Supplementary Material for Few-Shot Class-Incremental Learning

Xiaoyu Tao[1], Xiaopeng Hong[1,3]*, Xinyuan Chang[2], Songlin Dong[1], Xing Wei[2], Yihong Gong[2]
[1]Faculty of Electronic and Information Engineering, Xi'an Jiaotong University
[2]School of Software Engineering, Xi'an Jiaotong University
[3]Research Center for Artificial Intelligence, Peng Cheng Laboratory
txy666793@stu.xjtu.edu.cn, hongxiaopeng@mail.xjtu.edu.cn, cxy19960919@stu.xjtu.edu.cn,
dsl972731417@stu.xjtu.edu.cn, xingxjtu@gmail.com, ygong@mail.xjtu.edu.cn

1. Experimental Details

1.1. Setups and Explanations of Neural Gas

Given an input feature vector $\mathbf{f} \in \mathcal{F} \subseteq \mathbb{R}^n$, the *neural gas* (NG) net matches it to the node j whose centroid $\mathbf{m}_j \in \mathbb{R}^n$ has the minimum distance $d(\mathbf{f}, \mathbf{m}_j)$ to $\mathbf{f}$. In our implementation, we simply use the *Euclidean distance* as the distance measure, which is written in the following L_2 norm form:

$$d(\mathbf{f}, \mathbf{m}_j) = \|\mathbf{f} - \mathbf{m}_j\|_2.$$

To train a NG net of N nodes on the base class data, we first extract the feature set $\mathcal{F}^{(1)}$ from $\mathcal{D}^{(1)}$. We initialize NG net by randomly selecting N feature vectors from $\mathcal{F}^{(1)}$ as the initial centroid vectors of N nodes. The number of nodes is determined according to diversity of $\mathcal{F}^{(1)}$. We ensure the number of NG nodes is larger than the number of classes, so that each class has at least one node for correspondence. Heuristically, we set $N = 400$ for all datasets.

Each node is adapted to $\mathbf{f}$ using Eq. (3) in the main paper. For node r_i whose rank is i, the contribution by the input vector $\mathbf{f}$ is measured using the decay function $e^{-i/\alpha}$. That is, if node r_i has a large rank $i \gg \alpha$, its distance with the input $d(\mathbf{f}, \mathbf{m}_{r_i})$ is very large, and we neglect the adaptation to speed up the training. For this purpose, we can set α to a smaller value (e.g. $\alpha = 10$ in our experiments.) This can reduce the time complexity of "sorting" from $O(N \log_2 N)$ to $O(\log_2 N)$.

The topology-preserving mechanism is achieved by the *competitive Hebbian learning*, where a topological connection between node i and j is established and maintained, if the two nodes are always simultaneously response to the input (i.e., the nearest and second nearest to the input). The "age" of the connection a_{ij} is used to record how long the two nodes have not been activated simultaneously. If $a_{ij} > T$, the connection is removed. We set $T = 200$ for training on the base class data according to [8]. Noting that if the number of training iterations is smaller, the value of T should be decreased to a smaller value, correspondingly.

*Corresponding author

1.2. The Structure of Baseline CNNs

The QuickNet model used in the experiments is originally defined in the Caffe package [3]. Table 4 shows the QuickNet structure in detail.

Table 4. The structure of QuickNet for evaluating on the CIFAR100 and miniImageNet dataset.

Name	layer type	filters	filter size	stride	pad
conv1	conv	32	5	1	2
pool1	max pool	-	3	2	0
relu1	relu	-	-	-	-
conv2	conv	32	5	1	2
relu2	relu	-	-	-	-
pool2	ave pool	-	3	2	0
conv3	conv	64	5	1	2
relu3	relu	-	-	-	-
pool3	ave pool	-	3	2	0
fc1	fc	64	-	-	-
fc2	fc	100	-	-	-

1.3. Comparative Results

Tables 5-8 report the test accuracy of all methods at different sessions. We run each method for 10 times and report the mean and standard deviation of the accuracy. The four tables correspond to the four subfigures of Figure 4 in the main paper.

Figure 6 compares the confusion matrix of the classification results at the last session, produced by Ft-CNN, EEIL* [1], NCM* [2] and our TOPIC. The naïve finetuning approach tends to misclassify all past classes (i.e., 0-94) to the newly learned classes (i.e., 95-99), indicating *catastrophic forgetting*. EEIL* and NCM* can alleviate forgetting to some extent, while still tend to misclassify old class test samples as new classes due to overfitting. Our method,

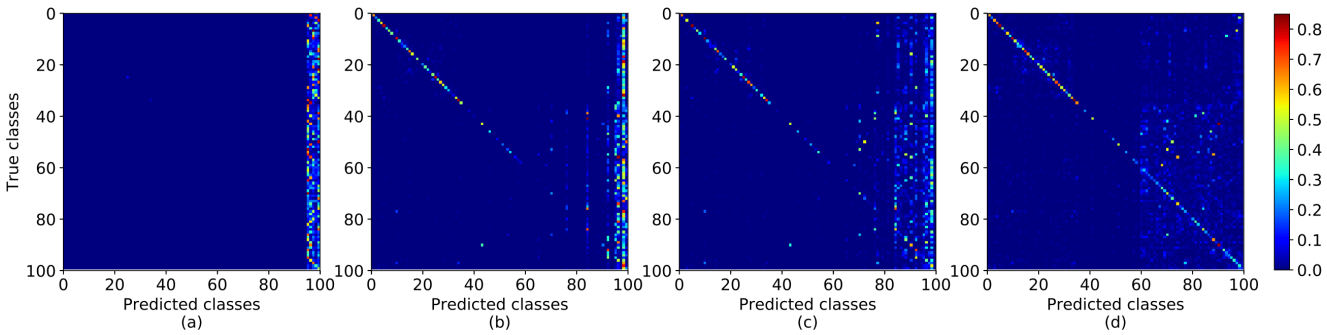


Figure 6. Comparison of the confusion matrices produced by (a) Ft-CNN, (b) EEIL*, (c) NCM*, and (d) our TOPIC on miniImageNet with ResNet18.

named "TOPIC", produces a much better confusion matrix, where the activations are mainly distributed at the diagonal line, indicating higher recognition performance over all encounter class. It demonstrate the effectiveness of solving FSCIL by avoiding both "forgetting old" and "overfitting new".

1.4. Acquisition-Memory Curve

Current researches typically use the "accuracy" metric to measure the performance of the models [4, 7]. To get a better trade-off between the test accuracy of the old and new classes, it is natural to tune the hyper-parameters or use training tricks, such as early stopping and lower learning rate [6, 5]. Therefore, simply reporting test accuracy may not fully demonstrate the models' effectiveness. For more comprehensive measurement, we develop the *acquisition-memory* (AM) curve, which measures the models' abilities of both remembering old class knowledge (*memory*) and acquiring new class knowledge (*acquisition*). First, we record the test accuracy of the intermediate models (i.e. the snapshots during training) on both old and new classes. Then we treat the accuracy of the old and new classes as the vertical and horizontal axes, and draw a curve using the records. We can further compute the F-score to determine the optimal trade-off point to get the best accuracy.

We compute AM curves at each training session for all methods. Figure 7 shows the AM curves computed with ResNet18 on miniImageNet. The horizontal "Acquisition Acc." and vertical "Memory Acc." axes stand for the test accuracy on new and old classes, respectively. From the AM curves, we can observed an amplified differences of the curves with the learning proceeds. We can observe a sharp drop of the "Ft-CNN" curve once finetuning on new class data, indicating *catastrophic forgetting* occurs. Other methods' curves fall much slower, thanks to the techniques for mitigating forgetting. After several training sessions, the differences of the curves are amplified, due to the cumulative effect of forgetting. Compared with other methods, our TOPIC ("AL-MML" setting) has the best "Memory Acc." after learning long sequence of sessions, indicating stronger ability of mitigating forgetting when learning new.

We can further find the trade-off point with considerable performance on both old and new classes, and use the corresponding model (snapshot) for learning subsequent sessions. We compute the F_β-score for each points on AM curve, and pick the one with the maximum score as the trade-off point:

$$F_\beta = (1+\beta^2)\frac{M \cdot A}{\beta^2 M + A}.$$

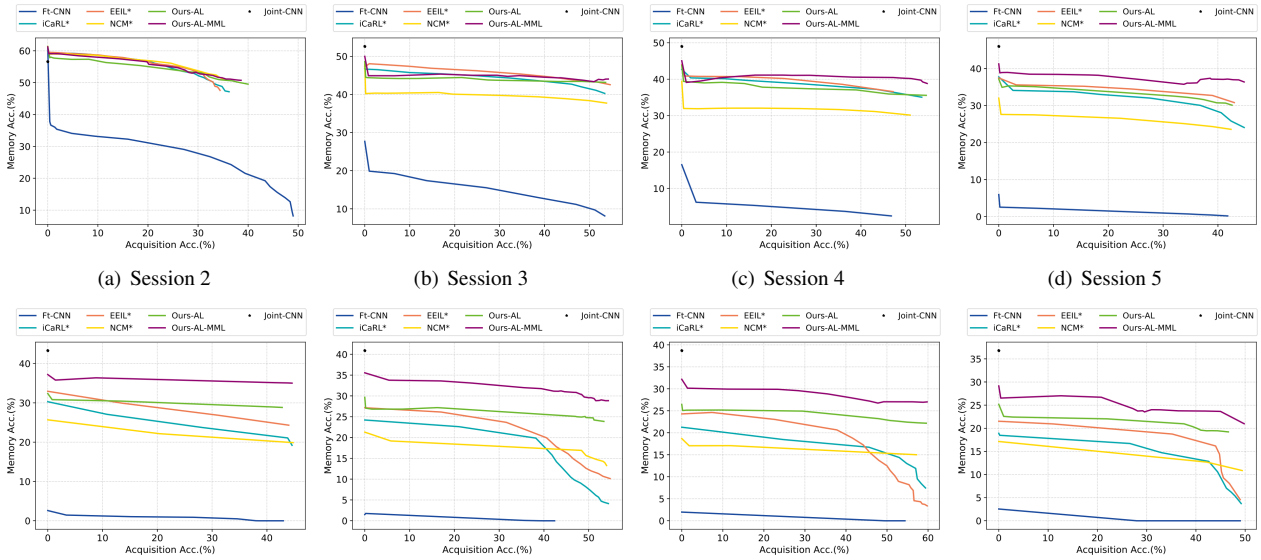
We can set β to a smaller value to remember more knowledge of old classes, or a larger value for enhancing the learning of new classes. In our comparison experiments, we set $\beta = 0.5$ when finding trade-off points for all methods.

1.5. Ablation Study

Table 10 reports the test accuracy achieved by different *loss terms* at different sessions, with ResNet18 on miniImageNet dataset, which corresponds to Table 2 in the main paper.

Tables 11 and 12 report the test accuracy of different methods under the *5-way 5-shot* and *5-way full-shot* setting, which correspond to the two subfigures of Figure 5 in the main paper.

Table 13 shows detailed comparative results between NG node and *exemplar* based method. For the exemplar based method, we randomly select exemplars from old class training samples, and extract their feature vectors as the representatives of feature space. When performing AL, we try to fix the feature vectors of the exemplars. While for MML, we pull the input feature vector to its nearest exemplar vector with the same label, while pulling exemplars of different labels away from each other. From Table 13, we can observe that the our NG based approach is more effective when the memory for storing the centroids of old data is small. When the memory grows larger and larger, the randomly sampled exemplars become more representative of the old class data, with a better performance closer to NG.



(a) Session 2 (b) Session 3 (c) Session 4 (d) Session 5

(e) Session 6 (f) Session 7 (g) Session 8 (h) Session 9

Figure 7. Comparison of the *acquisition-memory* (AM) curves at sessions 2-9, computed by ResNet18 on miniImageNet.

Table 5. Comparison results on CIFAR100 with QuickNet.

Method	sessions								
	1	2	3	4	5	6	7	8	9
Ft-CNN	57.78±0.40	13.09±0.20	6.40±0.21	3.00±0.27	2.24±0.18	2.86±0.19	1.46±0.27	2.47±0.19	2.00±0.22
Joint-CNN	57.78±0.40	53.30±0.03	49.50±0.03	46.20±0.03	43.80±0.04	41.20±0.03	39.10±0.02	37.80±0.03	35.90±0.03
iCaRL*	57.78±0.40	46.31±1.33	33.79±1.40	28.59±1.01	24.98±0.91	21.33±0.79	19.07±0.73	17.05±0.80	16.25±1.06
EEIL*	57.78±0.40	41.32±1.23	35.19±1.32	29.95±0.93	25.65±0.96	23.20±0.70	22.19±0.54	20.61±0.87	18.53±1.04
NCM*	57.78±0.40	48.91±1.31	41.91±1.57	38.05±1.02	30.61±0.90	26.68±0.67	24.79±0.68	22.15±0.71	19.50±1.18
Ours-AL	**57.78±0.40**	**49.52±1.25**	**44.32±1.45**	**39.59±1.22**	**33.72±1.14**	**30.65±0.82**	**27.36±0.60**	**25.06±0.57**	**23.12±1.25**
Ours-AL-MML	**57.58±0.40**	**49.49±1.10**	**44.12±1.37**	**39.82±0.85**	**35.07±0.93**	**31.42±0.61**	**27.82±0.41**	**25.47±0.65**	**24.17±1.17**

Table 6. Comparison results on CIFAR100 with ResNet18.

Method	sessions								
	1	2	3	4	5	6	7	8	9
Ft-CNN	64.10±0.40	36.91±0.20	15.37±0.38	9.80±0.25	6.67±0.13	3.80±0.19	3.70±0.17	3.14±0.23	2.65±0.11
Joint-CNN	64.10±0.40	59.30±0.27	54.90±0.08	51.20±0.03	48.10±0.02	45.80±0.03	42.80±0.02	40.90±0.02	38.90±0.03
iCaRL*	64.10±0.40	53.28±0.57	41.69±1.38	34.13±0.91	27.93±0.79	25.06±0.50	20.41±0.85	15.48±0.95	13.73±0.85
EEIL*	64.10±0.40	53.11±0.51	43.71±1.20	35.15±0.81	28.96±0.83	24.98±0.59	21.01±0.59	17.26±0.80	15.85±0.64
NCM*	64.10±0.40	53.05±0.55	43.96±1.56	36.97±0.83	31.61±0.60	26.73±0.65	21.23±0.53	16.78±1.00	13.54±0.68
Ours-AL	**64.10±0.40**	**56.03±0.61**	**47.89±1.34**	**42.99±1.03**	**38.02±0.62**	**34.60±0.68**	**31.67±0.56**	**28.35±0.61**	**25.86±0.91**
Ours-AL-MML	**64.10±0.40**	**55.88±0.42**	**47.07±1.18**	**45.16±0.82**	**40.11±0.79**	**36.38±0.50**	**33.96±0.69**	**31.55±0.54**	**29.37±0.80**

1.6. Hyper-parameter Setting

In the main paper, we set hyper-parameters $\lambda_1 = 0.5$, $\lambda_2 = 0.002$ and $\gamma = 1$ for balancing the strength of different loss terms. These hyper-parameters are tuned using the grid search technique. Concretely, at the initial experiments, we construct a temporary validation set by randomly selecting 100 samples from the 500 training samples for each base class, and select 5 training samples for each new class. We train the networks with different settings of these hyper-parameters. For grid search,

Table 7. Comparison results on miniImageNet with QuickNet.

Method	sessions								
	1	2	3	4	5	6	7	8	9
Ft-CNN	50.71±0.30	11.38±0.22	2.27±0.21	2.56±0.23	1.57±0.15	2.12±0.16	2.24±0.21	2.67±0.16	1.89±0.17
Joint-CNN	50.71±0.30	46.80±0.03	43.50±0.03	40.60±0.02	38.00±0.03	35.80±0.03	33.80±0.04	32.00±0.03	30.40±0.03
iCaRL*	50.71±0.30	37.55±1.26	31.65±1.23	26.49±1.15	23.33±1.26	20.75±1.15	17.08±1.07	14.69±0.93	11.05±1.12
EEIL*	50.71±0.30	39.20±1.22	33.55±1.17	29.84±0.93	26.47±1.02	22.41±0.87	18.79±0.98	16.74±0.94	13.59±1.02
NCM*	50.71±0.30	36.49±1.13	30.44±1.36	25.40±1.17	22.08±1.11	19.68±0.92	15.95±0.97	13.09±1.02	10.84±1.14
Ours-AL	**50.71±0.30**	**37.49±1.24**	**32.32±1.54**	**28.02±0.93**	**24.90±0.86**	**22.63±0.69**	**19.75±0.85**	**17.75±0.71**	**14.50±1.13**
Ours-AL-MML	**50.71±0.30**	**38.55±1.03**	**34.35±1.09**	**30.66±0.83**	**27.81±0.85**	**24.94±0.60**	**22.22±0.68**	**19.97±0.85**	**18.36±1.03**

Table 8. Comparison results on miniImageNet with ResNet18.

Method	sessions								
	1	2	3	4	5	6	7	8	9
Ft-CNN	61.31±0.30	27.22±0.26	16.37±0.37	6.08±0.22	2.54±0.24	1.56±0.12	1.93±0.17	2.60±0.25	1.40±0.13
Joint-CNN	61.31±0.30	56.60±0.16	52.60±0.05	49.00±0.03	46.00±0.03	43.30±0.04	40.90±0.03	38.70±0.02	36.80±0.03
iCaRL*	61.31±0.30	46.32±0.85	42.94±1.57	37.63±1.44	30.49±1.36	24.00±1.53	20.89±1.66	18.80±1.69	17.21±1.49
EEIL*	61.31±0.30	46.58±0.86	44.00±1.53	37.29±1.35	33.14±1.20	27.12±1.47	24.10±1.54	21.57±1.75	19.58±1.26
NCM*	61.31±0.30	47.80±0.77	39.31±1.58	31.91±1.37	25.68±1.28	21.35±1.43	18.67±1.50	17.24±1.52	14.17±1.37
Ours-AL	**61.31±0.30**	**48.58±0.64**	**43.77±1.15**	**37.19±1.12**	**32.38±0.90**	**29.67±1.36**	**26.44±1.38**	**25.18±1.34**	**21.80±1.14**
Ours-AL-MML	**61.31±0.30**	**50.09±0.70**	**45.17±1.04**	**41.16±1.02**	**37.48±0.83**	**35.52±1.17**	**32.19±1.10**	**29.46±1.13**	**24.42±0.96**

Table 9. Comparison results on CUB200 with ResNet18. Nothing that the comparative methods with their original learning rate settings have much worse test accuracy on CUB200. We carefully tune their learning rates and greatly boost their original accuracy. In the table below, we use * to denote the settings with the improved accuracy.

Method	sessions										
	1	2	3	4	5	6	7	8	9	10	11
Ft-CNN	68.68±0.90	43.70±0.83	25.05±0.94	17.72±0.91	18.08±1.11	16.95±1.19	15.10±0.99	10.60±0.87	8.93±0.90	8.93±1.15	8.47±0.89
Ft-CNN*	68.68±0.90	44.81±0.94	32.26±0.96	25.83±0.84	25.62±0.93	25.22±1.13	20.84±1.05	16.77±1.01	18.82±0.96	18.25±0.93	17.18±0.95
Joint-CNN	68.68±0.90	62.43±0.83	57.23±0.94	52.80±0.99	49.50±1.04	46.10±1.02	42.80±0.96	40.10±0.95	38.70±0.92	37.10±1.10	35.60±0.85
iCaRL	68.68±0.90	60.50±0.94	46.19±0.89	31.87±0.83	29.07±0.92	21.86±0.94	21.22±0.97	19.15±0.96	16.50±1.07	14.46±1.12	14.14±1.06
iCaRL*	68.68±0.90	52.65±0.93	48.61±0.91	44.16±0.89	36.62±0.93	29.52±0.97	27.83±0.86	26.26±0.88	24.01±0.85	23.89±0.92	21.16±1.00
EEIL	68.68±0.90	57.64±0.98	42.91±0.90	28.16±0.92	27.05±0.94	25.52±0.98	25.08±1.06	22.06±1.02	19.93±0.89	19.74±0.98	19.61±1.10
EEIL*	68.68±0.90	53.63±1.09	47.91±0.96	44.20±0.98	36.30±0.89	27.46±0.97	25.93±0.94	24.70±1.04	23.95±1.03	24.13±0.89	22.11±0.91
NCM	68.68±0.90	62.55±0.89	50.33±0.85	45.07±0.96	38.25±0.87	32.58±0.90	28.71±0.98	26.28±0.89	23.80±0.96	19.91±1.09	17.82±0.93
NCM*	68.68±0.90	57.12±0.93	44.21±0.91	28.78±0.94	26.71±0.95	25.66±0.98	24.62±0.89	21.52±0.88	20.12±0.94	20.06±0.96	19.87±1.05
Ours-AL	**68.68±0.90**	**61.01±0.92**	**55.35±0.94**	**50.01±0.89**	**42.42±0.92**	**39.07±0.91**	**35.47±0.90**	**32.87±0.88**	**30.04±0.93**	**25.91±0.91**	**24.85±0.87**
Ours-AL-MML	**68.68±0.90**	**62.49±0.91**	**54.81±0.92**	**49.99±0.87**	**45.25±0.85**	**41.40±0.88**	**38.35±0.96**	**35.36±0.87**	**32.22±0.90**	**28.31±0.92**	**26.28±0.91**

we first try different hyper-parameters in a wide range $\{1e^{-4}, 0.001, 0.01, 0.1, 1, 10\}$ to determine their scales. Then we search these hyper-parameters in a smaller range (e.g. $[0.1, 1]$ for λ_1.) We observed that the validation accuracy has very small fluctuations (less than 0.5%) when changing the hyper-parameters in a smaller range. Thus, we set the hyper-parameters to those values with reasonable validation accuracy.

References

[1] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248, 2018.

[2] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[3] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama,

Table 10. Comparison results of different loss terms on miniImageNet with ResNet18.

Method	sessions								
	1	2	3	4	5	6	7	8	9
DL	61.31±0.30	46.85±0.45	42.34±1.29	36.56±1.22	30.63±1.08	27.64±1.42	24.61±1.40	22.06±1.21	18.69±1.28
DL-MML	61.31±0.30	48.14±0.85	42.83±1.13	38.35±1.32	32.76±1.14	30.02±0.94	27.70±1.14	25.43±1.43	20.55±1.36
AL w/o. $\mathbf{\Lambda}$	61.31±0.30	48.55±0.71	42.73±1.21	36.73±1.21	32.59±1.02	28.40±1.48	25.23±1.31	23.69±1.39	21.36±1.20
AL	61.31±0.30	48.58±0.64	43.77±1.15	37.19±1.12	32.38±0.90	29.67±1.36	26.44±1.38	25.18±1.34	21.80±1.14
AL-Min*	61.31±0.30	50.60±0.79	45.14±1.13	41.03±1.31	35.69±0.91	33.64±1.30	30.11±1.34	27.79±1.40	24.18±1.16
AL-Max*	61.31±0.30	48.49±0.80	43.03±1.23	38.53±1.33	34.24±1.15	31.79±1.41	28.96±1.40	26.09±1.47	23.80±1.25
AL-MML	61.31±0.30	50.09±0.70	45.17±1.04	41.16±1.02	37.48±0.83	35.52±1.17	32.19±1.10	29.46±1.13	24.42±0.96
AL-MML-DL	61.31±0.30	50.00±0.65	44.23±1.15	39.85±1.16	36.02±0.93	32.95±1.05	29.78±1.20	27.17±1.39	23.49±1.18

Table 11. Comparison results of the 5-way 10-shot setting on miniImageNet with ResNet18.

Method	sessions								
	1	2	3	4	5	6	7	8	9
Ft-CNN	61.31±0.30	25.20±0.23	14.30±0.33	6.30±0.24	2.70±0.16	1.90±0.17	2.50±0.23	2.80±0.15	2.10±0.19
Joint-CNN	61.31±0.30	56.60±0.13	52.60±0.07	49.00±0.03	46.00±0.03	43.30±0.03	40.90±0.02	38.70±0.02	36.80±0.03
iCaRL*	61.31±0.30	45.18±0.80	42.53±1.48	35.96±1.46	27.99±1.35	21.82±1.51	17.91±1.21	16.13±1.26	15.36±1.47
EEIL*	61.31±0.30	45.97±0.83	42.56±1.51	37.75±1.31	31.16±1.27	27.08±1.48	24.42±1.12	22.42±1.42	19.82±0.98
NCM*	61.31±0.30	48.45±0.75	40.07±1.34	33.29±1.30	26.20±1.21	21.29±1.42	18.13±1.37	15.81±1.25	13.85±1.37
Ours-AL	**61.31±0.30**	**50.06±0.65**	**45.47±1.11**	**39.84±1.16**	**33.96±0.93**	**31.39±1.33**	**28.19±1.30**	**26.24±1.37**	**22.35±0.98**
Ours-AL-MML	**61.31±0.30**	**51.20±0.72**	**46.94±1.03**	**41.89±1.06**	**39.05±0.80**	**36.78±1.17**	**34.19±1.35**	**31.86±1.12**	**28.03±0.99**

Table 12. Comparison results of the 5-way full-shot setting on miniImageNet with ResNet18.

Method	sessions								
	1	2	3	4	5	6	7	8	9
Ft-CNN	61.31±0.30	43.43±0.27	37.57±0.32	34.64±0.28	33.24±0.39	29.80±0.28	26.74±0.25	26.53±0.19	26.13±0.24
Joint-CNN	61.31±0.30	59.31±0.11	57.84±0.07	56.32±0.03	55.85±0.03	54.23±0.03	53.95±0.02	53.71±0.03	53.54±0.03
iCaRL*	61.31±0.30	57.25±0.86	53.10±1.01	50.03±0.91	48.33±0.83	39.12±0.82	38.29±1.24	38.17±1.16	37.97±0.96
EEIL*	61.31±0.30	57.42±0.97	53.84±0.97	51.25±1.05	49.56±0.81	40.07±0.94	38.76±0.93	38.41±0.97	38.70±1.09
NCM*	61.31±0.30	57.12±1.13	54.23±0.92	52.17±1.13	50.52±0.91	45.82±1.02	44.44±0.91	44.01±1.15	43.71±1.06
Ours-AL	**61.31±0.30**	**57.63±0.83**	**55.31±0.92**	**53.31±1.05**	**51.48±0.94**	**45.00±0.83**	**44.38±1.12**	**44.27±1.05**	**43.54±0.91**
Ours-AL-MML	**61.31±0.30**	**56.75±0.92**	**54.33±1.31**	**52.64±1.11**	**51.21±1.47**	**46.25±1.18**	**45.43±0.92**	**44.78±1.24**	**44.95±1.09**

Table 13. Comparison of the final test accuracy achieved by "exemplars" and NG nodes under different memory size. Experiments are performed with ResNet18 on CIFAR100.

Memory	50	100	200	400	800	1600
Exemplars	19.21±0.95	22.32±0.93	26.94±0.90	28.25±0.85	28.69±0.80	28.89±0.79
NG nodes	22.37±0.98	25.72±0.91	28.56±0.85	29.37±0.80	29.54±0.84	29.35±0.74

and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014.

[4] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[5] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgletting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4652–4662, 2017.

[6] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

[7] B Pfülb and A Gepperth. A comprehensive, application-oriented study of catastrophic forgetting in dnns. 2018.

[8] Martinetz Thomas and Schulten Klaus. A ”neural-gas” network learns topologies. *Artificial Neural Networks*, 1991.