Regularizing Second-Order Influences for Continual Learning (Supplementary Material)

Zhicheng Sun[1], Yadong Mu[1,2*], Gang Hua[3]
[1]Peking University, [2]Peng Cheng Laboratory, [3]Wormpex AI Research
{sunzc,myd}@pku.edu.cn, ganghua@gmail.com

For clarity, blue characters will be used for referring to sections and equations in the main paper, while red and green characters refer to tables, equations and citations in this supplementary material.

A. Notation

Table 1 summarizes the used notation for quick lookup.

Symbol	Description
$\mathcal{Z}_t$	Available data at the t-th step
$\mathcal{Z}_{1:t}$	Seen data till the t-th step
$\mathcal{C}_t$	Coreset at the t-th step
m	Maximum coreset size
$L(z, \theta)$	Loss of parameter θ on sample z
$\hat{\theta}_t$	Optimal point at the t-th step
$\hat{\theta}_{\epsilon,z}$	Optimal point after z is upweighted by ϵ
$H_{\hat{\theta}_t}$	Hessian of $\hat{\theta}_t$ on coreset $\mathcal{C}_t$
$H_{\hat{\theta}_t,z}$	Hessian of $\hat{\theta}_t$ on sample z
s_t	Inverse Hessian-vector product at the t-th step
$\mathcal{I}(z)$	Influence of z on the test loss
$\mathcal{I}_{\epsilon,z}(z')$	Influence of z' after z is upweighted by ϵ
$\mathcal{I}^{(2)}(z, z')$	Second-order influence of z and z'
$\Delta I(z')$	Total interference on the influence of z'
$\mathcal{R}(\cdot)$	Our proposed regularizer

Table 1. Notation in the main paper.

B. Continual learning framework

The pseudocode for our learning procedure is presented in Alg. 1. Following ER [7], the model is trained on a mini-batch composed of the current task data and replay examples at each time step. Meanwhile, to reduce the computational cost imposed by the selection algorithm, the replay buffer is updated only in the last epoch of each task. For the settings of hyperparameters, please refer to Sect. 4.1.

Algorithm 1 Learning Procedure for Task T

1: **Input:** Dataset $\mathcal{Z}$ of task T, coreset $\mathcal{C}_{t-1}$ from the last round of selection, the number of epochs e_{max}, model parameter θ, learning rate η.
2: **for** $e = 1$ **to** e_{max} **do**
3:   **for** each batch $\mathcal{Z}_t \in \mathcal{Z}$ **do**
4:     Sample a replay batch $\mathcal{B}_\mathcal{C} \in \mathcal{C}_{t-1}$
5:     $\theta \leftarrow \theta - \eta \nabla_\theta \sum_{z \in \mathcal{Z}_t \cup \mathcal{B}_\mathcal{C}} L(z, \theta)$
6:     **if** $e = e_{\text{max}}$ **then**
7:       Update coreset $\mathcal{C}_t \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t$ by Sect. 3.6
8:       $t \leftarrow t + 1$

C. Derivation of influence functions

As a background introduction, this section provides the derivation of the first-order influence score $\mathcal{I}(z)$ in Eq. (4), following the idea by Koh and Liang [10].

It begins with upweighting an interested sample z by an infinitesimal amount ϵ, after which the perturbed optimal point $\hat{\theta}_{\epsilon,z}$ can be written as follows:

$$\hat{\theta}_{\epsilon,z} = \arg\min_\theta \sum_{z_i \in \mathcal{C}_t} L(z_i, \theta) + \epsilon L(z, \theta). \tag{1}$$

Its first-order optimality condition states that:

$$0 = \sum_{z_i \in \mathcal{C}_t} \nabla_\theta L(z_i, \hat{\theta}_{\epsilon,z}) + \epsilon \nabla_\theta L(z, \hat{\theta}_{\epsilon,z}). \tag{2}$$

To exploit the known optimal point $\hat{\theta}_t$, we apply the first-order Taylor expansion on the right-hand side:

$$\begin{aligned} 0 \approx & \Big[\sum_{z_i \in \mathcal{C}_t} \nabla_\theta L(z_i, \hat{\theta}_t) + \epsilon \nabla_\theta L(z, \hat{\theta}_t)\Big] \\ & + \Big[\sum_{z_i \in \mathcal{C}_t} \nabla^2_\theta L(z_i, \hat{\theta}_t) + \epsilon \nabla^2_\theta L(z, \hat{\theta}_t)\Big](\hat{\theta}_{\epsilon,z} - \hat{\theta}_t), \end{aligned} \tag{3}$$

where $o(\|\hat{\theta}_{\epsilon,z} - \hat{\theta}_t\|)$ terms are dropped. It is also assumed that L is twice-differentiable and convex in θ. Using the optimality condition $\sum_{z_i \in \mathcal{C}_t} \nabla_\theta L(z_i, \hat{\theta}_t) = 0$ and the notation $H_{\hat{\theta}_t} = \sum_{z_i \in \mathcal{C}_t} \nabla^2_\theta L(z_i, \hat{\theta}_t)$, it can be simplified to:

$$\hat{\theta}_{\epsilon,z} - \hat{\theta}_t \approx H^{-1}_{\hat{\theta}_t} \nabla_\theta L(z, \hat{\theta}_t)\epsilon, \tag{4}$$

where $o(\epsilon)$ terms are neglected. This yields the derivate of

*Corresponding author.

$\hat{\theta}_{\epsilon,z}$ w.r.t. ϵ:

$$\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\Big|_{\epsilon=0} = -H_{\hat{\theta}_t}^{-1}\nabla_\theta L(z,\hat{\theta}_t). \tag{5}$$

Finally, the influence of a particular sample z on the test loss can be computed by the chain rule:

$$\begin{aligned}\mathcal{I}(z) &= \sum_{z_i\in\mathcal{C}_{t-1}\cup\mathcal{Z}_t}\frac{dL(z_i,\hat{\theta}_{\epsilon,z})}{d\epsilon}\Big|_{\epsilon=0}\\ &= \sum_{z_i\in\mathcal{C}_{t-1}\cup\mathcal{Z}_t}\nabla_\theta L(z_i,\hat{\theta}_t)^\top\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\Big|_{\epsilon=0}\\ &= -\sum_{z_i\in\mathcal{C}_{t-1}\cup\mathcal{Z}_t}\nabla_\theta L(z_i,\hat{\theta}_t)^\top H_{\hat{\theta}_t}^{-1}\nabla_\theta L(z,\hat{\theta}_t).\end{aligned} \tag{6}$$

D. Derivation of the second-order influence

This section demonstrates the derivation of the second-order effects $\mathcal{I}^{(2)}(z,z')$ in Eq. (8) of Sect. 3.3. Note that the derivation below applies to Eq. (6) as well, since they share a similar form.

In that case, the influence score of a subsequent sample z' after the previous z is upweighted by ϵ is as follows:

$$\begin{aligned}\mathcal{I}_{\epsilon,z}(z') = &-\Big(\sum_{z_i\in\mathcal{C}_t\cup\mathcal{Z}_{t+1}}\nabla_\theta L(z_i,\hat{\theta}_{t+1})+\epsilon\nabla_\theta L(z,\hat{\theta}_{t+1})\Big)^\top\\ &\Big(H_{\hat{\theta}_{t+1}}+\epsilon H_{\hat{\theta}_{t+1},z}\Big)^{-1}\nabla_\theta L(z',\hat{\theta}_{t+1}).\end{aligned} \tag{7}$$

The inverse matrix therein can be effectively approximated with a Neumann series as $\epsilon\to 0$:

$$\begin{aligned}(A+\epsilon B)^{-1} &= A^{-1}(I+\epsilon BA^{-1})^{-1}\\ &= A^{-1}\sum_{k=0}^{\infty}(-\epsilon BA^{-1})^k\\ &= A^{-1}-\epsilon A^{-1}BA^{-1}+o(\epsilon).\end{aligned} \tag{8}$$

Take $A = H_{\hat{\theta}_{t+1}}$ and $B = H_{\hat{\theta}_{t+1},z}$ and substitute into Eq. (7), then we get:

$$\begin{aligned}\mathcal{I}_{\epsilon,z}(z') = &-\Big(\sum_{z_i\in\mathcal{C}_t\cup\mathcal{Z}_{t+1}}\nabla_\theta L(z_i,\hat{\theta}_{t+1})+\epsilon\nabla_\theta L(z,\hat{\theta}_{t+1})\Big)^\top\\ &\Big(H_{\hat{\theta}_{t+1}}^{-1}-\epsilon H_{\hat{\theta}_{t+1}}^{-1}H_{\hat{\theta}_{t+1},z}H_{\hat{\theta}_{t+1}}^{-1}+o(\epsilon)\Big)\nabla_\theta L(z',\hat{\theta}_{t+1}),\end{aligned} \tag{9}$$

which can be further rearranged into:

$$\begin{aligned}\mathcal{I}_{\epsilon,z}(z') = &-\sum_{z_i\in\mathcal{C}_t\cup\mathcal{Z}_{t+1}}\nabla_\theta L(z_i,\hat{\theta}_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1}\nabla_\theta L(z',\hat{\theta}_{t+1})\\ &+\epsilon\sum_{z_i\in\mathcal{C}_t\cup\mathcal{Z}_{t+1}}\nabla_\theta L(z_i,\hat{\theta}_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1}H_{\hat{\theta}_{t+1},z}H_{\hat{\theta}_{t+1}}^{-1}\nabla_\theta L(z',\hat{\theta}_{t+1})\\ &-\epsilon\nabla_\theta L(z,\hat{\theta}_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1}\nabla_\theta L(z',\hat{\theta}_{t+1})\\ &+o(\epsilon).\end{aligned} \tag{10}$$

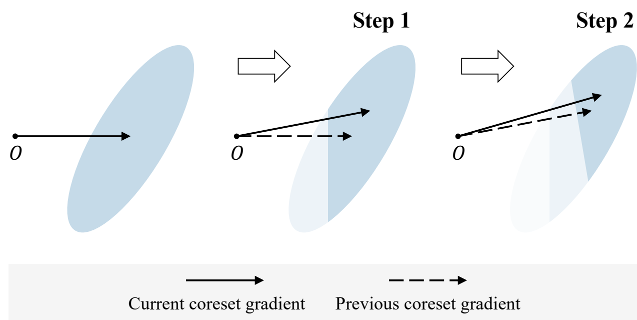

Figure 1. Illustration of two consecutive selection steps based on influence functions. The latter selection turns out to be non-ideal, as evidenced by its decision boundary (between high and low density regions indicated by color intensity) being rotated under the interference of the previous step on gradient information.

With notation $s_{t+1} = H_{\hat{\theta}_{t+1}}^{-1}\sum_{z_i\in\mathcal{C}_t\cup\mathcal{Z}_{t+1}}\nabla_\theta L(z_i,\hat{\theta}_{t+1})$, its derivative w.r.t. ϵ can be written as:

$$\begin{aligned}&\mathcal{I}^{(2)}(z,z') = \frac{d\mathcal{I}_{\epsilon,z}(z')}{d\epsilon}\Big|_{\epsilon=0}\\ &= -(\nabla_\theta L(z,\hat{\theta}_{t+1})-H_{\hat{\theta}_{t+1},z}s_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1}\nabla_\theta L(z',\hat{\theta}_{t+1}).\end{aligned} \tag{11}$$

E. Intuition behind the deviration

To illustrate the physical meaning behind the equations, this section presents Figure 1 as an intuitive example of the second-order effects on sample selection.

It is depicted that after two rounds of selection, the samples are more concentrated in the upper right corner. On a closer look, the prior selection alters the overall gradient, thereby distorting the next selection boundary which is inherently orthogonal to the gradient (by the inner product defined in Sect. 3.3). The final result is thus biased and less diversified.

The illustrated example, which focuses on the drift of decision boundary due to the deviation in coreset gradient, is characterized by our first case of second-order influences in Eq. (6). Complementarily, the disturbance to Hessian-related information is tackled in the second case of Eq. (8).

F. Comparison with group influences

Our second-order influences have a different origin from the group influences proposed by Basu *et al.* [3]. The group effects [3, 9] in their work arise from the interaction within a group of reweighted datapoints on the inner objective, so they are limited to jointly optimized samples. Our second-order terms, derived from separate analyses of inner and outer objectives, in contrast, have no such restrictions and apply to sequentially incoming data.

G. Connection to diversity

This section presents an algebraic view of the connection between our regularizer and gradient diversity, as a complement to the geometric perspective in Sect. 3.5.

Let $\mathcal{R}^o(\mathcal{C}_t)$ and $\mathcal{R}^i(\mathcal{C}_t)$ denote the regularizers under the $\mu = 0$ and identical Hessian settings, respectively. They are expressed as:

$$\mathcal{R}^o(\mathcal{C}_t) = \Big\| \sum_{z \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t} \nabla_\theta L(z, \hat{\theta}_t) - \sum_{z \in \mathcal{C}_t} \nabla_\theta L(z, \hat{\theta}_t) \Big\|,$$
$$\mathcal{R}^i(\mathcal{C}_t) = \Big\| (1 - \alpha\mu) \sum_{z \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t} \nabla_\theta L(z, \hat{\theta}_t) - \sum_{z \in \mathcal{C}_t} \nabla_\theta L(z, \hat{\theta}_t) \Big\|. \tag{12}$$

where α is a coefficient related only to the coreset size. The comparison of the two regularizers yields:

$$\mathcal{R}^i(\mathcal{C}_t)^2 - \mathcal{R}^o(\mathcal{C}_t)^2 = \underbrace{(-2\alpha\mu + \alpha^2\mu^2) \Big\| \sum_{z \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t} \nabla_\theta L(z, \hat{\theta}_t) \Big\|^2}_{\text{constant}}$$
$$+ \underbrace{2\alpha\mu \Big(\sum_{z \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t} \nabla_\theta L(z, \hat{\theta}_t) \Big)^\top \Big(\sum_{z \in \mathcal{C}_t} \nabla_\theta L(z, \hat{\theta}_t) \Big)}_{\text{diversity}}, \tag{13}$$

in which the latter term enforces the coreset gradient to be less aligned with the main gradient. Thus, the regularizer $\mathcal{R}^i(\mathcal{C}_t)$ additionally encourages the inclusion of gradients in other directions and promotes gradient diversity.

H. Taylor expansion of the regularizer

To optimize the new equivalent form of our regularizer in Eq. (14), we perform a first-order Taylor expansion near the initial weight $w^o_{t,i}$:

$$\mathcal{R}(w_t) \approx \mathcal{R}(w^o_t) - \sum_{z_i \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t} \beta^T (\nabla_\theta L(z_i, \hat{\theta}_t) - \mu H_{\hat{\theta}_t, z_i} s_t)(w_{t,i} - w^o_{t,i}), \tag{14}$$

where β is a vector independent of $w_{t,i}$:

$$\beta = \sum_{z_i \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t} \frac{(1 - w^o_{t,i})(\nabla_\theta L(z_i, \hat{\theta}_t) - \mu H_{\hat{\theta}_t, z_i} s_t)}{\mathcal{R}(w^o_t)}. \tag{15}$$

The result is a linear combination of $w_{t,i}$, and thus can be minimized with greedy heuristics, *i.e.*, by iteratively setting the $w_{t,i}$ with the largest coefficient to zero.

I. Additional results

Time cost with Hessian-vector product. The overhead in evaluating the Hessian-vector product is 0.014±0.001 seconds per step on Split CIFAR-10. This is fairly small compared to the base cost of 0.368±0.029 seconds per step for computing first-order influence functions.

Method	Class-incremental	Task-incremental
Grad matching	39.56±1.52 •	88.98±0.95 •
Grad diversity	43.94±2.03 •	87.82±1.38 •
Vanilla IF	47.09±0.85 •	90.78±1.21
Ours	**52.81±1.26**	**92.43±1.11**

Table 2. Comparison with only gradient regularization, in terms of ACC (%) on Split CIFAR-10 with $m = 500$. • indicates significant improvement with p-value less than 0.05 in paired t-tests.

Method	Class-incremental	Task-incremental
iCaRL [12]	47.87±0.47 •	90.35±1.13
BiC [14]	51.49±1.37	90.99±0.78
Ours	**52.81±1.26**	**92.43±1.11**
ER-ACE [5]	56.86±0.64 •	89.59±3.23
ER-ACE + Ours	**60.57±0.93**	**91.84±0.71**

Table 3. Comparison with multi-epoch methods and ER variant in 50-epoch learning. Detailed settings follow Table 2.

Comparison with only gradient regularization. Combination of memory replay with gradient regularization based approaches can partly bypass the interference issue. However, it lacks efficiency in buffering the most critical samples for performance preservation. We verify this point through the comparisons in Table 2, which empirically justifies the motivation of our proposed influence-based scheme.

Comparison with multi-epoch competitors. Additional comparisons with the classical multi-epoch methods iCaRL [12] and BiC [14] are given in Table 3, which confirm the edge of our method in 50-epoch learning. Results are presented with standard deviations.

In combination with ER-ACE. Table 3 further tests our strategy on the more advanced replay framework ER-ACE [5] instead of the previously adopted ER [7]. It is observed that the proposed method combines well with ER-ACE and yields a 3.71% gain in class-incremental learning.

Additional comparison. To compare with other replay-based competitors such as OCS [15], GCR [15] and Bilevel [4], as well as some regularization-based methods such as Stable SGD [11] and EWC [8], we reimplement our approach using the codebase of OCS. Its framework differs in mainly two aspects: (1) Methods are evaluated on two task-incremental benchmarks, including 20-split CIFAR-100 and a mixture of five datasets from different domains. (2) Each learning stage features much fewer training epochs, so that the resulting ACC will be lower

Method	Split CIFAR-100		Multiple Datasets	
	ACC (%)	BWT	ACC (%)	BWT
iCaRL [12]	60.3	-0.04	-	-
EWC [8]	49.5	-0.48	42.7	-0.28
A-GEM [6]	50.7	-0.19	-	-
ER [7]	46.9	-0.21	-	-
GSS [2]	59.7	-0.04	60.2	-0.07
ER-MIR [1]	60.2	-0.04	56.9	-0.11
Stable SGD [11]	57.4	-0.07	53.4	-0.16
Bilevel [4]	60.1	-0.04	58.1	-0.08
OCS [15]	60.5	-0.04	61.5	**-0.03**
GCR [13]	60.9	-	-	-
Vanilla IF	60.0	-0.05	59.7	-0.07
Ours	**61.2**	**-0.04**	**61.6**	-0.05

Table 4. Comparison with another group of baseline methods in task-incremental evaluations. The results of most methods come from the summary in OCS [15], while the result of GCR [13] is provided in its supplementary material.

than before, while the forgetting metric BWT will be significantly better.

As shown in Tab. 4, our approach continues to deliver considerable improvement over the base strategy Vanilla IF in new evaluations. Like many replay-based methods, we outperform regularization-based methods by a large margin. Furthermore, our method surpasses the top two competitors OCS and GCR in terms of ACC on both benchmarks. These results once again demonstrate the superiority of our approach in continual learning.

References

[1] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *NeurIPS*, pages 11849–11860. 2019. 4

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, pages 11817–11826, 2019. 4

[3] Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *ICML*, pages 715–724, 2020. 2

[4] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. In *NeurIPS*, pages 14879–14890, 2020. 3, 4

[5] 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 *ICLR*, 2022. 3

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 4

[7] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. In *ICML Workshops*, 2019. 1, 3, 4

[8] 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. 3, 4

[9] Pang Wei Koh, Kai-Siang Ang, Hubert HK Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *NeurIPS*, pages 5254–5264, 2019. 2

[10] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, pages 1885–1894, 2017. 1

[11] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *NeurIPS*, pages 7308–7320, 2020. 3, 4

[12] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 3, 4

[13] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. GCR: Gradient coreset based replay buffer selection for continual learning. In *CVPR*, pages 99–108, 2022. 4

[14] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 3

[15] Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. In *ICLR*, 2022. 3, 4