Supplementary Material for the Paper : Class-Incremental Learning for Action Recognition in Videos

Jaeyoo Park Minsoo Kang Bohyung Han
ECE & ASRI, Seoul National University
{bellos1203,kminsoo,bhhan}@snu.ac.kr

1. Relation about Parameter Regularization Methods

Our approach has something common with some parameter regularization techniques—Elastic Weight Consolidation (EWC) [4] and Synaptic Intelligence (SI) [9]—in the sense that they propose weighting schemes in continual learning scenarios. Both EWC and SI attempt to regularize model parameters using the weights given by either first-order information or cumulative trajectories in the parameter space, respectively; they desire to learn proper representations by backpropagation. On the other hand, our algorithm is based on knowledge distillation, where the learned representations are regularized using the weights given by the impact of individual activation changes with respect to the final loss, and expects the model parameters to be learned for generating the desirable features. Contrary to EWC and SI, which attempt to preserve the representations of old tasks indirectly via parameter regularization, our method optimizes the representation directly, which would be more effective for class-incremental continual learning.

2. Comparison with Parameter Regularization Method

To demonstrate the performance of our approach compared to parameter regularization methods, which we discuss in Section 3.6 of the main paper, we present the results from Elastic Weight Consolidation (EWC) [4] and naïve fine-tuning (FT) under the same memory constraint with ours in Table 1. Although both EWC and our approach attempt to maintain important information in the previous tasks, the proposed method optimizes the objective function directly via knowledge distillation and achieves superior performance. Note that a similar discussion has been made in [3, 7] as well.

Table 1. Class-incremental action recognition performance evaluation on UCF101 and HMDB51 between fine-tuning (FT), EWC and the proposed method. Note that "E" indicates the existence of exemplars. The bold-faced number means the best performance. EWC slightly outperforms the fine-tuning, while our approach surpasses both methods by large margins.

	UCF101			HMDB51	
Num. of classes	10×5 stages	5×10 stages	2×25 stages	5×5 stages	1×25 stages
FT + E	67.65	66.67	65.36	38.58	34.83
EWC [4] + E	69.70	68.12	67.00	39.98	35.94
TCD (Ours) w/o $\mathcal{L}_{\text{ortho}}$	**73.09**	**72.61**	**71.33**	**45.14**	**46.11**
TCD (Ours)	**74.89**	**73.43**	**72.19**	**45.34**	**46.66**

3. Compatibility with Bias Correction Method

In order to show the compatibility of our distillation objective to other kinds of algorithms, we combine the proposed method with an existing bias correction method, Bias Correction (BiC) [8]. For a fair comparison, we replace the classifier of our model with a linear classifier. We set the ratio between training/validation split on the exemplars to $4:1$ to perform BiC method, as we use 5 exemplars per class. Table 2 illustrates the results on UCF101. The performance gap between BiC combined with ours and BiC become larger when the number of incremental steps increase, which implies the robustness of the proposed approach.

Table 2. Compatibility of our distillation loss with the bias correction (BiC) method on UCF101. The bold-faced number indicates the best performance.

Num. of classes	10×5 stages	5×10 stages	2×25 stages
BiC	77.00	74.94	68.85
BiC + TCD (Ours)	**77.22**	**75.63**	**72.00**

4. Effect of Number of Input Frames

We set 8 frames as the input size following the convention of action recognition models [5, 6]. However, our algorithm also works well with different input sizes, which incur a trade-off between accuracy and cost. Table 3 demonstrates the results by varying the number of frames on UCF101 with the same backbone model, TSM. It shows that our algorithm consistently outperforms PODNet regardless of the number of input frames.

Table 3. Effect of input size on UCF101 with 10 stages.

# of frames	16		8		4		1	
Classifier	CNN	NME	CNN	NME	CNN	NME	CNN	NME
PODNet	73.36	74.80	71.58	73.75	70.93	73.47	68.89	71.98
TCD (Ours)	**75.17**	**76.13**	**73.43**	**75.35**	**71.17**	**74.18**	**69.14**	**72.93**

5. Implementation Details

For all experiment, we set the initial learning rate as 0.001 and adopt the SGD optimizer with weight decay of 0.0005. The learning rate is divided by 10 after 20 and 30 epochs. We construct our Local Similarity Classifier (LSC) by using 3 proxies and allow η to be trained throughout the training procedure. For UCF101, we set $\alpha = 1.0$ for the intermediate features and 0.01 for the logit, and set $\beta = 0.1$ for $\mathcal{L}_{\text{ortho}}$. For HMDB51, we set $\alpha = 3.0$ for the intermediate features and 0.1 for the logit, and set $\beta = 0.3$. For Something-Something V2, we set $\alpha = 0.5$ for the intermediate features and 10.0 for the logit, and set $\beta = 10^{-3}$. Following PODNet [1] and UCIR [2], we further multiply an adaptive scaling factor $\lambda = \sqrt{\frac{|\mathcal{C}_{1:k}|}{|\mathcal{C}_k|}}$ to α at each incremental step k, where $\mathcal{C}_{1:k} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_k$ denotes the number of class observed until the incremental step k.

To compare our method to the existing continual learning methods, we reimplement each algorithm and search the hyperparameters using a grid search. For UCIR [2] and PODNet [1], we explore the hyperparameters for the distillation losses, $a \cdot 10^b$, with $a \in \{1, 3, 5\}$ and $b \in \{-2, \cdots, 2\}$. As a result, we set the weight for UCIR as 5 and set the rest of the hyperparameters as same as the original paper. PODNet has two distillation terms, the distillation term for intermediate features and the logit. We set the weight of each loss term to $(0.5, 3.0)$, $(0.1, 1.0)$, and $(1.0, 5.0)$ for UCF101, HMDB51, and Something-Something V2 respectively. The margin for the LSC is set to 0.6 for both PODNet and ours.

References

[1] Arthur Douillard, Matthieu Cord, Charles Ollion, and Thomas Robert. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *ECCV*, 2020.

[2] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a Unified Classifier Incrementally via Rebalancing. In *CVPR*, 2019.

[3] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines. *arXiv preprint arXiv:1810.12488*, 2018.

[4] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[5] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. In *ICCV*, 2019.

[6] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video Classification with Channel-Separated Convolutional Networks. In *ICCV*, 2019.

[7] Gido M van de Ven and Andreas S Tolias. Three Scenarios for Continual Learning. *arXiv preprint arXiv:1904.07734*, 2019.

[8] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large Scale Incremental Learning. In *CVPR*, 2019.

[9] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning through Synaptic Intelligence. In *ICML*, 2017.