



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/241286/>

Version: Accepted Version

Article:

Wang, X., Liu, T., Zhou, M. et al. (2026) Adapting to dissimilar tasks for continual learning via gradient norm regularisation. *Neurocomputing*, 695. 134015. ISSN: 0925-2312

<https://doi.org/10.1016/j.neucom.2026.134015>

© 2026 The Authors. Except as otherwise noted, this author-accepted version of a journal article published in *Neurocomputing* is made available via the University of Sheffield Research Publications and Copyright Policy under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Adapting to Dissimilar Tasks for Continual Learning via Gradient Norm Regularisation

Xulong Wang^a, Tong Liu^a, Menghui Zhou^a, Yu Zhang^a, Zhipeng Yuan^a, Kang Liu^b and Po Yang^{a,*}

^aSchool of Computer Science, University of Sheffield, Sheffield, UK

^bDepartment of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Keywords:

continual learning
task similarity measure
knowledge transfer
catastrophic forgetting

ABSTRACT

An essential challenge of continual learning (CL) is that the knowledge gained from old tasks might be erased as new tasks are learnt, which is called “catastrophic forgetting” (CF). Most CL methods assume tasks are drawn independently and identically distributed (i.i.d.); however, they ignore that as the number of tasks grows, some tasks inevitably share similarities, which could be leveraged for knowledge protection and transfer. This paper aims to enhance the adaptation of continual learning by leveraging task similarity. Specifically, we define task similarity as the change in empirical loss before training and show how it correlates with CF. As task similarity decreases, gradient norms increase, causing more severe forgetting. Building on this insight, we propose incorporating a Gradient Norm Regularisation (GNR) approach into the CL process. During training on a new task, we slow down the update magnitude for parameters deemed dissimilar. Meanwhile, gradient directions are adjusted to improve adaptability. Once training is complete, dissimilar parameters are consolidated to prevent them from being easily altered in future tasks. Experiments on four benchmark datasets show that our method outperforms other regularisation techniques and nearly matches the performance of multi-task learning. By plugging into existing regularisation and replay methods, GNR enhances the adaptation of continual learning. These results confirm that gradient norms can serve as an effective supervisory signal to balance knowledge protection and transfer. An extended experiment on a real-world pest classification dataset validates GNR’s generalisability and the soundness of the task-similarity assumption in practical applications. We make the code of GNR publicly available at <https://github.com/wang-xulong/GNR.git>.

1. Introduction

Many practical vision applications need to continually learn new tasks over time without storing or revisiting earlier training data because of storage limits, data ownership or privacy concerns. For example, a smartphone photo app may need to enrol new faces in its face recognition model without retaining previous images. In this setting, continual learning, also known as lifelong learning, aims to enable models to tackle a sequence of tasks while retaining relevant knowledge from earlier experiences (Fig. 1). However, since the network uses a single set of parameters for all tasks and only sees current data, learning a new task can interfere with and even overwrite previously acquired knowledge. This challenge is called catastrophic forgetting (CF) [38].

To mitigate CF of old tasks, many methods have been proposed for continual learning as maintenance of old knowledge and are generally classified into three types [44]: memory-based [30, 41, 40, 4], architecture-based [8, 32, 24, 53] and regularisation [17, 1, 25, 9, 42, 27, 28]. Memory-based methods selectively retain important data [41, 40] from old tasks or utilise a generative network [4], and train them jointly with new task data. However, the privacy of data and the complexity of generative models limit their application

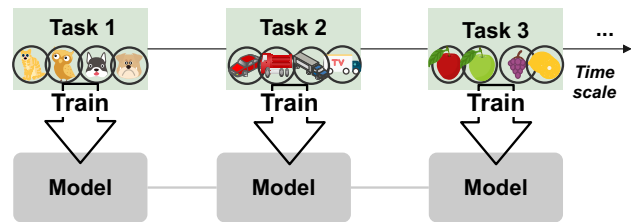


Figure 1: The illustration of continual learning. Each classification task is assumed to be drawn independently and identically distributed, with the model shared across all tasks.

scenarios. Architecture-based methods [31, 8, 32, 24, 53] isolate a few specific parameters (or neurons) [32, 31] that are important for each task or expend additional task-specific parameters [8, 24, 53] when a new task comes. As a result, isolated specific parameters limit their capability to transfer knowledge between tasks, and additional task-specific architecture requires substantial computation costs, constraining practical applications. Regularisation approaches incorporate penalty techniques into the gradient updates while training on a new task (Fig. 1 (b) Regularisation). Unlike other categories, regularisation methods do not require additional costs to retain historical data or dynamically expand the architecture, making them efficient and potentially integrable with other categories. Despite these advances, most continual learning methods implicitly assume tasks are pairwise dissimilar and ignore that, as more

*Corresponding author at: Department of Computer Science, Sheffield University, Sheffield, UK

✉ x1.wang@sheffield.ac.uk (X. Wang); t.liu@sheffield.ac.uk (T. Liu); menghui.zhou@sheffield.ac.uk (M. Zhou); yu.zhang1@sheffield.ac.uk (Y. Zhang); zhipeng.yuan1@sheffield.ac.uk (Z. Yuan); kang1.liu@polyu.edu.hk (K. Liu); po.yang@sheffield.ac.uk (P. Yang)
ORCID(s):

tasks are added, some tasks will inevitably exhibit significant similarity.

Recently, researchers have begun exploring the role of task similarity in continual learning, aiming to strike a better balance between retaining prior knowledge and facilitating effective transfer. For instance, a teacher-student framework was developed to analytically study CF in two-layer neural networks, demonstrating that intermediate task similarity led to the greatest forgetting while disentangling the roles of feature and readout similarities in transfer and forgetting dynamics [22]. Building upon this, another study further investigated how the interplay between input feature and readout similarity influenced knowledge transfer and retention, finding that high feature similarity with low readout similarity was particularly harmful, while regularisation methods effectively mitigated these challenges [12]. Ultimately, these theoretical insights enable continual learning systems to abandon a one-size-fits-all treatment of tasks and dynamically tailor knowledge protection and transfer strategies to the task similarity.

Building on this idea, continual learning with forgetting avoidance and knowledge transfer (CAT) and soft-masking of parameter-level gradient flow (SPG) are two representative approaches that simultaneously address knowledge protection and transfer [15, 18]. CAT relies on a shared “knowledge base”(KB) and a task similarity detection module. When a new task t arrives, this module creates two models. One reuses the shared base with its KB frozen, and the other is trained from scratch. It compares their performance to label each past task as similar or dissimilar. Similar tasks are unmasked, so t 's gradients can modify their shared weights, while dissimilar tasks remain masked, blocking updates and protecting their parameters. However, as tasks increase, CAT mask memory and compute costs become a bottleneck. SPG computes a soft, parameter-level importance mask from gradients on all previously seen tasks. During training on a new task, each parameter's gradient is softly scaled down in proportion to its importance, which protects old knowledge without completely freezing weights. This scheme scales with a fixed memory cost and balances stability and plasticity across sequences of similar or dissimilar tasks.

Incorporating task similarity detection and soft masking has been shown to enhance the performance and efficiency of continual learning. However, as the number of tasks grows, two key challenges emerge: the overhead of task similarity detection and the precision of protecting past tasks. For each new task, CAT must fine-tune the shared base and train a new model from scratch for every prior task, so the computational cost grows linearly with the number of tasks. Moreover, like other CL strategies, CAT and SPG consider only how past tasks drive forgetting and overlook the fact that new tasks can also cause it. As shown in Fig. 2, the same task 1 suffers different amounts of forgetting depending on task 2's similarity. SPG computes importance scores in a single pass by normalising gradient magnitudes across layers. This makes it vulnerable to noise from small batches. That noise is treated

as a real signal and skews the importance masks, causing them to either over-constrain new task learning or fail to protect old knowledge. Additionally, SPOT[46] has empirically established the link between loss-based similarity and CF. Their work highlights that task similarity is inherently reflected in the magnitude and direction of gradients during the initial phase of learning. However, while SPOT provides an effective tool to quantify the risk of forgetting, it does not propose a concrete mechanism to mitigate this challenge.

In this paper, to tackle these issues, we first propose a loss-based change measure as an alternative to iterative model fine-tuning. The first gradient update on a new task correlates with catastrophic forgetting: the greater the change in loss, the more severe the forgetting. This measure provides new insights into reducing the computational cost of task similarity detection. Secondly, for mitigating CF, we introduce a **Gradient Norm Regularisation(GNR)** approach that brings task similarity into the course of learning. By adding a gradient norm term, GNR slows down the variation of parameters dissimilar to those tasks. After training on a task, parameters that still exhibit gradient activity are incorporated as a soft mask to strengthen long-term memory stability. Experiments on four standardised classification benchmarks show that leveraging task similarity can effectively mitigate CF. An additional evaluation on the 100-class insect classification benchmark [51] demonstrates GNR's superior continual learning performance in realistic settings where tasks share similarities. Overall, we make the following contributions:

- We introduce a measure based on changes in empirical loss to quantify task similarity, instead of relying on existing methods that require two predefined models and repeated fine-tuning. This reduces the computational overhead of task similarity detection.
- We propose a gradient norm regularisation(GNR) approach that leverages task similarity to constrain gradient updates, effectively mitigating catastrophic forgetting while maintaining adaptability to new tasks. GNR could plug into existing regularisation and replay methods to effectively improve the adaptability of continual learning.

2. Related work

2.1. The effect of task similarity

Recent evidence suggested that forgetting is associated with task similarity. By analysing neural network weights, [23, 39, 22] found that forgetting is most severe for tasks with intermediate similarity. For achieving the learning of a mixed sequence of similar and dissimilar tasks, [15] encouraged to learn similar tasks, where there is little forgetting and more positive knowledge transfer. But forgetting mainly affects the learning of dissimilar tasks. Adhering to similar beliefs, [14] introduced a sensitivity measure to find cooperative relations between tasks while mitigating inter-task interference. On the direction of optimisation, TAG

[33] intelligently adapted the learning rate based on task-relatedness. Despite these studies empirically mitigating CF by using task similarity, why dissimilar tasks would lead to CF is largely understudied.

2.2. Task similarity estimation

Recent works to study the calculation of task similarity fall under the categories of data distribution and parameter distribution. First, [23, 39] utilised a data-mixing framework [55] to precisely control the similarity of data distributions to facilitate analysis of their impact on CF. As one of the SOTA methods of this type, optimal transport-based method (OOTD) [2] compare the distances between data distributions for various tasks. Second, the similarity based on parameter distribution is also related to forgetting. [39, 35, 5] employed *Central Kernel Alignment* [19](CKA) as a measure of task similarity to facilitate precise analysis of intrinsic properties of the neural network representation space. And [37] studied the influence of task sequence complexity and its sequential heterogeneity via normalised cosine distance. [26] proposed a task similarity measure based on the angular direction of gradients in the subspace of trust regions, which affects the magnitude of gradient projection. Unlike this, our measure of task similarity is based on task loss, which is more general and easily obtainable without any specifically constructed subspaces. Despite the growing interest in this topic, there is limited understanding of whether these estimation methods can be effectively applied to CL scenarios.

2.3. Mitigating catastrophic forgetting

Recently, a large number of CL methods have been proposed to mitigate catastrophic forgetting. Following the setting of [44, 48], those works can be divided into three main categories.

Regularisation-based methods [17, 1, 25, 9, 42, 27, 28], attempted to stabilise some important parameters or features from past tasks by adding regular terms. Elastic weight consolidation (EWC) [17], used the diagonal terms of *Fisher information matrix* as a proxy for the important network representation. Memory Aware Synapses (MAS) [1] attempted to analyse how sensitive the important parameters from the past tasks were. In addition, inspired by knowledge distillation, Learning without Forgetting (LwF) [25] kept a soft target from prior tasks to guide the new task training. They were all inspired by a Bayesian perspective. With a frequentist opinion, some of the methods [9, 42] tried to use the changes of gradients to revise the model training. For example, Orthogonal Gradient Descent (OGD) [9] emphasised that the gradient update direction of the new task should be orthogonal to the direction of the old task in order to protect the knowledge of the old task. Regularisation-based methods provide a way to mitigate CF in terms of the parameter's importance, they do not require additional costs to retain historical data or parameters and thereby potentially allow them to be integrated with other categories. Recently, a framework [47] introduced a refresh learning mechanism to mitigate over-memorisation by unlearning current task

data before relearning it. Despite both utilising gradient norms, our GNR approach differs by treating the gradient norm as an active supervisory signal rather than a means to flush out unimportant information during the training phase. Furthermore, while the refresh mechanism modifies model parameters via gradient ascent to shed outdated data, GNR operates directly on the gradient itself as detailed in Algorithm 2. By calculating interpolated gradients to identify flat minima and applying parameter-level soft masks to scale the updates, GNR ensures that previous task knowledge is preserved throughout the learning lifecycle.

Memory-based methods [30, 41, 40, 4, 52, 11] work on the idea that maintains a memory of the knowledge from old tasks, which is known as *episodic memory*. It can be regarded as a data-based regularisation method. iCaRL [40] selected and stored exemplars closest to the feature mean of each class by having a fixed memory. Averaged Gradient Episodic Memory (A-GEM) [4] proposed to establish a dynamic task descriptor in new task training, while [43] introduced a deep generative replay model with a generative adversarial network to replay the knowledge from past tasks. While memory-based methods retain key examples from previous tasks, there are challenges with such methods in privacy-related scenarios.

Architecture-based methods [32, 24] aim to find and isolate parameters for specific tasks by model compression so that the subsets of prior tasks' parameters after isolating can guarantee maximal stability. PackNet [32] was able to sequentially prune previous tasks' model parameters into a single network while ensuring minimal decrease in performance by performing iterative pruning and network re-training. To achieve this goal, the network was trained on the available parameters, and then some insignificant parameters were pruned away while keeping the performance of the model from deteriorating significantly. Learn-to-Grow [24] showed an evolving neural structure by separating the neural structure optimisation component and a parameter learning component. However, the isolation parameter triggers a limit on model capacity saturation and blocks the ability of knowledge transfer. Additional architecture expansion requires substantial computation costs, constraining practical applications.

3. Task similarity measure for CL

Motivation: CAT represents the state of the art in estimating task similarity for continual learning by using a transfer network alongside a reference network. Although effective, the repeated training of these networks imposes substantial computational overhead. Our analysis of CAT shows that its similarity module essentially measures task transferability. Inspired by this insight, we propose approximating the similarity between a new task and all previous tasks using the change in empirical loss on those tasks during the first gradient update. *We hypothesise that tasks exhibiting larger loss changes during fine-tuning are less similar to*

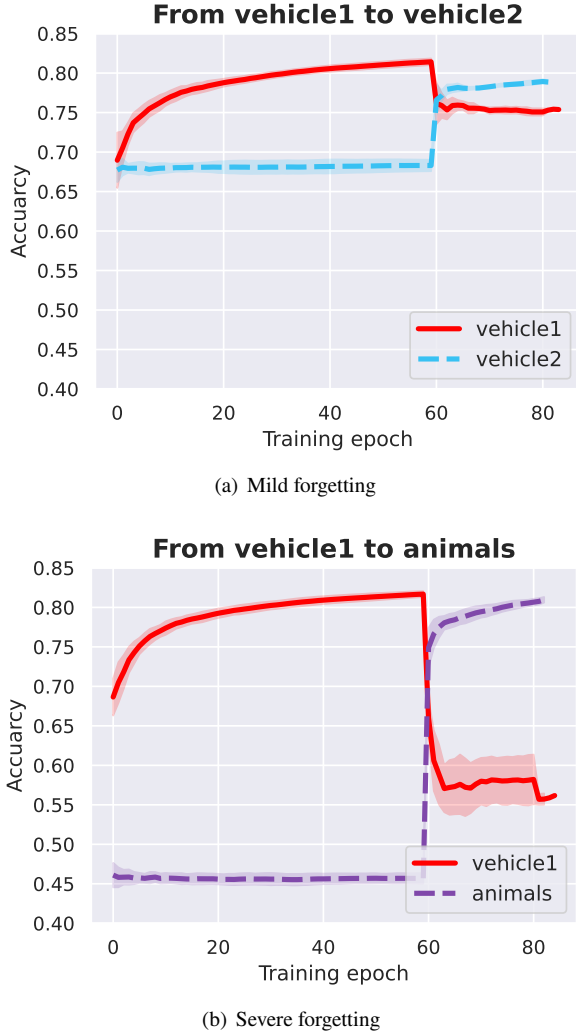


Figure 2: Two-stage accuracy curves. The solid red line tracks performance on the first task (vehicle 1) and the dashed blue/purple line tracks the second task (vehicle 2 in (a), animals in (b)). In (a), introducing a highly similar task leads to gains on vehicle 2 with only a small drop in vehicle 1 accuracy. In (b), introducing a dissimilar task still drives fast learning of animals but causes substantial forgetting of vehicle 1.

earlier tasks, which in turn leads to more severe catastrophic forgetting.

To validate this hypothesis, we conduct the following experiment. We use the Split CIFAR10 dataset, which includes five binary classification tasks (see Supplementary Section S3 for details). First, we train a model on the vehicle classification task as stage one. Then, in stage two, we train the same model on each of the other binary tasks, such as animal classification or other vehicle types. Figure 2 shows the resulting performance and forgetting on the first task.

Measure: Following prior work[22, 36], we consider a two-stage task sequence: given t tasks, we first learn task 1, and then fit the remaining $t - 1$ tasks, separately. We examine how this second stage of training impacts the forgetting of task 1. The amount of task forgetting can be characterised as

the changes of empirical loss for task 1:

$$\mathcal{F} \triangleq L_1(w_2^*) - L_1(w_1^*) \quad (1)$$

where w_1^* and w_2^* are the optimum parameters after training has been completed for the first and second tasks sequentially.

However, obtaining $L_1(w_2^*)$ is infeasible at point w_1^* , which is also incompatible with our goal. Next, we will show an effective alternative to solve this problem.

Let $\mathcal{D}_1$ be the task 1 data. For a given training batch $\mathcal{B}$ from task 2, we define the w_2 to represent the parameters after a step gradient update. The similarity measure $\mathcal{S}$ from task 1 to task 2 is defined as follows:

$$\text{sim}(1 \rightarrow 2) = 1 - \frac{\mathcal{L}_1(\mathcal{B}, w_2)}{\mathcal{L}_1(\mathcal{D}_1, w_1) + \epsilon} \quad (2)$$

$$\mathcal{S}_{1 \rightarrow 2} = e^{\mu * \text{sim}(1 \rightarrow 2)} \quad (3)$$

Equation 2 evaluates the change of loss in the first gradient update of the new task before and after training. A small constant ϵ is added to the denominator to improve numerical stability and avoid division by zero. Equation 3 aims to normalise the similarity, and μ is the adjustment factor. Noting that the $\text{sim}(1 \rightarrow 2)$ is commonly negative at the learning beginning of a new task, implying a negative transfer of learning.

Furthermore, we conduct the following theoretical analyses.

Claim: Let the old task $\mathcal{L}_1(\mathcal{D}_1, w_1)$ around the neighbourhood of the local minimum be the Lipschitz continuous and β -smooth loss function. Given that task 1 as the old task, task 2 induces higher similarity than task 3 on task 1, separately we have:

$$\|g_2\| \leq \|g_3\| \quad (4)$$

where g_2 and g_3 are the gradient of the task 2 and task 3, separately. The proof will be provided in Supplementary Material S1.

From the claim presented above, it follows that lower task similarity leads to larger gradient norms, which in turn increases the amount of forgetting. The theoretical bound in Eq. 4 establishes a monotonic relationship between task similarity and gradient norm under β -smoothness and near-stationarity assumptions. This motivates the practical similarity score in Eq. 2–3: rather than computing the exact bound (which requires the unknown smoothness constant β), we use the normalised change in empirical loss after one gradient step as an efficiently computable, monotonic proxy for the theoretical interference bound. The exponential normalisation in Eq. 3 further maps this proxy to a bounded similarity score. From an optimisation perspective, the gradient norm is correlated with the curvature of the loss function for new tasks, with larger gradient norms indicating greater differences between old and new tasks. On the other hand, prior study[36] similarly suggests that sharper loss surfaces (corresponding to higher curvature) are

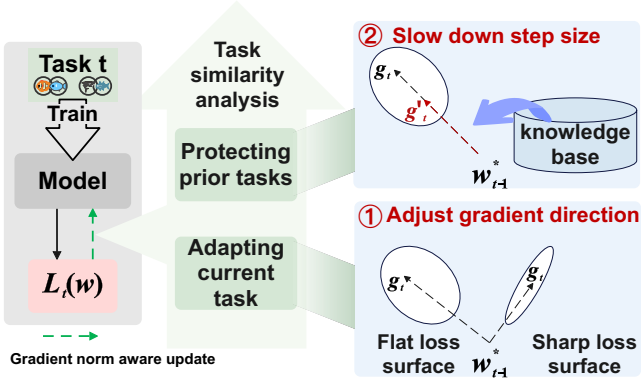


Figure 3: Illustration of gradient norm regularisation (GNR): The process starts by adjusting the gradient direction to avoid sharp loss surfaces, which promote stable adaptation to the new task. Second, the step size is slowed to protect prior knowledge, preventing interference from the new task's updates on previously important parameters.

more prone to catastrophic forgetting. In the next section, we will present a simple yet effective plug-in approach that balances knowledge retention and adaptability in continual learning systems by controlling the gradient norm.

4. Proposed approach: GNR

In this section, we first introduce the problem formulation in CL. We then propose the gradient norm regularisation (GNR) algorithm in Sec.4.2 and deconstruct its components as gradient-norm-aware update and gradient-norm-aware consolidation in Sec.4.3 and Sec.4.4 respectively.

4.1. Problem formulation

Consider a supervised CL problem where we receive a stream of the tasks in sequential manners: $(x_1, y_1), \dots, (x_t, y_t), \dots, (x_{\mathcal{T}}, y_{\mathcal{T}})$, where x_t is the input data and y_t is the ground truth label of task t . Let w_t be the set of model representation(weights) after training on task t . Our training goal is to learn $w_{\mathcal{T}}$, minimizing the training objective $\mathcal{L}_{\mathcal{T}}(w_{\mathcal{T}}) \in \mathbb{R}$, without significantly worsening loss on the previous tasks $\mathcal{L}_{i \in \{1, \dots, \mathcal{T}-1\}}(w_{\mathcal{T}})$. Cross-entropy is utilised in classification tasks.

4.2. Algorithm of GNR

Armed above observation in Sec. 3, to reduce forgetting, we restrict the updating of those key parameters that have lower task similarity. To achieve that, the key of the proposed approach is to consider the gradient norm as an essential measure for estimating task similarity. The GNR approach is composed of two modules (Fig. 3). Specifically, (i) The module of gradient-norm-aware update adds a gradient norm penalty during the new task t learning. We have the gradient update:

$$g = g_t + \lambda \nabla_w \|g_t\| \quad (5)$$

Algorithm 1 Gradient norm-based regularisation

Input: $\mathcal{D}^{train}, \eta, \alpha, r, \gamma$

Output: $\hat{w}$

```

1: Initialise:  $w \leftarrow w_0, \gamma \leftarrow \gamma^{\leq 0}$ 
2: for  $t = 1, 2, \dots, \mathcal{T}$  do
3:   while Until convergence do
4:     Get batch data pairs  $B_t \leftarrow (X_t, Y_t) \sim \mathcal{D}_t^{train}$ 
5:     Get gradient  $g'_t \leftarrow GU(B_t, w, \alpha, r, \gamma^{\leq t-1})$ 
6:     SGD optimizer  $w'_t \leftarrow w_t - \eta g'_t$ 
7:   end while
8:   Update knowledge base  $\gamma^{\leq t} = GC(\mathcal{D}_t^{train}, w'_t)$ 
9: end for
10: return Final optimisation  $\hat{w}$ 
    
```

where $g_t = \nabla_w \mathcal{L}_t(w)$ denotes the gradient of parameter w and λ be the regularisation term. Due to the constraint on the gradient norm, the model is encouraged to learn a smoother gradient direction on a flatter loss surface. A flatter loss surface is less sensitive to changes in parameters, leading to more stable learning and smoother updates. This helps the model adapt more effectively to the new task. Secondly, in the new gradient direction, there may be parameters that are important to previous tasks. Directly updating these parameters could lead to catastrophic forgetting. To prevent this, we consult the knowledge base ($\gamma^{\leq t-1}$) to assess the importance of these parameters, and then slow down their updates accordingly:

$$g' = (1 - \gamma^{\leq t-1})g \quad (6)$$

where $\gamma^{\leq t-1} \in [0, 1]$ and $\gamma^{\leq t-1} = \text{soft_masking}(\|g\|)$ denotes the consolidated crucial gradient norm for the previous tasks from 1 to $t-1$. Details will be found in section 4.4.

(ii) The module of gradient-norm-aware consolidation accumulated the crucial gradient norms by using the soft-masking technique after learning was finished. Soft-masking [18, 16] is a parameter-level penalty to restrict the important gradient flow for old tasks. In this work, we regard the gradient norm as an instance of a gradient flow, following the parameters operation of the soft-mask design. In cognitive science, people tend to remember unexpected events, such as surprises [34]. In our experiments, we observe that certain samples still produce large gradient norms in the trained model. Inspired by this, instead of saving these 'surprising' samples directly, we record their gradient norms, thereby maintaining a knowledge base.

In conclusion, GNR provides a unified protection mechanism, leveraging gradient norm as a signal to preserve past task knowledge while integrating connections to new tasks, ensuring effective adaptation to new tasks while minimising CF. Algorithm 1 illustrates how GNR works.

4.3. Gradient-Norm-Aware update

In the Gradient-Norm-Aware update(GU) module, we induce the gradient to change its direction and size when the gradient update of the current task is not similar to that of the previous task.

(1) Adjusting gradient direction. First, we evade updating the gradient in a direction on the sharp loss surface by constraining the gradient norm. Inspired by previous research [36], a flat loss surface helps reduce the amount of forgetting. We then transform this constraint problem into a sharpness-aware minimisation (SAM) optimisation. For achieving flatness, SAM scheme [10, 56] has become a simple yet effective technique to search for flat optimal minima by adding parameter perturbation. We note that adding the gradient norm penalty to the loss function does not directly adjust the magnitude of the gradient. Instead, differentiating this augmented loss with respect to the parameters yields the derivative of the gradient norm (Eq. 5), which involves the Hessian matrix and therefore effectively constrains the curvature of the loss landscape. Together with the motivation in Eq. 4, minimising the gradient norm simultaneously reduces the upper bound on inter-task interference.

Motivated by the above studies, we induce the gradient norm to find a flat task minima. In the optimisation procedure, we need to calculate the gradient of Eq. 5 as follows:

$$\nabla_w \mathcal{L}(w) = \nabla_w \mathcal{L}_t(w) + \lambda \nabla_w (\|\nabla_w \mathcal{L}_t(w)\|) \quad (7)$$

$$= \nabla_w \mathcal{L}_t(w) + \lambda \nabla_w^2 \mathcal{L}_t(w) \frac{\nabla_w \mathcal{L}_t(w)}{\|\nabla_w \mathcal{L}_t(w)\|} \quad (8)$$

where $\nabla_w^2 \mathcal{L}_t(w)$ represents the Hessian matrix, and the simplification of Eq. 7 to Eq.8 follows the chain derivation rule.

However, Eq.8 involves the computation of the Hessian matrix, which is computationally infeasible in the high-dimensional space of neural networks. Inspired by [56], Taylor second-order expansion was employed to approximate the Hessian matrix at the w . Adding a subtle perturbation Δw at gradient $\nabla_w \mathcal{L}_t(w)$, we have :

$$\nabla_w \mathcal{L}_t(w + \Delta w) = \nabla_w \mathcal{L}_t(w) + \nabla_w^2 \mathcal{L}_t(w) \Delta w + \mathcal{O}(\|\Delta w\|^2) \quad (9)$$

For getting the second term approximation in Eq.8, Let $\Delta w = r\mathbf{v}$ where r is perturb ratio and $\mathbf{v} = \frac{\nabla_w \mathcal{L}_t(w)}{\|\nabla_w \mathcal{L}_t(w)\|}$. Therefore, integrating Eq.9 into Eq.8, we have:

$$\begin{aligned} \nabla_w \mathcal{L}(w) &= (1 - \alpha) \nabla_w \mathcal{L}_t(w) \\ &+ \alpha \nabla_w \mathcal{L}_t(w + r \frac{\nabla_w \mathcal{L}_t(w)}{\|\nabla_w \mathcal{L}_t(w)\|}) \end{aligned} \quad (10)$$

where $\alpha = \frac{\lambda}{r}$ represents the balance ratio, aiming to balance the penalty and perturbation range. We give the detailed procedures for the simplification of Eq.10 in Supplementary Material S2.

(2) Slowing down the step size of the gradient. When a gradient direction corresponds to a flat loss surface, directly updating in that direction can still lead to catastrophic forgetting, particularly in long-sequence tasks. This occurs because such directions may contain important information about past tasks, which backwards propagation tends to modify. To address this, we propose step-size adaptation

Algorithm 2 Gradient-Norm-Aware update (GU)

Input: $B_t, w, \alpha, r, \gamma^{\leq t-1}$

Output: g_t

- 1: Calculate the first gradient $g_1 = \nabla_w \mathcal{L}_t(w)$
 - 2: Add perturbation $r \frac{g_1}{\|g_1\|}$ on parameter w so that
 $w' = w + r \frac{g_1}{\|g_1\|}$
 - 3: Calculate the second gradient $g_2 = \nabla_w \mathcal{L}_t(w')$
 - 4: Calculate interpolated gradient $g_t = (1 - \alpha)g_1 + \alpha g_2$
 - 5: # Slow down the step size of the gradient:
 - 6: **for** layer of feature extractor $l = 1, 2, \dots, L$ **do**
 - 7: $g'_t \leftarrow (1 - \gamma_l^{\leq t-1})g_t$
 - 8: **end for**
 - 9: **return** Task similarity aware gradient g'_t
-

Algorithm 3 Gradient-Norm-Aware consolidation (GC)

Input: D_t^{train}, w'_t

Output: $\gamma^{\leq t}$

- 1: **for** $\tau = 1, 2, \dots, t$ **do**
 - 2: Compute loss $\mathcal{L}^{t,\tau}$ of task τ at training task t in Equation 11
 - 3: **for** layer $l = 1, 2, \dots, L$ **do**
 - 4: $\gamma^{t,\tau} \leftarrow$ Equation 12
 - 5: **end for**
 - 6: **end for**
 - 7: **for** layer $l = 1, 2, \dots, L$ **do**
 - 8: $\gamma_l^t \leftarrow \max(\gamma_l^{t,1}, \dots, \gamma_l^{t,t}), \gamma_l^{\leq t} \leftarrow \max(\gamma_l^t, \gamma_l^{\leq t-1})$
 - 9: **end for**
 - 10: **return** knowledge base $\gamma^{\leq t} = \{\gamma_l^{\leq t}\}_{l=1}^L$
-

strategies that slow down parameter changes in these crucial directions. Specifically, we utilise the gradient norm from historical tasks as a measure of parameter importance through a soft-masking technique, thereby reducing the update step of the gradient:

$$\nabla_w \mathcal{L}(w)' = (1 - \gamma^{\leq t-1}) \nabla_w \mathcal{L}(w)$$

where $\gamma^{\leq t-1}$ serves as a knowledge base, recording the importance of each parameter for the past $t - 1$ tasks. The generation of this will be explained in the following section. Algorithm 2 provides the work procedures of the gradient-norm-aware update.

4.4. Gradient-Norm-Aware consolidation

In this section, we calculate the important gradient norm in the soft-masking for the previous $t - 1$ tasks and update the soft-masking value as the previous t tasks. When $t \geq 2$, the soft-masking module is activated and we now show its computation.

For the previous t tasks' model $\mathcal{M}_\tau$ ($1 \leq \tau \leq t$), we calculate their losses with $(X_t, Y_t) \sim D_t^{train}$ separately as follows:

$$\mathcal{L}^{t,\tau}(w) = \begin{cases} \mathcal{L}(\mathcal{M}_\tau(X_t), Y_t) & (\tau = t) \\ \text{Sum}(\mathcal{M}_\tau(X_t)) & (\tau < t) \end{cases} \quad (11)$$

where $\mathcal{L}^{t,\tau}(w)$ denotes the task τ 's loss when task t learning is finished. For the current task $\tau = t$, cross-entropy loss for classification is utilised as $\mathcal{L}^{t,t}$. For the previous task $\tau < t$, the loss $\mathcal{L}^{t,\tau}$ is defined as the sum of the logits of the previous tasks. The gradient norm-based soft-masking is z -scored as standard normal distribution to remove the scale between different layers:

$$\gamma_l^{t,\tau} = \left| \tanh \left(\text{Norm} \left(\|\nabla_{w_l} \mathcal{L}^{t,\tau}(w_l)\| \right) \right) \right| \quad (12)$$

$$\text{Norm}(x) = \frac{x - \text{Mean}(x)}{\sqrt{\text{Var}(x)}} \quad (13)$$

For achieving the memory stability of the model, the gradient norm-based soft-masking will be accumulated by means of calculating $\max()$ on the parameter level (as shown in line 8 of the Algorithm 3).

Noting that we only need to maintain the soft-masking size the same as the number of parameters, regardless of the number of tasks, since the value $\gamma^{\leq t}$ in each task update is the gradient norm maximum of the tasks seen so far. Algorithm 3 provides the work procedures of gradient-norm-aware consolidation.

5. Experiments

5.1. Experimental setup

Benchmarks. We perform our approaches on four benchmark datasets for CL, where “-t” denotes the total of t tasks created from the original dataset (t takes 10 or 20):

(1) **CIFAR100-t**: CIFAR100 [20] is a dataset that has images of 100 classes, which are grouped into 20 super-classes. We split the 100/ t classes equally into t tasks.

(2) **TinyImageNet-t**[49]: TinyImageNet contains 100K images of 200 classes (500 for each class) downsized to 64×64 coloured images from ImageNet. Each class has 500 training images, 50 validation images and 50 test images. We split the 200/ t classes equally into t tasks.

All experiments follow the Task-Incremental Learning (Task-IL) scenario [45]: the task identity is known at inference time and the model predicts only among classes belonging to the current task.

Evaluation Metrics. Following the existing CL works, we report the following metrics:

(1) **Average accuracy** can be defined as $A_{\mathcal{T}} = \frac{1}{\mathcal{T}} \sum_{i=1}^{\mathcal{T}} a_{\mathcal{T},i}$, where $\mathcal{T}$ is the total number of tasks seen so far, and $a_{t,i}$ is the accuracy on the task i after the model has learnt task t . It represents the average of accuracy after the model has learnt $\mathcal{T}$ tasks.

(2) **Average forgetting** represents the maximum gap between the best accuracy and the final accuracy of each task. It is calculated by $F = \frac{1}{\mathcal{T}-1} \sum_{i=1}^{\mathcal{T}-1} \max_{t \in \{1, \dots, \mathcal{T}-1\}} (a_{t,i} - a_{\mathcal{T},i})$ when the model has learnt $\mathcal{T}$ tasks.

Training baselines and details.

We compare 6 baselines. 4 regularisation methods EWC [17], SI [54], Lwf [25] and SPG [18]. Additionally, Naive fine-tunes greedily train a sequence of tasks without dealing

Table 1

DeepLabV3-Resnet50 on peach tree disease detection

Tasks	Similarity	Accuracy	Forgetting
1 → 2	0.58	86.69±1.73	13.32±2.33
1 → 3	0.12	85.76±1.58	14.15±2.14
1 → 4	0.47	85.60±1.43	15.14±2.34
1 → 5	0.13	85.90±1.74	14.59±1.91

with forgetting. Multi-task learning (MTL) is jointly learning all tasks, which has no forgetting and is usually considered to be the upper bound of performance. Furthermore, to evaluate the versatility of the proposed plug-in, GNR was combined with three representative memory-based techniques, namely replay, A-GEM[4], and DER++[3].

Alexnet [21] is employed as the backbone and stochastic gradient descent as the optimiser. The batch size is 64. 5 models with different random seeds are trained for each benchmark. Each seed determines a task order, which is used to evaluate the sensitivity of different algorithms to the task order. All results are reported as average $\pm$ standard deviation. For a fair comparison, EWC, SI, and Lwf are reproduced using the avalanche library [29].

5.2. The effectiveness of the proposed measure

In this section, we analyse the flexibility and model-diagnostics of the proposed measure and provide some inspiring insights. For ease of observation, we focus on the two-phase CL and analyse the effect of task similarity by fixing the first task.

Training protocol. We split CIFAR10 into five tasks, each of which performs a 2-classification. We compared 4 types of baseline methods for evaluating task similarity: Optimal transport distance (OTDD) [2] is to measure distances in the data distribution space. Cosine similarity is to calculate model weights (Cosine). central kernel alignment (CKA) [19] computes similarity in the neural network representation space. Domain similarity [7] is to measure domain-specific similarity in feature space. More information about the task settings can be found in Supplementary Material S3.

Task similarity estimation on Split CIFAR10. As shown in Fig. S2(a) in Supplementary Material, we compared the similarity on different tasks(2,3,4 and 5) during phase-2 learning by using different measures. CKA does not distinguish similarities between tasks This may be due to the fact that the requirement for the calculation of similarity is at the start of the second phase of learning. Other methods are reasonably distinct for different tasks. So, how effective is this distinguishing effect on the model's accuracy and forgetting at the end of the second phase of training? As shown in Fig. S2(b) in Supplementary Material, although cosine similarity has some relevance to the prediction of accuracy, it is difficult to predict the probability of forgetting occurring. This may be due to the direction of the gradient changing with uncertainty during the learning of new tasks. Domain and ours are better than others in correlation. Estimating values are highly positively correlated with accuracy and highly

Table 2
Summarised results on four benchmarks

Methods	C-10		C-20		T-10		T-20	
	Accuracy $\uparrow$	Forgetting $\downarrow$	Accuracy $\uparrow$	Forgetting $\downarrow$	Accuracy $\uparrow$	Forgetting $\downarrow$	Accuracy $\uparrow$	Forgetting $\downarrow$
Naive	46.45 $\pm$ 3.86	25.85 $\pm$ 6.53	49.39 $\pm$ 3.96	31.78 $\pm$ 2.38	34.80 $\pm$ 1.60	19.96 $\pm$ 2.00	36.21 $\pm$ 2.29	24.07 $\pm$ 1.68
EWC[17]	63.88 $\pm$ 2.08	16.77 $\pm$ 3.14	45.66 $\pm$ 3.45	37.75 $\pm$ 3.14	34.22 $\pm$ 2.34	20.74 $\pm$ 1.41	37.83 $\pm$ 1.94	20.51 $\pm$ 2.73
SI[54]	55.99 $\pm$ 5.75	20.10 $\pm$ 6.82	43.71 $\pm$ 12.09	35.96 $\pm$ 2.76	40.02 $\pm$ 1.34	16.30 $\pm$ 1.22	46.90 $\pm$ 1.67	18.05 $\pm$ 2.58
MAS[1]	58.46 $\pm$ 3.73	2.96$\pm$1.62	53.54 $\pm$ 16.89	4.89$\pm$1.74	47.52 $\pm$ 1.91	2.02$\pm$1.43	53.44 $\pm$ 1.55	4.94 $\pm$ 0.95
LwF[25]	67.83 $\pm$ 2.87	10.83 $\pm$ 2.58	66.75 $\pm$ 4.62	16.39 $\pm$ 3.43	47.10 $\pm$ 1.80	6.67 $\pm$ 1.45	44.32 $\pm$ 2.14	17.61 $\pm$ 1.31
SPG[18]	64.61 $\pm$ 1.74	8.26 $\pm$ 1.29	65.43 $\pm$ 0.98	8.74 $\pm$ 1.69	46.71 $\pm$ 1.87	4.96 $\pm$ 1.16	58.51 $\pm$ 0.90	4.43 $\pm$ 0.44
DER++[3]	71.20 $\pm$ 0.50	6.60 $\pm$ 0.50	76.00 $\pm$ 0.70	7.00 $\pm$ 1.10	48.80 $\pm$ 1.50	6.90 $\pm$ 0.60	56.80 $\pm$ 1.80	6.80 $\pm$ 0.50
GNR(Ours)	72.99$\pm$1.60	5.39 $\pm$ 0.60	74.72$\pm$1.72	6.21 $\pm$ 1.32	50.06$\pm$1.13	3.49 $\pm$ 0.49	59.22$\pm$1.35	3.28$\pm$0.71
MTL	73.11 $\pm$ 5.49	-	78.52 $\pm$ 1.97	-	49.53 $\pm$ 4.09	-	60.95 $\pm$ 2.37	-

For accuracy, the higher the better $\uparrow$. For forgetting, the lower the better $\downarrow$.

They are shown in per cent. The best CL approach for each benchmark is **bolded**.

negatively correlated with forgetting. However, domain similarity needs to provide a pre-trained model which provides a good mapping of the feature space for the computation of similarity. In our experiments of Fig. S2 in Supplementary Material, we used Resnet18 based on ImageNet pre-training as the backbone. Despite domain similarity got good performance in two-phase continual learning, it is difficult to adapt itself to a continual learning environment. The proposed method's estimation of task similarity is calculated exactly on the basis of the difference in distribution with previous tasks in the feature space. Compared to domain similarity, which uses all the samples to estimate the similarity, the proposed method is plug-and-play, requiring only one batch training.

Task similarity estimation in real-world applications. In this part, we employ a real-world agriculture dataset for peach tree disease detection to prove the effectiveness of the proposed measure. This consists of five tasks that are nearly identical in terms of image style and classification target, with the exception of their shooting time. We aim to observe the pattern of disease development over time. More details about the dataset will be found in Supplementary Material S3.

Table.1 shows the result of functional similarity with accuracy/forgetting. When we consider task 1 (shooting date: 27/07/21) as the first task in learning phase 1, task 2(shooting date: 14/09/21) has the highest functional similarity, accuracy and the lowest amount of forgetting. This implies that the tasks with the closest shooting dates have the highest similarity and are easier to transfer knowledge. The other tasks were shot in 2022, so the similarity was reduced. Additionally, task 4 (the same month in 2022) exhibits a similar seasonal similarity. we utilise the DeepLabV3-ResNet50 [6] as the network architecture. In summary, the proposed measure is model-agnostic. Additionally, we observed a pattern in the development of peach tree diseases as they progressed over time: Disease distribution is most similar at close time points, and disease occurrence has a pattern of annual periods.

5.3. Comparison with other baselines

Table. 2 reports the average accuracy and average forgetting on four benchmarks. This result revealed that our proposed method is generally better than other regularisation baselines. Except for C-20, the accuracy of GNR remained within 1% distance of multi-task learning, which implies that the representation learned by GNR has favourable generalisation properties. In addition, we took a different task order of continual learning in the five randomised trials, which led to a high standard deviation in accuracy across methods. Despite our efforts, it was difficult to avoid training failures on tasks that appeared in some sequences. Sensitivity to task sequencing was shown in results with large standard deviations: The accuracy of MAS was 53.54 ± 16.89 on C-20. The accuracy of SI was 43.71 ± 12.09 on C-20. In contrast, GNR performed stably for different task sequences, sharing the lowest standard deviation of accuracy on the four benchmarks. Note that MAS has the lowest forgetting on the C-10, C-20 and T-10 as a result of MAS being less adaptable to new tasks. This suggests that MAS limits the model's adaptability. It is worth noting that GNR was not tuned for hyperparameters and was uniformly set with $r = 0.02$ and $\alpha = 0.3$. For the other baselines, we adjusted the hyperparameters according to the settings reported in the respective papers. We also include DER++ [3] in Table 2 for cross-category comparison. GNR surpasses DER++ in accuracy on three out of four benchmarks and achieves lower forgetting on all four, despite not maintaining any memory buffer.

5.4. Using GNR with other baselines

A key property of GNR is that it operates as a plug-in regularisation term. To verify this compatibility, we examine whether GNR can be effectively combined with replay-based methods, which represent the dominant family of CL approaches. In this section, we used the weight regularisation method EWC and the function regularisation method Lwf as a baseline. Meanwhile, we introduced three representative memory-based techniques, namely replay, A-GEM[4], and DER++[3]. Furthermore, we incorporate the refresh [47] mechanism as a robust baseline plug-in. This approach is

Table 3

Accuracy: GNR brings improvements on 4 benchmarks

Methods	C-10			C-20			T-10			T-20		
	Accuracy	Δ	Δ_{mtl}	Accuracy	Δ	Δ_{mtl}	Accuracy	Δ	Δ_{mtl}	Accuracy	Δ	Δ_{mtl}
EWC	63.8±2.0			45.6±3.4			34.2±2.3			37.8±1.9		
EWC+GNR	72.6±0.6	8.7↑	-0.5	73.9±1.2	28.2↑	-4.5	50.7±0.6	16.4↑	1.1	59.8±0.9	22↑	-1.1
LwF	67.8±2.8			66.7±4.6			47.1±1.8			44.3±2.1		
LwF+GNR	75.3±1.1	7.4↑	2.2	76.5±1.8	9.7↑	-2.0	53.5±0.6	6.4↑	4	61.5±0.8	17.2↑	0.6
Replay	69.8±0.9			78.5±0.8			42.5±2.1			50.4±2.6		
Replay+GNR	73.6±0.8	3.7↑	0.5	79.3±1.1	0.8↑	0.8	46.7±1.3	4.1↑	-2.8	55.2±1.8	4.7↑	-5.6
A-GEM	64.0±1.0			60.7±3.9			40.7±1.4			46.4±2.5		
A-GEM+GNR	74.3±1.6	10.3↑	1.2	76.8±1.3	16↑	-1.6	51.7±1.1	10.9↑	2.1	60.0±1.4	13.5↑	-0.9
DER++	71.2±0.5			76.0±0.7			48.8±1.5			56.8±1.8		
DER++ + GNR	71.8±0.6	0.6↑	-1.2	78.4±0.4	2.3↑	-0.1	49.8±1.3	1.0↑	-0.3	58.3±1.8	1.5↑	-2.6
DER++ + refresh	72.6±0.7	1.3↑	-0.4	79.7±0.5	3.6↑	1.2	48.0±1.6	-0.8	-1.5	56.6±1.8	-0.2	-4.3

$\Delta = ACC_{method+GNR} - ACC_{method}$ represents the performance gain GNR brings to the method.

$\Delta_{MTL} = ACC_{method+GNR} - ACC_{MTL}$ denotes the gap between the approach to integrating the GNR and the MTL

A positive Δ represents the performance improvement.

A positive Δ_{MTL} denotes the performance beyond the level of MTL.

Table 4

Forgetting: GNR brings improvements on 4 benchmarks

Methods	C-10		C-20		T-10		T-20	
	Forgetting	Δ	Forgetting	Δ	Forgetting	Δ	Forgetting	Δ
EWC	16.8±1.7		37.8±3.1		20.7±1.4		20.5±2.7	
EWC+GNR	4.5±0.2	-12.3↑	4.3±1.3	-33.4↑	3.5±0.2	-17.2↑	2.8±1.1	-17.7↑
LwF	10.8±2.6		16.4±3.4		6.7±1.5		17.6±1.3	
LwF+GNR	0.9±0.3	-9.9↑	1.2±0.4	-15.2↑	-0.2±0.3	-6.9↑	0.6±0.5	-17.0↑
Replay	2.8±4.1		3.5±0.9		10.1±1.1		11.0±1.0	
Replay+GNR	-1.0±0.6	-3.8↑	-1.0±0.4	-4.5↑	3.1±0.6	-7.0↑	3.3±0.3	-7.7↑
A-GEM	11.1±0.6		18.0±4.3		13.8±0.6		16.2±1.9	
A-GEM+GNR	1.0±0.9	-10.0↑	1.0±1.0	-17.0↑	0.8±0.5	-12.9↑	1.1±0.5	-15.2↑
DER++	6.6±0.5		7.0±1.1		6.9±0.6		6.8±0.5	
DER++ + GNR	2.6±0.6	-3.9↑	1.2±0.4	-5.8↑	2.0±0.3	-4.8↑	1.6±0.4	-5.2↑
DER++ + refresh	7.3±0.3	0.7	4.8±0.6	-2.2↑	7.0±0.7	0.1	7.2±0.7	0.3

$\Delta = Forgetting_{method+GNR} - Forgetting_{method}$ denotes the performance gain GNR brings to the method.

A negative Δ indicates a decrease in the amount of forgetting.

designed to mitigate the risk of over-memorisation by unlearning current task data before the relearning process. Our aim in this section is to verify that GNR can be used as a plug-in strategy in combination with many CL algorithms. Table.3 and 4 report the accuracy and forgetting results, respectively, for the integration of GNR into several continual learning algorithms. The results demonstrate that GNR consistently improves performance by increasing accuracy and reducing forgetting. For example, **A-GEM+GNR** showed a significant improvement in accuracy on C-10 (+10.3%) and C-20 (+16%), while also reducing forgetting on C-10 by -10.0%. Similarly, **Replay+GNR** achieved notable gains in accuracy and a reduction in forgetting, particularly on T-10. Additionally, the positive values of Δ_{MTL} across many methods indicate that GNR enhances the model's ability to generalise across multiple tasks, contributing to

a performance boost akin to that of multi-task learning. A notable example is **LwF+GNR**, which exhibited a Δ_{MTL} of 2.2% on C-20, reflecting a substantial improvement in generalisation. These findings highlight the effectiveness of GNR in enhancing continual learning methods, making them more robust in handling sequential tasks while retaining previously acquired knowledge.

We introduce the refresh learning plugin into the DER++ baseline to analyse its impact on the stability-plasticity trade-off. While the refresh mechanism demonstrates considerable plasticity through notable accuracy gains in simpler baselines such as C-10 and C-20, its efficacy diminishes in more complex task sequences (T-10, T-20) where accuracy gains eventually regress. Crucially, the unlearn-relearn strategy frequently fails to stop knowledge erasure, which results in an actual increase in CF across several benchmarks. In

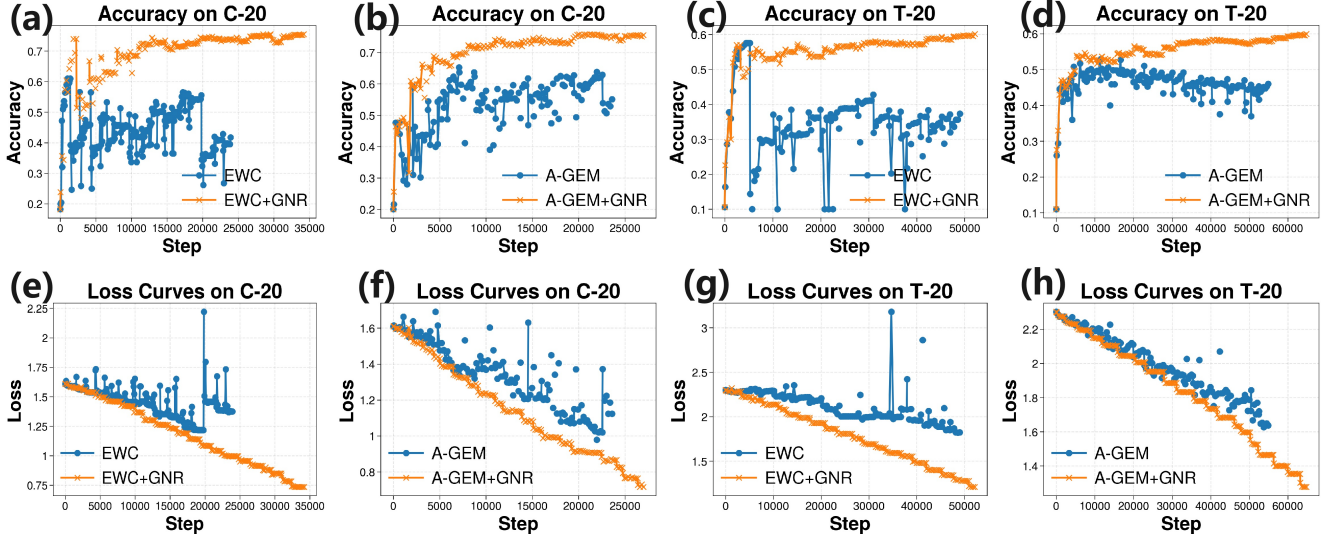


Figure 4: Adaptation analysis. Accuracy and loss curves for various continual learning algorithms (EWC and A-GEM) with and without the integration of GNR across different datasets (C-20 and T-20). Panels (a) and (b) show the accuracy on C-20, (c) and (d) show the accuracy on T-20, (e) and (f) display the loss curves for C-20, and (g) and (h) show the loss curves for T-20. The results illustrate the impact of GNR on improving model performance by mitigating oscillations and accelerating convergence, especially for longer sequences.

contrast, GNR establishes a balance between adaptation and retention by delivering consistent performance improvements while achieving a profound reduction in forgetting across all tested datasets. By operationalising the gradient norm as a supervisory signal, GNR ensures that knowledge preservation scales effectively with increasing task complexity. This robust performance shows the advantage of similarity-aware gradient modulation over methods focused solely on capacity recovery through information flushing. Additionally, refresh learning is suited to buffer-rich scenarios that prioritise the enhancement of overall recognition accuracy. Under resource constraints, its aggressive unlearning mechanism may inadvertently remove the memory of prior tasks that lack sufficient episodic samples to anchor existing knowledge (Supplementary 4).

In summary, our experiments empirically demonstrate that GNR can easily be integrated with other continual learning algorithms. This finding underscores the important role of task similarity, a factor that has been overlooked in many previous studies, suggesting that GNR leverages task relationships to significantly improve continual learning performance.

We note that for some base methods, the combined base+GNR does not surpass standalone GNR (e.g., EWC+GNR on C-10). This is expected: standalone GNR already integrates both GU and GC into a unified pipeline, whereas adding an external regulariser may introduce conflicting constraints that limit its full potential. Nevertheless, GNR as a plug-in *consistently improves* every base method (all $\Delta > 0$ in accuracy across Tables 3–4), confirming its reliability as a drop-in component that consistently strengthens existing CL pipelines.

5.5. Adaptation analysis

In this section, we compare the impact of adding GNR to regularisation-based methods (EWC) and memory-based methods (A-GEM) for long sequences (C-20, T-20), focusing on the model’s adaptability (Fig. 4). The results show that without GNR, both methods struggle to maintain stable performance over time, with noticeable oscillations in the loss and accuracy curves, particularly for the more complex T-20 dataset (Fig. 4). Specifically, the accuracy and loss curves for EWC and A-GEM (blue) exhibit fluctuations and slower convergence. In contrast, when GNR is integrated (orange curves), the models display smoother, more consistent improvements in accuracy and faster loss reduction, especially evident in A-GEM+GNR. This highlights that GNR provides the necessary stabilisation for models to better adapt to long-term learning tasks.

By mitigating oscillations and accelerating convergence, GNR enhances the robustness and efficiency of lifelong learning algorithms, making them more reliable for sequential task learning in real-world applications.

5.6. A case of practical application: pest classification

Pest recognition tasks in agricultural environments often involve long-term, staged learning, and the application of CL helps enhance the model’s stability and generalisation in such contexts. In order to further validate the application of GNR in practical scenarios, we utilised the IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition [50], which contains over 100 insect species commonly found in agricultural settings, offering a diverse set of images for pest classification tasks (Fig. 5). The IP102 dataset offers

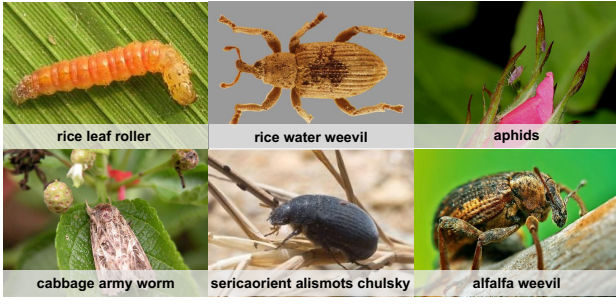


Figure 5: Example images of the IP102 dataset.

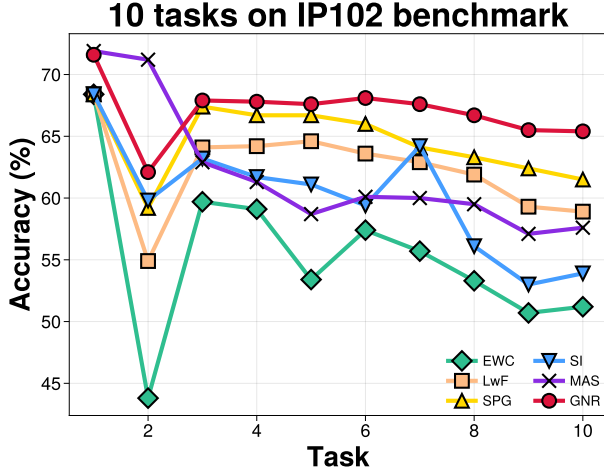
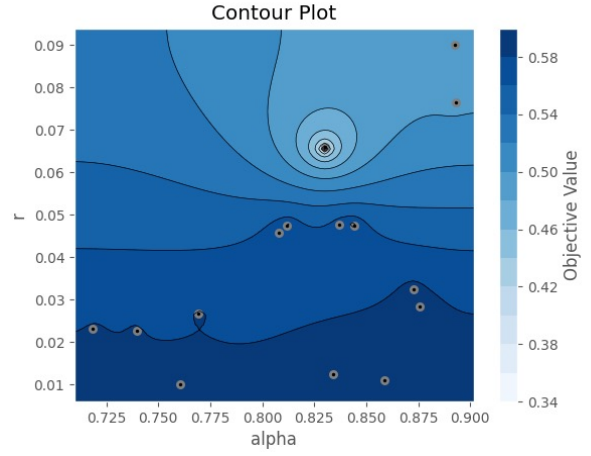
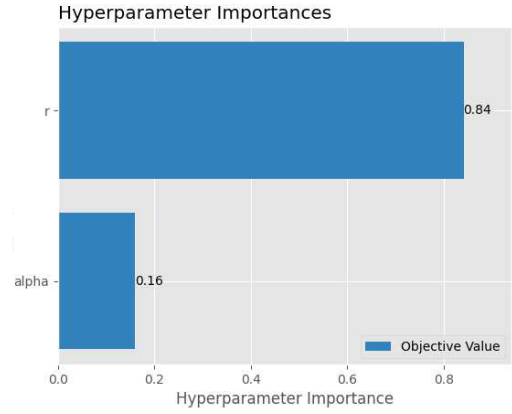


Figure 6: Accuracy comparison of six continual learning methods (EWC, LwF, SPG, SI, MAS, and GNR) on the IP102 benchmark across 10 sequential tasks.

significant advantages, including its large-scale and diverse collection of over 75,000 images across 102 insect species, which supports effective insect pest recognition research. However, it also presents challenges such as class imbalance and intra- and inter-class variability, which require attention to the model's generalisation ability and performance on minority classes in practical applications. In this work, we selected 100 classes from this benchmark and divided them into 10 tasks, with each task consisting of 10 classes. Fig.6 illustrates the average accuracy across tasks for six different regularisation methods on this benchmark. The main findings indicated that, across most tasks, GNR achieved the highest accuracy compared to the other regularisation methods. As the number of tasks increased, GNR showed a relatively smooth and mild decline in accuracy. In contrast, EWC and MAS showed more significant drops in accuracy over time, particularly after the early tasks. SI, SPG, and LwF exhibited similar trends, with LwF outperforming SPG in the later tasks. Notably, for the first task, GNR demonstrated the best performance, effectively learning better representations than the other methods. Overall, the stability of GNR and its smaller performance decline (compared to other methods) make it an ideal choice for long-term



(a) Optimal parameter distribution on T-20



(b) Parameter importance on T-20

Figure 7: Hyper-parameter analysis. r and α are hyper-parameters. The objective value denotes accuracy.

tasks, making it suitable for dynamic environments such as seasonal changes or variations in agricultural settings.

6. Parameter analyses

In this section, we will analyse the parameters of two optimal parameters r and α that are involved in the learning process of gradient-norm-aware update.

For perturb ratio r lays down a search range centred on the task minimum and with a radius range of r . This search range should not be set too large to prevent affecting the gradient descent optimisation process. For balance ratio α aims to balance the penalties between r and λ . Based on the empirical analysis of [10, 56], the effective range of values should be as follows: $r \in [0.01, 0.1]$ and $\alpha \in [0.7, 0.9]$. Following the empirical search range, 15 models with different random seeds are trained for each benchmark.

We report the hyper parameter analysis on T-20 in Fig.7. As shown in the contour plot in Fig.7(a), the optimal solution falls in the range of $r \in [0.01, 0.02]$, while good performance exists for alpha at either 0.7 or 0.8. Furthermore, as shown in Fig. 7(b), we estimate the importance of the parameters using functional analysis of variance [13]. Compared

Table 5

Wall-clock time (seconds) on C-20 and T-20, averaged over five seeds.[†]

Method	C-20		T-20	
	Exp	Epoch	Exp	Epoch
GNR	34.95	0.77	185.12	4.53
MAS	21.45	0.55	38.82	2.26
LwF	30.93	0.60	103.99	3.11
SPG	30.90	0.62	129.37	3.03
EWC	64.08	1.41	199.69	6.52
SI	64.66	1.66	109.35	7.84

[†]Exp: wall-clock time for one experience;
Epoch: mean time for one training epoch.

Table 6

Peak GPU memory (MB) on C-20 and T-20, averaged over five seeds.[†]

Method	C-20		T-20	
	Mean	Max	Mean	Max
GNR	191	191	535	536
MAS	264	274	723	791
LwF	240	245	733	760
SPG	388	645	1101	1795
EWC	915	1654	2577	4555
SI	228	230	651	653

[†]Mean: average peak memory across experiences;
Max: highest peak during the entire run.

to α , the setting of r is critical to model performance. We obtained similar results in other benchmarks.

In summary, r requires careful setting in order to ensure that the model plasticity gains from the appropriate perturbation. The optimal hyper parameter range is as follows: $r \in [0.01, 0.04]$ and $\alpha \in [0.7, 0.85]$.

6.1. Computational overhead analysis

To validate the efficiency claims, Tables 5 and 6 report computational overhead averaged over five seeds for GNR and all regularisation baselines on C-20 and T-20, covering wall-clock time per experience and per epoch, as well as mean and max peak GPU memory. All experiments are conducted on the same hardware with identical batch size and training configuration.

As shown in Tables 5 and 6, GNR achieves the lowest GPU memory among all methods on both benchmarks, with virtually no transient spike (max $\approx$ mean), whereas EWC and SPG exhibit large mean-to-max memory inflation. In terms of per-epoch time, GNR is nearly twice as fast as EWC and SI due to avoiding heavy auxiliary statistics, while its moderate overhead relative to MAS, LwF, and SPG stems from the additional backward pass in the SAM-style update. Note that per-experience time also reflects the number of early-stopping epochs and is therefore reported for completeness rather than as a direct overhead metric.

Overall, GNR offers the most stable and memory-efficient profile among all compared regularisation methods.

6.2. Component-level ablation

To isolate the contribution of each GNR component, we evaluate three variants: (1) **GU-only**, which applies only the gradient-norm-aware update without consolidation; (2) **GC-only**, which applies only the gradient-norm-aware consolidation (soft masking) without the perturbed update; and (3) **GNR** (GU+GC), the full method. Table 7 reports the results on all four benchmarks.

The results reveal that the two components play complementary roles. **GC-only** is the dominant component, providing the majority of both accuracy gains and forgetting reduction compared to the Naive baseline (Table 2). This is expected, as the consolidation mechanism directly protects important parameters via soft masking. **GU-only**, without any knowledge preservation mechanism, performs comparably to the Naive baseline, confirming that the SAM-style perturbation alone does not prevent forgetting. Notably, this is a direct consequence of the architectural coupling between GU and GC: in Algorithm 2 (line 7), the gradient update is scaled by $(1 - \gamma^{\leq t-1})$, where $\gamma^{\leq t-1}$ is the knowledge base produced by GC. When GC is removed, $\gamma^{\leq t-1} = 0$ throughout training, so the scaling factor becomes $(1 - 0) = 1$ and GU reduces to plain SAM-style gradient interpolation without any parameter-level protection. In other words, the GU-only variant only performs gradient perturbation for flat-minima seeking, but lacks the ability to selectively slow down updates on important parameters. However, combining GU with GC (the full GNR) consistently yields further improvements: +1.5–2.9% accuracy on CIFAR-100 benchmarks and an additional 1–2% reduction in forgetting across all benchmarks. This demonstrates that the gradient-norm-aware update helps the optimiser find flatter minima that are more amenable to the consolidation step, validating the complementary design of the two modules.

7. Conclusion

This paper presents a Gradient Norm Regularisation (GNR) approach to mitigate catastrophic forgetting in continual learning. By proposing a task similarity quantification method based on empirical loss, it reveals a strong connection between task similarity and catastrophic forgetting, demonstrating that task dissimilarity exacerbates forgetting. Based on this, we enable dynamic adjustments to task adaptability, effectively preserving knowledge across tasks. These findings offer new insights for improving continual learning systems. Experimentally, GNR outperforms existing regularisation methods on multiple benchmark datasets and achieves significant improvements in knowledge retention and task adaptability. Additionally, the method's application to a real-world pest classification dataset validates its practical applicability in dynamic environments. GNR integrates seamlessly with other continual learning algorithms, providing a versatile solution of continual learning systems for real-world, evolving tasks. Future research will focus on

Table 7Component-level ablation study. Accuracy (%) $\uparrow$ and Forgetting (%) $\downarrow$ on four benchmarks.

Variant	C-10		C-20		T-10		T-20	
	Acc. $\uparrow$	Fgt. $\downarrow$	Acc. $\uparrow$	Fgt. $\downarrow$	Acc. $\uparrow$	Fgt. $\downarrow$	Acc. $\uparrow$	Fgt. $\downarrow$
GU-only	65.1 $\pm$ 3.4	15.7 $\pm$ 2.8	52.6 $\pm$ 4.9	30.9 $\pm$ 5.3	37.3 $\pm$ 1.0	18.7 $\pm$ 0.4	41.0 $\pm$ 1.3	21.0 $\pm$ 0.4
GC-only	71.2 $\pm$ 1.0	6.0 $\pm$ 0.6	72.0 $\pm$ 1.4	7.0 $\pm$ 1.4	50.2 $\pm$ 1.1	4.6 $\pm$ 0.7	59.4 $\pm$ 1.0	4.2 $\pm$ 1.3
GNR (GU+GC)	72.7$\pm$1.9	4.2$\pm$0.4	74.8$\pm$0.7	4.7$\pm$1.5	50.3$\pm$1.4	3.4$\pm$0.9	59.6$\pm$1.1	3.5$\pm$0.4

improving model representations for continual learning to enhance adaptability while preserving knowledge.

Acknowledgment

This work was supported by the China Scholarship Council (No.202107030007) and the Engineering and Physical Sciences Research Council (EPSRC) Doctoral Training Partnership (EP/T517835/1).

References

- [1] Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T., 2018. Memory aware synapses: Learning what (not) to forget, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 139–154.
- [2] Alvarez-Melis, D., Fusi, N., 2020. Geometric dataset distances via optimal transport. Advances in Neural Information Processing Systems 33, 21428–21439.
- [3] Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S., 2020. Dark experience for general continual learning: a strong, simple baseline. Advances in neural information processing systems 33, 15920–15930.
- [4] Chaudhry, A., Marc'Aurelio, R., Rohrbach, M., Elhoseiny, M., 2019. Efficient lifelong learning with a-gem, in: 7th International Conference on Learning Representations, ICLR 2019, International Conference on Learning Representations, ICLR.
- [5] Chen, H.J., Cheng, A.C., Juan, D.C., Wei, W., Sun, M., 2020. Mitigating forgetting in online continual learning via instance-aware parameterization. Advances in Neural Information Processing Systems 33, 17466–17477.
- [6] Chen, L.C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. arXiv preprint:1706.05587.
- [7] Cui, Y., Song, Y., Sun, C., Howard, A., Belongie, S., 2018. Large scale fine-grained categorization and domain-specific transfer learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4109–4118.
- [8] Du, F., Yang, Y., Zhao, Z., Zeng, Z., 2023. Efficient perturbation inference and expandable network for continual learning. Neural Networks 159, 97–106.
- [9] Farajtabar, M., Azizan, N., Mott, A., Li, A., 2020. Orthogonal gradient descent for continual learning, in: International Conference on Artificial Intelligence and Statistics, PMLR. pp. 3762–3773.
- [10] Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B., 2020. Sharpness-aware minimization for efficiently improving generalization, in: International Conference on Learning Representations.
- [11] Graffieti, G., Maltoni, D., Pellegrini, L., Lomonaco, V., 2023. Generative negative replay for continual learning. Neural Networks 162, 369–383.
- [12] Hiratani, N., 2024. Disentangling and mitigating the impact of task similarity for continual learning, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems.
- [13] Hutter, F., Hoos, H., Leyton-Brown, K., 2014. An efficient approach for assessing hyperparameter importance, in: International conference on machine learning, PMLR. pp. 754–762.
- [14] Jin, H., Kim, E., 2022. Helpful or harmful: Inter-task association in continual learning, in: European Conference on Computer Vision, Springer. pp. 519–535.
- [15] Ke, Z., Liu, B., Huang, X., 2020. Continual learning of a mixed sequence of similar and dissimilar tasks. Advances in Neural Information Processing Systems 33, 18493–18504.
- [16] Ke, Z., Liu, B., Xiong, W., Celikyilmaz, A., Li, H., 2023. Sub-network discovery and soft-masking for continual learning of mixed tasks, in: The 2023 Conference on Empirical Methods in Natural Language Processing.
- [17] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al., 2017. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences 114, 3521–3526.
- [18] Konishi, T., Kurokawa, M., Ono, C., Ke, Z., Kim, G., Liu, B., 2023. Parameter-level soft-masking for continual learning, in: International Conference on Machine Learning, PMLR. pp. 17492–17505.
- [19] Kornblith, S., Norouzi, M., Lee, H., Hinton, G., 2019. Similarity of neural network representations revisited, in: International Conference on Machine Learning, PMLR. pp. 3519–3529.
- [20] Krizhevsky, A., Hinton, G., et al., 2009. Learning multiple layers of features from tiny images.
- [21] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25.
- [22] Lee, S., Goldt, S., Saxe, A., 2021. Continual learning in the teacher-student setup: Impact of task similarity, in: International Conference on Machine Learning, PMLR. pp. 6109–6119.
- [23] Lee, S., Mannelli, S.S., Clopath, C., Goldt, S., Saxe, A., 2022. Maslow's hammer in catastrophic forgetting: Node re-use vs. node activation, in: International Conference on Machine Learning, PMLR. pp. 12455–12477.
- [24] Li, X., Zhou, Y., Wu, T., Socher, R., Xiong, C., 2019. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting, in: International Conference on Machine Learning, PMLR. pp. 3925–3934.
- [25] Li, Z., Hoiem, D., 2017. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence 40, 2935–2947.
- [26] Lin, S., Yang, L., Fan, D., Zhang, J., 2022. Trgp: Trust region gradient projection for continual learning, in: The Tenth International Conference on Learning Representations.
- [27] Liu, T., Wang, X., Huang, H., Yang, P., 2023. Weak regression enhanced lifelong learning for improved performance and reduced training data, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 1587–1596.
- [28] Liu, T., Wang, X., Yang, P., Chen, S., Harris, C.J., 2024. Unsupervised transfer aided lifelong regression for learning new tasks without target output. IEEE Transactions on Knowledge and Data Engineering.
- [29] Lomonaco, V., Pellegrini, L., Cossu, A., Carta, A., Graffieti, G., Hayes, T.L., De Lange, M., Masana, M., Pomponi, J., Van de Ven, G.M., et al., 2021. Avalanche: an end-to-end library for continual

- learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3600–3610.
- [30] Lopez-Paz, D., Ranzato, M., 2017. Gradient episodic memory for continual learning, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6470–6479.
- [31] Mallya, A., Davis, D., Lazebnik, S., 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights, in: Proceedings of the European conference on computer vision (ECCV), pp. 67–82.
- [32] Mallya, A., Lazebnik, S., 2018. Packnet: Adding multiple tasks to a single network by iterative pruning, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 7765–7773.
- [33] Malviya, P., Ravindran, B., Chandar, S., 2022. Tag: Task-based accumulated gradients for lifelong learning, in: Conference on Lifelong Learning Agents, PMLR. pp. 366–389.
- [34] Mandler, G., 2014. The structure of value: Accounting for taste, in: Affect and cognition. Psychology Press, pp. 3–36.
- [35] Mirzadeh, S.I., Farajtabar, M., Gorur, D., Pascanu, R., Ghasemzadeh, H., 2021. Linear mode connectivity in multitask and continual learning, in: The Ninth International Conference on Learning Representations (ICLR 2022).
- [36] Mirzadeh, S.I., Farajtabar, M., Pascanu, R., Ghasemzadeh, H., 2020. Understanding the role of training regimes in continual learning. Advances in Neural Information Processing Systems 33, 7308–7320.
- [37] Nguyen, C.V., Achille, A., Lam, M., Hassner, T., Mahadevan, V., Soatto, S., 2019. Toward understanding catastrophic forgetting in continual learning. arXiv preprint:1908.01091 .
- [38] Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S., 2019. Continual lifelong learning with neural networks: A review. Neural networks 113, 54–71.
- [39] Ramasesh, V.V., Dyer, E., Raghu, M., 2020. Anatomy of catastrophic forgetting: Hidden representations and task semantics, in: International Conference on Learning Representations.
- [40] Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H., 2017. icarl: Incremental classifier and representation learning, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 2001–2010.
- [41] Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., Tesauro, G., 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference, in: International Conference on Learning Representations.
- [42] Saha, G., Garg, I., Roy, K., 2020. Gradient projection memory for continual learning, in: International Conference on Learning Representations.
- [43] Shin, H., Lee, J.K., Kim, J., Kim, J., 2017. Continual learning with deep generative replay, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 2994–3003.
- [44] Sodhani, S., Faramarzi, M., Mehta, S.V., Malviya, P., Abdelsalam, M., Janarthanan, J., Chandar, S., 2022. An introduction to lifelong supervised learning. arXiv preprint:2207.04354 .
- [45] van de Ven, G.M., Tuytelaars, T., Tolias, A.S., 2022. Three types of incremental learning. Nature Machine Intelligence , 1–13.
- [46] Wang, X., Zhang, Y., Liu, T., Yuan, Z., Liu, K., Lanfranchi, V., Yang, P., 2025. Spot: An efficient training-free task similarity quantification method for continual learning. Pattern Recognition Letters 197, 168–174.
- [47] Wang, Z., Li, Y., Shen, L., Huang, H., 2024. A unified and general framework for continual learning, in: The Twelfth International Conference on Learning Representations.
- [48] Wickramasinghe, B., Saha, G., Roy, K., 2023. Continual learning: A review of techniques, challenges and future directions. IEEE Transactions on Artificial Intelligence .
- [49] Wu, J., Zhang, Q., Xu, G., 2017. Tiny imagenet challenge. Technical report .
- [50] Wu, X., Zhan, C., Lai, Y., Cheng, M.M., Yang, J., 2019a. Ip102: A large-scale benchmark dataset for insect pest recognition, in: IEEE CVPR, pp. 8787–8796.
- [51] Wu, X., Zhan, C., Lai, Y.K., Cheng, M.M., Yang, J., 2019b. Ip102: A large-scale benchmark dataset for insect pest recognition, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8787–8796.
- [52] Xiao, Z., Du, Z., Wang, R., Gan, R., Li, J., 2023. Online continual learning with declarative memory. Neural Networks 163, 146–155.
- [53] Yoon, J., Yang, E., Lee, J., Hwang, S.J., 2018. Lifelong learning with dynamically expandable networks, in: 6th International Conference on Learning Representations, ICLR 2018, International Conference on Learning Representations, ICLR.
- [54] Zenke, F., Poole, B., Ganguli, S., 2017. Continual learning through synaptic intelligence, in: International conference on machine learning, PMLR. pp. 3987–3995.
- [55] Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D., 2018. mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations.
- [56] Zhao, Y., Zhang, H., Hu, X., 2022. Penalizing gradient norm for efficiently improving generalization in deep learning, in: International Conference on Machine Learning, PMLR. pp. 26982–26992.