

Enhancing Pretrained Model-based Continual Representation Learning via Guided Random Projection

Ruilin Li^{1,2*}, Heming Zou^{4*}, Xiufeng Yan⁶, Zheming Liang⁵, Jie Yang^{1,2},
Chenliang Li^{1✉}, and Xue Yang^{3✉}

¹ School of Cyber Science and Engineering, Wuhan University

² Shanghai Innovation Institute, ³ Shanghai Jiao Tong University

⁴ Tsinghua University, ⁵ USTC, ⁶ CUMT

ruilin-li@whu.edu.cn, yangxue-2019-sjtu@sjtu.edu.cn

Abstract. Recent paradigms in Random Projection Layer (RPL)-based continual representation learning have demonstrated superior performance when building upon a pre-trained model (PTM). These methods insert a randomly initialized RPL after a PTM to enhance feature representation in the initial stage. Subsequently, a linear classification head is used for analytic updates in the continual learning stage. However, under severe domain gaps between pre-trained representations and target domains, a randomly initialized RPL exhibits limited expressivity under large domain shifts. While largely scaling up the RPL dimension can improve expressivity, it also induces an ill-conditioned feature matrix, thereby destabilizing the recursive analytic updates of the linear head. To this end, we propose the Stochastic Continual Learner with MemoryGuard Supervisory Mechanism (SCL-MGSM). Unlike random initialization, MGSM constructs the projection layer via a principled, data-guided mechanism that progressively selects target-aligned random bases to adapt the PTM representation to downstream tasks. This facilitates the construction of a compact yet expressive RPL while improving the numerical stability of analytic updates. Extensive experiments on multiple exemplar-free Class Incremental Learning (CIL) benchmarks demonstrate that SCL-MGSM achieves superior performance compared to state-of-the-art methods. **Project Page:** <https://rlinl.github.io/SCL/>

Keywords: Pre-trained Models · Exemplar-free Continual Learning

1 Introduction

Although pre-trained models (PTMs) constitute strong representational backbones, their continual adaptation to downstream tasks is severely impeded by catastrophic forgetting [6, 11, 15, 36]. This challenge is particularly pronounced in exemplar-free CIL, which operates without task IDs or historical samples [40].

* Equal contribution, ✉ Corresponding author.

Recently, Random Projection Layer (RPL)-based methods [12, 13, 16, 19, 25, 42–45] have emerged as a promising paradigm, particularly for exemplar-free settings. Typically, the PTM is adapted on an initial task and then frozen. For each new task, data are projected through the frozen PTM and RPL, and only the linear head is updated via recursive least squares and its variants, which are algebraically equivalent to joint ridge regression on all observed data [7].

The efficacy of this paradigm is theoretically grounded in the geometry of high-dimensional spaces. As formalized by [24], scaling up the RPL dimension expands the available null space, thereby providing the necessary geometric degrees of freedom to identify the intersection of task-specific solution spaces and prevent forgetting. Driven by this “wider-is-better” theoretical insight, existing methods often resort to inflating the RPL to extremely high dimensions (e.g., >10k random bases) to ensure sufficient separability and mitigate the domain gap [25].

However, a critical gap exists between this theoretical ideal and its practical realization. Under a severe domain gap between the pre-trained representations and target domains, unguided random bases are unlikely to cover task-relevant regions in the feature space, causing the RPL to lack the expressivity required for downstream tasks. While aggressively scaling up the RPL dimension theoretically mitigates this issue by improving linear separability and expanding the null space to accommodate continual learning, practically it yields a highly ill-conditioned random-feature matrix [24]. This ill-conditioning forces the ridge-regression solver to rely on additional mechanisms to maintain numerical stability [16, 21]. Stabilizing analytic updates for a classification head with extremely large dimensions incurs substantial computational overhead. We posit that such stability and expressivity should arise from the intrinsic quality of the RPL. Currently, a principled way to configure the RPL that maintains high expressivity in a low

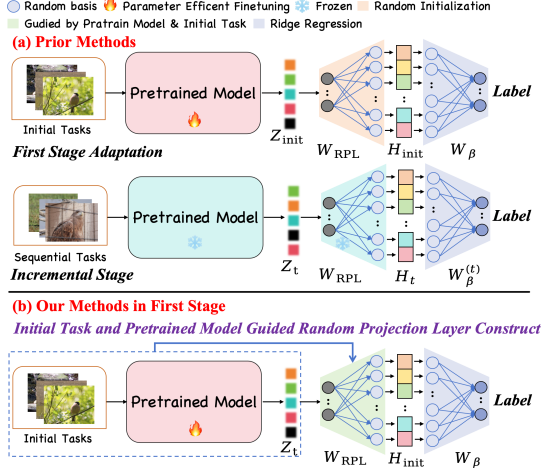


Fig. 1: Overview of the prior RPL-based CIL paradigm and comparison of initial-stage RPL construction. (a) Prior Methods: After first stage adaptation, frozen PTM extracts features Z_{init} , which are projected through a randomly initialized RPL (W_{RPL}) to obtain high-dimensional features H_{init} , followed by computing classifier weights W_{β} . During incremental learning, new features Z_t pass through the same frozen W_{RPL} to H_t , and only W_{β} is updated to $W_{\beta}^{(t)}$ via recursive ridge regression. **(b) Our Method:** We leverage the initial task and PTM to inform MGSM-guided RPL construction. During incremental learning, W_{β} is updated recursively as in (a).

dimension while preserving the numerical stability required by analytic classifiers is still lacking.

To address this dilemma, we draw inspiration from first-session adaptation (FSA) [21], where the PTM is fine-tuned on the initial task via Parameter-Efficient Fine-Tuning (PEFT) [3, 10, 14] and then frozen for subsequent tasks to narrow the domain gap. FSA substantially improves performance and has been widely adopted in both RPL-based [16, 19, 21] and prototype-based methods [38, 39]. Motivated by this insight, we go beyond adapting only the PTM and use the initial task to also guide the construction of the RPL. **We therefore propose the Stochastic Continual Learner with MemoryGuard Supervisory Mechanism (SCL-MGSM).** In the initial stage, MGSM constructs the RPL via a principled, data-guided mechanism: candidate random bases are sampled from an adaptively updated distribution and progressively selected by a target-aligned residual criterion. This produces a compact yet expressive RPL whose dimension is adaptively determined rather than fixed a priori. The convergence of this construction process is theoretically guaranteed. **In continual learning stages**, this RPL remains frozen. The compact RPL, with less collinear bases, yields better-conditioned feature matrices, improving the numerical stability required by recursive ridge-regression updates. An overview is provided in Figure 1. The core contributions of this work are as follows:

1. We propose to go beyond adapting only the PTM and leverage the initial task to also guide the construction of the RPL. This enables an expressive RPL, without resorting to extremely high-dimensional projections.
2. We introduce the MemoryGuard Supervisory Mechanism (MGSM), a principled, data-guided mechanism that employs a target-aligned residual criterion to progressively select informative and non-redundant random bases. With convergence guarantees (Theorem 1), MGSM constructs a compact, well-conditioned random feature space that supports stable recursive ridge-regression updates without external compensations.
3. Extensive experiments on multiple exemplar-free CIL benchmarks demonstrate that SCL-MGSM achieves superior performance and efficiency.

2 Related Works

Continual representation learning with Pre-trained Models. Adapting PTMs to learn new classes sequentially without forgetting previous ones is a significant challenge. Recent works primarily follow several strategies: (1) replay-based methods, which store historical exemplars and interleave them with current-task training to mitigate forgetting [17, 34]; (2) PEFT-based methods, which freeze the majority of PTM parameters and introduce lightweight trainable modules for task-specific adaptation [2, 26–28, 30, 35]; and (3) prototype-based methods, which incrementally maintain class prototypes or subspace statistics for efficient classifier updates [38, 39].

Random Projection Layer based Analytic Continual Learning. Recent RPL-based analytic continual learning has demonstrated strong performance when

building upon PTMs. ACIL [42] and G-ACIL [41] insert a randomly initialized RPL between the PTM and the classifier to enhance feature representation, and recursively update the classifier via analytic learning. RanPAC [16] adopts the same RPL strategy and further introduces first-session adaptation (FSA), where the PTM is fine-tuned on the initial task and then frozen for subsequent tasks, to adapt the PTM representations to downstream tasks. This significantly improves performance and is also effective for prototype-based methods. LoRanPAC [25] further scales up the RPL dimension in the first stage to enhance expressivity, and incorporates an additional mechanism to stabilize recursive updates for subsequent tasks, further improving performance. AnaCP [19] introduces a contrastive projection layer to adapt features for each task, also yielding notable improvements. Our work follows the RPL-based paradigm. Beyond fine-tuning the PTM on the initial task, we use initial-task data to guide RPL construction. This improves RPL expressivity without resorting to extremely high-dimensional projections, yielding better performance while preserving computational efficiency.

3 Revisiting RPL-based Analytic Continual Learning

3.1 Exemplar-free Class-Incremental Learning Setting

In the CIL setting, the model receives a sequence of datasets $\mathcal{D}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_t}$ for tasks $t = 1, 2, \dots, T$, where $\mathbf{x}_i \in \mathbb{R}^{c \times w \times h}$ denotes an input sample and $y_i \in \mathcal{Y}_t$ is its label. Class sets are disjoint across tasks, i.e., $\mathcal{Y}_t \cap \mathcal{Y}_{\hat{t}} = \emptyset$ for $t \neq \hat{t}$, and task identifiers are unavailable during both training and inference. In the exemplar-free setting, no data from previous stages are stored, so only the current-task dataset $\mathcal{D}_t$ is accessible at stage t .

3.2 The Supervisory Mechanism of SCNs

We briefly review the Stochastic Configuration Supervisory Mechanism (SCSM) of [29], a representative of classical constructive neural networks [5, 22, 23, 31], from which our proposed MGSM departs. Consider a single-hidden-layer network with $L - 1$ hidden units, $f_{L-1} = \mathbf{H}_{L-1} \mathbf{W}_{\beta_{L-1}}$, where $\mathbf{H}_{L-1} = [\mathbf{h}_1, \dots, \mathbf{h}_{L-1}] \in \mathbb{R}^{N \times (L-1)}$ is the hidden output matrix with each random basis $\mathbf{h}_i = g(\mathbf{X} \mathbf{w}_i + b_i) \in \mathbb{R}^N$, and $\mathbf{W}_{\beta_{L-1}} \in \mathbb{R}^{(L-1) \times m}$ are the output weights. Here $\mathbf{X} \in \mathbb{R}^{N \times d}$ is the input data, $g(\cdot)$ is an activation function, and $\mathbf{w}_i, b_i$ are randomly sampled input weight and bias. For multi-output residuals $\mathbf{e}_{L-1} = [\mathbf{e}_{L-1,1}, \dots, \mathbf{e}_{L-1,m}]$, a new random basis $\mathbf{h}_L$ is accepted only if each output component satisfies:

$$\langle \mathbf{e}_{L-1,q}, \mathbf{h}_L \rangle^2 \geq \|\mathbf{h}_L\|^2 \delta_{L,q}, \quad q = 1, \dots, m, \quad (1)$$

where $\delta_{L,q} = (1 - r - \mu_L) \|\mathbf{e}_{L-1,q}\|^2$, with contraction rate $r \in (0, 1)$ and vanishing tolerance $0 \leq \mu_L \leq 1 - r$, $\mu_L \rightarrow 0$. Under mild regularity conditions, this greedy line-projection criterion guarantees $\lim_{L \rightarrow \infty} \|\mathbf{e}_L\| = 0$ [29]. In our CIL setting, this criterion often yields compact random features at initialization, but the task-specific selection can bias bases toward current-task residuals and harm cross-stage robustness. To address this limitation, we introduce a novel supervisory mechanism in Section 4.

3.3 Prior RPL-based Continual Representation Learning Methods Framework

Recent RPL-based methods [16, 42] adopt a three-stage pipeline for exemplar-free CIL.

(1) Feature Extraction: Given the first-stage dataset $\mathcal{D}_1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_1}$, denoted as $\mathcal{D}_{\text{init}}$ (with $N := N_1$), input samples are fed into the PTM (frozen after first-session adaptation) to obtain feature representations:

$$\mathbf{Z}_{\text{init}} = \phi(\mathbf{X}_{\text{init}}; \Theta) \in \mathbb{R}^{N \times d}, \quad (2)$$

where $\mathbf{X}_{\text{init}} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top$, $\phi(\cdot; \Theta)$ denotes the frozen PTM mapping, and d is the embedding dimension.

(2) RPL Initialization: A Random Projection Layer is constructed by randomly sampling a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times L}$ and a bias vector $\mathbf{b} \in \mathbb{R}^L$ from a predefined distribution, yielding the random feature matrix $\mathbf{H}_{\text{init}} = g(\mathbf{Z}_{\text{init}}\mathbf{W} + \mathbf{1}_N\mathbf{b}^\top) \in \mathbb{R}^{N \times L}$. Driven by the “wider-is-better” theoretical insight [4, 24], recent methods often inflate $L \gg N$ to ensure sufficient linear separability and expand the null space for accommodating continual learning.

(3) Classifier Analytic Update: Given initial-task random features $\mathbf{H}_{\text{init}}$ and label matrix $\mathbf{Y}_{\text{init}} = [\mathbf{y}_1, \dots, \mathbf{y}_N]^\top \in \{0, 1\}^{N \times C_{\text{init}}}$, where $\mathbf{y}_i$ is the one-hot encoding of y_i and $C_{\text{init}} = |\mathcal{Y}_1|$ is the number of initial-task classes, the classifier is first obtained by ridge-regularized least squares:

$$\mathbf{W}_\beta^{(1)} = \arg \min_{\mathbf{W}_\beta} \|\mathbf{H}_{\text{init}}\mathbf{W}_\beta - \mathbf{Y}_{\text{init}}\|_F^2 + \lambda \|\mathbf{W}_\beta\|_F^2. \quad (3)$$

The corresponding sufficient statistic is

$$\mathbf{P}_{\text{init}} = \mathbf{H}_{\text{init}}^\top \mathbf{H}_{\text{init}} + \lambda \mathbf{I}. \quad (4)$$

For subsequent tasks $t = 2, 3, \dots$, recursive least squares (RLS) updates [7] are:

$$\mathbf{P}_t = \mathbf{P}_{t-1} + \mathbf{H}_t^\top \mathbf{H}_t, \quad (5)$$

$$\mathbf{W}_\beta^{(t)} = \mathbf{W}_\beta^{(t-1)} + \mathbf{P}_t^{-1} \mathbf{H}_t^\top (\mathbf{Y}_t - \mathbf{H}_t \mathbf{W}_\beta^{(t-1)}), \quad (6)$$

which is algebraically equivalent to joint ridge regression on all observed data, requiring only current-task data and the sufficient statistics $(\mathbf{W}_\beta^{(t-1)}, \mathbf{P}_{t-1})$. Method-specific variants may modify the update form but follow the same recursive update principle. In practice, this recursion is numerically reliable when $\mathbf{P}_t$ remains well-conditioned. If the RPL dimension is very large, unguided random bases can become highly collinear, which may increase the condition number of $\mathbf{H}_k^\top \mathbf{H}_k$ and degrade the conditioning of $\mathbf{P}_t$ when regularization is insufficient relative to the spectral spread. Under finite-precision arithmetic, the inverse in Eq. (6) may then amplify numerical errors and destabilize incremental learning. **This suggests a practical trade-off in RPL-based continual learning: unguidedly increasing the number of random bases can improve representational expressivity, but it can also reduce the numerical robustness of recursive updates.**

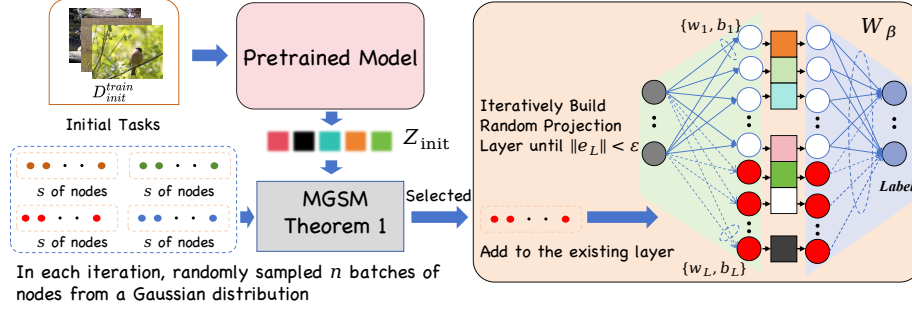


Fig. 2: Overview of MGSM-guided RPL construction in SCL-MGSM. Data from any stage can serve as the initialization set to build the RPL from scratch. Random hidden units are progressively sampled, evaluated by MGSM, and appended to the RPL only if they satisfy the supervisory criterion. The construction terminates once the residual converges below a predefined threshold ε . See Appendix B for details.

4 Method

We propose SCL-MGSM, whose core is the MemoryGuard Supervisory Mechanism (MGSM), a principled data-guided framework for progressively constructing an expressive yet well-conditioned RPL. Motivated by first-session adaptation (FSA) [21], MGSM leverages initial-task data and the pretrained model to guide RPL construction, iteratively selecting target-aligned, non-redundant random bases rather than sampling them all at once. The overall framework is illustrated in Fig. 2. In Section 4.1, we detail the algorithmic workflow of MGSM step by step (see Algorithm 1 for the complete procedure). In Section 4.2, we explain how MGSM yields an expressive RPL whose well-conditioned feature matrix ensures numerically stable recursive analytic updates.

4.1 SCL-MGSM Construction

Random Sampling of Hidden Units. Let $\mathbf{Z}_t$ denote the stage- t backbone features obtained via Eq. (2), with $\mathbf{Z}_{\text{init}}$ as the initialization-stage instance. Each candidate hidden unit is defined as

$$\mathbf{h}_i = g(\mathbf{Z}_{\text{init}}\mathbf{w}_i + b_i) \in \mathbb{R}^{N \times 1}, \quad (7)$$

where $\mathbf{w}_i$ and b_i are the input weight and bias of the i -th hidden unit, both randomly sampled from $\mathcal{N}(0, \xi^2)$ with scaling factor ξ . When a candidate is accepted by MGSM, its activation vector $\mathbf{h}_i$ is appended as a column of the RPL matrix.

Incremental Construction of the RPL. SCL-MGSM incrementally constructs the RPL via an iterative forward configuration. After $(L - s)$ hidden units have been accepted, their activations on $\mathbf{Z}_{\text{init}}$ form the current RPL output

$$\mathbf{H}_{L-s} = [\mathbf{h}_1, \dots, \mathbf{h}_{L-s}] \in \mathbb{R}^{N \times (L-s)}. \quad (8)$$

The corresponding classifier is

$$f_{L-s}(\mathbf{Z}_{\text{init}}) = \mathbf{H}_{L-s} \mathbf{W}_{\beta_{L-s}}, \quad (9)$$

where $\mathbf{W}_{\beta_{L-s}} \in \mathbb{R}^{(L-s) \times C}$ are the readout weights mapping the RPL output to the C class scores.

To decide whether further expansion is required, we define the current multi-output residual (with label matrix $\mathbf{Y} \in \mathbb{R}^{N \times C}$)

$$\mathbf{E}_{L-s} = \mathbf{Y} - \mathbf{H}_{L-s} \mathbf{W}_{\beta_{L-s}}, \quad (10)$$

and monitor its Frobenius norm $\|\mathbf{E}_{L-s}\|_F$. If $\|\mathbf{E}_{L-s}\|_F \leq \varepsilon$ for a predefined tolerance $\varepsilon > 0$, the RPL construction completes.

Otherwise, we continue to expand the RPL: at each iteration, we draw $B_{\max}$ candidate blocks. For each $j \in \{1, \dots, B_{\max}\}$, we independently sample s hidden units with parameters drawn from $\mathcal{N}(0, \xi^2)$ and compute their activations on $\mathbf{Z}_{\text{init}}$ to obtain a candidate matrix $\mathbf{H}_s^{(j)} \in \mathbb{R}^{N \times s}$. Using the augmented hidden matrix $[\mathbf{H}_{L-s}, \mathbf{H}_s^{(j)}]$, we evaluate the j -th candidate block by the MGSM acceptance criterion (formalized in Theorem 1).

Among all candidates that satisfy the criterion, we select the one achieving the largest improvement and accept it, so that the output matrix is updated to

$$\mathbf{H}_L = [\mathbf{H}_{L-s}, \mathbf{H}_s^{(j^*)}]. \quad (11)$$

MGSM Acceptance Criterion. We now formalize the block-wise selection rule used above. Let $\mathbf{H}_s \in \mathbb{R}^{N \times s}$ be the output matrix of a batch of s new hidden units, and define the augmented hidden matrix $\mathbf{H}_L = [\mathbf{H}_{L-s}, \mathbf{H}_s]$. Set

$$\mathbf{S} = \left(\mathbf{H}_s^\top \mathbf{H}_s + \lambda \mathbf{I}_s \right) - \mathbf{H}_s^\top \mathbf{H}_{L-s} \left(\mathbf{H}_{L-s}^\top \mathbf{H}_{L-s} + \lambda \mathbf{I}_{L-s} \right)^{-1} \mathbf{H}_{L-s}^\top \mathbf{H}_s. \quad (12)$$

For theoretical conciseness, we next state the criterion for a single-output target column. Let $\mathbf{e}_{L-s} = \mathbf{y} - \mathbf{H}_{L-s} \mathbf{W}_{\beta_{L-s}}$, and define

$$\mathbf{v} = \mathbf{H}_s^\top \mathbf{e}_{L-s}. \quad (13)$$

Theorem 1. Let $\mathbf{y} \in \mathbb{R}^N$ be the target vector, and let $\mathbf{H}_{L-s} \in \mathbb{R}^{N \times (L-s)}$ be the output matrix of the current network f_{L-s} . Suppose $\mathbf{W}_{\beta_{L-s}} = [\beta_1, \dots, \beta_{L-s}]^\top$ is the output weights, and define the current residual:

$$\mathbf{e}_{L-s} = \mathbf{y} - \mathbf{H}_{L-s} \mathbf{W}_{\beta_{L-s}}. \quad (14)$$

If the batch of new hidden units with output $\mathbf{H}_s$ satisfy the following inequalities:

$$2\mathbf{v}^\top \mathbf{S}^{-1} \mathbf{v} - \mathbf{v}^\top \left(\mathbf{S}^{-1} \mathbf{H}_s^\top \mathbf{H}_s \mathbf{S}^{-1} \right) \mathbf{v} \geq (1-r) \|\mathbf{e}_{L-s}\|^2, \quad (15)$$

where $0 < r < 1$ and the output weights $\mathbf{W}_{\beta_L^*} = [\beta_1^*, \dots, \beta_L^*]^\top$ are the ridge-regression weights that minimize



Fig. 3: Gaussian Initialization.

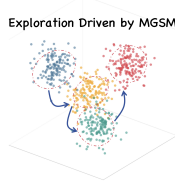


Fig. 4: Visualization of MGSM Exploration Strategy.

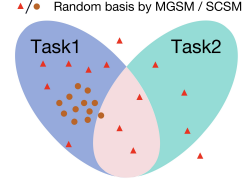


Fig. 5: Illustration of Random-Basis and Task-Relevant Regions.

$$\mathbf{W}_{\beta_L}^* = \arg \min_{\mathbf{W} \in \mathbb{R}^L} \|\mathbf{y} - \mathbf{H}_L \mathbf{W}\|_2^2 + \lambda \|\mathbf{W}\|_2^2. \quad (16)$$

then, we have $\lim_{L \rightarrow \infty} \|\mathbf{e}_L\| = 0$, where $\mathbf{e}_L = \mathbf{y} - \mathbf{H}_L \mathbf{W}_{\beta_L}^*$.

Proof. See Appendix A.

Remark (multi-output extension). For classification with $\mathbf{Y} \in \mathbb{R}^{N \times C}$, Theorem 1 is applied independently to each output column, which is equivalent to the Frobenius-norm formulation used in our construction.

If no candidate block is accepted under the current scaling ξ , we switch to the next value $\xi \in \mathcal{X}_\xi = \{\xi_{\min}, \xi_{\min} + \Delta\xi, \dots, \xi_{\max}\}$, resample $B_{\max}$ candidate blocks from $\mathcal{N}(0, \xi^2)$, and re-evaluate them by the same criterion. Empirically, exploring this discrete scaling set reliably yields admissible blocks, broadens the effective function space, and improves robustness to diverse task distributions, while keeping the PTM frozen and relying only on forward computations.

Initial SCL-MGSM Classifier. Let L^* denote the final number of hidden units in the constructed RPL. The SCL-MGSM classifier at the initial stage is

$$f_{L^*}(\mathbf{Z}_{\text{init}}) = \mathbf{H}_{L^*} \mathbf{W}_\beta = \sum_{i=1}^{L^*} \mathbf{h}_i \beta_i^\top \in \mathbb{R}^{N \times C}, \quad (17)$$

where $\mathbf{W}_\beta$ is obtained by ridge-regularized least squares. This procedure yields a task-relevant RPL characterized by the output matrix $\mathbf{H}_{L^*}$.

Incremental Update of Output Weights. The recursive update rule follows the generic RPL-based CIL formulation in Section 3, i.e., Eqs. (5)–(6). In our method, $\mathbf{H}_{\text{init}} = \mathbf{H}_{L^*}$, and thus Eq. (4) initializes the sufficient statistic for subsequent exemplar-free updates.

4.2 Underlying Rationale

We now explain how MGSM simultaneously enhances RPL expressivity and preserves the numerical stability required by recursive analytic updates.

MGSM-guided RPL construction enhances expressivity. First-session adaptation (FSA) adapts PTM representations using initial-task data [16, 21, 39], which presumably provides a useful inductive bias for subsequent tasks that share statistical similarities. Our work goes one step further by using the initial-task

data to guide RPL construction. SCL-MGSM incrementally configures the RPL via a block-wise, data-driven procedure: candidate random bases are sampled from an adaptively updated parameter distribution and accepted only if the target-aligned residual criterion is met. Otherwise, the sampling distribution is adjusted and the search is redirected to explore a different scale (Fig. 4). This multi-scale exploration steers the search toward parameter regions better suited for downstream continual learning tasks, producing a more functionally diverse set of random bases that enhances the expressivity of the RPL (Fig. 5). Theorem 1 guarantees the convergence of this construction process. In comparison, fixed-distribution sampling concentrates random bases on a narrow shell in parameter space (Fig. 3), so conventional random initialization can only improve expressivity by inflating the RPL dimension.

MGSM enables numerically stable analytic updates. MGSM-guided RPL achieves high expressivity without resorting to extremely high-dimensional projections, thereby avoiding the ill-conditioning that compromises analytic updates. Furthermore, compared to the supervisory mechanism of SCSM [29], which also aims to achieve expressivity in low dimensions, MGSM adopts a relaxed, target-aligned acceptance criterion instead of SCSM’s greedy residual-aligned criterion (Fig. 5). Combined with ridge regularization and block-wise updates, this relaxed criterion admits basis combinations that are not individually optimal but jointly effective, producing less collinear bases and better-conditioned Gram matrices for stable incremental updates. We elaborate on the relationship between MGSM and SCSM in Appendix B.

Algorithm 1 SCL-MGSM

Input: Initial Task $\mathcal{D}_1$, Incremental Tasks $\{\mathcal{D}_t\}_{t=2}^T$, predefined error ε , max batches B_{max} and number of nodes s .

- 1: *(Optional) First-session adaptation (FSA) of the PTM.*
- 2: **# Modeling RPL During Initial Training**
- 3: Extract features from frozen PTMs by Eq. (2)
- 4: **while** The predefined residual error is not satisfied **do**
- 5: Recruit randomly generated nodes based on MGSM by Eq. (15)
- 6: Update output weights by Eq. (16)
- 7: **end while**
- 8: Compute irreversible intermediate matrix $\mathbf{P}_{init}$ by Eq. (4)
- 9: **# Update Output Weights During Incremental Learning.**
- 10: **for** $t = 2, 3, \dots, T$ **do**
- 11: Extract features from frozen PTMs by Eq. (2)
- 12: Update Output Weights by Eq. (6)
- 13: Update irreversible intermediate matrix by Eq. (5)
- 14: **end for**

Output: SCL-MGSM

5 Experiments

5.1 Experiment Setting

Datasets: We conduct exemplar-free CIL experiments on four datasets, including ImageNet-R [8], ImageNet-A [9], ObjectNet [1], and Omnibenchmark [37].

ImageNet-R, ImageNet-A, and ObjectNet each contain 200 classes, while Omnibenchmark contains 300 classes. For brevity, we refer to them as IN-R, IN-A, ObjNet, and OmniB, respectively.

Compared Methods: We compare our proposed SCL-MGSM with four categories of CIL methods. (1) RPL-based methods: AnaCP [19], LoRanPAC [25], G-ACIL [41], RanPAC [16], KLDA [18]. (2) Prototype-based methods: EASE [39], SimpleCIL [38]. (3) Prompt-tuning-based methods: CODA-Prompt [27], APT [2]. (4) LoRA-based methods: SD-LoRA [33].

Evaluation Protocol. We report two metrics for all methods: average accuracy $A_{\text{avg}} = \frac{1}{t+1} \sum_{i=1}^t A_i$, where A_i is the test accuracy on $D_{1:i}^{\text{test}}$ after the i -th incremental session, and final accuracy A_{last} , which is the test accuracy on all learned tasks after the last incremental session.

Implementation Details. Unless stated otherwise, all methods use the frozen ViT-B/16-IN21K [32] as PTM. For first-session adaptation and ablation studies, we use AdaptFormer [3], SSF [14], and VPT [10] as PEFT methods. All results in Table 1 are averaged over 3 random seeds with standard error, where the initial-stage categories are non-overlapping across seeds. For SCL-MGSM, we set the residual tolerance $\varepsilon=0.01$ and the adaptive scaling range $\xi \in [0.0008, 0.004]$ with $\Delta\xi=0.0001$, shared across all four benchmarks; the final hidden size L^* is adaptively determined per dataset (Appendix C.2). Other hyperparameter settings can be found in Appendix C.1.

5.2 Main Results

As shown in Table 1, RPL-based methods consistently outperform prompt-tuning, LoRA-based, and prototype-based methods on most benchmarks, benefiting from random feature-space expansion. Among all compared methods, SCL-MGSM achieves the best A_{last} and A_{avg} across all eight evaluation settings, demonstrating consistent superiority over existing approaches. Specifically, on ImageNet-R with 20 sequential tasks, SCL-MGSM surpasses the second-best method by +2.05% in A_{last} and +1.85% in A_{avg} . On ObjectNet with 20 sequential tasks, the margins reach +1.77% and +2.49%, respectively. On ImageNet-A, SCL-MGSM consistently outperforms all competitors under both 40-task and 20-task protocols. Moreover, SCL-MGSM surpasses the joint linear probe on all datasets, and on ObjectNet the gap to joint fine-tuning, which serves as the upper bound, narrows to approximately 1%. Notably, the initial-stage categories are non-overlapping across the three seeds, yet SCL-MGSM yields consistent gains in all cases, confirming that MGSM-guided RPL construction generalizes across different initialization tasks. These gains are primarily attributed to MGSM, a data-guided mechanism in the initial stage that progressively selects target-aligned random bases with universal convergence guarantees (Theorem 1), constructing a compact yet expressive RPL that enhances the PTM’s representations, even without directly training the PTM on each task. The resulting well-conditioned feature space further benefits the stability of recursive analytic updates during incremental stages. The final hidden sizes of SCL-MGSM are reported in Appendix C.

Table 1: Comparison of CIL Performance Across Methods and Benchmarks. **Bold** and underline mark the best and second-best results. A_{last} and A_{avg} denote final and average accuracy (%). B- m Inc- n (m : initial classes, n : incremental classes; $m=0$: equal split). Joint linear probe trains only the classifier head on all tasks, while Joint fine-tuning updates the entire PTM on all tasks.

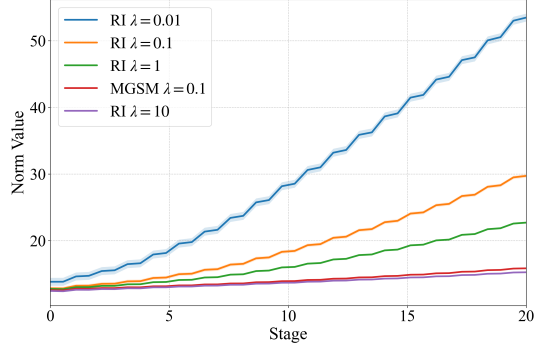
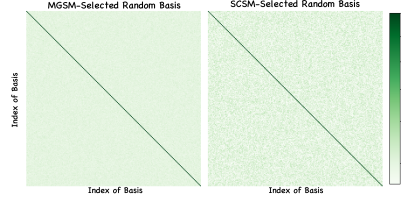
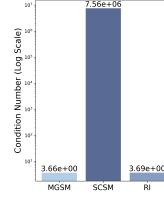
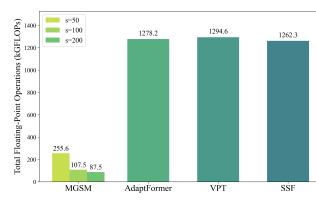
Method	ImageNet-R (B-0)				ImageNet-A (B-0)			
Joint linear probe	67.23 $\pm$ 0.05				50.98 $\pm$ 0.65			
Joint fine-tuning	83.67 $\pm$ 0.14				60.99 $\pm$ 0.32			
	Inc-5		Inc-10		Inc-5		Inc-10	
	A_{last}	A_{avg}	A_{last}	A_{avg}	A_{last}	A_{avg}	A_{last}	A_{avg}
CODA-Prompt	57.52 $\pm$ 0.68	65.71 $\pm$ 0.00	68.03 $\pm$ 0.42	74.57 $\pm$ 0.74	28.81 $\pm$ 0.60	41.45 $\pm$ 0.47	39.21 $\pm$ 0.83	50.17 $\pm$ 0.97
SD-LoRA	66.29 $\pm$ 1.10	73.67 $\pm$ 1.19	72.25 $\pm$ 0.72	78.14 $\pm$ 0.80	32.92 $\pm$ 1.07	46.00 $\pm$ 1.32	45.95 $\pm$ 0.53	58.41 $\pm$ 1.01
SimpleCIL	54.55 $\pm$ 0.00	61.63 $\pm$ 0.79	54.55 $\pm$ 0.00	61.22 $\pm$ 0.60	48.85 $\pm$ 0.00	60.49 $\pm$ 0.91	48.85 $\pm$ 0.00	59.96 $\pm$ 0.88
EASE	70.65 $\pm$ 0.52	76.69 $\pm$ 0.24	74.13 $\pm$ 0.57	80.38 $\pm$ 0.60	43.27 $\pm$ 2.55	55.80 $\pm$ 0.46	45.97 $\pm$ 1.93	58.09 $\pm$ 2.42
APT	70.55 $\pm$ 0.40	76.46 $\pm$ 0.71	75.07 $\pm$ 0.32	79.97 $\pm$ 0.57	37.19 $\pm$ 2.11	45.72 $\pm$ 3.80	51.57 $\pm$ 0.99	61.72 $\pm$ 0.44
KLDA	64.82 $\pm$ 0.30	71.02 $\pm$ 0.66	64.82 $\pm$ 0.30	70.69 $\pm$ 0.50	51.64 $\pm$ 0.16	58.23 $\pm$ 0.76	51.64 $\pm$ 0.16	57.69 $\pm$ 0.84
RanPAC	68.72 $\pm$ 0.40	74.84 $\pm$ 0.35	71.25 $\pm$ 0.45	76.58 $\pm$ 0.47	53.50 $\pm$ 1.58	60.56 $\pm$ 1.20	52.67 $\pm$ 0.39	61.59 $\pm$ 0.76
G-ACIL	67.05 $\pm$ 0.30	73.95 $\pm$ 0.44	67.05 $\pm$ 0.30	73.64 $\pm$ 0.29	45.05 $\pm$ 0.46	57.78 $\pm$ 0.29	45.05 $\pm$ 0.46	57.26 $\pm$ 0.37
LoRanPAC	70.78 $\pm$ 0.12	76.58 $\pm$ 0.20	71.81 $\pm$ 0.09	77.12 $\pm$ 0.06	54.84 $\pm$ 0.25	64.57 $\pm$ 0.38	54.62 $\pm$ 0.31	64.16 $\pm$ 0.55
AnaCP	71.34 $\pm$ 0.46	76.53 $\pm$ 0.80	73.60 $\pm$ 1.15	78.85 $\pm$ 0.32	53.08 $\pm$ 0.64	62.80 $\pm$ 0.55	55.22 $\pm$ 0.39	63.99 $\pm$ 0.61
SCL-MGSM	72.64$\pm$0.24	77.70$\pm$0.14	77.12$\pm$0.21	82.23$\pm$0.18	55.79$\pm$0.23	64.93$\pm$0.31	56.05$\pm$0.32	65.29$\pm$0.47
Method	ObjectNet (B-0)				OmniBenchmark (B-0)			
Joint linear probe	55.90 $\pm$ 0.28				78.56 $\pm$ 0.20			
Joint fine-tuning	67.53 $\pm$ 0.13				82.27 $\pm$ 0.26			
	Inc-5		Inc-10		Inc-5		Inc-10	
	A_{last}	A_{avg}	A_{last}	A_{avg}	A_{last}	A_{avg}	A_{last}	A_{avg}
CODA-Prompt	45.17 $\pm$ 0.49	56.44 $\pm$ 2.05	53.14 $\pm$ 0.07	63.68 $\pm$ 1.88	58.31 $\pm$ 0.80	69.56 $\pm$ 0.33	64.17 $\pm$ 0.58	74.06 $\pm$ 0.48
SD-LoRA	50.02 $\pm$ 1.24	61.95 $\pm$ 0.60	55.72 $\pm$ 0.98	66.74 $\pm$ 1.81	55.90 $\pm$ 1.55	69.22 $\pm$ 1.10	64.04 $\pm$ 0.15	74.25 $\pm$ 0.92
SimpleCIL	53.58 $\pm$ 0.00	63.15 $\pm$ 2.06	53.58 $\pm$ 0.00	62.70 $\pm$ 1.97	73.15 $\pm$ 0.00	81.10 $\pm$ 0.58	73.15 $\pm$ 0.00	80.85 $\pm$ 0.58
EASE	54.34 $\pm$ 0.71	65.30 $\pm$ 2.67	57.59 $\pm$ 0.33	68.34 $\pm$ 2.41	73.02 $\pm$ 0.20	81.09 $\pm$ 0.80	73.27 $\pm$ 0.09	81.04 $\pm$ 0.59
APT	56.05 $\pm$ 0.51	66.41 $\pm$ 1.02	60.54 $\pm$ 0.22	70.69 $\pm$ 1.60	62.99 $\pm$ 0.20	73.17 $\pm$ 0.23	66.12 $\pm$ 0.39	74.97 $\pm$ 0.04
KLDA	57.92 $\pm$ 0.04	67.23 $\pm$ 1.89	57.92 $\pm$ 0.04	66.76 $\pm$ 1.83	75.27 $\pm$ 0.04	83.48 $\pm$ 0.43	75.27 $\pm$ 0.04	83.23 $\pm$ 0.42
RanPAC	59.03 $\pm$ 0.41	69.02 $\pm$ 1.76	61.90 $\pm$ 0.41	71.27 $\pm$ 1.84	77.37 $\pm$ 0.23	85.33 $\pm$ 0.59	77.34 $\pm$ 0.17	85.00 $\pm$ 0.57
G-ACIL	57.60 $\pm$ 0.26	68.64 $\pm$ 2.39	57.60 $\pm$ 0.26	68.19 $\pm$ 2.29	76.41 $\pm$ 0.19	84.94 $\pm$ 0.75	76.41 $\pm$ 0.17	84.70 $\pm$ 0.73
LoRanPAC	61.89 $\pm$ 0.20	71.35 $\pm$ 1.93	63.57 $\pm$ 0.30	72.48 $\pm$ 1.82	76.68 $\pm$ 0.05	85.04 $\pm$ 0.80	76.70 $\pm$ 0.33	84.96 $\pm$ 0.69
AnaCP	62.66 $\pm$ 0.62	72.14 $\pm$ 0.15	64.68 $\pm$ 0.17	72.81 $\pm$ 0.94	78.64 $\pm$ 0.26	86.27 $\pm$ 1.08	78.81 $\pm$ 0.24	86.45 $\pm$ 0.71
SCL-MGSM	63.87$\pm$0.25	73.40$\pm$0.92	66.45$\pm$0.45	75.30$\pm$2.03	79.83$\pm$0.33	86.79$\pm$0.60	79.90$\pm$0.35	86.92$\pm$0.73

5.3 Mechanism Analysis

Expressivity under different RPL construction strategies. To assess whether MGSM improves RPL expressivity, we compare three construction strategies: MGSM, SCSM (the supervisory mechanism in SCNs [29]), and random initialization (RI). For a fair comparison, all methods yield an RPL with a final hidden size of 10,000 without FSA, and the output weights during incremental stages are computed via recursive ridge regression. As shown in Table 2, MGSM consistently outperforms both alternatives across all four benchmarks, achieving gains of approximately 5%–10% in A_{Last} over SCSM. On datasets with severe domain gaps (*e.g.* ImageNet-A), RI degrades substantially, confirming that an RPL composed of limited unguided random bases lacks the expressivity required for out-of-distribution data. SCSM, while data-guided, employs an overly strict greedy criterion that over-specializes to the initial task, often underperforming even RI on frozen PTM features. In contrast, MGSM’s relaxed, target-aligned criterion selects diverse, non-redundant bases that enrich RPL expressivity while keeping the projections adapted to downstream tasks. We further visualize the

Table 2: Performance Comparison of RPL Construction Strategies Without FSA.

Dataset	RPL	$A_{\text{Last}}(\%)$	$A_{\text{Avg}}(\%)$
<i>IN-R</i>	MGSM	70.09$\pm$0.13	75.90$\pm$0.08
	SCSM	64.78 $\pm$ 0.09	70.51 $\pm$ 0.44
	RI	63.00 $\pm$ 0.22	72.05 $\pm$ 0.39
<i>IN-A</i>	MGSM	53.63$\pm$0.74	62.45$\pm$0.59
	SCSM	44.85 $\pm$ 1.47	54.40 $\pm$ 1.88
	RI	47.31 $\pm$ 0.43	58.21 $\pm$ 0.37
<i>ObjNet</i>	MGSM	59.61$\pm$0.30	69.12$\pm$2.31
	SCSM	55.46 $\pm$ 0.33	64.98 $\pm$ 2.43
	RI	55.91 $\pm$ 0.14	66.08 $\pm$ 1.94
<i>OmniB</i>	MGSM	77.99$\pm$0.17	85.67$\pm$0.81
	SCSM	71.43 $\pm$ 0.11	79.34 $\pm$ 0.93
	RI	74.91 $\pm$ 0.04	82.54 $\pm$ 0.68

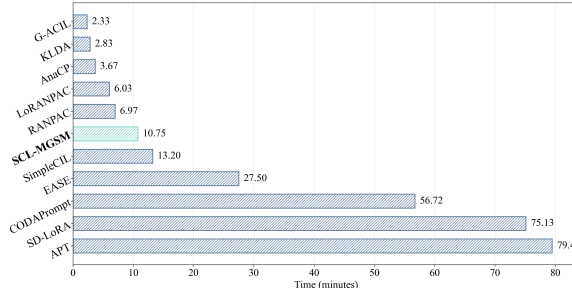
**Fig. 6: Comparison of P_t Norm Curves.****Fig. 7: Comparison of basis cosine similarity across different strategies.****Fig. 8: Comparison of Condition Numbers.****Fig. 9: Comparison of Computational Cost.**

basis quality on ImageNet-A in Figure 7: MGSM produces an almost diagonal cosine-similarity matrix, indicating near-orthogonal basis vectors that enrich the function space, whereas SCSM exhibits pronounced off-diagonal values reflecting high redundancy among bases. This further confirms that MGSM constructs a more expressive RPL, which contributes to stronger CIL performance.

Stability of MGSM-based continual learning. As discussed in Section 3, stable recursive updates require a well-conditioned random feature matrix. On ImageNet-A (Figure 8), we estimate the condition number of the random feature matrix on a randomly sampled subset of samples. Under MGSM-guided RPL, it is comparable to RI, whereas under SCSM-guided RPL it is markedly larger, indicating a substantially more ill-conditioned matrix and weaker numerical stability for recursive updates. We also track $\|P_t\|_F$ over incremental stages (Figure 6): for RI, $\|P_t\|_F$ grows steadily, and suppressing this growth requires an excessively large λ (e.g. $\lambda = 10$) that causes severe underfitting. MGSM with a moderate $\lambda = 0.1$ maintains a controlled $\|P_t\|_F$ comparable to heavily regularized RI. These results confirm that MGSM’s criterion contributes to both expressivity and stability.

Table 3: Performance Comparison of s and B_{max} on ImageNet-R (B-0 Inc-5).

METHOD	s	B_{max}	$A_{Avg}(\%)$	TIME (s)
MGSM	10	5	75.29 ± 0.15	92.37 ± 1.34
		10	75.31 ± 0.21	192.21 ± 2.64
		20	75.90 ± 0.18	262.56 ± 1.45
	100	5	74.82 ± 0.17	63.32 ± 0.81
		10	74.86 ± 0.21	70.37 ± 1.59
		20	75.17 ± 0.29	82.45 ± 2.33
SCSM	1	5	69.84 ± 0.34	268.61 ± 2.86
		10	69.89 ± 0.47	276.31 ± 2.79
		20	70.51 ± 0.29	332.18 ± 2.12

**Fig. 10: Comparison of Training Time Across Representative CIL Methods.**

5.4 Hyperparameter Sensitivity

Analysis of adaptive scaling of ξ . Figure 11 compares two fixed ξ values (0.08 and 0.008) with the adaptive (ADP) strategy on four datasets. ξ is a scaling factor that controls the sampling range for candidate random bases during RPL construction in SCL-MGSM. Both fixed ξ settings yield weaker results than the adaptive strategy (ADP), which outperforms them across all four benchmarks. A fixed ξ keeps the sampling range unchanged throughout the initial construction phase. As the RPL grows and its modeling capacity increases, candidates drawn from the same fixed range become less likely to pass the MGSM acceptance criterion, limiting further effective expansion of the RPL. The adaptive strategy adjusts ξ according to the current RPL state, so that later construction iterations can sample from a range that remains compatible with the MGSM criterion. This allows SCL-MGSM to explore a broader set of function spaces during construction, producing an RPL that better adapts to diverse downstream tasks.

Analysis of sensitivity to s and B_{max} . In Table 3, B_{max} is the maximum number of Gaussian-sampled batches, and s is the number of hidden units per batch. Larger s implies more concurrently integrated units, accelerating convergence. The same table indicates that a tenfold variation in s negligibly affects overall performance, allowing s to be increased for substantially faster configuration. Concurrently, a larger B_{max} enables broader function space exploration, improving optimal hidden unit selection probability. Furthermore, SCSM is significantly slower than MGSM, primarily because: first, its unit-by-unit configuration requires more forward computations, and second, its Moore-Penrose pseudoinverse calculation is computationally expensive.

5.5 Generalization across Pretrained Models

To verify that MGSM-guided RPL construction is not tied to a specific PTM, we further evaluate SCL-MGSM with DINO-v2 [20], a strong self-supervised Transformer backbone. As shown in Table 4, SCL-MGSM achieves the best A_{last} and A_{avg} on ImageNet-R, OmniBenchmark, and ObjectNet under both protocols, and attains the highest A_{last} on ImageNet-A under both protocols. Specifically, on

Table 4: Comparison of CIL performance across methods with DINO-v2 as the PTM. B- m Inc- n denotes m initial classes and n incremental classes ($m=0$: equal split). A_{last} and A_{avg} denote final and average accuracy (%), respectively. **Bold** and underline mark the best and second-best incremental results under each protocol.

Method	ImageNet-R (B-0)				ImageNet-A (B-0)			
Joint Linear Probe	82.47 $\pm$ 0.07				69.67 $\pm$ 0.37			
	Inc-5		Inc-10		Inc-5		Inc-10	
	A_{last}	A_{avg}	A_{last}	A_{avg}	A_{last}	A_{avg}	A_{last}	A_{avg}
SimpleCIL	75.35 $\pm$ 0.00	80.82 $\pm$ 0.94	75.35 $\pm$ 0.00	80.52 $\pm$ 0.91	68.93 $\pm$ 0.00	77.58 $\pm$ 1.13	68.93 $\pm$ 0.00	77.13 $\pm$ 1.07
KLDA	75.78 $\pm$ 0.17	80.76 $\pm$ 1.15	75.78 $\pm$ 0.16	80.79 $\pm$ 0.78	66.91 $\pm$ 0.63	70.13 $\pm$ 1.76	66.91 $\pm$ 0.63	70.15 $\pm$ 1.49
LoRanPAC	84.06 $\pm$ 0.10	88.34 $\pm$ 0.55	84.97 $\pm$ 0.21	89.11 $\pm$ 0.55	66.69 $\pm$ 0.65	73.60 $\pm$ 1.08	69.23 $\pm$ 0.44	75.28 $\pm$ 1.12
AnaCP	84.41 $\pm$ 0.40	88.93 $\pm$ 0.48	85.94 $\pm$ 0.33	89.82 $\pm$ 0.42	71.05 $\pm$ 0.72	79.45$\pm$0.72	71.98 $\pm$ 0.68	78.28 $\pm$ 1.18
RanPAC	83.67 $\pm$ 0.25	87.82 $\pm$ 0.49	84.56 $\pm$ 0.41	88.46 $\pm$ 0.69	48.56 $\pm$ 2.56	57.55 $\pm$ 4.60	64.65 $\pm$ 2.23	70.98 $\pm$ 1.28
G-ACIL	81.53 $\pm$ 0.19	86.33 $\pm$ 0.71	81.53 $\pm$ 0.19	86.12 $\pm$ 0.67	67.61 $\pm$ 0.58	75.91 $\pm$ 0.88	67.63 $\pm$ 0.60	75.45 $\pm$ 0.80
SCL-MGSM	84.61$\pm$0.79	89.26$\pm$0.17	86.40$\pm$0.16	90.18$\pm$0.45	71.60$\pm$0.42	78.66 $\pm$ 0.90	72.42$\pm$0.52	79.30$\pm$1.01
Method	ObjectNet (B-0)				OmniBenchmark (B-0)			
Joint Linear Probe	65.17 $\pm$ 0.28				82.41 $\pm$ 0.05			
	Inc-5		Inc-10		Inc-5		Inc-10	
	A_{last}	A_{avg}	A_{last}	A_{avg}	A_{last}	A_{avg}	A_{last}	A_{avg}
SimpleCIL	56.13 $\pm$ 0.00	66.33 $\pm$ 2.55	56.13 $\pm$ 0.00	65.84 $\pm$ 2.43	73.48 $\pm$ 0.00	81.02 $\pm$ 1.06	73.48 $\pm$ 0.00	80.79 $\pm$ 1.04
KLDA	59.99 $\pm$ 0.19	69.66 $\pm$ 0.68	59.99 $\pm$ 0.18	67.91 $\pm$ 1.67	77.50 $\pm$ 0.14	86.29 $\pm$ 1.40	77.51 $\pm$ 0.12	85.03 $\pm$ 1.23
LoRanPAC	67.82 $\pm$ 0.26	77.04 $\pm$ 1.91	69.72 $\pm$ 0.15	78.25 $\pm$ 1.82	78.62 $\pm$ 0.24	86.64 $\pm$ 0.58	79.22 $\pm$ 0.41	86.83 $\pm$ 0.56
AnaCP	71.61 $\pm$ 0.22	78.74 $\pm$ 2.10	71.76 $\pm$ 0.63	79.56 $\pm$ 1.49	81.32 $\pm$ 0.27	88.49 $\pm$ 0.55	81.46 $\pm$ 0.41	88.42 $\pm$ 0.56
RanPAC	66.95 $\pm$ 0.44	75.78 $\pm$ 2.34	68.78 $\pm$ 0.46	77.10 $\pm$ 1.71	79.80 $\pm$ 0.17	87.21 $\pm$ 0.70	79.96 $\pm$ 0.27	87.08 $\pm$ 0.52
G-ACIL	63.58 $\pm$ 0.19	73.20 $\pm$ 2.22	63.58 $\pm$ 0.18	72.77 $\pm$ 2.15	80.14 $\pm$ 0.19	87.58 $\pm$ 0.44	80.14 $\pm$ 0.19	87.39 $\pm$ 0.45
SCL-MGSM	72.04$\pm$1.32	80.34$\pm$1.96	72.50$\pm$0.59	80.23$\pm$1.76	83.40$\pm$0.10	89.38$\pm$0.71	83.43$\pm$0.20	89.34$\pm$0.81

OmniBenchmark Inc-5, SCL-MGSM surpasses the second-best method (AnaCP) by +2.08% in A_{last} . Notably, on OmniBenchmark SCL-MGSM surpasses the JOINT LINEAR PROBE oracle (83.40% vs. 82.41%) despite operating in a strictly incremental setting. Relative to Table 1, DINO-v2 consistently improves all compared methods, yet SCL-MGSM retains its advantage over the other RPL-based baselines, indicating that MGSM-guided subspace modeling continues to benefit from stronger feature representations rather than saturating as the backbone improves. We further verify on a convolutional neural network-based pretrained model in Appendix C, where SCL-MGSM continues to deliver strong performance, further demonstrating that MGSM generalizes across diverse PTM backbones.

5.6 Additional Analysis

Computational efficiency. Figure 10 reports the end-to-end training time on ImageNet-R (Inc-10). RPL-based methods achieve the best CIL performance while being generally faster than other paradigms. Among them, SCL-MGSM introduces only a minor overhead over other RPL-based baselines, in exchange for notably improved performance. The additional cost of MGSM stems solely from the increased number of recursive inverse updates during RPL construction (Figure 9), which remains far below PEFT. In the extreme case, one can pair a frozen PTM directly with MGSM, bypassing FSA entirely, to eliminate the adaptation cost while still benefiting from data-guided RPL construction. Additional analysis is provided in the appendix. Moreover, during incremental learning, compared with directly enlarging the hidden dimension to improve expressivity, SCL-MGSM

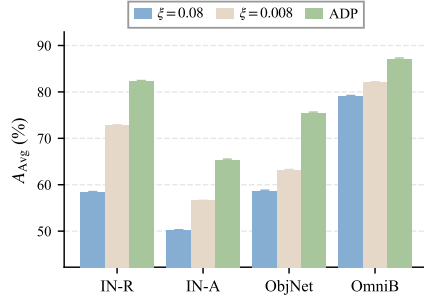


Fig. 11: Parameter Analysis of MGSM on Three Datasets. ADP refers to adaptive ξ adjustment.

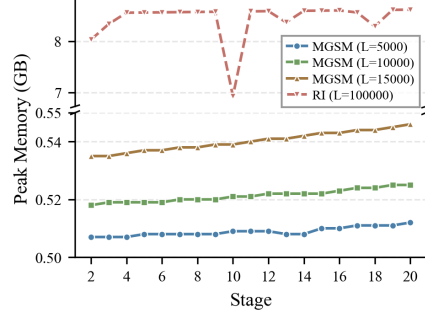


Fig. 12: Comparison of Peak GPU Memory.

is significantly more memory-efficient. For example, constructing a 100,000-dimensional RPL via RI and employing the stabilized recursive solver of [25] for incremental updates demands tens of times more peak GPU memory across CIL stages (Figure 12), indicating that improving performance by unguidedly enlarging the RPL quickly encounters resource bottlenecks.

More analysis. We present additional analysis in the appendix, including 1) the result on a convolutional neural network-based pretrained model, 2) the impact of first-session adaptation on RPL-based methods, 3) the comparison of average forgetting across methods, and 4) the RPL construction cost of MGSM against random initialization baselines.

6 Conclusion

In this work, we propose SCL-MGSM to enhance pretrained model-based continual representation learning via guided random projection. MGSM employs a target-aligned residual criterion to progressively select informative and non-redundant random bases, constructing a compact yet expressive RPL whose dimension is adaptively determined rather than fixed a priori. On this well-conditioned random feature space, the analytic classifier is updated via recursive ridge regression without external stabilization mechanisms. Extensive experiments on seven exemplar-free CIL benchmarks demonstrate that SCL-MGSM achieves superior performance and training efficiency. Moreover, SCL-MGSM is compatible with both first-session adaptation and diverse PTM backbones, demonstrating strong adaptability across training protocols and architectural families.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62506229), Natural Science Foundation of Shanghai under 25ZR1402268, and Shanghai QiYuan Innovation Foundation.

References

1. Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems* **32** (2019)
2. Chen, H., Wang, P., Zhou, Z., Zhang, X., Wu, Z., Jiang, Y.G.: Achieving more with less: Additive prompt tuning for rehearsal-free class-incremental learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 340–349 (2025)
3. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems* **35**, 16664–16678 (2022)
4. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers* (3), 326–334 (1965)
5. Fahlman, S., Lebiere, C.: The cascade-correlation learning architecture. *Advances in neural information processing systems* **2** (1989)
6. French, R.M.: Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences* **3**(4), 128–135 (1999)
7. Greville, T.N.: Some applications of the pseudoinverse of a matrix. *SIAM review* **2**(1), 15–22 (1960)
8. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 8340–8349 (2021)
9. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 15262–15271 (2021)
10. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: *European conference on computer vision*. pp. 709–727. Springer (2022)
11. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* **114**(13), 3521–3526 (2017)
12. Li, D., Wang, T., Chen, J., Dai, W., Zeng, Z.: Harnessing neural unit dynamics for effective and scalable class-incremental learning. *arXiv preprint arXiv:2406.02428* (2024)
13. Li, D., Zeng, Z.: Crnet: A fast continual learning framework with random theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(9), 10731–10744 (2023)
14. Lian, D., Zhou, D., Feng, J., Wang, X.: Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems* **35**, 109–123 (2022)
15. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: *Psychology of learning and motivation*, vol. 24, pp. 109–165. Elsevier (1989)
16. McDonnell, M.D., Gong, D., Parvaneh, A., Abbasnejad, E., Van den Hengel, A.: Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems* **36**, 12022–12053 (2023)

17. Meng, Z., Zhang, J., Yang, C., Zhan, Z., Zhao, P., Wang, Y.: Diffclass: Diffusion-based class incremental learning. In: European Conference on Computer Vision. pp. 142–159. Springer (2024)
18. Momeni, S., Mazumder, S., Liu, B.: Continual learning using a kernel-based method over foundation models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 19528–19536 (2025)
19. Momeni, S., Xiao, C., Liu, B.: Anacp: Toward upper-bound continual learning via analytic contrastive projection. *Advances in Neural Information Processing Systems* **38**, 72923–72944 (2026)
20. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
21. Panos, A., Kobe, Y., Reino, D.O., Aljundi, R., Turner, R.E.: First session adaptation: A strong replay-free baseline for class-incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18820–18830 (2023)
22. Pao, Y.H., Park, G.H., Sobajic, D.J.: Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* **6**(2), 163–180 (1994)
23. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. *Neural computation* **3**(2), 246–257 (1991)
24. Peng, L., Giampouras, P., Vidal, R.: The ideal continual learner: An agent that never forgets. In: International Conference on Machine Learning. pp. 27585–27610. PMLR (2023)
25. Peng, L., Litwin, J.E., Agterberg, J., Ribeiro, A., Vidal, R.: Loranpac: Low-rank random features and pre-trained models for bridging theory and practice in continual learning. In: 13th International Conference on Learning Representations, ICLR 2025. pp. 23483–23529. International Conference on Learning Representations, ICLR (2025)
26. Qiao, J., Zhang, Z., Tan, X., Chen, C., Qu, Y., Peng, Y., Xie, Y.: Prompt gradient projection for continual learning. In: International Conference on Learning Representations. vol. 2024, pp. 16294–16315 (2024)
27. Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11909–11919 (2023)
28. Sun, H.L., Zhou, D.W., Zhao, H., Gan, L., Zhan, D.C., Ye, H.J.: Mos: Model surgery for pre-trained model-based class-incremental learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 20699–20707 (2025)
29. Wang, D., Li, M.: Stochastic configuration networks: Fundamentals and algorithms. *IEEE transactions on cybernetics* **47**(10), 3466–3479 (2017)
30. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 139–149 (2022)
31. Wei, T., Wang, C., Rui, Y., Chen, C.W.: Network morphism. In: International conference on machine learning. pp. 564–572. PMLR (2016)
32. Wightman, R.: Pytorch image models. <https://github.com/rwightman/pytorch-image-models> (2019). <https://doi.org/10.5281/zenodo.4414861>
33. Wu, Y., Piao, H., Huang, L.K., Wang, R., Li, W., Pfister, H., Meng, D., Ma, K., Wei, Y.: Sd-lora: Scalable decoupled low-rank adaptation for class incremental learning. In: 13th International Conference on Learning Representations, ICLR 2025. pp. 10049–10065. International Conference on Learning Representations, ICLR (2025)

34. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3014–3023 (2021)
35. Yu, J., Zhuge, Y., Zhang, L., Hu, P., Wang, D., Lu, H., He, Y.: Boosting continual learning of vision-language models via mixture-of-experts adapters. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23219–23230 (2024)
36. Zhang, L., Qian, L., Li, R., Li, T., Zhang, W.: Leveraging multi-modal and historical knowledge graphs for continual robot navigation. *Visual Intelligence* **4**(1), 15 (2026)
37. Zhang, Y., Yin, Z., Shao, J., Liu, Z.: Benchmarking omni-vision representation through the lens of visual realms. In: European conference on computer vision. pp. 594–611. Springer (2022)
38. Zhou, D.W., Cai, Z.W., Ye, H.J., Zhan, D.C., Liu, Z.: Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision* **133**(3), 1012–1032 (2025)
39. Zhou, D.W., Sun, H.L., Ye, H.J., Zhan, D.C.: Expandable subspace ensemble for pre-trained model-based class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23554–23564 (2024)
40. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5871–5880 (2021)
41. Zhuang, H., Chen, Y., Fang, D., He, R., Tong, K., Wei, H., Zeng, Z., Chen, C.: GacI: Exemplar-free generalized analytic continual learning. *Advances in neural information processing systems* **37**, 83024–83047 (2024)
42. Zhuang, H., Weng, Z., Wei, H., Xie, R., Toh, K.A., Lin, Z.: Acil: Analytic class-incremental learning with absolute memorization and privacy protection. *Advances in Neural Information Processing Systems* **35**, 11602–11614 (2022)
43. Zou, H., Zang, Y., Ji, X.: Structural features of the fly olfactory circuit mitigate the stability-plasticity dilemma in continual learning. *arXiv preprint arXiv:2502.01427* (2025)
44. Zou, H., Zang, Y., Xu, W., Ji, X.: Fly-cl: A fly-inspired framework for enhancing efficient decorrelation and reduced training time in pre-trained model-based continual representation learning. *arXiv preprint arXiv:2510.16877* (2025)
45. Zou, H., Zang, Y., Xu, W., Zhu, Y., Ji, X.: Flylora: Boosting task decoupling and parameter efficiency via implicit rank-wise mixture-of-experts. *Advances in Neural Information Processing Systems* **38**, 10386–10419 (2026)