have achieved satisfactory results, they hold a primitive attitude toward the usage of old data. For instance, LUMP stores random old data, which cannot exclude outliers or too-similar data. Additionally, mixing old data with new data may result in unstable optimization and hinders performance, despite its ability to prevent overfitting. Similarly, CaSSLe relies solely on the past model and neglects old data. Because the model unavoidably loses old knowledge when learning new data, CaSSLe is unsuitable for learning longer incremental data sets. Intuitively, we may improve these methods as long as we can better utilize the old data. But towards the effective usage of the old data, there are two critical challenges to solve. The first challenge is how to select the representative data subset from the original data set without label feedback. The second one is how to balance the stability for the old knowledge and the plasticity for the new knowledge while replaying the stored data to the model.

Regarding the above two challenges in exploring old data, we design a novel method called *E*ffective *D*ata *S*election and *R*eplay (EDSR) for unsupervised continual learning, which is a strong integration of the two approaches in Fig. 1. To select representative old data without the feedback from labels, we first analyze that entropy is an effective evaluation metric and theoretically explain how to reduce the problem into a simple one for efficient optimization. To replay the stored data more effectively, we adopt a strategy to replay the stored data by distilling knowledge from the past model to the current one. Additionally, we enhance the replaying effectiveness by adding data-related noises to the data representations during distillation. This technique enables the model to learn a more diverse and general set of representations. As a result, the updated model gains the flexibility to adjust the old representation space for learning the new data. The major contributions of EDSR are summarized as follows:

- In this paper, we identify that the main reason hindering the performance of existing unsupervised continual learning models is insufficient exploration of old data sets. Therefore, we present a novel method to effectively select and replay old data for unsupervised continual learning, named EDSR.
- We solve the challenge of selecting representative data subset without label feedback by proposing a novel entropy-based data selection method.
- We address the stability-plasticity trade-off problem in data replay by indirectly learning the stored data via distillation and adding data-related noise to the data representation.
- Our empirical study demonstrates the outstanding performance of EDSR on benchmark computer vision and tabular datasets In particular, EDSR shows strong resistance to forgetting old knowledge while maintaining high accuracy.

TABLE I
SUMMARY ON IMPORTANT NOTATIONS

Notations	Meanings
$x^i \in X^i$	The sample x^i from the set X^i in i-th increment.
$y^i \in Y^i$	The label y^i from the set Y^i in i-th increment.
M^i	$M^i \subset X^i$ is the memory set for selected data after i-th increment, where $\|M^i\| < s$.
$x^m \in \bigcup_i M^i$	The sample x^m from the whole memory set.
n	n specifically indexes the new data, $x^n \in X^n$.
$\tilde{f}(\cdot), \hat{f}(\cdot)$	The model $f(\cdot)$ before and after learning X^n.
$\mathbf{x} \in \mathbb{R}^d$	d-dimensional vector representation of x.
$\mathbf{x}_1, \mathbf{x}_2$	The representations of augmentations x_1, x_2 from x.
$\tilde{\mathbf{x}}, \hat{\mathbf{x}}$	The representations of x extracted by $\tilde{f}(\cdot)$ and $\hat{f}(\cdot)$

II. PRELIMINARY AND RELATED WORKS

In this section, we first introduce the contrastive self-supervised learning in Sec. II-A, then introduce the supervised and unsupervised continue learning in Sec. II-B. The important notations in this paper are listed in Tab. I. Generally, we denote the sample with lowercase characters (e.g., x) and the set with uppercase characters (e.g., $x \in X$). Vector and matrix are denoted with bold lowercase (e.g., $\mathbf{x}$) and bold uppercase $\mathbf{A}$. The subscripts are for the functionality and the superscripts indicate the source set.

A. Contrastive Self-Supervised Learning

In this paper, we mainly focus on training the model without labeled data, i.e., unsupervised learning. In the past decades, various promising approaches have been proposed to represent data samples from unlabeled data sets, including Variational Autoencoders (VAE) [27], Generative Adversarial Networks (GAN) [36], and Contrastive Self-Supervised Learning (CSSL) [37]. Among these approaches, CSSL has emerged as the most effective approach [28]–[32], [38], [39]. CSSL aims to make the representations of two positive samples close to each other while pushing representations from negative ones away. Given a set of training samples X, CSSL first selects one sample $x \in X$ and its negative version $x^- \in X$, where x^- is regarded as different from x significantly (e.g., x and x^- belong to different class). Then, CSSL usually augments x into different views x_1 and x_2, which are positive samples of x. Let

$\mathbf{x} := f(x)$ denote the vector representation learned from the CSSL model $f(\cdot)$. The core idea of CSSL is to maximize the similarity between $\mathbf{x}_1$ and $\mathbf{x}_2$, while minimizing the similarity between $\mathbf{x}_j$ with $\mathbf{x}^-$ (j is 1 or 2). Formally, the loss of a CSSL model can be formulated as:

$$\mathcal{L}_{css}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}^-) = -Sim(\mathbf{x}_1, \mathbf{x}_2) + Sim(\mathbf{x}_j, \mathbf{x}^-), \quad (1)$$

where $Sim(\cdot, \cdot)$ measures the similarity of two inputs, e.g., cosine similarity. As in Eq. (1), classic data augmentation methods will more focus on how to create effective positive samples $\mathbf{x}_1$ and $\mathbf{x}_2$, while CCSL models tend to design better contrastive loss $\mathcal{L}_{css}(\cdot)$.

1) Data Augmentation for $\mathbf{x}_1$ and $\mathbf{x}_2$: The augmentation is a function $T(\cdot; O)$ composed of a set of transformation operations O [40]. Each operation $o \in O$ transforms the sample by a specific rule, and is parameterized by the possibility of being selected and the magnitude of transformation strength. For example, an example set of operations is $\{\text{cutout}, \text{rotate}, \text{flip}, \text{colorContrast}, \text{resize}\} \subset O$ [31], [38], [41]. Supposing that the selected subset of operations is $O_{sub} = \{o_k(\cdot) : 1 \leq k \leq |O_{sub}|\} \subset O$, then the augmentation is to sequentially apply these operations to the sample x:

$$x_{(1)} = o_1(x),\ x_{(k)} = o_k(x_{(k-1)}), \forall k \leq |O_{sub}|, \quad (2)$$

and eventually $x_j = T(x; O_{sub}) = x_{(|O_{sub}|)}$ ($j \in \{1, 2\}$). In recent years, how to design an effective augmentation method to boost the model performance becomes a hot topic [40], [42]–[45]. However, this is not our focus and we use the same augmentation methods as in SimSiam [31].

2) CSSL Loss $\mathcal{L}_{css}(\cdot)$: SimCLR [28] and MoCo [29] are the first CSSL methods to achieve comparable unsupervised performance to supervised learning, but they require large training batches and significant run-time memory [31], which limits their application. Recent CSSL methods, such as SimSiam [31], BYOL [32], and BarlowTwins [38], simplify training requirements by comparing only the representations from positive samples $\mathbf{x}_1, \mathbf{x}_2$ and achieve state-of-the-art performance. SimSiam [31] constructs its encoder network $f(\cdot)$ with a convolutional neural network (CNN) model [46] and a multi-layer perceptron (MLP) [47]. Different from the classic formulation in Eq. (1) that directly compares $\mathbf{x}_1$ and $\mathbf{x}_2$, it uses one representation to predict the other, i.e., predicting $\mathbf{x}_2$ based on $\mathbf{x}_1$ and vice versa. Following this idea, Simsiam adds a representation-predictor $h(\cdot)$ to the encoder and its objective is defined as:

$$\mathcal{L}_{css}(\mathbf{x}_1, \mathbf{x}_2) = -\frac{1}{2}[Sim(h(\mathbf{x}_1), \text{sg}(\mathbf{x}_2)) + Sim(h(\mathbf{x}_2), \text{sg}(\mathbf{x}_1))], \quad (3)$$

where $\text{sg}(\cdot)$ is the stop gradient operation that blocks backward propagation through the corresponding input and makes its input a prediction target.

BarlowTwins [38] only uses the encoder network $f(\cdot)$. It defines a cross-correlation matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$ between representations of two augmented views of the same input batch. It strengthens the correlation between two augmentations of the same input while reducing the correlation between different inputs, which is achieved by making $\mathbf{C}$ an identity matrix. The objective of BarlowTwins is defined as follows:

$$\mathcal{L}_{css}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\alpha=1}^{d}(1 - \mathbf{C}_{\alpha,\alpha})^2 + \lambda \cdot \sum_{\alpha=1}^{d}\sum_{\beta} \mathbf{C}_{\alpha,\beta}^2, \quad (4)$$

where d is the dimension of the representation space, $\beta \in \{1, \cdots, d\} \setminus \alpha$, λ is a weighting parameter, $Batch$ is the batch size, and $\mathbf{C}_{\alpha,\beta} = \frac{\sum_{b=1}^{Batch} \mathbf{x}_1^{(b,\alpha)} \mathbf{x}_2^{(b,\beta)}}{\sqrt{\sum_{b=1}^{Batch} (\mathbf{x}_1^{(b,\alpha)})^2} \sqrt{\sum_{b=1}^{Batch} (\mathbf{x}_2^{(b,\beta)})^2}}$ is the sum of all the similarities between the α dimension of $\mathbf{x}_1$ and the β dimension of $\mathbf{x}_2$ in the batch.

B. Continual Learning

1) Supervised Continual Learning: Continual learning [1], [2] aims to train a model that can learn a sequence of different datasets, while optimizing on new data and preserving knowledge of old data. It is similar to transfer learning [48]–[50] and dynamic learning [51], [52], but has the unique requirement of not accessing old data. Formally, the supervised continual learning (SCL) problem is defined as:

Definition 1 (Supervised Continual Learning Problem): Given a sequence of n datasets $\{(X^1, Y^1), ..., (X^n, Y^n)\}$, let the models before and after learning the new data (X^n, Y^n) be denoted as $\tilde{f}(\cdot)$ and $\hat{f}(\cdot)$, respectively. Then, the SCL problem is to minimize the supervised objective on (X^n, Y^n) while maintaining the model's performance on old data $\{(X^i, Y^i)\}_{i<n}$:

$$\hat{f} = \arg\min_{f} \sum_{(x^n, y^n)} \mathcal{L}_{sup}(f(x^n), y^n), \quad (5)$$

$$s.t. \sum_{(x^i, y^i)} |\mathcal{L}_{sup}(\hat{f}(x^i), y^i) - \mathcal{L}_{sup}(\tilde{f}(x^i), y^i)| \leq \delta, \forall i < n, \quad (6)$$

where $(x^i, y^i) \in (X^i, Y^i)$, $1 \leq i \leq n$, $\mathcal{L}_{sup}(\cdot)$ is the supervised learning loss, e.g., cross-entropy loss, and δ is the error threshold for forgetting.

Through solving Def. 1, SCL aims to optimize the performance of the new model $\hat{f}(\cdot)$ on the new data (X^n, Y^n), while guaranteeing its effectiveness on the old data to be within an error bar δ compared with the old model $\tilde{f}(\cdot)$. However, the old data sets are inaccessible during the learning of X^n under the continual setting. Thus, the constraint in Eq. (6) cannot be satisfied directly, which results in the Catastrophic Forgetting problem. To address this problem, several categories of continual learning methods have been proposed, including regularization-based, model adaptation, and memory-based methods. Regularization-based methods [16]–[19], [21], [53]–[55] restrict the parameter change during learning X^n to prevent forgetting. Model adaptation methods [4], [7]–[9], [20], [56]–[58] use extra parameters to accommodate increasingly more knowledge, while memory-based methods [10]–[14], [59], [60] first store or generate old data, then review them to reduce forgetting. Among these three categories, memory-based methods generally achieve the best performance. This shows that the old data is the strongest

anchor for recovering the old knowledge. Besides, the methods using knowledge distillation [35] techniques have achieved success [10], [16], [18], [19], [21], [55], [58], [60] by transferring the old knowledge from a past model to the current one. The medium of distillation can be the prediction [16], [55], feature [21], [60], or model parameters [19].

2) Unsupervised Continual Learning: In the unsupervised setting, labels are absent for input data and the data sets become $\{X^1, ..., X^n\}$. This makes most SCL methods no longer applicable, especially for those SCL methods relying on memory. Because without label feedback, it is challenging to identify representative data and reuse them. For example, iCarl [10] requires the mean prediction of each label to select important data, while GEM [11] relies on the gradients from old classes to redirect those from new classes to avoid forgetting. It has also been studied that SCL methods may face the performance collapse issue when adapted to unsupervised cases [24], [33].

In this subsection, we mainly discuss the CSSL-based UCL as it is the most effective unsupervised continual learning method. When the data label y is unavailable, CSSL-based UCL proposes to replace the supervised loss $\mathcal{L}_{sup}(\cdot)$ in Eq. (5) and Eq. (6) by the CSSL loss $\mathcal{L}_{css}(\cdot)$. For simplicity, $\mathbf{x}^i := f(x^i)$ is the representation of $x^i \in X^i$. Besides, $\hat{\mathbf{x}}^i_j$ and $\tilde{\mathbf{x}}^i_j$, $j \in \{1, 2\}$, are the representations extracted by $\hat{f}(\cdot)$ and $\tilde{f}(\cdot)$. The UCL problem is formally formulated as:

Definition 2 (Unsupervised Continual Learning Problem): Given a sequence of n unlabeled datasets $\{X^1, ..., X^n\}$, let the models before and after learning X^n be denoted as $\tilde{f}(\cdot)$ and $\hat{f}(\cdot)$, respectively. Then, the CSSL-based UCL problem aims to minimize the contrastive self-supervised objective on the new data X^n while maintaining the model's performance on old data $\{X^i\}_{i<n}$:

$$\hat{f} = \arg\min_{f} \sum_{x^n} \mathcal{L}_{css}(\mathbf{x}^n_1, \mathbf{x}^n_2), \tag{7}$$

$$s.t., \sum_{x^i} |\mathcal{L}_{css}(\hat{\mathbf{x}}^i_1, \hat{\mathbf{x}}^i_2) - \mathcal{L}_{css}(\tilde{\mathbf{x}}^i_1, \tilde{\mathbf{x}}^i_2)| \leq \delta, \forall i < n, \tag{8}$$

where $x^i \in X^i$, $i \leq n$, $\mathcal{L}_{css}(\cdot)$ is the CSSL loss, and δ is the error threshold for forgetting.

Upon solving Def. 2, CSSL-based UCL uses $\mathcal{L}_{css}(\cdot)$ to learn X^n in Eq. (7) and maintain the old knowledge in Eq. (8) without acquiring label information. However, it is still hard to guarantee the error bar δ under this unsupervised setting, which requires the UCL methods to devise novel approaches for continual learning. Currently, several CSSL-based UCL methods have been proposed, including memory-based methods LUMP [24] and Lin et al. [61], regularization-based methods CaSSLe [33] and PFR [62]. LUMP maintains one fixed memory buffer M that stores randomly selected old data. When learning a sample $x^n \in X^n$ of new data, LUMP draws a stored old data $x^m \in M$ from memory and synthesizes an input $\bar{x}^n$ by mixing x^n up with x^m, which is formulated as $\bar{x}^n = \omega x^n + (1-\omega)x^m, \omega \in (0, 1)$. Afterwards, LUMP sends $\bar{x}^n$ into $\mathcal{L}_{css}(\cdot)$, which gives $\mathcal{L}_{css}(\bar{\mathbf{x}}^n_1, \bar{\mathbf{x}}^n_2)$, to solve Def. 2. In such a way, the forgetting issue is eased by simultaneously learning both old and new data. Lin et al. store data based on k-means and maintain the representation distances between stored and new data to prevent forgetting. However, both methods lack theoretical analysis of how the stored data can be representative.

For regularization-based methods, PFR and CaSSLe design a knowledge distillation [35] loss $\mathcal{L}_{dis}(\cdot)$ based on $\mathcal{L}_{css}(\cdot)$. The mechanism of knowledge distillation is that, given the same input data, align the outputs from the current model $f(\cdot)$ and the old model $\tilde{f}(\cdot)$ to transfer the old knowledge from $\tilde{f}(\cdot)$ to $f(\cdot)$ and prevent forgetting. When based on $\mathcal{L}_{css}(\cdot)$, however, direct alignment is ineffective due to the high-dimensional representation space. So that before alignment, both methods add a projector $p_{dis}(\cdot)$ that projects the output from the current representation space to the old one in order to improve the effectiveness. Formally, $\mathcal{L}_{dis}(\cdot)$ is defined as:

$$\mathcal{L}_{dis}(\mathbf{x}^n_1, \tilde{\mathbf{x}}^n_1) = \mathcal{L}_{css}(p_{dis}(\mathbf{x}^n_1), \tilde{\mathbf{x}}^n_1), \tag{9}$$

where the concrete definition of $\mathcal{L}_{dis}(\cdot)$ varies with different $\mathcal{L}_{css}(\cdot)$. From Eq. (9), given an augmented input x^n_1, $\mathbf{x}^n_1$ is first projected to the old representation space by $p_{dis}(\cdot)$, then aligned with $\tilde{\mathbf{x}}^n_1$ to transfer knowledge from $\tilde{f}(\cdot)$ to $f(\cdot)$. In practice, $\mathcal{L}_{dis}(\mathbf{x}^n_2, \tilde{\mathbf{x}}^n_2)$ is also added to distillation process.

Besides the aforementioned methods, there are also works [63], [64] focusing on online UCL, where the inputs are an unlabeled data stream. Our setting is different from this stream setting by that, ours receives datasets, and each dataset can be repeatedly learned until optimization.

III. Methods

As discussed in Sec. I and Sec. II, accessing and reusing the old data is of great importance to continual learning, and many SCL methods achieve success through designing effective data selection and replay mechanisms. However, existing UCL works tend to ignore storing and replaying old data with an effective way. Therefore, our method EDSR will bring up this gap. When concerning about data selection and replay, the UCL problem definition is transformed as follows:

Definition 3 (EDSR's UCL Problem): Given a sequence of n unlabeled datasets $\{X^1, ..., X^n\}$, suppose that some data samples $\{M^i_*\}_{i<n}$ have been stored from $\{X^i\}_{i<n}$, as old data $\{X^i\}_{i<n}$ is unavailable for learning X^n under the setting of continual learning. Let the models before and after learning X^n be denoted as $\tilde{f}(\cdot)$ and $\hat{f}(\cdot)$, respectively. Then, EDSR's UCL Problem aims to train the model on the data $\{M^i_*\}_{i<n}$ and X_n to maximize its performance on all n data increments, and select informative data subsets $M^n_* \subset X^n$ for next increment. The problem is formally formulated as follows:

$$(1)\ \hat{f} = \operatorname{argmin}_{\tilde{f}} \sum_{x^n} \mathcal{L}_{css}(\mathbf{x}^n_1, \mathbf{x}^n_2) + \sum_{x^m} \mathcal{L}_{rpl}(\mathbf{x}^m_1, \mathbf{x}^m_2), \tag{10}$$

$$(2)\ M^n_* = \arg \max_{M^n \subset X^n, |M^n| \leq s} inf(X^n, M^n), \tag{11}$$

where the new data sample x^n is from the new data increment X^n, the sample x^m is from the past memorized data $\{M^i_*\}_{i<n}$,

$\mathcal{L}_{css}(\cdot)$ is the contrastive self-supervised objective for learning new data, $\mathcal{L}_{rpl}(\cdot)$ is the replay objective for learning stored data, $inf(\cdot,\cdot)$ calculates the information of X^n contained in M^n, and s is the memory budget.

Upon solving Def. 3, EDSR tries to address the UCL problem in two steps. First, training the model $\tilde{f}$ by constrasting the new data $x^n \in X^n$ with the contrastive loss $\mathcal{L}_{css}(\cdot)$ and replaying the stored data $x^m \in \{M_*^i\}_{i<n}$ with the replay loss $\mathcal{L}_{rpl}(\cdot)$. Second, EDSR assumes that the more information of $\{X^i\}_{i<n}$ contained in $\{M_*^i\}_{i<n}$, the better performance the model achieves through replaying them. Thus, $M_*^n \subset X^n$ is selected for next data increment.

Due to the absence of labels, effective data selection in Eq. (11) and data replay in Eq. (10) are both non-trivial tasks. Thus, in Sec. III-A, we introduce an entropy-based data selection method with theoretical validation to address the data selection challenge. Then, in Sec. III-B, we propose to replay the data through noise-enhanced distillation to solve the data replay challenge. Finally, the model framework is presented in Sec. III-C.

A. Entropy-Based Data Selection

In this subsection, we build up our data selection method for the UCL problem. As reflected in Eq. (11), after learning new data X^n, the goal of data selection is to identify the most informative subset M^n from X^n under a limited memory budget s, with the expectation that M^n can contain as much knowledge about X^n as possible. From the perspective of information theory [65], it indicates that maximizing the mutual information between M^n and X^n. By noting the mutual information as $MI(\cdot)$, the objective of data selection in Eq. (11) is transformed as follows:

$$M_*^n = \arg\max_{M^n \subset X^n, |M^n| \leq s} MI(X^n, M^n), \tag{12}$$

where s is the memory budget.

However, it is hard to optimize Eq. (12) directly, since it is discrete optimization with constraint and the searching complexity of M^n is $\mathcal{O}(|X^n|!/(s!(|X^n|-s)!)$. To solve this issue, we first transform $MI(X^n, M^n)$ as $MI(X^n, M^n) = H(M^n) - H(M^n|X^n)$ [65], where $H(\cdot)$ denotes the entropy. Since M^n is a subset of X^n, $H(M^n|X^n) = 0$. Thus, maximizing the mutual information in Eq. (12) is equivalent to maximize $H(M^n)$:

$$M_*^n = \arg\max_{M^n \subset X^n, |M^n| \leq s} H(M^n). \tag{13}$$

Intuitively, entropy maximization can be achieved by selecting distant data in the representation space, as distant representations reflect dissimilar images. However, such a heuristic has no guarantee on the effectiveness. The better approach is to obtain an entropy estimation function. Unfortunately, because X^n has only limited samples, directly using it to estimate the entropy distribution of X^n's input space is hard.

In this paper, we propose to use the lossy coding length [65] to evaluate the entropy of the input data. Lossy coding length describes the minimal number of bits that are needed to fully encode the data. The higher entropy the data has, the more encoding bits are required. So that we can maximize Eq. (13) by finding M^n that require largest lossy coding length. Because the representation consolidates the information of its input data and implicitly contains the label knowledge, we will use the lossy coding length of the representations of M^n to evaluate this length more effectively. Let $\hat{\mathbf{M}}^n := \hat{f}(M^n)$ be the representations extracted from M^n by the model optimized on X^n. Then as suggested in [66], [67], the entropy of M^n is defined as follows:

$$H(M^n) = \frac{|M^n|+d}{2} \log \cdot \det(\mathbf{I}_{|M^n|} + \frac{d}{|M^n|\epsilon^2} Cov(\hat{\mathbf{M}}^n)),$$

where $\mathbf{I}_{|M^n|}$ is the identity matrix of dimension $|M^n| \times |M^n|$, d is the representation dimension, $\det(\cdot)$ calculates the determinant of a matrix, $Cov(\mathbf{A}) = \mathbf{A}^T\mathbf{A}$, and ϵ is the decoding error of transforming the representation into bits. From the equation, the entropy of M^n is proportional to the magnitude of representation covariance, which is intuitive since a higher covariance means a larger chaos. Because d, ϵ, and $|M^n|$, are constants, the maximization of entropy in Eq. (13) is equivalent to optimize the variable $Cov(\hat{\mathbf{M}}^n)$. But it is still hard to maximize the determinant of a covariance matrix $\det(\mathbf{I}_{|M^n|} + \frac{d}{|M^n|\epsilon^2} Cov(\hat{\mathbf{M}}^n))$, which requires us to further simplify this formulation.

Recall that in matrix analysis, the determinant has an identical equation that replaces itself with a simple trace calculation $\mathrm{Tr}(\cdot)$, that is $\det \cdot \exp(\mathbf{A}) = \exp \cdot \mathrm{Tr}(\mathbf{A})$ [68]. We reformulate this identical equation as $\det(\mathbf{A}) = \exp \cdot \mathrm{Tr} \cdot \log(\mathbf{A})$ and use it to do the simplification. By using abbreviations $\mu_{M^n} = \frac{|M^n|+d}{2}$ and $\lambda_{M^n} = \frac{d}{|M^n|\epsilon^2}$, the simplification process is as follows:

$$\begin{aligned} H(M^n) &= \mu_{M^n} \log \cdot \det(\mathbf{I}_{|M^n|} + \lambda_{M^n} Cov(\hat{\mathbf{M}}^n)) \\ &= \mu_{M^n} \log \cdot \exp \cdot \mathrm{Tr} \cdot \log(\mathbf{I}_{|M^n|} + \lambda_{M^n} Cov(\hat{\mathbf{M}}^n)) \\ &= \mu_{M^n} \mathrm{Tr} \cdot \log(\mathbf{I}_{|M^n|} + \lambda_{M^n} Cov(\hat{\mathbf{M}}^n)) \\ &\propto \mu_{M^n} \mathrm{Tr}(\mathbf{I}_{|M^n|} + \lambda_{M^n} Cov(\hat{\mathbf{M}}^n)) \\ &\propto \mathrm{Tr}(Cov(\hat{\mathbf{M}}^n)), \end{aligned} \tag{14}$$

where the constants μ_{M^n}, λ_{M^n} and $\mathbf{I}_{|M^n|}$ are omitted in Eq. (14) without affecting the results. Finally, we can optimize the following objective to select representative data:

$$M_*^n = \arg\max_{M^n \subset X^n, |M^n| \leq s} \mathrm{Tr}(Cov(\hat{\mathbf{M}}^n)). \tag{15}$$

Since the trace of a covariance matrix is the sum of its singular values, $\mathrm{Tr}(Cov(\hat{\mathbf{M}}')) < \mathrm{Tr}(Cov(\hat{\mathbf{M}}'')), \forall M' \subset M''$, where M' and M'' are two random memory sets. Thus, the representations of X^n, $\hat{\mathbf{X}}^n := \hat{f}(X^n)$, has the largest entropy, and maximizing Eq. (15) is equivalent to finding a subset from $\hat{\mathbf{M}}^n$ that maintains the highest singular values. Intuitively, based on the meaning of singular values, Eq. (15) means that the most representative data subset is the one that can best reconstruct the representation space of the original data. This equivalence also satisfies our original goal of maximizing the mutual information between M^n and X^n. In practice, we

maximize the sum of singular values of $\hat{\mathbf{M}}^n$ via Principal Component Analysis [69].

B. Noise-Enhanced Data Replay

In this subsection, we introduce how we construct the data replay loss $\mathcal{L}_{rpl}(\cdot)$ in Eq. (10). To achieve an effective data replay, the two major concerns are the stability for the old data and the plasticity for the new data. Naively, the stored data $\{M^i\}_{i<n}$ can be directly replayed via $\mathcal{L}_{css}(\cdot)$. However, this replay method leads to the over-fitting issue and addresses neither of the above concerns. Because $\mathcal{L}_{css}(\cdot)$ requires abundant samples from the same class to learn effective representations, which the limited samples in $\{M^i\}_{i<n}$ cannot satisfy. Recently, there are several SCL methods [23], [70] adapting CSSL techniques to continual learning, for example PASS [23] replays the prototypes of each class with augmentation. Unfortunately, these methods rely on the label feedback and thus are unsuitable for the unsupervised setting.

Therefore, we propose a noise-enhanced data replay loss $\mathcal{L}_{rpl}(\cdot)$ to better strike the balance between stability and plasticity when replaying the limited and unlabeled memory. To avoid the over-fitting issue mentioned above, $\mathcal{L}_{rpl}(\cdot)$ replays the old data via knowledge distillation. Recall that the distillation process is to transfer the old knowledge from the old model $\tilde{f}(\cdot)$ to the current model $f(\cdot)$ via aligning their outputs on the same input. During this process, $\tilde{f}(\cdot)$ is the major source of old knowledge, and the stored data act as the media of knowledge transfer. So that the quantity of stored data can hardly cause negative effect. $\mathcal{L}_{rpl}(\cdot)$ uses the same distillation mechanism as CaSSLe for its state-of-the-art performance.

Besides replaying via distillation, we also propose to add representation-level noises to each sample in the memory during replay, in order to enlarge the transferable old knowledge. This intuition is inspired from the current research that similar samples will have overlapping representations after augmentation [71]. Introducing a perturbation to the representation of a sample can thus relate it to its similar neighbors and broaden the learnable representation space. In particular, this representation-level noise is a d dimensional vector $\boldsymbol{\sigma}$ following standard normal distribution and has a data-dependent magnitude $r(x^m)$. Without loss of generality, we demonstrate the calculation of $r(x^m)$ after selecting M_*^n from X^n. For $x^m \in M_*^n$, $r(x^m)$ is the standard deviation of the representations among the k nearest neighbors (kNN) of x^m from X^n. $r(x^m)$ is defined as follows:

$$r(x^m) = \mathrm{Std}(\{\hat{\mathbf{x}}' : x' \in \mathrm{Nei}(x^m|X^n)\}),$$

where $\mathrm{Nei}(x^m|X^n)$ is the kNN of x^m in X^n, $\mathrm{Std}(\cdot)$ calculates the standard deviation of the representations and $\hat{\mathbf{x}}'$ is extracted by the model $\hat{f}(\cdot)$ optimized on X^n. This data-dependent magnitude $r(x^m)$ helps scale the noise to a meaningful range.

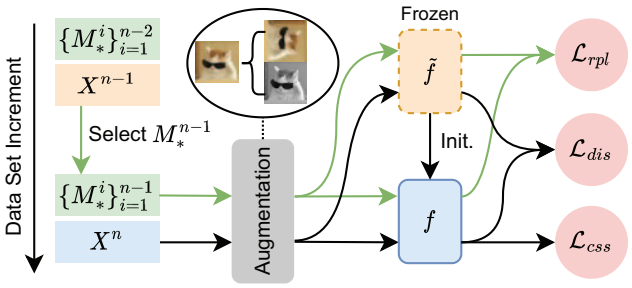


Fig. 2. The framework of EDSR. At the training stage, the model learns X^n with $\mathcal{L}_{css}(\cdot)$ and $\mathcal{L}_{dis}(\cdot)$, and replays $\{M_*^i\}_{i<n}$ with $\mathcal{L}_{rpl}(\cdot)$. After learning X^n, $M_*^n \subset X^n$ is selected based on entropy and stored.

After adding the noise $r(x^m)\cdot\boldsymbol{\sigma}$ to the distillation process, $\mathcal{L}_{rpl}(\cdot)$ is formulated as follows:

$$\begin{aligned}\mathcal{L}_{rpl}(\mathbf{x}_1^m, \tilde{\mathbf{x}}_1^m | r(x^m)) &= \mathcal{L}_{dis}(\mathbf{x}_1^m, \tilde{\mathbf{x}}_1^m + r(x^m)\cdot\boldsymbol{\sigma}) \\ &= \mathcal{L}_{css}(p_{dis}(\mathbf{x}_1^m), \tilde{\mathbf{x}}_1^m + r(x^m)\cdot\boldsymbol{\sigma}),\end{aligned} \tag{16}$$

where x^m is augmented to x_1^m and $\tilde{\mathbf{x}}_1^m$ is the representation extracted by the old model $\tilde{f}(\cdot)$ before learning X^n. During distillation, $\mathbf{x}_1^m$ is aligned with $\tilde{\mathbf{x}}_1^m + r(x^m)\cdot\boldsymbol{\sigma}$, whose noise-enhanced old knowledge is thus distilled into the new model. $\mathcal{L}_{rpl}(\cdot)$ strikes a good balance between stability and plasticity, because noise-enhanced distillation revokes abundant old knowledge for the model stability, while the noise itself also allows the model plasticity for new knowledge.

C. Framework

After introducing how we store representative data in Sec. III-A and how we replay them in Sec. III-B, we now show the integrated framework of our method, EDSR, in Fig. 2. The framework contains two stages, the training stage and the selecting stage.

1) Training Stage: When training the model on the new stage n, $f(\cdot)$ is optimized simultaneously on X^n and $\{M_*^i\}_{i<n}$. For X^n, $\mathcal{L}_{css}(\cdot)$ is applied to acquire new knowledge and $\mathcal{L}_{dis}(\cdot)$ to assist forgetting prevention. On the other hand, $\{M_*^i\}_{i<n}$ is solely trained on $\mathcal{L}_{rpl}(\cdot)$ to further recover the old knowledge while enabling plasticity for new knowledge. The final objective of our model is defined as follows,

$$\begin{aligned}\mathcal{L}(X^n, \bigcup_{i<n} M_*^i) = &\sum_{x^n} \mathcal{L}_{css}(\mathbf{x}_1^n, \mathbf{x}_2^n) \\ &+ \sum_{x^n} \frac{1}{2}(\mathcal{L}_{dis}(\mathbf{x}_1^n, \tilde{\mathbf{x}}_1^n) + \mathcal{L}_{dis}(\mathbf{x}_2^n, \tilde{\mathbf{x}}_2^n)) \\ &+ \sum_{x^m} \frac{1}{2}\mathcal{L}_{rpl}(\mathbf{x}_1^m, \tilde{\mathbf{x}}_1^m | r(x^m)),\end{aligned}$$

where $x^n \in X^n$ and $x^m \in \bigcup_{i<n} M_*^i$.

2) Selecting Stage.: After optimization on X^n, the model enters the selecting stage. At this stage, $\hat{\mathbf{X}}^n$ is first extracted by $\hat{f}(\cdot)$ on X^n, during which X^n are not augmented. Then we select s data that maintain the highest singular values of $\hat{\mathbf{X}}^n$ and form M_*^n. Finally, we update the memory to $\{M_*^i\}_{i\le n}$.

TABLE II
DATA SET SUMMARY. *Positive Rate* IS THE RATIO OF TABULAR DATA WITH POSITIVE CLASSES.

	Name	#Train data	#Test data	#Classes	Image size
Image Data	CIFAR-10	50,000	10,000	10	32*32
	CIFAR-100	50,000	10,000	100	32*32
	Tiny-ImageNet	50,000	10,000	100	64*64
	DomainNet-real	120,906	52,041	345	64*64
	Name	Size	#Input dim.	#Classes	Positive ratio
Tabular Data	Bank	45,211	16	2	11.70%
	Shoppers	12,330	17	2	15.47%
	Income	32,561	14	2	24.08%
	BlastChar	7,043	20	2	26.54%
	Shrutime	10,000	10	2	20.37%

IV. EXPERIMENTS

A. Experiment Setup

1) Data Sets: We adopt image data and tabular data to evaluate existing works and EDSR. The data set summary is in Tab. II and the details are as follows.

Image Data. Four benchmark data sets are selected for our main experiments, which are CIFAR-10 [72], CIFAR-100 [72], Tiny-ImageNet [73] and DomainNet-real [74]. Images of CIFAR-10 and CIFAR-100 have 32×32 pixels, while those of Tiny-ImageNet and DomainNet-real are unified to 64×64 pixels. CIFAR-10 has 10 classes, each class has 5,000 training samples and 1,000 testing samples. CIFAR-100 and Tiny-ImageNet have 100 classes, each class has 500 training samples and 100 testing samples. DomainNet-real is a subset of DomainNet with only real images, which has 345 classes and in total 120,906 training samples and 52,041 testing samples.

Tabular Data. Five tabular data sets are selected to show our method effectiveness on tabular data, whose statistics are shown in Tab. II and sources listed in [20]. These data sets are concerned about binary person-characteristic classification. We randomly split 20% of each data set as their test set.

2) Task Description: As shown in Fig. 1, the continual learning model will be trained and evaluated on a sequence of data sets. We divide the data sets in Tab. II into several subsets, where each subset contains multiple unique classes. CIFAR-10 is divided into 5 subsets of 2 unique classes. CIFAR-100 and Tiny-ImageNet are divided into 20 subsets of 5 unique classes. DomainNet-real is divided into 15 subsets of 23 unique classes. The five tabular data sets form a sequence of 5 increments. Under the continual learning setting, when learning the new data increment X^n, only X^n with the stored data $\{M_*^i\}_{i<n}$ are fed into the model, i.e., old data $\{X^i\}_{i<n}$ are unavailable. Then, the model $\hat{f}$ after learning X^n will be tested on all n data increments $\{X^i\}_{i=1}^n$.

3) Evaluation Metrics: After learning the sequence of n data sets $\{X^i\}_{i\leq n}$, this performance is recorded in an accuracy matrix $\mathbf{A}$ and a forgetting matrix $\mathbf{F}$, which are indicated in Fig. 3. Each element $A_{i,j} \in \mathbf{A}$ is the test accuracy on the old data X^j after learning a new data set X^i, where $j \leq i$. Following this definition, the elements $A_{i,j}$ whose $j > i$ are

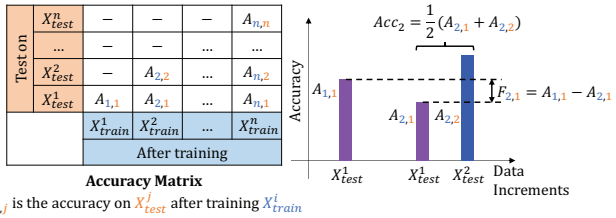


Fig. 3. An The accuracy matrix $\mathbf{A}$ and an illustration for the metrics.

meaningless and ignored. After learning X^i, Acc_i is averaged on the accuracy of learned data sets $\{X^j\}_{j\leq i}$:

$$Acc_i = \frac{1}{i}\sum_{j\leq i} A_{i,j}. \tag{17}$$

To intuitively reflect the forgetting issue of UCL models, we can calculate $F_{i,j} = \max_{i'\leq i}(A_{i',j} - A_{i,j})$ to evaluate the accuracy decrease on the old data set X^j after learning the new data set X^i. A smaller $F_{i,j}$ means that the knowledge from old X^j is less forgotten after learning the new X^i, thus is better. Naturally, $F_{i,i} = 0$ for any i since the model will not forget X_i after learning X^i. Similar to Acc_i, Fgt_i is averaged on the accuracy decrease of old data sets $\{X^j\}_{j<i}$:

$$Fgt_i = \frac{1}{i-1}\sum_{j<i} F_{i,j}, \tag{18}$$

where $F_{i,i}$ is excluded from average as it is always 0. In the following experiments, without specification, we simplify Acc_n as Acc and Fgt_n as Fgt, where n is the number of subsets of the chosen benchmark data set.

Inheriting the meaning from $\mathbf{A}$ and $\mathbf{F}$, the better performance means a higher Acc and a lower Fgt. When presenting the results, we use **Bold** to note the best results and use Underline to note the second best.

4) Baselines: We compare our method with the following state of the art baselines. Because the unsupervised setting, those SCL methods [10], [11], [22], [23] that strictly require labels are excluded from comparison. From the extendable SCL methods, we select the regularization-based method, SI [54], and memory-based method DER [60] .

- **Finetune.** Finetune means training the model continually without any forgetting prevention. At time step n, $f(\cdot)$ is trained on X^n with $\mathcal{L}_{css}(\cdot)$ only. It is the simplest method and serves as the vanilla baseline.
- **SI.** SI [54] is a SCL method that prevents forgetting by preserving the important old parameters. SI uses the gradient from the objective loss to identify the parameter importance, thus can be adapted to unsupervised setting.
- **DER.** DER [60] is a SCL method that randomly stores old data and replay them via distillation. Both characteristics enable DER to be applied in unsupervised setting. Especially, its distillation is based on the output from the CNN backbone model instead of representations.
- **LUMP.** LUMP [24] randomly stores old data and replay the stored data via mixing them with new data .

TABLE III
THE MODEL COMPARISON ON FOUR BENCHMARK IMAGE DATA SETS. FOR METHODS WITH MEMORY, THE LIMIT IS 256 FOR CIFAR-10, 640 FOR CIFAR-100 AND TINY-IMAGENET, AND 960 FOR DOMAINNET-REAL. MULTITASK IS EXCLUDED FROM COMPARISON SINCE IT ACCESSES OLD DATA.

Model	CIFAR-10		CIFAR-100		Tiny-ImageNet		DomainNet-real	
	$Acc \uparrow$	$Fgt \downarrow$	$Acc \uparrow$	$Fgt \downarrow$	$Acc \uparrow$	$Fgt \downarrow$	$Acc \uparrow$	$Fgt \downarrow$
Multitask	95.76 ± 0.08	-	86.31 ± 0.38	-	85.09 ± 0.01	-	75.37±0.07	-
Finetune	89.02 ± 0.05	5.79 ± 0.07	75.88 ± 2.18	5.23 ± 3.96	71.03 ± 1.31	10.01 ± 0.73	68.46 ± 0.16	7.10 ± 0.07
SI [54]	91.06 ± 0.08	3.79 ± 0.11	78.93 ± 1.15	8.37 ± 1.30	71.37 ± 0.82	9.99 ± 0.47	68.81 ± 0.06	6.57 ± 0.10
DER [60]	90.17 ± 0.62	5.15 ± 0.78	76.70 ± 0.45	9.21 ± 0.69	72.78 ± 0.59	8.58 ± 0.36	68.96 ± 0.23	6.79 ± 0.19
LUMP [24]	91.05 ± 0.37	2.11 ± 0.23	83.41 ± 0.14	4.12 ± 0.17	77.58 ± 0.24	4.24 ± 0.34	66.54 ± 0.06	6.11 ± 0.57
CaSSLe [33]	92.28 ± 0.13	0.62 ± 0.05	83.67 ± 0.35	1.33 ± 0.15	78.76 ± 0.25	2.48 ± 0.40	70.78 ± 0.23	0.55 ± 0.12
Our EDSR	**93.14 ± 0.08**	**0.12 ± 0.06**	**85.42 ± 0.20**	**0.57 ± 0.14**	**81.19 ± 0.22**	**1.77 ± 0.28**	**71.58 ± 0.27**	**0.24 ± 0.11**

- **CaSSLe.** CaSSLe [33] does not store old data and uses knowledge distillation to prevent forgetting.
- **Multitask.** Multitask model does not follow the continual setting, and all the data sets are learned at once, i.e. $f(\cdot)$ is optimized jointly on $\{X^1, ..., X^n\}$ by $\mathcal{L}_{css}(\cdot)$. Thus, it is regarded as the upper bound of continual learning models.

5) Implementation Details: The experiments are run on Nvidia RTX-2080 GPUs and Nvidia RTX-3090 GPUs. Our code is publicly available at Github.

For the data augmentation methods, we select $\{\text{crop}, \text{horizontalFlip}, \text{colorJitter}, \text{grayScale}, \text{grayScale}, \text{gaussianBlur}\}$ for image data and adopt an effective technique tabularCrop [75] for tabular data. To extract representations from the augmented data, the model $f(\cdot)$ is a concatenation of a ResNet-18 model [76] and a 2-layer MLP for image data, or a 7-layer MLP for tabular data. Especially for tabular data, the first layer of $f(\cdot)$ is data-specific and unifies various input table dimensions into the same hidden dimension. The image representations are of 2048 dimensions and the tabular ones are of 128 dimensions. The distillation projector $p_{dis}(\cdot)$ is a 2-layer MLP with the same dimension as the representation.

We select the $\mathcal{L}_{css}(\cdot)$ from SimSiam [31] for its effectiveness and low GPU memory requirement. The representation predictor $h(\cdot)$ of SimSiam a 2-layer MLP with the same dimensions as representations. The optimizer is selected as stochastic gradient descent for image data and Adam [77] for tabular data. Unless specified, we train all the models for 200 epochs on CIFAR-10 and CIFAR-100, 300 epochs on Tiny-ImageNet, 150 epochs on DomainNet-real, and 1000 epochs on tabular data. The results are averaged for 4 separate runs for image data and 10 runs for tabular data. To evaluate the quality of the representations without introducing extra parameters, the KNN Classifier [78] is selected, which is also a common evaluation method in CSSL [24], [31], [33].

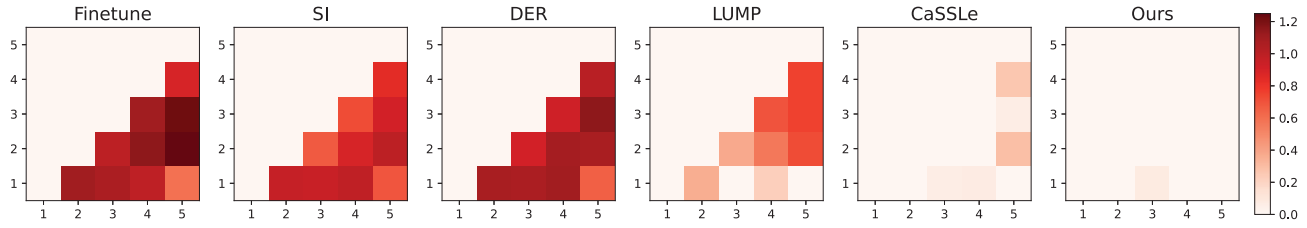
For our method, the only hyper-parameter is the number of neighbors for calculating the noise of $\mathcal{L}_{rpl}(\cdot)$, which is set to 100 in CIFAR-10, 10 in CIFAR-100, Tiny-ImageNet and DomainNet-real and 100 in five tabular data sets. We follow the hyper-parameter settings in LUMP [24] to set up the baseline methods SI, DER and LUMP.

B. Main Experiment

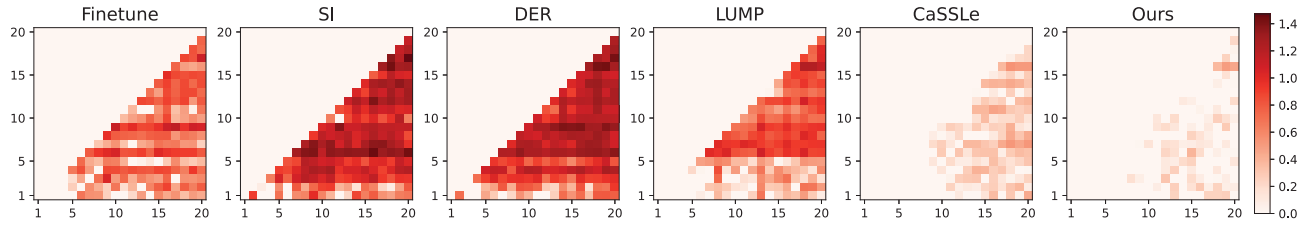
In this subsection, we compare the overall accuracy and forgetting ratio between baselines with the proposed EDSR on four benchmark image data sets. The results and discussions on the tabular data set are shown in Sec. IV-E. Comparing previous memory-based methods (DER and LUMP) with the memory-free methods (SI, CaSSLe), the latter have larger Acc and smaller Fgt. This ineffective usage of memory comes with two reasons. First is that DER and LUMP randomly select old data, but the high data quality is essential for unsupervised learning to effectively revoke the old knowledge. Secondly, their designs of data replay cannot exploit the advantage of $\mathcal{L}_{css}(\cdot)$. DER uses the backbone output instead of the representation, which neglects the rich information in representations. LUMP applies mixup, which brings ambiguity and restricts $\mathcal{L}_{css}(\cdot)$ from being fully optimized on either old or new data. This drawback is especially sever for the complex data set DomainNet-real, where LUMP is even worse than Finetune. We overcome two problems by selecting representative data and replaying via representation-level distillation. Thus, we outperform memory-free methods significantly and achieve the leading accuracy and low forgetting issue.

To provide more insights to the forgetting ratio in Tab. III, we visualize the forgetting matrix $\mathbf{F}$ of the models learned on four image data sets in Fig. 4. The color shade reflects the forgetting magnitude, and the smaller forgetting leads to lighter color. Intuitively the forgetting issue in the Finetune model is more severe than in continual learning models since it does not apply any technique to prevent forgetting. Interestingly, the Finetune model has smaller forgetting compared with two SCL methods SI and DER on CIFAR-100. That is because the Finetune model obtains smaller accuracies on new data sets, leaving small space for forgetting. Moreover, the UCL methods (LUMP, CaSSLe, and Ours) have much smaller forgetting scores than the SCL methods SI and DER, because UCL methods are designed to be more suitable to $\mathcal{L}_{css}(\cdot)$. Compared with LUMP that only uses stored data, CaSSLe explores the past model to include more complete old knowledge, making CaSSLe forget less. Corresponding to the lowest forgetting ratio of Tab. III, our method leverages both stored data and past model, which achieves the lightest color in Fig. 4. Notice that the length of data set sequence is the max number of red boxes for a row in Fig. 4, which varies between 5 and 20. And regardless of short or long data set sequences, our method effectively remains the lowest forgetting.

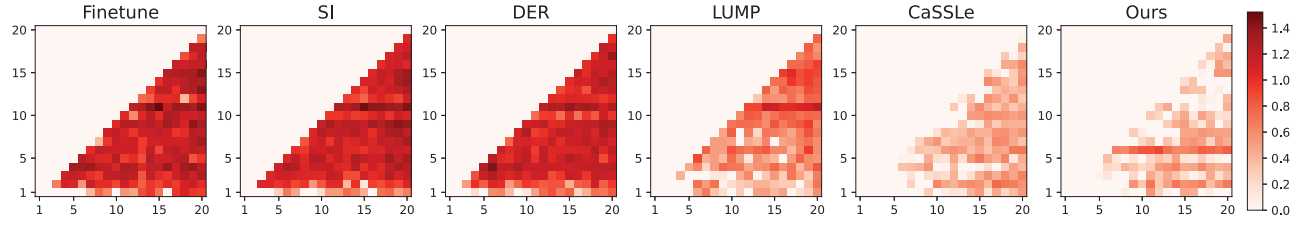
To evaluate the model plasticity and provide insights to accuracy in Tab. III, we plot the new data set accuracy (i.e., $A_{i,i}$) at i-th increment in Fig. 5. We omit the comparison



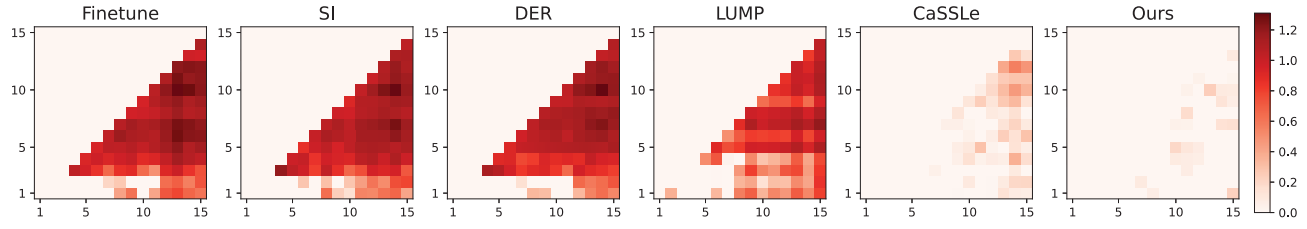
(a) Forgetting in CIFAR-10



(b) Forgetting in CIFAR-100



(c) Forgetting in Tiny-ImageNet



(d) Forgetting in DomainNet-real

Fig. 4. The forgetting matrix $\mathbf{F}$ of the models learned on four image data sets. $F_{i,j}$ evaluates the accuracy decrease on the old data set X^j after learning the new data set X^i. The reported values are the logarithms of forgetting and indicated by color. The lighter the color, the better the model (smaller forgetting).

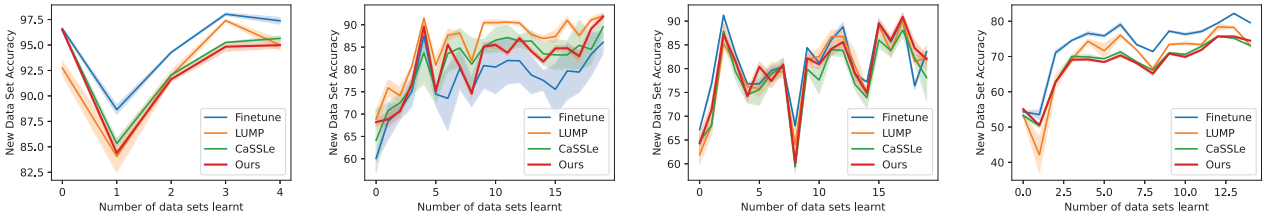
TABLE IV
EXPERIMENTS ON DIFFERENT REPLAYING METHODS. THE DATA ARE SELECTED BASED ON HIGH-ENTROPY DATA. THE REPORTED VALUE SS AVERAGE ACCURACY Acc.

Dataset	**No Replay** (CaSSLe)	**Loss for Data Replay**		
		$\mathcal{L}_{css}(\cdot)$	$\mathcal{L}_{dis}(\cdot)$	$\mathcal{L}_{rpl}(\cdot)$
CIFAR-10	92.28±0.13	91.38±0.13	**93.17±0.26**	93.14±0.08
CIFAR-100	83.67±0.35	73.63±3.33	85.23±0.31	**85.42±0.20**
TIny-ImageNet	78.76±0.25	62.15±0.89	80.27±0.56	**81.19±0.22**

between SCL methods as their Accs are significantly lower than UCL ones. A higher accuracy on the new data set reflects a better plasticity of the method. The new accuracy is fluctuating during data set increments, because the learning difficulty of data sets varies from one to the other. It is interesting that the new accuracy of ours is not advanced. Because our method has the strongest forgetting prevention mechanism, which trades the model plasticity for stability to achieve the best Acc and Fgt on all the learned data sets. This is also reflected from CaSSLe, which has the second best Acc and Fgt but generally has the second lowest new data set accuracy. Furthermore, the standard deviations of Finetune and CaSSLe are higher than those of the replay methods (LUMP and Ours) as depicted in Fig. 5b. This is because CIFAR-100 has smaller (500 images per data set) and simpler (32×32 pixels) new data sets, which makes the model prone to overfitting and sensitive to initial states. The replay methods overcome this issue by providing the old data to diversify the new data sets.

C. Ablation Studies

Here we report the performance of several variants of EDSR to investigate some key components in this paper, including how injecting noise improves the replay performance, how to select data, and how different designs of $\mathcal{L}_{css}(\cdot)$ affect the model effectiveness. Note that the original setting is based on selecting high-entropy data and replaying through noise-enhanced distillation $\mathcal{L}_{rpl}(\cdot)$.



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny-ImageNet (d) DomainNet-real

Fig. 5. Experiments on the new data set accuracy.

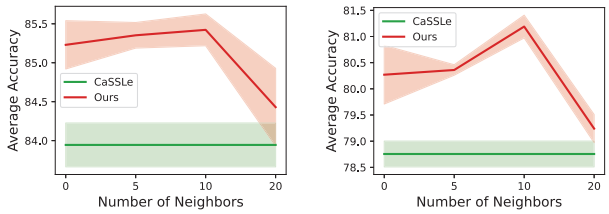
1) How to Replay Data: To show the improvement brought by replaying the old data with noisy distillation, we present the experiments in Tab. IV, where the comparison among using $\mathcal{L}_{css}(\cdot)$, $\mathcal{L}_{dis}(\cdot)$, and $\mathcal{L}_{rpl}(\cdot)$ to replay the stored data is presented. The data selection method is by high-entropy data. It is interesting that replaying with $\mathcal{L}_{css}(\cdot)$ is worse than no data replay. This validates the issue of over-fitting on the stored data under the UCL setting, which is caused by the inability of representing the whole old data by a few stored, unlabeled data. Fortunately, the issue is solved by the distillation mechanism that compensates the old knowledge with the old model, as shown in the results of replaying with $\mathcal{L}_{dis}(\cdot)$ and $\mathcal{L}_{rpl}(\cdot)$. Furthermore, $\mathcal{L}_{rpl}(\cdot)$ adds noises to enlarge the learnable old knowledge, making it outperform $\mathcal{L}_{dis}(\cdot)$ in data replay. Especially, the advantage of $\mathcal{L}_{rpl}(\cdot)$ increases with more complex data sets.

2) How to select data: In Tab. V, the effectiveness of replaying different data selection methods are evaluated. Besides entropy based data selection, other data selection methods are:

- **Random**: randomly select data within the memory size.
- **Distant**: data with maximum distance among each other are selected based on K-means++ seeding algorithm [79].
- **K-means**: store the cluster centers of K-means [80], where the cluster number is proportional to the memory size.
- **Min-Var**: [61] forms the same amount of clusters as the number of classes, and store the data whose augmented views have minimum representation variance.

Note that the memory limit is 320 for CIFAR-10 and 640 for CIFAR-100 and Tiny-ImageNet. As shown in Tab. IV, $\mathcal{L}_{css}(\cdot)$ is less effective in data replay. Therefore, we only show the variants of different selection models under compilation of $\mathcal{L}_{dis}(\cdot)$ and $\mathcal{L}_{rpl}(\cdot)$ in Tab. V. The results under $\mathcal{L}_{dis}(\cdot)$ are related to no additional tricks and directly reflect the effectiveness of different selection methods. Furthermore, the results under $\mathcal{L}_{rpl}(\cdot)$ show how our technique is applicable to different selection methods and improve their performances.

We first compare the data selection methods with the results under $\mathcal{L}_{dis}(\cdot)$. Compared with no old data replay (CaSSLe), any selection methods can bring notable improvements. Interestingly, compared with vanilla data selection methods (Random, Distant), clustering methods (K-means, Min-Var) do not always have advantages. This is because data from clusters are prone to dense areas in the representation space, which can be locally important but require a careful choice of cluster number to maintain high entropy. As a result, the clustering methods do not generalizes well on different data sets. On the other hand, directly selecting high entropy data is more effective and generalizes better.



(a) CIFAR-100 (b) Tiny-ImageNet

Fig. 6. Experiments on the effect of different number of neighbors for calculating noise in $\mathcal{L}_{rpl}(\cdot)$. We plot unchanged CaSSLe just for comparison.

Afterwards, we evaluate the advantages of $\mathcal{L}_{rpl}(\cdot)$ over $\mathcal{L}_{dis}(\cdot)$. Across different data selection methods, replaying with $\mathcal{L}_{rpl}(\cdot)$ generally achieves higher Acc and lower Fgt. This reflects that when distilling the old data with noise, old knowledge is better recovered and over-fitting issue is less likely to happen. The improvement from noise in CIFAR-10 is small, because storing 256 data sample is large enough to represent the old data sets, and adding noise can only stabilize the performance.

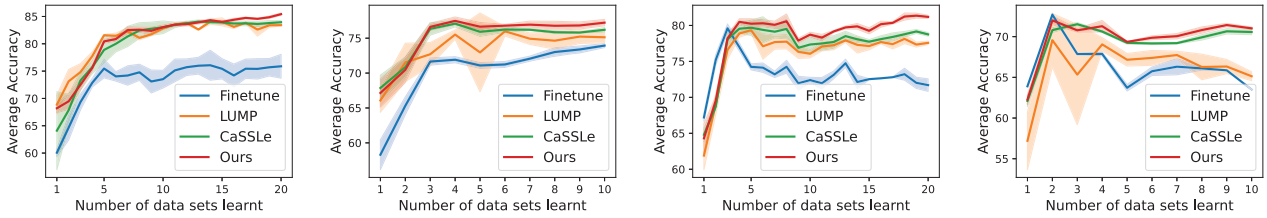
3) Different CSSL losses: In the main experiments, we base our method on $\mathcal{L}_{css}(\cdot)$ from SimSiam. Here we switch to $\mathcal{L}_{css}(\cdot)$ of BarlowTwins to evaluate the effect from different $\mathcal{L}_{css}(\cdot)$, the results are shown in Tab. VI. After the substitution of $\mathcal{L}_{css}(\cdot)$, CaSSLe and Ours have decreased performances, especially CaSSLe is significantly inferior to LUMP. This is because the input representations should differ only from their encoder models to ensure the effectiveness of distillation. As shown in Eq. (3), $\mathcal{L}_{css}(\cdot)$ of SimSiam can satisfy this requirement, based on which CaSSLe and Ours give good performances. However, from Eq. (4), $\mathcal{L}_{css}(\cdot)$ of BarlowTwins considers the correlation matrix between two batches of representations, which unavoidably compares inputs from different data and different models simultaneously. This brings chaos to the distillation process, confusing the knowledge from the past model. LUMP escapes from this chaos as it relies solely on the old data. On the other hand, despite the reduced effectiveness of distillation, our effective usage of old data still makes our performance better than CaSSLe.

D. Sensitivity Analysis

To evaluate the sensitivity of EDSR in different learning settings, we present several experiments that are related to the hyper-parameter, the subset setting of data sets, and the amount of stored data.

TABLE V
EXPERIMENTS ON DIFFERENT STORAGE METHODS AND WHETHER ADD NOISES. MEMORY SIZES ARE THE SAME FOR ALL THE METHODS, 256 FOR CIFAR-10 AND 640 FOR CIFAR-100 AND TINY-IMAGENET.

DataSet		No Replay (CaSSLe)	Random	K-means [80]	Min-Var [61]	Distant [79]	High Entropy
			Replay with $\mathcal{L}_{dis}(\cdot)$				
CIFAR-10	$Acc \uparrow$	92.28 ± 0.13	93.04 ± 0.11	93.08 ± 0.19	92.96 ± 0.15	93.03 ± 0.03	**93.17 ± 0.26**
	$Fgt \downarrow$	0.62 ± 0.05	**0.07 ± 0.05**	0.20 ± 0.09	0.19 ± 0.07	0.14 ± 0.05	0.08 ± 0.01
CIFAR-100	$Acc \uparrow$	83.67 ± 0.35	84.94 ± 0.68	84.71 ± 0.52	85.14 ± 0.23	84.87 ± 0.41	**85.23 ± 0.31**
	$Fgt \downarrow$	1.33 ± 0.15	0.89 ± 0.13	0.75 ± 0.20	**0.48 ± 0.10**	0.81 ± 0.09	0.73 ± 0.03
Tiny-ImageNet	$Acc \uparrow$	78.76 ± 0.25	79.50 ± 0.35	**80.36 ± 0.16**	79.83 ± 0.35	79.55 ± 0.17	80.27 ± 0.56
	$Fgt \downarrow$	2.48 ± 0.40	2.02 ± 0.20	1.45 ± 0.19	1.41 ± 0.11	1.77 ± 0.33	**1.31 ± 0.41**
			Replay with $\mathcal{L}_{rpl}(\cdot)$				
CIFAR-10	$Acc \uparrow$	92.28 ± 0.13	92.96 ± 0.08	92.90 ± 0.13	93.07 ± 0.19	93.05 ± 0.24	**93.14 ± 0.08**
	$Fgt \downarrow$	0.62 ± 0.05	0.06 ± 0.02	0.18 ± 0.02	0.20 ± 0.09	0.16 ± 0.09	**0.12 ± 0.06**
CIFAR-100	$Acc \uparrow$	83.67 ± 0.35	85.04 ± 0.39	85.22 ± 0.27	85.26 ± 0.38	84.95 ± 0.45	**85.42 ± 0.20**
	$Fgt \downarrow$	1.33 ± 0.15	0.70 ± 0.11	0.72 ± 0.19	0.71 ± 0.02	0.91 ± 0.11	**0.57 ± 0.14**
Tiny-ImageNet	$Acc \uparrow$	78.76 ± 0.25	79.81 ± 0.22	80.66 ± 0.35	79.89 ± 0.10	79.99 ± 0.59	**81.19 ± 0.22**
	$Fgt \downarrow$	2.48 ± 0.40	1.54 ± 0.19	1.79 ± 0.35	**1.51 ± 0.09**	1.70 ± 0.41	1.77 ± 0.28



(a) CIFAR-100: 20 subsets (b) CIFAR-100: 10 subsets (c) Tiny-ImageNet: 20 subsets (d) Tiny-ImageNet: 10 subsets

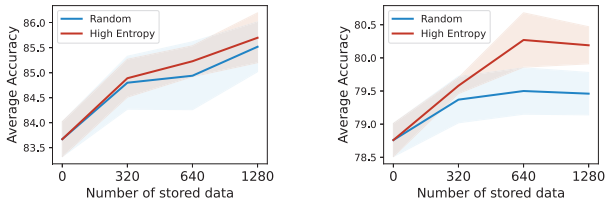
Fig. 7. Experiments on learning CIFAR-100 and Tiny-ImageNet in two settings, 20 subsets of 5 classes and 10 subsets of 10 classes.

TABLE VI
EXPERIMENTS OF DIFFERENT CSSL LOSSES. AVERAGE ACCURACY Acc IS REPORTED.

Methods	SimSiam		BarlowTwins	
	CIFAR-100	Tiny-ImageNet	CIFAR-100	Tiny-ImageNet
Multitask	86.31 ± 0.38	85.09 ± 0.01	87.16 ± 0.52	83.01 ± 0.10
Finetune	75.51 ± 0.64	57.13 ± 10.31	71.97 ± 0.54	68.81 ± 0.29
LUMP	83.41 ± 0.14	77.58 ± 0.24	**83.14 ± 0.87**	75.02 ± 0.36
CaSSLe	83.67 ± 0.35	78.76 ± 0.25	79.60 ± 0.80	70.30 ± 1.44
Ours	**85.42 ± 0.20**	**81.19 ± 0.22**	80.66 ± 1.67	**75.59 ± 1.11**

1) Hyper-parameter Study: Here we discuss how the selection of hyper-parameter, i.e. the number of neighbors for calculating noise in $\mathcal{L}_{rpl}(\cdot)$, affects the model performance, whose results are shown in Fig. 6. The settings other than the neighbor number are the same as in the main experiment. Note that $\mathcal{L}_{rpl}(\cdot)$ with 0 neighbors is equivalent to $\mathcal{L}_{dis}(\cdot)$. Across the three data sets, with the increase of neighbor number, the effectiveness of $\mathcal{L}_{rpl}(\cdot)$ increases then decreases. Because when the neighbor number is within a suitable range, the neighbors share similar features as the anchor sample, and adding the noise includes these useful knowledge into distillation. After the neighbor number becomes too large, the remote neighbors differ largely from the anchor, which makes adding the noise misleading and negatively affects the model performance. Besides Acc magnitude, within a proper range of neighbor number, adding noises brings down the standard deviation as more old knowledge is stably learned.

2) Different Data Settings: To investigate the influence of changing the data set setting, we compare the model performance on learning CIFAR-100 and Tiny-ImageNet in two settings in Fig. 7. In addition to the original setting, the new one divides them into 10 subsets of 10 classes. Considering the complexity of data sets, CIFAR-100 is trained for 300 epochs in the new setting. 32 samples are stored for each data subset, thus 640 for the original split and 320 for the new split. CIFAR-10 is omitted because 10 classes cannot be further divided. Interestingly, at the beginning of increments, Acc_i of all methods is increasing instead of decreasing, even for Finetune which faces the most severe forgetting. Because the sizes of the first several data sets are too small for $\mathcal{L}_{css}(\cdot)$ to generate effective representations, making them inadequately learned. After later increments, the larger amount of data improves the effectiveness of the model and increases the Acc's of early data sets. Across different methods and data settings, ours stably outperforms the others.



(a) CIFAR-100 (b) Tiny-ImageNet

Fig. 8. Experiments on the effect of different amounts of stored data.

3) Different Amount of Stored Data: To investigate how the number of stored data affects the model performance, we present the results on CIFAR-100 and Tiny-ImageNet in Fig. 8. We omit the noise in $\mathcal{L}_{rpl}(\cdot)$ to remove the disturbance from its improvements on replay. We also include random data selection to further show the effectiveness of high-entropy data selection. From Fig. 8, it is intuitive that storing more data leads to better performances. The Acc drop in Fig. 8b is because more epochs are needed to learn the larger and more complex stored data. Interestingly, with the increasing

TABLE VII
RESULTS OF LEARNING TABULAR DATA SETS.

Method	Multitask	Finetune	CaSSLe	Ours
Acc	80.38±0.31	80.82±0.49	81.09±0.73	**81.27±0.47**
Frg	-	0.79±0.41	0.69±0.37	**0.52±0.26**

memory size, the performance difference between random data and high entropy data increases then decreases. This is affected by the old knowledge contained in the stored data. When the memory size is small, the stored data can contain limited old knowledge, thus the difference between selection methods is small. Then with the increase of memory size, the high entropy selection brings more representative data into the memory, making the performance difference larger. Finally, when the memory size is sufficiently large, random selection is also likely to find representative data, thus the performance difference becomes smaller. However, because continual learning lays a small limit on the memory size, guaranteeing high entropy is the most suitable selection approach. This advantage is also reflected by the smaller standard deviation and smaller average forgetting.

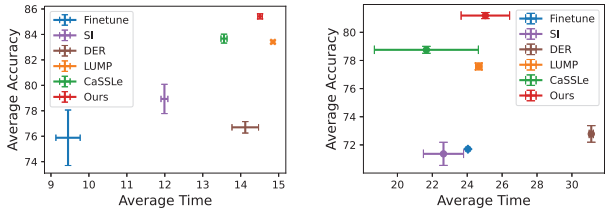
E. Generalization to Tabular Data

To evaluate the effectiveness of EDSR on other data type, here we present the experiments on tabular data that are listed in Tab. VII. LUMP is omitted from comparison because its mix-up technique cannot be extended to non-unified input dimensions of tabular data sets. We store 1% of each tabular data set into the memory. It is interesting that the results of Multitask is worse than the continual methods, especially Finetune. This is because the sizes of data sets are unbalanced, and those smaller data sets are worse learned when combined with the larger ones. The continual methods avoids this issue by learning different data sets individually. And EDSR achieves the best performance, which validates our effectiveness on other data type than image.

F. Efficiency-Effectiveness Study

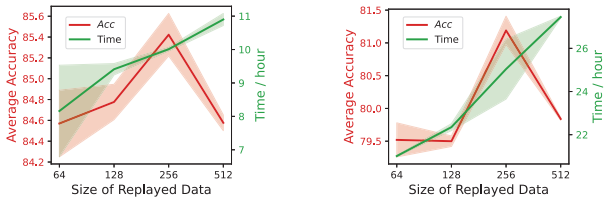
As our EDSR replays old data to improve the performance, we show the trade-off between time efficiency and effectiveness in Fig. 9. Time costs CIFAR-100 are collected from RTX 2080 GPUs and those of Tiny-ImageNet are collected from RTX 3090 GPUs. Compared with SCL methods (SI and DER), UCL methods (LUMP, CaSSLe and Ours) spend longer time to achieve better performances. Within the UCL methods, LUMP and Ours are less efficient due to the utilization of old data. However, compared with the additional time spent, our effectiveness improvement is more significant.

Additionally, we study the efficiency-effectiveness trade-off within our method design, and results are shown in Fig. 10. The number of replayed old data in each batch is the major factor affecting our efficiency, while data selection takes much smaller time than training. Thus, we present the study on different sizes of replayed data in Fig. 10, and the memory budget is 640. With the increase of replayed data size, the time cost keeps increasing, while the effectiveness increases then decreases. The decreased effectiveness is because replaying too many stored data prohibits learning new knowledge. It shows that 256 may be a proper size to achieve maximum



(a) CIFAR-100 (b) CIFAR-100

Fig. 9. Visualization of efficiency-effectiveness analysis (time is in hours)



(a) CIFAR-100 (b) Tiny-ImageNet

Fig. 10. Time vs *Acc* for different replay size.

effectiveness without sacrificing efficiency too much. To better strike the balance, it may be a potential way to sample the stored data from the memory based on their similarities to the new data during replay. In such a way, more valuable stored data can be replayed.

V. CONCLUSION

In this paper, we propose an effective unsupervised continual learning framework, namely EDSR, to help models learn new knowledge without forgetting old knowledge. Specifically, EDSR consists of data selection and data replay. To select and store representative data subset without label assistance, we analyze that such subset should have the largest entropy in representation space. To replay the stored data while balancing the model stability for old data and the plasticity for new data, we propose to distill the noise-enhanced knowledge from the old model. The performance of EDSR is demonstrated by extensive experiments on real-world datasets. For the future works, it is worth trying to study the continual learning problem on graphs, such as graph neural networks [81], [82], unsupervised graph learning [83], and knowledge graph embedding [84], [85].

ACKNOWLEDGMENT

The corresponding authors are Shuangyin Li and Shimin Di. Lei Chen's work is partially supported by National Science Foundation of China (NSFC) under Grant No. U22B2060, the Hong Kong RGC GRF Project 16213620, RIF Project R6020-19, AOE Project AoE/E-603/18, Theme-based project TRS T41-603/20R, CRF Project C2004-21G, China NSFC No. 61729201, Guangdong Basic and Applied Basic Research Foundation 2019B151530001, Hong Kong ITC ITF grants MHX/078/21 and PRP/004/22FX, Microsoft Research Asia Collaborative Research Grant and HKUST-Webank joint research lab grants. Xiaofang Zhou's work is supported by the JC STEM Lab of Data Science Foundations funded by The Hong Kong Jockey Club Charities Trust, HKUST-China Unicom Joint Lab on Smart Society, and HKUST-HKPC Joint Lab on Industrial AI and Robotics Research. The authors appreciate the support of the deep learning computing framework MindSpore (https://www.mindspore.cn/en/).

REFERENCES

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," Neural Networks, vol. 113, pp. 54–71, 2019. [Online]. Available: https://doi.org/10.1016/j.neunet.2019.01.012

[2] M. D. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," TPAMI, vol. 44, no. 7, pp. 3366–3385, 2022. [Online]. Available: https://doi.org/10.1109/TPAMI.2021.3057446

[3] R. M. French and N. Chater, "Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting," Neural Comput., vol. 14, no. 7, pp. 1755–1769, 2002. [Online]. Available: https://doi.org/10.1162/08997660260028700

[4] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, pp. 7120–7129. [Online]. Available: https://doi.org/10.1109/CVPR.2017.753

[5] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 3, pp. 651–663, 2020. [Online]. Available: https://doi.org/10.1109/TPAMI.2018.2884462

[6] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," CoRR, vol. abs/1701.08734, 2017. [Online]. Available: http://arxiv.org/abs/1701.08734

[7] J. Serrà, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 4555–4564. [Online]. Available: http://proceedings.mlr.press/v80/serra18a.html

[8] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 7765–7773. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Mallya_PackNet_Adding_Multiple_CVPR_2018_paper.html

[9] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach, "Adversarial continual learning," in Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI, ser. Lecture Notes in Computer Science, vol. 12356. Springer, 2020, pp. 386–402. [Online]. Available: https://doi.org/10.1007/978-3-030-58621-8_23

[10] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, pp. 5533–5542. [Online]. Available: https://doi.org/10.1109/CVPR.2017.587

[11] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 6467–6476. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html

[12] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in NeurIPS, Long Beach, CA, USA, 2017, pp. 2990–2999. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/0efbe98067c6c73dba1250d2beaa81f9-Abstract.html

[13] N. Kamra, U. Gupta, and Y. Liu, "Deep generative dual memory network for continual learning," CoRR, vol. abs/1710.10368, 2017. [Online]. Available: http://arxiv.org/abs/1710.10368

[14] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=Hkf2_sC5FX

[15] Y. Liu, Y. Su, A. Liu, B. Schiele, and Q. Sun, "Mnemonics training: Multi-class incremental learning without forgetting," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. Computer Vision Foundation / IEEE, 2020, pp. 12 242–12 251. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Liu_Mnemonics_Training_Multi-Class_Incremental_Learning_Without_Forgetting_CVPR_2020_paper.html

[16] Z. Li and D. Hoiem, "Learning without forgetting," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 12, pp. 2935–2947, 2018. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2773081

[17] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," CoRR, vol. abs/1612.00796, 2016.

[18] X. Hu, K. Tang, C. Miao, X. Hua, and H. Zhang, "Distilling causal effect of data in class-incremental learning," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 2021, pp. 3957–3966. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Hu_Distilling_Causal_Effect_of_Data_in_Class-Incremental_Learning_CVPR_2021_paper.html

[19] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12365. Springer, 2020, pp. 86–102. [Online]. Available: https://doi.org/10.1007/978-3-030-58565-5_6

[20] H. Liu, S. Di, and L. Chen, "Incremental tabular learning on heterogeneous feature space," Proc. ACM Manag. Data, vol. 1, no. 1, may 2023. [Online]. Available: https://doi.org/10.1145/3588698

[21] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, 2019, pp. 831–839. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Hou_Learning_a_Unified_Classifier_Incrementally_via_Rebalancing_CVPR_2019_paper.html

[22] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 11 816–11 825. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/e562cd9c0768d5464b64cf61da7fc6bb-Abstract.html

[23] F. Zhu, X. Zhang, C. Wang, F. Yin, and C. Liu, "Prototype augmentation and self-supervision for incremental learning," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 2021, pp. 5871–5880. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Zhu_Prototype_Augmentation_and_Self-Supervision_for_Incremental_Learning_CVPR_2021_paper.html

[24] D. Madaan, J. Yoon, Y. Li, Y. Liu, and S. J. Hwang, "Representational continuity for unsupervised continual learning," in International Conference on Learning Representations, 2022. [Online]. Available: https://openreview.net/forum?id=9Hrka5PA7LW

[25] A. Achille, T. Eccles, L. Matthey, C. P. Burgess, N. Watters, A. Lerchner, and I. Higgins, "Life-long disentangled representation learning with cross-domain latent homologies," in NeurIPS, 2018, pp. 9895–9905. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/a0afdf1ac166b8652ffe9dee6eac779e-Abstract.html

[26] D. Rao, F. Visin, A. A. Rusu, R. Pascanu, Y. W. Teh, and R. Hadsell, "Continual unsupervised representation learning," in NeurIPS, 2019, pp. 7645–7655. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/861578d797aeb0634f77aff3f488cca2-Abstract.html

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[28] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 1597–1607. [Online]. Available: http://proceedings.mlr.press/v119/chen20j.html

[29] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. Computer Vision Foundation / IEEE, 2020, pp. 9726–9735. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.00975

[30] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/70feb62b69f16e0238f741fab228fec2-Abstract.html

[31] X. Chen and K. He, "Exploring simple siamese representation learning," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 2021, pp. 15 750–15 758. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Chen_Exploring_Simple_Siamese_Representation_Learning_CVPR_2021_paper.html

[32] J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - A new approach to self-supervised learning," in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html

[33] E. Fini, V. G. T. da Costa, X. Alameda-Pineda, E. Ricci, K. Alahari, and J. Mairal, "Self-supervised models are continual learners," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. IEEE, 2022, pp. 9611–9620. [Online]. Available: https://doi.org/10.1109/CVPR52688.2022.00940

[34] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[35] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," CoRR, vol. abs/1503.02531, 2015.

[36] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," CoRR, vol. abs/1406.2661, 2014. [Online]. Available: http://arxiv.org/abs/1406.2661

[37] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," 2020. [Online]. Available: https://arxiv.org/abs/2011.00362

[38] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 12 310–12 320. [Online]. Available: http://proceedings.mlr.press/v139/zbontar21a.html

[39] H. Li, S. Di, Z. Li, L. Chen, and J. Cao, "Black-box adversarial attack and defense on graph neural networks," in 38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022. IEEE, 2022, pp. 1017–1030. [Online]. Available: https://doi.org/10.1109/ICDE53745.2022.00081

[40] C. J. Reed, S. Metzger, A. Srinivas, T. Darrell, and K. Keutzer, "Selfaugment: Automatic augmentation policies for self-supervised learning," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 2021, pp. 2674–2683. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Reed_SelfAugment_Automatic_Augmentation_Policies_for_Self-Supervised_Learning_CVPR_2021_paper.html

[41] S. Di and L. Chen, "Message function search for knowledge graph embedding," in Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023, Y. Ding, J. Tang, J. F. Sequeda, L. Aroyo, C. Castillo, and G. Houben, Eds. ACM, 2023, pp. 2633–2644. [Online]. Available: https://doi.org/10.1145/3543507.3583546

[42] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical data augmentation with no separate search," CoRR, vol. abs/1909.13719, 2019. [Online]. Available: http://arxiv.org/abs/1909.13719

[43] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 6662–6672. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/6add07cf50424b14fdf649da87843d01-Abstract.html

[44] C. Hou, J. Zhang, and T. Zhou, "When to learn what: Model-adaptive data augmentation curriculum," CoRR, vol. abs/2309.04747, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2309.04747

[45] H. Li, X. Lin, and L. Chen, "Fine-grained entity typing via label noise reduction and data augmentation," in Database Systems for Advanced Applications - 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11-14, 2021, Proceedings, Part I, ser. Lecture Notes in Computer Science, C. S. Jensen, E. Lim, D. Yang, W. Lee, V. S. Tseng, V. Kalogeraki, J. Huang, and C. Shen, Eds., vol. 12681. Springer, 2021, pp. 356–374. [Online]. Available: https://doi.org/10.1007/978-3-030-73194-6_24

[46] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," Pattern Recognit., vol. 77, pp. 354–377, 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.10.013

[47] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," Atmospheric environment, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[48] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[49] S. Di, J. Peng, Y. Shen, and L. Chen, "Transfer learning via feature isomorphism discovery," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018, Y. Guo and F. Farooq, Eds. ACM, 2018, pp. 1301–1309. [Online]. Available: https://doi.org/10.1145/3219819.3220029

[50] S. Di, Y. Shen, and L. Chen, "Relation extraction via domain-aware transfer learning," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds. ACM, 2019, pp. 1348–1357. [Online]. Available: https://doi.org/10.1145/3292500.3330890

[51] H. Li and L. Chen, "Cache-based GNN system for dynamic graphs," in CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, and H. Tong, Eds. ACM, 2021, pp. 937–946. [Online]. Available: https://doi.org/10.1145/3459637.3482237

[52] ——, "Early: Efficient and reliable graph neural network for dynamic graphs," Proc. ACM Manag. Data, vol. 1, no. 2, jun 2023. [Online]. Available: https://doi.org/10.1145/3589308

[53] X. Liu, M. Masana, L. Herranz, J. van de Weijer, A. M. López, and A. D. Bagdanov, "Rotate your networks: Better weight consolidation and less catastrophic forgetting," in 24th International Conference on Pattern Recognition, ICPR 2018, Beijing, China, August 20-24, 2018. IEEE Computer Society, 2018, pp. 2262–2268. [Online]. Available: https://doi.org/10.1109/ICPR.2018.8545895

[54] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 3987–3995. [Online]. Available: http://proceedings.mlr.press/v70/zenke17a.html

[55] P. Dhar, R. V. Singh, K. Peng, Z. Wu, and R. Chellappa, "Learning without memorizing," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, 2019, pp. 5138–5146. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Dhar_Learning_Without_Memorizing_CVPR_2019_paper.html

[56] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," CoRR, vol. abs/1606.04671, 2016. [Online]. Available: http://arxiv.org/abs/1606.04671

[57] S. Mirzadeh, M. Farajtabar, D. Görür, R. Pascanu, and H. Ghasemzadeh, "Linear mode connectivity in multitask and continual learning," in 9th International Conference on Learning Representations, ICLR 2021,

Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=Fmg_fQYUejf

[58] Y. Liu, S. Parisot, G. G. Slabaugh, X. Jia, A. Leonardis, and T. Tuytelaars, "More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning," in Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVI, ser. Lecture Notes in Computer Science, vol. 12371. Springer, 2020, pp. 699–716. [Online]. Available: https://doi.org/10.1007/978-3-030-58574-7_42

[59] A. Iscen, J. Zhang, S. Lazebnik, and C. Schmid, "Memory-efficient incremental learning through feature adaptation," in Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVI, ser. Lecture Notes in Computer Science, vol. 12361. Springer, 2020, pp. 699–715. [Online]. Available: https://doi.org/10.1007/978-3-030-58517-4_41

[60] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html

[61] Z. Lin, Y. Wang, and H. Lin, "Continual contrastive learning for image classification," in 2022 IEEE International Conference on Multimedia and Expo (ICME). IEEE, jul 2022. [Online]. Available: https://doi.org/10.1109%2Ficme52920.2022.9859995

[62] A. Gomez-Villa, B. Twardowski, L. Yu, A. D. Bagdanov, and J. van de Weijer, "Continually learning self-supervised representations with projected functional regularization," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 3866–3876.

[63] S. Purushwalkam, P. Morgado, and A. Gupta, "The challenges of continuous self-supervised learning," in Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI, ser. Lecture Notes in Computer Science, vol. 13686. Springer, 2022, pp. 702–721. [Online]. Available: https://doi.org/10.1007/978-3-031-19809-0_40

[64] J. S. Smith, C. E. Taylor, S. Baer, and C. Dovrolis, "Unsupervised progressive learning and the STAM architecture," in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021. ijcai.org, 2021, pp. 2979–2987. [Online]. Available: https://doi.org/10.24963/ijcai.2021/410

[65] E. T. Jaynes, "Information theory and statistical mechanics," Physical review, vol. 106, no. 4, p. 620, 1957.

[66] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 9, pp. 1546–1562, 2007. [Online]. Available: https://doi.org/10.1109/TPAMI.2007.1085

[67] X. Liu, Z. Wang, Y. Li, and S. Wang, "Self-supervised learning via maximum entropy coding," in NeurIPS, 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/dc709714c52b35f2f34aca2a92b06bc8-Abstract-Conference.html

[68] R. A. Horn and C. R. Johnson, Matrix Analysis, 2nd ed. Cambridge University Press, 2012.

[69] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," The London, Edinburgh, and Dublin philosophical magazine and journal of science, vol. 2, no. 11, pp. 559–572, 1901.

[70] H. Cha, J. Lee, and J. Shin, "Co^{2}l: Contrastive continual learning," in 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. IEEE, 2021, pp. 9496–9505. [Online]. Available: https://doi.org/10.1109/ICCV48922.2021.00938

[71] Y. Wang, Q. Zhang, Y. Wang, J. Yang, and Z. Lin, "Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap," in The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=ECvgmYVyeUz

[72] A. Krizhevsky, "Learning multiple layers of features from tiny images," pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[73] M. A. mnmoustafa, "Tiny imagenet," 2017. [Online]. Available: https://kaggle.com/competitions/tiny-imagenet

[74] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. IEEE, 2019, pp. 1406–1415. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00149

[75] D. Bahri, H. Jiang, Y. Tay, and D. Metzler, "Scarf: Self-supervised contrastive learning using random feature corruption," in International Conference on Learning Representations, 2022. [Online]. Available: https://openreview.net/forum?id=CuV_qYkmKb3

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[78] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 3733–3742. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Wu_Unsupervised_Feature_Learning_CVPR_2018_paper.html

[79] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007, N. Bansal, K. Pruhs, and C. Stein, Eds. SIAM, 2007, pp. 1027–1035. [Online]. Available: http://dl.acm.org/citation.cfm?id=1283383.1283494

[80] J. MacQueen, "Classification and analysis of multivariate observations," in 5th Berkeley Symp. Math. Statist. Probability. University of California Los Angeles LA USA, 1967, pp. 281–297.

[81] Z. Wang, S. Di, and L. Chen, "Autogel: An automated graph neural network with explicit link information," in Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 24 509–24 522. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/cd3afef9b8b89558cd56638c3631868a-Abstract.html

[82] ——, "A message passing neural network space for better capturing data-dependent receptive fields," in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023, A. K. Singh, Y. Sun, L. Akoglu, D. Gunopulos, X. Yan, R. Kumar, F. Ozcan, and J. Ye, Eds. ACM, 2023, pp. 2489–2501. [Online]. Available: https://doi.org/10.1145/3580305.3599243

[83] Z. Wang, S. Di, L. Chen, and X. Zhou, "Search to fine-tune pre-trained graph neural networks for graph-level tasks," CoRR, vol. abs/2308.06960, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2308.06960

[84] S. Di, Q. Yao, and L. Chen, "Searching to sparsify tensor decomposition for n-ary relational data," in WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021, J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds. ACM / IW3C2, 2021, pp. 4043–4054. [Online]. Available: https://doi.org/10.1145/3442381.3449853

[85] S. Di, Q. Yao, Y. Zhang, and L. Chen, "Efficient relation-aware scoring function search for knowledge graph embedding," in 37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021. IEEE, 2021, pp. 1104–1115. [Online]. Available: https://doi.org/10.1109/ICDE51399.2021.00100