GKEAL: Gaussian Kernel Embedded Analytic Learning for Few-shot Class Incremental Task

Huiping Zhuang1*, Zhenyu Weng2, Run He1, Zhiping Lin2, Ziqian Zeng1
1Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, China
2School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
*corresponding: hpzhuang@scut.edu.cn

Abstract

Few-shot class incremental learning (FSCIL) aims to address catastrophic forgetting during class incremental learning in a few-shot learning setting. In this paper, we approach the FSCIL by adopting analytic learning, a technique that converts network training into linear problems. This is inspired by the fact that the recursive implementation (batch-by-batch learning) of analytic learning gives identical weights to that produced by training on the entire dataset at once. The recursive implementation and the weight-identical property highly resemble the FSCIL setting (phase-by-phase learning) and its goal of avoiding catastrophic forgetting. By bridging the FSCIL with the analytic learning, we propose a Gaussian kernel embedded analytic learning (GKEAL) for FSCIL. The key components of GKEAL include the kernel analytic module which allows the GKEAL to conduct FSCIL in a recursive manner, and the augmented feature concatenation module that balances the preference between old and new tasks especially effectively under the few-shot setting. Our experiments show that the GKEAL gives state-of-the-art performance on several benchmark datasets.

1. Introduction

Class-incremental learning (CIL) [20] can continuously absorbs new category knowledge in a phase-by-phase manner with data coming separately in each phase, after training a classification network. This is important as data can be scattered at various times and locations in a non-identical independent way. The few-shot class incremental learning (FSCIL) [23] further imposes an inefficiency constraint on the data availability. That is, only a few data samples, i.e., few-shot, for each new class is allowed, leading to a more challenging incremental learning problem.

The major challenge for FSCIL follows from the CIL’s, namely the catastrophic forgetting. The performance on old (base) tasks is tremendously discounted after learning new tasks. This is caused by the lack of training data for old tasks, tricking models to focus only on new tasks. The forgetting issue is also referred to as task-recency bias, in favor of newly learned tasks in prediction. The forgetting issue in FSCIL manifests more quickly due to over-fitting than that in the conventional CIL setting as the training samples become scarce for new tasks.

To handle the forgetting, conventional CIL sparks various contributions, which mainly include the Bias correction-based CIL [1, 9], Regularization-based CIL [11, 13] and Replay-based CIL [17, 20]. They work well in addressing the catastrophic forgetting in CIL. However, the few-shot constraint in FSCIL renders the CIL solutions obsolete (see [23] or our experiments). There have been several works [22,23,31] taking into account the few-shot constraint, outperforming the conventional CIL. These FSCIL techniques take inspirations from existing CIL variants [22] or the few-shot learning angle (e.g., prototype-based [31]) to present catastrophic forgetting.
In this paper, inspired by analytic learning \[33, 34\]—a technique converting network training into linear problems—we approach the FSCIL in an unique angle by incorporating traditional machine learning techniques. The analytic learning allows the training to be implemented in a recursive manner where training data are scattered into multiple batches. Yet the weights trained recursively are identical to those trained by pouring the entire data in one go \[33\]. We may call this weight-invariant (or weight-identical) property. Such recursive form and its weight-invariant property highly resemble the incremental learning paradigm and its objective of avoiding (catastrophic) forgetting respectively (see Figure 1). Following this intuition, we propose a Gaussian kernel embedded analytic learning (GKEAL) for FSCIL. The GKEAL adopts traditional machine learning tools such as least squares (LS) and matrix inverse to avoid forgetting. The key contributions are summarized as follows.

- We introduce GKEAL by treating the FSCIL as a recursive learning problem to avoid forgetting. We prove that the GKEAL in the FSCIL setting follows the same weight-invariant property as that in analytic learning.
- To bridge analytic learning into the FSCIL realm, the GKEAL replaces the classifier at a network’s final layer with a kernel analytic module (KAM). The KAM contains a Gaussian kernel embedding process for extracting more discriminative feature, and an LS solution that allows the GKEAL to learn new tasks in a recursive manner.
- To mitigate the data imbalance between the base and new tasks, an augmented feature concatenation (AFC) module is introduced, which effectively balances the network’s base-new task preference.
- Experiments on benchmark datasets show that the GKEAL outperforms the state-of-the-art methods by a considerable margin. Ablation study is also provided, giving thorough analysis of the hyperparameters introduced, as well as strong supports to our theoretical claims.

2. Related Works

Class-Incremental Learning. Existing CIL methods mainly include three categories, namely Bias correction-based CIL, Regularization-based CIL and Replay-based CIL. The bias correction-based CIL aims to address the task-recency bias. The end-to-end incremental learning (EEIL) \[1\] introduces a task balance stage in order to reduce the bias. In \[27\], the bias is corrected by introducing an additional trainable layer. The method named LUCIR \[9\] replaces the softmax layer with a cosine normalization alternative to reduce target-recency bias.

The regularization-based CIL imposes certain constraints on the objective functions in order to prevent forgetting. In \[11\], the elastic weight consolidation (EWC) estimates the importance with a Fisher information matrix and constrains those weights. The EWC is later enhanced by \[15\] which seeks a better Fisher matrix approximation. In \[13\], the learning without forgetting method refrains the activations of old tasks from changing too much while absorbing new tasks.

The relay-based CIL has recently become the favored CIL branch due to its competitive performance to resist the catastrophic forgetting by allowing a small mount of historical data. The incremental classifier and representation Learning (iCaRL) \[20\] introduced such a setting. Following the iCaRL, the PODNet proposed in \[4\] employs a spatial-based distillation component, achieving relatively outstanding results especially for large-phase CIL. The AANets \[17\] balances the stability and plasticity with a stable block and a plastic block. The reinforcement memory management \[18\] handles the forgetting issue adopting reinforcement learning, and plugging it into PODNet and AANets leads to better performance.

Few-shot Learning. The few-shot learning (FSL) addresses the scenario where each category/task is given only a few training samples. Existing FSL methods are mainly optimization-based \[5, 10, 16\] and metric-based \[6, 28, 29\]. Optimization techniques allow a fast adaptation to new few-shot tasks by learning an optimization algorithm. Metric-based methods alleviate distance metrics, e.g., DeepEMD \[30\], in order to measure the deviation between samples.

Few-shot Class-Incremental Learning. The FSCIL \[2, 23, 32\] jointly incorporates the settings of CIL and FSL by performing CIL tasks with each phase containing limited samples for new tasks. The topology-preserving knowledge Incrementer (TOPIC) framework \[23\] mitigates the forgetting issue by stabilizing a neural gas network’s topology. In \[31\], the continually evolved classifier (CEC) separates each class with an independent classifier, and adopts a graph model to propagate context information between classifiers. The F2M \[22\] overcomes the catastrophic forgetting via finding flat minima. This is achieved by injecting noise during base training, suggesting to take the focus of FSCIL back in the base training stage.

Analytic Learning. The analytic learning is developed to avoid the limitations imposed by back-propagation (BP) so that the training can be completed within one epoch. It is also known as pseudoinverse learning \[7\] due to adopting matrix inverse. The analytic learning begins in the shallow learning. For instance, the radial basis network \[19\] trains the parameters using an LS estimation in the final layer after conducting a kernel transformation in the first layer. Analytic learning with multiple layers \[24, 26, 34\] usually converts the nonlinear layers into linear segments, so that LS solutions can be employed in a one-epoch training style. To address the memory issue in analytic learning, the block-wise recursive Moore-Penrose learning (BRMP) \[33\] is developed, allowing analytic learning to stream new samples.
without forgetting the impact of previous learned knowledge. This coincides well with the incremental scenarios and it is introduced to CIL realm in the analytic class-incremental learning (ACIL) [36] which has state-of-the-art performance via a frozen backbone and linear classifier trained by analytic learning. These become the key inspirations for the proposed GKEAL.

3. The Proposed Method

In this section, algorithmic details of the proposed GKEAL are provided. Firstly, the GKEAL adopts a classifier re-training phase on the base task, in which the last-layer classifier is replaced by the KAM. Subsequently the FSCIL tasks are conducted in a recursive manner. The FSCIL discussed in this paper is restricted to classification problems. An overview of GKEAL is depicted in Figure 2.

3.1. Base Training

Base Training via BP. Prior to FSCIL, the network is first trained with BP on the base task (see Figure 2(a)). Here we discuss a commonly seen CNN structure consisting of a stack of CNN layers (or known as CNN backbone) as feature extractor followed by a fully-connected network (FCN) as classifier. Let \( W_{\text{CNN}} \) and \( W_{\text{FCN}} \) represent the weights for the CNN backbone and the FCN classifier. Given an input \( X \in \mathbb{R}^{w\times h\times 3} \) (e.g., color image as an example), the output of the network is

\[
Y = f_{\text{softmax}}(f_{\text{flat}}(f_{\text{CNN}}(X, W_{\text{CNN}}))W_{\text{FCN}})
\]

where \( f_{\text{CNN}}(X, W_{\text{CNN}}) \) indicates the output of the CNN backbone after feeding the input \( X \); \( f_{\text{flat}} \) is a flattening operator, which flattens an \( m \times \text{D} \) tensor into a 1-D vector; \( f_{\text{softmax}} \) is the softmax function.

Analytic Initialization. With the network trained on base dataset, the GKEAL seeks to detach the CNN backbone, shown as follows.

\[
X_{0}^{(\text{cnn})} = f_{\text{flat}}(f_{\text{CNN}}(X_{0}^{\text{train}}, W_{\text{CNN}}))
\]

where \( X_{0}^{(\text{cnn})} \in \mathbb{R}^{N_{0}\times d_{\text{cnn}}} \) is the embedding of \( X_{0}^{\text{train}} \). Subsequently we conduct a Gaussian kernel embedding (GKE) process to obtain kernelized embedding \( X_{0}^{(ke)} \), i.e.,

\[
X_{0}^{(ke)} = g(e_{1},...,e_{r})(X_{0}^{(\text{cnn})})
\]

where \( g \) indicates the GKE module. The \( j \)th row of \( X_{0}^{(ke)} \) writes

\[
X_{0}^{(ke)}[j,:] = [e^{-\beta\|X_{0}^{(\text{cnn})}[j,:]-c_{1}\|^2} \ldots e^{-\beta\|X_{0}^{(\text{cnn})}[j,:]-c_{r}\|^2}]
\]

where \( \{c_{1},...,c_{r}\} \) is a set of center vectors randomly selected from the rows of \( X_{0}^{(\text{cnn})} \), and \( \beta \) is a width-adjusting parameter. Next, the kernelized embedding \( X_{0}^{(ke)} \) is mapped onto the label matrix \( Y_{0}^{\text{train}} \) using a linear regression procedure via solving

\[
\arg\min_{W_{\text{FCN}}} \left\| Y_{0}^{\text{train}} - X_{0}^{(ke)}W_{\text{FCN}}^{(0)} \right\|_{F}^{2} + \gamma \left\| W_{\text{FCN}}^{(0)} \right\|_{F}^{2}
\]

where \( \| \cdot \|_{F} \) indicates the Frobenius norm, and \( \gamma \) regularizes the objective function, with an optimal solution

\[
W_{\text{FCN}}^{(0)} = (X_{0}^{(ke)T}X_{0}^{(ke)} + \gamma I)^{-1}X_{0}^{(ke)TY_{0}^{\text{train}}}
\]

where \( W_{\text{FCN}}^{(0)} \) indicates the estimated weight of the FCN layer, and \( \cdot^{T} \) is the matrix transpose operator.

3.2. Few-shot Class Incremental Learning

Upon completing the AInit process, the FSCIL begins. Let

\[
Y_{0:k-1}^{\text{train}} = \begin{bmatrix}
Y_{0}^{\text{train}} & 0 & 0 & \ldots & 0 \\
0 & Y_{1}^{\text{train}} & 0 & \ldots & 0 \\
\vdots & & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & Y_{k-1}^{\text{train}}
\end{bmatrix}, \quad X_{0:k-1}^{\text{train}} = \begin{bmatrix}
X_{0}^{(ke)} \\
X_{1}^{(ke)} \\
\vdots \\
X_{k-1}^{(ke)}
\end{bmatrix}
\]

where the structure of \( Y_{0:k-1}^{\text{train}} \) is sparse due to disjoint tasks, and \( X_{j}^{(ke)} \) is the \( j \)th embedding via

\[
X_{j}^{(ke)} = g(e_{1},...,e_{r})(f_{\text{CNN}}(X_{j}^{\text{train}}, W_{\text{CNN}})).
\]

Without loss of generality, the learning problem in (4) given \( D_{0:k-1}^{\text{train}} \) can be expanded to

\[
\arg\min_{W_{\text{FCN}}^{(k-1)}} \left\| Y_{0:k-1}^{\text{train}} - X_{0:k-1}^{(ke)}W_{\text{FCN}}^{(k-1)} \right\|_{F}^{2} + \gamma \left\| W_{\text{FCN}}^{(k-1)} \right\|_{F}^{2}
\]
Theorem 3.1. Given \( \mathcal{D}_k^{\text{train}} \), \( R_{k-1} \) and \( \hat{W}_{\text{FCN}}^{(k-1)} \) in (7), \( \hat{W}_{\text{FCN}}^{(k)} \) can be obtained recursively via

\[
\hat{W}_{\text{FCN}}^{(k)} = \begin{bmatrix}
\hat{W}_{\text{FCN}}^{(k-1)} - R_k X_{k}^{(\text{fe})\text{T}} Y_{\text{train}}^{k-1} \\
R_k X_{k}^{(\text{fe})\text{T}} Y_{\text{train}}^{k-1}
\end{bmatrix},
\]

(9)

where

\[
R_k = R_{k-1} - R_{k-1} X_{k}^{(\text{fe})\text{T}} (I + X_k^{(\text{fe})\text{T}} R_{k-1} X_k^{(\text{fe})\text{T}})^{-1} X_k^{(\text{fe})\text{T}} R_{k-1}. \tag{10}
\]

**Proof.** See Supplementary material. \( \square \)

As shown in (9), the solution expresses FSCIL in two folds: 1) The right part of weight is built for new tasks by taking only new information (i.e., \( R_k \), \( X_k^{(\text{fe})\text{T}} \) and \( Y_{\text{train}}^{k} \)). 2) The left part absorbs both new and old tasks (i.e., \( \hat{W}_{\text{FCN}}^{(k-1)} \)) knowledge. This pattern makes sense as the incremental learning should preserve the learned knowledge while accepting the new information’s influence.

Why KAM and LS solution. The KAM and its LS solution are essential in terms of handling data-inefficiency scenarios where the FSCIL naturally belongs. The LS solution has anti-over-fitting nature \([34]\), which is useful given very limited data during each incremental phase. The GKE process also tends to give good performance with small data availability \([35]\).

**Freezing the Backbone.** As shown in (6), the kernelized embedding is obtained by feeding the input through the CNN backbone trained on the base dataset. That is, the backbone’s parameters are not trainable during the FSCIL. Such a freezing action prevents the network from updating itself with new class features. This seems to result in certain performance decline, which is true in the traditional CIL case. However, in the few-shot scenario, data from new classes are scarce compared with that of the base training. These scarce samples are less likely to make vital contributions in improving the backbone’s feature extraction power during the FSCIL. Such a decision (i.e., freezing the back-
bone) also happens in many other FSCIL methods (e.g., CEC [31]). It is a reasonable call to free the backbone in exchange for GKEAL’s no-forgetting property for FSCIL.

3.3. Augmented Feature Concatenation For New Tasks

Theorem 3.1 has given an overview of the proposed GKEAL. However, unlike conventional incremental learning, FSCIL has the issue of sample imbalance. That is, the learning favors the base classes as the base samples are visited more frequently. Here we mitigate the imbalance via an augmented feature concatenation (AFC) process by amplifying the impact of new tasks.

In (9), instead of directly adopting $X_k^{(fe)}$, the AFC process augments and concatenates the feature by

$$X_k^{(a)} = [g(x_1, \ldots, x_3)(\text{FCN}(A_k(X_k^{(w)}), W_{\text{CNN}})), g(x_1, \ldots, x_3)(\text{FCN}(A_k(X_k^{(w)}), W_{\text{CNN}})), \ldots, g(x_1, \ldots, x_3)(\text{FCN}(A_k(X_k^{(w)}), W_{\text{CNN}}))],$$

where we use $\leftarrow$ to re-define $X_k^{(fe)}$ and $Y_k^{\text{train}}$ for convenience. Here $A_k(X_k^{\text{train}})$ indicates the $c$th data augmentation on $X_k^{\text{train}}$. We use commonly seen augmentation techniques such as random horizontal flip, random cropping and normalizing. $C$ here is referred to as the augmentation count. Note that the output $Y_k^{\text{train}}$ is a concatenated matrix stacked $C$ times with the original label. This resembles the multi-epoch training where data for each epoch are augmented randomly, but is not the same since the AFC process updates the weights “in one go” after concatenating the augmented features. Here $C \in \mathbb{Z}$ is an additional hyperparameter balancing the knowledge between old and new tasks. This rewrites (9) as

$$W_{\text{FCN}}^{(k)} = \begin{bmatrix} W_{\text{FCN}}^{(k-1)} & R_k X_k^{(fe)T} & Y_k^{\text{train}} \end{bmatrix}.$$  

By augmenting the few-shot data of new tasks, the solution can avoid being too focused on base classes. For demonstration purpose, we have $X_k^{(fe)T} \approx CX_k^{(fe)T}$ and $X_k^{(fe)T} \approx CX_k^{(fe)T}$. This rewrites (12) as

$$W_{\text{FCN}}^{(k)} \approx \begin{bmatrix} W_{\text{FCN}}^{(k-1)} - CR_k X_k^{(fe)T} Y_k^{\text{train}} \begin{bmatrix} C & 0 \end{bmatrix} \text{gain reduced} + \gamma I \end{bmatrix}.$$

The update formula with AFC process points out that the output channels for new tasks (i.e., right-side weight in (13)) are amplified up to $C$ times while the gains for old tasks are cut down (i.e., left-side weight in (13)). This analysis can later be supported in experiments (see Figure 4(d)).

The AFC works similarly to a regular data augmentation technique, but differs in that it augments the features and labels in matrix form and participates the calculation in a single shot. This allows certain interpretability such as that in (13). Regular data augmentation joins the calculation in a mini-batch form in a sequential manner, whose interpretability is buried during the iterations. We summarize the proposed GKEAL in Algorithm 1.

Algorithm 1 GKEAL

**Require:** Data $\mathcal{D}_{\text{train}}$, number of kernel vectors $I$, regularization parameter $\gamma$, width parameter $\beta$, augmentation count $C$.

1. **BP-based base training:** Train networks with BP on the base dataset.
2. **Analytic initialization (with $\mathcal{D}_{\text{train}}$):** i) Obtain kernelized embedding with (2) and (1); ii) Obtain base weight $W_{\text{FCN}}^{(0)}$ with (5). iii) Obtain and store $R_0 = (X_0^{(fe)} X_0^{(fe)T} + \gamma I)^{-1}$.
3. for $k = 1$ to $K$ (with $\mathcal{D}_{\text{train}}^{(k-1)}$ and $R_{k-1}$) do
   4. i) Obtain augmented kernelized embedding with (11);
   5. ii) Update $R_k$ with (10);
   6. iii) Update weight $W_{\text{FCN}}^{(k)}$ with (12);
end for

4. Experiments

In this section, we compare the proposed GKEAL with several state-of-the-art methods, including CIL-converted methods (i.e., iCaRL [20], EEIL [1], LUCIR [9]) and techniques specifically designed for FSCIL (i.e., TOPIC [23], CEC [31], F2M [22], MetaFSCIL [3] and Entropy-reg [14]). In addition, ablation study and parameter analysis are also included to reveal the contributions of GKEAL’s components.

4.1. Experimental Setup

**Dataset and Data Split.** We evaluate the performance of GKEAL by training ResNet [8] on CIFAR-100 [12], CUB200-2011 [25] and mini-ImageNet [21], which have 100, 200 and 100 image classes respectively. These benchmark Data have image sizes of $32 \times 32$, $224 \times 224$ and $84 \times 84$ respectively. All compared methods follow the
I containing few-shot samples. For CIFAR-100 and mini-ImageNet, the base training set includes 60 classes. For CUB200-2011, the number of classes in the base training set is 100. The FSCIL is conducted in a 5-way 5-shot (5 classes with 5 samples in each class for each phase) manner. All methods experience decreasing accuracy evolution of CIFAR-100 (Figure 3(a)), CUB200-2011 (total of 10 phases excluding the base training). A further reason is FSCIL’s accuracies. The reason behind this degradation is twofold. Firstly, the network takes in new data classes in each phase, and the performance drop rate (PD), i.e., PD = 1 − kthphase / testk, is included for evaluation. Reporting the performance drop is meaningful as some methods may give good results mainly due to a well-trained network on the base dataset. For results in ablation study and parameter analysis, we report the last-phase accuracy Ak for convenience.

4.2. Comparison with State-of-the-arts

As an overview, we depict the compared methods’ accuracy evolution of CIFAR-100 (Figure 3(a)), CUB200-2011 (Figure 3(b)) and mini-ImageNet (Figure 3(c)) with respect to (w.r.t.) each phase. All methods experience decreasing accuracies. The reason behind this degradation is twofold. Firstly, the network takes in new data classes in each phase, giving it more choices to distinguish from, naturally leading to accuracy decrease. The second reason is FSCIL’s implementation details. For the training on the base dataset, we allow various training strategies for compared methods to achieve their desired performances. For instance, on CIFAR-100 F2M [22] requires 240 epochs for base training while the CEC [31], MetaFSCIL [3] and Entropy-reg [14] needs 100. For our GKEAL, for CIFAR-100 or mini-ImageNet, the base training set includes 60 classes. For mini-ImageNet, the base training set includes 60 classes. For CUB200-2011, the number of classes in the base training set is 100. The FSCIL is conducted in a 5-way 5-shot (5 classes with 5 samples in each class for each phase) manner for CIFAR-100 or mini-ImageNet (total of 8 phases excluding the base training) and a 10-way 5-shot manner for CUB200-2011 (total of 10 phases excluding the base training).

Evaluation Protocol. Following [23], the performance for the kth phase is evaluated by the average accuracy, i.e., the test accuracy on the seen classes (i.e., DAk), denoted by Ak. Also, the performance drop rate (PD), i.e., PD = 1 − AK, is included for evaluation. Reporting the performance drop is meaningful as some methods may give good results mainly due to a well-trained network on the base dataset. For results in ablation study and parameter analysis, we report the last-phase accuracy Ak for convenience.

**Table 1. The accuracy among the compared methods on mini-ImageNet.**

<table>
<thead>
<tr>
<th>Phase</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>PD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>iCaRL [20]</td>
<td>61.31</td>
<td>64.32</td>
<td>42.94</td>
<td>37.63</td>
<td>30.49</td>
<td>24.00</td>
<td>20.89</td>
<td>18.80</td>
<td>17.21</td>
<td>44.10</td>
</tr>
<tr>
<td>EEIL [1]</td>
<td>61.31</td>
<td>64.58</td>
<td>44.00</td>
<td>37.29</td>
<td>33.14</td>
<td>27.12</td>
<td>24.10</td>
<td>21.57</td>
<td>19.58</td>
<td>41.73</td>
</tr>
<tr>
<td>LUCIR [9]</td>
<td>61.31</td>
<td>47.80</td>
<td>39.31</td>
<td>31.91</td>
<td>25.68</td>
<td>21.35</td>
<td>18.67</td>
<td>17.24</td>
<td>14.17</td>
<td>47.14</td>
</tr>
<tr>
<td>TOPIC [23]</td>
<td>61.31</td>
<td>50.09</td>
<td>45.17</td>
<td>41.16</td>
<td>37.48</td>
<td>35.52</td>
<td>32.19</td>
<td>29.46</td>
<td>24.42</td>
<td>36.89</td>
</tr>
<tr>
<td>CEC [31]</td>
<td>72.00</td>
<td>66.83</td>
<td>62.97</td>
<td>59.43</td>
<td>56.70</td>
<td>53.73</td>
<td>51.19</td>
<td>49.24</td>
<td>47.63</td>
<td>24.37</td>
</tr>
<tr>
<td>F2M [22]</td>
<td>72.05</td>
<td>67.47</td>
<td>63.16</td>
<td>59.70</td>
<td>56.71</td>
<td>53.77</td>
<td>51.11</td>
<td>49.21</td>
<td>47.84</td>
<td>24.21</td>
</tr>
<tr>
<td>MetaFSCIL [3]</td>
<td>72.04</td>
<td>67.94</td>
<td>63.77</td>
<td>60.29</td>
<td>57.58</td>
<td>55.16</td>
<td>52.99</td>
<td>50.79</td>
<td>49.19</td>
<td>22.85</td>
</tr>
<tr>
<td>Entropy-reg [14]</td>
<td>71.84</td>
<td>67.12</td>
<td>63.21</td>
<td>59.77</td>
<td>57.01</td>
<td>53.95</td>
<td>51.55</td>
<td>49.52</td>
<td>48.21</td>
<td>23.63</td>
</tr>
</tbody>
</table>

**Table 2. The accuracy among the compared methods on CIFAR-100.**

<table>
<thead>
<tr>
<th>Phase</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>PD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>iCaRL [20]</td>
<td>64.10</td>
<td>53.28</td>
<td>41.69</td>
<td>34.13</td>
<td>27.93</td>
<td>25.06</td>
<td>20.41</td>
<td>15.48</td>
<td>13.73</td>
<td>50.37</td>
</tr>
<tr>
<td>EEIL [1]</td>
<td>64.10</td>
<td>53.11</td>
<td>43.71</td>
<td>35.15</td>
<td>28.96</td>
<td>24.98</td>
<td>21.01</td>
<td>17.26</td>
<td>15.85</td>
<td>48.25</td>
</tr>
<tr>
<td>LUCIR [9]</td>
<td>64.10</td>
<td>53.05</td>
<td>43.96</td>
<td>36.97</td>
<td>31.61</td>
<td>26.73</td>
<td>21.23</td>
<td>16.78</td>
<td>13.54</td>
<td>50.56</td>
</tr>
<tr>
<td>TOPIC [23]</td>
<td>64.10</td>
<td>55.88</td>
<td>47.07</td>
<td>45.16</td>
<td>40.11</td>
<td>36.38</td>
<td>33.96</td>
<td>31.55</td>
<td>29.37</td>
<td>34.73</td>
</tr>
<tr>
<td>CEC [31]</td>
<td>73.07</td>
<td>68.88</td>
<td>65.26</td>
<td>61.19</td>
<td>58.09</td>
<td>55.57</td>
<td>53.22</td>
<td>51.34</td>
<td>49.14</td>
<td>23.93</td>
</tr>
<tr>
<td>F2M [22]</td>
<td>71.45</td>
<td>68.10</td>
<td>64.43</td>
<td>60.80</td>
<td>57.76</td>
<td>55.26</td>
<td>53.53</td>
<td>51.17</td>
<td>49.35</td>
<td>22.06</td>
</tr>
<tr>
<td>MetaFSCIL [3]</td>
<td>74.50</td>
<td>70.10</td>
<td>68.54</td>
<td>62.77</td>
<td>59.48</td>
<td>56.52</td>
<td>54.36</td>
<td>52.56</td>
<td>49.97</td>
<td>24.53</td>
</tr>
<tr>
<td>Entropy-reg [14]</td>
<td>74.40</td>
<td>70.20</td>
<td>66.54</td>
<td>62.51</td>
<td>59.71</td>
<td>56.58</td>
<td>54.52</td>
<td>52.39</td>
<td>50.14</td>
<td>24.26</td>
</tr>
</tbody>
</table>

**GKEAL (I, β, C = 10k, 10, 200) | 73.59 | 68.90 | 65.33 | 62.29 | 59.39 | 56.70 | 54.20 | 52.59 | 51.31 | 22.28**

| GKEAL (I, β, C = 5k, 10, 200) | 74.01 | 70.45 | 67.01 | 63.08 | 60.01 | 57.30 | 55.50 | 53.39 | 51.40 | 22.61 |
Table 3. The accuracy among the compared methods on CUB200-2011.

<table>
<thead>
<tr>
<th>Method</th>
<th>Phase: 0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>PDΔ</th>
</tr>
</thead>
<tbody>
<tr>
<td>iCaRL [20]</td>
<td>68.68</td>
<td>52.65</td>
<td>48.61</td>
<td>44.16</td>
<td>36.62</td>
<td>29.52</td>
<td>27.83</td>
<td>26.26</td>
<td>24.01</td>
<td>23.89</td>
<td>21.16</td>
<td>47.52</td>
</tr>
<tr>
<td>EEIL [1]</td>
<td>68.68</td>
<td>53.63</td>
<td>47.91</td>
<td>44.20</td>
<td>36.30</td>
<td>27.46</td>
<td>25.93</td>
<td>24.70</td>
<td>23.95</td>
<td>24.13</td>
<td>22.11</td>
<td>46.57</td>
</tr>
<tr>
<td>TOPIC [23]</td>
<td>68.68</td>
<td>62.49</td>
<td>54.81</td>
<td>49.99</td>
<td>45.25</td>
<td>41.40</td>
<td>38.35</td>
<td>35.36</td>
<td>32.22</td>
<td>28.31</td>
<td>26.26</td>
<td>42.40</td>
</tr>
<tr>
<td>CEC [31]</td>
<td>75.85</td>
<td>71.94</td>
<td>68.50</td>
<td>63.50</td>
<td>62.43</td>
<td>58.27</td>
<td>57.73</td>
<td>55.81</td>
<td>54.83</td>
<td>53.52</td>
<td>52.28</td>
<td>23.57</td>
</tr>
<tr>
<td>F2M [22]</td>
<td>77.13</td>
<td>73.92</td>
<td>70.27</td>
<td>66.37</td>
<td>64.34</td>
<td>61.69</td>
<td>60.52</td>
<td>59.38</td>
<td>57.15</td>
<td>56.94</td>
<td>55.89</td>
<td>21.24</td>
</tr>
<tr>
<td>MetaFSCIL [3]</td>
<td>75.90</td>
<td>72.41</td>
<td>68.78</td>
<td>64.78</td>
<td>62.96</td>
<td>59.99</td>
<td>58.30</td>
<td>56.85</td>
<td>54.78</td>
<td>53.82</td>
<td>52.64</td>
<td>23.26</td>
</tr>
<tr>
<td>Entropy-reg [14]</td>
<td>75.90</td>
<td>72.14</td>
<td>68.64</td>
<td>63.76</td>
<td>62.58</td>
<td>59.11</td>
<td>57.82</td>
<td>55.89</td>
<td>54.92</td>
<td>53.58</td>
<td>52.39</td>
<td>23.51</td>
</tr>
<tr>
<td>GKEAL (I, β, C = 10k, 15, 10)</td>
<td>78.88</td>
<td>75.62</td>
<td>72.32</td>
<td>68.62</td>
<td>67.23</td>
<td>64.26</td>
<td>62.98</td>
<td>61.89</td>
<td>60.20</td>
<td>59.21</td>
<td>58.67</td>
<td>20.21</td>
</tr>
</tbody>
</table>

main concern, i.e., catastrophic forgetting, causing accuracy drop. As observed in the Figure 3, the CEC and F2M are giving similarly competitive results, outperforming iCaRL, EEIL, LUCIR and TOPIC by large margins. This is a reasonable observation as the iCaRL, EEIL and LUCIR are not specifically designed for FSCIL tasks, so they suffer rather significantly from over-fitting. For instance, the iCaRL, a rather strong baseline in traditional CIL, achieves the average accuracy below 20%. The TOPIC is the first FSCIL baseline, giving a slightly better performance over that of the CIL methods, but cannot compete with the recent FSCIL techniques (e.g., CEC and F2M). For the most recent FSCIL techniques like MetaFSCIL and Entropy-reg, they slightly outperform CEC and F2M.

Our GKEAL even outperforms the results of MetaFSCIL and Entropy-reg by a considerable amount (see gaps between the red curves and the second best curves in Figure 3), showing improved accuracy in each phase on all three datasets. The hyperparameters selected for CIFAR-100, CUB200-2011 and mini-ImageNet are \{I, β, C = 5k, 10, 200\}, \{I, β, C = 10k, 15, 10\} and \{I, β, C = 10k, 10, 200\} respectively. For convenience, the detailed results on mini-ImageNet are tabulated in Table 1 as a further support. The GKEAL achieves a 51.31% accuracy at the last phase, overtaking the second best result (MetaFSCIL’s 49.19%) by 2.12%. In particular, the PD score of GKEAL is 22.28%, which is also the lowest, indicating a less forgetting among the compared methods. The detailed results for CIFAR-100 and CUB200-2011 are shown in Table 2 and Table 3, where the results show similar patterns.

4.3. Ablation Study and Parameter Analysis

Ablation Study. Here we conduct an ablation study to justify the contributions of the GKE module and AFC module. Both modules are important to GKEAL. In particular, the GKE gives a critical contribution. As shown in Table 4 (first 2 rows), the analytic learning does not work with the original extracted features by the BP algorithm. That is, lacking the GKE module results in catastrophic accuracy loss (e.g., a drop of last-phase accuracy from 51.21% to 7.22%). This is surprising as a mere linear layer tuned by BP can have reasonable achievement. This is because the matrix inverse experiences ill-conditioned scenario, leading to a breakdown of the solution. On the other hand, the AFC also contributes significantly. It improves the last-phase accuracy by 6.42% from 43.79% to 51.21%. However, the AFC alone cannot help improve the performance. This is because without the GKE, the analytic learning itself might go wrong (performance drop from 74.80% to 13.20%), AFC’s new-old task balancing advantage is not demonstrated as a plug-in to our GKEAL. When GKE is applied, the AFC works effectively. As shown in Figure 4(d), the knowledge of base (old) classes and new classes are balanced by tuning the hyper-parameter C. Larger C emphasizes/weakens the base/new class knowledge, which well supports our theoretical claim in Eq. 13.

Hyperparameter Analysis. Upon proving the importance of the GKE and AFC modules, we further evaluate the impacts of the introduced kernel vector size \(I\), width parameter \(β\) and augmentation count \(C\). As shown in Figure 4(a), in general the GKEAL hungers a quite large \(I\). The best performing parameters for CIFAR-100, CUB200-2011 and mini-ImageNet are at \(I = 8k\), 10k, 12k. This is because the LS solution itself is prone to under-fitting, which can be compensated by an increased dimensionality. This requires an increased number of parameters, which is a common limitation of analytic learning. Investigating a more condense structure is in our future plans.

For parameter \(β\) in the GKE module, as shown Figure 4(b), there is a comfortable range for \(β\) at around \(β ∈ [5, 15]\) for CIFAR-100/mini-ImageNet that gives good results, exceeding either bound would cause performance decay. In particular, too small or too large \(β\) values corrupt the training process (giving very low accuracy performance). For
CUB200-2011, the performance peaks at around \( \beta = 15 \).

For parameter \( C \), as shown in Figure 4(c), training on CIFAR-100/mini-ImageNet achieves the best results at around \( C = 200 \). Although training on CUB200-2011 prefers a much smaller \( C = 10 \), there is no intrinsic difference among these datasets. This is because CIFAR-100/mini-ImageNet has 500 samples per base class while CUB200-2011 has only 30. The augmentation ratios w.r.t. the base dataset are \( \frac{200}{500} = 0.4 \) for CIFAR-100/mini-ImageNet and \( \frac{10}{30} = 0.33 \) for CUB200-2011, which are close enough among these three datasets. In particular, we include the accuracies of CIFAR-100 reported on the base dataset \( D_{0}^{\text{test}} \) and the new class dataset \( D_{1:K}^{\text{test}} \) separately (see the dash lines in Figure 4(d)). We observe a consistent increasing (decreasing) performance pattern for base (new) classes with a larger \( C \) value, indicating that the AFC indeed balances the focus between the base and new classes.

5. Conclusion

In this paper, we propose Gaussian kernel embedded analytic learning (GKEAL) to handle the few-shot class incremental learning task. One key component of GKEAL is the kernel analytic module, containing a Gaussian embedding process which re-embeds the feature trained on the base dataset to produce more discriminative embeddings and a least-square classifier. The augmented feature concatenation module is another key contribution that balances base-new knowledge to enhance overall performance. Our experiments have conducted various empirical analysis (e.g., ablation study and parameter analysis), showing outstanding performance compared with the state-of-the-art methods.

Contributions

The contributions of authors in this work can be described as follows. Huiping Zhuang proposed the idea of GKEAL and did the experiments with Zhenyu Weng and Run He. Huiping Zhuang, Zhenyu Weng, Zhiping Lin and Ziqian Zeng contributed to the paper writing. Huiping Zhuang and Run He contributed to the rebuttal and revised the paper together with Zhiping Lin and Ziqian Zeng. All the authors contributed to the paper structure.

References


[23] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 5, 6, 7


[28] Han-Jia Ye, Hexiang Hu, and De-Chuan Zhan. Learning adaptive classifiers synthesis for generalized few-


[31] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12455–12464, June 2021. 1, 2, 5, 6, 7

[32] Hanbin Zhao, Yongjian Fu, Mintong Kang, Qi Tian, Fei Wu, and Xi Li. Mgsvf: Multi-grained slow vs. fast framework for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2


