Exemplar-Free Continual Transformer with Convolutions (Supplementary Material)

Anurag Roy[1] Vinay K. Verma[2,†] Sravan Voonna[1] Kripabandhu Ghosh[3] Saptarshi Ghosh[1] Abir Das[1]
[1]IIT Kharagpur, [2]IML Amazon India, [3]IISER Kolkata
{anurag_roy@,vg.sravan@,saptarshi@cse.,abir@cse.}iitkgp.ac.in, {vinayugc,kripa.ghosh}@gmail.com

This supplementary material contains the following.

- Section 1: Classification Accuracies on 5-Datasets over more baselines.
- Section 2 Classification Accuracies of ConTraCon with different task orders.
- Section 3 Upper Bounds on classification accuracies on various datasets.
- Section 4: Augmentations used for task prediction

1. Results on 5-Datasets

First proposed by Ebrahimi et. al. [4], 5-Datasets is composed of CIFAR-10 [7], MNIST [8], SVHN [9], Fashion MNIST [12] and notMNIST where classification on each of these datasets is a task. The variation / diversity in the dataset for each of the tasks in 5-Dataset sets it apart from the other benchmark datasets used in this paper. As the dataset is very recently proposed, some of the competitive continual learning approaches did not have chance to validate on this dataset. Hence, we ran a few recent approaches on this dataset and compared with ConTraCon. Specifically, we ran DER++ (NeurIPS2020) [2] and FDR (ICLR2019) [1] on this very challenging dataset.

Table 1 shows the results on this dataset. The newly added experimental results are highlighted in blue. While FDR [1] and DER++ [2] use ResNet18 as the backbone, GPM and EFT uses a variation of it to reduce the number of learnable parameters with an eye to avoid possible overfitting. It can be noted that ConTraCon significantly beats FDR and DER++ on this challenging and diverse continual learning dataset with almost 15% gain over the best of the two in (Task Incremental Learning) TIL and around 20% better in the (Class Incremental Learning) CIL setting. Additionally ConTraCon uses much less parameters (almost 33% less) compared to both FDR and DER++ showing the capability of our proposed task adaptable convolution to handle diverse tasks for continual learning in TIL as well as CIL settings.

[†]Work started before joining Amazon

2. Results with Different Task Orders

ConTraCon uses convolution-based task adaptation over the original backbone (in our case CCT [6]), pre-trained on the first or initial task. To show the robustness of our approach on the choice of the initial task, we chose to experiment with different initial tasks out of the tasks available for CIFAR-100/10.

For this purpose, we perform two variations of the experiment originally performed for the main paper (ref. Table 2 of the main paper) – (1) Train ConTraCon with *reversed* task order as followed in main paper so that the initial task there becomes the final task here and vice-versa, and (2) Train ConTraCon with a random task-order.

We observe that, with task-order reversed, the model achieves an average classification accuracy of 84.83% in the Task Incremental Learning (TIL) setup while the same accuracy for the original task-order followed in Table 2 of the main paper is 85.69%. With random task-order, ConTraCon achieves an average classification accuracy of 83.82%.

Note that, even with different task-orders, ConTraCon's performance is always better than that of the state-of-the-art approaches.

3. Upper Bounds on Classification Accuracies

To better understand the performance of ConTraCon, we calculate the upper bounds, i.e., the maximum achievable performance by the backbone architecture. Specifically, we train each task on a separate backbone architecture, thereby having a per-task parameter increase of 100%. Using this setup, we calculate the upper-bounds for all the datasets and task-splits. Table 2 shows a comparative study of the performance variation between the upper-bound and ConTraCon. On average, we observe that ConTraCon's performance is $\sim 1-4\%$ below the corresponding upper-bounds while requiring only 0.7% of the number of parameters required per task for the upper-bound performances.

Model	Approach	Backbone	# Params	5-Datasets	
				TIL	CIL
FDR (ICLR2019) [1]	Rehearsal	ResNet18	11.2 M	72.45	38.21
DER++ (NeurIPS2020) [2]	Rehearsal	ResNet18	11.2 M	80.45	45.03
EFT (CVPR2021) [11]	Dynamic Arch	ResNet18	4.9 M (32k)	94.75	52.04
GPM (ICLR2021) [10]	Regularization	ResNet18	**1.2 M**	90.60	–
Dytox (CVPR2022) [3]	Rehearsal	Transformer	10.7 M	77.12	**67.13**
ConTraCon (proposed)	Dynamic Arch	Transformer	3.9 M (28k)	**95.10**	65.21

Table 1: Classification accuracy on 5-Datasets. Mentioned in brackets, are the number of additional parameters required to learn each new task for dynamic architecture-based approaches like EFT and ConTraCon. EFT and GPM use a reduced version of the resnet18 architecture as their backbone. New baseline approaches are highlighted in blue. The rehearsal based approaches use a buffer of size 500

Dataset	Upper Bound		ConTraCon	
	TIL	# Params	TIL	# Params
CIFAR-100/5	85	3.1 M	79.37	26k
CIFAR-100/10	89.30	3.1 M	85.96	26k
CIFAR-100/20	93.16	3.1 M	88.94	26k
ImageNet-100/10	80.67	3.6 M	76.78	28k
TinyImageNet-200/10	70.66	3.6 M	62.76	28k
5-Datasets	96.42	3.9 M	95.10	28k

Table 2: Comparison of the performance of ConTraCon (proposed model) with the upper-bound (the maximum achievable performance) calculated by training the backbone separately for each task. The values under TIL denote the average classification accuracy in the Task Incremental Learning setup. # Params denotes the number of parameters required to learn each new incoming task.

4. Augmentations Used

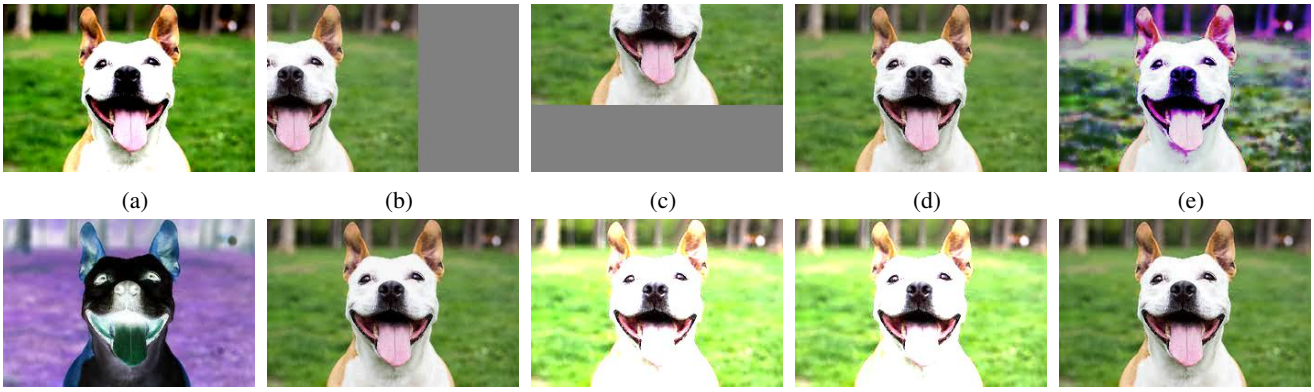
Entropy-based task prediction performs poorly due to cross-entropy training loss function. This results in high confidence (i.e., low entropy) predictions even for out-of-distribution inputs [5]. Hence, to overcome this, we calculate the entropy of the average predictions of different augmentations of the input image (as discussed in Section 3.4 of the main paper). Specifically, for an input image during test time (shown in Fig. 1), we augment the test image in 10 different ways. After that we pass these through various task specific models and calculate the entropy of each of the predicted probability distributions to ultimately get the task ids. In this supplementary material, we provide the details of the augmentations we used for this purpose. The augmented versions of the image in Fig. 1 are shown in Fig. 2a–Fig. 2j. The augmentations we use are as follows:

Figure 1: Unaugmented image

- Increase Contrast: This augmentation increases the color content of an image. We increase the contrast by magnitude 1.6 as shown in Fig. 2a.
- Translate along X-axis: As shown in Fig. 2b, we translate the image along x-axis by a magnitude of 0.4.
- Translate along Y-axis: We translate the image along y-axis by a magnitude of 0.4 as shown in Fig. 2c.
- Increase sharpness: This augmentation helps in increasing the detail of the image by making the edges clearer. As shown in Fig. 2d and Fig. 2j, we increase the sharpness of the input image by a magnitude of 1.3.
- Equalize Image: This augmentation equalizes the histogram of the given image as shown in Fig. 2e.
- Invert Image color: This augmentation involves reversing the color of the image as shown in Fig. 2f
- Posterize : An image is posterized by reducing the number of bits for each color channel. As shown in Fig. 2g we posterize by magnitude 5.
- Increase Brightness: As shown in Fig. 2h and Fig. 2i, we increase the brightness by magnitudes 1.9 and 1.7 respectively.

References

[1] Ari Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. In *Inter-*



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 2: Augmentations used for Task-id prediction (a) increased contrast (b) translation along x axis (c) Translation along y-axis (d) increased sharpness (e) Equalized image (f) Inverted image (g) posterized image (h) increased brightness (i) increased brightness (j) increased sharpness.

national Conference on Learning Representations, 2019. 1, 2

[2] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020. 1, 2

[3] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[4] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *European Conference on Computer Vision*, pages 386–402. Springer, 2020. 1

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 2

[6] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021. 1

[7] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009. 1

[8] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 1

[9] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 1

[10] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. 2

[11] Vinay Kumar Verma, Kevin J Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin. Efficient feature transformations for discriminative and generative continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13865–13875, 2021. 2

[12] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv*, 2017. 1