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

Zhicheng Sun\textsuperscript{1}, Yadong Mu\textsuperscript{1,2*}, Gang Hua\textsuperscript{3}
\textsuperscript{1}Peking University, \textsuperscript{2}Peng Cheng Laboratory, \textsuperscript{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.

B. Continual learning framework

The pseudocode for our learning procedure is presented in Alg. 1. Following ER [7], the model is trained on a minibatch 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.

C. Derivation of influence functions

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

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

$$\hat{\theta}_{\epsilon,z} = \arg\min_\theta \sum_{z_i \in C_t} L(z_i, \theta) + \epsilon L(z, \theta).$$

(1)

Its first-order optimality condition states that:

$$0 = \sum_{z_i \in C_t} \nabla_\theta L(z_i, \hat{\theta}_{\epsilon,z}) + \epsilon \nabla_\theta L(z, \hat{\theta}_{\epsilon,z}).$$

(2)

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

$$0 \approx \left[ \sum_{z_i \in C_t} \nabla_\theta L(z_i, \hat{\theta}_t) + \epsilon \nabla_\theta L(z, \hat{\theta}_t) \right]$$

$$+ \left[ \sum_{z_i \in C_t} \nabla^2_\theta L(z_i, \hat{\theta}_t) + \epsilon \nabla^2_\theta L(z, \hat{\theta}_t) \right] (\hat{\theta}_{\epsilon,z} - \hat{\theta}_t).$$

(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 $\theta$. Using the optimality condition $\sum_{z_i \in C_t} \nabla_\theta L(z_i, \hat{\theta}_t) = 0$ and the notation $H_{\hat{\theta}_t} = \sum_{z_i \in C_t} \nabla^2_\theta L(z_i, \hat{\theta}_t)$, it can be simplified to:

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

(4)

where $o(\epsilon)$ terms are neglected. This yields the derivat of
the derivation below applies to Eq. (6) as well, since they can be further rearranged into: 

\[ A \]

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

\[ I(z) = \sum_{z_i \in C \cup Z_{t+1}} \frac{dL(z_i, \hat{\theta}_{t+1})}{d\epsilon} \bigg|_{\epsilon=0} \]

\[ = \sum_{z_i \in C \cup Z_{t+1}} \nabla_{\epsilon} L(z_i, \hat{\theta}_t)^\top \frac{d\hat{\theta}_{t+1}}{d\epsilon} \bigg|_{\epsilon=0} \]

\[ = - \sum_{z_i \in C \cup Z_{t+1}} \nabla_{\epsilon} L(z_i, \hat{\theta}_t)^\top H_{\hat{\theta}_t}^{-1} \nabla_{\epsilon} L(z, \hat{\theta}_t). \tag{6} \]

## D. Derivation of the second-order influence

This section demonstrates the derivation of the second-order effects \( I^{(2)}(z, z') \) 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 \( \epsilon \) as is follows:

\[ I_{\epsilon,z}(z') = - \left( \sum_{z_i \in C \cup Z_{t+1}} \nabla_{\epsilon} L(z_i, \hat{\theta}_{t+1}) + \epsilon \nabla_{\epsilon} L(z, \hat{\theta}_t) \right)^\top \]

\[ \left( H_{\hat{\theta}_{t+1}} + \epsilon H_{\hat{\theta}_{t+1},z} \right)^{-1} \nabla_{\epsilon} L(z', \hat{\theta}_{t+1}). \tag{7} \]

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

\[ (A + \epsilon B)^{-1} = A^{-1} (I + \epsilon BA^{-1})^{-1} \]

\[ = A^{-1} \sum_{k=0}^{\infty} (-\epsilon BA^{-1})^k \]

\[ = A^{-1} - \epsilon A^{-1} BA^{-1} + o(\epsilon). \tag{8} \]

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

\[ I_{\epsilon,z}(z') = - \left( \sum_{z_i \in C \cup Z_{t+1}} \nabla_{\epsilon} L(z_i, \hat{\theta}_{t+1}) + \epsilon \nabla_{\epsilon} L(z, \hat{\theta}_t) \right)^\top \]

\[ \left( H_{\hat{\theta}_{t+1}}^{-1} - \epsilon H_{\hat{\theta}_{t+1}}^{-1} H_{\hat{\theta}_{t+1},z} H_{\hat{\theta}_{t+1}}^{-1} + o(\epsilon) \right) \nabla_{\epsilon} L(z', \hat{\theta}_{t+1}), \tag{9} \]

which can be further rearranged into:

\[ I_{\epsilon,z}(z') = - \sum_{z_i \in C \cup Z_{t+1}} \nabla_{\epsilon} L(z_i, \hat{\theta}_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1} \nabla_{\epsilon} L(z', \hat{\theta}_{t+1}) \]

\[ + \epsilon \sum_{z_i \in C \cup Z_{t+1}} \nabla_{\epsilon} L(z_i, \hat{\theta}_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1} H_{\hat{\theta}_{t+1},z} H_{\hat{\theta}_{t+1}}^{-1} \nabla_{\epsilon} L(z', \hat{\theta}_{t+1}) \]

\[ - \epsilon \nabla_{\epsilon} L(z, \hat{\theta}_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1} \nabla_{\epsilon} L(z', \hat{\theta}_{t+1}) \]

\[ + o(\epsilon). \tag{10} \]

### E. Intuition behind the deviation

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

\[
\mathcal{R}^0(C_t) = \left\| \sum_{z \in C_{t-1} \cup Z_t} \nabla_\theta L(z, \hat{\theta}_t) - \sum_{z \in C_t} \nabla_\theta L(z, \hat{\theta}_t) \right\|, \\
\mathcal{R}^i(C_t) = \left\| (1 - \alpha \mu) \sum_{z \in C_{t-1} \cup Z_t} \nabla_\theta L(z, \hat{\theta}_t) - \sum_{z \in C_t} \nabla_\theta L(z, \hat{\theta}_t) \right\|, \\
\]

where \( \alpha \) is a coefficient related only to the coreset size. The comparison of the two regularizers yields:

\[
\mathcal{R}^i(C_t)^2 - \mathcal{R}^0(C_t)^2 = (-2\alpha \mu + \alpha^2 \mu^2) \left\| \sum_{z \in C_{t-1} \cup Z_t} \nabla_\theta L(z, \hat{\theta}_t) \right\|^2 + 2\alpha \mu \left( \sum_{z \in C_{t-1} \cup Z_t} \nabla_\theta L(z, \hat{\theta}_t) \right) \left( \sum_{z \in C_t} \nabla_\theta L(z, \hat{\theta}_t) \right),
\]

in which the latter term enforces the coreset gradient to be less aligned with the main gradient. Thus, the regularizer \( \mathcal{R}^i(C_t) \) 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 \( w_{t,i}^0 \):

\[
\mathcal{R}(w_t) \approx \mathcal{R}(w_t^0) - \sum_{z_i \in C_{t-1} \cup Z_t} \beta^T (\nabla_\theta L(z_i, \hat{\theta}_t) - \mu H_{\hat{\theta}_t, z_i} s_t) (w_{t,i} - w_{t,i}^0),
\]

where \( \beta \) is a vector independent of \( w_{t,i} \):

\[
\beta = \sum_{z_i \in C_{t-1} \cup Z_t} (1 - w_{t,i}^0) (\nabla_\theta L(z_i, \hat{\theta}_t) - \mu H_{\hat{\theta}_t, z_i} s_t) / \mathcal{R}(w_t^0).
\]

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.

### 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Class-incremental</th>
<th>Task-incremental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grad matching</td>
<td>39.56±1.52 ●</td>
<td>88.98±0.95 ●</td>
</tr>
<tr>
<td>Grad diversity</td>
<td>43.94±2.03 ●</td>
<td>87.82±1.38 ●</td>
</tr>
<tr>
<td>Vanilla IF</td>
<td>47.09±0.85 ●</td>
<td>90.78±1.21</td>
</tr>
<tr>
<td>Ours</td>
<td>52.81±1.26 ●</td>
<td>92.43±1.11</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Class-incremental</th>
<th>Task-incremental</th>
</tr>
</thead>
<tbody>
<tr>
<td>iCaRL [12]</td>
<td>47.87±0.47 ●</td>
<td>90.35±1.13</td>
</tr>
<tr>
<td>BiC [14]</td>
<td>51.49±1.37</td>
<td>90.99±0.78</td>
</tr>
<tr>
<td>Ours</td>
<td>52.81±1.26 ●</td>
<td>92.43±1.11</td>
</tr>
<tr>
<td>ER-ACE [5]</td>
<td>56.86±0.64 ●</td>
<td>89.59±3.23</td>
</tr>
<tr>
<td>ER-ACE + Ours</td>
<td>60.57±0.93 ●</td>
<td>91.84±0.71</td>
</tr>
</tbody>
</table>

### 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.
### 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Split CIFAR-100</th>
<th>Multiple Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC (%)</td>
<td>BWT</td>
</tr>
<tr>
<td>iCaRL [12]</td>
<td>60.3</td>
<td>-0.04</td>
</tr>
<tr>
<td>EWC [8]</td>
<td>49.5</td>
<td>-0.48</td>
</tr>
<tr>
<td>A-GEM [6]</td>
<td>50.7</td>
<td>-0.19</td>
</tr>
<tr>
<td>ER [7]</td>
<td>46.9</td>
<td>-0.21</td>
</tr>
<tr>
<td>GSS [2]</td>
<td>59.7</td>
<td>-0.04</td>
</tr>
<tr>
<td>ER-MIR [11]</td>
<td>60.2</td>
<td>-0.04</td>
</tr>
<tr>
<td>Stable SGD [11]</td>
<td>57.4</td>
<td>-0.07</td>
</tr>
<tr>
<td>Bilevel [4]</td>
<td>60.1</td>
<td>-0.04</td>
</tr>
<tr>
<td>OCS [15]</td>
<td>60.5</td>
<td>-0.04</td>
</tr>
<tr>
<td>GCR [13]</td>
<td>60.9</td>
<td>-</td>
</tr>
<tr>
<td>Vanilla IF</td>
<td>60.0</td>
<td>-0.05</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>61.2</strong></td>
<td><strong>-0.04</strong></td>
</tr>
</tbody>
</table>

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


