Model Sensitivity Aware Continual Learning

Zhenyi Wang and Heng Huang
Department of Computer Science, Institute of Health Computing
University of Maryland College Park
College Park, MD, 20742
zwang@umd.edu;heng@umd.edu

Abstract

Continual learning (CL) aims to adapt to non-stationary data distributions while retaining previously acquired knowledge. However, CL models typically face a trade-off between preserving old task knowledge and excelling in new task performance. Existing approaches often sacrifice one for the other. To overcome this limitation, orthogonal to existing approaches, we propose a novel perspective that views the CL model ability in preserving old knowledge and performing well in new task as a matter of model sensitivity to parameter updates. *Excessive* parameter sensitivity can lead to two drawbacks: (1) significant forgetting of previous knowledge; and (2) overfitting to new tasks. To reduce parameter sensitivity, we optimize the model's performance based on the parameter distribution, which achieves the worst-case CL performance within a distribution neighborhood. This innovative learning paradigm offers dual benefits: (1) reduced forgetting of old knowledge by mitigating drastic changes in model predictions under small parameter updates; and (2) enhanced new task performance by preventing overfitting to new tasks. Consequently, our method achieves superior ability in retaining old knowledge and achieving excellent new task performance simultaneously. Importantly, our approach is compatible with existing CL methodologies, allowing seamless integration while delivering significant improvements in effectiveness, efficiency, and versatility with both theoretical and empirical supports.

1 Introduction

Continual learning (CL) embodies a dynamic approach aimed at adapting to non-stationary data distributions that evolve over time. However, in pursuit of this goal, CL encounters a significant challenge: the trade-off between preserving previously acquired knowledge and effectively learning new knowledge. As the model assimilates new information, it often swiftly erases previously learned knowledge, resulting in catastrophic forgetting (CF) on past tasks [44, 54]. Effectively addressing CF during CL is essential to preserve previously acquired information. On the other hand, effectively learning new information is equally crucial for CL models to adapt to new tasks and environments.

Existing approaches in CL often face a dilemma: they either prioritize preserving old knowledge or excelling in new task performance, often at the expense of the other. When a CL model prioritizes preserving old knowledge, it focuses on retaining information from previous tasks while minimizing interference or forgetting. However, excessive emphasis on old knowledge can limit the model's ability to adapt to new tasks. Conversely, when a model prioritizes new task performance, it aims to quickly adapt to new tasks or data distributions. Yet, this emphasis on new tasks can potentially degrade performance on previously learned tasks.

To overcome the aforementioned limitations, orthogonal to existing approaches, we introduce the concept of model sensitivity and approach the challenge of balancing old knowledge retention and new task performance in CL from the perspective of model parameter sensitivity. When a CL

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

model exhibits high sensitivity to parameter changes, it leads to two significant issues: (1) *Increased Forgetting*: Excessive sensitivity in model parameters can cause abrupt and substantial changes in model predictions with minor parameter adjustments during CL. This phenomenon results in significant forgetting of previous tasks. (2) *Diminished New Task Performance*: High sensitivity in model parameters can also result in severe overfitting on new tasks. Overfitting occurs when a model memorizes the training data instead of generalizing patterns that can be applied to unseen data. High parameter sensitivity means that even minor alterations in the training data can induce substantial modifications in the learned model. This renders the model excessively tailored to the training data and reduces its adaptability to new, unseen data, consequently leading to suboptimal performance on new tasks.

To reduce the CL model parameter sensitivity under model updates, we aim to ensure that even minor alterations in model parameters do not substantially impair CL model performance. This is accomplished by optimizing the model's performance based on the worst-case scenario of parameter distributions within a distribution neighborhood. However, finding the optimal worst-case CL model parameter distribution is challenging since the space of all possible distributions within the neighborhood is an infinite-dimensional space [32]. To efficiently solve this problem, we parameterize the optimal worst-case CL model parameter distribution as Gaussian distribution. We propose a natural-gradient descent (NGD) method to efficiently inference the mean and covariance of the Gaussian distribution since NGD incorporates the information geometry of the parameter space by adapting the step size based on the curvature of the cost function. This adaptive approach leads to faster convergence compared to conventional gradient descent methods, particularly in high-dimensional spaces where the curvature exhibits notable variations. This is especially beneficial for CL models. However, calculating the natural gradient is computationally expensive due to the explicit calculation of Fisher information matrix (FIM). We thus update the worst-case CL parameters in the expectation parameter space, rather than the traditional natural parameter space, of the Gaussian distribution, thereby eliminating the need for explicit calculation of the FIM.

Our method offers dual benefits: (1) *Reduced Forgetting*: By mitigating parameter sensitivity and avoiding drastic changes in model prediction, our approach effectively reduces the loss of previously learned task knowledge. (2) *Improved New Task Performance*: Through decreased parameter sensitivity, the model becomes less susceptible to overfitting on new task training data. This reduced vulnerability to minor fluctuations fosters the learning of more generalized patterns rather than memorizing specific examples. As a result, the model demonstrates enhanced generalization capabilities on new tasks. Therefore, our method simultaneously achieves superior performance in retaining previously learned knowledge and excelling in new task performance.

We provide a thorough theoretical analysis for our method. Firstly, the theory illustrates that our approach implicitly reduces the variance of loss against different parameter variations, thereby indicating reduced model parameter sensitivity. Secondly, our method tightens the generalization bound of CL models, suggesting enhanced generalization. Furthermore, our extensive experiments across multiple datasets, compared to various state-of-the-art (SOTA) baseline methods, reveal substantial enhancements in overall performance across all learned tasks, backward transfer, and new task test accuracy. These results indicate significantly enhanced CL model ability in preserving old knowledge and achieving better performance on new task with our method. Additionally, our proposed approach seamlessly integrates with existing CL methodologies, functioning as a versatile plug-in. This demonstrates the effectiveness, efficiency, and versatility of our method.

Our contributions can be summarized as follows:

- We tackle the challenge of both retaining old task knowledge and excelling in new task in CL from a novel perspective by mitigating model parameter sensitivity.
- We introduce a novel CL approach aimed at reducing model parameter sensitivity by optimizing CL model performance under the worst-case parameter distribution within a distribution neighborhood. Additionally, we propose an efficient learning algorithm to identify the worst-case parameter distribution.
- We provide comprehensive theoretical analyses that substantiate our method's ability to decrease model parameter sensitivity and improve model generalization.
- Extensive experiments conducted across multiple datasets demonstrate the efficacy and versatility of our proposed method.

2 Related Works

CL aims to learn non-stationary data distributions without forgetting previously learned knowledge. The CL scenarios can be further categorized into three scenarios: task-incremental learning (Task-IL), domain-incremental learning (Domain-IL) and class-incremental learning (Class-IL) [67]. Task-IL and Class-IL are most representative scenarios in CL, we thus focus on these two scenarios. Existing approaches for CL can be categorized into five classes: (1) *regularization-based* methods incorporate regularization terms either in model weights or outputs into the loss function to mitigate catastrophic forgetting when learning new tasks, including [28, 62, 84, 55, 11, 1, 22, 10, 39]; (2) *memory replay-based* methods address the challenge of catastrophic forgetting by explicitly storing and replaying a subset of past experiences (samples from previous tasks) while learning new tasks, including [40, 57, 15, 7, 51, 68, 3, 8, 75, 4, 74, 76, 61, 78, 77, 83, 36, 73, 72]; (3) *gradient-projection-based* methods aim to mitigate catastrophic forgetting by projecting gradient updates onto subspaces that minimize interference with previously learned tasks, including [13, 17, 60, 71, 38, 52, 82]; (4) *architecture-based* methods involve dynamically adapting and modifying the neural network architecture to accommodate new tasks while preserving performance on previously learned tasks, including [41, 63, 34, 23]; (5) *Bayesian-based* methods leverage principles from Bayesian inference to manage the uncertainty and learning of new tasks while preserving knowledge from previous tasks, including [48, 58, 30, 25, 21, 49, 66, 59].

In contrast to existing methods, which often necessitate a trade-off between retaining old knowledge and learning new knowledge, sacrificing one for the other, our approach takes a different path. It sets itself apart from these existing methods by offering an orthogonal solution that preserves old task knowledge while simultaneously enhancing new task performance. This novel perspective is achieved by reducing parameter sensitivity.

Connection with existing flat-minima/SWAD approaches: (1) Connection and difference with sharpness-aware minimization (SAM) [18, 27, 45] related approach: Our method is fundamentally different from SAM-based CL in two aspects. (i) *Deterministic vs. Probabilistic Approach*: SAM uses a fixed deterministic neighborhood, which can be restrictive in practice since updates are constrained within a fixed ball. In contrast, our method employs a probabilistic distributional approach, offering two distinct advantages: (a) The distributional neighborhood is more flexible and covers a broader range of parameter variations by sampling from a neighborhood distribution, and (b) Stochastic Gradient Descent (SGD) introduces noise during CL. Our distributional approach accounts for this noise, making it a more realistic model in practice and providing stronger guarantees against parameter sensitivity. (ii) *Uniform vs. Parameter-specific sensitivity without explicit calculation of FIM*: SAM uniformly updates all parameters, overlooking the varying importance and sensitivity of each parameter in the context of CL. Our method, on the other hand, considers these differences and treats parameters uniquely through the natural gradient without needing to explicitly calculating the FIM. This distinction is crucial for CL, as each parameter has different sensitivity to forgetting—a factor that SAM does not address. (2) Connection and difference with model averaging flatness seeking approach: SWA [24] and SWAD [9], which aim to achieve flatter minima by averaging multiple models during training. However, these approaches are memory-intensive and inefficient for CL, as they require storing multiple sets of model parameters, which compromises memory efficiency.

3 Method

In this section, we first present the preliminary in section 3.1 and then present the model sensitivity aware continual learning in section 3.2.

3.1 Preliminary

Continual Learning Setup The standard CL problem involves learning a sequence of T tasks, represented as $\mathcal{D}^{tr} = \{\mathcal{D}_1^{tr}, \mathcal{D}_2^{tr}, \cdots, \mathcal{D}_T^{tr}\}$. The training dataset $\mathcal{D}_k^{tr}$ for the k^{th} task contains a collection of triplets: $(\boldsymbol{x}_i^k, y_i^k)_{i=1}^{n_k}$, where $\boldsymbol{x}_i^k$ denotes the i^{th} data example specific to task k, y_i^k represents the associated data label for $\boldsymbol{x}_i^k$. The primary objective is to train a neural network function, parameterized by $\boldsymbol{\theta}$, denoted as $g_{\boldsymbol{\theta}}(\boldsymbol{x})$. The goal is to achieve good performance on the test datasets from all the learned tasks, represented as $\mathcal{D}^{te} = \{\mathcal{D}_1^{te}, \mathcal{D}_2^{te}, \cdots, \mathcal{D}_T^{te}\}$, while ensuring that knowledge

acquired from previous tasks is not forgotten. The CL loss function is defined as the following:

$$\mathcal{L}^{CL}(\boldsymbol{\theta}) := \mathcal{L}_{CE}(\boldsymbol{x}, y; \boldsymbol{\theta}) + \zeta \mathcal{L}_f(\boldsymbol{\theta}) \tag{1}$$

where $\mathcal{L}_{CE}(\boldsymbol{x}, y; \boldsymbol{\theta})$ is the current task cross-entropy loss function. $\mathcal{L}_f(\boldsymbol{\theta})$ is the forgetting-mitigation loss, e.g., memory-replay, weight-regularization and gradient-projection loss, etc. ζ is a constant that balances the weight between the loss of the new task and the loss of the previous tasks.

Exponential Family of Distributions The exponential family distribution [70] is defined as:

$$P_{\boldsymbol{\phi}}(\boldsymbol{\theta}) := h(\boldsymbol{\theta}) exp(\langle \boldsymbol{\phi}, \boldsymbol{\Omega}(\boldsymbol{\theta}) \rangle - Z(\boldsymbol{\phi})) \tag{2}$$

Where $:=$ denotes a definition. In existing literature [70], $\boldsymbol{\phi}$ are called the natural parameters for defining the distribution, $P_{\boldsymbol{\phi}}(\boldsymbol{\theta})$. $h(\boldsymbol{\theta})$ is the base measure, $\boldsymbol{\Omega}(\boldsymbol{\theta})$ is the sufficient statistic, $Z(\boldsymbol{\phi}) := \log \int h(\boldsymbol{\theta}) exp(\langle \boldsymbol{\phi}, \boldsymbol{\Omega}(\boldsymbol{\theta}) \rangle) d\boldsymbol{\theta}$ is the log-partition function, $\langle , \rangle$ denotes the dot product between two vectors. We denote the expectation parameters as $\boldsymbol{\lambda} := \mathbb{E}_{P_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \boldsymbol{\Omega}(\boldsymbol{\theta})$. We can write multivariate Gaussian distribution as canonical form of exponential family as:

$$f(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) := \frac{1}{(2\pi)^{\frac{d}{2}} det(\boldsymbol{\Sigma})^{\frac{1}{2}}} exp\{-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu})\} \tag{3}$$

$$= exp\{\boldsymbol{\theta}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} - \frac{1}{2} \boldsymbol{\theta}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\theta} - \frac{1}{2}[d \log 2\pi + \log |\boldsymbol{\Sigma}| + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}]\} \tag{4}$$

Therefore, the correspondence between $f(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and exponential family distribution in Eq.(2) can be expressed as the following:

$$\boldsymbol{\phi} := (\boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}, -\frac{1}{2} \boldsymbol{\Sigma}^{-1}), \quad \boldsymbol{\Omega}(\boldsymbol{\theta}) := (\boldsymbol{\theta}, \boldsymbol{\theta}\boldsymbol{\theta}^T) \tag{5}$$

$$\boldsymbol{\lambda}^1 := \mathbb{E}_{f(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})} \boldsymbol{\theta} = \boldsymbol{\mu}, \quad \boldsymbol{\lambda}^2 := \mathbb{E}_{f(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})} \boldsymbol{\theta}\boldsymbol{\theta}^T = \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma} \tag{6}$$

Derivations details of Eq.(6) can be found in Appendix B.1. In the following section, we use exponential family distributions to parameterize the worst-case of CL model parameter distribution since this enables us to efficiently calculate the natural gradient in the expectation parameter space $\boldsymbol{\lambda}$ without needing to explicitly calculate the Fisher information matrix (FIM) in natural parameter space $\boldsymbol{\phi}$.

3.2 Model Sensitivity Aware Continual Learning

Learning Objective Specifically, we propose the following CL learning objective to reduce the CL parameter sensitivity under model parameter updates:

$$\min_{\boldsymbol{\mu}} \max_{\mathbb{U} \in \mathcal{U}} \mathbb{E}_{\boldsymbol{\theta} \sim \mathbb{U}(\boldsymbol{\theta})} \mathcal{L}^{CL}(\boldsymbol{\theta}) \tag{7}$$
$$\text{s.t. } \mathcal{U} = \{\mathbb{U} : D_{\mathrm{KL}}(\mathbb{U}, \mathbb{V}) \leq \epsilon\}$$

where $\mathcal{U}$ denotes the uncertainty set. $D_{\mathrm{KL}}(\mathbb{U}, \mathbb{V})$ denotes the KL divergence between the current CL model parameter distribution $\mathbb{V}$ and the neighbour CL model parameter distribution $\mathbb{U}$. ϵ is a small constant. $\max_{\mathbb{U} \in \mathcal{U}} \mathbb{E}_{\boldsymbol{\theta} \sim \mathbb{U}(\boldsymbol{\theta})} \mathcal{L}^{CL}(\boldsymbol{\theta})$ aims to find the worst-case CL model parameter distribution within a neighbourhood. We choose probabilistic distributional neighbourhood due to two-fold reasons: (1) the distributional neighbourhood covers more flexible parameter space; and (2) widely used SGD method incurs update noise during CL, thereby distributional neighbourhood provides stronger guarantee against parameter sensitivity. It is important to note that the outer minimization is performed with respect to $\boldsymbol{\mu}$, the expectation of $\boldsymbol{\theta}$, since during inference, only $\boldsymbol{\mu}$ is used as the model parameter for predictions.

Objective for Learning the Worst-Case CL Parameter Distribution We convert the constrained inner maximization optimization in Eq. (7) into the following unconstrained optimization to find the worst-case CL model parameter distribution.

$$\arg\min_{\mathbb{U}} [H(\mathbb{U}) := -\mathbb{E}_{\boldsymbol{\theta} \sim \mathbb{U}(\boldsymbol{\theta})} \mathcal{L}^{CL}(\boldsymbol{\theta}) + \alpha D_{\mathrm{KL}}(\mathbb{U}, \mathbb{V})] \tag{8}$$

where $\alpha > 0$ is a constant. However, solving Eq. (8) is intractable since the optimization target is in an infinite-dimensional function space [32]. For computation efficiency, we set the current CL

model parameter distribution as $\mathbb{V}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, where $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ denote the mean vector and covariance matrix, respectively. We set the neighbourhood distribution as $\mathbb{U}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the mean vector and covariance matrix, respectively. To further improve computational efficiency, we constrain the covariance matrix to be diagonal matrix, i.e., $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma}^2)$ and $\boldsymbol{\Sigma}_0 = \text{diag}(\boldsymbol{\rho}^2)$. We denote the density function of $\mathbb{U}(\boldsymbol{\theta})$ and $\mathbb{V}(\boldsymbol{\theta})$ as $u(\boldsymbol{\theta})$ and $v(\boldsymbol{\theta})$, respectively. We express the loss function in Eq. (8) as the following:

$$H(\mathbb{U}) = \mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) := -\mathcal{L}^{CL}(\boldsymbol{\theta}) + \alpha[\log u(\boldsymbol{\theta}) - \log v(\boldsymbol{\theta})]] \tag{9}$$

By parameterizing the distribution $\mathbb{U}$ as exponential family distribution in Eq. (4), our goal is to learn the parameters $\boldsymbol{\phi}$ in Eq. (5) with natural gradient descent (NGD) [42] as the following equation:

$$\boldsymbol{\phi}_{i+1} = \boldsymbol{\phi}_i - \eta \boldsymbol{F}^{-1} \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}_i) \tag{10}$$

where $\boldsymbol{F}$ is the FIM. We opt for NGD because it adjusts the step size according to the curvature of the cost function, making convergence faster than traditional gradient descent methods. This is especially advantageous in high-dimensional spaces where the curvature and parameter-wise sensitivity vary significantly, benefiting CL models. However, computing the natural gradient is computationally intensive due to the need to calculate the FIM. To address this, we develop an efficient update method in the dual space, specifically the expectation parameter space $\boldsymbol{\lambda}$, rather than the natural parameter space $\boldsymbol{\phi}$, eliminating the need for explicit FIM calculation. In the following, we will use $\mathcal{L}(\boldsymbol{\phi})$ and $\mathcal{L}(\boldsymbol{\lambda})$ interchangeably, as they represent the same loss function only parameterized in different spaces. We leverage the relation between NGD in natural parameter space and gradient descent in expectation parameter space (in Appendix A.1), NGD can be performed without explicitly computing the FIM. This update in its dual space leads to significantly more efficient parameter updates and promising computational advantages.

NGD for Efficiently Finding the Worst-Case Gaussian Distribution In the following, we present specific algorithms for updating the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ with NGD to find the worst-case Gaussian distribution, i.e., $\mathbb{U}^* := \arg\min_{\mathbb{U}} H(\mathbb{U})$. We can get the following updates for mean $\boldsymbol{\mu}$ and diagonal covariance $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma}^2)$ (detailed derivations can be found in Appendix B):

$$\boldsymbol{\mu}_{i+1} = \boldsymbol{\mu}_i + \eta \boldsymbol{\Sigma}_{i+1}[\nabla_{\boldsymbol{\theta}} \mathcal{L}^{CL}(\boldsymbol{\theta}) - \alpha(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)\boldsymbol{\Sigma}_0^{-1}] \tag{11}$$

$$\boldsymbol{\Sigma}_{i+1}^{-1} = (1 - \eta\alpha)\boldsymbol{\Sigma}_i^{-1} + \eta[-\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \mathcal{L}^{CL}(\boldsymbol{\theta}) + \alpha\boldsymbol{\Sigma}_0^{-1}] \tag{12}$$

By plug-in $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma}^2)$ and $\boldsymbol{\Sigma}_0 = \text{diag}(\boldsymbol{\rho}^2)$ into the above equations, we can obtain the following updates:

$$\boldsymbol{\mu}_{i+1} = \boldsymbol{\mu}_i + \eta \boldsymbol{\sigma}^2_{i+1}[\nabla_{\boldsymbol{\theta}} \mathcal{L}^{CL}(\boldsymbol{\theta}_i) - \alpha(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)\boldsymbol{\rho}^{-2}] \tag{13}$$

$$\boldsymbol{\sigma}_{i+1}^{-2} = (1 - \eta\alpha)\boldsymbol{\sigma}_i^{-2} + \eta[-\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \mathcal{L}^{CL}(\boldsymbol{\theta}_i) + \alpha\boldsymbol{\rho}^{-2}] \tag{14}$$

In practice, we set $\alpha = 1.0$ to reduce the reliance on hyperparameters. However, computing the diagonal Hessian matrix $\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \mathcal{L}^{CL}(\boldsymbol{\theta})$ in Eq. (14) is a computationally challenging task. Following [42], we efficiently approximate the Hessian as the following:

$$\nabla^2_{\boldsymbol{\theta}^k\boldsymbol{\theta}^k} \mathcal{L}^{CL}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{j=1}^{j=N} [\nabla_{\boldsymbol{\theta}^k} \mathcal{L}_j^{CL}(\boldsymbol{\theta})]^2 \tag{15}$$

where N is the number of training data points for the current task, $\mathcal{L}_j^{CL}(\boldsymbol{\theta})$ denotes the loss function for the data point j, $\boldsymbol{\theta}^k$ denotes the k^{th} element of the model parameter $\boldsymbol{\theta}$. It is crucial to note that this Hessian approximation is computed only once after learning each task and involves calculating only the diagonal elements, i.e., $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma}^2)$. As a result, the overall computational cost throughout the continual learning process remains low. Additionally, this update mechanism maintains the same number of learnable parameters as existing methods, ensuring fair comparisons. This is because, during the learning of each task, only the mean parameters of the Gaussian distribution are updated.

Learning Algorithm We name our method as **M**odel sensitivity **A**ware **C**ontinual **L**earning (**MACL**). The detailed algorithm is present in Algorithm 1.

Algorithm 1 Model Sensitivity Aware Continual Learning

REQUIRE: model parameters $\boldsymbol{\theta}$, CL model learning rate β, worst-case Gaussian learning rate η, number of CL tasks T, number of CL steps K for each task, distribution neighbourhood regularization strengths $\alpha = 1.0$. Randomly initialized diagonal covariance matrix, i.e., diag$(\boldsymbol{\sigma}^2)$.
for $n = 1$ to T **do**
  for $i = 1$ to K **do**
    calculate the CL loss function according to Eq. (1)
    update the worst-case Gaussian mean $\boldsymbol{\mu}$ (i.e., $\boldsymbol{\theta}$) by $\boldsymbol{\theta}_i' = \boldsymbol{\theta}_i + \eta\boldsymbol{\sigma}_n^2[\nabla_{\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}_i) - (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0)\boldsymbol{\rho}^{-2}]$
    sample parameters from the worst-case CL model parameter distribution. $\boldsymbol{\theta}' = \boldsymbol{\theta}_i' + \boldsymbol{\sigma}_n\boldsymbol{\zeta}$, where $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    update CL model parameters $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}' - \beta\nabla_{\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}')$
  end for
  update the worst-case Gaussian covariance $\boldsymbol{\sigma}$ by $\boldsymbol{\sigma}_{n+1}^{-2} = (1-\eta)\boldsymbol{\sigma}_n^{-2} + \eta[-\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \boldsymbol{\rho}^{-2}]$
  where the Hessian is calculated by $\nabla^2_{\boldsymbol{\theta}^k\boldsymbol{\theta}^k}\mathcal{L}^{CL}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{j=1}^{j=N}[\nabla_{\boldsymbol{\theta}^k}\mathcal{L}_j^{CL}(\boldsymbol{\theta})]^2$ according to Eq. (15)
end for

4 Theoretical Analysis

In this section, we build the theoretical connection between MACL and parameter sensitivity in Theorem 4.2 and the generalization analysis in Theorem 4.3. Due to the space limitations, we provide the theorem proof in Appendix A.2. Let's first look at the inner maximization problem in Eq. (7).

$$\max_{\mathbb{U}\in\mathcal{U}} \int \mathcal{L}^{CL}(\boldsymbol{\theta}) d\mathbb{U}(\boldsymbol{\theta}), \quad \text{s.t. } \mathcal{U} = \{\mathbb{U} : D_{\mathrm{KL}}(\mathbb{U}, \mathbb{V}) \leq \epsilon\} \tag{16}$$

Lemma 4.1. $D_{\mathrm{KL}}(\mathbb{U}, \mathbb{V}) = \int u(\boldsymbol{\theta}) \log(\frac{u(\boldsymbol{\theta})}{v(\boldsymbol{\theta})}) d\boldsymbol{\theta} \leq \int \frac{(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))^2}{v(\boldsymbol{\theta})} d\boldsymbol{\theta}$

Theorem 4.2. *Assume* $\int ||\frac{1}{v(\boldsymbol{\theta})}||_\infty d\boldsymbol{\theta} \leq M$*, we can obtain the following conclusion for Eq. (16):*

$$\max_{\mathbb{U}\in\mathcal{U}} \int \mathcal{L}^{CL}(\boldsymbol{\theta}) d\mathbb{U}(\boldsymbol{\theta}) = \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} + \sqrt{\frac{\epsilon\mathbb{E}(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2}{M}} \tag{17}$$

where $\overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} := \int \mathcal{L}^{CL}(\boldsymbol{\theta}) d\mathbb{V}(\boldsymbol{\theta})$. $Var(\mathcal{L}^{CL}(\boldsymbol{\theta}))$ denotes the variance of $\mathcal{L}^{CL}(\boldsymbol{\theta})$ with respect to different model parameters variations, i.e., $Var(\mathcal{L}^{CL}(\boldsymbol{\theta})) = \mathbb{E}(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2 = \int(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2 d\boldsymbol{\theta}$. In this context, $Var(\mathcal{L}^{CL}(\boldsymbol{\theta}))$ serves as a measure of the CL model's sensitivity to parameter updates. Essentially, a smaller loss variance indicates lower parameter sensitivity in the CL model. However, directly optimizing the loss variance within the parameter distribution neighborhood is impractical, as it requires computing the loss variation across a large number of different sets of CL model parameters and training data points. In contrast, our method (MACL) offers an efficient and effective alternative. MACL implicitly minimizes the loss variance across different model parameter variations by optimizing CL performance solely on the worst-case CL model parameter distribution. In the following, inspired by UDIL [64], we further provide the following generalization bound for CL:

Theorem 4.3 (Generalization bound of MACL)**.** *Let q be the number of CL model parameters and n be the number of training data points. The CL loss $\mathcal{L}^{CL}(\boldsymbol{\theta}) \leq C$ (C is a constant). With high probability of $1-\delta$, the following bound holds:*

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \sum_i^{i=T} \mathcal{L}_{\mathcal{D}_i}^{CL}(\boldsymbol{\theta}) \leq \max_{\mathbb{U}\in\mathcal{U}} \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}} \mathcal{L}^{CL}(\boldsymbol{\theta}) + \frac{C}{N_T + \zeta\sum_{i=1}^{i=T-1} N_i} + \sqrt{\frac{\tau^2(\sqrt{q} + \sqrt{2\log(N_T + \zeta\sum_{i=1}^{i=T-1} N_i)})^2 + R + 2\log(\frac{N_T + \zeta\sum_{i=1}^{i=T-1} N_i}{\delta})}{4(N_T + \zeta\sum_{i=1}^{i=T-1} N_i - 1)}} \tag{18}$$

Where τ is a constant. We denote the number of data examples for task $1, \cdots, T-1$ in the memory buffer $\mathcal{M}$ during training on task T as $N_1, N_2, \cdots, N_{T-1}$ when using memory replay based approach or the number of training data points when using regularization based approach. $\mathcal{L}_{\mathcal{D}_i}^{CL}(\boldsymbol{\theta})$ denotes the CL loss on the data from data distribution $\mathcal{D}_i$ of task i (generalization error), i.e., it is defined as: $\mathcal{L}_{\mathcal{D}_i}^{CL}(\boldsymbol{\theta}) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\mathcal{L}(\boldsymbol{x}, y, \boldsymbol{\theta})$. $\mathcal{L}^{CL}(\boldsymbol{\theta})$ denotes the empirical CL loss as Eq. (1). $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the CL model parameter posterior distribution parameterized with Gaussian distribution.

Generalization bound implication: (1) When using a memory-replay approach, the number of samples from new tasks often exceeds the number of samples in the memory buffer, causing data imbalance. This imbalance, where fewer samples from previous tasks are stored, affects the second and third terms in the generalization bound. The bound suggests that as the number of samples in the memory buffer increases (i.e., $\sum_{i=1}^{i=T-1} N_i \uparrow$), these terms tighten, leading to a tighter generalization upper bound. This is because $\lim_{x\to\infty}[h(x) := \frac{\log x}{x}] = 0$, meaning the generalization improves with a larger buffer, aligning with the intuition that more memory buffer data leads to better performance. (2) In the regularization-based approach, $\zeta \sum_{i=1}^{i=T-1} N_i$ is treated as the effective sample size for previous tasks since the loss is approximated in the absence of earlier data. The parameter ζ controls the trade-off between learning the new task and retaining knowledge from past tasks. A larger ζ increases regularization, preventing the model from deviating too much from the parameters learned on previous tasks. This leads to higher empirical loss on the new task (first term), but tighter bounds (second and third terms), indicating that knowledge from previous tasks is retained effectively. This prioritizes stability over learning flexibility for the new task.

5 Experiments

5.1 Setup

Datasets We conduct experiments on several datasets, including CIFAR10 (10 classes), CIFAR100 (100 classes) [29], and Tiny-ImageNet (200 classes) [80], to assess the effectiveness of MACL in task incremental learning (Task-IL) and class incremental learning (Class-IL). In addition, we also conduct experiments on 5-dataset [79, 5], CUB200 [69] and ImageNet-R [20] (in Appendix). Following the approach in [7], we split the CIFAR-10 dataset into five tasks, each with two distinct classes. We divided the CIFAR-100 dataset into ten tasks, each containing ten classes. We split the Tiny-ImageNet dataset into ten tasks, each comprising twenty classes. More dataset statistics can be found in Appendix E.1.

Baselines We compare to the following various SOTA CL methods. (1) Regularization-based methods, including oEWC [62], synaptic intelligence (SI) [84], Learning without Forgetting (LwF) [35], Classifier-Projection Regularization (CPR) [10]. (2) Bayesian-based methods, including NCL [25]. (3) Architecture-based methods, including HAT [63]. (4) Memory-based methods, including ER [15], A-GEM [14], iCaRL[55], GSS [2], HAL [12], DER++ [7], ER-ACE [8] and LODE [36]. (5) Gradient-projection-based methods: Gradient Projection Memory (GPM) [60].

Implementation Details Following [7], we use ResNet18 [19] as the backbone network for all the CL datasets and compared baseline methods. For the baselines that are included in the open-source code of DER++ [7], we use the same hyperparameters provided in DER++ [7] for the compared methods. For the baselines not included in the open-source code of DER++, e.g., GPM, LODE, etc, we use the open-source code from their original paper for comparisons. For the hyperparameters in our method, we set $\alpha = 1.0$ across all the datasets to minimize the model's dependence on hyperparameters. For η, we set $\eta = 1e-5$ for CIFAR10 and CIFAR100, and $\eta = 1e-6$ for Tiny-ImageNet. The η is selected from the range of $[1e-4, 1e-5, 1e-6, 1e-7]$. Following [7, 14], the hyperparameter is determined through the validation sets split from the training sets from the first three tasks. Similar to [7], we train all the CL models using the standard SGD optimizer to update the CL model. The batch size and replay buffer batch size are set to 32. We use a single NVIDIA A5000 GPU with 24GB memory to run the experiments. Each experiment result is averaged for 10 runs with mean and standard deviation.

5.2 Results

We evaluate the performance of different CL methods with (1) overall accuracy; (2) new task accuracy; and (3) backward transfer in the following.

Overall Accuracy (ACC) ACC is the average accuracy across the entire task sequence. We present the results on CIFAR10, CIFAR-100 and Tiny-ImageNet in Table 1. We can observe that our method substantially improve over various SOTA baseline methods up to 3% to 4% on CIFAR100, TinyImageNet by integrating MACL with existing CL methods. This overall performance improvement is attributed to the reduced parameter sensitivity.

Table 1: **Task-IL and class-IL** overall accuracy on CIFAR10, CIFAR-100 and Tiny-ImageNet, respectively with memory size 500. '—' indicates not applicable/available.

Method	CIFAR-10		CIFAR-100		Tiny-ImageNet	
	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
fine-tuning	19.62 ± 0.05	61.02 ± 3.33	9.29 ± 0.33	33.78 ± 0.42	7.92 ± 0.26	18.31 ± 0.68
Joint train	92.20 ± 0.15	98.31 ± 0.12	71.32 ± 0.21	91.31 ± 0.17	59.99 ± 0.19	82.04 ± 0.10
SI	19.48 ± 0.17	68.05 ± 5.91	9.41 ± 0.24	31.08 ± 1.65	6.58 ± 0.31	36.32 ± 0.13
LwF	19.61 ± 0.05	63.29 ± 2.35	9.70 ± 0.23	28.07 ± 1.96	8.46 ± 0.22	15.85 ± 0.58
NCL	19.53 ± 0.32	64.49 ± 4.06	8.12 ± 0.28	20.92 ± 2.32	7.56 ± 0.36	16.29 ± 0.87
GPM	—	90.68 ± 3.29	—	72.48 ± 0.40	—	—
UCB	—	79.28 ± 1.87	—	57.15 ± 1.67	—	—
HAT	—	92.56 ± 0.78	—	72.06 ± 0.50	—	—
A-GEM	22.67 ± 0.57	89.48 ± 1.45	9.30 ± 0.32	48.06 ± 0.57	8.06 ± 0.04	25.33 ± 0.49
GSS	49.73 ± 4.78	91.02 ± 1.57	13.60 ± 2.98	57.50 ± 1.93	—	—
HAL	41.79 ± 4.46	84.54 ± 2.36	9.05 ± 2.76	42.94 ± 1.80	—	—
oEWC	19.49 ± 0.12	64.31 ± 4.31	8.24 ± 0.21	21.2 ± 2.08	7.42 ± 0.31	15.19 ± 0.82
oEWC+MACL	$\mathbf{20.55 \pm 0.71}$	$\mathbf{66.95 \pm 2.46}$	$\mathbf{8.82 \pm 0.50}$	$\mathbf{23.42 \pm 1.93}$	$\mathbf{7.86 \pm 0.23}$	$\mathbf{17.43 \pm 0.93}$
CPR(EWC)	19.61 ± 3.67	65.23 ± 3.87	8.42 ± 0.37	21.43 ± 2.57	7.67 ± 0.23	15.58 ± 0.91
CPR(EWC)+MACL	$\mathbf{20.58 \pm 2.56}$	$\mathbf{67.28 \pm 3.75}$	$\mathbf{9.15 \pm 0.63}$	$\mathbf{22.87 \pm 1.78}$	$\mathbf{8.10 \pm 0.49}$	$\mathbf{17.96 \pm 0.82}$
GPM	—	—	—	72.48 ± 0.40	—	30.72 ± 0.27
GPM+MACL	—	—	—	$\mathbf{74.51 \pm 0.36}$	—	$\mathbf{35.06 \pm 0.38}$
iCaRL	—	—	44.16 ± 1.53	84.06 ± 0.42	23.71 ± 0.23	59.24 ± 0.16
iCaRL+MACL	—	—	$\mathbf{48.27 \pm 0.95}$	$\mathbf{84.55 \pm 0.51}$	$\mathbf{24.18 \pm 0.58}$	$\mathbf{59.45 \pm 0.32}$
ER	57.74 ± 0.27	93.61 ± 0.27	20.98 ± 0.35	73.37 ± 0.43	$\mathbf{9.99 \pm 0.29}$	48.64 ± 0.46
ER+MACL	$\mathbf{63.74} \pm 1.24$	$\mathbf{93.78 \pm 0.36}$	$\mathbf{22.18 \pm 0.27}$	$\mathbf{74.87 \pm 0.51}$	9.87 ± 0.15	$\mathbf{51.25 \pm 0.37}$
DER++	72.70 ± 1.36	93.88 ± 0.50	36.37 ± 0.85	75.64 ± 0.60	19.38 ± 1.41	51.91 ± 0.68
DER+++MACL	$\mathbf{74.53 \pm 0.79}$	$\mathbf{94.72 \pm 0.65}$	$\mathbf{39.42 \pm 0.82}$	$\mathbf{77.53 \pm 0.89}$	$\mathbf{20.17 \pm 1.56}$	$\mathbf{54.03 \pm 0.79}$
ER-ACE	71.83 ± 1.42	94.12 ± 0.61	37.05 ± 0.36	75.97 ± 0.69	20.43 ± 0.97	52.59 ± 0.75
ER-ACE+MACL	$\mathbf{73.21 \pm 0.96}$	$\mathbf{94.98 \pm 0.72}$	$\mathbf{40.28 \pm 0.39}$	$\mathbf{77.65 \pm 0.76}$	$\mathbf{21.89 \pm 0.83}$	$\mathbf{53.95 \pm 0.78}$
LODE	75.45 ± 0.90	$\mathbf{94.41 \pm 0.22}$	38.95 ± 0.93	78.92 ± 0.67	19.87 ± 0.72	60.18 ± 0.65
LODE+MACL	$\mathbf{76.41 \pm 0.67}$	94.32 ± 0.24	$\mathbf{40.67 \pm 0.89}$	$\mathbf{40.03 \pm 0.51}$	$\mathbf{21.09 \pm 0.97}$	$\mathbf{61.79 \pm 0.86}$

New Task Accuracy To evaluate the new task performance of the proposed CL method, we evaluate the new task performance during CL by integrating MACL with DER++ and GPM in Figure 1. The results show that MACL can significantly improves the new task performance for different CL methods, indicating that reducing the model parameter sensitivity is beneficial to improve new task performance during CL.

Backward Transfer Backward transfer (BWT) quantifies the degree of forgetting observed on previously learned tasks. When BWT > 0, it indicates that learning the current new task positively influences the performance on previously learned tasks. Conversely, when BWT ≤ 0, it signals that learning the current new task may result in forgetting previously acquired knowledge. We evaluate BWT in Table 2. We can observe that our method significantly improves BWT by up to 5% through integrating MACL with existing CL methods. This indicates that reducing parameter sensitivity can substantially reduce forgetting on previously learned knowledge. These empirical analysis also verify our theoretical analysis that our method implicitly improves the stability by reducing loss variance.

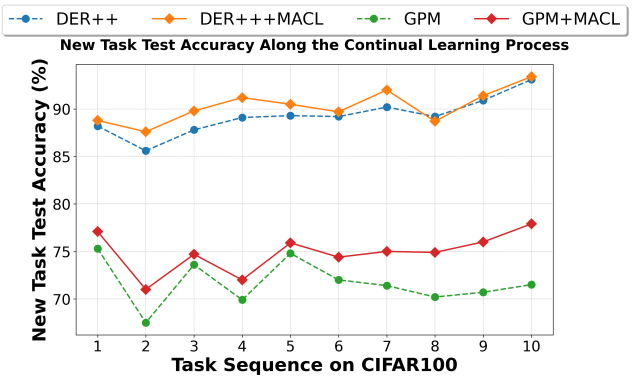

Figure 1: new task performance during CL.

Table 2: **Backward Transfer** of different CL methods with memory size 500.

Method	CIFAR10		CIFAR100		Tiny-ImageNet	
	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
finetuning	-96.39 ± 0.12	-46.24 ± 2.12	-89.68 ± 0.96	-62.46 ± 0.78	-78.94 ± 0.81	-67.34 ± 0.79
AGEM	-94.01 ± 1.16	-14.26 ± 1.18	-88.5 ± 1.56	-45.43 ± 2.32	-78.03 ± 0.78	-59.28 ± 1.08
GSS	-62.88 ± 2.67	-7.73 ± 3.99	-82.17 ± 4.16	-33.98 ± 1.54	—	—
HAL	-62.21 ± 4.34	-5.41 ± 1.10	-49.29 ± 2.82	-13.60 ± 1.04	—	—
ER	-45.35 ± 0.07	-3.54 ± 0.35	-74.84 ± 1.38	-16.81 ± 0.97	**-75.24 ± 0.76**	-31.98 ± 1.35
ER+MACL	**-34.43 ± 0.82**	**-3.31 ± 0.32**	**-73.17 ± 0.69**	**-15.73 ± 0.78**	-75.29 ± 0.37	**-29.32 ± 0.42**
DER++	-22.38 ± 4.41	-4.66 ± 1.15	-53.89 ± 1.85	-14.72 ± 0.96	-64.6 ± 0.56	-27.21 ± 1.23
DER++ MACL	**-21.87 ± 1.67**	**-3.09 ± 1.31**	**-48.62 ± 1.56**	**-13.62 ± 0.35**	**-62.23 ± 0.78**	**-27.10 ± 0.43**
ER-ACE	-13.64 ± 0.95	-3.28 ± 0.83	-39.51 ± 1.23	-14.57 ± 0.39	-46.07 ± 0.83	**-28.35 ± 0.16**
ER-ACE+MACL	**-12.76 ± 1.23**	**-3.15 ± 0.57**	**-33.86 ± 1.37**	**-13.89 ± 0.57**	**-42.29 ± 0.50**	-28.41 ± 0.23
LODE	-16.37 ± 0.67	**-2.93 ± 0.19**	-53.23 ± 1.72	-15.24 ± 0.76	-55.89 ± 0.98	-19.13 ± 0.56
LODE+MACL	**-16.25 ± 0.73**	-3.16 ± 0.45	**-52.67 ± 1.35**	**-15.11 ± 0.53**	**-55.61 ± 1.15**	**-18.17 ± 0.83**

5.3 Ablation Study

Hyperparameter Analysis We evaluate the sensitivity of the hyperparameters η in Table 5 in Appendix D.1. Our observations indicate that when parameter sensitivity is not reduced, i.e., $\eta = 0$, the CL model performs poorly. As we gradually increase the reduction of parameter sensitivity, the CL model's performance improves. This improvement is because appropriately reducing parameter sensitivity helps mitigate forgetting and enhances learning for new tasks, thus boosting overall CL performance. However, if the reduction in parameter sensitivity is increased excessively, the model's performance deteriorates. This is because an overly constrained model, while minimizing forgetting, struggles to learn new tasks effectively, resulting in worse performance.

Effect of Memory Size To assess the impact of varying memory buffer sizes, we present the results in Table 3. The results demonstrate that compared to different baseline methods, our MACL plug-in also enhances the performance of baseline methods with a memory size of 2000.

Table 3: **Task-IL and class-IL** overall accuracy on CIFAR-100 and Tiny-ImageNet, respectively with memory size 2000.

Algorithm	CIFAR-100		Tiny-ImageNet	
Method	Class-IL	Task-IL	Class-IL	Task-IL
ER	36.06 ± 0.72	81.09 ± 0.45	15.16 ± 0.78	58.19 ± 0.69
ER+MACL	**37.83 ± 0.94**	**83.37 ± 1.35**	**17.08 ± 0.73**	**59.51 ± 0.53**
DER++	50.72 ± 0.71	82.43 ± 0.38	24.21 ± 1.09	62.22 ± 0.87
DER+++MACL	**52.79 ± 0.85**	**84.07 ± 0.79**	**27.55 ± 1.43**	**64.28 ± 0.95**
LODE	54.32 ± 0.56	85.79 ± 0.67	31.03 ± 1.27	**70.05 ± 0.59**
LODE+MACL	**54.76 ± 0.68**	**86.53 ± 0.58**	**32.16 ± 1.12**	69.79 ± 0.53

Benefit of NGD To evaluate the benefits of using NGD over gradient descent (GD) for calculating the worst-case Gaussian distribution, we present comparison results in Table 6 in Appendix D.2. The results show that NGD outperforms GD because NGD better captures parameter importance, which helps preserve old knowledge while effectively adapting to new tasks.

Efficiency Evaluation To assess the efficiency of our proposed method, we compare the running time of integration of different CL methods with MACL and corresponding CL methods alone on CIFAR100, as shown in Table 15 in Appendix D.8. The results indicate that incorporating MACL increases the computational cost by only 55% to 61% compared to the corresponding CL methods alone. This demonstrates the high efficiency of our method, as it introduces only small additional training cost.

Effect of Different Architectures To evaluate the impact of different architectures, we compared various approaches using both ViT and ResNet32. For the ResNet32 experiments, we followed the setup in [85], integrating MACL with MEMO [86] and comparing it to MEMO alone, using a memory buffer size of 2000 on CIFAR100. Additionally, we conducted experiments with a pre-trained Vision Transformer (ViT) [16], specifically the vit-base-patch16-224 model pre-trained on ImageNet1K. On CIFAR100, we integrated MACL with DER++, using a memory size of 500, and demonstrated that using a pre-trained ViT significantly improves CL performance. Moreover, combining MACL with DER++ further enhances CL performance with the pre-trained ViT. The results are presented in the Appendix.

Long Task Sequence To assess the effectiveness of the proposed approach across varying task lengths, we conducted experiments by splitting Tiny-ImageNet into sequences of 10 and 20 tasks. The Task-IL and Class-IL results for integrating DER++ with MACL, using a memory buffer size of 500, are presented in Table 4. These results demonstrate that even with longer task sequences, our method still significantly outperforms DER++.

Table 4: Overall accuracy of integrating DER++ with MACL using a memory buffer of 500 and longer task sequence on Tiny-ImageNet.

number of tasks	10	20
Class-IL	19.38 ± 1.41	15.02 ± 0.53
Class-IL+ MACL	$\mathbf{20.17 \pm 1.56}$	$\mathbf{16.08 \pm 0.81}$
Task-IL	51.91 ± 0.68	51.65 ± 1.36
Task-IL + MACL	$\mathbf{54.03 \pm 0.79}$	$\mathbf{54.96 \pm 0.72}$

Online CL Under the online CL setting, we evaluate the effectiveness of the proposed approach on CIFAR100 and Tiny-ImageNet by comparing with MKD [46] and PCR [37]. The results are put in the Appendix.

5-datasets results To assess the effectiveness of MACL on the 5-Datasets benchmark [79, 5], which includes CIFAR-10, MNIST [33], Fashion-MNIST [81], SVHN [47], and notMNIST [6], we conducted experiments. This dataset provides a diverse range of CL tasks. We performed experiments on 5-Datasets, using a memory buffer size of 500, with MACL. The detailed results are provided in the Appendix.

ImageNet-R and CUB200 results We further evaluate the effectiveness of MACL on CUB200 [69] and ImageNet-R [20], the results are shown in the Appendix.

6 Conclusion

In this paper, we address the challenge of balancing learning new tasks while preserving knowledge from previous ones in continual learning. We propose a model sensitivity-aware continual learning method that enhances both the model's ability to retain old knowledge and improve performance on new tasks. Specifically, our goal is to reduce model parameter sensitivity by optimizing CL performance for the worst-case parameter distribution within the neighborhood of the current model's parameter distribution. This approach improves stability in preserving old knowledge and mitigates overfitting on new tasks. We provide a comprehensive theoretical analysis of the proposed method, and extensive experiments on multiple datasets demonstrate its effectiveness, efficiency, and versatility.

Limitation Discussion Our method introduces additional training cost compared to existing continual learning approaches.

Broader Impacts

Our work advances continual learning, which is beneficial to develop more adaptable and efficient AI. Our work has no negative societal impacts.

Acknowledgments

This work was partially supported by NSF IIS 2347592, 2347604, 2348159, 2348169, DBI 2405416, CCF 2348306, CNS 2347617.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision*, pages 139–154, 2018.

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems 30*, 2019.

[3] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022.

[4] Lorenzo Bonicelli, Matteo Boschini, Angelo Porrello, Concetto Spampinato, and Simone Calderara. On the effectiveness of lipschitz-driven rehearsal in continual learning. *Advances in Neural Information Processing Systems*, 35:31886–31901, 2022.

[5] Jorg Bornschein, Alexandre Galashov, Ross Hemsley, Amal Rannen-Triki, Yutian Chen, Arslan Chaudhry, Xu Owen He, Arthur Douillard, Massimo Caccia, Qixuan Feng, et al. Nevis' 22: A stream of 100 tasks sampled from 30 years of computer vision research. *Journal of Machine Learning Research*, 24(308):1–77, 2023.

[6] Yaroslav Bulatov. Notmnist dataset. 2011.

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

[8] 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 *International Conference on Learning Representations*, 2022.

[9] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

[10] Sungmin Cha, Hsiang Hsu, Taebaek Hwang, Flavio Calmon, and Taesup Moon. Cpr: Classifier-projection regularization for continual learning. In *International Conference on Learning Representations*, 2021.

[11] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision*, pages 532–547, 2018.

[12] Arslan Chaudhry, Albert Gordo, Puneet K Dokania, Philip HS Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. *Association for the Advancement of Artificial Intelligence*, 2021.

[13] Arslan Chaudhry, Naeemullah Khan, Puneet Dokania, and Philip Torr. Continual learning in low-rank orthogonal subspaces. *Advances in Neural Information Processing Systems*, 33:9900–9911, 2020.

[14] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *Proceedings of the International Conference on Learning Representations*, 2019.

[15] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *https://arxiv.org/abs/1902.10486*, 2019.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[17] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020.

[18] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[19] 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*, pages 770–778, 2016.

[20] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, 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*, pages 8340–8349, 2021.

[21] Christian Henning, Maria Cervera, Francesco D'Angelo, Johannes Von Oswald, Regina Traber, Benjamin Ehret, Seijin Kobayashi, Benjamin F Grewe, and João Sacramento. Posterior meta-replay for continual learning. *Advances in Neural Information Processing Systems*, 34:14135–14149, 2021.

[22] 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*, pages 831–839, 2019.

[23] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] P Izmailov, AG Wilson, D Podoprikhin, D Vetrov, and T Garipov. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence*, pages 876–885, 2018.

[25] Ta-Chu Kao, Kristopher Jensen, Gido van de Ven, Alberto Bernacchia, and Guillaume Hennequin. Natural continual learning: success is a journey, not (just) a destination. *Advances in neural information processing systems*, 34:28067–28079, 2021.

[26] Mohammad Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. In *International conference on machine learning*, pages 2611–2620. PMLR, 2018.

[27] Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher sam: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pages 11148–11161. PMLR, 2022.

[28] 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.

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[30] Richard Kurle, Botond Cseke, Alexej Klushyn, Patrick Van Der Smagt, and Stephan Günnemann. Continual learning with bayesian neural networks for non-stationary data. In *International Conference on Learning Representations*, 2019.

[31] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of statistics*, pages 1302–1338, 2000.

[32] Peter D Lax. *Functional analysis*, volume 55. John Wiley & Sons, 2002.

[33] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[34] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pages 3925–3934. PMLR, 2019.

[35] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[36] Yan-Shuo Liang and Wu-Jun Li. Loss decoupling for task-agnostic continual learning. *Advances in Neural Information Processing Systems*, 36, 2023.

[37] 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*, pages 24246–24255, 2023.

[38] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Trgp: Trust region gradient projection for continual learning. In *International Conference on Learning Representations*, 2022.

[39] Chenxi Liu, Zhenyi Wang, Tianyi Xiong, Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. Few-shot class incremental learning with attention-aware self-adaptive prompt. In *European Conference on Computer Vision*, 2024.

[40] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[41] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[42] James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[43] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.

[44] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[45] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. *Journal of Machine Learning Research*, 24(214):1–50, 2023.

[46] Nicolas Michel, Maorong Wang, Ling Xiao, and Toshihiko Yamasaki. Rethinking momentum knowledge distillation in online continual learning. In *Forty-first International Conference on Machine Learning*, 2024.

[47] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[48] Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

[49] Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. Continual deep learning by functional regularisation of memorable past. *Advances in Neural Information Processing Systems*, 33:4453–4464, 2020.

[50] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[51] Quang Pham, Chenghao Liu, and Steven HOI. Dualnet: Continual learning, fast and slow. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[52] Jingyang Qiao, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, Yuan Xie, et al. Prompt gradient projection for continual learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[53] Garvesh Raskutti and Sayan Mukherjee. The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457, 2015.

[54] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

[55] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[56] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[57] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019.

[58] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

[59] Tim GJ Rudner, Freddie Bickford Smith, Qixuan Feng, Yee Whye Teh, and Yarin Gal. Continual learning via sequential function-space variational inference. In *International Conference on Machine Learning*, pages 18871–18887. PMLR, 2022.

[60] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *ICLR*, 2021.

[61] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Error sensitivity modulation based experience replay: Mitigating abrupt representation drift in continual learning. In *The Eleventh International Conference on Learning Representations*, 2023.

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

[63] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pages 4548–4557. PMLR, 2018.

[64] Haizhou Shi and Hao Wang. A unified approach to domain incremental learning with memory: Theory and algorithm. *Advances in Neural Information Processing Systems*, 36, 2023.

[65] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. 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*, pages 11909–11919, 2023.

[66] Michalis K. Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*, 2020.

[67] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

[68] Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9385–9394, 2021.

[69] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[70] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[71] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021.

[72] Zhenyi Wang, Yan Li, Li Shen, and Heng Huang. A unified and general framework for continual learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[73] Zhenyi Wang, Li Shen, Tiehang Duan, Qiuling Suo, Le Fang, Wei Liu, and Mingchen Gao. Distributionally robust memory evolution with generalized divergence for continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[74] Zhenyi Wang, Li Shen, Tiehang Duan, Donglin Zhan, Le Fang, and Mingchen Gao. Learning to learn and remember super long multi-domain task sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7982–7992, June 2022.

[75] Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Tiehang Duan, and Mingchen Gao. Improving task-free continual learning by distributionally robust memory evolution. In *International Conference on Machine Learning*, pages 22985–22998. PMLR, 2022.

[76] Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Donglin Zhan, Tiehang Duan, and Mingchen Gao. Meta-learning with less forgetting on large-scale non-stationary task distributions. In *European Conference on Computer Vision*, pages 221–238. Springer, 2022.

[77] Zhenyi Wang, Li Shen, Donglin Zhan, Qiuling Suo, Yanjun Zhu, Tiehang Duan, and Mingchen Gao. Metamix: Towards corruption-robust continual learning with temporally self-adaptive data transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24521–24531, 2023.

[78] Zifeng Wang, Zheng Zhan, Yifan Gong, Yucai Shao, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. Dualhsic: Hsic-bottleneck and alignment for continual learning. In *International Conference on Machine Learning*, 2023.

[79] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022.

[80] Jiayu Wu, Qixiang Zhang, and Guoxi Xu. Tiny imagenet challenge. *Technical report*, 2017.

[81] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[82] Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. Hebbian learning based orthogonal projection for continual learning of spiking neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[83] Enneng Yang, Li Shen, Zhenyi Wang, Tongliang Liu, and Guibing Guo. An efficient dataset condensation plugin and its application to continual learning. *Advances in Neural Information Processing Systems*, 36, 2023.

[84] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[85] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[86] Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2023.

A Theorem Proof

A.1 Duality in Natural Gradient Descent for Exponential Family Distribution

Theorem A.1. *Gradient of the loss $\mathcal{L}(\boldsymbol{\lambda})$ with respect to the expectation parameter $\boldsymbol{\lambda}$, i.e., $\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda})$, is equal to the natural gradient with respect to natural parameter $\boldsymbol{\phi}$, i.e., $\boldsymbol{F}^{-1}\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi})$. This can be expressed as the following:*

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda}) = \boldsymbol{F}^{-1}\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}) \tag{19}$$

In particular, NGD in natural parameter space can be equivalently performed through gradient descent with respect to the expectation parameters as the following:

$$\boldsymbol{\phi}_{i+1} = \boldsymbol{\phi}_i - \eta\boldsymbol{F}^{-1}\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}_i) = \boldsymbol{\phi}_i - \eta\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda}_i) \tag{20}$$

where η is the learning rate and $\boldsymbol{F}$ is the Fisher information matrix (FIM).

Proof. The exponential family distribution is defined as the following:

$$P_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = exp(\langle\boldsymbol{\phi}, \boldsymbol{\Omega}(\boldsymbol{\theta})\rangle - Z(\boldsymbol{\phi})) \tag{21}$$

According to the expectation of the score function is 0, we can obtain the following

$$\mathbf{0} = \mathbb{E}_{P_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\nabla_{\boldsymbol{\phi}}\log P_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = \mathbb{E}_{P_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[\boldsymbol{\Omega}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\phi}}Z(\boldsymbol{\phi})] = \boldsymbol{\lambda} - \nabla_{\boldsymbol{\phi}}Z(\boldsymbol{\phi}) \tag{22}$$

Therefore,

$$\boldsymbol{\lambda} = \nabla_{\boldsymbol{\phi}}Z(\boldsymbol{\phi}) \tag{23}$$

where the first equality is due to the fact that the expectation of the score function is zero.

We then derive the Fisher information matrix (FIM) as the following:

$$\begin{aligned}
\boldsymbol{F}(\boldsymbol{\phi}) &:= \mathbb{E}_{P_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[-\nabla^2_{\boldsymbol{\phi}}\log P_{\boldsymbol{\phi}}(\boldsymbol{\theta})] && (24)\\
&= \mathbb{E}_{P_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[-\nabla_{\boldsymbol{\phi}}(\nabla_{\boldsymbol{\phi}}\log P_{\boldsymbol{\phi}}(\boldsymbol{\theta}))] && (25)\\
&= \mathbb{E}_{P_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[-\nabla_{\boldsymbol{\phi}}(\nabla_{\boldsymbol{\phi}}(\langle\boldsymbol{\phi}, \boldsymbol{\Omega}(\boldsymbol{\theta})\rangle - Z(\boldsymbol{\phi}))] && (26)\\
&= \mathbb{E}_{P_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[-\nabla_{\boldsymbol{\phi}}(\boldsymbol{\Omega}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\phi}}Z(\boldsymbol{\phi}))] && (27)\\
&= \nabla_{\boldsymbol{\phi}}\boldsymbol{\lambda} && (28)\\
&= \nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\phi}}Z(\boldsymbol{\phi}) && (29)\\
&= \nabla^2_{\boldsymbol{\phi}}Z(\boldsymbol{\phi}) && (30)
\end{aligned}$$

where $:=$ denotes defined as. Then,

$$\nabla_{\boldsymbol{\phi}}\boldsymbol{\lambda} = \nabla^2_{\boldsymbol{\phi}}Z(\boldsymbol{\phi}) = \boldsymbol{F} \tag{31}$$

Next,

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\lambda}}\boldsymbol{\phi}\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}) = [\nabla_{\boldsymbol{\phi}}\boldsymbol{\lambda}]^{-1}\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}) = \boldsymbol{F}^{-1}\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}) \tag{32}$$

□

More general results on manifold can be found in [53].

A.2 Theoretical and Generalization Analysis of MACL

Lemma A.2. $D_{\mathrm{KL}}(\mathbb{U}, \mathbb{V}) = \int u(\boldsymbol{\theta})\log(\frac{u(\boldsymbol{\theta})}{v(\boldsymbol{\theta})})d\boldsymbol{\theta} \leq \int \frac{(u(\boldsymbol{\theta})-v(\boldsymbol{\theta}))^2}{v(\boldsymbol{\theta})}d\boldsymbol{\theta}$

Proof.

$$D_{\mathrm{KL}}(\mathbb{U},\mathbb{V}) = \int u(\boldsymbol{\theta})\log(\frac{u(\boldsymbol{\theta})}{v(\boldsymbol{\theta})})d\boldsymbol{\theta} \tag{33}$$

$$\leq \log\int \frac{u(\boldsymbol{\theta})^2}{v(\boldsymbol{\theta})}d\boldsymbol{\theta} \ \text{ (by Jensen's inequality)} \tag{34}$$

$$\leq \int \frac{u(\boldsymbol{\theta})^2}{v(\boldsymbol{\theta})} - 1 d\boldsymbol{\theta} \ \ (\log(1+\boldsymbol{x}) \leq \boldsymbol{x}) \tag{35}$$

$$= \int \frac{(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))^2}{v(\boldsymbol{\theta})}d\boldsymbol{\theta} \tag{36}$$

where the last equality is because

$$\int \frac{(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))^2}{v(\boldsymbol{\theta})}d\boldsymbol{\theta} = \int \frac{u(\boldsymbol{\theta})^2}{v(\boldsymbol{\theta})} - 2\int u(\boldsymbol{\theta})d\boldsymbol{\theta} + \int v(\boldsymbol{\theta})d\boldsymbol{\theta} = \int \frac{u(\boldsymbol{\theta})^2}{v(\boldsymbol{\theta})} - 1 \tag{37}$$

Since $\int u(\boldsymbol{\theta})d\boldsymbol{\theta} = \int v(\boldsymbol{\theta})d\boldsymbol{\theta} = 1$

□

Theorem A.3. *Assume $\int ||\frac{1}{v(\boldsymbol{\theta})}||_\infty d\boldsymbol{\theta} \leq M$, we can obtain the following conclusion for Eq. (16):*

$$\max_{\mathbb{U}\in\mathcal{U}} \int \mathcal{L}^{CL}(\boldsymbol{\theta})d\mathbb{U}(\boldsymbol{\theta}) = \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} + \sqrt{\frac{\epsilon\mathbb{E}(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2}{M}} \tag{38}$$

where $\overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} := \int \mathcal{L}^{CL}(\boldsymbol{\theta})d\mathbb{V}(\boldsymbol{\theta})$. We denote the variance of the random variable $\mathcal{L}^{CL}(\boldsymbol{\theta})$ as $Var(\mathcal{L}^{CL}(\boldsymbol{\theta})) = \mathbb{E}(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2 = \int(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2 d\boldsymbol{\theta}$.

Proof. We define a new distribution $\mathbb{Z} := \mathbb{U} - \mathbb{V}$.

$$\int \mathcal{L}^{CL}(\boldsymbol{\theta})d\mathbb{U}(\boldsymbol{\theta}) = \int \mathcal{L}^{CL}(\boldsymbol{\theta})d(\mathbb{Z}(\boldsymbol{\theta}) + \mathbb{V}(\boldsymbol{\theta})) = \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} + \int \mathcal{L}^{CL}(\boldsymbol{\theta})d\mathbb{Z}(\boldsymbol{\theta}) \tag{39}$$

$$= \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} + \int(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})d\mathbb{Z}(\boldsymbol{\theta}) + \int \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})}d\mathbb{Z}(\boldsymbol{\theta}) \tag{40}$$

By Lemma 4.1 and Hölder's inequality, we can obtain the following:

$$D_{\mathrm{KL}}(\mathbb{U},\mathbb{V}) = \int u(\boldsymbol{\theta})\log(\frac{u(\boldsymbol{\theta})}{v(\boldsymbol{\theta})})d\boldsymbol{\theta} \leq \int \frac{(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))^2}{v(\boldsymbol{\theta})}d\boldsymbol{\theta} \tag{41}$$

$$\leq \int(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))^2 d\boldsymbol{\theta}\int ||\frac{1}{v(\boldsymbol{\theta})}||_\infty d\boldsymbol{\theta} \tag{42}$$

$$\leq \int(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))^2 d\boldsymbol{\theta}M \leq \epsilon \tag{43}$$

Therefore,

$$\int(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))^2 d\boldsymbol{\theta} \leq \frac{\epsilon}{M} \tag{44}$$

$$\int(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})d\mathbb{Z}(\boldsymbol{\theta}) = \int(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))d\boldsymbol{\theta} \tag{45}$$

$$\leq \sqrt{\int(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2 d\boldsymbol{\theta}\int(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))^2 d\boldsymbol{\theta}} \text{ (Cauchy-Schwarz inequality)} \tag{46}$$

$$\leq \sqrt{\frac{\epsilon\mathbb{E}(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2}{M}} \tag{47}$$

The equality holds when the following condition holds:

$$u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}) = a(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})}) \tag{48}$$

where a is a constant.

$$\int \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} d\mathbb{Z}(\boldsymbol{\theta}) = \int \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})}(u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))d\boldsymbol{\theta} \tag{49}$$

$$= \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} \int (u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))d\boldsymbol{\theta} \tag{50}$$

$$= 0 \tag{51}$$

The last equality is because $\int (u(\boldsymbol{\theta}) - v(\boldsymbol{\theta}))d\boldsymbol{\theta} = \int u(\boldsymbol{\theta})d\boldsymbol{\theta} - \int v(\boldsymbol{\theta})d\boldsymbol{\theta} = 1 - 1 = 0$

Therefore, we can obtain the following conclusion:

$$\max_{\mathbb{U}\in\mathcal{U}} \int \mathcal{L}^{CL}(\boldsymbol{\theta}) d\mathbb{U}(\boldsymbol{\theta}) = \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})} + \sqrt{\frac{\epsilon \mathbb{E}(\mathcal{L}^{CL}(\boldsymbol{\theta}) - \overline{\mathcal{L}^{CL}(\boldsymbol{\theta})})^2}{M}} \tag{52}$$

□

In this context, the CL loss variance across various sets of model parameters $Var(\mathcal{L}^{CL}(\boldsymbol{\theta}))$ serves as a measure of the CL model's sensitivity to parameter updates. Essentially, a smaller loss variance indicates lower parameter sensitivity in the CL model. However, directly optimizing the loss variance within the parameter distribution neighborhood is impractical, as it requires computing the loss variance across a large number of different sets of CL model parameters and training data points. In contrast, our method (MACL) offers an efficient and effective alternative. MACL implicitly minimizes the loss variance across different model parameter variations by optimizing CL performance solely on the worst-case CL model parameter distribution.

We denote the prior distribution as $\mathbb{V}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and posterior distribution as $\mathbb{U}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$

Theorem A.4 (Generalization bound of MACL)**.** *Let q be the number of CL model parameters and n be the number of training data points. The CL loss $\mathcal{L}^{CL}(\boldsymbol{\theta}) \leq C$ (C is a constant). With high probability of $1-\delta$, the following bound holds:*

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \sum_{i}^{i=T} \mathcal{L}^{CL}_{\mathcal{D}_i}(\boldsymbol{\theta}) \leq \max_{\mathbb{U}\in\mathcal{U}} \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}} \mathcal{L}^{CL}(\boldsymbol{\theta}) + \frac{C}{N_T + \zeta \sum_{i=1}^{i=T-1} N_i} + \sqrt{\frac{\tau^2(\sqrt{q} + \sqrt{2\log(N_T + \zeta \sum_{i=1}^{i=T-1} N_i)})^2 + R + 2\log(\frac{N_T + \zeta \sum_{i=1}^{i=T-1} N_i}{\delta})}{4(N_T + \zeta \sum_{i=1}^{i=T-1} N_i - 1)}} \tag{53}$$

Where τ is a constant. We denote the number of data examples for task $1, \cdots, T-1$ in the memory buffer $\mathcal{M}$ during training on task T as $N_1, N_2, \cdots, N_{T-1}$ when using memory replay based approach or the number of training data points when using regularization based approach. $\mathcal{L}^{CL}_{\mathcal{D}_i}(\boldsymbol{\theta})$ denotes the CL loss on the data from data distribution $\mathcal{D}_i$ (generalization error), i.e., it is defined as: $\mathcal{L}^{CL}_{\mathcal{D}_i}(\boldsymbol{\theta}) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_i} \mathcal{L}(\boldsymbol{x}, y, \boldsymbol{\theta})$. $\mathcal{L}^{CL}(\boldsymbol{\theta})$ denotes the empirical CL loss as Eq. (1). $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the CL model parameter posterior distribution parameterized with Gaussian distribution.

Proof. We apply the PAC-Bayes theorem [43] that for any prior distribution, with probability $1-\delta$ over the CL training dataset $\mathcal{T}$, the following bound holds:

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}[\mathcal{L}^{CL}_{\mathcal{D}}(\boldsymbol{\theta})] \leq \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}[\mathcal{L}^{CL}_{\mathcal{T}}(\boldsymbol{\theta})] + \sqrt{\frac{D_{\mathrm{KL}}(\mathbb{U}(\boldsymbol{\theta})||\mathbb{V}(\boldsymbol{\theta})) + \log(\frac{n}{\delta})}{2(n-1)}} \tag{54}$$

The KL divergence between posterior and prior distribution can be calculated as the following:

$$D_{\mathrm{KL}}(\mathbb{U}(\boldsymbol{\theta})||\mathbb{V}(\boldsymbol{\theta})) = \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}[\log(\mathbb{U}(\boldsymbol{\theta})) - \log(\mathbb{V}(\boldsymbol{\theta}))] \tag{55}$$

$$= \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}_s|} - \frac{1}{2}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}(\boldsymbol{\theta}-\boldsymbol{\mu}_s)^T\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_s) + \frac{1}{2}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}(\boldsymbol{\theta}-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_p) \tag{56}$$

$$= \frac{1}{2}[\log\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}_s|} - q + (\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p) + Tr(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}_s)] \tag{57}$$

We assume the following inequality:

$$\log\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}_s|} + Tr(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}_s) \leq R + q, \quad R \geq 0 \tag{58}$$

Therefore,

$$D_{\mathrm{KL}}(\mathbb{U}(\boldsymbol{\theta})||\mathbb{V}(\boldsymbol{\theta})) \leq \frac{1}{2}[R + (\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)] \tag{59}$$

According to [50], we have the following identity:

For a random variable $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\boldsymbol{\theta}-\boldsymbol{\mu}')^T\boldsymbol{A}(\boldsymbol{\theta}-\boldsymbol{\mu}') = (\boldsymbol{\mu}-\boldsymbol{\mu}')^T\boldsymbol{A}(\boldsymbol{\mu}-\boldsymbol{\mu}') + Tr(\boldsymbol{A}\boldsymbol{\Sigma}) \tag{60}$$

where Tr denotes the trace of A matrix. Therefore, according to Eq. (60), we have the following two equations 61 and 62.

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}(\boldsymbol{\theta}-\boldsymbol{\mu}_s)^T\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_s) = (\boldsymbol{\mu}_s-\boldsymbol{\mu}_s)^T\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_s) + Tr(\boldsymbol{\Sigma}_s^{-1}\boldsymbol{\Sigma}_s) = q \tag{61}$$

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}(\boldsymbol{\theta}-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_p) = (\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p) + Tr(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}_s) \tag{62}$$

We set $\boldsymbol{\gamma} = \boldsymbol{\Sigma}_p^{-\frac{1}{2}}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)$. Then, $||\boldsymbol{\gamma}||^2 = (\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)$.

If $\boldsymbol{\gamma} \sim N(\mathbf{0}, \tau^2\boldsymbol{I})$, according to [31], we have the following inequality with probability of $1-\frac{1}{n}$

$$||\boldsymbol{\gamma}||^2 \leq \tau^2(q + 2\sqrt{q\log n} + 2\log n) \leq \tau^2(\sqrt{q} + \sqrt{2\log n})^2 \tag{63}$$

Then we partition the space of $\boldsymbol{\mu}_s$ into two disjoint area that satisfy $(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p) \leq 2\epsilon - R$ and $(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p) > 2\epsilon - R$.

(1) In the case of $(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p) \leq 2\epsilon - R$, we take the maximum loss over $\boldsymbol{\mu}_s$, we have the following inequality:

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}[\mathcal{L}_{\mathcal{T}}^{CL}(\boldsymbol{\theta})] \leq \max_{(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)\leq 2\epsilon-R} \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}\mathcal{L}^{CL}(\boldsymbol{\theta}) \tag{64}$$

(2) For the case of $(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p) > 2\epsilon - R$, we have $\mathcal{L}_{\mathcal{T}}^{CL}(\boldsymbol{\theta}) \leq C$

Combining case (1) and (2), we can obtain the following generalization bound:

$$D_{\mathrm{KL}}(\mathbb{U},\mathbb{V}) \leq \frac{1}{2}[(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p) + R + q - q] \leq \frac{1}{2}[||\boldsymbol{\gamma}||^2 + R] \tag{65}$$

$$\leq \frac{1}{2}[\tau^2(\sqrt{q} + \sqrt{2\log n})^2 + R] \tag{66}$$

We have the following bound with probability of $1-\frac{1}{n}$:

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}[\mathcal{L}_{\mathcal{T}}^{CL}(\boldsymbol{\theta})] \leq (1-\frac{1}{n}) \max_{(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_s-\boldsymbol{\mu}_p)\leq 2\epsilon-R} \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \frac{C}{n} \tag{67}$$

$$\leq (1-\frac{1}{n}) \max_{D_{\mathrm{KL}}(\mathbb{U},\mathbb{V})\leq\epsilon} \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}(\boldsymbol{\theta})}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \frac{C}{n} \tag{68}$$

Then, we can obtain the following generalization bound with probability of $1-\frac{1}{n}$:

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\sum_{i}^{i=T}\mathcal{L}_{\mathcal{D}_i}^{CL}(\boldsymbol{\theta}) \leq \max_{\mathbb{U}\in\mathcal{U}}\mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{U}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \frac{C}{N_T+\zeta\sum_{i=1}^{i=T-1}N_i} + \tag{69}$$

$$\sqrt{\frac{\tau^2(\sqrt{q}+\sqrt{2\log(N_T+\zeta\sum_{i=1}^{i=T-1}N_i)})^2+R+2\log(\frac{N_T+\zeta\sum_{i=1}^{i=T-1}N_i}{\delta})}{4(N_T+\zeta\sum_{i=1}^{i=T-1}N_i-1)}}$$

□

In this theorem, we provide the theoretical guarantee for the generalization analysis of our proposed method. This bound indicates by optimizing the MACL loss, our method tighten/reduce the generalization error of the CL method, thus improving the overall performance of our method.

B Equation Derivation

B.1 Exponential Family Distribution Details

According to the definition of expectation, we can obtain the following equation:

$$\boldsymbol{\lambda}^1 := \mathbb{E}_{f(\boldsymbol{\theta};\boldsymbol{\mu},\boldsymbol{\Sigma})}\boldsymbol{\theta} = \boldsymbol{\mu} \tag{70}$$

According to the definition of covariance matrix,

$$\boldsymbol{\Sigma} := \mathbb{E}[(\boldsymbol{\theta}-\boldsymbol{\mu})(\boldsymbol{\theta}-\boldsymbol{\mu})^T] \tag{71}$$

$$= \mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^T - 2\boldsymbol{\mu}\boldsymbol{\theta} + \boldsymbol{\mu}\boldsymbol{\mu}^T] \tag{72}$$

$$= \mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T \tag{73}$$

By rearranging the above equation, we can obtain the following:

$$\mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^T] = \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma} \tag{74}$$

Then, the conclusion follows:

$$\boldsymbol{\lambda}^1 := \mathbb{E}_{f(\boldsymbol{\theta};\boldsymbol{\mu},\boldsymbol{\Sigma})}\boldsymbol{\theta} = \boldsymbol{\mu}, \quad \boldsymbol{\lambda}^2 := \mathbb{E}_{f(\boldsymbol{\theta};\boldsymbol{\mu},\boldsymbol{\Sigma})}\boldsymbol{\theta}\boldsymbol{\theta}^T = \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma} \tag{75}$$

B.2 Worst-Case Gaussian Distribution NGD Derivations

Gradient of Loss $\mathcal{L}(\boldsymbol{\lambda})$ With Respect to $\boldsymbol{\lambda}$ Taking gradient with respect to $\boldsymbol{\lambda}$ as the following:

$$\nabla_{\boldsymbol{\lambda}^1}\mathcal{L}(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\mu}}\mathcal{L}(\boldsymbol{\lambda})\frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\lambda}^1} + \nabla_{\boldsymbol{\Sigma}}\mathcal{L}(\boldsymbol{\lambda})\frac{\partial\boldsymbol{\Sigma}}{\partial\boldsymbol{\lambda}^1} = \nabla_{\boldsymbol{\mu}}\mathcal{L}(\boldsymbol{\lambda}) - 2\nabla_{\boldsymbol{\Sigma}}\mathcal{L}(\boldsymbol{\lambda})\boldsymbol{\mu} \tag{76}$$

In Eq. (76), the second equality is because the identity: $\frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\lambda}^1} = \mathbf{1}, \quad \frac{\partial\boldsymbol{\Sigma}}{\partial\boldsymbol{\lambda}^1} = \frac{\partial\boldsymbol{\Sigma}}{\partial\boldsymbol{\mu}} = -2\boldsymbol{\mu}$. (by Eq. (75))

$$\nabla_{\boldsymbol{\lambda}^2}\mathcal{L}(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\mu}}\mathcal{L}(\boldsymbol{\lambda})\frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\lambda}^2} + \nabla_{\boldsymbol{\Sigma}}\mathcal{L}(\boldsymbol{\lambda})\frac{\partial\boldsymbol{\Sigma}}{\partial\boldsymbol{\lambda}^2} = \nabla_{\boldsymbol{\Sigma}}\mathcal{L}(\boldsymbol{\lambda}) \tag{77}$$

In Eq. (77), the second equality is because the identity: $\frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\lambda}^2} = \mathbf{0}, \quad \frac{\partial\boldsymbol{\Sigma}}{\partial\boldsymbol{\lambda}^2} = \mathbf{1}$ (by Eq. (75))

According to Eq. (5), we set the natural parameters as:

$$\boldsymbol{\phi}^1 := \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \quad \boldsymbol{\phi}^2 := -\frac{1}{2}\boldsymbol{\Sigma}^{-1} \tag{78}$$

- (1) *NGD with respect to* ϕ^2: According to Eq. (20 and 77), NGD with respect to ϕ^2 can be obtained as:

$$-\frac{1}{2}\boldsymbol{\Sigma}_{i+1}^{-1} = -\frac{1}{2}\boldsymbol{\Sigma}_{i}^{-1} - \eta\nabla_{\boldsymbol{\lambda}^2}\mathcal{L}(\boldsymbol{\lambda}_i) = -\frac{1}{2}\boldsymbol{\Sigma}_{i}^{-1} - \eta\nabla_{\boldsymbol{\Sigma}}\mathcal{L}(\boldsymbol{\lambda}_i) \tag{79}$$

  Then, obtain the following update:

$$\boldsymbol{\Sigma}_{i+1}^{-1} = \boldsymbol{\Sigma}_{i}^{-1} + 2\eta\nabla_{\boldsymbol{\Sigma}}\mathcal{L}(\boldsymbol{\lambda}_i) \tag{80}$$

- (2) *NGD with respect to* ϕ^1: Similarly, according to Eq. (20 and 76), NGD with respect to ϕ^1 can be obtained as:

$$\boldsymbol{\Sigma}_{i+1}^{-1}\boldsymbol{\mu}_{i+1} = \boldsymbol{\Sigma}_{i}^{-1}\boldsymbol{\mu}_i - \eta(\nabla_{\boldsymbol{\mu}}\mathcal{L}(\boldsymbol{\lambda}_i) - 2\nabla_{\boldsymbol{\Sigma}}\mathcal{L}(\boldsymbol{\lambda}_i)\boldsymbol{\mu}_i) \tag{81}$$

  By simplifying and rearranging Eq. (81), the following update for $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}_{i+1} = \boldsymbol{\mu}_i - \eta\boldsymbol{\Sigma}_{i+1}\nabla_{\boldsymbol{\mu}}\mathcal{L}(\boldsymbol{\lambda}_i) \tag{82}$$

Mean and Covariance Updates Derivations Following the results in [56, 26], we can obtain the following equation:

$$\nabla_{\boldsymbol{\mu}}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{83}$$

$$\nabla_{\boldsymbol{\Sigma}}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{84}$$

Then, we only need to calculate $\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ and $\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma})$. Here, since we assumed a general CL Gaussian distribution for the current CL parameter distribution, i.e., $\mathbb{V}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)$ and neighbourhood distribution, i.e., $\mathbb{U}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\Sigma})$. The detailed derivations for the gradient are present in the following:

$$\begin{aligned}
\nabla_{\boldsymbol{\mu}}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma}) &= -\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla_{\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \alpha\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[\nabla_{\boldsymbol{\theta}}\log u(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}}\log v(\boldsymbol{\theta})] && (85)\\
&= -\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla_{\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) - \alpha\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}(\boldsymbol{\theta}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1} + \alpha\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}(\boldsymbol{\theta}-\boldsymbol{\mu}_0)\boldsymbol{\Sigma}_0^{-1} && (86)\\
&= \mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[-\nabla_{\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \alpha(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\boldsymbol{\Sigma}_0^{-1}] && (87)
\end{aligned}$$

$$\begin{aligned}
\nabla_{\boldsymbol{\Sigma}}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma}) &= -\frac{1}{2}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \alpha\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[\log u(\boldsymbol{\theta}) - \log v(\boldsymbol{\theta})] && (88)\\
&= -\frac{1}{2}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \frac{\alpha}{2}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\log u(\boldsymbol{\theta}) - \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\log v(\boldsymbol{\theta})] && (89)\\
&= -\frac{1}{2}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \frac{\alpha}{2}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[-\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}_0^{-1}] && (90)\\
&= \frac{1}{2}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[-\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) - \alpha\boldsymbol{\Sigma}^{-1} + \alpha\boldsymbol{\Sigma}_0^{-1}] && (91)
\end{aligned}$$

Plug-in the gradient derivation into Eq. (82 and 80), we can obtain the following results:

$$\begin{aligned}
\boldsymbol{\Sigma}_{i+1}^{-1} &= \boldsymbol{\Sigma}_{i}^{-1} + \eta\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[-\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) - \alpha\boldsymbol{\Sigma}_i^{-1} + \alpha\boldsymbol{\Sigma}_0^{-1}] && (92)\\
&= (1-\eta\alpha)\boldsymbol{\Sigma}_{i}^{-1} + \eta\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[-\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \alpha\boldsymbol{\Sigma}_0^{-1}] && (93)
\end{aligned}$$

$$\boldsymbol{\mu}_{i+1} = \boldsymbol{\mu}_i - \eta\boldsymbol{\Sigma}_{i+1}\mathbb{E}_{\boldsymbol{\theta}\sim u(\boldsymbol{\theta})}[-\nabla_{\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}) + \alpha(\boldsymbol{\mu}_i-\boldsymbol{\mu}_0)\boldsymbol{\Sigma}_0^{-1}] \tag{94}$$

Finally, by using single sample from distribution $\mathbb{U}$ with density $\boldsymbol{\theta} \sim u(\boldsymbol{\theta})$ to approximate the expectation. By plug-in $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma}^2)$ and $\boldsymbol{\Sigma}_0 = \text{diag}(\boldsymbol{\rho}^2)$ into the above equations, we can obtain the following updates:

$$\begin{aligned}
\boldsymbol{\mu}_{i+1} &= \boldsymbol{\mu}_i + \eta\boldsymbol{\sigma}^2_{i+1}[\nabla_{\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}_i) - \alpha(\boldsymbol{\mu}_i-\boldsymbol{\mu}_0)\boldsymbol{\rho}^{-2}] && (95)\\
\boldsymbol{\sigma}^{-2}_{i+1} &= (1-\eta\alpha)\boldsymbol{\sigma}_i^{-2} + \eta[-\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{L}^{CL}(\boldsymbol{\theta}_i) + \alpha\boldsymbol{\rho}^{-2}] && (96)
\end{aligned}$$

C Baseline Details

- **EWC** [28]: EWC endeavors to alleviate forgetting in continual learning through the utilization of a weighted weight regularization technique based on the Fisher information matrix.
- **CPR** [10]: Drawing on neural networks with wide local minima and principles from information theory, CPR introduces an extra regularization term. This term aims to maximize the entropy of a classifier's output probabilities, thereby reaching wider local minima to enhance generalization.
- **GPM** [60]: A CL model acquires new skills by adjusting its parameters through gradient steps that move orthogonal to the gradient subspaces considered vital for previous tasks. The Gradient Projection Memory (GPM) establishes these subspaces by analyzing network activations following the completion of each task using Singular Value Decomposition (SVD), then preserves them in memory.
- **HAT** [63]: HAT is a task-driven hard attention mechanism that retains information from prior tasks while ensuring it doesn't interfere with the current task's learning process.
- **A-GEM** [14]: AGEM aims to guarantee that, at every training step, the average loss of episodic memory over past tasks does not rise, thus mitigating the risk of forgetting previously acquired knowledge.
- **Gradient-Based Sample Selection (GSS-Greedy)** [2]: The goal is to populate the memory buffer with a diverse set of examples, using the data gradient as a feature for sample selection. For comparison, we opt for the efficient GSS-Greedy version.
- **ER** [15]: This method stores a subset of examples from previous tasks using reservoir sampling [15]. During each iteration, we randomly replay a subset of examples from the memory buffer.
- **DER++** [7]: This method combines experience replay with knowledge distillation to further improve the effectiveness of experience replay.
- **ER-ACE** [8]: They discovered that ER causes significant overlap between the representations of newly added classes and previous ones, resulting in highly disruptive parameter updates. From this empirical analysis, they proposed a new method to address this issue by protecting the learned representations from drastic adaptations when incorporating new classes. Their approach uses an asymmetric update rule that pushes new classes to adapt to the older ones, rather than the reverse. This technique is particularly effective at task boundaries, where much of the forgetting typically happens.
- **LODE** [36]: They conducted an in-depth analysis of the impacts of distinguishing between new and old classes, as well as among new classes, finding that these two learning objectives result in varying degrees of forgetting. Consequently, combining these objectives negatively affects the performance of the CL model. To address this, LODE separates the two objectives for new tasks by decoupling the loss associated with them. This approach allows LODE to assign different weights to each objective, leading to better performance compared to methods that use a coupled loss.

D More Experimental Results

D.1 Hyperparameter Sensitivity Analysis

Table 5: Analysis of hyperparameter η on CIFAR100 and Tiny-ImageNet in the setting of task-IL.

η	0.0	1e-7	1e-6	1e-5	3×1e-5
CIFAR100	75.64±0.60	77.16±0.42	77.69±0.37	77.53±0.89	76.97±0.46
Tiny-ImageNet	51.91±0.68	53.12±0.82	54.03±0.79	54.46±0.91	51.62±0.55

D.2 Benefit of NGD

Table 6: Benefit of MACL-NGD vs. MACL-GD on CIFAR100 and Tiny-ImageNet in the setting of task-IL.

method	DER++	DER++(MACL-NGD)	DER++(MACL-GD)
CIFAR100	75.64±0.60	77.53±0.89	76.62±0.53
Tiny-ImageNet	51.91±0.68	54.03±0.79	52.97±0.71

D.3 Online CL results

Table 7: Online CL Results on CIFAR100 under the blurry boundary setting

Memory Size	1000	2000	5000
MKD(PCR)	35.6 ± 0.66	44.95 ± 0.42	54.87 ± 0.39
MKD(PCR) + MACL	**37.2 ± 0.53**	**46.17 ± 0.51**	**56.21 ± 0.43**

Table 8: Online CL Results on Tiny-ImageNet under the blurry boundary setting

Memory Size	2000	5000	10000
MKD(PCR)	17.33 ± 1.28	29.58 ± 0.60	38.02 ± 1.64
MKD(PCR) + MACL	**18.21 ± 1.32**	**30.69 ± 0.71**	**38.73 ± 1.56**

D.4 Prompt-based CL results

We conducted an experiment integrating MACL with the SOTA prompt-based CL method, CODA-Prompt [65]. Our method operates on the parameters of prompt components and corresponding keys/attention vectors.

Table 9: CODA Prompt Results on ImageNet-R

Number of Tasks	10	20
CODA-P	75.45 ± 0.56	72.37 ± 1.19
CODA-P + MACL	**76.39 ± 0.67**	**73.42 ± 1.23**

D.5 5-datasets results

Table 10: Comparison of methods on Class-IL and Task-IL on 5-datasets.

Method	Class-IL	Task-IL
ER	66.03 ± 1.37	92.58 ± 1.26
ER+MACL	**67.32 ± 1.18**	**93.21 ± 1.08**
DER++	85.92 ± 0.33	87.16 ± 0.21
DER++MACL	**87.23 ± 0.51**	**87.51 ± 0.30**

D.6 Effect of Different Architectures

Table 11: Overall accuracy with ResNet32 using a memory buffer of 2000 by integrating with MEMO.

	MEMO	MEMO+MACL
accuracy	58.49	59.61

Table 12: Overall accuracy with ViT using a memory buffer of 500 by integrating DER++ with MACL.

	Class-IL	Task-IL
DER++	76.21 ± 0.67	96.72 ± 0.31
DER++ MACL	77.83 ± 0.80	97.31 ± 0.46

D.7 ImageNet-R and CUB200 results

We conducted experiment on the recent CL datasets of ImageNet-R and CUB200 with pre-trained Vision Transformer (ViT), i.e., vit-base-patch16-224 as the backbone following the codebase of DER++. The results (memory size of 500) are shown in the following table.

Table 13: ImageNet-R Results

Method	Class-IL	Task-IL
DER++	58.29 ± 1.78	86.93 ± 0.32
DER++MACL	**60.51 ± 1.65**	**87.56 ± 0.41**
LODE	74.98 ± 0.21	90.22 ± 0.39
LODE+MACL	**75.51 ± 0.26**	**90.81 ± 0.28**

Table 14: CUB200 Results

Method	Class-IL	Task-IL
DER++	41.81 ± 1.69	87.16 ± 1.09
DER++MACL	**43.07 ± 1.53**	**88.03 ± 0.97**
LODE	66.87 ± 0.35	93.12 ± 0.56
LODE+MACL	**67.53 ± 0.51**	**93.42 ± 0.37**

D.8 Efficiency Evaluation

Table 15: Running efficiency of MACL on CIFAR100 by training for a single epoch on CIFAR100.

CL method	w/o MACL	w/ MACL
DER++	8.7	13.5
ER-ACE	6.3	10.2
LODE	13.2	20.8

E Experiment Setup

E.1 Dataset Statistics

Table 16: Dataset Statistics

Dataset	**Seq-CIFAR10**	**Seq-CIFAR100**	**Seq-TinyImageNet**
Number of Tasks	5	10	10
Number of Classes	10	100	200
Number of Training Samples	50,000	50,000	100,000
Number of Test Samples	10,000	10,000	10,000

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction summarize the main contributions in our paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We dicussed the limitations of our work after conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the full set of assumptions and a complete (and correct) proof in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our contribution is a new continual learning algorithm. We described full implementation details for our proposed algorithm.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code will be released.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provided detailed implementation details regarding training and test details, data splits, hyperparameters and type of optimizer.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provided results standard deviation with multiple experiment runs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided sufficient information on the computer resources in implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed the societal impacts afer conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the code package produced by DER++ and the dataset used, e.g., CIFAR10, CIFAR100, TinyImageNet.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.