Coarse-To-Fine Incremental Few-Shot Learning

Xiang Xiang[1⋆](✉), Yuwen Tan[1], Qian Wan[1⋆⋆], Jing Ma[1], Alan L. Yuille[2], and Gregory D. Hager[2]

[1] Key Lab of Image Processing and Intelligent Control, Ministry of Education
School of AI and Automation, Huazhong Univ. of Science and Tech., China
[2] Department of Computer Science, Johns Hopkins University, USA

Abstract. Different from fine-tuning models pre-trained on a large-scale dataset of preset classes, class-incremental learning (CIL) aims to recognize novel classes over time without forgetting pre-trained classes. However, a given model will be challenged by test images with finer-grained classes, e.g., a basenji is at most recognized as a dog. Such images form a new training set (i.e., support set) so that the incremental model is hoped to recognize a basenji (i.e., query) as a basenji next time. This paper formulates such a hybrid natural problem of coarse-to-fine few-shot (C2FS) recognition as a CIL problem named C2FSCIL, and proposes a simple, effective, and theoretically-sound strategy Knowe: to learn, freeze, and normalize a classifier's weights from fine labels, once learning an embedding space contrastively from coarse labels. Besides, as CIL aims at a stability-plasticity balance, new overall performance metrics are proposed. In that sense, on CIFAR-100, BREEDS, and tieredImageNet, Knowe outperforms all recent relevant CIL or FSCIL methods.

Keywords: theory, class-incremental learning, coarse-to-fine, few shots

1 Introduction

Product visual search is normally driven by a deep model pre-trained on a large-scale private image-set, while at inference it needs to recognize consumer images at a finer granularity. For example, given a tree-like product catalog at Amazon.com, there is a class hierarchy per tree. However, it is rare to add the root-like categories, such as breads, fruits, meat, *etc.* in the Fresh department and breakfast, snacks, beverages, *etc.* in the Gourmet Food department. That is because such catalogs have set the routine by semantic abstraction and approximate summarization. It is standard to pre-train models at a relatively static scale in such 'super-categories' or coarse levels. Such a model is expected to evolve on-the-fly [30] over time as being used, because fine-tuning (FT) it for specific novel classes induces an increasing number of separate models retrained, and thus is inefficient. In practice, it is common to add leaf-like categories along the use of

⋆ Correspondence to xex@hust.edu.cn. Also with China's Belt & Road Joint Lab on Measur. & Contr. Tech., and Nat. Key Lab of S&T on Multi-Spectral Info Processing.
⋆⋆ Visiting student from Wuhan Research Inst. of Posts & Telecommunications, China.

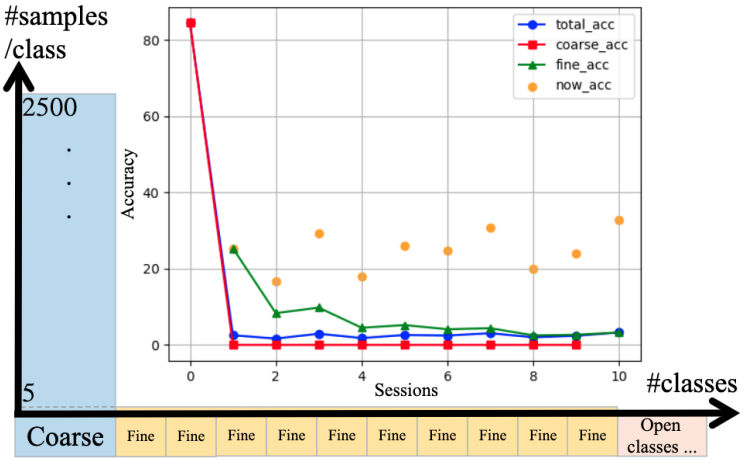

Fig. 1: Catastrophic forgetting when FT-ing a coarsely-trained model on fine samples presently available w/o freezing any weight. We pre-set 10 sessions from CIFAR-100 [21]. There is a fine-class accuracy from the 1st session and yet no coarse-class accuracy as all samples are with fine labels.

Amazon, *e.g.*, under Fruits/Snacks, there is a long list that changes daily. Like humans, new labels of an item can be perceived later and then refine models' knowledge. However, it is rare to pre-train models at such dynamic scale. Such expectations of coarse-to-fine knowledge expansion is also valid for vision-driven autonomous systems. Model developers have a lot of coarsely-labeled samples for training but cannot predict what will be input after deploying it. For example, a self-driving car needs to gradually grow its perception capabilities as it runs.

In this paper, we are interested in a coarse-to-fine recognition problem that fits the class-incremental learning (CIL) setting. Moreover, fine classes appear asynchronously, which again fits CIL. It is also a few-shot learning problem, as there is no time to collect abundant samples per new class. We name such an incremental few-shot learning problem Coarse-to-Fine Few-Shot Class-Incremental Learning (C2FSCIL), and aim to propose a method that can evolve a generic model to both avoid catastrophic forgetting of source-blind coarse classes and prevent over-fitting the new few-shot fine-grained classes. However, **what exactly is the knowledge?** Incremental learning (IL) is aimed for the learning model to adapt to new data without forgetting its existing knowledge. *Catastrophic forgetting* is a concept in connectionist networks [35,20] and occurs when the new weight vector is completely inappropriate as a solution for the originally learned pattern. In deep learning (DL), knowledge distillation (KD) is one of the most effective approaches to IL, while there lacks a consensus about what exactly the knowledge is in deep networks. Will it be the weight vectors?

Is a coarsely-learned embedding space generalizable? We aim to achieve a superior performance at both the coarse and fine granularity. Considering the diversity of fine labels, it is infeasible to train a comprehensive fine-grained model beforehand. Instead, can a model be trained, using coarsely-

labeled samples, to classify finely-labeled samples with accuracy comparable to that of a model trained with fine labels [11]? Our hypothesis is yes; then, the next question is how to pre-train a generalizable base model? How to explore a finer embedding space from coarse labels? Namely, what type of knowledge is useful for fine classes and how can we learn and preserve them [7]?

Can we balance old knowledge and current learning? (*a.k.a.*, the stability–plasticity dilemma [34,52]). We aim to remember cues of both the pre-trained base classes and fine classes in the previous few-shot sessions. Our hypothesis is yes and our preference is a linear classifier as it is flexible, data in-demanding, and efficient to train as well as simple for derivation. The next question is how a linear classifier evolves the model effectively with a few shots and yet a balanced performance. As presumed, if the knowledge is weights, then freezing weights retains knowledge while updating weights evolves it.

To answer the questions, we propose a new problem for incrementally learning coarse-to-fine with a few shots and a way to measure balanced performance. We theoretically analyze why learning, freezing, and normalizing weights effectively solves the problem with a base model contrastively learned from coarse labels.

2 Related Work

Weak supervision. Judging from the fine-class stage (Fig. 1 middle to right), if we combine a pre-training set and the support set as a holistic training set, then the few-shot fine-grained recognition using a model pre-trained on coarse samples are similar to the *weakly-supervised learning* and *learning from coarse labels* [6,11,54,55], *e.g.*, C2FS [6]. Ristel *et. al.* investigates how coarse labels can be used to recognize sub-categories using random forests [42] (say, NCM [43]).

Open-Set Learning. Judging from the coarse-class stage [5] (see the left side of Fig. 1), CIL [38] can be dated back to the SVM [23] and random forest [43,42], where a new class can be added as a new node, and now seen as a progressive case of *continual/lifelong learning* [8,30], where CF is a challenge as data are hidden. The topology structure is also favored in DL [49,48]. *Few-shot learning* (FSL) measures models' ability to quickly adapt to new tasks [50] and has a flavor of CIL considering novel classes in the support set [13,39,49,10,56].

Incremental Learning (IL). IL allows a model to be continually updated on new data without forgetting, instead of training a model once on all data. There are two settings: class-IL [33] and task-IL [8]. They share main approaches, such as regularization and rehearsal methods. Regularization methods prevent the drift of consolidated weights and optimize network parameters for the current task, *e.g.*, parameter control in EWC [20]. CIL is our focus and aims at learning a classifier that maintains a good performance on all classes seen in different sessions. Li *et.al.* first introduces KD [12] to IL in LwF [26] by modifying a cross-entropy loss to retain the knowledge. Recent works focus on retaining old-class samples to compute the KD loss. For example, iCaRL [38] learns both features and strong classifiers by combining KD and feature learning, *e.g.*, NME.

Incremental Few-Shot Learning (IFSL). In the IFSL [39] or similarly FSCIL [49] setting, samples in the incremental session are relatively scarce, dif-

ferent from conventional CIL. While IFSL is based on meta learning, IFSL and DFSL [13] both utilize attentions. In FSCIL, a model named TOPIC is proposed, which contains a single neural gas (NG) network to learn feature-space topologies as knowledge, and adjust NG to preserve the stabilization and enhance the adaptation. In [10], Dong *et. al.* propose an exemplar relation KD-IL framework to balance the tasks of old-knowledge preserving and new-knowledge adaptation as done in [53]. CEC [56] is proposed to separate classifier from the embedding learner, and use a graph attention network to propagate context cues between classifiers for adaptation. In [16], Hou *et. al.* address the imbalance between old and new classes by cosine normalization [51,13,16].

Operating Weights for IL. The IL literature since 2017 has seen various weight operations (op. for short) in the sense of consolidation (*e.g.*, EWC [20]), aligning [57,14], normalization [57,58], standardization [4], regularization [20,36], aggregation [28], calibration [47], rectification [46], transfer [25,29], sharing [41], masking [31], imprinting [37], picking [18], scaling [3], merging [24], pruning [32], quantizaton [45], weight importance [19], assignment [17], restricting weights to be positive [57], constraining weight changes [22], and so on.

Different from existing settings [10,49,56] that focus on remembering the pre-trained base classes only, our setting requires remembering the knowledge gained in both the base coarse and previous fine sessions. We add finer classes instead of new classes at the same granularity. Our setting requires a balance between coarse and fine performance unexplored by existing works . C2FS can be seen as going from our Session-0 to Session-1, while our setting has more incremental sessions and is a derived clean one among the **mixed** setups in IIRC [1]. **Different** from exisiting approaches, we do not follow rehearsal methods, namely, our model learns without memorizing samples [9]. However, retaining samples is often infeasible, say, when learning on-the-fly [30]. Even if there is memory for storing previous samples, there often is a budget, buffer, or queue. Thus, we aim to examine the extreme case of knowledge forgetting, and thus design IFSL methods to the upper-bound extent. For example, although in [22] they do not use any base-class training samples and keep the weights of the base classifier frozen, they still use previous samples in their third phase.

3 A New Problem C2FSCIL

Given a model parameterized by $\mathbf{\Theta}$ and pre-trained on $\mathbb{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where $y_i \in \mathbb{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, ..., \mathcal{Y}_R\}$, a set of R coarse labels $\mathcal{Y}$, we have a stream of C-way K-shot support sets $\mathbb{S}^{(1)}, \mathbb{S}^{(2)}, ..., \mathbb{S}^{(t)}, ..., \mathbb{S}^{(T)}$ where $\mathbb{S}^{(t)} = \{(\mathbf{x}_j^{(t)}, y_j^{(t)})\}_{j=1}^{C \cdot K}$ and $y_j^{(t)} \in \mathbb{Z}^{(t)} = \{\mathcal{Z}_1^{(t)}, ..., \mathcal{Z}_C^{(t)}\}$, a set of C fine-grained labels $\mathcal{Z}$. Then, we adapt our model to $\mathbb{S}^{(1)}, \mathbb{S}^{(2)}, ..., \mathbb{S}^{(t)}$ over time and update the parameter set $\mathbf{\Theta}$ from $\mathbf{\Theta}^{(0)}$ all the way to $\mathbf{\Theta}^{(t)}$. For testing, we also have a stream of $(C \cdot t + R)$-way H-shot query sets $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, ..., \mathbb{Q}^{(t)}, ..., \mathbb{Q}^{(T)}$ where $\mathbb{Q}^{(t)} = \{(\mathbf{x}_k^{(t)}, y_k^{(t)})\}_{k=1}^{(Ct+R)H}$ and $y_k^{(t)} \in \cup_{l=1}^{t} \mathbb{Z}^{(l)} \cup \mathbb{Y}$, which is the generalized union of all label sets till the t-th session. Notably, $\mathbb{Z}^{(t_1)} \cap \mathbb{Z}^{(t_2)} = \varnothing$, $\forall t_1, t_2$. We assume no sample can be retained

(unlike rehearsal methods) and the CIL stage only includes (sub-classes of) base classes. At the t-th session, only the support set $\mathbb{S}^{(t)}$ can be used for training. We set the base of our subsequent theoretical analysis with two definitions [52]. Notably, as we only analyze the last layer, we take off the layer index l therein.

Definition A (Stability). *When the model $\mathbf{\Theta}$ is being trained in the t-th session, $\Delta\mathbf{w}_{t,s}$ in each session should lie in the null space of the uncentered feature covariance matrix $\bar{\mathcal{X}}_{t-1} = [\mathbf{X}_{1,1}^T, ..., \mathbf{X}_{t-1,t-1}^T]^T$, namely, if $\bar{\mathcal{X}}_{t-1}\Delta\mathbf{w}_{t,s} = 0$ holds, then $\mathbf{\Theta}$ is stable at the t-th session's s-th step.*

Note $\mathbf{w}$ is the classification-layer's weight vector, $\Delta\mathbf{w}$ is the change of $\mathbf{w}$, t indexes the session, and s indexes the training step. $\mathbf{X}_{p,p}$ where $p < t$ in $\bar{\mathcal{X}}_{t-1}$ is the input features of classification-layer on p-th session using classification-layer's weight trained on p-th session. We call it the absolute stability.

Definition B (Plasticity). *Assume that the network $\mathbf{\Theta}$ is being trained in the t-th session, and $\mathbf{g}_{t,s} = \{g_{t,s}^1, ..., g_{t,s}^L\}$ denotes the parameter update generated by Gradient Descent for training $\mathbf{\Theta}$ at step s. If $\langle\Delta\mathbf{w}_{t,s}, \mathbf{g}_{t,s}\rangle > 0$ holds, then $\mathbf{\Theta}$ preserves plasticity at the t-th session's s-th step.*

If the inequality condition holds, the $\mathbf{\Theta}$'s loss deceases and thus $\mathbf{\Theta}$ is learning.

4 A Simple Approach Knowe

4.1 Learning Embedding-Weights Contrastively

Now, we elaborate on how we train a generalizable base embedding space [50,27]. cWe follow ANCOR [6] to use MoCo [15] as the backbone, and keep two network streams each of which contains a backbone network with the last-layer FC replaced by a Multi-Layer Perceptron (MLP). The hidden layer of two streams' MLP outputs intermediate $\mathbf{q}$ and $\mathbf{k}$, respectively. Given coarse labels, the total loss is defined as $\mathcal{L}^c = \mathcal{L}_{Con} + \mathcal{L}_{CE}^c$ where

$$\mathcal{L}_{Con} = -\sum_{n=1}^{N} log\frac{exp(\mathbf{q}_n^{\mathrm{T}}\mathbf{k}_n^+/\tau)}{exp(\mathbf{q}_n^{\mathrm{T}}\mathbf{k}_n^+/\tau) + \sum_{m\neq n} exp(\mathbf{q}_n^{\mathrm{T}}\mathbf{k}_m^-/\tau)}, \tag{1}$$

and $\mathcal{L}_{CE}^c$ is the standard cross-entropy loss that captures the inter-class cues. We also use angular normalization [6] to improve their synergy.

Note that m, n index samples, τ is a temperature parameter, $\mathbf{k}_m^-$ denotes the intermediate output of the m-th sample, a negative sample, in the same class with the n-th sample, a positive sample, so as to capture intra-class cues (fine cues), and reduce unnecessary noises to the subsequent fine-grained classification [54]. $\mathcal{L}_{Con}$ will be small when $\mathbf{q}_n$ is similar with $\mathbf{k}_n^+$ and different from $\mathbf{k}_m^-$.

4.2 Freezing Memorized Classifier-Weights

In the t-th incremental session, the task is similar to FSL where a support set $\mathbb{S}^{(t)}$ is offered to train a model to be evaluated on a query set $\mathbb{Q}^{(t)}$. However, FSL only evaluates the classification accuracy of the classes appeared in the support set $\mathbb{S}^{(t)}$. In our setting, the query set $\mathbb{Q}^{(t)}$ contains base classes, and

all classes in previous support sets. No matter freezing embedding-weights helps or not, it does not hurt. We do so, hoping it to reduce model complexity to avoid over-fitting. As past samples are not retained, we freeze the classifier-weights of past classes to implicitly retain the label information and only train the augmented weight matrix $\mathbf{W}$ where in the t-th session, we have $\mathbf{W}_{[B:E]} = [\mathbf{w}_1^{(t)}|\mathbf{w}_2^{(t)}|...|\mathbf{w}_C^{(t)}]_{d\times C}$ with $B = R + C \cdot (t-1) + 1$, $E = R + C \cdot t$ for $t \geq 1$, except $\mathbf{W}_{[:R]} = [\mathbf{w}_1^{(0)}|\mathbf{w}_2^{(0)}|...|\mathbf{w}_R^{(0)}]_{d\times R}$ where d is the feature dimension.

4.3 Normalizing Classifier-Weights

In the last layer, we set the bias term to 0. For a sample $\mathbf{x}$, once a neuron has its output logit $o = \mathbf{w}^{\mathrm{T}}\mathbf{f}(\mathbf{x})$ ready, then a Softmax activation function $Smx(\cdot)$ is applied to convert o to a probability so that we can classify $\mathbf{x}$. ($^{\mathrm{T}}$ is transpose)

However, such an inner-product linear classifier often favors new classes [16]. Instead, we compute the logit using the normalized inner-product [51] (*a.k.a.*, cosine similarity, cosine normalization [13,16]) as $\tilde{o} = \tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{f}}(\mathbf{x})$ where $\mathcal{L}_2$-normalized $\tilde{\mathbf{f}}(\mathbf{x}) = \mathbf{f}(\mathbf{x})/\|\mathbf{f}(\mathbf{x})\|_2$ and $\tilde{\mathbf{w}}_i = \mathbf{w}_i/\|\mathbf{w}_i\|_2$, and then apply Softmax to the rescaled logit $\tilde{o}$ as

$$p_i(\mathbf{x}) = Smx(\tilde{o}/\lambda) = \frac{exp(\tilde{\mathbf{w}}_i^{\mathrm{T}}\tilde{\mathbf{f}}(\mathbf{x})/\lambda)}{\sum_j exp(\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{f}}(\mathbf{x})/\lambda)} \tag{2}$$

where i is the class index, λ is a temperature parameter that rescales the Softmax distribution, as $\tilde{o}$ is ranged of $[-1, 1]$. In the t-th session, we minimize the following cross-entropy loss on the support set $\mathbb{S}^{(t)}$:

$$\mathcal{L}_{CE}^{\mathbb{S}^{(t)}} = -\frac{1}{C \cdot K}\sum_{n=1}^{C\cdot K}\sum_{i=1}^{R+t*C}\delta_{y_n^{(t)}=i} log[p_i(\mathbf{x}_n^{(t)})] \tag{3}$$

where $\delta_{y_n^{(t)}=i}$ is the indicator function.

5 Theoretical Analysis of Knowe for Stability-Plasticity

In this theory, we decouple the embedding learner and classifier, a linear FC layer, freeze weights of the embedding learner, and use the conventional Softmax cross-entropy loss. Different from the conventional FC layer, we freeze weights of neurons corresponding to previous classes. Now we extend **Def. A** and **B**.

Definition 1 (Stability Decay). *For the same input sample, let $\tilde{\mathbf{o}}_i^{(t)}$ denote the output of the i-th neuron in the last layer in the t-th session. After the loss reaches the minimum, we define the decay of stability as $\mathcal{D} = \sum_i(\frac{\tilde{\mathbf{o}}_i^{(T)} - \tilde{\mathbf{o}}_i^{(t)}}{\tilde{\mathbf{o}}_i^{(t)}})^2$.*

Definition 2 (Relative Stability). *Given models $\mathbf{\Theta}_a$ and $\mathbf{\Theta}_b$, if $0 \leq \mathcal{D}_a < \mathcal{D}_b$, then we say $\mathbf{\Theta}_a$ is more stable than $\mathbf{\Theta}_b$.*

Assuming embedding-weights are frozen, then we have:

Proposition 1 (Normalizing or freezing weights improves stability; doing both improves the most). *Given $\mathbf{\Theta}_a$, if we only normalize weights of a linear FC*

classifier, we obtain $\mathbf{\Theta}_b$; if we only freeze them, we obtain $\mathbf{\Theta}_c$; if we do both, we obtain $\mathbf{\Theta}_d$. Then, $\mathcal{D}_d < \mathcal{D}_b < \mathcal{D}_a$ and $\mathcal{D}_d < \mathcal{D}_c < \mathcal{D}_a$.

Proof. (1) Stability Degree of model $\mathbf{\Theta}_a$.

It is assumed that the training for all sessions will reach the minimum loss. For the training sample m in the 0-th session, the probability that m belongs to superclass is one, i.e., $p^m_{t,c_{super}} = 1$ and $p^m_{t,i} = 0(i \neq c_{super})$. According to $p^m_i = \frac{\exp(o^m_i)}{\sum_{j=1}^{I}\exp(o^m_j)}$, the following conditions are satisfied,

$$\tilde{\mathbf{o}}^{(t)}_{c_{super}} = a(a \in \mathbb{R}), \tilde{\mathbf{o}}^{(t)}_i(i \neq c_{super}) = -\infty. \tag{4}$$

After training of T-th session has reached the minimum loss, $\tilde{\mathbf{o}}^{(T)}_{c_{sub}} = b(b \in \mathbb{R}), \tilde{\mathbf{o}}^{(T)}_i(i \neq c_{sub}) = -\infty$, then,

$$\mathcal{D}_a = \sum_i (\frac{\tilde{\mathbf{o}}^{(T)}_i - \tilde{\mathbf{o}}^{(t)}_i}{\tilde{\mathbf{o}}^{(t)}_i})^2 = \quad (\frac{-\infty - a}{a})^2 + (\frac{b-(-\infty)}{-\infty})^2 = \infty. \tag{5}$$

Similarly, we can analyze the stability degree for $\mathbf{\Theta}_b$, $\mathbf{\Theta}_c$, $\mathbf{\Theta}_d$. Please see the full proof in *Appendix*. Our second claim is about normalization for plasticity.

Proposition 2 (Weights normalized, plasticity remains). *To train our FC classifier, if we denote the loss as $\mathcal{L}(\mathbf{w})$ where $\mathbf{w}$ is normalized, the weight update at each step as $\Delta\mathbf{w}$, and the learning rate as α, then we have $f\mathcal{L}(\mathbf{w}-\alpha\Delta\mathbf{w}) < \mathcal{L}(\mathbf{w})$.*

Proof. For a sample m whose feature vector is $\mathbf{x}$, the output of i-th neuron is

$$\mathbf{o}_i = \sigma(\mathbf{x}\cdot\mathbf{w}^i) = \cos\theta_i = \frac{\mathbf{x}\cdot\mathbf{w}^i}{\|\mathbf{x}\|_2\|\mathbf{w}^i\|_2}. \tag{6}$$

The probability of sample m belonging to i-th class is

$$p_i = \frac{\exp(\mathbf{o}_i)}{\sum_{j=1}\exp(\mathbf{o}_j)} \tag{7}$$

And the loss of training is denoted as

$$\mathcal{L}(\mathbf{w}) = -\sum_i y_i log(p_i) \tag{8}$$

where y_i denotes the label of sample m. Denote the weights update of the i-th neuron in linear FC layer as $\Delta\mathbf{w}^i$, then

$$\Delta\mathbf{w}^i = \begin{cases} (p_i - 1)(\frac{\mathbf{x}}{\|\mathbf{x}\|_2\|\mathbf{w}^i\|_2} - \frac{\mathbf{w}^i(\mathbf{x}\cdot\mathbf{w}^i)}{\|\mathbf{x}\|_2\|\mathbf{w}^i\|_2^3}), & i = c \\ p_i(\frac{\mathbf{x}}{\|\mathbf{x}\|_2\|\mathbf{w}^i\|_2} - \frac{\mathbf{w}^i(\mathbf{x}\cdot\mathbf{w}^i)}{\|\mathbf{x}\|_2\|\mathbf{w}^i\|_2^3}), & i \neq c \end{cases} \tag{9}$$

According to $\hat{\mathbf{w}} = \mathbf{w} - \alpha\Delta\mathbf{w}$, we have

$$\hat{\mathbf{w}}^i = \begin{cases} \mathbf{w}^i + \alpha(1-p_i)\frac{1}{\|\mathbf{w}^i\|_2}(\frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\mathbf{w}^i}{\|\mathbf{w}^i\|_2}\cos\theta_i), & i = c \\ \mathbf{w}^i - \alpha p_i\frac{1}{\|\mathbf{w}^i\|_2}(\frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\mathbf{w}^i}{\|\mathbf{w}^i\|_2}\cos\theta_i), & i \neq c \end{cases} \tag{10}$$

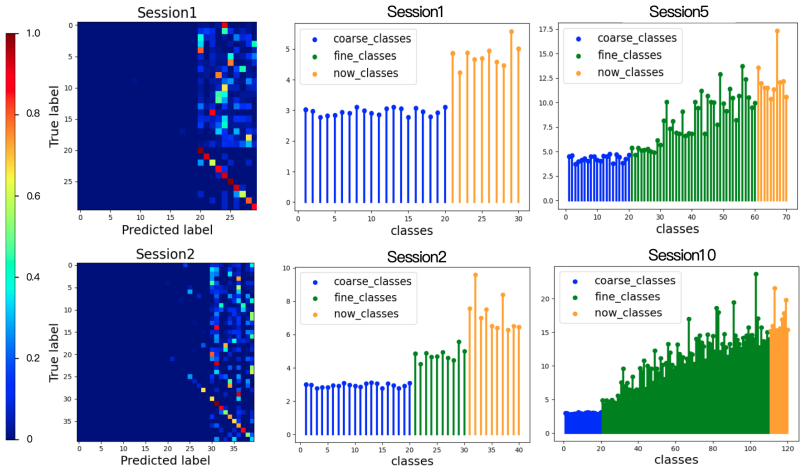

Fig. 2: 10-way 5-shot confusion matrix (left) and visualization of the norm of raw weights (mid-right) in the last layer for old/new classes. As each session can only access labels of the present classes, a linear classifier will have a larger weight for the current classes' neurons, inducing the queries of previous classes to be likely assigned into current classes' region (left) in the embedding space. (CIFAR-100)

By denoting $h(\alpha) \triangleq \mathcal{L}(\mathbf{w} - \alpha\Delta\mathbf{w})$, according to Taylor's theorem, we have

$$\mathcal{L}(\mathbf{w} - \alpha\Delta\mathbf{w}) = \mathcal{L}(\mathbf{w}) - \alpha \left\langle \Delta\mathbf{w}, \mathbf{g} \right\rangle + o(\alpha) \tag{11}$$

where $\frac{|o(\alpha)|}{\alpha} \to 0$ when $\alpha \to 0$. Therefore, there exists $\overline{\alpha} > 0$ such that

$$|o(\alpha)| < \alpha | \left\langle \Delta\mathbf{w}, \mathbf{g} \right\rangle |, \forall \alpha \in (0, \overline{\alpha}) \tag{12}$$

With $\mathbf{g} = \frac{\partial \mathcal{L}(\hat{\mathbf{w}})}{\partial \hat{\mathbf{w}}} = \Delta\hat{\mathbf{w}}$, the calculation leads to the conclusion that $\langle \Delta\mathbf{w}, \mathbf{g} \rangle = \sum_i \Delta\mathbf{w}^i \Delta\hat{\mathbf{w}}^i > 0$, and thus $\mathcal{L}(\mathbf{w} - \alpha\Delta\mathbf{w}) < \mathcal{L}(\mathbf{w})$ for all $\alpha \in (0, \overline{\alpha})$. Weights update $\Delta\mathbf{w}$ is the descent direction.

Notably, freezing the weights does not affect plasticity. As shown in Fig. 2, samples of classes seen in the 1st session are totally classified to classes seen in the 2nd session while only samples of the present classes can be correctly classified. We plot weight norms to find them grow and propose a conjecture implying a need of normalization.

Conjecture 1 (FC weights grow over time). *Let $\|\mathbf{W}^{(t)}\|_F$ denotes the Frobenius norm of the weight matrix formed by all weight vectors in the FC layer for new classes in the t-th session. With training converged and norm outliers ignored, it holds that $\|\mathbf{W}^{(t)}\|_F > \|\mathbf{W}^{(t-1)}\|_F, \forall t \in \{1, ..., T\}$.*

Analysis. For a conventional linear FC layer, the output of neural network directly determines the probability of which class the sample belongs to. Thus, we use $\Delta\mathbf{o}_i$ to represent the reward ($\Delta\mathbf{o}_i > 0$) or penalty ($\Delta\mathbf{o}_i < 0$) for different neurons after sample $\mathbf{x}$ with label c is trained, where $\mathbf{o}_i = \mathbf{x} \cdot \mathbf{w}^i$ is the output

of the i-th neuron and $\alpha > 0$ is the learning rate. Then, we have

$$\Delta \mathbf{o}_i = \begin{cases} \alpha(1 - p_i)\|\mathbf{x}\|_2^2 \geq 0, & i = c \\ -\alpha p_i \|\mathbf{x}\|_2^2 \leq 0, & i \neq c \end{cases} \tag{13}$$

For a sample m with super-class label c_{super} and sub-class label c_{sub}, when we train sample m only with label c_{super} and reach a relatively good state in the 0-th session, we will get $p^m_{c_{super}} \to 1$ and $p^m_i (i \neq c_{super}) \to 0$. When we train sample m only with label c_{sub} in other sessions and reach a relatively good state, the penalty for superclass of sample m will be much larger than other classes, meanwhile the reward for subclass of sample m will be much larger too. Therefore, if i belongs to previously-seen classes, $i \neq c$ will hold most of the time during training. Thus, previously-seen classes will keep being penalized during the gradient descent. As a result, the weights of previously-seen classes are prone to be smaller than those for the newly added classes. And because we train new classes in stages and reach a relatively good state (say, the training loss converges to small value) for all sessions, the FC weights will piecewisely grow over time. Therefore, the model is consequently biased towards new classes.

As for freezing embedding weights. We have (see the analysis in *Appendix*):
Conjecture 2 (Sufficient & necessary condition of no impact of freezing embedding-weights). $p \vee q \Leftrightarrow \neg r$ *where*

p: classifier-weights are normalized,

q: classifier-weights are frozen,

r: freezing embedding-weights improves the performance.

6 Experiments

6.1 New Overall Performance Measures

In this section , we evaluate the model after each session with the query set $\mathbb{Q}^{(t)}$, and report the Top-1 accuracy. The base session only contains coarse labels, and thus is evaluated by the coarse-grained classification accuracy $\mathcal{A}_c$. We evaluate $\mathcal{A}_c$, the fine-grained accuracy $\mathcal{A}_f$, and the total accuracy $\mathcal{A}_t$ per incremental session, except the last session when only fine labels are available and $\mathcal{A}_c$ is not evaluated. We average $\mathcal{A}_t$ to obtain an overall performance score as

$$\bar{\mathcal{A}} = \frac{1}{T+1} \sum_{i=0}^{T} \mathcal{A}_t^i. \tag{14}$$

Inspired by [3], we define the fine-class forgetting rate

$$\mathcal{F}_f^t = \frac{\mathcal{A}_f^{t-1} - \mathcal{A}_f^t}{\mathcal{A}_f^{t-1}}, \tag{15}$$

and the forgetting rate for the base coarse class as

$$\mathcal{F}_c^t = \frac{\mathcal{A}_c^0 - \mathcal{A}_c^t}{\mathcal{A}_c^0}. \tag{16}$$

With them, we can evaluate the model with an overall measure to represent the catastrophic forgetting rate as

$$\mathcal{F} = \frac{1}{T-1}(\sum_{t=2}^{T} \mathcal{F}_f^t * \frac{c_t}{N_f} + \sum_{t=1}^{T-1} \mathcal{F}_c^t * (1 - \frac{c_t}{N_f})) \tag{17}$$

where T is the number of incremental sessions; c_t is the number of appeared fine classes until the t-th session, and N_f is fine-class total number; $\mathcal{A}_c$ and $\mathcal{A}_f$ are the accuracy of coarse and fine classes per session, respectively.

6.2 Datasets and Results

CIFAR-100 contains 60,000 images from 100 fine classes, each of which has 500 training images and 100 test images [21]. They can be grouped into 20 coarse classes, each of which includes 5 fine classes, *e.g.*, *trees* contains *maple, oak, pine, palm*, and *willow*. The 100 fine classes are divided into 10 10-way 5-shot sessions.

BREEDS is derived from ImageNet with class hierarchy re-calibrated by [44] and contains 4 subsets named living17, nonliving26, entity13, and entity30. They have 17, 26, 13, 30 coarse classes, $4, 4, 20, 8$ fine classes per coarse class, 88K, 132K, 334K, 307K training images, 3.4K, 5.2K, 13K, 12K test images.

tieredImageNet (tIN) is a subset of ImageNet and contains 608 classes [40] that are grouped into 34 high-level super-classes to ensure that the training classes are distinct enough from the test classes semantically. The train/val/test set have $20, 6, 8$ coarse classes, $351, 97, 160$ fine classes, 448K, 124K, 206K images. Table 1 summarizes our performance and Fig. 3 visualizes confusion matrices.

6.3 Implementation Details

We use ResNet-50 on BREEDS, '-12' on CIFAR100 and '-12' on tIN, train $\mathbf{\Theta}^{(0)}$ except FC using ANCOR, use SGD with a momentum 0.9, as well as set weight decay to 5e-4, batch size to 256, τ to 0.2, and λ to 0.5. The learning rate is 0.12 for $\mathbf{\Theta}^{(0)}$, and is 0.1 for $\mathbf{\Theta}^{(1)}$,$\mathbf{\Theta}^{(2)}$, *etc.* for 200 epochs. See also project page [3].

6.4 Ablation Study

Impact of base contrastive learning. As shown in Fig 4a, Knowe obtains a better performance than not using MoCo in Knowe's base, which verifies that the

[3] https://github.com/HAIV-Lab/Knowe

Dataset	coarse#	fine#	total#	sessions	way/shot	queries	$\bar{\mathcal{A}}$	$\mathcal{F}$
CIFAR-100	20	100	120	10	10/5	15	38.50	0.42
living17	17	68	85	7	10/1	15	54.62	0.33
nonliving26	26	104	130	11	10/1	15	48.41	0.25
entity13	13	260	273	13	20/1	15	41.45	0.38
entity30	30	240	270	8	30/1	15	47.79	0.32
tieredImageNet	20	351	371	10	36/5	15	33.24	0.39

Table 1: Dataset setting and performance. # is class num.

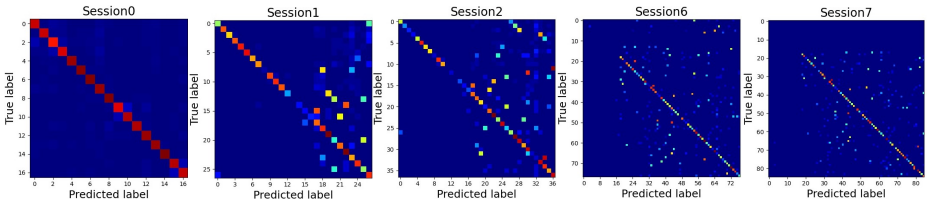

Fig. 3: Confusion matrices of Knowe tested on BREEDS-living17.

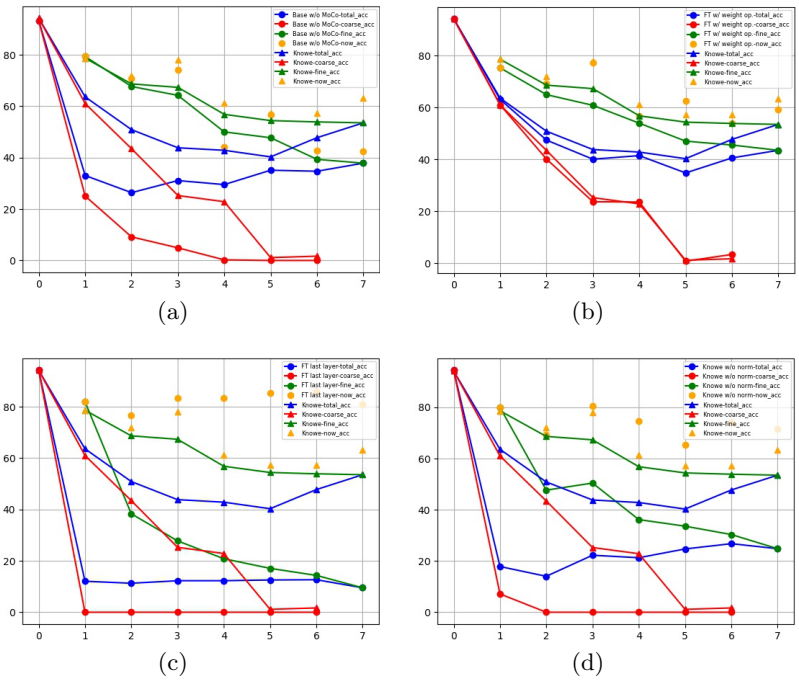

Fig. 4: Ablation study on living17. (a) Contrastive learning? (b) Freezing embedding-weights? (c) Freezing classifier-weights? (d) Normalizing weights?

contrastively-learned base model helps fine-grained recognition. Starting from almost the same fine accuracy in the 2nd session, the gap between w/ MoCo and w/o MoCo increases, as the former stably outperforms the latter on current classes. It verifies that the former can learn more fine knowledge than the latter. Given there are only a few fine-class samples, the extra fine-grained knowledge is likely from the contrastively-learned base model.

Impact of freezing embedding-weights. Fig. 4b illustrates that freezing embedding-weights induces a slightly better performance than not freezing them. *If classifier-weights are normalized and frozen, freezing embedding-weights does not help* ($p \wedge q \Rightarrow \neg r$), which is shown by small changes of $\bar{\mathcal{A}}$ and $\mathcal{F}$ in Table 2.

Impact of freezing memorized classifier-weights. As shown in Fig 4c, there is severe CF of both fine and coarse knowledge when not freezing the

Mehtod	Contr. learn.	Decoupled	Frozen	Normalization	Total accuracy per session 0	1	2	3	4	5	6	7	8	$\bar{\mathcal{A}}\uparrow$	$\mathcal{F}\downarrow$
(a) Base w/o MoCo		✓	✓	✓	93.18	33.04	26.37	31.08	29.51	35.10	34.71	37.84	N/A	40.10	0.50
(b) FT w/ weight op.	✓		✓	✓	94.21	63.14	47.45	40.10	41.47	34.80	40.59	43.53	N/A	50.66	0.35
(c) FT last layer	✓	✓		✓	94.21	12.06	11.28	12.26	12.26	12.55	12.65	9.51	N/A	22.09	0.66
(d) Knowe w/o norm.	✓	✓	✓		94.50	17.84	14.02	22.26	21.28	24.71	26.77	24.80	N/A	30.77	0.57
LwF+ [26]	✓				**94.50**	*61.47*	*44.61*	27.45	19.12	11.28	6.37	4.22	N/A	33.63	0.51
ScaIL [3]	✓				**94.50**	38.63	25.59	31.08	30.29	35.10	37.84	41.08	N/A	41.76	0.48
Weight Align+ [57]	✓	✓	✓		**94.50**	50.98	37.94	*38.43*	*37.06*	*35.20*	*39.80*	*43.24*	N/A	*47.14*	*0.40*
Subsp. Reg.+ [2]	✓	✓	✓		**94.50**	59.41	39.51	33.43	29.31	25.59	27.84	26.47	N/A	42.01	*0.40*
Knowe (Ours)	✓	✓	✓	✓	*94.21*	**63.63**	**50.88**	**43.82**	**42.84**	**40.29**	**47.75**	**53.53**	N/A	**54.62**	**0.33**
ANCOR [6]	✓				94.50	11.86	11.18	12.35	11.77	12.55	10.78	9.02	N/A	21.75	0.66
Jt. train. (upp. bd.)	✓	✓	✓	✓	94.21	63.63	58.53	52.26	46.28	47.75	36.96	42.75	N/A	55.29	0.25
LwF+[26]	✓				**89.48**	**65.03**	*48.69*	22.72	9.36	6.03	4.61	2.86	3.33	28.01	0.47
ScaIL[3]	✓				**89.48**	39.25	25.50	22.44	23.69	25.75	30.81	32.08	35.25	36.03	0.48
Weight Align+ [57]	✓	✓	✓		**89.48**	47.36	37.06	*31.72*	*30.56*	*32.28*	*34.11*	*36.39*	*37.06*	*41.78*	*0.42*
Subsp. Reg.+ [2]	✓	✓	✓		**89.48**	42.39	28.94	20.86	16.14	16.44	16.75	16.17	16.06	29.25	0.48
Knowe (Ours)	✓	✓	✓	✓	*87.90*	*63.22*	**49.22**	**37.75**	**34.78**	**36.25**	**38.03**	**40.08**	**42.83**	**47.79**	**0.32**
ANCOR[6]	✓				89.48	8.67	8.28	9.50	6.83	8.75	9.53	8.19	8.69	17.55	0.61
Jt. train. (upp. bd.)	✓	✓	✓	✓	87.90	63.22	56.56	53.72	47.36	44.78	41.61	38.06	36.75	52.22	0.20

Table 2: Performance on BREEDS living17 (top) and entity30 (bottom).

weights of previously-seen classes, which implies that little knowledge is retained. Although embedding-weights are frozen and classifier-weights are normalized, the coarse knowledge is totally forgotten. It implies that, *if classifier-weights are normalized and yet not frozen, freezing the embedding-weights does not help* ($p \wedge \neg q \Rightarrow \neg r$). It can be explained that fine-tuning on a few samples normally induces little change to the embedding-weights and yet great change to classifier-weights. Moreover, the model without freezing classifier-weights performs much worse than Knowe that freezes previous weights. The gap of the fine accuracy increases over time and is larger than the gap of the present accuracy. It implies that they also differ in the performance of previous fine classes.

Impact of normalizing classifier-weight. Fig 2 has already shown that, with a linear classifier, the weight norms of new classes totally surpass the weight norms of previous classes, which causes that the linear classifier biases towards new classes (*i.e.*, any sample of previous class can be classified as a new class). That implies a need of normlizing the classifer-weights. As shown in Fig 4d, when we freeze weights of previous classes and only tune the weights of new classes w/o normalization, the model performs stably worse, which verifies that normalizing classifier-weights plays a positive role.

More about freezing embedding-weights. We know $\neg p \wedge q \Rightarrow \neg r$. Thus, we have a **Conjecture 3**: $p \vee q \Rightarrow \neg r$, meaning *if classifier-weights are either normalized or frozen, then freezing embedding-weights does not help*. A decent *now_acc* seems a condition for weight freezing and normalization to be effective.

6.5 Performance Comparison and Analysis

Table 2,3,4 and Fig. 5 compare Knowe with SOTA FSCIL or CIL methods including LwF [26], ScaIL [3], Weight Aligning [57] and Subspace Regularizers (Sub. Reg.) [2]. Joint training is non-IL and an *acc* upper bound in principle.

Overall average *acc* $\bar{\mathcal{A}}$ and forgetting rate $\mathcal{F}$. As shown in Table 2,3,4, Knowe has the smallest $\mathcal{F}$ and the largest $\bar{\mathcal{A}}$ on all datasets. From both metrics, Weight

Method	Contr. learn.	Decoupled	Frozen	Normalization	0	1	2	3	4	5	6	7	8	9	10	11	12	13	$\bar{A}\uparrow$	$\mathcal{F}\downarrow$
LwF+ [26]	✓				**86.94**	*65.51*	**58.14**	*44.17*	22.76	14.36	9.68	6.92	5.90	5.19	5.32	3.40	N/A	N/A	27.36	0.38
ScaIL [3]	✓				**86.94**	36.09	24.10	21.47	23.27	23.65	27.95	31.80	34.23	36.09	*37.76*	*38.14*	N/A	N/A	35.12	0.43
Weight Align.+ [57]	✓	✓	✓		**86.94**	61.41	46.03	40.00	35.77	*34.10*	*35.96*	*33.21*	*35.51*	*36.60*	37.56	37.76	N/A	N/A	*43.40*	*0.29*
Subsp. Reg.+ [2]	✓	✓	✓		**86.94**	63.59	52.56	42.95	*35.96*	31.41	28.01	26.15	23.27	19.68	19.36	20.19	N/A	N/A	37.51	**0.25**
Knowe (Ours)	✓	✓	✓	✓	*86.23*	**65.90**	*53.08*	**46.80**	**42.82**	**38.91**	**41.22**	**39.10**	**40.06**	**41.80**	**42.44**	**42.63**	N/A	N/A	**48.41**	**0.25**
ANCOR [6]	✓				**86.94**	5.83	6.03	6.92	5.90	6.60	7.63	7.05	7.05	7.50	7.44	2.63	N/A	N/A	13.13	0.61
Jt. train. (upp. bd.)	✓	✓	✓	✓	86.23	65.90	60.51	59.04	53.53	53.85	46.73	46.60	43.85	36.67	37.31	36.80	N/A	N/A	52.25	0.16
LwF+[26]	✓				**92.03**	*59.10*	*43.64*	18.49	10.49	6.82	3.59	2.54	3.10	2.56	2.10	2.23	1.77	1.54	17.86	0.52
ScaIL[3]	✓				**92.03**	37.10	13.92	13.36	14.87	18.36	21.72	23.28	24.33	27.62	29.59	31.54	32.36	34.08	29.58	0.49
Weight Align+ [57]	✓	✓	✓		**92.03**	36.74	24.15	*20.51*	*22.31*	*24.82*	*26.41*	*26.85*	*27.26*	*31.49*	*32.26*	*35.28*	*36.72*	*37.69*	*33.89*	0.46
Subsp. Reg.+ [2]	✓	✓	✓		**92.03**	52.72	28.95	15.92	12.08	10.82	10.90	11.49	12.05	12.03	11.77	11.72	12.54	14.36	22.10	*0.45*
Knowe (Ours)	✓	✓	✓	✓	*91.35*	**66.90**	**45.69**	**35.54**	**30.56**	**29.21**	**30.10**	**29.95**	**30.85**	**33.74**	**35.36**	**38.54**	**40.26**	**42.21**	**41.45**	**0.38**
ANCOR[6]	✓				92.03	5.36	5.67	5.49	5.18	6.51	5.82	4.80	5.39	6.28	5.36	5.13	5.26	5.62	11.71	0.57
Jt. train. (upp. bd.)	✓	✓	✓	✓	91.35	66.90	57.54	49.92	50.59	48.64	47.69	44.41	41.72	39.13	39.62	40.72	38.49	37.26	49.57	0.24

Table 3: Performance on BREEDS nonliving26 (top) and entity13 (bottom).

Method	Contr. learn.	Decoupled	Frozen	Normalization	0	1	2	3	4	5	6	7	8	9	10	$\bar{A}\uparrow$	$\mathcal{F}\downarrow$
LwF+[26]	✓				**78.39**	**41.87**	28.00	23.80	14.93	10.53	8.00	8.80	6.47	7.33	6.73	21.35	0.51
ScaIL[3]	✓				**78.39**	14.47	14.13	18.07	21.00	25.20	26.20	31.87	*32.60*	36.53	38.20	30.61	0.52
Weight Align+ [57]	✓	✓	✓		**78.39**	13.20	14.13	18.20	21.20	24.60	*26.93*	*32.33*	*32.60*	*38.93*	*38.46*	30.82	0.53
Subsp. Reg.+ [2]	✓	✓	✓		**78.39**	*41.47*	**31.80**	**32.87**	*26.73*	*25.73*	25.27	26.73	24.27	25.73	24.00	*33.00*	*0.43*
Knowe (Ours)	✓	✓	✓	✓	*72.07*	36.00	*28.13*	*30.27*	**32.20**	**31.20**	**30.93**	**36.33**	**39.27**	**43.20**	**43.93**	**38.50**	**0.42**
ANCOR[6]	✓				78.39	7.93	7.13	8.27	7.80	8.60	6.40	7.53	6.93	8.20	8.33	14.14	0.59
Jt. train. (upp. bd.)	✓	✓	✓	✓	72.07	36.00	37.07	40.27	40.13	41.33	38.60	41.13	40.47	41.40	43.47	42.90	0.33
LwF+[26]	✓				**87.64**	**69.36**	13.88	4.22	4.05	4.03	3.02	2.74	1.44	1.05	1.06	17.50	0.55
ScaIL[3]	✓				**87.64**	48.51	**33.12**	**26.15**	**22.66**	*22.77*	23.42	22.72	23.38	25.17	26.65	*32.93*	*0.40*
Weight Align+ [57]	✓	✓	✓		**87.64**	25.13	18.63	18.37	20.08	22.20	*24.22*	**24.73**	*26.71*	*29.00*	*30.45*	29.74	0.48
Subsp. Reg.+ [2]	✓	✓	✓		**87.64**	*49.73*	*32.06*	24.35	20.95	20.76	20.84	21.12	21.79	23.15	24.31	31.52	0.42
Knowe (Ours)	✓	✓	✓	✓	*76.15*	48.24	30.60	*25.60*	*22.34*	**23.48**	**24.79**	*24.69*	**27.65**	**30.26**	**31.87**	**33.24**	**0.39**
ANCOR[6]	✓				87.64	7.10	6.69	6.55	6.36	6.57	6.42	6.55	6.55	6.40	5.17	13.82	0.61
Jt. train. (upp. bd.)	✓	✓	✓	✓	76.15	48.24	39.89	34.09	32.21	30.85	28.81	29.86	28.57	28.74	29.06	36.95	0.32

Table 4: Comparison with others on CIFAR-100 (top table) and tieredImageNet.

Aligning ranks 2nd on BREEDS, Sub. Reg. ranks 2nd on CIFAR-100, and ScaIL ranks 2nd on tIN (consistent across two metrics). LwF has poor numbers, which implies that, with no samples retained, KD does not help.

Total accuracy per session decreases over time yet slower and slower for Knowe and SOTA methods. However, outstanding ones decrease first and then rise, because that the proportion of fine classes in the query set gets higher and their accuracy plays a leading role in the total accuracy. Knowe is the best, with a strong rising trend, which satisfies the aim of CIL the most and envisions Knowe continuing performing well when more sessions are added (Table 3) . Sub. Reg. and Weight Align. often have 2nd-best numbers (both freeze weights); ScaIL and LwF occasionally do.

Coarse class accuracy decreases over time unavoidably (see Fig. 5), while Knowe and SOTA methods slow down the decay, with comparable rates. As IL methods, Weight Aligning, ScaIL, and LwF do not forget knowledge totally although they do not operate weights as done by Knowe. As an non-IL approach, ANCOR totally forgets old knowledge from the 1st session because it fine-tunes on the few fine shots without any extra operation to retain coarse knowledge. The joint training on all fine classes till the present is non-IL, and in principle should bound the fine-class performance. Interestingly, it also suffers less from coarse *acc* decay, the rate of which is much lower (Fig. 5). Differently, the cause

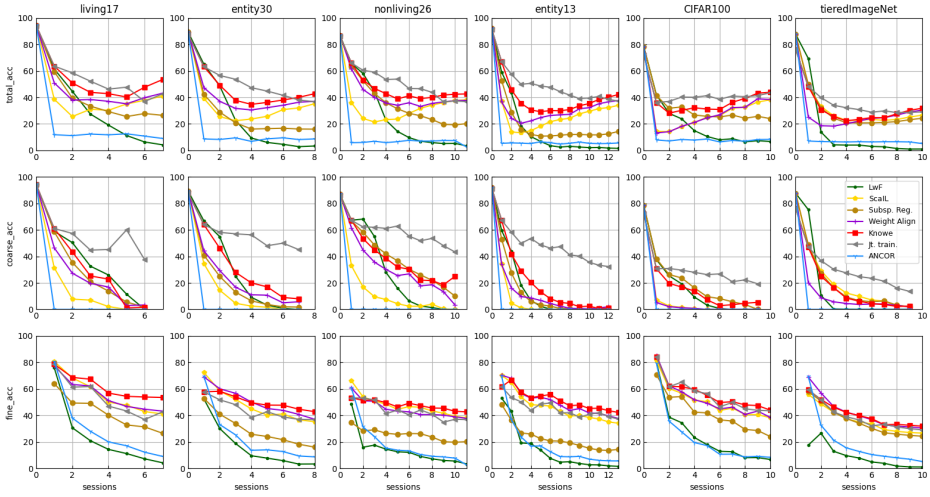

Fig. 5: Accuracy comparison on all datasets. Top-down: total, coarse, fine.

can be imbalance between increasing fine classes and coarse classes. Knowe's performance is very competitive and indeed bounded by joint training.

Fine class's total accuracy normally decreases over time yet slower and slower for Knowe and SOTA methods (Fig. 5), and can be maintained in a similar range for most methods, among which Knowe often stays the highest, ScaIl and Weight Aligning are in the middle, Sub Reg. often stays in a low level, and LwF and ANCOR perform stably the worst. Knowe is the most balanced, while Sub. Reg. biases towards stability that is its drawback. Joint training does not bound the accuracy, possibly due to few shots.

Compared works. All empirically-compared CIL methods and ours are no-rehearsal ones. On the other hand, joint training is rehearsal-based.

7 Conclusion

In this paper, we present a new problem together with new metrics, and theoretically analyze why a simple approach can solve it well in the sense of getting more balanced performance than the SOTA. While it is not new to freeze or normalize weights, we are unaware of them previously being presented as a principled approach (to CIL) that is as simple as fine-tuning. It makes pre-trained big models more useful for finer-grained tasks. For C2FSCIL with a linear classifier, weights seem to be the knowledge. However, how generic are our findings in practice? Can they be applied to general FSCIL? If yes, we are more comfortable with that answer, but then how does a class hierarchy make a difference? Future work will include examining those questions, non-linear classifiers, and so on.

Acknowledgement. This research was supported by Nat. NSFC (62176100), Nat. Key R&D Program of China (2021ZD0201300), HUST Independent Innovation Res. Fund (2021XXJS096), Sichuan Univ. Interdisciplinary Innovation Res. Fund (RD-03-202108), and MoE Key Lab of Image Processing & Intell. Contr.

References

1. Abdelsalam, M., Faramarzi, M., Sodhani, S., Chandar, S.: IIRC: Incremental Implicitly-Refined Classification. In: CVPR (2021)
2. Akyürek, A.F., Akyürek, E., Wijaya, D., Andreas, J.: Subspace regularizers for few-shot class incremental learning. Arxiv preprint:2110.07059 (2021)
3. Belouadah, E., Popescu, A.: ScaIL: Classifier weights scaling for class incremental learning. In: IEEE/CVF Winter Conference on Applications of Computer Vision (2020)
4. Belouadah, E., Popescu, A., Kanellos, I.: Initial classifier weights replay for memoryless class incremental learning. ArXiv preprint:2008.13710 (2020)
5. Bendale, A., Boult, T.: Towards open world recognition. In: CVPR (2015)
6. Bukchin, G., Schwartz, E., Saenko, K., Shahar, O., Feris, R., Giryes, R., Karlinsky, L.: Fine-grained angular contrastive learning with coarse labels. In: CVPR (2021)
7. Cha, H., Lee, J., Shin, J.: Co2l: Contrastive continual learning. In: ICCV (2021)
8. Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
9. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: CVPR (2019)
10. Dong, S., Hong, X., Tao, X., Chang, X., Wei, X.: Few-shot class-incremental learning via relation knowledge distillation. In: AAAI (2021)
11. Fotakis, D., Kalavasis, A., Kontonis, V., Tzamos, C.: Efficient algorithms for learning from coarse labels. In: 34th Annual Conference on Learning Theory (2021)
12. Geoffrey, H., Oriol, V., Jeff, D.: Distilling the knowledge in a neural network. In: NeurIPS (2015)
13. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: CVPR (2018)
14. He, C., Wang, R., Chen, X.: A tale of two cils: The connections between class incremental learning and class imbalanced learning, and beyond. In: CVPR Workshops (2021)
15. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
16. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: CVPR (2019)
17. Hu, X., Tang, K., Miao, C., Hua, X.S., Zhang, H.: Distilling causal effect of data in class-incremental learning. In: CVPR (2021)
18. Hung, S.C.Y., Tu, C.H., Wu, C.E., Chen, C.H., Chan, Y.M., Chen, C.S.: Compacting, picking and growing for unforgetting continual learning. In: NeurIPS (2019)
19. Jung, S., Ahn, H., Cha, S., Moon, T.: Continual learning with node-importance based adaptive group sparse regularization. In: NeurIPS (2020)
20. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017)
21. Krizhevsky, A.: Learning multiple layers of features from tiny images. Unvieristy of Toronto: Technical Report (2009)
22. Kukleva, A., Kuehne, H., Schiele, B.: Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In: ICCV (2021)

23. Kuzborskij, I., Orabona, F., Caputo, B.: From n to n+1: Multiclass transfer incremental learning. In: CVPR (2013)
24. Lee, J., Hong, H.G., Joo, D., Kim, J.: Continual learning with extended kronecker-factored approximate curvature. In: CVPR (2020)
25. Lee, S.W., Kim, J.H., Jun, J., Ha, J.W., Zhang, B.T.: Overcoming catastrophic forgetting by incremental moment matching. In: NIPS (2017)
26. Li, Z., Hoiem, D.: Learning without forgetting. In: ECCV (2016)
27. Liu, C., Fu, Y., Xu, C., Yang, S., Li, J., Wang, C., Zhang, L.: Learning a few-shot embedding model with contrastive learning. In: AAAI (2021)
28. Liu, Y., Schiele, B., Sun, Q.: Adaptive aggregation networks for class-incremental learning. In: CVPR (2021)
29. Liu, Y., Su, Y., Liu, A.A., Schiele, B., Sun, Q.: Mnemonics training: Multi-class incremental learning without forgetting. In: CVPR (2020)
30. Mai, Z., Li, R., Jeong, J., Quispe, D., Kim, H., Sanner, S.: Online continual learning in image classification: An empirical survey. ArXiv preprint:2101.10423 (2021)
31. Mallya, A., Davis, D., Lazebnik, S.: Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In: ECCV (2018)
32. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: CVPR (2018)
33. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: survey and performance evaluation on image classification. ArXiv preprint:2010.15277 (2020)
34. Mermillod, M., Bugaiska, A., Bonin, P.: The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. Frontiers in Psychology **4** (2013)
35. M.French, R.: Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences **3** (1999)
36. Pan, P., Swaroop, S., Immer, A., Eschenhagen, R., Turner, R.E., Khan, M.E.: Continual deep learning by functional regularisation of memorable past. In: NeurIPS (2020)
37. Qi, H., Brown, M., Lowe, D.G.: Low-shot learning with imprinted weights. In: CVPR (2018)
38. Rebuff, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCaRL: Incremental Classifier and Representation Learning. In: CVPR (2017)
39. Ren, M., Liao, R., Fetaya, E., Zemel, R.S.: Incremental few-shot learning with attention attractor networks. In: NeurIPS (2019)
40. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. In: ICLR (2018)
41. Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., Tesauro, G.: Learning to learn without forgetting by maximizing transfer and minimizing interference. In: ICLR (2019)
42. Ristin, M., Gall, J., Guillaumin, M., Gool, L.V.: From categories to subcategories: Large-scale image classification with partial class label refinement. In: CVPR (2015)
43. Ristin, M., Guillaumin, M., Gall, J., Gool, L.V.: Incremental Learning of NCM Forests for Large-Scale Image Classification. In: CVPR (2014)
44. Santurkar, S., Tsipras, D., Madry, A.: Breeds: Benchmarks for subpopulation shift. ArXiv preprint:2008.04859 (2020)
45. Shi, Y., Yuan, L., Chen, Y., Feng, J.: Continual learning via bit-level information preserving. In: CVPR (2021)

46. Singh, P., Mazumder, P., Rai, P., Namboodiri, V.P.: Rectification-based knowledge retention for continual learning. In: CVPR (2021)
47. Singh, P., Verma, V.K., Mazumder, P., Carin, L., Rai, P.: Calibrating cnns for lifelong learning. In: NeurIPS (2020)
48. Tao, X., Chang, X., Hong, X., Wei, X., Gong, Y.: Topology-preserving class-incremental learning. In: ECCV (2020)
49. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: CVPR (2020)
50. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: a good embedding is all you need? In: ECCV (2020)
51. Wang, F., Xiang, X., Cheng, J., Yuille, A.L.: Normface: L2 hypersphere embedding for face verification. In: ACM Conference on Multimedia (2017)
52. Wang, S., Li, X., Sun, J., Xu, Z.: Training networks in null space of feature covariance for continual learning. In: CVPR (2021)
53. Wu, G., Gong, S., Li, P.: Striking a balance between stability and plasticity for class-incremental learning. In: ICCV (2021)
54. Xu, Y., Qian, Q., Li, H., Jin, R., Hu, J.: Weakly supervised representation learning with coarse labels. In: ICCV (2021)
55. Yang, J., Yang, H., Chen, L.: Towards cross-granularity few-shot learning: Coarse-to-fine pseudo-labeling with visual-semantic meta-embedding. In: ACM Conference on Multimedia (2021)
56. Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P., Xu, Y.: Few-shot incremental learning with continually evolved classifiers. In: CVPR (2021)
57. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.: Maintaining discrimination and fairness in class incremental learning. In: CVPR (2020)
58. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: CVPR (2021)