Dual-Loop Online Meta-Learning with Subspace-Aware Memory Refresh for Online Class Incremental Learning

Di Shang
Institute of Automation
Chinese Academy of Sciences
Beijing, China
shangdi2023@ia.ac.cn

Yanfeng Lu*
Institute of Automation
Chinese Academy of Sciences
Beijing, China
yanfeng.lv@ia.ac.cn

Zhiyuan Li
Institute of Automation
Chinese Academy of Sciences
Beijing, China
lizhiyuan2022@ia.ac.cn

Guoqi Li*
Institute of Automation
Chinese Academy of Sciences
Beijing, China
guoqi.li@ia.ac.cn

Lu Zhang
Institute of Automation
Chinese Academy of Sciences
Beijing, China
lu.zhang@ia.ac.cn

***Abstract*—Online class-incremental learning (OCIL) requires a model to learn a growing set of classes from a strictly one-pass, non-stationary stream, where each sample is observed only once, leading to insufficient learning of new classes and unstable representations. Despite the strong advantages of replay-based methods in OCIL, imbalanced replay and sample loss under a finite buffer further suppress plasticity, disrupt the stability-plasticity balance, and make performance highly sensitive to buffer size. Specifically, we introduce a dual-loop online meta-learning framework that meta-learns both model parameters and drift-adaptive update rules at the sample level, promoting more transferable representations and stronger plasticity under insufficient new-sample learning. We further propose Structured Replay with Subspace-Aware Refresh, which uses class-subspace relations to enable uniform replay and cyclic buffer reuse, improving buffer-size robustness and providing a more representative replay signal for meta-learning transferable representations. Extensive evaluations on standard OCIL benchmarks (T-ImageNet, CIFAR100, MNIST-P) and a challenging few-shot OCIL setting (CUB200) demonstrate state-of-the-art performance with consistently lower forgetting and stronger forward transfer. Our method remains robust across a wide range of buffer budgets while incurring runtime comparable to practical replay baselines.**

***Index Terms*—Online Class incremental learning, Computer vision, efficient replay memory**

I. Introduction

Real-world agents face evolving data distributions, motivating continual learning to acquire new knowledge sequentially while avoiding catastrophic forgetting (i.e., degraded performance on previously learned classes) [1]–[3]. In practice, however, data often arrive as a strictly one-pass stream, making repeated retraining infeasible under compute and storage constraints, especially in online-critical domains such as autonomous driving and robotic manipulation [4], [5]. This setting motivates online class-incremental learning (OCIL), a sub-scenario of continual learning. Compared with general continual learning, OCIL assumes strictly one-pass data without revisits and requires task-agnostic inference over a growing unified label space [6], [7]. Solving OCIL is key to long-horizon deployment without repeated retraining.

Unlike offline continual learning, which can revisit data across multiple epochs, the one-pass constraint in OCIL often limits optimization of new samples, leading to unstable feature representations [8]. Updates based on unstable representations may disrupt previously acquired knowledge, exacerbating catastrophic forgetting and sharpening the stability–plasticity tension. A wide range of OCIL methods have been proposed to address these challenges [9]–[11]. Among them, replay-based approaches are widely adopted and empirically strong [10], [12], [13], as they interleave buffered past samples with incoming data to retain historical knowledge. However, under strict one-pass constraints, stronger retention often comes with heavier reliance on replay, which can reduce the effective training signal for new classes and suppress plasticity. Moreover, without explicit mechanisms to preserve class-level coverage, buffer update and sampling policies may progressively underrepresent early and long-tail classes, while naïve buffer refresh that simply discards older samples when memory is full can further erase class-specific information. As a result, with fixed memory and growing tasks, performance can become highly buffer-sensitive. Together, these factors make it difficult to robustly balance stability and plasticity across memory budgets, posing a practical bottleneck for OCIL.

To address these challenges, we propose a Dual-Loop

Corresponding author: Yanfeng Lu and Guoqi Li. This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences under (Grants XDA0450000, XDA0450202), the National Science Foundation for Distinguished Young Scholars (62325603), the Central Government Guidance Funds for Local Science and Technology Development (YDZX2025124) and the National Natural Science Foundation of China (62236009, U22A20103, 91948303, 61627808).

Online Meta-Learning training strategy with Subspace-Aware Memory Refresh, which better balances stability and plasticity and improves robustness to replay-buffer capacity. (i) Specifically, we introduce a dual-loop online meta-learning framework that meta-learns both model parameters and drift-adaptive update rules (e.g., step sizes and update directions) at the sample level, promoting more transferable representations and stronger plasticity in non-stationary one-pass OCIL.(ii) We further introduce a Structured Replay Strategy with Subspace-Aware Memory Refresh to help the training scheme meta-learn more general representations and improve robustness to replay-buffer size. Structured replay keeps replay balanced via class-aware buckets and uniform cross-bucket sampling, providing a more representative signal for learning transferable representations. The subspace-aware refresh then cyclically reuses fixed memory and constrains updates using discarded class subspaces, preserving prior knowledge while maintaining high plasticity for new tasks. (iii) Extensive experiments show that our method achieves state-of-the-art continual learning under the standard OCIL protocol on T-ImageNet, CIFAR100, and MNIST-P, while maintaining a strong balance between stability and plasticity. This advantage becomes even more pronounced in the few-shot OCIL setting on CUB200. In addition, our approach is robust across varying buffer budgets with runtime comparable to most replay baselines. Further analysis highlights that two components are complementary: dual-loop online meta-learning primarily boosts plasticity, while the Structured Replay Strategy with Subspace-Aware Memory Refresh mainly improves stability-plasticity balance and buffer robustness. Code and data will be made publicly available upon acceptance.

II. Related Work

A. Online Continual Learning

online class-incremental learning (OCIL) considers learning from a non-stationary stream under a strict one-pass constraint, where past samples cannot be revisited [6], [7]. This makes the stability–plasticity balance particularly challenging, since insufficient consolidation of new data can destabilize representations and amplify interference with previously learned knowledge. Current OCIL methods can be broadly categorized into regularization-based and replay-based approaches. Regularization methods reduce forgetting by adding penalties that restrict changes to parameters important for past knowledge, like EWC and GEM. MetaCL [9] uses meta-optimization to modulate gradients and preserve important parameters. Under strict one-pass OCIL, however, such constraints may limit rapid consolidation of newly arriving classes, and their effectiveness can depend on the reliability of online importance estimates. In contrast, replay-based methods interleave buffered past samples with incoming data to retain historical knowledge, and have been widely adopted due to strong empirical performance in OCIL. ER [12] combines buffered and incoming samples with a cross-entropy loss, but uniform replay may suffer from imbalanced coverage and buffer-size sensitivity. To improve retrieval beyond uniform replay, MIR-ER [10] prioritizes samples likely to be most affected by the next update, and ESM-ER [14] combines error-sensitive sampling with episodic-semantic replay. Although beneficial for retention, it may rely more heavily on rehearsal, reducing effective learning from newly arriving data under one-pass constraints. From a memory-structure perspective, CLS-ER [15] maintains two semantic memories via exponential moving averages of model weights and a small episodic buffer. This design reduces exemplar overfitting, but may be less adaptive to evolving feature distributions and sensitive to buffer size. EEIL [16] combines exemplar rehearsal with knowledge distillation to preserve previous knowledge. CCL-DC [13] co-trains two peer learners with an entropy-regularized distillation objective, yet its effectiveness can vary with buffer size. Overall, strict one-pass OCIL poses a fundamental challenge in consolidating new classes, and existing replay-based designs may face a trade-off between retention and rapid adaptation, especially under limited memory. These considerations motivate methods that improve new-class consolidation while maintaining robust and balanced replay under memory constraints.

B. Meta-learning

Meta-learning trains over a distribution of tasks to capture shared structure that enables rapid adaptation to new tasks with only a few gradient steps [17], [18]. Classic methods are MAML [17], Reptile [19], and Meta-SGD [18]. A canonical gradient-based formulation (e.g., MAML [17])minimizes the post-adaptation loss across tasks:

$$\min_{\theta} \; \mathbb{E}_{T\sim p(T)}[\,\mathcal{L}_T^{\mathrm{qry}}(f_{\theta'})\,], \quad \theta' = \theta - \beta\,\nabla_\theta \mathcal{L}_T^{\mathrm{sup}}(f_\theta)\,, \quad (1)$$

where θ denotes the model parameters, β is the meta learning rate, and $\mathcal{L}_T^{\mathrm{sup}}$ and $\mathcal{L}_T^{\mathrm{qry}}$ are the losses computed on the support and query sets of task T, respectively.

III. Dual-Loop Online Meta-Learning with Subspace-Aware Memory Refresh

As illustrated in Fig. 1 and Algorithm 1, we propose a dual-loop online meta-learning strategy with subspace-aware memory refresh that balances stability and plasticity while making the method robust to replay buffer size. Specifically, we proposed a dual-loop meta-learning strategy which meta-learns both model parameters and drift-adaptive update rules at the sample level, promoting more transferable representations and stronger plasticity. For memory management, We further introduce a Structured Replay Strategy with Subspace-Aware Memory Refresh which uses class-subspace relations to enable uniform replay and cyclic buffer reuse, improving buffer-size robustness and helping meta-learn transferable representations. The training strategy improves plasticity via online meta-updates, while memory management supplies compatible replay samples to stabilize meta-learning across buffer sizes. Together, they form a synergistic framework that jointly improves stability and plasticity. Details of each component and how they interact are presented in the following sections.

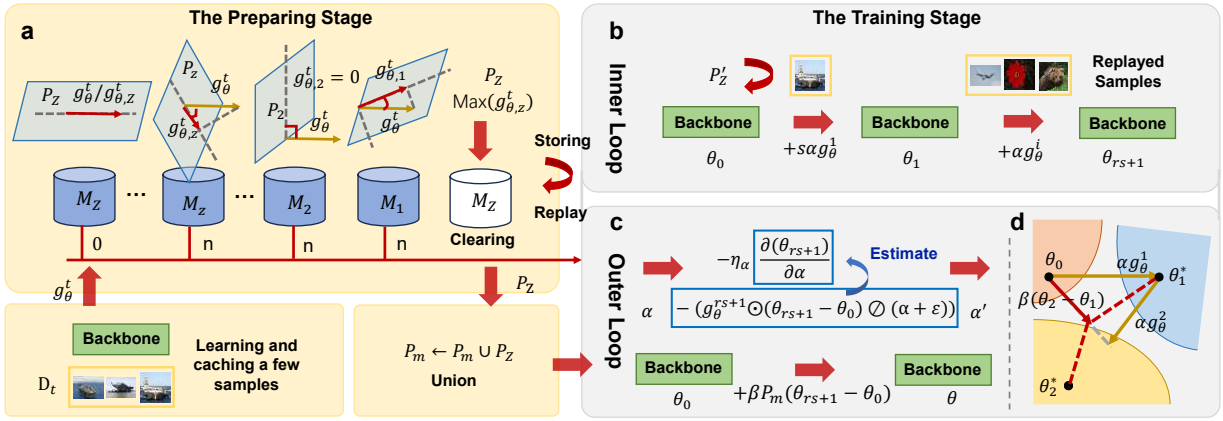


Fig. 1: **Overview of our method.** (a) Structured replaying with subspace-aware memory refresh. (b) and (c) Dual-Loop online meta-learning training. (d) Learning task-shared representation by meta learning. P is the orthogonal subspace of classes in each bucket. g is the gradient. θ is the model parameters. α and β are Hyperparameter.

A. *Dual-Loop Online Meta-Learning Training Strategy*

Replay-based OCIL under the one-pass constraint faces two key issues: new samples are under-optimized and replay can suppress plasticity for learning new classes. As shown in Fig. 1(b),(c),(d), we propose a dual-loop online meta-learning framework. It encourages cross-task generalization and preserves the plasticity in OCIL. Our approach offers two innovations. First, we move away from typical meta-learning setups. Instead, we use a dual-loop design for sample-level online meta-learning. This helps the model learn effectively from each instance in only one pass. Second, we develop a distribution-drift–adaptive online hyperparameter meta-optimization mechanism. This balances theoretical grounding with practical, efficient implementation. These designs promotes transferable representations and reduces over-alignment with old-task representations caused by replay, improve plasticity. We describe the method in detail below.

a) Sample-level Online Meta-learning: Unlike classical meta-learning, which optimizes over fully observed tasks from a fixed dataset, OCIL must learn from a non-stationary stream where each sample is observed only once. To approximate previously encountered tasks, we use experience replay. We also replace task-level meta-objectives with a streaming optimization protocol. As shown in Fig. 1 (b), at each step, the inner loop updates parameters using mixed mini-batches that combine the current sample with replayed exemplars. As shown in Fig. 1 (c), the outer loop performs a meta-update based on the changes induced by the inner loop, encouraging the model to capture transferable structure across evolving tasks under the one-pass constraint.

Inner Loop. As shown in lines 10–18 of Algorithm 1 and Fig. 1 (b), we construct a mixed set $\mathcal{S}_t = \{(x, y)\} \cup \texttt{sample}(rs, \mathcal{M})$ consisting of the incoming sample (x, y) and rs replayed exemplars from the memory $\mathcal{M}$. We then perform sample-level sequential updates over $\mathcal{S}_t$. Let $(x_i, y_i) \in \mathcal{S}_t$ denote the i-th element and $g_\theta^i = \nabla_\theta \mathcal{L}(x_i, y_i; \theta_{i-1})$. The inner-loop update is

$$\theta_i = \begin{cases} \theta_{i-1} - s\alpha\, g_\theta^i, & (x_i, y_i) \in D_t, \\ \theta_{i-1} - \alpha\, g_\theta^i, & (x_i, y_i) \notin D_t, \end{cases} \tag{2}$$

where the hyperparameter for replayed samples is α, and s scales α for the incoming sample. This mechanism is specifically designed to tackle insufficient optimization of newly arriving samples under OCIL's one-pass constraint. By scaling up the gradient update amplitude for incoming samples, we ensure the model allocates sufficient optimization focus to new data. This approach avoids under-fitting to new classes while retaining the contribution of replayed exemplars.

Outer-Loop. As shown in lines 19 of Algorithm 1 and Fig. 1(c), the outer loop performs a first-order meta-update [19] using the parameter change from inner loop:

$$\theta \leftarrow \theta_0 + \beta\, P_m\big(\theta_{r_s+1} - \theta_0\big), \tag{3}$$

where θ_{r_s+1} denotes the parameters after the inner loop, and β is the outer-loop step size. The projector P_m constrains the outer-loop meta-update to reduce interference with previously learned knowledge. More details are given in Sec. III-B. As illustrated in Fig. 1 (d), combining gradient directions from new and replayed data steers updates toward a shared descent direction that reduces both losses. This leads to more general representation and reduces over-alignment with old-task representations caused by replay.

b) Online Hyperparameter Meta-optimization: In OCIL, the data distribution drifts over time, making fixed step sizes brittle. As shown in lines 20 of Algorithm 1 and Fig. 1(c), we therefore adapt the inner-loop hyperparameter α online via meta-learning in the outer loop:

$$\alpha \leftarrow \alpha - \eta_\alpha \frac{\partial L(\theta_{r_s+1})}{\partial \alpha}, \tag{4}$$

where α is a parameter-wise step-size vector (with the same dimensionality as θ). In principle, $\partial\theta/\partial\alpha$ is a Jacobian. To make the hypergradient tractable, we keep only its diagonal

part, i.e., we ignore cross-parameter couplings and retain only per-parameter influence. Under this diagonal approximation, we use the element-wise chain rule

$$\frac{\partial L(\theta_{r_s+1})}{\partial \alpha} \approx \frac{\partial L(\theta_{r_s+1})}{\partial \theta} \odot \frac{\partial \theta_{r_s+1}}{\partial \alpha}. \tag{5}$$

$r_s + 1$ is the number of inner updates, and inner update is:

$$\theta_i = \theta_{i-1} - (w_i \alpha) \odot g_\theta^i, \quad i = 1, \ldots, r_s + 1, \tag{6}$$

where $w_i \in \{1, s\}$ scales the step size ($w_i = s$ for the incoming sample and $w_i = 1$ for replayed exemplars), and $\odot$ denotes the Hadamard product. Differentiating (6) w.r.t. α yields the recursion:

$$\frac{\partial \theta_i}{\partial \alpha} = \frac{\partial \theta_{i-1}}{\partial \alpha} - w_i g_\theta^i - (w_i \alpha) \odot \underbrace{\frac{\partial g_\theta^i}{\partial \theta_{i-1}}}_{H_i} \frac{\partial \theta_{i-1}}{\partial \alpha}, \qquad \frac{\partial \theta_0}{\partial \alpha} = 0. \tag{7}$$

Under a first-order approximation that ignores the second-order term $(w_i \alpha) \odot H_i \frac{\partial \theta_{i-1}}{\partial \alpha}$, we obtain

$$\frac{\partial \theta_{r_s+1}}{\partial \alpha} \approx - \sum_{i=1}^{r_s+1} w_i g_\theta^i. \tag{8}$$

Plugging (8) into (5) gives

$$\frac{\partial L(\theta_{r_s+1})}{\partial \alpha} \approx - g_\theta^{r_s+1} \odot \sum_{i=1}^{r_s+1} w_i g_\theta^i, \tag{9}$$

where $g_\theta^{r_s+1} := \frac{\partial L(\theta_{r_s+1})}{\partial \theta}$ denotes the gradient of the outer loss at θ_{r_s+1}. Summing (6) over $i = 1, \ldots, r_s + 1$ gives

$$\theta_{r_s+1} - \theta_0 = -\alpha \odot \sum_{i=1}^{r_s+1} w_i g_\theta^i, \tag{10}$$

$$\sum_{i=1}^{r_s+1} w_i g_\theta^i = -(\theta_{r_s+1} - \theta_0) \oslash (\alpha + \varepsilon), \tag{11}$$

where $\oslash$ is element-wise division and $\varepsilon > 0$ avoids division by zero. Substituting (11) into (9) and (3):

$$\alpha \leftarrow \alpha - \eta_\alpha (g_\theta^{r_s+1} \odot \left(\theta_{r_s+1} - \theta_0\right) \oslash (\alpha + \varepsilon)). \tag{12}$$

Directly updating a high-dimensional parameter-wise α can be noisy. We therefore optionally tie step sizes within predefined parameter groups, yielding a lower-dimensional vector $\{\alpha_g\}_{g \in \mathcal{G}}$. Let $\mathcal{I}_g$ denote the index set of parameters in group g. Given the element-wise estimate in (12), we aggregate it within each group:

$$\alpha \leftarrow \alpha - \eta_\alpha \mathrm{Agg}_{j \in \mathcal{I}_g} \left(g_{\theta,j}^{r_s+1} \left(\theta_{r_s+1,j} - \theta_{0,j}\right) / (\alpha_g + \varepsilon) \right), \tag{13}$$

where $\mathrm{Agg}(\cdot)$ is mean. We then broadcast α_g to all parameters in group g during the inner updates in (6).

B. Structured Replay with Subspace-Aware Memory Refresh

Under the one-pass and fixed-memory constraints of OCIL, replay-based methods become increasingly buffer-size sensitive as the task-to-memory ratio grows. Uniform reservoir sampling under-represents earlier and long-tail tasks, degrading historical coverage, and naive buffer refresh further discards old knowledge, undermining long-term stability. To alleviate these limitations, we propose an integrated replay optimization framework that combines structured replay with subspace-aware memory refresh. Structured replay organizes the buffer into class-aware buckets and performs cross-bucket uniform sampling to mitigate imbalance and improve historical coverage. When the buffer saturates, subspace-aware refresh enables cyclic memory reuse while preserving prior knowledge via subspace characterization and projector fusion, reducing refresh-induced forgetting. This replay module also provides more compatible historical samples, stabilizing the dual-loop online meta-learning under strict memory budgets.

a) Structured Replay Strategy.: With fixed memory and a growing number of classes, uniform reservoir sampling progressively under-represents earlier and long-tail classes, leading to a class-imbalanced buffer. Consequently, replay provides weak coverage of historical data and exacerbates forgetting. We therefore design a structured replay mechanism to enforce balanced sampling across tasks. As shown in line 11 of Algorithm 1, Fig. 1(a). Instead of maintaining a single unordered pool, we partition the replay buffer into Z class-aware buckets $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_Z\}$, where each bucket corresponds to a set of past classes. During training, we enforce balanced historical coverage via cross-bucket uniform sampling: we select z buckets (either uniformly at random or via round-robin cycling) and draw m samples from each, yielding $r_s = m \times z$ replay examples. These replay items are then combined with the incoming sample (x, y) to form a mixed mini-batch. When the number of buckets is large, we cycle through $\{\mathcal{M}_j\}$ in a round-robin manner to cap the batch size. This caps the replay-to-new ratio per mini-batch to prevent replay gradients from dominating, while round-robin cycling ensures coverage of older tasks over time.

b) Subspace-Aware Memory Refresh: Structured replay fixes sample imbalance, but when the buffer is full in fixed-memory OCIL, refreshing it can drop old samples, weakening long-term stability. To resolve this, we propose a subspace-aware memory refresh mechanism that achieves cyclic utilization of fixed memory while preserving the integrity of prior knowledge. As shown in Fig. 1, we represent the orthogonal subspace of classes in each bucket with a projection operator and use subspace orthogonality to pick the most orthogonal buckets for cyclic reuse. During refresh, we fold the replaced bucket's projection operator into a global constraint operator P_m, so that historical knowledge is retained. This mechanism bridges the gap between memory efficiency and knowledge retention, further alleviating the stability-plasticity tension. We elaborate on the detailed workflow below.

Task Subspace Characterization and Projector. For each

new task in OCIL, we characterize orthogonal subspace of classes by constructing an orthogonal-complement projector:

$$P_t \;=\; I_d \;-\; D_t\big(D_t^\top D_t + \gamma I_n\big)^{-1} D_t^\top, \tag{14}$$

where $D_t \in \mathbb{R}^{d\times n}$ stacks features extracted from the backbone's penultimate layer (d is the feature dimension and n the number of samples). Here I_d and I_n are identity matrices of sizes d and n, respectively, and γ is a small damping term that stabilizes the inversion in the one-pass setting, where $D_t^\top D_t$ may be ill-conditioned or singular.

Memory Refresh and Subspace-Based Replacement. For each bucket M_t, we store its projector P_t. As shown in lines 2-9 of Algorithm 1 and Fig. 1(a), when the buffer is full, we trigger a refresh and choose the bucket to replace using a subspace orthogonality criterion. Concretely, for the incoming classes we form a representative subset C and compute its loss gradient at the pre-update parameters θ_0, then select

$$j^* \leftarrow \arg\max_{j\in\{1,\dots,Z\}} \left\|P_j \nabla_\theta \mathcal{L}(C;\theta_0)\right\|_2, \tag{15}$$

where P_j encodes the orthogonal-complement subspace of bucket M_j. A larger value implies the new task is more orthogonal to bucket M_j (i.e., less feature overlap). By replacing the most orthogonal bucket, the new task becomes more aligned with the remaining buckets' subspaces, enabling more effective replay and strengthening generalizable representations. Meanwhile, merging the replaced bucket's projector into the global constraint introduces the smallest additional restriction on the new gradients, thereby preserving plasticity.

Projector Merging and Global Constraint Update. As shown in lines 2-9 of Algorithm 1 and Fig. 1(a), after selecting M_{j^*} for replacement, we merge its projector P_{j^*} into the global constraint operator P_m, which summarizes the orthogonal-complement subspace of previously seen but unstored data. Because P_{j^*} and P_m are only approximately projectors under regularization (i.e., not perfectly idempotent), $I - P_i$ cannot project onto the target subspace accurately. We first symmetrize them to reduce numerical asymmetry:

$$P_i \leftarrow \tfrac{1}{2}(P_i + P_i^\top), \quad i \in \{j^*, m\}. \tag{16}$$

We construct a stacked matrix A to integrate the subspace information of P_{j^*} and P_m:

$$A = \begin{bmatrix} I - P_{j^*} \\ I - P_m \end{bmatrix}, \tag{17}$$

where $I - P_{j^*}$ and $I - P_m$ correspond to the task-feature subspaces of the evicted bucket and the previously summarized data, respectively. The approximate null space of A captures directions that are simultaneously orthogonal to both subspaces. To extract it efficiently in the streaming setting, we compute the thin SVD (retaining only non-zero singular values to reduce computational complexity for streaming scenarios):

$$A = U\Sigma V^\top, \qquad \Sigma = \mathrm{diag}(\sigma_1 \geq \cdots \geq \sigma_K \geq 0), \tag{18}$$

and select the right singular vectors associated with near-zero singular values using the threshold $\tau = \varepsilon\,\sigma_{\max}(A)$, where $\varepsilon \in [10^{-8}, 10^{-6}]$:

$$\mathcal{K} = \{k : \sigma_k \leq \tau\}, \qquad V_\cap = [v_k]_{k\in\mathcal{K}}. \tag{19}$$

Finally, we update the global constraint by projecting onto the extracted null-space component:

$$P_m^* \approx V_\cap V_\cap^\top. \tag{20}$$

Practical Application of P_m. As in (3), we apply P_m^* in the outer loop to project parameter updates onto directions orthogonal to the historical subspaces, preserving prior knowledge while still allowing effective learning on the new task.

Algorithm 1 Training Strategy of Proposed Method

Input: θ_0 denotes the model parameters, $\{M_i\}_{i=1}^z$ denotes occupied replay memory in Z replay memory blocks, P_t and P_m denotes the projector that constrains gradient update. α is the hyperparameter. β is the meta learning rate.

Output: θ denotes the parameters of a model trained.

1: $M_{now} \leftarrow M_{z+1}$
2: **if** $z = Z$ **then**
3: $C = sample(D_t)$
4: $j^* \leftarrow \arg\max_{j\in\{1,\dots,Z\}}\{P_j \nabla_\theta \mathcal{L}(C;\, \theta_0, \phi_0)\}$
5: $P_{j^*} \leftarrow (P_{j^*} + P_{j^*}^\top)/2$
6: $A \leftarrow \begin{bmatrix} I - P_{j^*} \\ I - P_m \end{bmatrix}$ SVD: $A = U\Sigma V^\top$
7: Select $\mathcal{K} = \{k : \sigma_k \leq \varepsilon\sigma_{\max}(A)\}$ $V_\cap \leftarrow [v_k]_{k\in\mathcal{K}}$
8: $P_m^* \approx V_\cap V_\cap^\top$ $M_{now} \leftarrow M_{j^*} \leftarrow \{\}$
9: **end if**
10: **for** (x, y) in D_t **do**
11: $S_t = (x, y) \cup sample(rs, M)$
12: **for** (x_i, y_i) in S **do**
13: **if** $(x_i, y_i) \in D_t$ **then**
14: $\theta_i = \theta_{i-1} - s\alpha g_\theta^i$ P_t modified
15: **else**
16: $\theta_i = \theta_{i-1} - \alpha g_\theta^i$
17: **end if**
18: **end for**
19: $\theta = \theta_0 + \beta P_m(\theta_{rs+1} - \theta_0)$ $M_{now} = M_{now} \cup \{(x, y)\}$
20: $\alpha \leftarrow \alpha - \eta_\alpha(g_\theta^{r_s+1} \odot \big(\theta_{r_s+1} - \theta_0\big) \oslash (\alpha + \varepsilon))$
21: **end for**
22: **return** θ

IV. Experiments

A. Experimental Setup

a) Datasets: We evaluate our method under the online class-incremental learning (OCIL) protocol on four image-classification benchmarks, covering both standard and few-shot regimes. MNIST-P [20] contains 200 classes in 20 sessions constructed by applying fixed pixel permutations to grayscale digit images, CIFAR-100 [21] contains 100 classes in 10 sessions, and T-ImageNet [22] is a hierarchical subset of

ImageNet [22] with 64 classes in 16 sessions; for MNIST-P and CIFAR-100, each class provides 20 training samples. For T-ImageNet, we keep the first session at its natural sample count and set the others to 50 samples per class. We further consider few-shot OCIL on CUB-200 [23], an imbalanced fine-grained bird benchmark with 200 classes in 10 sessions, where we keep the first session at their natural sample count and set the others to 5 samples per class.

b) Implementation Details: We adopt a one-pass setting (epoch= 1, batch size= 1) in which past data are available only via a bounded replay buffer, and all methods follow the same stream order and memory budget on each benchmark. We use a fully connected network for MNIST-P with a buffer size of 500, a convolutional network for CIFAR-100 with a buffer size of 250, and ResNet-50 for CUB-200 and T-ImageNet with buffer sizes of 250 and 500, respectively. All experiments are implemented in PyTorch and executed on a single NVIDIA A800-SXM4-80GB GPU.

c) Metrics: Following [24], we evaluate the model on all classes in T sessions throughout training. After finishing session t, we test on every session $j \in \{1, \ldots, T\}$ and obtain an accuracy matrix $\mathbf{R} \in \mathbb{R}^{T\times T}$, where $R_{t,j}$ is the test accuracy on session j after learning task t. Let r_j denote the test accuracy on task j from random initialization. We report the following metrics. **Average Accuracy (ACC):** $ACC \triangleq \frac{1}{T}\sum_{j=1}^{T} R_{T,j}$. It measures the mean performance over the all classes after learning. **Forgetting Measure (FM):** $FM \triangleq \frac{1}{T-1}\sum_{j=1}^{T-1}\left(R_{T,j} - R_{j,j}\right)$. It averages the performance drop on past sessions since first learning. More negative means more forgetting. **Forward Transfer (FT):** $FT \triangleq \frac{1}{T-1}\sum_{i=2}^{T}\left(R_{i-1,i} - r_i\right)$. It measures transfer to the next session before training, relative to random initialization; higher is better.

d) Baselines: We categorize the baselines into: (i) **regularization-based** methods that restrict updates to preserve past knowledge, including EWC [25], MetaCL [9], and GEM [26]; (ii) **replay-based** methods that rely on a memory buffer (optionally with sample selection or distillation), including ER [12], MIR [10], EEIL [16], CCL-DC [13], CLS-ER [15], and ESM-ER [14]. We further include fine-tuning and a joint-training oracle as lower and upper bounds.

B. Results and Analysis

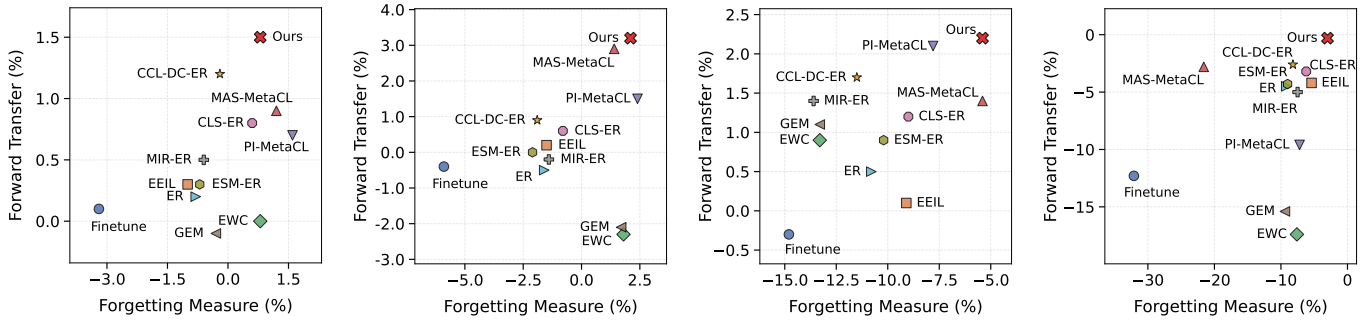
a) Average Accuracy on Standard OCIL: To evaluate overall continual learning capability of methods under the standard OCIL protocol, as shown in Tab. I, we measure the final Average Accuracy (ACC) on three benchmarks: T-ImageNet, CIFAR100, and MNIST-P. Our method yields consistent and substantial improvements across all three benchmarks, surpassing the previous state-of-the-art method CCL-DC-ER by $+1.6\%$ ACC on T-ImageNet (40.2% vs. 38.6%), $+3.3\%$ ACC on CIFAR100 (43.1% vs. 39.8%), and $+3.0\%$ ACC on MNIST-P (75.0% vs. 72.0%). Notably, on CIFAR100 and MNIST-P, our method even exceeds joint training, suggesting that the proposed dual-loop meta-learning strategy learns more transferable and general representations from the non-stationary stream rather than simply fitting each session. In combination with subspace-aware memory refresh that preserves class-level coverage, this design mitigates catastrophic forgetting and eases the stability–plasticity tension under fixed memory constraints.

TABLE I: Average Accuracy (ACC, %) on T-ImageNet, CIFAR100, and MNIST-P under standard OCIL, and on CUB-200 under few-shot OCIL. Results are reported as mean ± std over N runs.

	T-Imagenet	CIFAR100	MNIST-P	CUB200
Methods	ACC(%)	ACC(%)	ACC(%)	ACC(%)
Fine tune	28.4 ±1.1	31.7 ± 1.0	46.8 ± 0.6	17.2 ± 1.0
EWC	32.1 ±0.6	34.5 ± 0.6	66.6 ± 0.2	29.5 ± 1.1
MAS-MetaCL	33.2 ±0.4	37.5 ± 0.9	57.7 ± 0.6	28.7 ± 1.2
PI-MetaCL	32.5 ±1.4	38.2 ± 0.4	56.0 ± 0.4	31.5 ± 1.0
GEM	33.5 ±0.3	32.3 ± 0.4	65.7 ± 0.5	26.4 ± 1.0
EEIL	31.3 ±1.2	33.1 ± 0.3	63.2 ± 0.4	21.5 ± 1.0
ER	35.0 ± 0.5	36.1 ± 0.4	68.8 ± 0.2	31.2 ± 1.0
MIR-ER	36.8 ± 0.5	38.2 ± 0.5	70.2 ± 0.1	32.8 ± 1.0
CLS-ER	37.8 ± 0.4	39.0 ± 0.4	71.0 ± 0.3	33.5 ± 1.0
ESM-ER	36.0 ± 0.4	35.9 ± 0.3	69.8 ± 0.4	31.8 ± 1.0
CCL-DC-ER	38.6 ± 0.4	39.8 ± 0.4	72.0 ± 0.2	34.6 ± 1.0
Ours	**40.2 ± 0.4**	**43.1 ± 0.2**	**75.0 ± 0.3**	**38.1 ± 0.9**
Joint training	40.3 ± 0.2	40.0 ± 0.5	68.2 ± 0.2	35.6 ± 0.8

b) Stability and Plasticity on Standard OCIL: We further evaluate stability and plasticity of methods under the standard OCIL protocol. On T-ImageNet, CIFAR100, and MNIST-P, we assess the Forgetting Measure (higher means less forgetting) for stability and the Forward Transfer (higher means better transfer) for plasticity. These results are shown in Fig 2 (a), (b) and (c). Our method consistently excels at Forward Transfer across all benchmarks, indicating strong plasticity in a non-stationary one-pass stream. This aligns with our dual-loop online meta-learning, which meta-learns parameters and update rules at the sample level. While some baselines achieve better forward transfer on certain datasets, they typically show more forgetting (further to the left). Methods with slightly higher Forgetting Measure generally have much lower forward transfer (further down), reflecting an imbalanced stability-plasticity behavior. In contrast, our method remains closer to the top-right region across three benchmarks, demonstrating improved plasticity without sacrificing stability. This validates that dual-loop meta-learning and subspace-aware memory refresh work together: the former learns generalizable representations and robust updates, while the latter preserves representative class coverage under fixed memory.

c) Performance on few-shot OCIL.: To further test the stability-plasticity balance under extremely limited supervision, we evaluate on CUB200 under the few-shot OCIL setting (see Sec. IV-A for the protocol). As shown in Tab. I, our method achieves the best final ACC on CUB200, reaching $38.1\% \pm 0.9$. This improves over the prior SOTA CCL-DC-ER by $+3.5$ points ($34.6\% \pm 1.0$). Notably, this gain is larger than on the standard OCIL benchmarks: $+1.6$ on T-ImageNet, $+3.3$



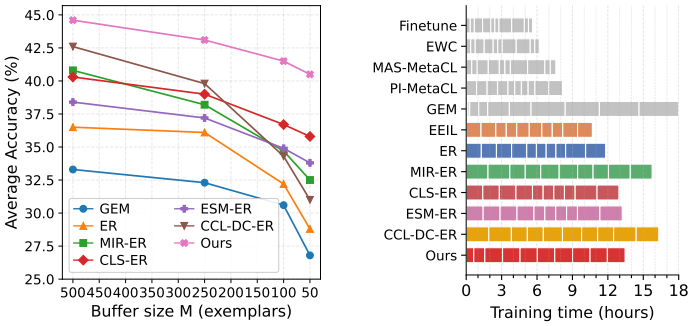
(a) T-Imagenet (b) CIFAR100 (c) MNIST-P (d) CUB200

Fig. 2: Stability–plasticity on (a) T-ImageNet, (b) CIFAR100 and (c) MNIST-P under standard OCIL, and on (d) CUB200 under few-shot OCIL. We plot Forgetting Measure (x-axis) against Forward Transfer (y-axis), where higher is better for both. Each marker denotes a method, and the top-right region corresponds to a better balance between stability and plasticity.

on CIFAR100, and +3.0 on MNIST-P. This suggests that our advantage is even more pronounced when the stream is more data-scarce, and the stability-plasticity tension is amplified. The stability-plasticity plot in Fig. 2(d) supports the same conclusion. Compared with the standard OCIL benchmarks, our method on CUB200 achieves a more decisive dominance. It outperforms all baselines on both forward transfer and forgetting measure. This behavior aligns with our design: dual-loop online meta-learning meta-learns parameters and update rules at the sample level to improve data efficiency and plasticity. Meanwhile, subspace-aware memory refresh maintains representative class coverage within a fixed memory budget, stabilizing learning and reducing interference.

d) Robustness to Buffer Size.: To evaluate the robustness of replay-based methods to memory capacity under the one-pass OCIL constraint, we vary the replay buffer size on CIFAR100 and report ACC for $M \in 50, 100, 250, 500$ in Fig. 3(a). Our method consistently achieves the best accuracy and degrades more slowly as the buffer shrinks, indicating stronger robustness to buffer size. Importantly, the performance gap widens in the low-memory regime: at M=50 and M=100, we outperform the strongest competitor CLS-ER by +4.7% and +4.8%, while at M=250 and M=500 we improve over CCL-DC-ER by +3.3% and +2%, respectively. This robustness comes from our structured replay with subspace aware memory refresh. Class aware bucketing and cross bucket sampling maintain coverage of early and long tail classes as the class to memory ratio grows. When the buffer is full, the subspace aware refresh cyclically reuses fixed memory and constrains updates using discarded class subspaces, reducing the loss of task specific information that often makes replay highly sensitive to buffer capacity. Together, these mechanisms alleviate the buffer sensitivity bottleneck in one pass OCIL and maintain stable performance across memory budgets.

e) Efficiency Analysis: In data-stream learning, low processing time is crucial to keep pace with incoming data. We therefore report the wall-clock training time measured at the end of the last task. All methods are evaluated under identical



(a) ACC vs. Replay Buffer Size (b) Training time

Fig. 3: Impact of replay buffer size on average accuracy and computational cost on CIFAR-100 under standard OCIL. (a) Average accuracy (%) trends across replay buffer sizes M (number of stored exemplars). (b) Total training time (hours) under the same setting, with M=250 fixed for all methods.

settings. As shown in Fig. 3(b), lightweight regularization methods are the fastest, while replay-based methods naturally require more time. Importantly, our method has a runtime comparable to standard replay baselines (e.g., ER/CLS-ER/ESM-ER,12–13 hours), indicating only a modest overhead. In contrast, more computation-intensive replay strategies (e.g., GEM, MIR-ER, CCL-DC-ER) are substantially slower, taking roughly 15–18 hours. Overall, the results suggest that our approach achieves a favorable efficiency profile, matching the runtime of strong and practical replay baselines while avoiding the substantial overhead of more expensive methods.

C. Ablation Study

We analyze two core components of our framework: the dual-loop online meta-learning training strategy (OM) and the structured replay strategy with subspace-aware memory refresh (SR). OM is designed to improve adaptation in the one-pass, non-stationary OCIL stream, while SR aims to mitigate the fixed-memory limitation by maintaining more balanced

TABLE II: Ablation results (ACC%, higher is better) on standard OCIL (T-ImageNet/CIFAR100/MNIST-P) and few-shot OCIL (CUB-200). SR: SGD + structured replay with subspace-aware refresh; OM: dual-loop online meta-learning + ER; Ours: OM + SR (full model).

	T-Imagenet	CIFAR100	MNIST-P	CUB200
Methods	ACC(%)	ACC(%)	ACC(%)	ACC(%)
SR	36.9 ±1.1	38.3 ± 1.0	70.5 ± 0.6	31.6 ± 0.4
OM	35.2 ±1.2	39.1 ± 0.3	71.3 ± 0.4	33.2 ± 1.0
Ours	**40.2 ± 0.4**	**43.1 ± 0.2**	**75.0 ± 0.3**	**38.1 ± 0.9**

and informative replay. We evaluate OM-only and SR-only variants on standard OCIL (T-ImageNet/CIFAR100/MNIST-P) and few-shot OCIL (CUB200) under the same protocol and memory budget as the full model. Tab. II reports the average accuracy. Across all benchmarks, the full model consistently outperforms either component alone, showing that OM and SR offer complementary rather than redundant benefits. Specifically, Ours improves ACC from 35.2 to 40.2 on T-ImageNet (+5.0 over OM) and from 36.9 to 40.2 (+3.3 over SR), with similar gains on CIFAR100 (43.1 vs. 39.1/+4.0 and 38.3/+4.8) and MNIST-P (75.0 vs. 71.3/+3.7 and 70.5/+4.5). The largest improvement appears on few-shot CUB-200, where Ours reaches 38.1 compared to 33.2 (+4.9) and 31.6 (+6.5), indicating stronger synergy under scarce supervision. Notably, OM and SR show different strengths across stream difficulty. On more homogeneous benchmarks (CIFAR100/MNIST-P/CUB200), OM outperforms SR, suggesting meta-updates work well when replay shift is mild. On T-ImageNet, where classes are hierarchical and more diverse, SR performs better, supporting the view that higher heterogeneity exacerbates buffer imbalance and weakens the effect of meta-learning. Overall, OM and SR are complementary, and removing either component causes a clear performance drop, confirming that both are necessary to achieve robust gains across diverse online continual learning scenarios.

V. CONCLUSION

We proposed a Dual-Loop Online Meta-Learning framework with Subspace-Aware Memory Refresh for one-pass OCIL, targeting catastrophic forgetting, the stability–plasticity trade-off, and buffer sensitivity. Across four benchmarks, our method achieves state-of-the-art accuracy, with particularly strong gains in few-shot and low-memory settings, while keeping runtime comparable to practical replay baselines. Ablations further verify the complementarity of online meta-learning and structured replay: the former improves plasticity via sample-level meta-updates, and the latter enhances stability and robustness through class-aware memory management. Future work will extend this task-agnostic training strategy to more complex tasks (e.g., reinforcement learning).

REFERENCES

[1] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE transactions on pattern analysis and machine intelligence*, pp. 5362–5383, 2024.

[2] M. De L, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, pp. 3366–3385, 2021.

[3] S. Tian, L. Li, W. Li, H. Ran, X. Ning, and P. Tiwari, "A survey on few-shot class-incremental learning," *Neural Networks*, pp. 307–324, 2024.

[4] K. Shaheen, M. A. Hanif, O. Hasan, and M. Shafique, "Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks," *Journal of Intelligent & Robotic Systems*, p. 9, 2022.

[5] S. Raghavan, J. He, and F. Zhu, "Online class-incremental learning for real-world food image classification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024.

[6] Z. Cai, O. Sener, and V. Koltun, "Online continual learning with natural distribution shifts: An empirical study with visual data," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[7] H. Koh, D. Kim, J. W. Ha, and J. Choi, "Online continual learning on class incremental blurry task configuration with anytime inference," in *International Conference on Learning Representations*, 2022.

[8] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," *Advances in neural information processing systems*, 2020.

[9] C. Le, X. Wei, B. Wang, L. Zhang, and Z. Chen, "Learning continually from low-shot data stream," *arXiv preprint arXiv:1908.10223*, 2019.

[10] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Online continual learning with maximally interfered retrieval," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 849–11 860.

[11] E. Fini, S. Lathuiliere, E. Sangineto, M. Nabi, and E. Ricci, "Online continual learning under extreme memory constraints," in *European Conference on Computer Vision*. Springer, 2020, pp. 720–735.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[13] M. Wang, N. Michel, L. Xiao, and T. Yamasaki, "Improving plasticity in online continual learning via collaborative learning," in *CVPR*, 2024.

[14] F. Sarfraz, E. Arani, and B. Zonooz, "Error sensitivity modulation based experience replay: Mitigating abrupt representation drift in continual learning," in *ICLR*, 2023.

[15] E. Arani, F. Sarfraz, and B. Zonooz, "Learning fast, learning slow: A general continual learning method based on complementary learning system," in *International Conference on Learning Representations*, 2022.

[16] P. Singh, V. K. Verma, P. Mazumder, L. C, and P. Rai, "Calibrating cnns for lifelong learning," *Advances in Neural Information Processing Systems*, 2020.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.

[18] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[19] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[20] I. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[21] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[22] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *ICLR*, 2018.

[23] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," California Institute of Technology, Tech. Rep., 2011.

[24] A. M. Das, G. Bhatt, M. M. Bhalerao, V. R. Gao, R. Yang, and J. Bilmes, "Continual active learning," 2022.

[25] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, 2017.

[26] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, 2017.