Supplementary Material for FOSTER: Feature Boosting and Compression for Class-Incremental Learning

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye✉, and De-Chuan Zhan

State Key Laboratory for Novel Software Technology, Nanjing University
wangfuyun@smail.nju.edu.cn,{zhoudw, yehj, zhandc}@lamda.nju.edu.cn

Abstract. The ability to continuously learn new knowledge is considered to be one of the most important symbols of strong artificial intelligence. However, deep neural networks suffer from the severe problem known as catastrophic forgetting when training on new classes. Inspired by the idea that gradient boosting algorithm continuously creates new weak classifiers to fit the residuals between the target and the ensemble model, we propose FOSTER, where we dynamically expand and compress the model when new tasks come, empowering the model to learn new categories adaptively. In the supplementary material, we provide:

(i) Rationality analysis of the substitution (Sec. 1).
(ii) Influence of the initialization of the weight **O** (Sec. 2).
(iii) Introduction to compared methods (Sec. 3).
(iv) Visualization of detailed performance (Sec. 4).

1 Rationality Analysis of the Substitution.

We argue that our simplification of replacing the sum of softmax with softmax of logits sum and substituting the distance metric $\mathrm{Dis}(\cdot,\cdot)$ for the Kullback-Leibler divergence (KLD) $\mathrm{KL}(\cdot \mid\mid \cdot)$. KLD can evaluate the residual between the target and the output by calculating the distance between the target label distribution and the output distribution of categories. KLD is more suitable for classification tasks, and there are some works [2,6] that point out that the KLD has many advantages in many aspects, including faster optimization and better feature representation. Typically, to reflect the relative magnitude of each output, we use non-linear activation softmax to transform the output logits into the output probability. Namely, $p_1, p_2, \ldots, p_{|\hat{\mathcal{Y}}_t|}$, where $0 \le p_i \le 1$, $\sum_{i=1}^{|\hat{\mathcal{Y}}_t} p_i = 1$ and $|\hat{\mathcal{Y}}_t|$ is the number of all seen categories. In classification tasks, the target label is usually set to 1, and the non-target label is set to 0. Therefore, we expect the output of the boosting model can be constrained between 0 and 1. Simply combining the softmax outputs of the original model F_{t-1} and $\mathcal{F}_t$ can not satisfy the constraints. Suppose that the output of F_{t-1} and $\mathcal{F}_t$ in class i are p_i^o and p_i^n, the combination of p_i^n and p_i^o is not in line with our expectation since $0 \le p_i^o + p_i^n \le 2$. By replacing the sum of softmax with softmax of logits sum, we can limit the output of the boosting model between 0 and 1, and the judgment of the two models can still be integrated.

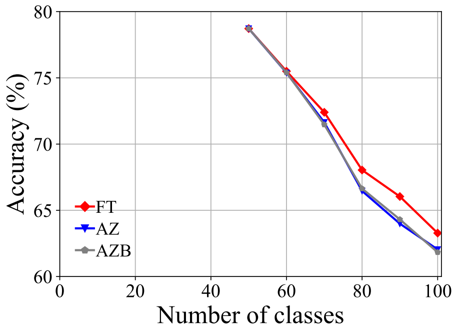


Fig. 1: **Influence of different initialization strategies.**The red line represents FT, the blue line represents AZ, and the gray line represents AZB. The performance of FT is slightly better than AZ and AZB. The performance gap between AZ and AZB is negligible.

2 Influence of the Initialization of the Weight O

In this section, we discuss the effect of the initialization of the weight **O** in the super linear classifier of our boosting model.

$$\mathbf{W}_t^\top = \begin{bmatrix} \mathbf{W}_{t-1}^\top & (\mathcal{W}_t^{(o)})^\top \\ \mathbf{O} & (\mathcal{W}_t^{(n)})^\top \end{bmatrix} . \tag{1}$$

In the main paper, we set **O** to all zero as our default initialization strategy. Therefore, the outputs of the original model for new categories are zero, thus having nothing to do with the classification of new classes.

Here, we introduce three different initialization strategies, including fine-tune (FT), all-zero (AZ), and all-zero with bias (AZB), to further explore the impact of different initialization strategies on performance. Among them, FT is directly training **O** without any restrictions. AZ sets the outputs of the old model on the new class to all zero, and thus the outputs of the model on the new class logits only contain the output of the new model, and the old model does not provide any judgment on the new class. Based on AZ, AZB adds bias learning to balance the logits of the old and new categories. Fig. 1 illustrates the comparison of performance on CIFAR-100 [3] B50 with 5 steps with different initialization strategies. We can see that the performance of using FT initialization strategy is slightly better than that of using AZ and AZB initialization strategies, but the difference is not significant. The performance gap between AZ and AZB is negligible, indicating that the influence of bias is weak.

3 Introduction to Compared Methods

In this section, we will describe in detail the methods compared in the main paper.

Fine-tune: Fine-tune is the baseline method that simply updates its parameters when a new task comes, suffering from catastrophic forgetting. By default, weights corresponding to the outputs of previous classes in the final linear classifier are not updated.

Replay: Replay utilizes the rehearsal strategy to alleviate the catastrophic forgetting compared to Fine-tune. We use herding as the default way of choosing exemplars from the old data.

iCaRL [5]: iCaRL combines cross-entropy loss with knowledge distillation loss together. It retains an old model to help the new model maintain the discrimination ability through knowledge distillation on old categories. To mitigate the classification bias caused by the imbalanced dataset when learning new tasks, iCaRL calculates the center of exemplars for each category and uses NME as the classifier for evaluation.

BiC [8]: BiC performs an additional bias correction process compared to iCaRL, retaining a small validation set to estimate the classification bias resulting from imbalanced training. The final logits are computed by

$$q_k = \begin{cases} o_k & 1 \le k \le n \\ \alpha o_k + \beta & n+1 \le k \le n+m \end{cases}, \tag{2}$$

where n is the number of old categories and m is the number of new ones. the bias correction step is to estimate the appropriate α and β.

WA [10]: During the process of incremental learning, the norms of the weight vectors of new classes are much larger than those of old classes. Based on that, WA proposes an approach called Weight Alignment to correct the biased weights in the final classifier by aligning the norms of the weight vectors of new classes to those of old classes.

$$\widehat{\mathbf{W}}_{new} = \gamma \cdot \mathbf{W}_{new}, \tag{3}$$

where $\gamma = \frac{\text{Mean}(\boldsymbol{Norm}_{\text{old}}\ )}{\text{Mean}(\boldsymbol{Norm}_{new})}$.

PODNet [1]: PODNet proposes a novel spatial-based distillation loss that can be applied throughout the model. PODNet has greater performance on long runs of small incremental tasks.

DER [9]: DER preserves old feature extractors to maintain knowledge for old categories. When new tasks come, DER creates a new feature extractor and concatenates it with old feature extractors to form a higher dimensional feature space. In order to reduce the number of parameters, DER uses the pruning method proposed in HAT [7], but the number of parameters still increases with the number of tasks. DER can be seen as a particular case of our Boosting model. When we set the weight $\mathbf{O}$ of boosting model can be trainable, and remove feature enhancement and logits alignment proposed in the main paper, boosting model can be reduced to DER.

4 Visualization of Detailed Performance

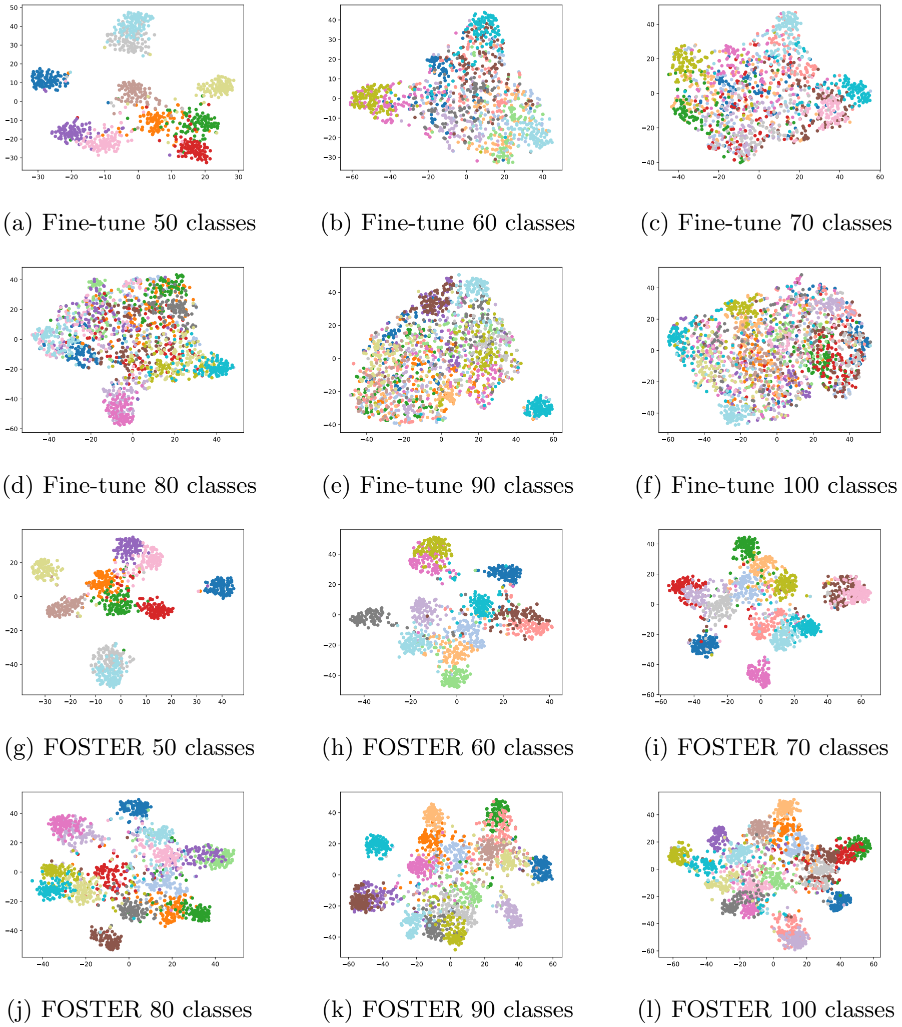
Visualizing Feature Representation. We visualize the feature representations of the test data by t-SNE [4]. Fig. 2 illustrates the comparison of baseline method, fine-tune, with our FOSTER in the setting of CIFAR-100 [3] B50 with 5 steps. As shown in Fig. 2a and Fig. 2g, in the base task, all categories can form good clusters with explicit classification boundaries. However, as shown in Fig. 2b, Fig. 2c, Fig. 2d, Fig. 2e, and Fig. 2f, in stages of incremental learning, the result of category clustering becomes very poor without clear classification boundaries. In the last stage which is shown in Fig.2f, feature points of each category are scattered. On the contrary, as shown in Fig. 2g, Fig. 2h, Fig. 2i, Fig. 2j, Fig. 2k, and Fig. 2l. our FOSTER method can make all categories form good clusters at each incremental learning stage, and has a clear classification boundary, indicating that our FOSTER method is a very effective strategy in feature representation learning and overcoming catastrophic forgetting.

Visualizing Confusion Matrix. To compare with other methods, we visualize the confusion matrices of different methods at the last stage in Fig. 3. In these confusion matrices, the vertical axis represents the real label, and the horizontal axis represents the label predicted by the model. Warmer colors indicate higher prediction rates, and cold colors indicate lower ones. Therefore, the warmer the point color on the diagonal and the colder the color on the other points, the better the performance of the model. Fig. 3a shows the confusion matrix of fine-tune. The brightest colors on the right and colder colors elsewhere suggest that the fine-tune method has a strong classification bias, tending to classify inputs into new categories and suffering from severe catastrophic forgetting. Fig. 3b shows the confusion matrix of iCaRL [5]. iCaRL has obvious performance improvement compared with fine-tune. However, the columns on the right are still bright, indicating that they also have a strong classification bias. In addition, the points on the diagonal have obvious discontinuities, indicating that they cannot make all categories achieve good accuracy. Fig. 3c shows the confusion matrices of WA [10]. Benefiting from Weight Alignment, WA significantly reduces classification bias compared with iCaRL. The rightmost columns have no obvious brightness. Nevertheless, its accuracy in old classes is not high enough. As shown in the figure, most of his color brightness at the diagonal position of the old class is between 0.2 and 0.4. Fig. 3d shows the confusion matrices of DER [9]. DER achieves good results in both old and new categories, but the brightness of the upper right corner shows that it still suffers from classification bias and has room for improvement. As shown in Fig. 3e, our method FOSTER performs well in all categories and well balances the accuracy of the old and new classes.

References

1. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: ECCV. pp. 86–102. Springer (2020)

2. Hu, Z., Hong, L.J.: Kullback-leibler divergence constrained distributionally robust optimization. Available at Optimization Online pp. 1695–1724 (2013)
3. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
4. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
5. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: CVPR. pp. 2001–2010 (2017)
6. Rubinstein, R.Y., Kroese, D.P.: The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning, vol. 133. Springer (2004)
7. Serra, J., Suris, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: ICML. pp. 4548–4557. PMLR (2018)
8. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: CVPR. pp. 374–382 (2019)
9. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: CVPR. pp. 3014–3023 (2021)
10. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: CVPR. pp. 13208–13217 (2020)



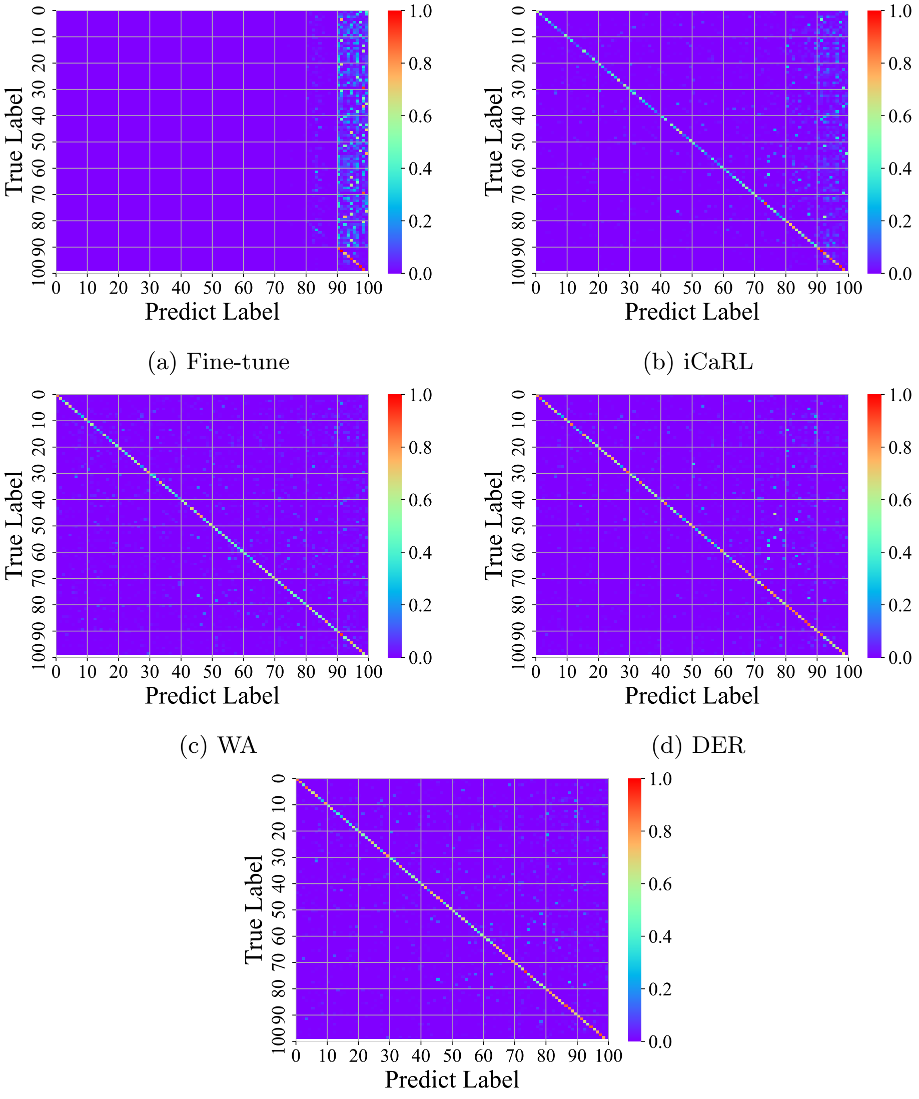
(a) Fine-tune 50 classes (b) Fine-tune 60 classes (c) Fine-tune 70 classes

(d) Fine-tune 80 classes (e) Fine-tune 90 classes (f) Fine-tune 100 classes

(g) FOSTER 50 classes (h) FOSTER 60 classes (i) FOSTER 70 classes

(j) FOSTER 80 classes (k) FOSTER 90 classes (l) FOSTER 100 classes

Fig. 2: **t-SNE [4] visualization of CIFAR-100 [3] B50 with 5 steps.** Figure (a)-(g) shows the t-SNE visualization of fine-tune method. Figure (h)-(l) shows the t-SNE visualization of our method FOSTER. **In order to achieve better results, we normalize each feature and randomly select one category in each five categories for visualization.**



(a) Fine-tune

(b) iCaRL

(c) WA

(d) DER

(e) FOSTER

Fig. 3: **Confusion matrices of different methods.** The vertical axis represents the real label, and the horizontal axis represents the label predicted by the model. The warmer the color of a point in the graph, the more samples it represents.