DISTRIBUTION MATCHING GUIDED REPLAY SAMPLE SELECTION FOR ONLINE CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

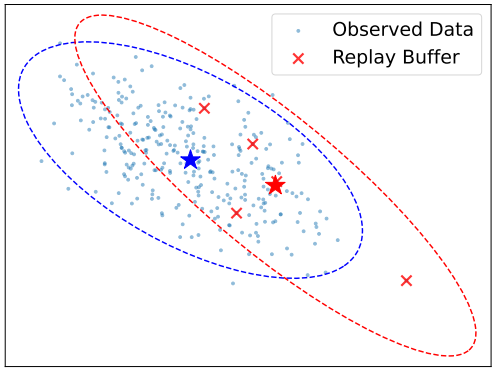
Online Class-Incremental Learning (OCIL) focuses on learning from continuously evolving data streams that are ubiquitous in dynamic environments. Data replay is an effective OCIL method, which maintains a replay buffer and learns from a mixture of replay samples and new samples to mitigate catastrophic forgetting. Previous studies have proposed various strategies for selecting replay samples; however, they often suffer from limited data availability and error accumulation, and only achieve suboptimal performance in practice. To this end, we propose a new replay sample selection strategy called *Distribution Matching Replay* (DMR). By continuously aligning the distribution of replay samples with that of all historical data, DMR significantly alleviates the issues of data availability and error accumulation. To make second-order distribution matching memory-efficient in the streaming setting, we further introduce a randomized sketch of the RFF covariance statistics, which approximates *Maximum Covariance Discrepancy* (MCD) without explicitly storing per-class covariance matrices. Extensive experiments on multiple benchmarks demonstrate that DMR outperforms state-of-the-art selection strategies and can be seamlessly integrated into various OCIL methods to further improve their performance.

1 INTRODUCTION

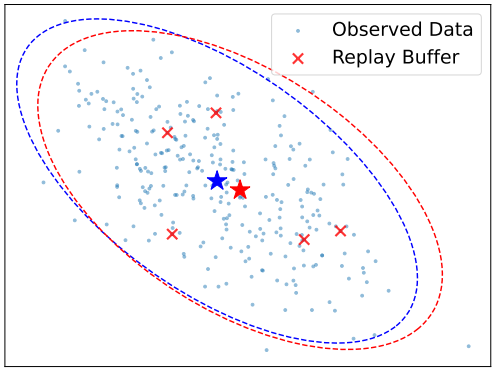
Recent studies have revealed that neural networks are susceptible to *Catastrophic Forgetting* (CF) when applied to learning with unstable data distributions. Neural networks can easily "forget" knowledge of old classes after learning new classes, resulting in significant performance degradation (McCloskey & Cohen, 1989). This restricts the applicability of neural networks to learning from streaming data commonly encountered in dynamic environments. To deal with this problem, various *Class-Incremental Learning* (CIL) methods have been proposed. Existing CIL methods can be roughly classified into two categories with different settings: *Task-Aware Class-Incremental Learning* (TACIL) and *Online Class-Incremental Learning* (OCIL). TACIL assumes that the data stream can be partitioned into a sequence of distinct tasks with disjoint classes, and the model can be trained offline on each task. On the other hand, in OCIL, the model needs to continuously learn from the data stream in a fully online manner without any previously known task identities or task boundaries.

In real-world scenarios, the task to which a sample belongs is usually unknown in advance, and the task boundaries are also unclear. Therefore, the assumption of TACIL is somewhat unrealistic. In contrast, OCIL makes no assumption on task boundaries or data partitioning. This makes it a more realistic and applicable – yet more challenging – CIL setting. In this paper, we focus on data replay-based OCIL methods (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019), which maintain a small buffer of previously encountered samples (replay samples), and learn from a mixture of new and replay samples to preserve previous knowledge and mitigate catastrophic forgetting. Compared with regularization (Kirkpatrick et al., 2017; Zenke et al., 2017; Li & Hoiem, 2018) and parameter isolation (Yoon et al., 2018; Lee et al., 2020), the replay-based methods strike a better balance between performance and efficiency and handle streaming data more easily, making it a good choice for OCIL.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Replay **w/o matching**. The distribution of replay buffer deviates far from the data observed so far.



(b) Replay **w/ matching**. The replay buffer and the data observed so far have similar distributions.

Figure 1: Illustration of DMR. The stars denote sample means; the ellipses are estimated with 99% confidence. DMR aligns the distribution of replay buffer with that of the data observed so far to maintain a representative and informative buffer.

The key to data replay is to design a strategy to select the most informative replay samples to retain in the buffer. Unfortunately, this is not easy to achieve due to the online and volatile streaming-data setting. At any point in time, the model can only utilize the current (most recently arrived) sample; all previously encountered samples, unless saved in the buffer, are permanently lost and no longer available. In such case, it is rather difficult to determine whether a replay sample is informative, *e.g.* by its similarity/adjacency to previous samples, or by its neighborhood sample density. This poses a significant challenge to the maintenance of optimal replay buffers in OCIL.

Previous studies have proposed a variety of replay sample selection strategies. Most of the strategies designed for OCIL follow two paradigms: (i) assuming that the combination of the replay buffer and the current sample $\mathcal{X}_{\text{replay}} \cup \mathcal{X}_{\text{curr}}$ effectively represents all samples $\mathcal{X}_{\text{all}}$ observed so far (Yoon et al., 2022; Tong et al., 2025); (ii) adopting model-based metrics, *e.g.* sample loss or gradients, as the criteria for selecting informative samples (Aljundi et al., 2019b; Shim et al., 2021; Buzzega et al., 2021; Kong et al., 2023). Unfortunately, these strategies do not fully address the data availability problem because $\mathcal{X}_{\text{replay}} \cup \mathcal{X}_{\text{curr}}$ only provides a limited approximation of $\mathcal{X}_{\text{all}}$, thus the selected replay samples are not strongly guaranteed to be highly informative. Moreover, the model-based strategies suffer from "error accumulation", since the replay buffer is limited in size and cannot capture all the information of historical samples. The model trained on a limited buffer is likely to be suboptimal, resulting in a suboptimal selection of replay samples, which in turn causes the model to deviate further away from optimality in subsequent training.

To tackle the aforementioned challenges, we propose a novel replay sample selection strategy called *Distribution Matching Replay* (DMR) as illustrated in Fig. 1. DMR maintains an informative and representative replay buffer by matching the distributions of the buffer and the historical data. Its underlying idea is derived from the classic two-sample problems in statistics. Specifically, given two sample sets independently obtained from two different distributions, various test statistics have been proposed to determine whether the difference between these two distributions is statistically significant. By viewing the replay buffer as one sample set and the historical data as the other, matching replay buffer with historical data is equivalent to minimizing the test statistic measuring the significance of difference.

For practical implementation, DMR measures distribution discrepancy with *Maximum Mean Discrepancy* (MMD) and *Maximum Covariance Discrepancy* (MCD). These two statistics capture complementary distributional information, with MMD characterizing discrepancies in kernel mean embeddings and MCD additionally accounting for second-order variation. We use *Random Fourier Features* (RFF) to maintain the corresponding kernel statistics online, and a randomized covariance sketch to make explicit second-order matching memory-efficient. These additive statistics support counterfactual candidate scoring without revisiting historical samples or physically modifying the replay buffer for every candidate.

The main contributions of our work can be summarized as follows:

- We alleviate limited historical-data availability in OCIL by proposing DMR, a distribution-based replay sample selection strategy that builds on the all-order distribution-matching property of RBF-MMD and maintains its RFF approximation online. To the best of our knowledge, DMR is the first method that achieves informative and representative online replay sample selection via distribution matching.
- We address the drawback of error accumulation in existing replay sample selection strategies by implicitly guiding replay sample selection with the information of the overall data distribution captured by DMR.
- We develop a memory-efficient sketched MCD for second-order distribution matching. It replaces each $d' \times d'$ RFF covariance matrix with an r-dimensional sketch, supports direct online and counterfactual add/remove updates, and admits high-probability approximation guarantees relative to full RFF-MCD.
- We demonstrate that DMR consistently outperforms state-of-the-art replay sample selection strategies and can be seamlessly integrated into various replay-based OCIL methods as a plugin to improve their performance.

2 Related Work

Model-centric replay sample selection. Model-centric strategies use the model itself to guide replay sample selection, trying to select samples that are useful to the model. One common approach is to select high-loss samples (Aljundi et al., 2019a; Buzzega et al., 2021), which contain knowledge that the model has not sufficiently learned. Some methods attempt to retain high-loss and low-loss samples at the same time to achieve a balance between past and current tasks (Millunzi et al., 2024). Another category of selection strategies uses model gradients as a metric for sample selection. For example, GSS (Aljundi et al., 2019b) formulates CIL as a constrained optimization problem and maximizes the variance of gradient directions to select diverse replay samples. GCR (Tiwari et al., 2022) minimizes a gradient approximation objective to select a weighted set of replay samples that approximate the gradients of the full dataset. There are also some approaches that utilize other information of the model to guide sample selection, *e.g.* adversarial Shapley values (Shim et al., 2021) and predicted probabilities of ground-truth labels (Kong et al., 2023).

Data-centric replay sample selection. Data-centric strategies aim to select informative replay samples based on the data characteristics of the samples themselves. A classic data-centric selection strategy is *herding* (Rebuffi et al., 2017), which selects representative replay samples by approximating class means. PICS (He & Zhu, 2021) performs selection on visual-similarity clusters to better handle high intra-class diversity. Other methods emphasize diversity or coverage: ESS (Wiewel & Yang, 2021) maximizes buffer-sample entropy, while DSS (Nokhwal & Kumar, 2023) performs a 2-approximate k-center selection. DMR differs from both paradigms by scoring candidates through class-conditional distribution matching against online summaries of all historical data, without using model-dependent importance signals.

3 Method

Without loss of generality, we suppose that a sequence of mini-batches $\mathcal{X}^{(t)} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[m_t]}$ with cardinality m_t arrives at each timestamp t, where $[a]$ denotes the integer set $\{1, 2, \ldots, a\}$. As an online setting, the data stream is volatile, i.e. only the current mini-batch $\mathcal{X}^{(t)}$ is available at time t, and each incoming sample is observed from the stream only once. The data stream consists of a sequence of m tasks $(\mathcal{T}_s)_{s\in[m]}$, where each task includes a distinct set of classes. For any sample $(\boldsymbol{x}_i, y_i) \in \mathcal{T}_s$, its task identity $\mathcal{T}_s$ is unknown during both training and inference, i.e., we adopt a *task-agnostic* setting. A replay buffer $\mathcal{X}_R$ with fixed cardinality k is maintained to mitigate forgetting. Given a neural network f parameterized by θ, the learning objective at timestamp t can be formulated as

$$\theta^\star = \operatorname*{argmin}_\theta \mathcal{L}(\mathcal{X}_S \cup \mathcal{X}^{(t)}; \theta),$$

where $\mathcal{L}$ is the loss function, and $\mathcal{X}_S \subseteq \mathcal{X}_R$ is a subset of $\mathcal{X}_R$ to reduce overfitting. The goal is to maintain an optimal $\mathcal{X}_R$ such that the learned f can generalize well.

3.1 Distribution Matching by MMD

An informative replay buffer $\mathcal{X}_R$ should enable a model trained on it to generalize well to all historical samples $\mathcal{X}_H = \mathcal{X}^{(1)} \cup \mathcal{X}^{(2)} \cup \cdots \cup \mathcal{X}^{(t)}$. Studies in various fields have revealed that the generalizability of a model depends largely on the distribution difference between the training set (replay buffer) and the test set (historical samples). For example, in *Domain Adaptation* (DA), it has been proved that the target-domain error $\epsilon_T(h)$ of model h is bounded by its source-domain error $\epsilon_S(h)$ and the empirical $\mathcal{H}\Delta\mathcal{H}$-divergence $\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$ between the source domain and the target domain (Ben-David et al., 2010). Therefore, we consider a method to select informative replay samples by matching the underlying distributions between the buffer and the historical data. To quantify this discrepancy, we use an *Integral Probability Metric* (IPM), which compares two probability measures P and Q through expectation differences over a function class $\mathcal{F}$:

$$d_{\mathcal{F}}(P, Q) = \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{\boldsymbol{x} \sim P}[f(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{y} \sim Q}[f(\boldsymbol{y})] \right|.$$

The choice of $\mathcal{F}$ trades off discriminative power and statistical tractability: $\mathcal{F}$ should be rich enough that $\mathrm{IPM}_{\mathcal{F}}(P, Q) = 0$ only when $P = Q$, while remaining sufficiently restricted for reliable estimation from finite samples. Setting $\mathcal{F}$ as the unit ball of a *Reproducing Kernel Hilbert Space* (RKHS) $\mathcal{H}$ with a universal kernel $\kappa(\cdot, \cdot)$ can achieve a good balance, yielding the *Maximum Mean Discrepancy* (MMD) metric. With kernel mean embeddings $\mu_P = \mathbb{E}_{\boldsymbol{x} \sim P}[\kappa(\boldsymbol{x}, \cdot)]$ and $\mu_Q = \mathbb{E}_{\boldsymbol{y} \sim Q}[\kappa(\boldsymbol{y}, \cdot)]$, MMD becomes

$$\begin{aligned} \mathrm{MMD}(P, Q) &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \langle f, \mu_P - \mu_Q \rangle_{\mathcal{H}} \\ &= \|\mu_P - \mu_Q\|_{\mathcal{H}} \end{aligned}$$

where the second equality follows from the Cauchy–Schwarz inequality. MMD with the RBF kernel has two properties that are particularly desirable for distribution matching. First, as a characteristic kernel, its kernel mean embedding is injective, so the population MMD is zero if and only if $P = Q$. Second, its feature representation implicitly involves statistics of all polynomial orders, as seen from the expansion

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2}\right) \sum_{j=0}^{\infty} \frac{(\boldsymbol{x}^\top \boldsymbol{y})^j}{j!\, \sigma^{2j}},$$

where σ is the RBF bandwidth. Since $(\boldsymbol{x}^\top \boldsymbol{y})^j$ contains polynomial terms of total degree j, the corresponding feature representation contains Gaussian-weighted polynomial terms of every order. Hence, matching RBF kernel mean embeddings can be interpreted as aligning an infinite collection of weighted moments. We therefore use MMD with the RBF kernel as the distribution discrepancy metric, and minimize it between the replay-buffer distribution and the historical-data distribution to realize distribution matching. For empirical samples, DMR uses the biased squared-MMD estimator, whose kernel form and estimation details are provided in Appendix A.

3.2 Empirical MMD Approximation by RFF

Applying MMD directly in OCIL faces two practical limitations. First, the historical data $\mathcal{X}_H$ observed so far is unavailable in OCIL. Since MMD computes $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for all $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ sample pairs in both $\mathcal{X}_R$ and $\mathcal{X}_H$, it is impossible to directly compute the MMD between the replay buffer and the historical data. Second, computing the entire kernel matrix is computationally expensive for large n – the time complexity is $\mathcal{O}(n^2 d)$ and the space complexity is $\mathcal{O}(n^2)$. In OCIL, data may potentially stream in without any limit, i.e. there is no upper bound on n. This makes the computational cost of MMD prohibitive.

To address these limitations, we approximate the empirical MMD with *Random Fourier Features* (RFF) (Rahimi & Recht, 2007). RFF replaces the implicit, potentially infinite-dimensional RKHS representation induced by a shift-invariant kernel with an explicit finite-dimensional Euclidean feature map. Concretely, each coordinate is a randomized cosine projection,

$$[z(\boldsymbol{x})]_j = \sqrt{2/d'}\, \cos(\mathbf{w}_j^\top \boldsymbol{x} + b_j), \qquad j = 1, \ldots, d',$$

where $\mathbf{w}_j$ is sampled from the kernel's spectral distribution and $b_j \sim \mathcal{U}_{[0,2\pi]}$. For the multi-kernel variant used in our experiments, each coordinate first samples one component bandwidth; the exact construction is provided in Appendix A. The resulting shared map $z : \mathbb{R}^d \to \mathbb{R}^{d'}$ satisfies $z(\boldsymbol{x})^\top z(\boldsymbol{y}) \approx \kappa(\boldsymbol{x}, \boldsymbol{y})$, turning kernel mean matching into ordinary feature-mean matching. Thus, the historical distribution can be summarized by an additive d'-dimensional statistic instead of retaining all historical samples or a growing kernel matrix. Replacing kernel evaluations by these finite-dimensional features gives

$$\widehat{\mathrm{MMD}}^2(X, Y) \approx \mathrm{MMD}^2_{\mathrm{RFF}}(X, Y) = \left\|\bar{z}(\boldsymbol{x}) - \bar{z}(\boldsymbol{y})\right\|_2^2, \tag{1}$$

where $\bar{z}(\boldsymbol{x})$ and $\bar{z}(\boldsymbol{y})$ are the empirical RFF means of the two sample sets. By Bochner's theorem (Rudin, 2017), the RFF inner product is an unbiased kernel estimator, and its approximation variance decreases as d' increases.

The RFF estimation of MMD (RFF-MMD) effectively tackles the two aforementioned problems in MMD and can be easily applied to OCIL. RFF-MMD does not require all historical data $\mathcal{X}_H$; it only needs to maintain a d'-dimensional feature mean vector, which is straightforward to update in an online manner. Moreover, compared with vanilla MMD, which requires $\mathcal{O}(n^2 d)$ time and $\mathcal{O}(n^2)$ kernel storage, RFF-MMD processes n samples in $\mathcal{O}(ndd')$ time and represents the historical distribution with only a d'-dimensional mean vector, apart from the shared $\mathcal{O}(dd')$ RFF projection.

3.3 Distribution Matching by MCD

Although population RBF-MMD is zero if and only if the two distributions coincide, the empirical objective used in DMR is estimated from finite samples and a finite-dimensional RFF map, and it does not explicitly emphasize covariance discrepancy in the feature space. We therefore complement mean-embedding matching with explicit covariance matching, following prior evidence that covariance alignment is useful for domain adaptation (Sun et al., 2016; Zong et al., 2016; Jiang et al., 2017).

We use the *Maximum Covariance Discrepancy* (MCD) (Zhang et al., 2020), which measures the distance between second-order moments in the RKHS:

$$\mathrm{MCD}^2(P, Q) = \left\|C[P] - C[Q]\right\|_{\mathrm{HS}}^2,$$

where $C[P] = \mathbb{E}_{\boldsymbol{x}\sim P}[\phi(\boldsymbol{x})\otimes\phi(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim P}[\phi(\boldsymbol{x})]\otimes\mathbb{E}_{\boldsymbol{x}\sim P}[\phi(\boldsymbol{x})]$ is the centered covariance operator. The empirical kernel estimator and its derivation are given in Appendix B.

RFF approximation and its memory bottleneck. As with MMD, directly evaluating kernel MCD requires unavailable historical samples and large kernel matrices. Using the same RFF mapping z as above, we approximate empirical MCD by the Frobenius distance between empirical RFF covariance matrices:

$$\widehat{\mathrm{MCD}}^2(X, Y) \approx \mathrm{MCD}^2_{\mathrm{RFF}}(X, Y) = \left\|\widehat{\mathbf{C}}_X - \widehat{\mathbf{C}}_Y\right\|_F^2, \tag{2}$$

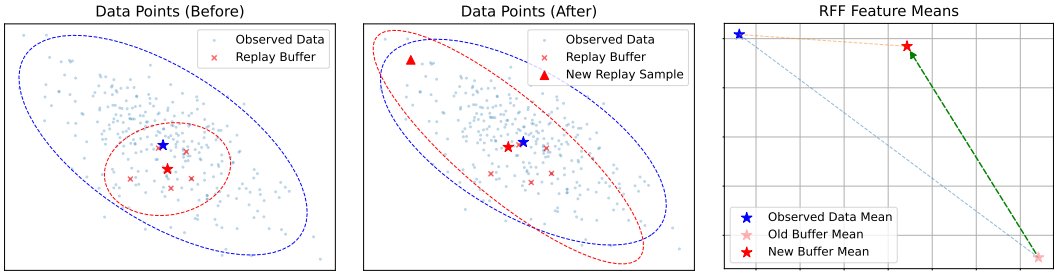
where $\widehat{\mathbf{C}}_X$ and $\widehat{\mathbf{C}}_Y$ are the empirical covariances of the RFF features; Appendix B gives the full derivation.

Sketched MCD. A direct implementation requires a $d' \times d'$ covariance matrix for each class, incurring $\mathcal{O}(d'^2)$ memory per class. DMR only needs the covariance discrepancy and its change after a candidate update, so we compress this second-order state. We use a degree-two tensorized random projection (Kar & Karnick, 2012; Meister et al., 2019). Let r be the sketch dimension and $\mathbf{U}, \mathbf{V} \in \{-1, +1\}^{r\times d'}$ be shared random-sign matrices. For $\mathbf{A} \in \mathbb{R}^{d'\times d'}$, define

$$[\mathcal{S}(\mathbf{A})]_j = \frac{1}{\sqrt{r}}\mathbf{u}_j^\top \mathbf{A}\mathbf{v}_j, \qquad j = 1, \ldots, r. \tag{3}$$

For an RFF feature $z \in \mathbb{R}^{d'}$, the sketch of its rank-one second-order feature can be computed without forming $zz^\top$:

$$\psi(z) \triangleq \mathcal{S}(zz^\top) = \frac{1}{\sqrt{r}}(\mathbf{U}z) \odot (\mathbf{V}z). \tag{4}$$



(a) Replay buffer **before** adding a new sample. A clear gap between the distributions can be observed.

(b) Replay buffer **after** adding a new sample. The distribution of the replay buffer gets much closer to the data observed so far.

(c) Adding a new sample reduces MMD (distance between feature means). The sample that causes the **greatest decrease in MMD** ($0.42 \rightarrow 0.19$) is added to the buffer.

Figure 2: DMR replay buffer update. For simplicity, we only show the addition of samples.

We then use

$$\widetilde{\mathrm{MCD}}^2_{\mathrm{RFF}}(X, Y) = \left\| \mathcal{S}(\widehat{\mathbf{C}}_X) - \mathcal{S}(\widehat{\mathbf{C}}_Y) \right\|_2^2, \tag{5}$$

which avoids materializing a $d' \times d'$ covariance matrix. Importantly, the sketch preserves the full RFF covariance discrepancy in expectation:

$$\mathbb{E}_{\mathcal{S}}\left[\|\mathcal{S}(\mathbf{A})\|_2^2 \right] = \|\mathbf{A}\|_F^2.$$

Thus, sketched MCD is unbiased for the squared Frobenius discrepancy of full RFF-MCD; increasing r reduces the approximation variance. The complete concentration bounds and candidate-selection guarantee are provided in Appendix D.

Online sufficient statistics. For each class and sample set, DMR maintains only

$$n, \qquad \mathbf{p} = \sum_{i=1}^{n} z_i, \qquad \mathbf{q} = \sum_{i=1}^{n} \psi(z_i). \tag{6}$$

With $\boldsymbol{\mu} = \mathbf{p}/n$, linearity of $\mathcal{S}$ gives the covariance sketch

$$\mathbf{c} = \mathcal{S}(\widehat{\mathbf{C}}) = \frac{\mathbf{q}}{n} - \psi(\boldsymbol{\mu}). \tag{7}$$

Thus the per-class distribution state consists of an $\mathcal{O}(d')$ mean statistic and an $\mathcal{O}(r)$ covariance sketch.

3.4 Replay Sample Selection

DMR maintains the above statistics separately for each class, using a single shared RFF map fixed after initialization. Historical statistics are updated with every incoming mini-batch, while replay statistics summarize only the current buffer. As illustrated in Fig. 2, we update the buffer by removing samples whose deletion least increases the class-wise distribution discrepancy and adding incoming samples whose insertion most decreases it.

For a replay sample $(\boldsymbol{x}_i, y_i)$ with $y_i = y$, let $\bar{z}_R^{(y)}$ and $\bar{z}_y$ denote the replay and historical RFF means. For removal, the MMD component is the increase in squared RFF-MMD,

$$\delta_{\mathrm{MMD},i} = \left\| \bar{z}_{R'}^{(y)} - \bar{z}_y \right\|_2^2 - \left\| \bar{z}_R^{(y)} - \bar{z}_y \right\|_2^2,$$

where $\mathcal{X}_{R'}^{(y)} = \mathcal{X}_R^{(y)} \setminus \{(\boldsymbol{x}_i, y_i)\}$. We remove the candidate with the smallest increase; for addition, $\delta_{\mathrm{MMD},i}$ is defined as the corresponding decrease and we select the largest. The MCD component $\delta_{\mathrm{MCD},i}$ is defined analogously from the covariance-sketch discrepancy in Eq. (5). Both components

are evaluated counterfactually from the compact additive statistics, without physically modifying the buffer for every candidate; the exact mean/sketch updates are given in Appendix C.

The final importance score is

$$\delta_i = (1-\alpha)\cdot\delta_{\mathrm{MMD},i} + \alpha\cdot\delta_{\mathrm{MCD},i}, \tag{8}$$

where α controls the contribution of second-order matching. We use *Reservoir Sampling* (Vitter, 1985) to determine the number k of occupied buffer slots to replace at each step; DMR then removes the k least informative replay samples and fills the available slots with the most informative incoming samples. The full counterfactual updates and algorithm pseudocode are given in Appendix C and Appendix E, respectively.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on four standard class-incremental benchmarks: **Seq-MNIST** (Lecun et al., 1998), **Seq-CIFAR-10**, **Seq-CIFAR-100** (Krizhevsky, 2009), and **Seq-Tiny-ImageNet** (Deng et al., 2009). All benchmarks follow the standard class-incremental setting; details are available in Appendix F. Although the benchmarks use class-based task splits, training and inference are task-agnostic and each sample is observed only once.

Evaluation metrics. We report *Average Accuracy* (Acc), $A_k = 1/k\sum_{i=1}^{k} a_{k,i}$, where $a_{k,i}$ is the accuracy on task i after learning through task k; higher is better. Results for *Average Forgetting* (Fgt) are reported in Appendix I.

Implementation details. We use an MLP with two hidden layers of size 100 for Seq-MNIST, and a ResNet-18 trained from scratch for the other datasets. All models are trained for **one** epoch per task. For DMR, we use a shared $d' = 512$ RFF map for an equally weighted mixture of $M = 10$ exponentially spaced RBF kernels, with each RFF dimension independently sampling one component bandwidth. We set $\alpha = 0.5$ in all experiments, and use sketch dimension $r = 64$ for Seq-MNIST/Seq-CIFAR-10 and $r = 256$ for Seq-CIFAR-100/Seq-Tiny-ImageNet. Bandwidth construction and remaining training details are provided in Appendix H; baseline hyperparameters follow the original papers.

4.2 Comparison with State-of-the-art Replay Sample Selection Strategies

Baselines. We compare DMR with six replay selection strategies: **GSS** (Aljundi et al., 2019b), **LABRS** (Buzzega et al., 2021), **ESS** (Wiewel & Yang, 2021), **DSS** (Nokhwal & Kumar, 2023), **CCS** (Kong et al., 2023), and **ABS** (Millunzi et al., 2024), together with *Reservoir Sampling* (**RS**) (Vitter, 1985). To isolate the selection policy, all strategies are integrated into vanilla ER (Chaudhry et al., 2019) without additional method-specific components. Baseline details are provided in Appendix G.

Results. Table 1 shows that DMR achieves the highest final average accuracy in all eight dataset–buffer settings, including the smallest buffers with only five replay samples per class on Seq-MNIST ($K = 50$) and Seq-Tiny-ImageNet ($K = 1000$). On Seq-CIFAR-100 and Seq-Tiny-ImageNet, DMR improves over RS by 1.9–4.1 percentage points (pp) across the four settings. On these two more challenging benchmarks, most competing selection strategies remain close to RS, whereas DMR retains a clear improvement at both buffer sizes. This pattern is consistent with our motivation that summarizing the historical class distributions provides a more reliable selection signal when a small current buffer cannot adequately represent complex past data. The advantage also persists when the buffer size increases, indicating that the benefit is not confined to the most memory-starved regime. The task-wise curves in Fig. 3 further show that the advantage is sustained throughout the stream. These results support distribution matching as an effective replay-selection signal under tight memory budgets. Consistent with the accuracy results, DMR also reduces catastrophic forgetting: it achieves the lowest average forgetting in six of the eight dataset–buffer settings and ranks second in the remaining two. This indicates that the accuracy gains are accompanied by stronger retention of earlier tasks; complete forgetting results are reported in Appendix I.

Table 1: Comparison of DMR with state-of-the-art sample selection strategies. Subcolumn headers indicate replay buffer size K. Results are reported as mean $\pm$ standard deviation over 5 independent trials. **Bold** and underlined values denote the best and second-best results for the dataset and buffer size, respectively.

METHOD	SEQ-MNIST		SEQ-CIFAR-10		SEQ-CIFAR-100		SEQ-TINY-IMAGENET	
	50	100	200	500	1000	2000	1000	2000
RS (1985)	59.0 $\pm$4.1	71.0 $\pm$2.8	25.6 $\pm$6.4	31.0 $\pm$4.3	14.5 $\pm$0.9	16.8 $\pm$0.2	7.7 $\pm$0.4	12.2 $\pm$0.6
GSS (2019b)	45.7 $\pm$5.4	68.9 $\pm$3.5	30.6 $\pm$5.9	37.8 $\pm$4.9	16.0 $\pm$0.6	19.0 $\pm$0.7	9.6 $\pm$0.9	13.4 $\pm$0.2
LABRS (2021)	63.1 $\pm$3.8	73.5 $\pm$1.9	26.1 $\pm$4.2	38.9 $\pm$3.8	12.3 $\pm$1.1	12.2 $\pm$1.2	7.0 $\pm$0.1	12.9 $\pm$0.3
ESS (2021)	66.8 $\pm$2.1	72.9 $\pm$2.5	26.5 $\pm$1.5	34.0 $\pm$2.7	15.4 $\pm$2.5	17.8 $\pm$1.7	6.9 $\pm$0.1	11.7 $\pm$0.4
DSS (2023)	63.9 $\pm$2.5	69.2 $\pm$1.3	24.8 $\pm$2.7	28.7 $\pm$2.2	14.4 $\pm$0.2	17.1 $\pm$1.1	7.1 $\pm$0.3	12.7 $\pm$0.2
CCS (2023)	64.4 $\pm$2.8	72.8 $\pm$2.4	26.5 $\pm$2.6	28.6 $\pm$1.5	10.2 $\pm$1.1	14.6 $\pm$1.6	6.1 $\pm$0.1	12.3 $\pm$0.2
ABS (2024)	60.4 $\pm$3.2	69.8 $\pm$1.9	26.0 $\pm$0.6	33.1 $\pm$3.0	12.6 $\pm$1.1	12.3 $\pm$0.9	7.8 $\pm$0.6	11.6 $\pm$0.3
DMR (Ours)	**76.4** $\pm$1.4	**82.1** $\pm$1.6	**31.7** $\pm$1.0	**43.9** $\pm$1.2	**18.3** $\pm$2.1	**20.9** $\pm$1.4	**10.1** $\pm$0.2	**14.1** $\pm$0.8

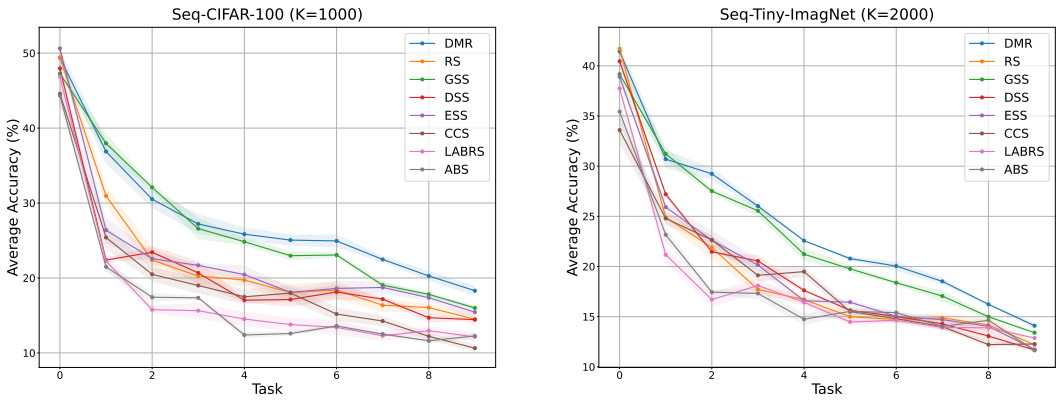


Figure 3: Average accuracies on observed tasks on Seq-CIFAR-100 with $K = 1000$ and Seq-Tiny-ImageNet with $K = 2000$. Shaded areas indicate 95% confidence intervals computed from the standard deviations across five independent trials.

4.3 INTEGRATION INTO EXISTING OCIL REPLAY METHODS

Baselines. We further replace the buffer update policy of four replay methods—**DER++** (Buzzega et al., 2020), **ER-ACE** (Caccia et al., 2022), **LODE** (Liang & Li, 2023), and **CLeWI** (Kozal et al., 2024)—with DMR, yielding the "w/ DMR" variants in Table 2.

Results. Across all valid dataset–buffer combinations, replacing the original buffer policy with DMR improves final average accuracy (30/30 comparisons in Table 2), with absolute gains ranging from 1.2 to 8.0 pp. Notably, DMR also improves LODE and CLeWI, whose vanilla variants already achieve substantially higher accuracies than DER++ on the harder benchmarks, suggesting that the gains are not merely compensating for a weak replay learner. Since these variants change only the buffer update policy while retaining the corresponding replay method, the consistent gains provide more direct evidence for the value of the selection strategy itself. Together, these results support DMR as a method-agnostic sample-selection plugin. The same pattern holds for forgetting: replacing the original buffer policy with DMR reduces average forgetting in all 30 valid comparisons. Thus, the benefits of DMR extend to retaining previously learned knowledge rather than improving final accuracy alone; detailed results are provided in Appendix I.

Table 2: Comparison of state-of-the-art replay methods with their DMR-variants ("w/ DMR"). Sub-column headers indicate replay buffer size K. Results are reported as mean $\pm$ standard deviation over 5 independent trials. **Bold** and underlined values denote the best and second-best results for the dataset and buffer size, respectively. "–" means the backbone does not support this method.

METHOD	SEQ-MNIST		SEQ-CIFAR-10		SEQ-CIFAR-100		SEQ-TINY-IMAGENET	
	50	100	200	500	1000	2000	1000	2000
DER++ (2020)	69.5±2.4	81.8±1.3	35.0±5.6	46.2±3.4	16.0±0.3	17.0±0.7	8.5±0.3	12.4±1.6
DER++ w/ DMR	77.1±1.7	86.4±1.6	39.7±3.4	52.5±1.7	17.2±1.8	20.9±1.5	10.6±1.0	15.2±0.7
ER-ACE (2022)	81.6±2.7	86.4±1.1	39.9±5.0	48.4±2.5	21.5±0.3	25.0±1.4	15.0±1.3	17.9±1.1
ER-ACE w/ DMR	87.6±1.4	89.3±1.5	47.9±1.7	55.0±1.2	**26.1±0.7**	**30.5±2.1**	17.6±1.7	22.0±0.6
LODE (2023)	85.1±1.6	88.2±0.4	52.5±1.8	57.2±1.3	22.8±0.8	23.8±2.0	17.1±0.4	22.0±1.1
LODE w/ DMR	**88.4±0.7**	**90.1±0.8**	**55.8±2.0**	**59.7±1.3**	25.1±1.6	29.8±1.5	20.6±1.1	23.7±0.3
CLeWI (2024)	–	–	52.3±1.7	57.3±1.1	22.8±1.4	24.7±1.0	18.5±0.8	24.0±1.5
CLeWI w/ DMR	–	–	54.7±1.6	58.6±1.7	25.8±1.2	26.9±1.8	**22.2±0.6**	**25.8±1.8**

4.4 ABLATION STUDIES

Component analysis. We isolate the contributions of mean matching, covariance matching, and RFF embedding. Mean matching provides the dominant gain: by itself, it improves RS by 9.2 pp on Seq-MNIST (71.0 $\rightarrow$ 80.2) and 5.2 pp on Seq-CIFAR-10 (31.0 $\rightarrow$ 36.2). Adding RFF embedding further raises accuracy to 81.8 and 42.5, with a particularly large gain on Seq-CIFAR-10. Covariance matching also provides consistent additional improvements, both without RFF embedding (81.0/37.8) and on top of it (82.1/43.9). The full configuration achieves the highest mean accuracy on both benchmarks, indicating that mean matching forms the core selection signal while RFF embedding and explicit covariance alignment provide complementary gains. Complete component results are reported in Appendix J.

Hyperparameter sensitivity and visualization. We further analyze the RFF dimension d', sketch dimension r, and mixing weight α. Increasing d' generally improves accuracy with diminishing returns, while the performance becomes relatively stable once r reaches a moderate value, indicating that a compact covariance sketch is sufficient in practice. The balanced setting $\alpha = 0.5$ performs best on both analyzed benchmarks, whereas heavily emphasizing MCD degrades accuracy, consistent with mean matching being the primary selection signal. Finally, t-SNE visualizations show that DMR selects replay buffers that better match the class distributions observed so far and are generally more diverse than those selected by the baselines. Full sensitivity curves and visualizations are provided in Appendix J.

5 CONCLUSION

We introduce a novel approach for selecting informative and representative replay samples in OCIL by matching the distributions of the replay buffer and the historical data. Building on the concept of distribution matching, we propose DMR, a novel sample selection strategy for replay-based OCIL. DMR maintains a representative replay buffer by minimizing RFF-MMD together with a memory-efficient sketched RFF-MCD between the buffer and the historical data. Through comprehensive experiments, we demonstrate that DMR outperforms state-of-the-art replay sample selection strategies, and can effectively improve the performance of various OCIL replay methods. In future work, we plan to improve DMR by adopting a dynamic mechanism to maintain first- and second-order distribution statistics. This would allow DMR to reduce memory overheads on simple tasks (one distribution for many classes) or improve performance on difficult tasks (many distributions for one class). Also, using dynamic feature means and covariance sketches makes it possible to extend DMR to *Online Domain-Incremental Learning* (ODIL).

AI USE STATEMENT

In this work, we used generative AI tools (i) to aid or polish writing, and (ii) for research ideation or execution. We used ChatGPT to assist in paper writing and to help verify the correctness of mathematical derivations and proofs. We also used OpenAI Codex to write some of the code.

We have reviewed all AI-assisted work. We have carefully reviewed the final version of the paper to ensure that its content is accurate, faithfully reflects our work, and is free from plagiarism/hallucination. We have carefully reviewed and verified all LLM-generated code to make sure that (i) it passed all hand-crafted tests; (ii) it faithfully reflects our design and is consistent with the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have submitted the relevant code in the supplementary materials, including the implementations of DMR and baselines, the Mammoth framework (upon which we built our implementations) (Buzzega et al., 2020), and other components required to reproduce this study and its experiments.

For more detailed information on the implementation of DMR, please refer to Algorithm 1 and relevant sections in the Appendix.

For more detailed information on the experiments, please refer to Appendix H.

REFERENCES

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*, pp. 11246–11255, 2019a.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*, pp. 11817–11826, 2019b.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, pp. 15920–15930, 2020.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking experience replay: a bag of tricks for continual learning. In *25th International Conference on Pattern Recognition (ICPR'20)*, pp. 2180–2187, 2021.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *10th International Conference on Learning Representations (ICLR'22)*, 2022.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision (ECCV'18)*, pp. 556–572, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pp. 248–255, 2009.

Jiangpeng He and Fengqing Zhu. Online continual learning for visual food classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW'21)*, pp. 2337–2346, 2021.

Wenhao Jiang, Cheng Deng, Wei Liu, Feiping Nie, Fu-lai Chung, and Heng Huang. Theoretic analysis and extremely easy algorithms for domain adaptive feature learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, pp. 1958–1964, 2017.

Purushottam Kar and Harish Karnick. Random Feature Maps for Dot Product Kernels. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pp. 583–591. PMLR, 2012.

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

Yajing Kong, Liu Liu, Maoying Qiao, Zhen Wang, and Dacheng Tao. Trust-region adaptive frequency for online continual learning. *International Journal of Computer Vision*, 131(7):1825–1839, April 2023. ISSN 0920-5691.

Jędrzej Kozal, Jan Wasilewski, Bartosz Krawczyk, and Michał Woźniak. Continual learning with weight interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'24)*, pp. 4187–4195, 2024. doi: 10.1109/CVPRW63382.2024.00422.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural Dirichlet process mixture model for task-free continual learning. In *8th International Conference on Learning Representations (ICLR'20)*, 2020.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

Yan-Shuo Liang and Wu-Jun Li. Loss decoupling for task-agnostic continual learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS'23)*, pp. 11151–11167, 2023.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30 (NIPS'17)*, pp. 6470–6479, 2017.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Michela Meister, Tamas Sarlos, and David Woodruff. Tight Dimensionality Reduction for Sketching Low Degree Polynomial Kernels. In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*, pp. 9475–9486, 2019.

Monica Millunzi, Lorenzo Bonicelli, Angelo Porrello, Jacopo Credi, Petter N. Kolm, and Simone Calderara. May the forgetting be with you: Alternate replay for learning with noisy labels. In *35th British Machine Vision Conference 2024 (BMVC'24)*, 2024.

Sahil Nokhwal and Nirman Kumar. DSS: A diverse sample selection method to preserve knowledge in class-incremental learning. In *10th International Conference on Soft Computing & Machine Intelligence (ISCMI'23)*, pp. 178–182, 2023.

Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*, pp. 239–247, 2013.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20 (NIPS'07)*, pp. 1177–1184, 2007.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pp. 2001–2010, 2017.

Walter Rudin. *Fourier analysis on groups*. Courier Dover Publications, 2017.

Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial Shapley value. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI'21)*, pp. 9630–9638, 2021.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pp. 2058–2065, 2016.

Rishabh Tiwari, Krishnateja Killamsetty, Rishabh K. Iyer, and Pradeep Shenoy. GCR: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pp. 99–108, 2022.

Ruilin Tong, Yuhang Liu, Javen Qinfeng Shi, and Dong Gong. Coreset selection via reducible loss in continual learning. In *13th International Conference on Learning Representations (ICLR'25)*, 2025.

Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985.

Felix Wiewel and Bin Yang. Entropy-based sample selection for online continual learning. In *28th European Signal Processing Conference (EUSIPCO'20)*, pp. 1477–1481, 2021. doi: 10.23919/Eusipco47968.2020.9287846.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *6th International Conference on Learning Representations (ICLR'18)*, 2018.

Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. In *10th International Conference on Learning Representations (ICLR'22)*, 2022.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, pp. 3987–3995, 2017.

Wenju Zhang, Xiang Zhang, Long Lan, and Zhigang Luo. Maximum mean and covariance discrepancy for unsupervised domain adaptation. *Neural Processing Letters*, 51:347–366, 2020.

Yuan Zong, Wenming Zheng, Tong Zhang, and Xiaohua Huang. Cross-corpus speech emotion recognition based on domain-adaptive least-squares regression. *IEEE Signal Processing Letters*, 23(5):585–589, 2016.

APPENDIX

A EMPIRICAL MMD AND RFF CONSTRUCTION

The squared population MMD can be written as

$$\mathrm{MMD}^2(P,Q) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim P}[\kappa(\boldsymbol{x},\boldsymbol{x}')] + \mathbb{E}_{\boldsymbol{y},\boldsymbol{y}'\sim Q}[\kappa(\boldsymbol{y},\boldsymbol{y}')] - 2\mathbb{E}_{\boldsymbol{x}\sim P,\boldsymbol{y}\sim Q}[\kappa(\boldsymbol{x},\boldsymbol{y})].$$

For samples $X = \{\boldsymbol{x}_i\}_{i=1}^m$ and $Y = \{\boldsymbol{y}_j\}_{j=1}^n$, DMR uses the biased empirical estimator

$$\begin{aligned}\widehat{\mathrm{MMD}}^2(X,Y) = \frac{1}{m^2}\sum_{i\in[m]}\sum_{j\in[m]}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) + \frac{1}{n^2}\sum_{i\in[n]}\sum_{j\in[n]}\kappa(\boldsymbol{y}_i,\boldsymbol{y}_j) \\ - \frac{2}{mn}\sum_{i\in[m]}\sum_{j\in[n]}\kappa(\boldsymbol{x}_i,\boldsymbol{y}_j).\end{aligned} \tag{9}$$

This is the squared distance between empirical kernel mean embeddings and is consistent for the population $\mathrm{MMD}^2(P,Q)$. We use the biased form because DMR relies on relative changes in a consistently defined discrepancy for sample selection; it also admits simple additive sufficient statistics and counterfactual updates.

For an equally weighted mixture of M RBF kernels,

$$\kappa_{\mathrm{MK}}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{M}\sum_{m=1}^{M}\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{h_m}\right),$$

where $h_m = 2\sigma_m^2$ is the bandwidth parameter of the m-th component. We construct a single d'-dimensional RFF map by sampling each feature dimension independently from the mixture spectral distribution. For $j = 1,\ldots,d'$, we sample

$$\begin{aligned} s_j &\sim \mathrm{Uniform}\{1,\ldots,M\}, \\ \mathbf{w}_j &\sim \mathcal{N}\left(0, \frac{2}{h_{s_j}}\mathbf{I}\right), \qquad b_j \sim \mathcal{U}_{[0,2\pi]}, \end{aligned}$$

and define

$$[z(\boldsymbol{x})]_j = \sqrt{2/d'}\,\cos(\mathbf{w}_j^\top\boldsymbol{x} + b_j).$$

This construction is an unbiased Monte Carlo approximation of the mixture kernel. For any fixed $\boldsymbol{x},\boldsymbol{y}$, averaging one feature over its random phase, component index, and frequency gives

$$\begin{aligned}&\mathbb{E}_{s_j,\mathbf{w}_j,b_j}\left[2\cos(\mathbf{w}_j^\top\boldsymbol{x}+b_j)\cos(\mathbf{w}_j^\top\boldsymbol{y}+b_j)\right] \\ &\quad= \mathbb{E}_{s_j}\mathbb{E}_{\mathbf{w}_j|s_j}\left[\cos\big(\mathbf{w}_j^\top(\boldsymbol{x}-\boldsymbol{y})\big)\right] = \frac{1}{M}\sum_{m=1}^{M}\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{h_m}\right) = \kappa_{\mathrm{MK}}(\boldsymbol{x},\boldsymbol{y}).\end{aligned}$$

Hence, $\mathbb{E}[z(\boldsymbol{x})^\top z(\boldsymbol{y})] = \kappa_{\mathrm{MK}}(\boldsymbol{x},\boldsymbol{y})$. Since the d' features are sampled independently and each summand is bounded, the variance of this kernel estimate decreases as $\mathcal{O}(1/d')$, giving the usual $\mathcal{O}_p(d'^{-1/2})$ fixed-pair Monte Carlo error. Thus, the approximation has the same convergence principle as standard RFF, but samples frequencies from the mixture spectral distribution. Each component receives a random number of feature dimensions, with expected allocation d'/M. Importantly, the resulting map still has total dimension d', comparable in size to a single d'-dimensional RFF map rather than an Md'-dimensional representation. Substituting this approximation into Eq. (9) yields the RFF-MMD form in Eq. (1).

B DERIVATION OF RFF-MCD

Note that MMD is the difference of mean functions, i.e., $\mathrm{MMD}(P,Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}}$. Analogously, MCD is defined as the difference of covariance operators: $\mathrm{MCD}(P,Q) = \|C_P - C_Q\|_{\mathrm{HS}}$ where

$$C_P = \int(\kappa(\boldsymbol{x},\cdot)-\mu_P)\otimes(\kappa(\boldsymbol{x},\cdot)-\mu_P)\mathrm{d}P(\boldsymbol{x}) = \int\kappa(\boldsymbol{x},\cdot)\otimes\kappa(\boldsymbol{x},\cdot)\mathrm{d}P(\boldsymbol{x}) - \mu_P\otimes\mu_P$$

and $x \otimes x$ is the tensor product generalized from $\boldsymbol{x}\boldsymbol{x}^\top$. Since $(\boldsymbol{x}\boldsymbol{y}^\top)\boldsymbol{z} = \boldsymbol{x}(\boldsymbol{y}^\top\boldsymbol{z})$, by defining $(x \otimes y)z = x\langle y, z\rangle_{\mathcal{H}}$ for any $x, y, z \in \mathcal{H}$, we can view C_P as an operator from $\mathcal{H}$ to $\mathcal{H}$ and define its squared Hilbert-Schmidt norm as

$$\|C_P\|_{\mathrm{HS}}^2 = \sum_{i\in\mathcal{I}} \|C_P e_i\|_{\mathcal{H}}^2$$

where $\{e_i \mid i \in \mathcal{I}\}$ is an orthogonal basis of $\mathcal{H}$. It can be shown that all the operator $A : \mathcal{H} \mapsto \mathcal{H}$ with finite Hilbert-Schmidt norm form a Hilbert space, written $\mathrm{HS}(\mathcal{H})$, with inner product

$$\langle A, B\rangle_{\mathrm{HS}} = \sum_{i\in\mathcal{I}} \langle Ae_i, Be_i\rangle_{\mathcal{H}} = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}} \langle Ae_i, e_j\rangle_{\mathcal{H}}\langle Be_i, e_j\rangle_{\mathcal{H}}.$$

where the second equality holds because $Ae_i = \sum_{j\in\mathcal{I}}\langle Ae_i, e_j\rangle_{\mathcal{H}} e_j$.

For any $x, y \in \mathcal{H}$ and any $L \in \mathrm{HS}(\mathcal{H})$, we have

$$\begin{aligned}
\langle L, x \otimes y\rangle_{\mathrm{HS}} &= \sum_{i\in\mathcal{I}} \langle Le_i, (x \otimes y)e_i\rangle_{\mathcal{H}} \\
&= \sum_{i\in\mathcal{I}} \langle Le_i, x\langle y, e_i\rangle_{\mathcal{H}}\rangle_{\mathcal{H}} \\
&= \left\langle L\sum_{i\in\mathcal{I}} e_i\langle y, e_i\rangle_{\mathcal{H}}, x\right\rangle_{\mathcal{H}} = \langle Ly, x\rangle_{\mathcal{H}}
\end{aligned}$$

In particular, if $L = a \otimes b$, then

$$\langle a \otimes b, x \otimes y\rangle_{\mathrm{HS}} = \langle (a \otimes b)y, x\rangle_{\mathcal{H}} = \langle a\langle b, y\rangle_{\mathcal{H}}, x\rangle_{\mathcal{H}} = \langle a, x\rangle_{\mathcal{H}}\langle b, y\rangle_{\mathcal{H}}$$

With the following reproducing property

$$\langle \kappa(\boldsymbol{x}, \cdot), \kappa(\boldsymbol{x}', \cdot)\rangle_{\mathcal{H}} = \kappa(\boldsymbol{x}, \boldsymbol{x}')$$

$$\langle \kappa(\boldsymbol{x}, \cdot), \mu_P\rangle_{\mathcal{H}} = \left\langle \kappa(\boldsymbol{x}, \cdot), \int \kappa(\boldsymbol{x}', \cdot)\mathrm{d}P(\boldsymbol{x}')\right\rangle_{\mathcal{H}} = \int \kappa(\boldsymbol{x}', \boldsymbol{x})\mathrm{d}P(\boldsymbol{x}')$$

$$\langle \mu_P, \mu_P\rangle_{\mathcal{H}} = \left\langle \int \kappa(\boldsymbol{x}, \cdot)\mathrm{d}P(\boldsymbol{x}), \int \kappa(\boldsymbol{x}', \cdot)\mathrm{d}P(\boldsymbol{x}')\right\rangle_{\mathcal{H}} = \iint \kappa(\boldsymbol{x}, \boldsymbol{x}')\mathrm{d}P(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}')$$

we have

$$\begin{aligned}
\|C_P\|_{\mathrm{HS}}^2 = &\left\langle \int \kappa(\boldsymbol{x}, \cdot) \otimes \kappa(\boldsymbol{x}, \cdot)\mathrm{d}P(\boldsymbol{x}), \int \kappa(\boldsymbol{x}', \cdot) \otimes \kappa(\boldsymbol{x}', \cdot)\mathrm{d}P(\boldsymbol{x}')\right\rangle_{\mathrm{HS}} \\
&+ \langle \mu_P \otimes \mu_P, \mu_P \otimes \mu_P\rangle_{\mathrm{HS}} \\
&- 2\left\langle \int \kappa(\boldsymbol{x}, \cdot) \otimes \kappa(\boldsymbol{x}, \cdot)\mathrm{d}P(\boldsymbol{x}), \mu_P \otimes \mu_P\right\rangle_{\mathrm{HS}} \\
= &\iint \kappa(\boldsymbol{x}, \boldsymbol{x}')^2\mathrm{d}P(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}') + \left(\iint \kappa(\boldsymbol{x}, \boldsymbol{x}')\mathrm{d}P(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}')\right)^2 \\
&- 2\int \left(\int \kappa(\boldsymbol{x}', \boldsymbol{x})\mathrm{d}P(\boldsymbol{x}')\right)^2 \mathrm{d}P(\boldsymbol{x}) \\
\|C_Q\|_{\mathrm{HS}}^2 = &\iint \kappa(\boldsymbol{x}, \boldsymbol{x}')^2\mathrm{d}Q(\boldsymbol{x})\mathrm{d}Q(\boldsymbol{x}') + \left(\iint \kappa(\boldsymbol{x}, \boldsymbol{x}')\mathrm{d}Q(\boldsymbol{x})\mathrm{d}Q(\boldsymbol{x}')\right)^2 \\
&- 2\int \left(\int \kappa(\boldsymbol{x}', \boldsymbol{x})\mathrm{d}Q(\boldsymbol{x}')\right)^2 \mathrm{d}Q(\boldsymbol{x}) \\
\langle C_P, C_Q\rangle_{\mathrm{HS}} = &\iint \kappa(\boldsymbol{x}, \boldsymbol{x}')^2\mathrm{d}P(\boldsymbol{x})\mathrm{d}Q(\boldsymbol{x}') + \left(\iint \kappa(\boldsymbol{x}, \boldsymbol{x}')\mathrm{d}P(\boldsymbol{x})\mathrm{d}Q(\boldsymbol{x}')\right)^2 \\
&- \int \left(\int \kappa(\boldsymbol{x}', \boldsymbol{x})\mathrm{d}Q(\boldsymbol{x}')\right)^2 \mathrm{d}P(\boldsymbol{x}) \\
&- \int \left(\int \kappa(\boldsymbol{x}', \boldsymbol{x})\mathrm{d}P(\boldsymbol{x}')\right)^2 \mathrm{d}Q(\boldsymbol{x})
\end{aligned}$$

and the squared MCD can be expressed by

$$\mathrm{MCD}^2(P,Q) = \|C_P - C_Q\|_{\mathrm{HS}}^2 = \|C_P\|_{\mathrm{HS}}^2 + \|C_Q\|_{\mathrm{HS}}^2 - 2\langle C_P, C_Q\rangle_{\mathrm{HS}}$$

In practice, since P and Q are unknown, we use the plug-in empirical estimator obtained by replacing the population covariance operators with empirical covariance operators normalized by their sample counts:

$$\begin{aligned}
\widehat{\|C_P\|_{\mathrm{HS}}^2} &= \frac{1}{m^2}\sum_{i\in[m]}\sum_{j\in[m]}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)^2 + \frac{1}{m^4}\left(\sum_{i\in[m]}\sum_{k\in[m]}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_k)\right)\left(\sum_{j\in[m]}\sum_{l\in[m]}\kappa(\boldsymbol{x}_j,\boldsymbol{x}_l)\right) \\
&\quad - \frac{2}{m^3}\sum_{i\in[m]}\left(\sum_{j\in[m]}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)\right)\left(\sum_{k\in[m]}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_k)\right) \\
&= \frac{1}{m^2}\sum_{i\in[m]}\sum_{j\in[m]}\left(\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)^2 + \frac{1}{m^2}\sum_{k\in[m]}\sum_{l\in[m]}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_k)\kappa(\boldsymbol{x}_j,\boldsymbol{x}_l)\right. \\
&\quad \left. -\frac{2}{m}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)\sum_{k\in[m]}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_k)\right) \\
&= \frac{1}{m^2}\sum_{i\in[m]}\sum_{j\in[m]}\left(\sum_{k\in[m]}\kappa(\boldsymbol{x}_i,\boldsymbol{x}_k)\left(\delta_{k,j}-\frac{1}{m}\right)\right)^2 \\
&= \frac{1}{m^2}\sum_{i\in[m]}\sum_{j\in[m]}\left[\mathbf{K}_{XX} - \frac{1}{m}\mathbf{K}_{XX}\mathbf{1}_m\mathbf{1}_m^\top\right]_{ij}\left[\mathbf{K}_{XX} - \frac{1}{m}\mathbf{K}_{XX}\mathbf{1}_m\mathbf{1}_m^\top\right]_{ji} \\
&= \frac{1}{m^2}\sum_{i\in[m]}[\mathbf{K}_{XX}\mathbf{L}_m\mathbf{K}_{XX}\mathbf{L}_m]_{ii} \\
&= \frac{1}{m^2}\mathrm{tr}(\mathbf{K}_{XX}\mathbf{L}_m\mathbf{K}_{XX}\mathbf{L}_m)
\end{aligned}$$

where $\mathbf{K}_{XX} = [\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)]_{i,j\in[m]}$ is the kernel matrix and $\mathbf{L}_m = \mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top$ is the centering matrix. Analogously we have $\widehat{\|C_Q\|_{\mathrm{HS}}^2} = \frac{1}{n^2}\mathrm{tr}(\mathbf{K}_{YY}\mathbf{L}_n\mathbf{K}_{YY}\mathbf{L}_n)$ and $\widehat{\langle C_P, C_Q\rangle}_{\mathrm{HS}} = \frac{1}{mn}\mathrm{tr}(\mathbf{K}_{XY}\mathbf{L}_n\mathbf{K}_{YX}\mathbf{L}_m)$. By putting together, we have the empirical estimator of MCD

$$\begin{aligned}
\widehat{\mathrm{MCD}}^2(X,Y) &= \widehat{\|C_P\|_{\mathrm{HS}}^2} + \widehat{\|C_Q\|_{\mathrm{HS}}^2} - 2\widehat{\langle C_P, C_Q\rangle}_{\mathrm{HS}} \\
&= \frac{1}{m^2}\mathrm{tr}(\mathbf{K}_{XX}\mathbf{L}_m\mathbf{K}_{XX}\mathbf{L}_m) + \frac{1}{n^2}\mathrm{tr}(\mathbf{K}_{YY}\mathbf{L}_n\mathbf{K}_{YY}\mathbf{L}_n) \\
&\quad - \frac{2}{mn}\mathrm{tr}(\mathbf{K}_{XY}\mathbf{L}_n\mathbf{K}_{YX}\mathbf{L}_m)
\end{aligned}$$

When RFF is applied, the kernel matrix $\mathbf{K}_{XX}$ can be approximated by $\mathbf{Z}_X^\top\mathbf{Z}_X$ and

$$\begin{aligned}
\|C_P\|_{\mathrm{HS}}^2 \approx \widehat{\|C_P\|_{\mathrm{HS}}^2} &= \frac{1}{m^2}\mathrm{tr}(\mathbf{K}_{XX}\mathbf{L}_m\mathbf{K}_{XX}\mathbf{L}_m) \\
&\approx \frac{1}{m^2}\mathrm{tr}(\mathbf{Z}_X^\top\mathbf{Z}_X\mathbf{L}_m\mathbf{Z}_X^\top\mathbf{Z}_X\mathbf{L}_m) \\
&= \left\|\frac{1}{m}\mathbf{Z}_X\mathbf{L}_m\mathbf{Z}_X^\top\right\|_F^2
\end{aligned}$$

Thus, under the RFF approximation, the empirical covariance operators are represented by $\widehat{\mathbf{C}}_X = \frac{1}{m}\mathbf{Z}_X\mathbf{L}_m\mathbf{Z}_X^\top$ and $\widehat{\mathbf{C}}_Y = \frac{1}{n}\mathbf{Z}_Y\mathbf{L}_n\mathbf{Z}_Y^\top$, yielding

$$\widehat{\mathrm{MCD}}^2(X,Y) \approx \left\|\widehat{\mathbf{C}}_X - \widehat{\mathbf{C}}_Y\right\|_F^2 = \left\|\frac{1}{m}\mathbf{Z}_X\mathbf{L}_m\mathbf{Z}_X^\top - \frac{1}{n}\mathbf{Z}_Y\mathbf{L}_n\mathbf{Z}_Y^\top\right\|_F^2.$$

C Online Statistics, Counterfactual Scoring, and Complexity

The main paper maintains, for each class and sample set, the additive statistics $(n, \mathbf{p}, \mathbf{q})$ in Eq. (6). With $\boldsymbol{\mu} = \mathbf{p}/n$, the empirical RFF covariance is

$$\widehat{\mathbf{C}} = \frac{1}{n}\sum_{i=1}^{n} z_i z_i^\top - \boldsymbol{\mu}\boldsymbol{\mu}^\top = \frac{1}{n}\left(\sum_{i=1}^{n} z_i z_i^\top - n\boldsymbol{\mu}\boldsymbol{\mu}^\top\right),$$

and linearity of $\mathcal{S}$ yields the compact covariance sketch in Eq. (7).

Candidate evaluation is counterfactual: the replay buffer need not be physically modified for every candidate. Let $n_R^{(y)}$ and $\bar{z}_R^{(y)}$ be the current replay count and RFF mean for class y. For a candidate with $z_i = z(\boldsymbol{x}_i)$, the means after removal and addition are

$$\bar{z}_{R,-}^{(y)} = \bar{z}_R^{(y)} - \frac{z_i - \bar{z}_R^{(y)}}{n_R^{(y)} - 1},$$
$$\bar{z}_{R,+}^{(y)} = \bar{z}_R^{(y)} + \frac{z_i - \bar{z}_R^{(y)}}{n_R^{(y)} + 1}.$$

For the MCD component, let $s \in \{-1, +1\}$ denote removal and addition. The sufficient statistics become

$$n' = n + s, \qquad \mathbf{p}' = \mathbf{p} + s z_i, \qquad \mathbf{q}' = \mathbf{q} + s\psi(z_i). \tag{10}$$

Substituting $(n', \mathbf{p}', \mathbf{q}')$ into Eq. (7), with $\boldsymbol{\mu}' = \mathbf{p}'/n'$, yields the counterfactual covariance sketch $\mathbf{c}' = \mathbf{q}'/n' - \psi(\boldsymbol{\mu}')$ exactly in sketch space. Let $\mathbf{c}_R^{(y)}$ and $\mathbf{c}_H^{(y)}$ denote replay and historical covariance sketches, define $\mathbf{e}_y = \mathbf{c}_R^{(y)} - \mathbf{c}_H^{(y)}$, and let $\Delta\mathbf{c}_i = \mathbf{c}'_{R,i} - \mathbf{c}_R^{(y)}$. Then

$$\widetilde{\mathrm{MCD}}_{\mathrm{RFF},\,i}^{\,2\,\prime} - \widetilde{\mathrm{MCD}}_{\mathrm{RFF}}^{\,2} = 2\mathbf{e}_y^\top \Delta\mathbf{c}_i + \|\Delta\mathbf{c}_i\|_2^2.$$

Thus, for removal we use the increase $\delta_{\mathrm{MCD},i} = \|\mathbf{e}_y + \Delta\mathbf{c}_i\|_2^2 - \|\mathbf{e}_y\|_2^2$, whereas for addition we use the corresponding decrease. Only after the best candidate is selected are the replay buffer and maintained statistics actually updated.

Space and scoring complexity. Full RFF-MCD requires $\mathcal{O}(d'^2)$ second-order state per class. Sketched MCD reduces this to $\mathcal{O}(r)$ per class, in addition to the $\mathcal{O}(d')$ first-order RFF statistic required by MMD. The RFF map is shared across classes and incurs a one-time $\mathcal{O}(dd')$ storage cost; the projection matrices $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{r\times d'}$ are likewise shared, introducing a one-time $\mathcal{O}(rd')$ global storage cost when explicitly materialized. Once z_i and $\psi(z_i)$ are available, evaluating a candidate's first- and second-order counterfactual scores requires only $\mathcal{O}(d' + r)$ vector operations and no covariance reconstruction. Computing $\psi(z_i)$ with the dense projection in Eq. (4) costs $\mathcal{O}(rd')$; structured tensor sketches can be substituted when lower projection-time overhead is desired (Pham & Pagh, 2013).

D Approximation Guarantee for the Covariance Sketch

The sketch in Eq. (3) is unbiased for the squared Frobenius norm. For any fixed $\mathbf{A} \in \mathbb{R}^{d'\times d'}$,

$$\mathbb{E}_{\mathcal{S}}\left[\|\mathcal{S}(\mathbf{A})\|_2^2\right] = \|\mathbf{A}\|_F^2.$$

A direct fourth-moment argument for the independent Rademacher rows gives the conservative variance bound

$$\mathrm{Var}_{\mathcal{S}}\left[\|\mathcal{S}(\mathbf{A})\|_2^2\right] \le \frac{8}{r}\|\mathbf{A}\|_F^4.$$

Consequently, Chebyshev's inequality yields the fixed-matrix guarantee

$$\Pr\left[\left|\|\mathcal{S}(\mathbf{A})\|_2^2 - \|\mathbf{A}\|_F^2\right| > \epsilon\|\mathbf{A}\|_F^2\right] \le \frac{8}{r\epsilon^2}. \tag{11}$$

Sharper concentration is known for the same tensorized random projection (Kar & Karnick, 2012; Meister et al., 2019). In particular, for degree two, Meister et al. (2019) show that in the high-probability regime $\delta < 1/d'^2$, a fixed vector in $\mathbb{R}^{d'^2}$ can be norm-preserved with sketch dimension

$$r = \Theta\left(\epsilon^{-2} \log \frac{d'}{\delta} + \epsilon^{-1} \log^2 \frac{d'}{\delta}\right).$$

Identifying $\mathrm{vec}(\mathbf{A})$ with such a tensor-space vector makes these results directly applicable to the RFF covariance difference. Applying them to $\mathbf{A} = \widehat{\mathbf{C}}_R - \widehat{\mathbf{C}}_H$ yields a multiplicative approximation to full RFF-MCD squared.

The same sketch also controls candidate-score perturbations. Suppose the ϵ-relative squared-norm preservation event in Eq. (11) holds for both the covariance discrepancy before a candidate update, $\mathbf{A}$, and after it, $\mathbf{A} + \Delta\mathbf{A}_i$. Since the MCD candidate score is a difference of squared norms, the triangle inequality gives

$$|\widetilde{\delta}_{\mathrm{MCD},i} - \delta_{\mathrm{MCD},i}| \le \epsilon \left(\|\mathbf{A}\|_F^2 + \|\mathbf{A} + \Delta\mathbf{A}_i\|_F^2\right).$$

For a finite set of candidates, the guarantee can be made simultaneous by applying the concentration bound with a union bound over the relevant before/after covariance discrepancies. Consequently, if the gap between the best candidate's full-RFF DMR score and every competing score exceeds the sum of their error bounds (scaled by α for the MCD component), the sketched and full-RFF versions select the same candidate. This provides a margin-dependent selection-consistency guarantee rather than requiring every potentially tiny score change to have a bounded relative error.

These guarantees characterize the approximation relative to *full RFF-MCD*; they are separate from the first RFF approximation in Eq. (2). They are also state-wise: the standard oblivious-sketch result directly applies to a fixed buffer state and finite candidate set that are not chosen adaptively as a function of the same sketch randomness. Since subsequent DMR buffers depend on earlier sketch-based selections, we do not claim a uniform high-probability guarantee over an unbounded adaptive online trajectory.

E ALGORITHM OF DMR

Algorithm 1 gives the complete DMR procedure corresponding to the scoring rules in the main paper and makes explicit how the historical and replay statistics are maintained and used for candidate scoring. For each class y, DMR keeps two tuples of sufficient statistics,

$$\mathcal{T}_H^{(y)} = (n_H^{(y)}, \mathbf{p}_H^{(y)}, \mathbf{q}_H^{(y)}), \qquad \mathcal{T}_R^{(y)} = (n_R^{(y)}, \mathbf{p}_R^{(y)}, \mathbf{q}_R^{(y)}),$$

for all historical samples and the current replay buffer, respectively. Given a tuple $\mathcal{T} = (n, \mathbf{p}, \mathbf{q})$, define its RFF mean and covariance sketch, for $n > 0$, as

$$\boldsymbol{\mu}(\mathcal{T}) = \frac{\mathbf{p}}{n}, \qquad \mathbf{c}(\mathcal{T}) = \frac{\mathbf{q}}{n} - \psi(\boldsymbol{\mu}(\mathcal{T})),$$

as in Eq. (7). For the empty tuple $(0, \mathbf{0}, \mathbf{0})$, we use the initialization convention $\boldsymbol{\mu} = \mathbf{0}$ and $\mathbf{c} = \mathbf{0}$, which makes first insertion and last removal well-defined without introducing additional state. The class-wise discrepancy represented by a replay tuple $\mathcal{T}$ is

$$\mathcal{D}_y(\mathcal{T}) = (1-\alpha)\left\|\boldsymbol{\mu}(\mathcal{T}) - \boldsymbol{\mu}(\mathcal{T}_H^{(y)})\right\|_2^2 + \alpha\left\|\mathbf{c}(\mathcal{T}) - \mathbf{c}(\mathcal{T}_H^{(y)})\right\|_2^2.$$

For a candidate with RFF feature z_i, adding or removing it only changes the corresponding replay tuple according to

$$\mathrm{Update}(\mathcal{T}, z_i, s) = \big(n + s,\ \mathbf{p} + s z_i,\ \mathbf{q} + s\psi(z_i)\big), \qquad s \in \{-1, +1\},$$

which is exactly the counterfactual update in Eq. (10). Therefore, a removal candidate is scored by the increase in $\mathcal{D}_y$, whereas an incoming candidate is scored by the decrease in $\mathcal{D}_y$. The historical statistics are updated with the entire incoming mini-batch before buffer selection, so they summarize all data observed so far irrespective of which incoming samples are eventually stored. In the pseudocode, $\mathrm{ReservoirQuota}(\mathcal{X}_R, \mathcal{X}_M, s_R)$ denotes the reservoir-sampling step used in the main

Algorithm 1 Distribution Matching Replay (detailed algorithm)

Require: Incoming mini-batch $\mathcal{X}_M$, replay buffer $\mathcal{X}_R$ with capacity s_R, shared RFF map z, shared sketch parameters $(\mathbf{U}, \mathbf{V})$, historical statistics $\{\mathcal{T}_H^{(y)}\}$, replay statistics $\{\mathcal{T}_R^{(y)}\}$, mixing weight α
Ensure: Updated replay buffer $\mathcal{X}_R$ and sufficient statistics

▷ *Encode the incoming mini-batch and update the historical target.*
for each $(\boldsymbol{x}_i, y_i) \in \mathcal{X}_M$ **do**
  if y_i is a newly encountered class **then**
    Zero-initialize $\mathcal{T}_H^{(y_i)}$ and $\mathcal{T}_R^{(y_i)}$
  end if
  $z_i \leftarrow z(\boldsymbol{x}_i)$ // cached only for the current mini-batch
  $\mathcal{T}_H^{(y_i)} \leftarrow \mathrm{Update}(\mathcal{T}_H^{(y_i)}, z_i, +1)$
end for
▷ *Determine how many occupied buffer slots should be replaced.*
$k \leftarrow \mathrm{ReservoirQuota}(\mathcal{X}_R, \mathcal{X}_M, s_R)$ // Reservoir sampling controls the replacement rate
▷ *Remove the least informative replay samples.*
for $j = 1, \ldots, k$ **do**
  for each $(\boldsymbol{x}_i, y_i) \in \mathcal{X}_R$ **do**
    $z_i \leftarrow z(\boldsymbol{x}_i)$
    $\mathcal{T}_i' \leftarrow \mathrm{Update}(\mathcal{T}_R^{(y_i)}, z_i, -1)$
    $\delta_i^- \leftarrow \mathcal{D}_{y_i}(\mathcal{T}_i') - \mathcal{D}_{y_i}(\mathcal{T}_R^{(y_i)})$ // increase after removal
  end for
  $i^\star \leftarrow \mathrm{argmin}_i\, \delta_i^-$
  $\mathcal{X}_R \leftarrow \mathcal{X}_R \setminus \{(\boldsymbol{x}_{i^\star}, y_{i^\star})\}$
  $\mathcal{T}_R^{(y_{i^\star})} \leftarrow \mathrm{Update}(\mathcal{T}_R^{(y_{i^\star})}, z_{i^\star}, -1)$
end for
▷ *Fill the available slots with the most informative incoming samples.*
while $|\mathcal{X}_R| < s_R$ **and** $\mathcal{X}_M \neq \varnothing$ **do**
  for each $(\boldsymbol{x}_i, y_i) \in \mathcal{X}_M$ **do**
    $\mathcal{T}_i' \leftarrow \mathrm{Update}(\mathcal{T}_R^{(y_i)}, z_i, +1)$
    $\delta_i^+ \leftarrow \mathcal{D}_{y_i}(\mathcal{T}_R^{(y_i)}) - \mathcal{D}_{y_i}(\mathcal{T}_i')$ // decrease after addition
  end for
  $i^\star \leftarrow \mathrm{argmax}_i\, \delta_i^+$
  $\mathcal{X}_R \leftarrow \mathcal{X}_R \cup \{(\boldsymbol{x}_{i^\star}, y_{i^\star})\}$
  $\mathcal{T}_R^{(y_{i^\star})} \leftarrow \mathrm{Update}(\mathcal{T}_R^{(y_{i^\star})}, z_{i^\star}, +1)$
  $\mathcal{X}_M \leftarrow \mathcal{X}_M \setminus \{(\boldsymbol{x}_{i^\star}, y_{i^\star})\}$
end while

paper: it returns the number k of currently occupied buffer slots to replace after accounting for any free capacity, with $0 \leq k \leq |\mathcal{X}_M|$.

The algorithm never constructs a $d' \times d'$ covariance matrix. Historical statistics are updated once per incoming sample, while replay statistics are changed only after a candidate is actually selected for removal or addition. All candidate evaluations are counterfactual vector updates of the same compact statistics, so the procedure directly implements the first- and second-order objectives in Eq. (8).

F Dataset Details

- **Seq-MNIST** divides the MNIST dataset (Lecun et al., 1998) into 5 disjoint tasks. Each task contains 2 classes and approximately 12,000 grayscale images of 28×28 pixels.

- **Seq-CIFAR-10** divides the CIFAR-10 dataset (Krizhevsky, 2009) into 5 disjoint tasks. Each task contains 2 classes and 10,000 colored images of 32×32 pixels.
- **Seq-CIFAR-100** divides the CIFAR-100 dataset (Krizhevsky, 2009) into 10 disjoint tasks. Each task contains 10 classes and 5,000 colored images of 32×32 pixels.
- **Seq-Tiny-ImageNet** divides the Tiny-ImageNet dataset (Deng et al., 2009) into 10 disjoint tasks. Each task contains 20 classes and 10,000 colored images of 64×64 pixels.

G Sample Selection Strategy Details

- **GSS** (Aljundi et al., 2019b): Select samples to maximize gradient direction diversity.
- **LABRS** (Buzzega et al., 2021): Prioritize saving samples with larger losses so as to learn from those samples that were not fitted. Also keeps the buffer class-balanced by removing old samples in the majority class.
- **ESS** (Wiewel & Yang, 2021): Select the most informative samples by maximizing the entropy of the buffer.
- **DSS** (Nokhwal & Kumar, 2023): Select samples that preserve class boundaries by running a 2-approximate k-center clustering algorithm on the dataset.
- **CCS** (Kong et al., 2023): Prioritize saving samples with higher confidence. The confidence of a sample is estimated by the model's predicted probability of the ground-truth label.
- **ABS** (Millunzi et al., 2024): Select past-task samples with the highest loss to ensure sample diversity and preserve complexity. Select current-task samples with the lowest loss to ensure sample purity.

H More Experimental Details

We use $M = 10$ equally weighted, exponentially spaced RBF bandwidths and construct a shared $d' = 512$ RFF map using the mixture-sampling scheme described in Appendix A. Each RFF dimension independently samples one of the ten bandwidths with uniform probability, so the number of dimensions assigned to each kernel is random (with expected allocation $d'/M = 51.2$), while the total feature dimension remains 512. The center bandwidth is set to the median pairwise distance in the replay buffer when the buffer first reaches capacity. Before this point, all observed samples are retained in the buffer and no sample selection is required; once the buffer is full, we instantiate the shared RFF map and initialize its historical and replay sufficient statistics from these retained samples. The bandwidth bank and RFF map are then kept fixed so that all subsequently accumulated statistics remain in a common feature space. For sketched MCD, we use a tensorized random projection with two independently sampled Rademacher matrices $U, V \in \{-1, +1\}^{r \times d'}$, whose entries take $+1$ and -1 with equal probability. For all experiments, we use a batch size of 16. For data replay, we follow the setting of *Experience Replay* (ER) (Chaudhry et al., 2019) and randomly sample a subset of the same size as the mini-batch (16 samples) from the replay buffer. We use SGD with a momentum of 0.9 as the optimizer. We set the learning rate as 1.0×10^{-3} for Seq-MNIST and Seq-CIFAR-10, and 8.0×10^{-3} for Seq-CIFAR-100 and Seq-Tiny-ImageNet.

I Experimental Results on Forgetting

We additionally experiment with the *Average Forgetting* (Fgt) (Chaudhry et al., 2018; Yoon et al., 2022; Kong et al., 2023) of different methods. Average forgetting is a widely adopted metric in CL that measures the amount of knowledge forgotten by a model during training. The average forgetting at task k is defined as $F_k = 1/(k-1) \sum_{i=1}^{k-1} \max_{j \in \{1,\dots,k-1\}} (a_{j,i} - a_{k,i})$, which reflects the average drop from the highest performance (accuracy) to the final performance in tasks $1, \dots, k$. Lower forgetting means the method suffers less from catastrophic forgetting.

The results on average forgetting are shown in Tables 3 and 4. Our DMR method is less prone to forgetting than most other selection strategies, and consistently reduces forgetting when combined with state-of-the-art replay methods. This demonstrates the effectiveness of DMR in preserving learned knowledge and improving model stability.

Table 3: Comparison of average forgetting (lower is better) for different selection strategies. Subcolumn headers indicate replay buffer size K. Results are reported as mean $\pm$ standard deviation over 5 independent trials.

METHOD	SEQ-MNIST		SEQ-CIFAR-10		SEQ-CIFAR-100		SEQ-TINY-IMAGENET	
	50	100	200	500	1000	2000	1000	2000
RS (1985)	28.6 ±4.3	20.5 ±2.9	40.1 ±6.6	35.5 ±3.2	12.9 ±1.2	11.1 ±0.5	15.9 ±0.8	12.4 ±0.7
GSS (2019b)	37.3 ±6.1	21.9 ±3.3	33.3 ±7.9	27.3 ±5.0	**8.1 ±0.9**	**5.4 ±0.7**	11.2 ±1.0	8.3 ±0.4
LABRS (2021)	24.9 ±4.3	18.8 ±2.9	39.5 ±4.9	28.1 ±3.7	10.9 ±1.7	10.2 ±0.8	14.2 ±1.1	9.9 ±0.7
ESS (2021)	22.5 ±1.9	19.2 ±2.6	38.5 ±1.2	34.7 ±3.3	13.2 ±2.3	10.0 ±2.7	16.0 ±0.1	12.9 ±0.7
DSS (2023)	21.1 ±2.8	21.2 ±1.4	40.5 ±2.6	39.7 ±1.7	13.5 ±0.2	10.4 ±1.1	11.8 ±0.4	8.6 ±0.3
CCS (2023)	23.4 ±2.9	18.0 ±2.7	36.1 ±2.5	27.0 ±1.6	11.5 ±1.6	11.3 ±1.4	12.9 ±0.5	9.1 ±0.8
ABS (2024)	27.3 ±2.7	21.9 ±1.9	38.7 ±0.9	34.9 ±4.5	10.7 ±0.8	11.4 ±0.9	14.1 ±0.4	11.0 ±0.6
DMR (Ours)	**14.7 ±1.4**	**11.4 ±1.3**	**33.2 ±1.9**	**22.7 ±2.1**	10.5 ±1.8	7.4 ±0.7	**10.7 ±0.6**	**7.8 ±1.3**

Table 4: Comparison of average forgetting (lower is better) for different replay methods and their DMR-variants. Subcolumn headers indicate replay buffer size K. Results are reported as mean $\pm$ standard deviation over 5 independent trials.

METHOD	SEQ-MNIST		SEQ-CIFAR-10		SEQ-CIFAR-100		SEQ-TINY-IMAGENET	
	50	100	200	500	1000	2000	1000	2000
DER++ (2020)	22.1 ±1.7	12.9 ±0.8	31.7 ±6.2	22.2 ±5.4	11.5 ±1.0	9.6 ±1.8	14.6 ±0.4	11.9 ±1.9
DER++ w/ DMR	16.7 ±1.0	9.3 ±1.5	24.4 ±5.3	11.2 ±3.4	8.9 ±1.6	7.0 ±1.2	13.5 ±0.7	9.8 ±0.2
ER-ACE (2022)	10.0 ±2.2	6.4 ±1.1	25.6 ±4.8	19.3 ±2.0	9.6 ±0.8	7.7 ±1.0	9.7 ±1.8	8.5 ±0.8
ER-ACE w/ DMR	5.3 ±2.0	4.3 ±1.8	13.4 ±2.9	6.8 ±0.6	4.6 ±1.4	3.6 ±1.5	7.5 ±1.4	4.9 ±1.2
LODE (2023)	4.7 ±1.6	2.6 ±0.4	12.1 ±3.0	9.6 ±1.1	5.6 ±0.2	5.1 ±1.9	6.9 ±0.6	3.7 ±0.5
LODE w/ DMR	**3.4 ±0.2**	**2.4 ±0.4**	6.1 ±2.9	3.2 ±1.0	2.8 ±1.5	**0.7 ±0.7**	**4.8 ±1.1**	**3.2 ±0.3**
CLeWI (2024)	–	–	9.4 ±0.2	8.2 ±0.6	6.1 ±0.8	4.7 ±0.3	8.6 ±1.0	4.5 ±0.6
CLeWI w/ DMR	–	–	**4.7 ±1.3**	**2.6 ±1.4**	**1.8 ±0.3**	1.1 ±0.9	4.9 ±0.7	3.8 ±1.4

J ABLATION STUDIES

This section reports the complete component results and the hyperparameter and visualization analyses referenced in the main paper.

Component analysis. Table 5 reports the complete component ablation results. Mean matching provides the primary improvement over reservoir sampling, while both RFF embedding and covariance matching yield additional gains. Combining all three components achieves the best performance on both datasets, supporting their complementary roles in DMR.

Hyperparameter analysis. Figure 4 illustrates the effect of RFF dimensionality d' on DMR. To isolate the effect of d', this ablation uses full RFF covariance matrices without sketching. We set the minimum value of d' to 2 for both datasets and the maximum value slightly above the original data dimension, i.e., 1024 for Seq-MNIST and 4096 for Seq-CIFAR-10. Overall, higher d' leads to higher average accuracy, while the gains gradually diminish as d' increases.

Figure 5 shows the effect of the MCD sketch dimensionality r. Increasing r from very small values yields clear gains, whereas the accuracy becomes relatively stable once the sketch reaches a moderate dimension. In particular, $r = 64$, used for these two benchmarks in the main experiments, attains

Table 5: Ablation study on the effectiveness of each component in DMR. "Mean" and "Cov" denote mean and covariance matching, respectively, while "Embed" denotes matching in the RFF feature space. Results are reported as mean ± standard deviation over 5 independent trials. Experiments are conducted on Seq-MNIST with $K = 100$ and Seq-CIFAR-10 with $K = 500$.

Component			Seq-MNIST	Seq-CIFAR-10
Mean	Cov	Embed		
×	×	×	71.0 ±2.8	31.0 ±4.3
✓	×	×	80.2 ±1.4	36.2 ±2.4
✓	✓	×	81.0 ±0.5	37.8 ±2.3
✓	×	✓	81.8 ±0.4	42.5 ±1.8
✓	✓	✓	**82.1 ±1.6**	**43.9 ±1.2**

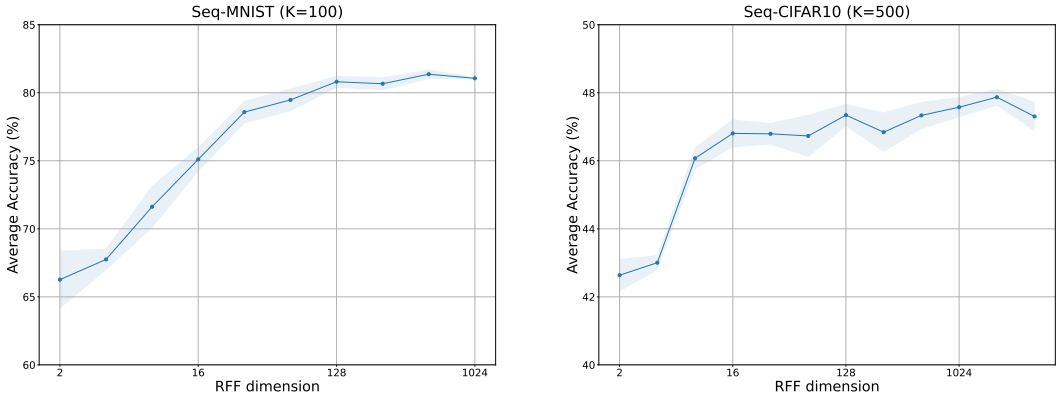


Figure 4: Illustration of the effect of RFF dimensionality d' on the final average accuracy of DMR. Experiments are conducted on Seq-MNIST ($K = 100$) and Seq-CIFAR-10 ($K = 500$). Shaded areas indicate 95% confidence intervals.

the highest mean accuracy on Seq-MNIST and is within 0.4 percentage points of the best mean accuracy on Seq-CIFAR-10. This indicates that a compact sketch can preserve the information needed for sample selection without requiring a large second-order state.

Figure 6 studies the mixing weight α, which controls the contribution of second-order (MCD) matching, while $1 - \alpha$ weights first-order (MMD) matching. The balanced setting $\alpha = 0.5$ achieves the highest mean accuracy on both datasets. Reducing α to 0.3 or 0.1 causes only modest changes, whereas increasing it to 0.9 lowers mean accuracy by 3.5 and 3.9 percentage points on Seq-MNIST and Seq-CIFAR-10, respectively, relative to $\alpha = 0.5$. This asymmetric sensitivity indicates that mean matching provides a particularly important signal for replay selection: overly emphasizing covariance matching while suppressing the mean term is detrimental. At the same time, the best performance at $\alpha = 0.5$ supports retaining both statistics rather than relying on either order alone, consistent with the component ablation.

Visualization. Figure 7 illustrates the t-SNE visualization of the replay samples selected by DMR and all seven baselines. The replay samples selected by DMR match the overall distributions of classes better than the baselines, and the replay buffers of DMR are generally more diverse.

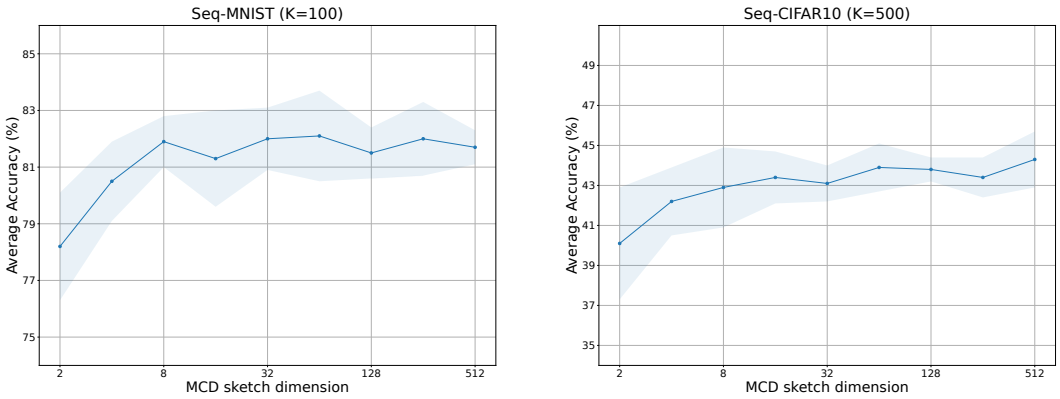


Figure 5: Illustration of the effect of MCD sketch dimensionality r on the final average accuracy of DMR. Experiments are conducted on Seq-MNIST ($K = 100$) and Seq-CIFAR-10 ($K = 500$). Shaded areas indicate 95% confidence intervals.

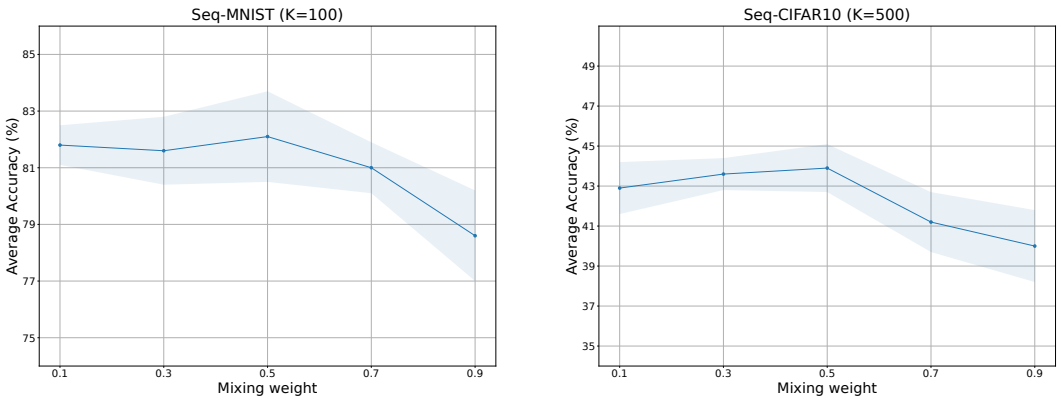


Figure 6: Illustration of the effect of mixing weight α on the final average accuracy of DMR. Experiments are conducted on Seq-MNIST ($K = 100$) and Seq-CIFAR-10 ($K = 500$). Shaded areas indicate 95% confidence intervals.

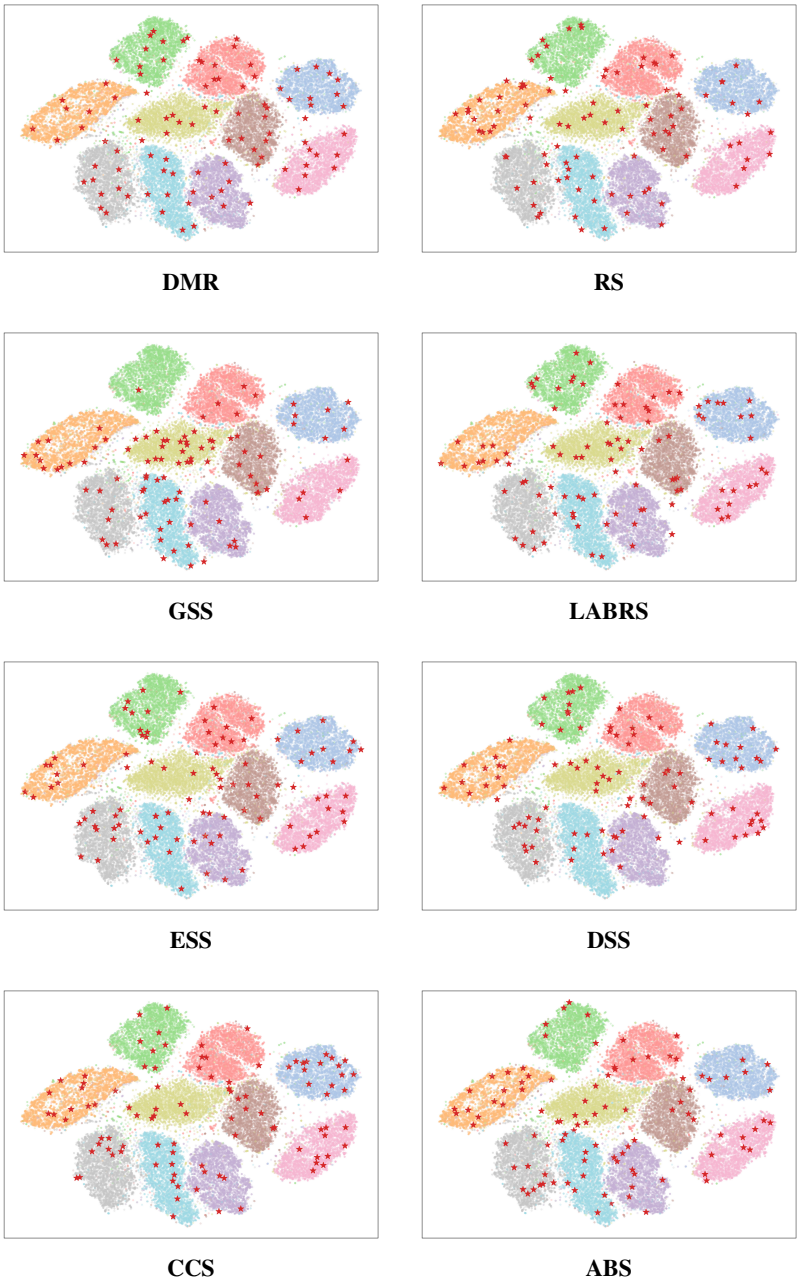


Figure 7: t-SNE visualization of replay samples selected by different strategies. Seq-MNIST dataset with $K = 100$ is used. Different colors indicate different classes. Stars indicate replay samples.