DUAL-STATE PREDICTIVE MOMENTUM FOR ONLINE CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Online continual learning (OCL) empowers models to learn from non-stationary streams in a single pass while retaining prior knowledge. In this context, exponential moving average (EMA)-based strategies have achieved impressive results and underpin many state-of-the-art approaches. However, their first-order temporal smoothing only tracks the average parameter level, leading to a low-fidelity approximation of the evolving low-loss center. To this end, we propose **D**ual-**S**tate **P**redictive **M**omentum (DSPM), which jointly estimates the level and trend states of the student trajectory through two momentum estimators and extrapolates them to construct a one-step predictive teacher. From the perspective of linear mode connectivity, we disclose the tracking bias of EMA and show that DSPM yields an asymptotically unbiased trajectory error and a tighter upper bound on the average empirical loss. We further introduce dual-granularity complementary distillation, which aligns teacher-student output distributions while preserving stable level-state representations, promoting classifier plasticity and representation stability. Extensive experiments on CIFAR-10, CIFAR-100, Tiny-ImageNet, and ImageNet-100 validate the effectiveness of DSPM. Specifically, DSPM improves model stability by 2.66%, 1.40%, 3.36%, and 5.79% over the latest state-of-the-art method on these benchmarks, respectively, with an average test accuracy gain of 1.08%. Our code is available at `https://anonymous.4open.science/r/DSPM`.

1 INTRODUCTION

In real-world scenarios, deep learning models are often exposed to continuously evolving data streams, rendering them highly vulnerable to catastrophic forgetting (Kirkpatrick et al., 2017) when adapting to new scenarios. Inspired by human-like learning, continual learning (CL) has been proposed as a powerful paradigm for mitigating catastrophic forgetting, holding particular significance for practical applications ranging from medical diagnosis (Luo et al., 2025), embodied AI (Feng et al., 2025) to autonomous driving (Liu et al., 2026). Generally, CL can be categorized into task-incremental, domain-incremental, and class-incremental scenarios according to task identity availability and label-space evolution (Van de Ven & Tolias, 2019). Currently, most CL methods follow an offline training protocol, where continual learners are trained on available data for multiple epochs. However, when deployed in realistic online scenarios, these methods may suffer substantial performance degradation (Wang et al., 2024b), since each incoming instance from the current task is accessed for training only once. To address this practical challenge in the CL realm, a growing body of work has investigated online continual learning (OCL) (Aljundi et al., 2019b; Gunasekara et al., 2023). In this work, we focus on the most challenging class-incremental OCL.

Various OCL methods have been proposed to enable models to learn continually from one-pass data streams while mitigating catastrophic forgetting. Among them, rehearsal-based methods (Chaudhry et al., 2019b; Aljundi et al., 2019a; Caccia et al., 2022; Lin et al., 2023; Wu et al., 2024) are among the earliest and most widely adopted approaches with remarkable success in OCL. Those methods aim to store selected exemplars from previous tasks in a memory buffer and replay them with incoming data to prevent forgetting. Beyond such input-level rehearsal, some methods rehearse intermediate representations (Hayes et al., 2020; Lin, 2024) or final prototypes (Buzzega et al., 2020; Wei et al., 2023) to improve the OCL performance of downstream tasks. Building on rehearsal, regularization techniques constrain important parameters through penalty terms to prevent substantial

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

parameter drift when learning new tasks, as in OCM (Guo et al., 2022), LPR (Yoo et al., 2024), and OCAR (Urettini & Carta, 2025). Additionally, to effectively improve model stability, some rehearsal studies (Buzzega et al., 2020; Yan et al., 2024; Wang et al., 2024b) incorporate knowledge distillation (KD) to enforce information consistency between historical and current models.

Complementary to rehearsal-based methods, many researchers have explored exponential moving average (EMA) for rehearsal-free OCL. EMA offers a direct yet effective methodology that can be seamlessly integrated with existing rehearsal-based methods. For example, TempEns (Soutif-Cormerais et al., 2023) developed a lightweight temporal ensemble by applying EMA to historical models, reducing the stability gap in rehearsal-based methods. To improve the weight distribution on past knowledge, SDP (Koh et al., 2023) used two EMA models to construct a hypo-exponential weight distribution. Since KD in OCL remains under-explored, MKD (Michel et al., 2024) constructed an evolving teacher whose parameters are updated as an EMA of the student parameters. Furthermore, GGSM (Qiu et al., 2026) leveraged 1-order and 2-order gradient information to generate a mask over an EMA-based teacher, which serves as a container of previous knowledge. In fact, the aforementioned EMA-based OCL methods lie in linear mode connectivity (Frankle et al., 2020; Marouf et al., 2024), which suggests that accuracy levels in the loss landscape are maintained along linear interpolation paths between two separate networks. By leveraging this property, EMA can strike an easy-to-implement stability-plasticity balance for continual learners (Lin et al., 2022). However, existing EMA-based OCL methods essentially behave as a 1-order linear low-pass filter over model checkpoints. They maintain a temporally smoothed estimate of historical knowledge with fixed exponential decay, but provide only a coarse approximation of the underlying optimal trajectory. Consequently, constructing a high-fidelity optimization trajectory remains an open problem for achieving better stability-plasticity balance in OCL.

To this end, we propose **D**ual-**S**tate **P**redictive **M**omentum (DSPM) for building high-fidelity optimization trajectories in OCL. First, we shed light on how EMA induces biased trajectories and derive its upper bound on the empirical loss. Second, DSPM focuses on predicting both the level state and trend state of past parameters using two coupled EMA models, as shown in Figure 1. This enables DSPM to construct a higher-fidelity trajectory with a tighter loss upper bound than EMA, which in turn yields better stability-plasticity balance. Third, we design a dual-granularity complementary distillation scheme that aligns output distributions between DSPM and the student model, while matching level-momentum representations with student representations. This scheme jointly preserves representation-level stability and classifier-level plasticity. In summary, the contributions of this work are as follows:

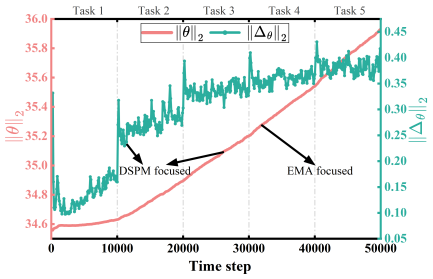


Figure 1: Evolution of $\|\theta\|_2$ and $\|\Delta_\theta\|_2$ on CIFAR-10 for the parameter dynamics focused by EMA and DSPM.

- A theoretical analysis of EMA-based OCL is provided through the lens of linear mode connectivity to disclose its first-order biased optimization trajectory, motivating the development of our proposed method for OCL.
- We propose a level and trend-based dual-state predictive momentum method to construct an adaptive high-fidelity parameter trajectory for OCL, resulting in a better balance of past and present knowledge than EMA.
- A dual-granularity complementary distillation scheme is designed to align the stability-oriented teacher and the plasticity-oriented student at both the representation and output distribution levels, thereby facilitating our low-loss linear connectivity.

2 Related Work

2.1 Offline Continual Learning

Typically, continual learning (CL) refers to offline CL, where the training data of the current task can be revisited over multiple epochs. It can be broadly categorized into four categories: regularization, rehearsal, parameter isolation, and representation-based methods (Wang et al., 2024a; 2025c).

Regularization-based methods (Kirkpatrick et al., 2017; Schwarz et al., 2018; Cui et al., 2023) introduce regularization terms to constrain variation of important model parameters, thereby protecting the previously gained knowledge. Such methods are typically proposed earlier than other types, owing to their ease of implementation and transparent theoretical foundations (Li & Zeng, 2023). Rehearsal-based methods (Chaudhry et al., 2019a; Zhang et al., 2024; Theofilou et al., 2025) mitigate catastrophic forgetting by maintaining a memory buffer of past samples and replaying them alongside incoming data. Parameter-isolation methods explicitly allocate disjoint parameter subspaces (Yang et al., 2023; Wang et al., 2025b) or architectural components (Wang et al., 2022; Guo et al., 2025) to different tasks, thereby reducing interference between old and new knowledge. Representation-based methods encompass a broad range of strategies, including meta-learning (Chi et al., 2022), self-supervised learning (Liu et al., 2025b), and pretraining (Wang et al., 2025a), which flexibly exploit the advantages of representation learning to enhance CL.

2.2 Online Continual Learning

Unlike offline CL, OCL represents a more practical setting, where each sample is observed for just one pass. This setting better accommodates privacy-preserving and real-time learning requirements. Rehearsal-based methods are widely used and achieve noticeable performance in OCL. For instance, ER (Rolnick et al., 2019) and DER++ (Buzzega et al., 2020) show that storing raw samples or output logits from previous tasks and replaying them during current-task learning can achieve favorable performance. Apart from input-output rehearsal, OnPro (Wei et al., 2023) extended ER with online prototype learning to improve feature discrimination, while ER-FSL (Lin, 2024) introduced feature subspace learning to decouple new sample learning from the rehearsed samples. Based on rehearsal, OCM (Guo et al., 2022) preserves past feature knowledge by maximizing mutual information. LPR (Yoo et al., 2024) stabilized OCL by imposing layerwise proximal regularization to preserve the hidden activations of rehearsed data. OCAR (Urettini & Carta, 2025) exploited Fisher information to regularize distributional changes on rehearsed data. Some rehearsal-based works further incorporate KD. MOSE (Yan et al., 2024) developed multi-level supervision and reverse self-distillation to address the overfitting-underfitting dilemma. CCL-DC (Wang et al., 2024b) combined collaborative continual learning with a distillation chain to enhance model plasticity.

Meanwhile, a growing body of work has explored exponential moving average (EMA) over model parameters as a rehearsal-free strategy for OCL. To address the stability gap in model evaluation, TempEns (Soutif-Cormerais et al., 2023) employs EMA to construct a lightweight temporal ensemble model at test time. Empirically, SDP (Koh et al., 2023) introduced two EMA models to design a hypo-exponential teacher for feature distillation. Aiming at effective KD in OCL, MKD (Michel et al., 2024) constructed a momentum teacher by updating its parameters as an EMA of the student's parameters. Instead of using a vanilla EMA teacher, GGSM (Qiu et al., 2026) exploited first- and second-order gradient information to generate parameter-wise masks for the EMA teacher. EMA offers a simple yet effective way to achieve plug-in OCL, making it easy to implement in real-world scenarios. However, existing EMA-based OCL methods have not yet fully recognized why EMA is effective in OCL, nor have they characterized the empirical loss along its optimization trajectory.

Discussion. In this paper, we argue that the optimization trajectory built by 1-order linear interpolation of EMA is biased in view of linear mode connectivity (Frankle et al., 2020; Marouf et al., 2024). By contrast, DSPM gains an unbiased and high-fidelity optimization trajectory by maintaining both level and trend momentum. Akin to SDP, MKD, and GGSM, our method also incorporates KD. Nevertheless, it differs by performing dual-granularity complementary distillation to jointly align output distributions and representations. In contrast to SDP, which uses two EMA models to construct a fixed hypo-exponential weighting trajectory, DSPM focuses on building a self-regulating trajectory from level and trend perspectives. MKD largely remains an empirical extension of vanilla EMA to KD. GGSM indeed utilizes gradient information to generate adaptive weights for a single EMA, without explicitly incorporating second-order momentum information about model parameters.

3 Preliminaries

In this section, we first introduce the concept of linear mode connectivity and then present the problem formulation of online continual learning, laying the groundwork for the subsequent analysis.

3.1 LINEAR MODE CONNECTIVITY

The optimization of deep learning models amounts to searching for low-loss regions in a high-dimensional and highly non-convex objective landscape. Given a model $\mathcal{F}(\boldsymbol{x};\boldsymbol{\theta}_0)$ initialized with parameters $\boldsymbol{\theta}_0$, we create two copies and train them up to iteration k, e.g., with different data augmentations or projection heads, yielding two optimized parameters $\hat{\boldsymbol{\theta}}^1_{0:k}$ and $\hat{\boldsymbol{\theta}}^2_{0:k}$. Under such conditions, many interesting phenomena have been observed. One notable example is linear model connectivity (Frankle et al., 2020; Wortsman et al., 2021), which reveals that there is a continuous low-loss linear path from $\hat{\boldsymbol{\theta}}^1_{0:k}$ to $\hat{\boldsymbol{\theta}}^2_{0:k}$ when $\boldsymbol{\theta}_0$ is not randomly initialized but is trained for a certain number of iterations. This condition can be readily satisfied, as two optima can be connected by a linear path once they share a common initialization $\boldsymbol{\theta}_0$. Currently, some studies (Lin et al., 2022; Wen et al., 2023) on offline CL have shown that a naive linear connection between any two task-specific optima may traverse a high-loss region.

3.2 PROBLEM FORMULATION

In the class-incremental OCL setting, we require a model $\mathcal{F}(\boldsymbol{x};\boldsymbol{\theta})$ to continuously update itself from a data stream, with each mini-batch containing new data sampled from a changing distribution. The stream consists of a sequence of tasks $t \in \{1,\ldots,\mathcal{T}\}$ with mutually disjoint label spaces, i.e. $\mathcal{C}_t \cap \mathcal{C}_{t'} = \emptyset$ for $t \neq t'$. The task t is associated with a task-specific data distribution $\mathcal{D}_t$ over input-label pairs, whose label space is $\mathcal{C}_t$. For task t, the model receives a mini-batch of data $\mathcal{B}_t = \{(\boldsymbol{x}^t_i, y^t_i)\}^{n_t}_{i=1}$, where $\mathcal{B}_t \sim \mathcal{D}_t$, $\boldsymbol{x}^t_i$ represents the input data, y^t_i is the corresponding label, and n_t is the number of samples. The model is expected to predict over the accumulated label space $\mathcal{C}_{1:c} = \cup^c_{j=1}\mathcal{C}_j$ under one-epoch training without task identity. The ideal optimization objective of OCL is to minimize the cumulative loss on all encountered tasks:

$$\boldsymbol{\theta}^*_t = \arg\min_{\boldsymbol{\theta}} \mathcal{J}_t(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{t} \mathcal{L}_i(\boldsymbol{\theta};\mathcal{B}_i) \tag{1}$$

The $\boldsymbol{\theta}^*_t$ represents the optimal parameter position that simultaneously accommodates both the past and current tasks, i.e., the low-loss center associated with all tasks observed so far. However, solving for $\boldsymbol{\theta}^*_t$ using all data $\mathcal{B}_{1:t}$ is impractical under the one-pass constraint of OCL. As such, this paper aims to construct a high-fidelity parameter optimization trajectory to approximate the ideal $\boldsymbol{\theta}^*_t$.

4 METHODOLOGY

4.1 MOTIVATIONS

In this subsection, we elaborate on the motivation of our study. First, we introduce a unified definition of trajectory accuracy to infer tracking error. Subsequently, the trajectory error of EMA is rigorously derived. Furthermore, we discuss the implications of this biased trajectory, motivating the development of our method.

Definition of Trajectory Accuracy. When the current task t emerges, the overall empirical risk across all tasks is marked as $\mathcal{J}_t(\boldsymbol{\theta})$. We can perform a second-order Taylor expansion of $\mathcal{J}_t(\boldsymbol{\theta})$ around the dynamic optimal point $\boldsymbol{\theta}^*_t$:

$$\mathcal{J}_t(\boldsymbol{\theta}) \approx \mathcal{J}_t(\boldsymbol{\theta}^*_t) + (\boldsymbol{\theta}-\boldsymbol{\theta}^*_t)^T\nabla\mathcal{J}_t(\boldsymbol{\theta}^*_t) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^*_t)^T\mathbf{H}_t\,(\boldsymbol{\theta}-\boldsymbol{\theta}^*_t) \tag{2}$$

where the Hessian $\mathbf{H}_t = \nabla^2\mathcal{J}_t(\boldsymbol{\theta}^*_t)$. Since $\boldsymbol{\theta}^*_t$ is the optimum model parameter, $\nabla\mathcal{J}_t(\boldsymbol{\theta}^*_t) \approx \mathbf{0}$. We further have:

$$\mathcal{J}_t(\boldsymbol{\theta}) \approx \mathcal{J}_t(\boldsymbol{\theta}^*_t) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^*_t)^T\mathbf{H}_t\,(\boldsymbol{\theta}-\boldsymbol{\theta}^*_t)\,. \tag{3}$$

If the maximum eigenvalue of $\mathbf{H}_t$ is Λ, the upper bound of $\mathcal{J}_t(\boldsymbol{\theta})$ is formulated as follows:

$$\mathcal{J}_t(\boldsymbol{\theta}) \leqslant \mathcal{J}_t(\boldsymbol{\theta}^*_t) + \frac{\Lambda}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}^*_t\|^2. \tag{4}$$

Therefore, the empirical loss for all tasks can be upper-bounded by encouraging the model parameters $\boldsymbol{\theta}$ to approach the current low-loss center $\boldsymbol{\theta}^*_t$. Accordingly, the accuracy of an optimization

trajectory can be characterized by the error between the parameters along the current trajectory γ_t and the low-loss center $\boldsymbol{\theta}_t^*$, defined as:

$$\varepsilon_t = \int_0^1 \|\gamma_t(\eta) - \boldsymbol{\theta}_t^*\|^2 d\eta \tag{5}$$

where $\eta \in [0, 1]$ denotes the interpolated coefficient along the linear connectivity trajectory.

Trajectory Error of EMA. The update formula of the standard EMA-based teacher is written as follows:

$$\boldsymbol{E}_t = \rho \boldsymbol{E}_{t-1} + (1-\rho)\boldsymbol{\theta}_t \tag{6}$$

where $\rho \in (0, 1)$ is the smoothing coefficient, and the EMA-based teacher is parameterized by $\boldsymbol{E}_t$ for task t. Note that $\boldsymbol{E}_0 = \boldsymbol{\theta}_0$. Under linear model connectivity, its optimization trajectory is built by linearly interpolating as follows:

$$\gamma_t^{EMA}(\eta) = \eta \boldsymbol{E}_{t-1} + (1-\eta)\boldsymbol{\theta}_t \tag{7}$$

The trajectory error of EMA is defined as $\boldsymbol{e}_t = \boldsymbol{E}_t - \boldsymbol{\theta}_t^* = \rho(\boldsymbol{E}_{t-1} - \boldsymbol{\theta}_t^*)$. Based on the local linear drift hypothesis introduced in Appendix A.1, the EMA error propagation can be derived as follows:

$$\boldsymbol{e}_t = \rho(\boldsymbol{E}_{t-1} - \boldsymbol{\theta}_{t-1}^* - \boldsymbol{\Delta}_{\boldsymbol{\theta}}) = \rho \boldsymbol{e}_{t-1} - \rho \boldsymbol{\Delta}_{\boldsymbol{\theta}}. \tag{8}$$

Clearly, equation 8 is a first-order constant coefficient linear nonhomogeneous difference formula and satisfies the steady-state equilibrium condition in Appendix A.2. Then, we have $\boldsymbol{e}_t = \boldsymbol{e}_{t-1} = \boldsymbol{e}^*$. Accordingly, the trajectory error of EMA is formulated as follows:

$$\boldsymbol{e}_t = \boldsymbol{E}_t - \boldsymbol{\theta}_t^* = -\frac{\rho}{1-\rho}\boldsymbol{\Delta}_{\boldsymbol{\theta}}. \tag{9}$$

We further obtain $\boldsymbol{E}_{t-1} - \boldsymbol{\theta}_t^* = -\frac{1}{1-\rho}\boldsymbol{\Delta}_{\boldsymbol{\theta}}$. Afterwards, the overall tracking error induced by the EMA-based optimization trajectory is derived as follows:

$$\varepsilon_t^{EMA} = \int_0^1 \left\|\gamma_t^{EMA}(\eta) - \boldsymbol{\theta}_t^*\right\|^2 d\eta = \int_0^1 \left\| -\frac{\eta}{1-\rho}\boldsymbol{\Delta}_{\boldsymbol{\theta}} \right\|^2 d\eta = \frac{\|\boldsymbol{\Delta}_{\boldsymbol{\theta}}\|^2}{3(1-\rho)^2}. \tag{10}$$

Finally, from equation 4, the average loss $\bar{\mathcal{J}}_{EMA}$ over this EMA-based optimization trajectory for the overall empirical risk across all tasks is quantified as:

$$\bar{\mathcal{J}}_{EMA} = \int_0^1 \mathcal{J}_t\left(\gamma_t^{EMA}(\eta)\right) d\eta \leqslant \mathcal{J}_t(\boldsymbol{\theta}_t^*) + \frac{\Lambda}{2}\int_0^1 \left\|\gamma_t^{EMA}(\eta) - \boldsymbol{\theta}_t^*\right\|^2 d\eta = \mathcal{J}_t(\boldsymbol{\theta}_t^*) + \frac{\Lambda}{2}\varepsilon_t^{EMA} \tag{11}$$

Remark 1. From the perspective of tracking error, equation 10 substantiates how the EMA coefficient ρ governs the stability-plasticity trade-off of the EMA-based teacher. A larger ρ slows teacher updates and preserves longer-term historical information, thereby enhancing stability but increasing the deviation from the current low-loss center and hindering plasticity. More importantly, it reveals that the EMA-induced trajectory error satisfies $\varepsilon_t^{EMA} > \frac{1}{3}\|\boldsymbol{\Delta}_{\boldsymbol{\theta}}\|^2$ for any $\rho \in (0, 1)$, indicating that EMA inevitably incurs a non-negligible first-order tracking bias. Since ε_t^{EMA} determines the upper bound of the average loss $\bar{\mathcal{J}}_{EMA}$, a smaller trajectory error leads to a tighter loss bound and yields a more favorable stability-plasticity trade-off. This motivates us to devise a high-fidelity trajectory.

4.2 Dual-State Predictive Momentum

To construct a high-fidelity optimization trajectory, we propose the dual-state predictive momentum (DSPM) method for KD-based OCL. The overview of our DSPM framework when applied to a baseline OCL method with a memory buffer is illustrated in Figure 2. DSPM consists of two peer continual learners with the same architecture: a student model $\mathcal{S}$ and a teacher model Γ. The teacher parameters are updated from the student parameters through the proposed DSPM strategy. The model update, objective function, and final model inference are detailed in the following.

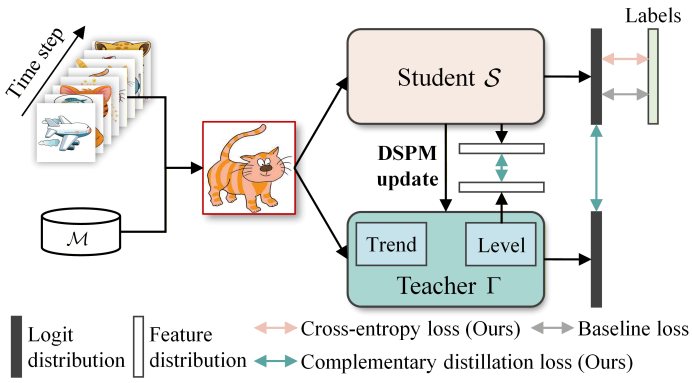


Figure 2: Overview of the DSPM framework when applied to a baseline with a memory buffer.

Model Update. In contrast to EMA for model update, which only tracks the exponentially averaged level state of historical student parameters, DSPM explicitly models both the level and trend states of the student trajectory through two coupled EMA estimators. It is formulated as follows:

$$\boldsymbol{L}_t = \alpha\boldsymbol{\theta}_t + (1-\alpha)(\boldsymbol{L}_{t-1} + \boldsymbol{B}_{t-1}) \tag{12}$$

$$\boldsymbol{B}_t = \beta(\boldsymbol{L}_t - \boldsymbol{L}_{t-1}) + (1-\beta)\boldsymbol{B}_{t-1} \tag{13}$$

$$\boldsymbol{T}_t = \boldsymbol{L}_t + \boldsymbol{B}_t \tag{14}$$

where $\alpha \in (0,1)$ and $\beta \in (0,1)$ are smoothing coefficients; $\boldsymbol{L}_t$ and $\boldsymbol{B}_t$ denote the parameter states corresponding to the level and trend components, respectively; $\boldsymbol{T}_t$ is the model parameter for the DSPM teacher Γ.

According to the theoretical validation of DSPM's unbiased trajectory error in Appendix A.3, we have obtained $\boldsymbol{T}_{t-1} = \boldsymbol{\theta}_t^*$. For the DSPM-based optimization trajectory

$$\gamma_t^{DSPM}(\eta) = \eta\boldsymbol{T}_{t-1} + (1-\eta)\boldsymbol{\theta}_t, \quad \eta \in [0,1], \tag{15}$$

the overall tracking error of DSPM is derived as follows:

$$\varepsilon_t^{DSPM} = \int_0^1 \left\|\gamma_t^{DSPM}(\eta) - \boldsymbol{\theta}_t^*\right\|^2 d\eta = 0. \tag{16}$$

Therefore, the average loss $\bar{\mathcal{J}}_{DSPM}$ along the DSPM-based optimization trajectory is given by:

$$\bar{\mathcal{J}}_{DSPM} = \int_0^1 \mathcal{J}_t\left(\gamma_t^{DSPM}(\eta)\right) d\eta \leqslant \mathcal{J}_t(\boldsymbol{\theta}_t^*). \tag{17}$$

Remark 2. Notably, DSPM degenerates into conventional EMA when the trend state is disabled, i.e., $\boldsymbol{B}_t \equiv \boldsymbol{0}$, indicating that DSPM provides a more general formulation. By extrapolating the trend state from past parameter dynamics, DSPM constructs a one-step predictive teacher that better anticipates the evolving low-loss region. From a path-wise perspective, this predictive mechanism reduces the distance between the linear connectivity path $\gamma_t^{DSPM}(\eta)$ and the dynamic low-loss center compared with the EMA path $\gamma_t^{EMA}(\eta)$. Consequently, DSPM yields a smaller trajectory-error term and a tighter empirical loss upper bound than EMA, as illustrated in equation 17, facilitating a superior plasticity–stability trade-off in OCL.

Objective Function. The objective function of the proposed DSPM consists of two components: the cross-entropy loss $\mathcal{L}_{CE}$ and the proposed dual-granularity complementary KD loss $\mathcal{L}_{CKD}$. For the current task t, the overall training objective is presented as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{CE}\left(\mathcal{S}(\hat{\boldsymbol{x}}^t), \boldsymbol{y}^t\right) + \mathcal{L}_{CKD}\left(\Gamma(\boldsymbol{x}^t), \mathcal{S}(\boldsymbol{x}^t)\right) \tag{18}$$

where Γ and $\mathcal{S}$ denote the teacher and student models parameterized by $\boldsymbol{T}$ and $\boldsymbol{\theta}$, respectively.

Specifically, $\mathcal{L}_{CKD}$ is defined as

$$\begin{aligned}\mathcal{L}_{CKD}\left(\Gamma(\boldsymbol{x}^t), \mathcal{S}(\boldsymbol{x}^t)\right) =& \frac{\mu_1}{2}\left[KL\left(\Gamma(\hat{\boldsymbol{x}}^t)/\tau, \mathcal{S}(\boldsymbol{x}^t)/\tau\right) + KL\left(\Gamma(\hat{\boldsymbol{x}}^t)/\tau, \mathcal{S}(\hat{\boldsymbol{x}}^t)/\tau\right)\right] \\ &+ \frac{\mu_2}{2}\left[\left\|\mathcal{F}_L(\hat{\boldsymbol{x}}^t) - \mathcal{F}_S(\boldsymbol{x}^t)\right\|^2 + \left\|\mathcal{F}_L(\hat{\boldsymbol{x}}^t) - \mathcal{F}_{\mathcal{S}}(\hat{\boldsymbol{x}}^t)\right\|^2\right]\end{aligned} \tag{19}$$

where KL is the Kullback–Leibler divergence, τ is the distillation temperature, and $\mathcal{F}_L$ and $\mathcal{F}_S$ are the feature extractors of the level state and the student model, respectively. The augmented input is generated by a data augmentation operation, i.e., $\hat{\boldsymbol{x}}^t = \mathrm{Aug}(\boldsymbol{x}^t)$, which enables dual-view distillation between the original and augmented views. Noteworthily, $\mathcal{L}_{CKD}$ anchors the level component in the feature space to stabilize the historical knowledge, while introducing the trend component in the logit space to provide momentum guidance for new-task adaptation. In this way, DSPM effectively balances stability and plasticity at both the representation and classification levels. We also provide PyTorch-like pseudo-code for the proposed DSPM in Algorithm 1 of Appendix A.4.

Model Inference. Usually, the student aims to achieve excellent performance on the current task. The teacher is expected to retain knowledge acquired from previous tasks, serving as a stability-oriented reference. Relying solely on either the teacher or student weights for inference may therefore be suboptimal. Consequently, following model averaging estimation (Michel et al., 2024), we average the teacher and student parameters for final online inference, yielding $\bar{\boldsymbol{\theta}}^* = \frac{\boldsymbol{T}+\boldsymbol{\theta}}{2}$.

5 Experiments

5.1 Experiment Setup

Datasets. We evaluate the effectiveness of our method on four image classification benchmark datasets: CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), Tiny-ImageNet (Le et al., 2015), and ImageNet-100 (Hou et al., 2019).

Baselines. To demonstrate the effectiveness and superiority of our method, we compare the proposed DSPM with eleven state-of-the-art OCL methods, including ER (Rolnick et al., 2019), DER++ (Buzzega et al., 2020), ER-ACE (Caccia et al., 2022), GSA (Guo et al., 2023), PCR (Lin et al., 2023), CCL-DC (Wang et al., 2024b), S6MOD (Liu et al., 2025a), SDP (Koh et al., 2023), TempEns (Soutif-Cormerais et al., 2023), MKD (Michel et al., 2024), and GGSM (Qiu et al., 2026). Among them, ER serves as a plain OCL baseline. SDP, TempEns, MKD, and GGSM are EMA-based OCL methods. For a fair comparison, methods that are originally rehearsal-free, including CCL-DC, S6MOD, EMA-based OCL methods, and our DSPM, are implemented on top of ER.

Metrics. We adopt average test accuracy (ACC) and average forgetting measure (AFM) (Liu et al., 2025a) as evaluation metrics, where ACC measures the overall CL performance and AFM reflects the degree of performance degradation or knowledge forgetting on previous tasks. All reported results of each metric are averaged over five repeated runs with random seeds 0, 1, 2, 3, and 4.

Implementation Details. For all methods, we adopt a full ResNet-18 (He et al., 2016) without pretraining as the feature extractor and use a linear layer as the classifier. Following (Qiu et al., 2026; Wang et al., 2024b; Michel et al., 2024), the streaming batch size $|\mathcal{B}|$ is set to 10, and the memory batch size is capped at 64. The temperature τ for KD is set to 4. Afterwards, we perform a hyperparameter search on CIFAR-100 with $\mathcal{M} = 2$k and use the selected hyperparameters across the other datasets. Our model is optimized using Adam with a learning rate of 0.0005. The teacher update coefficients α and β are set to 0.005 and 0.001, respectively, while the loss weights μ_1 and μ_2 are set to 6 and 0.05. For more details regarding the experiment implementation, please refer to Appendix B.1.

5.2 Results and Discussion

Unlike most OCL studies, which evaluate performance gains after integrating their method into different baselines, we directly compare DSPM with state-of-the-art OCL methods under a common ER framework. Experimental results for ACC and AFM are reported in Tables 1 and 2, respectively.

Average test accuracy. Table 1 shows the final ACC under different memory buffer sizes. DSPM achieves the best result in 8 out of 11 settings and ranks among the top two in 10 settings, demonstrating consistently strong overall performance. In particular, DSPM achieves the highest ACC on ImageNet-100 for all memory sizes, outperforming the second-best method by 1.34%, 1.13%, and 1.60%, respectively. It also obtains the best results on Tiny-ImageNet with $\mathcal{M}$=2k and 5k. On CIFAR-100, DSPM achieves the highest ACC for $\mathcal{M}$=1k and 2k and remains second-best for

Table 1: ACC (%, higher is better) on four benchmark datasets under different memory buffer sizes $\mathcal{M}$. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	CIFAR-10		CIFAR-100			Tiny-ImageNet			ImageNet-100		
Memory size $\mathcal{M}$	500	1000	1000	2000	5000	2000	5000	10000	2000	5000	10000
ER [NeurIPS'19]	56.68±1.89	62.32±4.13	24.47±0.72	31.89±1.45	39.41±1.81	10.82±0.79	19.16±1.42	24.71±2.52	19.06±0.90	33.30±1.74	36.72±1.09
DER++ [NeurIPS'20]	58.04±2.30	64.02±1.92	25.09±1.41	32.33±2.66	38.31±2.28	8.73±1.58	17.95±2.49	19.40±3.71	15.36±3.04	34.75±2.23	36.94±4.67
ER-ACE [ICLR'22]	53.26±3.04	59.94±2.40	28.36±1.99	34.21±1.53	39.39±1.31	13.56±1.00	20.84±0.43	25.92±1.07	27.16±0.57	38.37±1.20	39.14±0.35
GSA [CVPR'23]	60.34±1.97	66.54±2.28	27.72±1.57	35.08±1.37	41.41±1.65	12.44±1.17	19.59±1.30	25.34±1.43	26.29±0.59	41.03±0.99	42.18±0.93
PCR [CVPR'23]	60.61±2.23	61.66±1.39	30.68±0.81	38.63±1.01	45.27±0.78	12.47±3.56	20.41±2.84	23.85±4.21	19.89±6.24	31.35±3.01	36.99±4.70
SDP [ICLR'23]	58.85±1.38	65.93±2.15	28.48±1.18	35.21±0.73	41.19±1.11	15.32±0.47	23.22±0.31	26.97±1.10	22.95±0.13	32.12±0.23	35.64±0.25
TempEns [CoLLAs'23]	62.10±1.70	70.20±0.20	30.00±0.70	37.90±0.90	46.80±0.90	16.30±0.50	23.60±0.40	30.00±0.50	20.40±0.50	33.12±1.70	44.58±1.10
CCL-DC [CVPR'24]	66.43±2.48	74.10±1.71	33.43±1.06	44.45±1.04	**53.81±1.16**	16.56±1.63	29.39±1.23	37.73±0.85	30.12±1.09	43.11±1.49	48.64±2.40
MKD [ICLR'24]	**68.48±0.92**	74.33±0.68	38.50±0.50	45.20±0.20	52.10±0.50	23.95±0.65	32.22±0.88	**38.27±0.78**	30.67±0.46	39.70±1.07	44.92±0.98
S6MOD [CVPR'25]	57.88±3.30	65.80±2.16	26.50±2.23	34.55±1.66	39.61±3.16	10.94±1.47	19.67±1.36	25.62±1.73	21.51±0.96	31.22±1.23	36.97±2.63
GGSM [AAAI'26]	67.71±1.29	74.12±0.38	38.35±0.43	45.56±0.42	52.19±0.66	23.59±0.63	31.82±0.25	37.86±0.35	33.29±0.27	43.42±1.07	49.06±2.43
DSPM (Ours)	67.63±0.72	**74.51±0.48**	**38.68±0.33**	**45.59±0.50**	52.44±0.63	**24.83±0.59**	**32.30±0.45**	37.96±0.81	**34.63±0.48**	**44.55±0.66**	**50.66±0.90**

Table 2: AFM (%, lower is better) on four benchmark datasets under different memory buffer sizes $\mathcal{M}$. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	CIFAR-10		CIFAR-100			Tiny-ImageNet			ImageNet-100		
Memory Size $\mathcal{M}$	500	1000	1000	2000	5000	2000	5000	10000	2000	5000	10000
ER [NeurIPS'19]	33.16±3.50	20.94±6.79	32.65±1.78	22.20±2.26	13.29±1.98	58.38±1.69	46.87±1.60	40.77±2.45	41.94±1.41	23.38±2.10	20.29±1.93
DER++ [NeurIPS'20]	24.21±2.75	18.42±1.84	34.49±4.39	25.55±3.26	20.01±2.88	62.03±2.83	51.57±4.60	49.51±3.04	31.19±3.53	28.77±4.10	13.67±1.74
ER-ACE [ICLR'22]	**15.25±0.99**	11.73±2.71	15.33±0.63	11.13±1.57	6.26±1.50	20.66±0.29	18.93±0.58	16.53±0.52	24.20±1.00	16.79±0.68	11.76±0.91
GSA [CVPR'23]	21.10±2.17	16.09±0.76	30.24±0.90	19.97±1.66	9.02±0.90	29.03±0.65	25.05±0.68	22.48±0.90	47.99±1.02	30.70±1.65	18.04±0.87
PCR [CVPR'23]	27.88±4.97	26.53±14.52	34.34±1.03	25.88±0.49	25.88±0.49	38.23±6.45	28.43±4.88	26.40±8.60	35.58±4.85	25.76±4.00	13.78±1.34
SDP [ICLR'23]	26.62±4.40	16.09±2.00	20.71±1.50	14.74±1.58	10.42±1.55	50.34±0.84	23.53±0.63	39.67±0.92	23.91±1.83	17.77±1.49	12.75±1.13
TempEns [CoLLAs'23]	32.00±2.59	19.60±1.33	31.85±1.22	19.90±1.44	9.68±1.75	23.28±1.04	16.03±0.74	10.77±0.59	23.77±1.72	28.87±0.94	18.58±1.74
CCL-DC [CVPR'24]	30.22±3.75	19.85±2.55	43.28±1.67	29.35±1.50	16.88±1.99	69.56±1.54	53.13±0.85	42.63±0.80	29.41±1.63	28.48±1.50	18.64±0.81
MKD [ICLR'24]	21.69±2.80	11.08±1.05	16.46±0.87	11.02±0.77	5.43±1.41	23.98±1.12	17.51±1.11	12.30±0.55	24.66±0.81	16.97±1.37	11.85±0.59
S6MOD [CVPR'25]	31.25±3.44	16.71±2.86	30.96±2.07	30.96±2.07	12.97±2.61	58.03±2.19	45.83±1.49	37.62±1.53	30.41±2.46	20.61±1.34	15.31±1.26
GGSM [AAAI'26]	22.33±1.67	11.88±1.52	16.83±1.15	11.02±0.52	5.65±1.24	23.78±0.41	17.83±0.35	12.27±0.49	31.16±0.80	20.73±0.94	16.04±1.36
DSPM (Ours)	19.86±1.52	**9.05±1.34**	**14.68±1.18**	**9.26±0.43**	**5.35±1.70**	**18.55±1.09**	**14.77±0.87**	**10.48±0.73**	**23.60±1.43**	**15.75±0.85**	**11.20±0.77**

$\mathcal{M}$=5k. These results demonstrate the effectiveness of jointly modeling the level and trend states to construct a predictive parameter trajectory that more accurately tracks the evolving low-loss region.

Average forgetting measure. Table 2 presents the AFM under different memory buffer sizes. DSPM achieves the lowest AFM in 10 out of 11 settings and ranks second in the remaining setting. The advantage is particularly evident on the more challenging datasets. On Tiny-ImageNet with $\mathcal{M}$=2k, DSPM reduces AFM from the previous best result of 20.66% to 18.55%. On ImageNet-100, it consistently achieves the lowest AFM across all memory sizes, reaching 23.60%, 15.75%, and 11.20%, respectively. Similar improvements are observed on CIFAR-100, where DSPM obtains AFM values of 14.68%, 9.26%, and 5.35% as the memory size increases. Importantly, these reductions in forgetting are accompanied by competitive or superior ACC. This indicates that DSPM does not preserve historical knowledge by excessively constraining adaptation to incoming classes, but instead achieves a more favorable stability-plasticity trade-off.

Comparison with EMA-based methods. We further compare DSPM with existing EMA-based methods, including SDP, TempEns, MKD, and GGSM. Remarkably, DSPM outperforms the strongest competitor (MKD) in 9 out of 11 ACC settings and achieves a lower AFM across all 11 settings. Unlike conventional approaches that merely smooth historical parameters, impose fixed temporal weighting or apply gradient-guided masks, DSPM explicitly estimates the student trajectory's level and trend states. This enables one-step predictive parameter tracking, thereby alleviating the temporal bias inherent in first-order EMA. Consequently, DSPM facilitates robust knowledge retention while sustaining sufficient plasticity for incoming classes, achieving a superior stability-plasticity trade-off. These empirical findings align seamlessly with our theoretical analysis, confirming that DSPM reduces trajectory tracking bias and induces a tighter empirical loss upper bound.

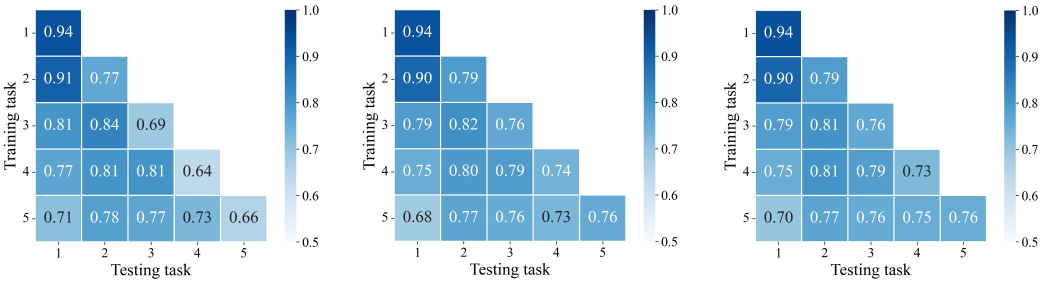
5.3 Ablation Studies

To validate the effectiveness of each DSPM component, we conduct an ablation study on three benchmarks, with the results reported in Table 3. Combining the level state with CKD_1 substantially improves the fine-tuning baseline across all benchmarks. This confirms the effectiveness of constructing a stability-oriented level teacher and transferring its output knowledge. Adding CKD_2

Table 3: Ablation study on different components of our method.

Component					CIFAR-10 ($\mathcal{M} = 1000$)		CIFAR-100 ($\mathcal{M} = 2000$)		ImageNet-100 ($\mathcal{M} = 5000$)	
Base	Level	Trend	CKD_1	CKD_2	ACC (%)	AFM (%)	ACC (%)	AFM (%)	ACC (%)	AFM (%)
✓					62.32±4.13	20.94±6.79	31.89±1.45	22.20±2.26	33.30±1.74	23.38±2.10
✓	✓		✓		72.94±0.97	8.74±0.33	44.09±0.35	8.32±0.62	41.29±0.61	15.87±1.11
✓	✓		✓	✓	73.40±0.77	**7.70±1.67**	44.48±0.41	**7.74±0.51**	43.61±0.84	**15.16±0.92**
✓	✓	✓	✓		74.10±0.59	9.45±1.58	44.84±0.83	9.56±0.76	43.70±0.61	16.52±1.17
✓	✓	✓	✓	✓	**74.51±0.48**	9.05±1.34	**45.59±0.50**	9.26±0.43	**44.55±0.66**	15.75±0.85

"Base" denotes the ER baseline; "CKD_1" and "CKD_2" denote the logit-level and feature-level distillation components of the proposed dual-granularity complementary KD strategy, respectively.



(a) Level+CKD_1 (b) Level+Trend+CKD_1 (c) Level+Trend+CKD_1+CKD_2

Figure 3: Effect of level, trend, and feature-level distillation components on test accuracy for each task over CIFAR-10 with $\mathcal{M} = 1000$.

further improves both metrics consistently, indicating that feature-level distillation provides additional representation stability beyond logit-level supervision. Compared with adding CKD_2 alone, introducing trend-state prediction yields larger ACC gains, especially on CIFAR-10 and ImageNet-100. This suggests that trend extrapolation improves plasticity by anticipating the evolving parameter trajectory, although the accompanying increase in AFM indicates a modest stability cost. Finally, combining the trend state with CKD_2 further improves ACC while reducing AFM compared with the trend-only counterpart on all benchmarks. These results demonstrate that CKD_2 effectively regularizes the predictive trajectory, enabling DSPM to exploit trend-based plasticity without incurring excessive forgetting. Additionally, Figure 3 illustrates the task-wise test accuracy on CIFAR-10 throughout the incremental training process. The comparison between Figures 3(a) and 3(b) shows that introducing the trend state substantially enhances the plasticity of the level state model. Further incorporating feature-level distillation improves model stability, as shown in Figure 3(c).

Note that supplementary experiments, including parameter sensitivity analysis, feature discrimination, empirical stability-plasticity validation via feature drift and temporal update alignment, and additional ablation studies, are provided in Appendix B.2 to offer further insights into DSPM.

6 Conclusion

In this paper, we studied the optimization trajectory of EMA-based teachers in OCL and found that conventional EMA suffers from first-order tracking bias when approximating the evolving low-loss center. To address this limitation, we proposed dual-state predictive momentum (DSPM), which maintains level and trend states of the student trajectory to construct a one-step predictive teacher. We theoretically showed that DSPM provides an asymptotically unbiased trajectory under local linear drift and induces a tighter upper bound on the average empirical loss than conventional EMA. To further enhance the teacher-student interaction, we introduced dual-granularity complementary distillation, which jointly aligns output distributions and stable representations. Extensive experiments on four benchmarks against state-of-the-art baselines demonstrate that DSPM consistently improves the stability-plasticity trade-off and delivers superior OCL performance.

Limitations and Future Work. DSPM is the first work to explore a local linear-drift approximation of the evolving low-loss center in OCL, enabling a tractable and unified characterization of trajectory bias. Nevertheless, real-world data streams may involve uncertain abrupt or highly nonlinear distribution shifts, suggesting that future work could investigate higher-order adaptive predictors or uncertainty-aware mechanisms for more reliable trend estimates.

STATEMENTS

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All experiments were conducted using established image-classification benchmarks, including CIFAR-10, CIFAR-100, Tiny-ImageNet, and ImageNet-100, which were obtained through their standard access channels and used in accordance with their applicable licenses and usage terms. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

We have taken extensive measures to support the reproducibility of this work. The source code and implementation configurations used in our experiments are publicly available in an anonymous repository. All benchmark datasets considered in this study are publicly accessible from their standard sources, facilitating evaluation under consistent data settings.

The paper and supplementary material provide detailed descriptions of the dataset splits, task construction, model architecture, optimization procedure, data augmentation strategies, batch and memory sizes, random seeds, hyperparameter search spaces, evaluation protocols, and hardware configuration. Results are averaged over five runs with fixed random seeds. We also provide the complete formulation of DSPM, including its parameter-update equations, training objective, implementation details, and PyTorch-style pseudocode. Together, the released code and documented experimental protocol are intended to enable independent replication and verification of our results.

We believe these measures will enable other researchers to reproduce our work and further advance the field.

REFERENCES

Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. In *Advances in neural information processing systems*, pp. 11872–11883, 2019a.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in neural information processing systems*, pp. 11817–11826, 2019b.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in neural information processing systems*, volume 33, pp. 15920–15930, 2020.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *Proceedings of the International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=N8MaByOzUfb.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *Proceedings of the International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019b.

Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. MetaFSCIL: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14166–14175, 2022.

Yuanning Cui, Yuxin Wang, Zequn Sun, Wenqiang Liu, Yiqiao Jiang, Kexin Han, and Wei Hu. Lifelong embedding learning and transfer for growing knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4217–4224, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

Saber Elaydi. *An introduction to difference equations*. Springer, 2005. URL https://link.springer.com/book/10.1007/0-387-27602-5.

Tongtong Feng, Xin Wang, Zekai Zhou, Ren Wang, Yuwei Zhan, Guangyao Li, Qing Li, and Wenwu Zhu. EvoAgent: Self-evolving agent with continual world model for long-horizon tasks. *arXiv preprint arXiv:2502.05907*, 2025.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the International conference on machine learning*, pp. 3259–3269, 2020.

Nuwan Gunasekara, Bernhard Pfahringer, Heitor Murilo Gomes, and Albert Bifet. Survey on online streaming continual learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 6628–6637, 2023.

Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *Proceedings of the International conference on machine learning*, pp. 8109–8126, 2022.

Yiduo Guo, Bing Liu, and Dongyan Zhao. Dealing with cross-task class discrimination in online continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11878–11887, 2023.

Zihao Guo, Qingyun Sun, Ziwei Zhang, Haonan Yuan, Huiping Zhuang, Xingcheng Fu, and Jianxin Li. GraphKeeper: Graph domain-incremental learning via knowledge disentanglement and preservation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 1145–1171, 2025.

Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *European conference on computer vision*, pp. 466–483, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 831–839, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Hyunseo Koh, Minhyuk Seo, Jihwan Bang, Hwanjun Song, Deokki Hong, Seulki Park, Jung-Woo Ha, and Jonghyun Choi. Online boundary-free continual learning by scheduled data prior. In *Proceedings of the International Conference on Learning Representations*, pp. 26701–26723, 2023.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. URL `https://cave.cs.toronto.edu/kriz/learning-features-2009-TR.pdf`.

Yann Le, Xuan Yang, et al. Tiny ImageNet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Depeng Li and Zhigang Zeng. CRNet: A fast continual learning framework with random theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10731–10744, 2023.

Guoliang Lin, Hanlu Chu, and Hanjiang Lai. Towards better plasticity-stability trade-off in incremental learning: A simple linear connector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 89–98, 2022.

Huiwei Lin. ER-FSL: Experience replay with feature subspace learning for online continual learning. *arXiv preprint arXiv:2407.12279*, 2024.

Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. PCR: Proxy-based contrastive replay for online class-incremental continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24246–24255, 2023.

Haochen Liu, Tianyu Li, Haohan Yang, Li Chen, Caojun Wang, Ke Guo, Haochen Tian, Hongchen Li, Hongyang Li, and Chen Lv. Reinforced refinement with self-aware expansion for end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(5): 5774–5792, 2026.

Sihao Liu, Yibo Yang, Xiaojie Li, David A Clifton, and Bernard Ghanem. Enhancing online continual learning with plug-and-play state space model and class-conditional mixture of discretization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20502–20511, 2025a.

Xiaohao Liu, Xiaobo Xia, See-Kiong Ng, and Tat-Seng Chua. Continual multimodal contrastive learning. In *Advances in Neural Information Processing Systems*, pp. 48414–48455, 2025b.

Yiwen Luo, Wuyang Li, Cheng Chen, Xiang Li, Tianming Liu, Tianye Niu, and Yixuan Yuan. LLM-guided decoupled probabilistic prompt for continual learning in medical image diagnosis. *IEEE Transactions on Medical Imaging*, 44(8):3439–3450, 2025.

Imad Eddine Marouf, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière. Weighted ensemble models are strong continual learners. In *European Conference on Computer Vision*, pp. 306–324, 2024.

Nicolas Michel, Maorong Wang, Ling Xiao, and Toshihiko Yamasaki. Rethinking momentum knowledge distillation in online continual learning. In *Proceedings of the International Conference on Machine Learning*, pp. 35607–35622, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, volume 32, pp. 8024–8035, 2019.

Benliu Qiu, Heqian Qiu, Lanxiao Wang, Taijin Zhao, Yu Dai, Lili Pan, and Hongliang Li. Parameter merging with gradient-guided supermasks in online continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24991–24999, 2026.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pp. 350–360, 2019.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *Proceedings of the International Conference on Machine Learning*, pp. 4528–4537, 2018.

Albin Soutif-Cormerais, Antonio Carta, and Joost van de Weijer. Improving online continual learning performance and stability with temporal ensembles. In *Proceedings of The Conference on Lifelong Learning Agents*, volume 232, pp. 828–845, 2023.

Paraskevi-Antonia Theofilou, Anuhya Thota, Stefanos Kollias, and Mamatha Thota. Stable-drift: A patient-aware latent drift replay method for stabilizing representations in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7340–7349, 2025.

Edoardo Urettini and Antonio Carta. Online curvature-aware replay: Leveraging second order information for online continual learning. In *Proceedings of the International Conference on Machine Learning*, pp. 60590–60609, 2025.

Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11):2579–2605, 2008.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10087–10098, 2025a.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024a.

Maorong Wang, Nicolas Michel, Ling Xiao, and Toshihiko Yamasaki. Improving plasticity in online continual learning via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23460–23469, 2024b.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance with self-supervision for incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2563–2580, 2025b.

Zhenyi Wang, Enneng Yang, Li Shen, and Heng Huang. A comprehensive survey of forgetting in deep learning beyond continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1464–1483, 2025c.

Zifeng Wang, Zheng Zhan, Yifan Gong, Geng Yuan, Wei Niu, Tong Jian, Bin Ren, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. SparCL: Sparse continual learning on the edge. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20366–20380, 2022.

Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 18764–18774, 2023.

Haitao Wen, Haoyang Cheng, Heqian Qiu, Lanxiao Wang, Lili Pan, and Hongliang Li. Optimizing mode connectivity for class incremental learning. In *Proceedings of the International Conference on Machine Learning*, pp. 36940–36957, 2023.

Mitchell Wortsman, Maxwell C Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. Learning neural network subspaces. In *Proceedings of the International Conference on Machine Learning*, pp. 11217–11227, 2021.

Yichen Wu, Hong Wang, Peilin Zhao, Yefeng Zheng, Ying Wei, and Long-Kai Huang. Mitigating catastrophic forgetting in online continual learning by modeling previous task interrelations via pareto optimization. In *Proceedings of the International conference on machine learning*, volume 235, pp. 53892–53908, 2024.

Hongwei Yan, Liyuan Wang, Kaisheng Ma, and Yi Zhong. Orchestrate latent expertise: Advancing online continual learning with multi-level supervision and reverse self-distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23670–23680, 2024.

Enneng Yang, Li Shen, Zhenyi Wang, Shiwei Liu, Guibing Guo, and Xingwei Wang. Data augmented flatness-aware gradient projection for continual learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5630–5639, 2023.

Jason Yoo, Yunpeng Liu, Frank Wood, and Geoff Pleiss. Layerwise proximal replay: a proximal point method for online continual learning. In *Proceedings of the International Conference on Machine Learning*, pp. 57199–57216, 2024.

Xikun Zhang, Dongjin Song, Yixin Chen, and Dacheng Tao. Topology-aware embedding memory for continual learning on expanding networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4326–4337, 2024.

A APPENDIX

A.1 HYPOTHESIS OF LOCAL LINEAR DRIFT

In OCL, as new data streams arrive continuously, the optimal parameter center $\boldsymbol{\theta}_t^*$ is typically not static. Assume that within a local time window, the dynamic low-loss center satisfies a linear drift:

$$\boldsymbol{\theta}_{t+1}^* = \boldsymbol{\theta}_t^* + \boldsymbol{\Delta}_{\boldsymbol{\theta}}. \tag{20}$$

where $\boldsymbol{\Delta}_{\boldsymbol{\theta}}$ indicates the linear drift of the low-loss center. Further assume that the current student model has adapted well to the current stage, that is $\boldsymbol{\theta}_t = \boldsymbol{\theta}_t^* + \delta_t$, where δ_t is the optimization error of the current student model relative to the low-loss center. In order to highlight the deviation of the teacher trajectory itself, we currently consider the ideal situation: $\boldsymbol{\theta}_t = \boldsymbol{\theta}_t^*$.

A.2 STEADY-STATE EQUILIBRIUM CONDITION

Assume that, within a local temporal window, the optimal parameter of the discrete tracking system follows a locally constant linear drift with $\boldsymbol{v} \in \mathbb{R}^k$. The corresponding error dynamics can be generally formulated as a k order discrete state-space model, as follows:

$$\boldsymbol{e}_t = \boldsymbol{H}\boldsymbol{e}_{t-1} + \boldsymbol{C}\boldsymbol{v} \tag{21}$$

where $\boldsymbol{e}_t \in \mathbb{R}^k$ denotes the state vector of tracking error, $\boldsymbol{H} \in \mathbb{R}^{k\times k}$ represents the state transition matrix, and $\boldsymbol{C} \in \mathbb{R}^{k\times k}$ is the constant driving matrix.

According to the asymptotic stability theory for discrete-time dynamic systems (Elaydi, 2005), if $\boldsymbol{H}$ is Schur stable, i.e., its spectral radius is strictly less than 1, then this linear nonhomogeneous difference system admits a unique globally asymptotically stable equilibrium. As $t \to \infty$, the transient response induced by the initial state vanishes, and the error state converges to the steady-state equilibrium point $\boldsymbol{e}^*$, as follows:

$$\lim_{t\to\infty} \boldsymbol{e}_t = \boldsymbol{e}^* = (\boldsymbol{I} - \boldsymbol{H})^{-1}\boldsymbol{C}\boldsymbol{v}. \tag{22}$$

As such, once the system reaches the steady-state equilibrium, the error state of adjacent time steps no longer changes, strictly satisfying $\boldsymbol{e}_t = \boldsymbol{e}_{t-1}$.

A.3 UNBIASED TRAJECTORY ERROR OF DSPM

The level and trend tracking errors of DSPM-based optimization trajectory are defined as follows:

$$\boldsymbol{e}_t^L = \boldsymbol{L}_t - \boldsymbol{\theta}_t^* \tag{23}$$

$$\boldsymbol{e}_t^B = \boldsymbol{B}_t - \boldsymbol{\Delta}_{\boldsymbol{\theta}}. \tag{24}$$

Recalling equation 12 and the hypothesis of local linear drift, the level state is updated as

$$\boldsymbol{L}_t = \alpha\boldsymbol{\theta}_t^* + (1-\alpha)(\boldsymbol{L}_{t-1} + \boldsymbol{B}_{t-1}). \tag{25}$$

Since $\boldsymbol{L}_{t-1} = \boldsymbol{\theta}_{t-1}^* + \boldsymbol{e}_{t-1}^L$ and $\boldsymbol{B}_{t-1} = \boldsymbol{\Delta}_\theta + \boldsymbol{e}_{t-1}^B$, we have $\boldsymbol{L}_{t-1} + \boldsymbol{B}_{t-1} = \boldsymbol{\theta}_t^* + \boldsymbol{e}_{t-1}^L + \boldsymbol{e}_{t-1}^B$. Substituting this relation into the above equation, the level error is derived as follows:

$$\boldsymbol{e}_t^L = \boldsymbol{L}_t - \boldsymbol{\theta}_t^* = (1-\alpha)(\boldsymbol{e}_{t-1}^L + \boldsymbol{e}_{t-1}^B). \tag{26}$$

Likewise, since $\boldsymbol{L}_t - \boldsymbol{L}_{t-1} = \boldsymbol{\Delta}_{\boldsymbol{\theta}} + \boldsymbol{e}_t^L - \boldsymbol{e}_{t-1}^L$, the trend state update in equation 13 can be expressed as:

$$\boldsymbol{B}_t = \beta(\boldsymbol{\Delta}_{\boldsymbol{\theta}} + \boldsymbol{e}_t^L - \boldsymbol{e}_{t-1}^L) + (1-\beta)(\boldsymbol{\Delta}_{\boldsymbol{\theta}} + \boldsymbol{e}_{t-1}^B). \tag{27}$$

Afterwards, we can obtain the recursive formula of the trend error below:

$$\boldsymbol{e}_t^B = \boldsymbol{B}_t - \boldsymbol{\Delta}_{\boldsymbol{\theta}} = \beta(\boldsymbol{e}_t^L - \boldsymbol{e}_{t-1}^L) + (1-\beta)\boldsymbol{e}_{t-1}^B = -\alpha\beta\boldsymbol{e}_{t-1}^L + (1-\alpha\beta)\boldsymbol{e}_{t-1}^B. \tag{28}$$

We define the joint state error as $\boldsymbol{e}_t$, leading to the following second-order discrete state-space model of DSPM error dynamics:

$$\boldsymbol{e}_t = \boldsymbol{H}\boldsymbol{e}_{t-1} \tag{29}$$

where $\boldsymbol{e}_t = \begin{bmatrix} \boldsymbol{e}_t^L \\ \boldsymbol{e}_t^B \end{bmatrix}$, and $\boldsymbol{H} = \begin{bmatrix} 1-\alpha & 1-\alpha \\ -\alpha\beta & 1-\alpha\beta \end{bmatrix}$.

Algorithm 1 PyTorch-like pseudo-code of DSPM.

```
# student: student model
# teacher: teacher model
# level: level state of teacher model
# optim: optimizer for student model
for x, y in dataloader:
  # Baseline loss
  loss_baseline = criterion_baseline(student, x, y)

  x_aug = transform(x) # data augmentation
  f_stu1, l_stu1 = student(x) # features and logits of student
  f_stu2, l_stu2 = student(x_aug)
  f_tea, l_tea = teacher(x_aug) # features and logits of teacher
  f_lev, l_lev = level(x_aug) # features and logits of level state

  # Proposed loss
  loss_ce = cross_entropy(l_stu2, y)
  loss_ckd1 = kl_div(softmax(l_tea/τ), softmax(l_stu1/τ)) + kl_div(softmax(l_tea/τ),
  ↪  softmax(l_stu2/τ)) # temperature τ
  loss_ckd2 = norm(f_lev - f_stu1)**2 + norm(f_lev - f_stu2)**2
  loss_ours = loss_ce + mu1*loss_ckd1 + mu2*loss_ckd2 # equation 18
  loss = loss_baseline + loss_ours

  optim.zero_grad()
  loss.backward()
  optim.step()
  update_dspm() # equations 12-14
```

As shown in ***Proof 1***, the transition matrix $\boldsymbol{H}$ is Schur stable. Hence, this homogeneous error system is asymptotically stable, and the joint tracking error converges to the unique equilibrium. We can obtain

$$\boldsymbol{e}_L^* = (1-\alpha)(\boldsymbol{e}_L^* + \boldsymbol{e}_B^*), \boldsymbol{e}_B^* = (1-\alpha)\boldsymbol{e}_B^*. \tag{30}$$

where $\boldsymbol{e}_L^*$ and $\boldsymbol{e}_B^*$ are equilibrium points of $\boldsymbol{e}_L$ and $\boldsymbol{e}_B$, respectively. It is derived that $\boldsymbol{e}_L^* = 0$ and $\boldsymbol{e}_B^* = 0$. Consequently, we can obtain $\boldsymbol{L}_t = \boldsymbol{\theta}_t^*$ and $\boldsymbol{B}_t = \boldsymbol{\Delta}_\theta$. Finally, we arrive at the following equation:

$$\boldsymbol{T}_{t-1} = \boldsymbol{L}_{t-1} + \boldsymbol{B}_{t-1} = \boldsymbol{\theta}_{t-1}^* + \boldsymbol{\Delta}_\theta = \boldsymbol{\theta}_t^*. \tag{31}$$

This indicates that the DSPM optimization trajectory is asymptotically unbiased for one-step prediction under a uniform hypothesis of local linear drift.

Proof 1 (Schur Stability of DSPM Error). For the DSPM's state transition matrix $\boldsymbol{H}$:

$$\boldsymbol{H} = \begin{bmatrix} 1-\alpha & 1-\alpha \\ -\alpha\beta & 1-\alpha\beta \end{bmatrix}, 0 < \alpha < 1, 0 < \beta < 1, \tag{32}$$

we can calculate its characteristic polynomial as follows:

$$p(\lambda) = \det(\lambda \boldsymbol{I} - \boldsymbol{H}) = (\lambda - 1 - \alpha)(\lambda - 1 + \alpha\beta) + \alpha\beta(1-\alpha). \tag{33}$$

Let $x = \lambda - 1$. Then, the eigenvalues of $\boldsymbol{H}$ can be written as $\lambda_i = 1 + x_i$, where x_i denotes the roots of the characteristic polynomial:

$$p(x) = x^2 + \alpha(1+\beta)x + \alpha\beta = 0 \tag{34}$$

By Vieta's formulas, the two roots x_1 and x_2 satisfy

$$x_1 + x_2 = -\alpha(1+\beta) < 0 \tag{35}$$

$$x_1 x_2 = \alpha\beta > 0. \tag{36}$$

As a result, those two roots x_1 and x_2 both are negative numbers with $x_1 < 0$ and $x_2 < 0$. Finally, we can yield

$$\lambda_1 = 1 + x_1 < 1, \lambda_2 = 1 + x_2 < 1. \tag{37}$$

Therefore, all eigenvalues of $\boldsymbol{H}$ lie strictly inside the unit circle, i.e., $\rho(H) < 1$. This has proved the Schur stability of the DSPM error.

A.4 Pseudo-Code of DSPM

In Algorithm 1, we provide PyTorch-like (Paszke et al., 2019) pseudo-code to outline the proposed DSPM, which can be readily integrated into existing baselines. For simplicity, the memory buffer is omitted in the pseudo-code. When rehearsal is adopted, the same training procedure applies by constructing each batch from both incoming stream data and memory data.

B APPENDIX

B.1 IMPLEMENTATION DETAILS

Datasets. As described in section 5.1, we evaluate the effectiveness of our method on four benchmark datasets. Each original dataset is split into multiple tasks with disjoint classes. For a fair comparison, the classes' allocation to tasks and the sequence of tasks are randomized using 5 fixed random seeds. Detailed information on dataset splits and task allocation is summarized as follows:

CIFAR-10 (Krizhevsky et al., 2009) consists of 10 classes, with 50,000 training images and 10,000 test images. Each image has a resolution of 32×32. CIFAR-10 is divided into 5 class-disjoint tasks with 2 classes per task in our experiments.

CIFAR-100 (Krizhevsky et al., 2009) contains 100 classes, with 50,000 training images and 10,000 test images. Each image has a resolution of 32×32. We split CIFAR-100 into 10 class-disjoint tasks with 10 classes per task.

Tiny-ImageNet (Le et al., 2015) has 200 classes with 100,000 training samples and 10,000 test samples. Each image has a resolution of 64×64. It is split into 100 non-overlapping tasks with 2 classes per task.

ImageNet-100 (Hou et al., 2019) is the first 100-class subset of ImageNet-1k (Deng et al., 2009), containing 1300 training images and 50 testing images per class. The images are 224×224 in size. It is split into 20 class-disjoint tasks with 5 classes per task.

Metrics. Two commonly used evaluation metrics, average test accuracy (ACC) and average forgetting measure (AFM) (Liu et al., 2025a), are adopted in this paper. A superior method is expected to yield higher ACC values and lower AFM values. For clarity, we present their detailed definitions in the following.

In CL, after completing all $\mathcal{T}$ tasks, the continual learner needs to be tested on all previously learned tasks $\{1, 2, ..., \mathcal{T}\}$. The ACC is defined as follows:

$$ACC = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \mathcal{R}_{\mathcal{T},t} \tag{38}$$

where $\mathcal{R}_{\mathcal{T},t}$ is the test accuracy on task t after learning task $\mathcal{T}$.

AFM measures the average performance drop from the best historical accuracy of each task to its final accuracy after learning all tasks. A lower AFM indicates better retention of previous knowledge, whereas a higher AFM suggests more severe forgetting. It is calculated as follows:

$$AFM = \frac{1}{\mathcal{T}-1} \sum_{t=2}^{\mathcal{T}} \max_{i \in \{1,...,t-1\}} (\mathcal{R}_{i,i} - \mathcal{R}_{t,i}) \tag{39}$$

Data Augmentation. Data augmentation has been shown to improve OCL performance, particularly in KD-based methods (Wang et al., 2024b; Michel et al., 2024; Qiu et al., 2026). However, different methods may benefit from different augmentation strengths, and simple augmentations can sometimes outperform more complex ones. Therefore, to ensure a fair and optimal comparison, we consider two augmentation strategies for all methods, as follows:

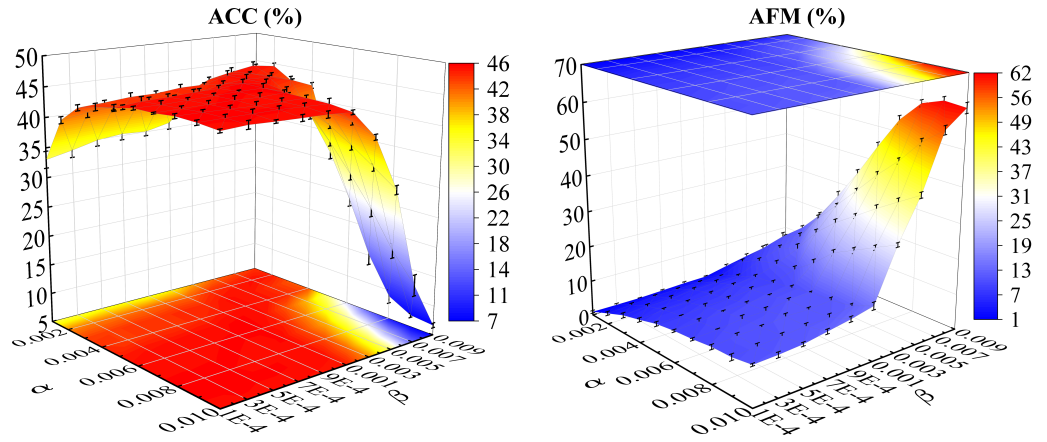
Full augmentation strategy incorporates a broader set of data transformation techniques. It involves a sequence of random cropping, horizontal flipping, color jitter, and random grayscale transformations. The color jitter parameters are set to (0.4, 0.4, 0.4, 0.1) with an application probability of 0.8, while random grayscale is applied with a probability of 0.2.

Partial augmentation strategy consists of a subset of the augmentations used in the full strategy. Specifically, it sequentially applies random cropping and random horizontal flipping, each with a probability of 0.5.

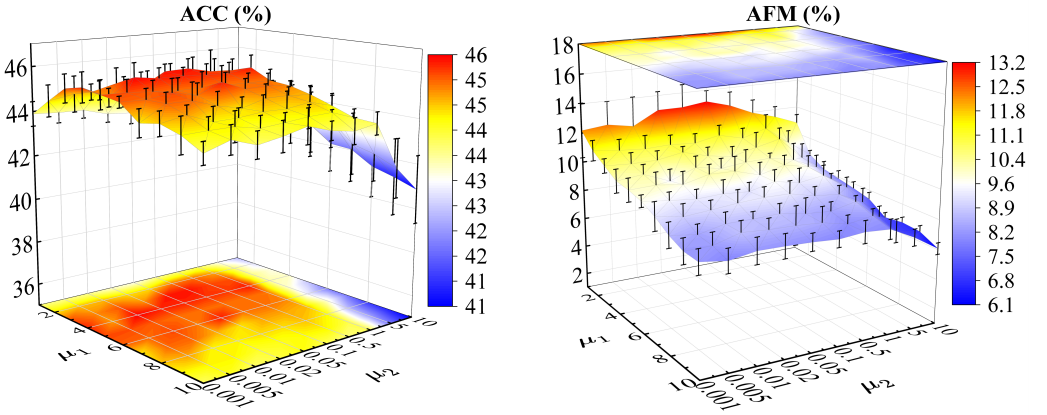
Hyperparameter Settings. For MKD and GGSM, we adopt the hyperparameter settings recommended in their original papers. For the remaining baselines, their hyperparameters are selected through grid search over the search ranges reported in Table 4. The grid search is performed on CIFAR-100 with a memory size of $\mathcal{M} = 2$k.

Table 4: Hyperparameter configuration ranges for different methods.

Method	Hyperparameter	Search range
ER	Optimizer	{SGD, Adam}
	Weight decay	{0, 1e-4}
	Learning rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}
	Momentum	{0, 0.9}
	Data augmentation	{full, partial}
DER++	Optimizer	{SGD, Adam}
	Weight decay	{0, 1e-4}
	Learning rate	{0.0001, 0.001, 0.01, 0.03}
	Momentum	{0, 0.9}
	Data augmentation	{full, partial}
	α	{0.1, 0.2, 0.5, 1.0}
	β	{0.5, 1.0}
ER-ACE	Optimizer	{SGD, Adam}
	Weight decay	{0, 1e-4}
	Learning rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}
	Momentum	{0, 0.9}
	Data augmentation	{full, partial}
GSA	Optimizer	{SGD, Adam}
	Weight decay	{0, 1e-4}
	Learning rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}
	Momentum	{0, 0.9}
	Data augmentation	{full, partial}
PCR	Optimizer	{Adam}
	Weight decay	{0}
	Learning rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005}
	Momentum	{0}
	Data augmentation	{full}
SDP	Optimizer	{Adam}
	Weight decay	{0}
	Learning rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005}
	Momentum	{0}
	Data augmentation	{full}
	μ	{10, 100, 1000, 10000}
	c^2	{0.5, 0.75, 0.9}
TempEns	Optimizer	{Adam}
	Weight decay	{0}
	Learning rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005}
	Momentum	{0}
	Data augmentation	{full}
	α	{0.01, 0.05, 0.1, 0.5, 0.9, 0.95, 0.99}
CCL-DC	Optimizer	{Adam}
	Weight decay	{0}
	Learning rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005}
	Momentum	{0}
	Data augmentation	{full}
	N	{1, 2, 3, 4}
	M	{7, 9, 11, 13, 15, 17}
	λ_1	{0.25, 0.5, 1, 2, 4, 8}
	λ_2	{0.25, 0.5, 1, 2, 4, 8}
S6MOD	Optimizer	{Adam}
	Weight decay	{0}
	Learning rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005}
	Momentum	{0}
	Data augmentation	{full}
	N	{8, 10}
	α	{0.01, 0.1, 0.5, 1, 5}
	β	{5×α}



(a) Only varying α and β



(b) Only varying μ_1 and μ_2

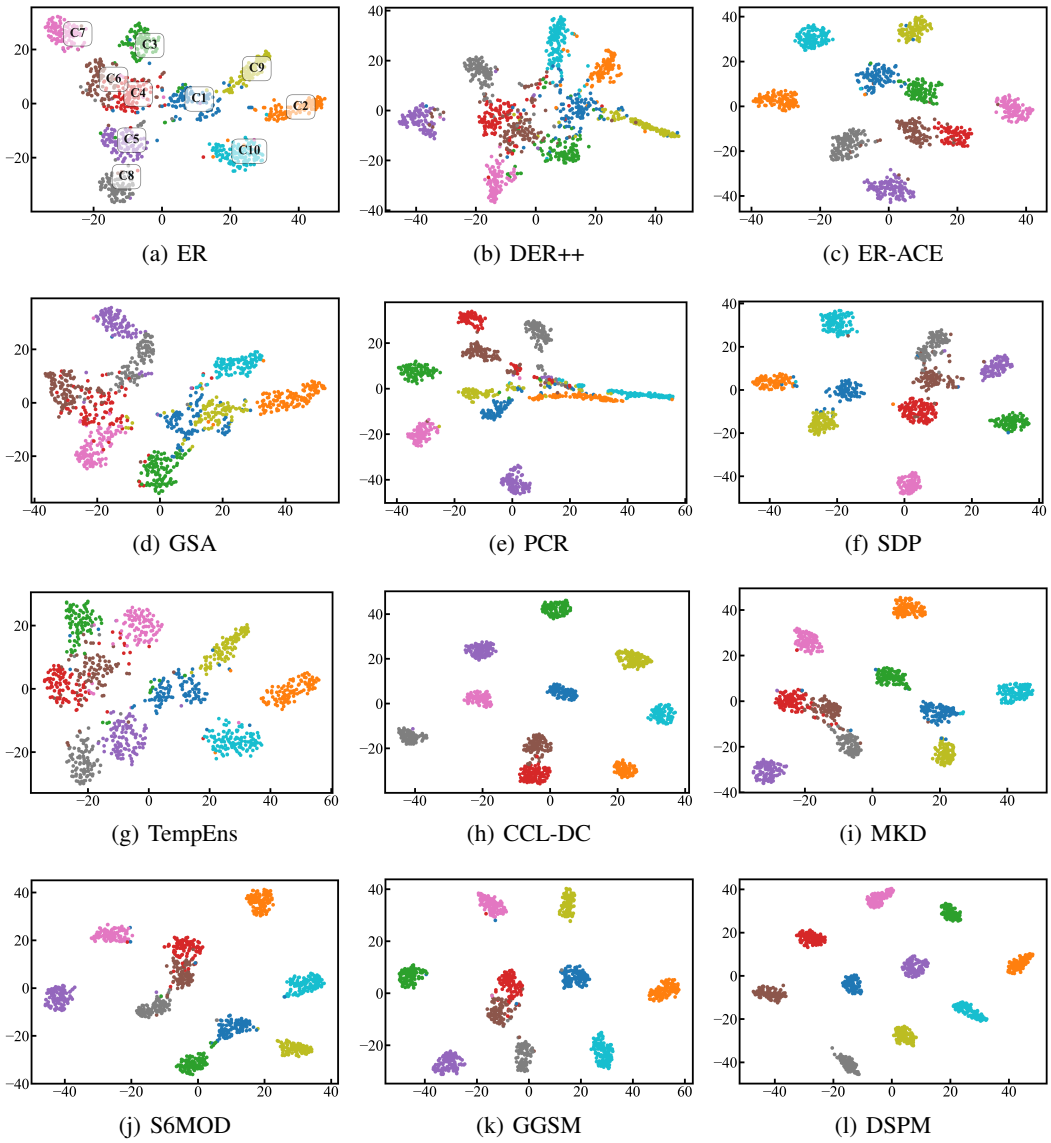
Figure 4: Impact of DSPM hyperparameters on ACC and AFM on CIFAR-100 with $\mathcal{M}$=2k.

B.2 Supplementary Experiments

Parameter Sensitivity Analysis. The hyperparameters of DSPM can be divided into two groups. The first group includes α and β, which are the smoothing coefficients of the level-state and trend-state models, respectively. The second group consists of μ_1 and μ_2, which serve as regularization weights in the objective function. Their corresponding experiments for parameter sensitivity analysis are conducted on CIFAR-100 with $\mathcal{M}$=2k, and the results are illustrated in Figure 4.

Initially, α and β are varied, while μ_1 and μ_2 are fixed at 6 and 0.05. The impact of α and β on ACC and AFM is explored, as depicted in Figure 4(a). When $\alpha \in [0.003, 0.01]$ and $\beta \in [0.0001, 0.002]$, the surfaces of ACC and AFM are fairly flat. We can also find that AFM increases with the rise of β, especially showing a significant increase when $\beta > 0.003$, indicating that a large β makes the trend state overly sensitive to short-term noise and mini-batch fluctuations, leading to teacher overshooting and degraded stability. Generally, $\beta < \alpha \ll 1$ yields better OCL performance. This is because α enables the teacher to stably follow the main optimization direction of the student parameters, while β provides a compensatory correction by estimating local trends at a slower update rate. It helps construct a higher-precision predictive optimization trajectory without amplifying noise. Notably, when $\alpha \in [0.005, 0.008]$ and $\beta \in [0.0005, 0.001]$, DSPM achieves a superior stability-plasticity trade-off in this OCL setting.

Subsequently, we conduct sensitivity analysis for μ_1 and μ_2 while fixing $\alpha = 0.005$ and $\beta = 0.001$, with the results shown in Figure 4(b). Increasing μ_1 strengthens logit-level knowledge transfer from the predictive teacher, improving ACC and reducing AFM. However, an excessively large μ_1



(a) ER (b) DER++ (c) ER-ACE

(d) GSA (e) PCR (f) SDP

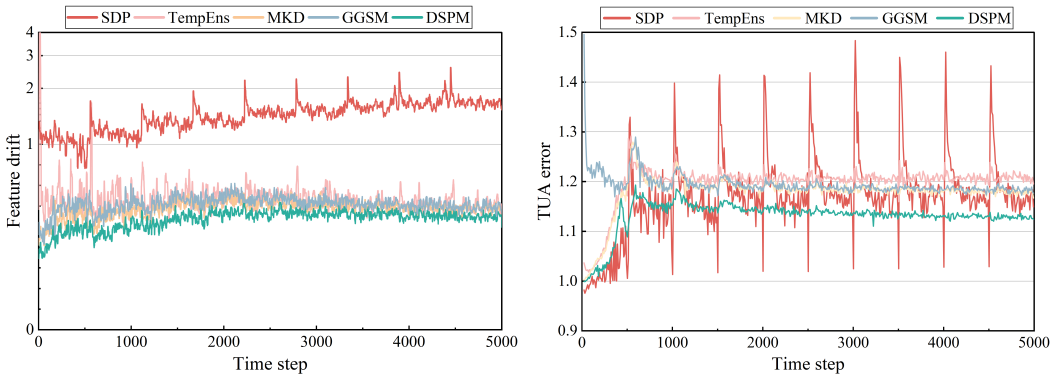
(g) TempEns (h) CCL-DC (i) MKD

(j) S6MOD (k) GGSM (l) DSPM

Figure 5: t-SNE visualization of memory data at the end of training on CIFAR-10 with $\mathcal{M}$=1k.

gradually degrades ACC, suggesting that overly strong logit-level regularization may restrict the student's adaptation to incoming classes. A similar pattern is observed for μ_2. Moderate feature-level distillation improves representation stability and mitigates forgetting, whereas an excessively large μ_2 reduces AFM at the expense of ACC, indicating over-regularization of the representation space. Overall, DSPM achieves a favorable trade-off over a broad region where $\mu_1 \in [4, 7]$ and $\mu_2 \in [0.001, 1]$. In this regime, the model benefits from stable teacher guidance while preserving sufficient student plasticity. Specifically, $\mu_1 = 6$ and $\mu_2 = 0.05$ are adopted as they provide strong ACC with low AFM, yielding a well-balanced objective for OCL.

Feature Discrimination. Learning discriminative features is essential for continual learners, as it promotes effective separation of semantic concepts and facilitates reliable classification. Thus, we present t-SNE visualizations Van der Maaten & Hinton (2008) on memory data at the end of training for baselines and our DSPM, as shown in Figure 5. We used a perplexity of 30 for all experiments. The corresponding class labels are annotated for each cluster in Figure 5(a). We can observe that recent state-of-the-art methods still struggle to distinguish between classes 6 and 4, as evidenced



(a) Feature drift (b) Temporal update alignment error

Figure 6: Feature drift and temporal update alignment error of EMA-based methods on CIFAR-100 with $\mathcal{M}$=2k.

Table 5: Additional ablation study on different components of our method.

Component					CIFAR-10 ($\mathcal{M}=1000$)		CIFAR-100 ($\mathcal{M}=2000$)		ImageNet-100 ($\mathcal{M}=5000$)	
Base	Level	Trend	CKD_1	CKD_2	ACC (%)	AFM (%)	ACC (%)	AFM (%)	ACC (%)	AFM (%)
✓					62.32±4.13	20.94±6.79	31.89±1.45	22.20±2.26	33.30±1.74	23.38±2.10
✓	✓			✓	70.28±0.78	16.80±1.41	38.80±0.84	12.33±1.08	40.30±0.76	15.57±1.00
✓	✓	✓		✓	67.52±1.57	22.75±4.10	36.86±0.93	23.58±0.98	37.14±0.78	25.13±1.39

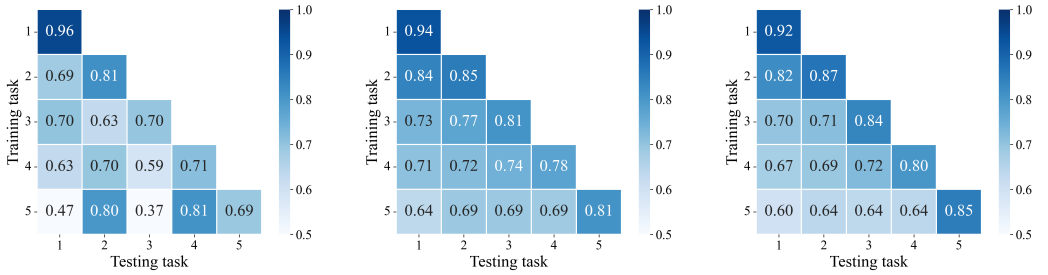
"Base" denotes the ER baseline; CKD_1" and "CKD_2" denote the logit-level and feature-level distillation components of the proposed dual-granularity complementary KD strategy, respectively.

by their overlapping feature distributions. In contrast, our method achieves more discriminative representations with clearer separation among different classes.

Feature Drift. Feature drift arises when a model updates its network parameters to learn a new task, causing the representations of previously learned data to shift in the feature space and potentially rendering the established decision boundaries unreliable. To this end, we empirically analyze the feature drift of EMA-based methods, including SDP, TempEns, MKD, GGSM, and our DSPM, to further evaluate their representation stability, as shown in Figure 6(a). Specifically, feature drift is measured as $\|\mathcal{F}(\boldsymbol{x}_{old};\boldsymbol{\theta}_t)-\mathcal{F}(\boldsymbol{x}_{old};\boldsymbol{\theta}_{t+1})\|_2$, where $\boldsymbol{x}_{old}$ denotes memory samples from previously learned classes. For clearer visualization, the resulting drift curves are smoothed using a median filter with a kernel size of 9. As we can see, SDP and TempEns exhibit substantial feature drift with pronounced fluctuations, including sharp spikes at task transitions. Recent EMA-based methods, such as MKD and GGSM, exhibit relatively small feature drift. Nevertheless, DSPM achieves the lowest and most stable feature drift among all compared methods.

Temporal Update Alignment. In addition to using feature drift to evaluate the stability of learned representations, we introduce temporal update alignment (TUA) to characterize the plasticity of EMA teachers constructed by EMA-based OCL methods. The TUA error is defined as $1-\cos\left(\boldsymbol{T}_t-\boldsymbol{T}_{t-1},\boldsymbol{\theta}_t-\boldsymbol{\theta}_{t-1}\right)$, which indicates whether the teacher can accurately track the student's optimization trajectory. Therefore, TUA provides an empirical measurement of the trajectory tracking fidelity of EMA teachers. A smaller TUA error indicates that the teacher more accurately follows the student's parameter updates, enabling more faithful knowledge transfer with better plasticity. Figure 6(b) illustrates the TUA errors of EMA-based methods. SDP exhibits a clear periodic variation pattern in TUA error that correlates with task transitions, indicating instability in tracking the student's update trajectory. The other methods maintain relatively stable TUA errors as training proceeds. However, GGSM suffers from the largest TUA error during the initial task. In contrast, DSPM achieves a lower TUA error from the beginning and further reduces the error after the initial task, consistently maintaining the smallest TUA values among all compared methods. This observation is consistent with the theoretical analysis of DSPM in Appendix A.3.

Additional Ablation Studies. Note that the trend state is derived from the level state and thus cannot



(a) Base (b) Base+Level+CKD$_2$ (c) Base+Level+Trend+CKD$_2$

Figure 7: Effect of base, level, and trend components under feature-level distillation on task-wise test accuracy on CIFAR-10 with $\mathcal{M}$=1k.

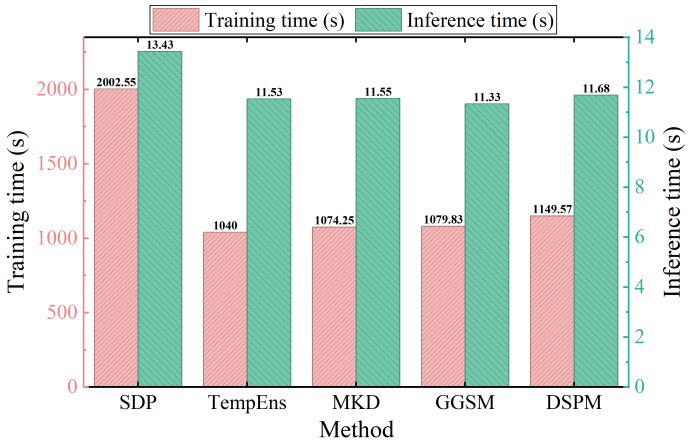


Figure 8: Training and inference times of EMA-based methods on CIFAR-100 with $\mathcal{M}$=2k.

be ablated independently, whereas CKD$_1$ and CKD$_2$ are separable modules. As a complement to Table 3, Table 5 further isolates the effect of CKD$_2$. Figure 7 demonstrates task-wise test accuracy for base, level, and trend components under feature-level distillation on CIFAR-10 with $\mathcal{M}$=1k. Augmenting the ER baseline with the level state and CKD$_2$ yields consistent improvements across all datasets, indicating that CKD$_2$ effectively mitigates forgetting by using the level state as a stable feature-space anchor. However, adding the trend state without CKD$_1$ degrades performance relative to the Level+CKD$_2$ variant. This result suggests that the trend state is not designed to serve as a standalone regularizer for the holistic DSPM. Instead, it constructs a predictive teacher whose main benefit should be transferred through logit-level distillation. Without CKD$_1$, the extrapolated trend information cannot be effectively utilized at the individual feature-distribution level and may introduce instability in OCL. These results demonstrate the complementary roles of CKD$_1$ and CKD$_2$: CKD$_1$ transfers the predictive knowledge of the trend-aware teacher, while CKD$_2$ preserves stable representations from the level state. Their combination enables DSPM to achieve a more effective balance between plasticity and stability.

Hardware and Computation. All experiments are conducted on NVIDIA RTX 5090 GPUs. Figure 8 reports the training and inference times (in seconds) of EMA-based methods on CIFAR-100 with $\mathcal{M}$=2k. TempEns achieves the lowest overall runtime because it employs a single EMA model without knowledge distillation. In contrast, SDP incurs the highest computational cost due to its use of two EMA models together with dynamic knowledge distillation. Compared with SDP, our DSPM substantially reduces runtime while maintaining computational overhead comparable to that of single-EMA-based methods.

B.3 AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing and grammatical correction to improve the clarity and readability of the manuscript. We did not use generative AI tools to generate the manuscript's substantive content or synthetic data, nor for any other tasks requiring mandatory disclosure. All AI-assisted edits were carefully reviewed and verified by the authors. We take full responsibility for the final content of this work, including any text produced or revised with the aid of generative AI.