Supplementary Material
IB-DRR - Incremental Learning with Information-Back Discrete Representation Replay

Jian Jiang Edoardo Cetin Oya Celiktutan
Centre for Robotics Research, Department of Engineering, King's College London, UK
{jian.jiang, edoardo.cetin, oya.celiktutan}@kcl.ac.uk

1. Qualitative Results of VQ-VAE in CIFAR-100 and ImageNet

In this section, we show the qualitative results of reconstructed images by the hierarchical VQ-VAE pretrained given half classes of CIFAR-100 (the one used in our main paper). Fig. 1 shows the model architecture of the hierarchical VQ-VAE.

Example reconstructed images from CIFAR-100 are provided in Fig. 2-(a). And those from ImageNet with resolution 32, 64, 224 are in Fig. 2-(b), Fig. 2-(c)), Fig. 2-(d)) respectively. Looking at the qualitative results presented in Fig. 2, we argue that our VQ-VAE is capable of reconstructing varying sizes of images adequately, despite being applied to a dataset (ImageNet) that is different from the training dataset (CIFAR-100). To answer why the VQ-VAE has such a good generalization ability, intuitively, a VQ-VAE is just a pixel reconstructor and the data (50 classes from CIFAR-100) used for pretraining contains diverse images that provides a considerably good pixel distribution for learning.

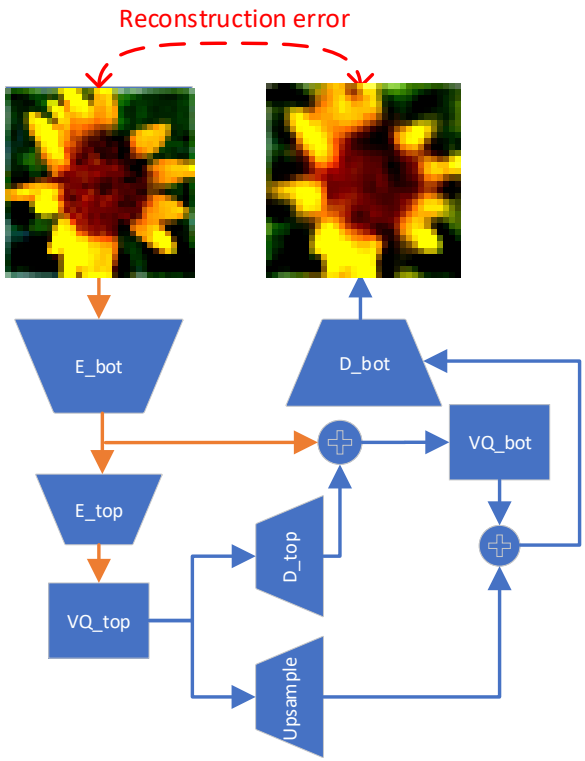


Figure 1. The hierarchical VQ-VAE architecture for incremental learning. E_, D_, VQ_ represent Encoder, Decoder, and Vector quantisation respectively.

2. Compression for Codes via Bit-Swap

Because we use the codebook size of 512, the discrete value of a code ranges from $[0, 511]$. Theoretically, a code can be saved using 9 bits ($2^9 = 512$), but in practice, a system saves codes data in the least unit of byte, so it causes 2 bytes, i.e., 16 bits to save an uncompressed code. According to entropy coding scheme, the entropy of a certain data distribution $p(\mathbf{x})$, defined by $H[\mathbf{x}] \triangleq \mathbb{E}[-\log p(\mathbf{x})]$, is the lower bound on how many average bits (called 'message length') a lossless compression method can achieve to encode data points coming from this distribution. Table 1 shows the entropy of top-level and bottom-level codes obtained by compressing a certain dataset via the VQ-VAE (the one used in our main paper). For example, the entropy of top codes and bottom codes of samples of 100 classes from CIFAR-100 are 8.6032 bits and 8.6404 bits respectively. And Bit-Swap models trained by 3000 epochs for each of them could achieve 8.8376 or 8.8450 average bits cost for compressing them respectively. We originally expected the more codes data (from 50 classes to 100 classes) can reveal a better distribution with less entropy in terms of a certain dataset. In contrast to our expectation, the entropy of codes distribution changed a little (less than 0.1) during the incremental setting on CIFAR-100, i.e., more new codes did not decrease the average overall entropy. Moreover, if we mixed the top and bottom codes, the entropy increased.

Dataset	top	bottom	top & bottom
CIFAR-100	8.6032	8.6404	8.7482
SubImg 224x224	6.7936	7.1016	7.4683

Table 1. Entropy of codes distributions for different dataset. 'SubImg' refers to a subset (contains 100 classes) of ImageNet

We also ran preliminary results on Subset-ImageNet that contained random 100 classes ($128,856$ training samples) from ImageNet-1K with an image resolution of 224×224. We obtained codes of ImageNet using the VQ-VAE pre-trained on CIFAR-100 as discussed in the main paper. For all top-level codes of Sub-ImageNet, after trained for 2100 epochs, a Bit-Swap cost 7.86 average bits to encode one top-level code. As for bot-level codes, another Bit-Swap trained for 1100 epochs cost 7.83 average bits to encode one bottom-level code. Our preliminary results showed that more iterations (epochs) could further improve the compression performance.

3. Implementation details of compared methods

We compare our methods with the following baseline methods and state-of-the-art methods with their original implementation settings on CIFAR-100:

- ***Upper Bound (UB)*** saves all (50,000) exemplars and trains from scratch a Resnet-18 for each new training phase. For training UB, we use the same hyperparameters as our DRR.
- ***GFR*** [6] requires no exemplar to be stored. It has a two-stage training: The first stage jointly trains a feature extractor and a classifier for 200 epochs, where they use a Resnet-18 and treat the former layers as the feature extractor. They save a copy of the old model after a training phase and initialize the new model with the weights of the old ones. The second stage trains a GAN used a frozen feature extractor and the classifier for 500 epochs.
- ***iCaRL*** [9] saves 20 exemplar per class and $2,000$ in total. It uses a rank function to select samples to be reserved, and a Resnet-32 is trained or fine-tuned for 160 epochs.
- ***LUCIR*** [4] saves 20 exemplar per class using class-rebalance strategies. It also uses the rank function introduced in iCaRL. We use the setup of LUCIR as presented in [7] where a Resnet-32 is used and is fine-tuned for 160 epochs in new training phases.
- ***Mnemonics*** [7] is built upon LUCIR while it saves 20 trainable exemplars per class instead. Two-level training includes model-level training for a Resnet-32 and exemplar-level training for exemplars. The classifier and learnable exemplars are fine-tuned in the training phase.
- ***LWF*** [5] is one of the most representative non-replay-based methods and no exemplar is saved. It incorporates knowledge distillation technique [3] as an extra regularisation term to consolidate previous knowledge when learning new knowledge. We also use the setup of LWF as presented in [7] where a Resnet-32 is used and is fine-tuned for 160 epochs in new training phases.

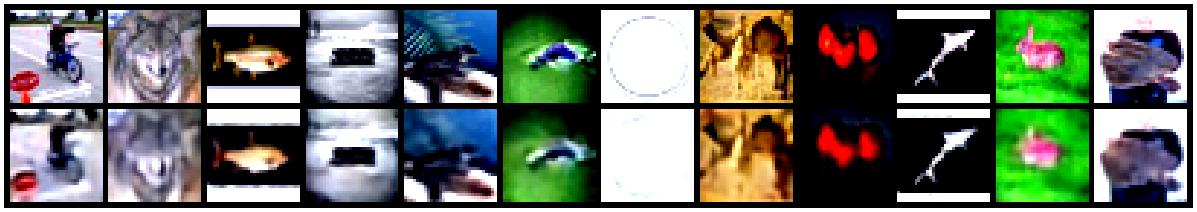
4. The Choice of Resnet for CIFAR-100

Residual Network (Resnet) was proposed by He *et al*. [2] in 2015. Resnet-18 was also proposed in [2], which had 18 layers with a hyperparameter 'in-planes' set to 64. The 'in-planes' was used to control the number of filters in layers. For example, with 'in-planes = 64', the layers in i_{th} ResBlock had $64 \times i$ filters. There were many variants of Resnet with a different number of layers and different 'in-planes'. In recent incremental learning works researchers prefer a Resnet-32 [7, 9, 8] with 'in-planes = 16' that had more layers but lower capacity than original Resnet-18 with 'in-planes = 64'. In our preliminary experiments, we found that Resnet-32 with 'in-planes = 16' resulted in underfitting of our models in some cases, e.g., if a strong Data Augmentation was applied. Our further experiments showed that if we increased the number of filters in Resnet-32, i.e., setting the hyper-parameter 'inplanes' from 16 to 64, our method performed even slightly better as compared to that used Resnet-18 with 'inplanes=64'. We conjectured that Resnet-32 with 'inplanes =16' was adequate only for replaying 20 (low resolutions) samples per class. However, for replaying more samples like ours and [10] or generated sample models (like GFR [6]), Resnet-18 or Resnet-32 with 'inplanes=64' was more suitable for CIFAR-100. Note that we only used simple data augmentation like 'horizontal flip' and 'random crop' for the experiments in the main paper, but our previous studies showed that DRR could benefit from a strong data augmentation strategy [1] as well as test-time augmentation strategy [11, 12]. We will further test IB-DRR with the above data augmentation strategies in the future.

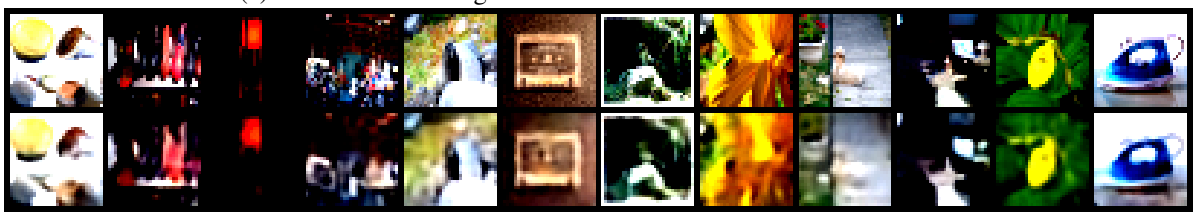
References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 2

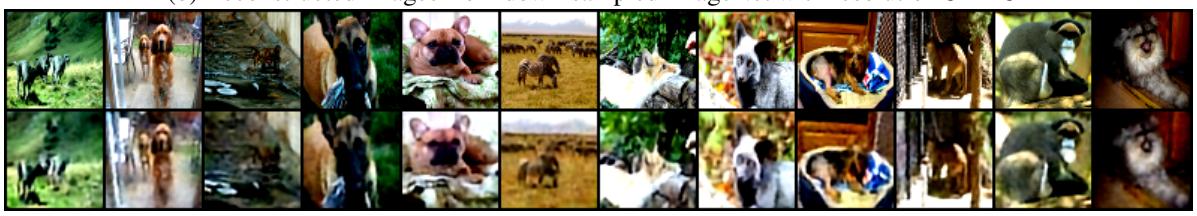
[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*



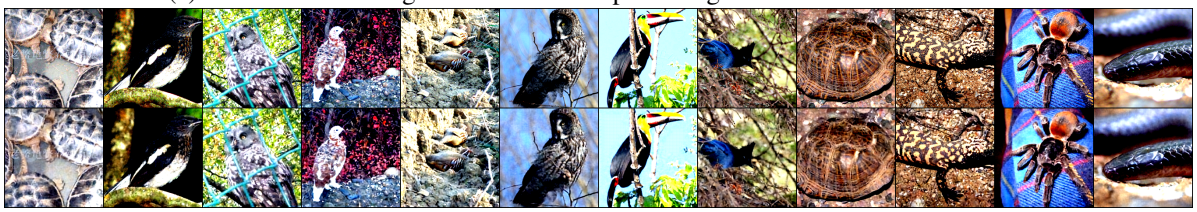
(a) Reconstructed images from CIFAR-100 with resolution 32×32



(b) Reconstructed images from down-sampled ImageNet with resolution 32×32



(c) Reconstructed images from down-sampled ImageNet with resolution 64×64



(d) Reconstructed images from original ImageNet with resolution 224×224

Figure 2. Reconstructed images at different resolutions using a VQ-VAE pre-trained with 25,000 images (32×32) from CIFAR-100.

ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 2

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. 2015. 2

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

[5] Zhizhong Li and Derek Hoiem. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, jun 2018. 2

[6] Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 226–227, 2020. 2

[7] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020. 2

[8] Dushyant Rao, Francesco Visin, Andrei A Rusu, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Continual unsupervised representation learning. *arXiv preprint arXiv:1910.14481*, 2019. 2

[9] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2

[10] Matthew Riemer, Tim Klinger, Djallel Bouneffouf, and Michele Franceschini. Scalable recollections for continual lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1352–1359, 2019. 2

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE con-*

ference on computer vision and pattern recognition, pages 2818–2826, 2016. 2