Exemplar-Supported Generative Reproduction for Class Incremental Learning Supplementary Material

Chen He[1,2]
chen.he@vipl.ict.ac.cn
Ruiping Wang[1,2]
wangruiping@ict.ac.cn
Shiguang Shan[1,2]
sgshan@ict.ac.cn
Xilin Chen[1,2]
xlchen@ict.ac.cn

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China

[2] University of Chinese Academy of Sciences, Beijing, 100049, China

In the supplementary material, we provide the implementation details (Section 4.1) and the preliminary results of the variants of our method that use a more "balanced" training set which performs better under the 10-class adding setting on CIFAR-100 (mentioned in Section 4.2).

1 Implementation Details

Exemplar-Supported Generative Reproduction (ESGR). For CIFAR-100 [2], we train a WGAN with gradient penalty (WGAN-GP) [1] for each class (under the one-class adding and the batch-of-class adding setting). Each WGAN-GP is trained for 10,000 iterations with a base Adam learning rate 0.001 and a batch size of 64. For ImageNet-Dogs, we add an auxiliary classifier [4] to WGAN-GP, making it an AC-WGAN-GP. Each generator is trained for 20,000 iterations with a base Adam learning rate 0.0002 and a batch size of 32. The generated images on CIFAR-100 and ImageNet-Dogs are shown in Figure 1 and Figure 3 respectively. All of the implementations of the generators are based on the codes of WGAN-GP [1]. According to the original implementations, the generator should be trained for 200,000 iterations, but the training loss and the inception scores won't change much and the generated images are good enough after 10,000 iterations on CIFAR-100 and 20,000 iterations on ImageNet-Dogs respectively. Thus, we take an early stop, which can save much training time for ESGR.

iCaRL [5]. For convenience's sake, we use the codes released on Github by the author and only change the network architectures to be the same as ours to make fair comparisons. For both LeNet and ResNet, we use the output of the layer before the final fully connected layer as the extracted features as iCaRL did.

Learning without Forgetting (LwF) [3]. Although LwF is not designed for this task, we still evaluate its performance. We simply refer to the implementation of *hybrid1* in the

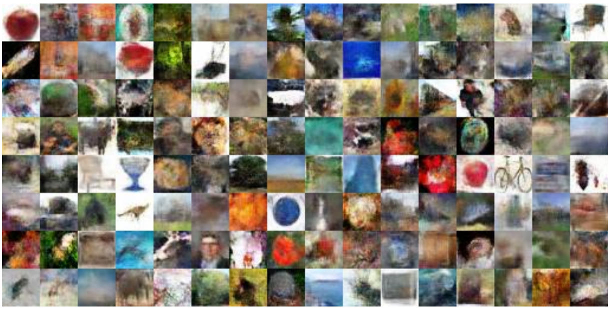

Figure 1: The generated images by ESGR on CIFAR-100.

Figure 2: The generated images by DGR on CIFAR-100 after the final training session.

codes released by iCaRL as the implementation of LwF (*hybrid1* is an implementation of LwF except that *hybrid1* uses sigmoid instead of softmax in the original paper). Supposing that we want to add N classes to an M-class classifier and make it an (M+N)-class classifier, the one-hot vectors for the old task and new task are set to be M-dimension and N-dimension respectively.

Deep generative replay (DGR) [6]. The author doesn't offer the codes nor conduct experiment on CIFAR-100 and ImageNet-Dogs, so we implement DGR ourselves. WGAN-GP is used as the generator, the network architecture of which is the same as that of ESGR. For CIFAR-100, we use a base Adam learning rate of 0.0001 and train 20,000 iterations for each training session. The generated images are shown in Figure 2. The ratio is set to be 0.8 for training the classifier since we find that 0.8 yields better results. For ImageNet-Dogs, we also train the generator for 20,000 iterations in each training session, but the generated images are getting unsatisfactory from the 8th training session (Figure 4). The reason might be that it is a difficult dataset and DGR cannot handle high-resolution pictures of so many classes. There is a possibility that careful tuning on the hyper-parameters may make DGR perform better, but it is a difficult task which is beyond the scope of this paper.

2 Results using a Balanced Version of ESGR

In Section 4.2, we mention that the less satisfactory result of our method under the 10-class adding setting on CIFAR-100 is because the training set is in essence "imbalanced" and we suggest that it can be easily solved by using generated data only for the new class for

Figure 3: The generated images by ESGR on ImageNet-Dogs. Although most of the generated images are distorted since it is a difficult dataset, for some of them we can still recognize what is in the picture.

Figure 4: The generated images by DGR on ImageNet-Dogs after the 8th training session. From the 8th training session, the training is less stable and the generated images become unsatisfactory.

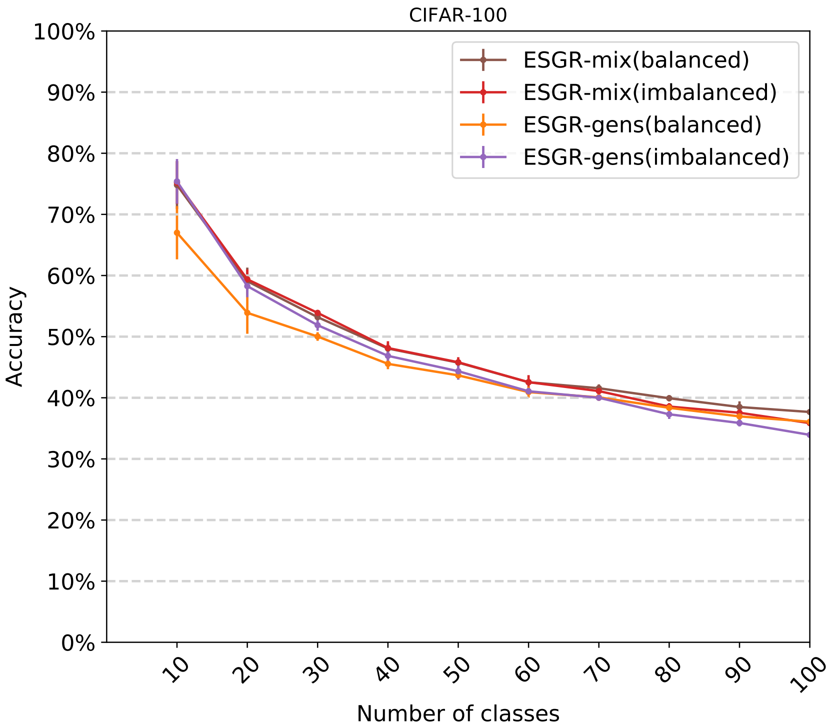

Figure 5: Accuracy curves of the original version of ESGR-*gens* and ESGR-*mix* and the "balanced" versions under the 10-class adding setting on CIFAR-100.

ESGR-*gens* and a proper combination of some randomly selected real data combined with generated data for ESGR-*mix* (please refer to the released codes for more details). Below are the results:

From Figure 5, we can see that the "balanced" version of ESGR-*gens* has a 1-2% improvement over the "imbalanced" one in the final evaluation. But when adding less than 70 seen classes, the original "imbalanced" version performs better. It is because there are more real data in the training set in the "imbalanced" version—an extreme case is when the algorithm sees the first 10 classes: for the "imbalanced version" the training set is composed of real samples only; for the "balanced version" the training set is composed of generated samples only. Apparently at this time the "imbalanced version" has better performance than the "balanced version". But as the number of seen classes increases, the negative influence of the "imbalance" increases, and the performance drops more quickly than the "balanced" version. Note that when adding more classes, let's say 1,000, the two curves will perform almost the same, since the portion of the real samples is too small to cause "imbalance". The "balanced" version of ESGR-*mix* shows similar phenomena and the reason is almost the same.

From Table 1, we can see that the forget rate drops quite a lot by adapting the original version to a "balanced" one. It is even lower than that of iCaRL, indicating that the imbalance

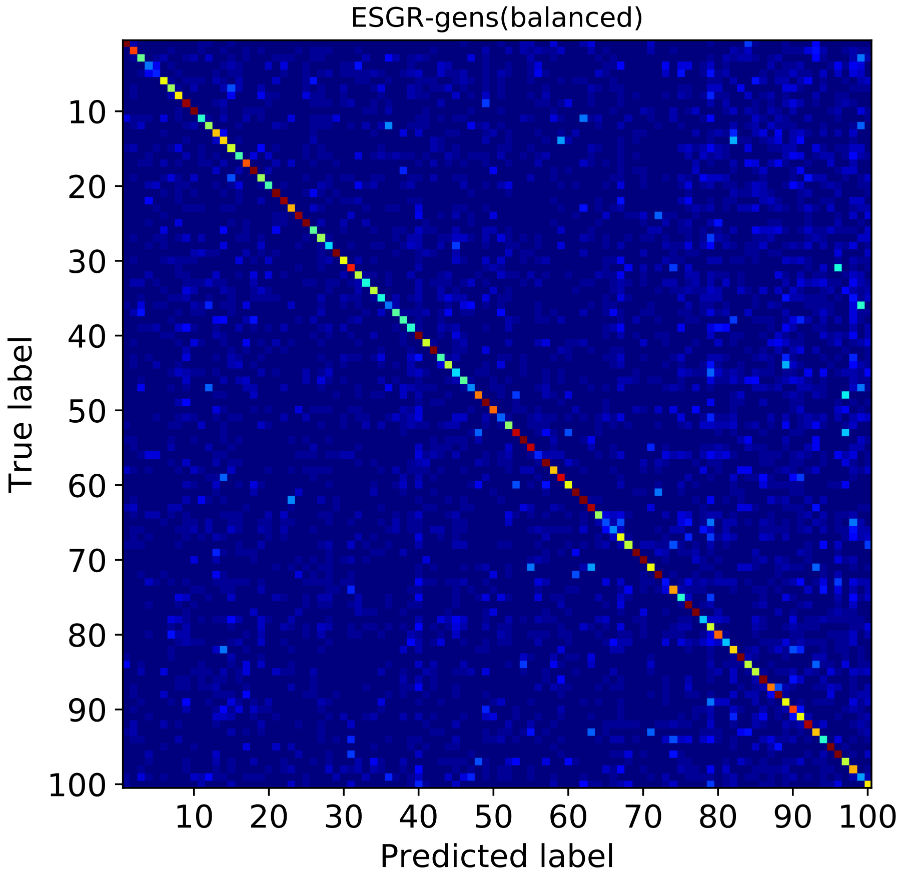

Figure 6: Visualization of the confusion matrix of the balanced version of ESGR-*gens* under the 10-class adding setting on CIFAR-100. Note that it doesn't have a distinct bar on the right of the image anymore, implying that a test sample may be predicted as each of the seen classes equally.

Table 1: Results of different methods under the 10-class adding setting on CIFAR-100

Method	Final	Adaptation	Forget Rate
Joint training	0.4611	0.5437	15.19%
ESGR-*gens* (imbalanced)	0.3388	0.6393	47.00%
ESGR-*gens* (balanced)	0.3597	0.4624	22.21%
ESGR-*mix* (imbalanced)	0.3576	0.6243	42.72%
ESGR-*mix* (balanced)	0.3803	0.5082	25.17%
iCaRL	0.356	0.4865	26.82%

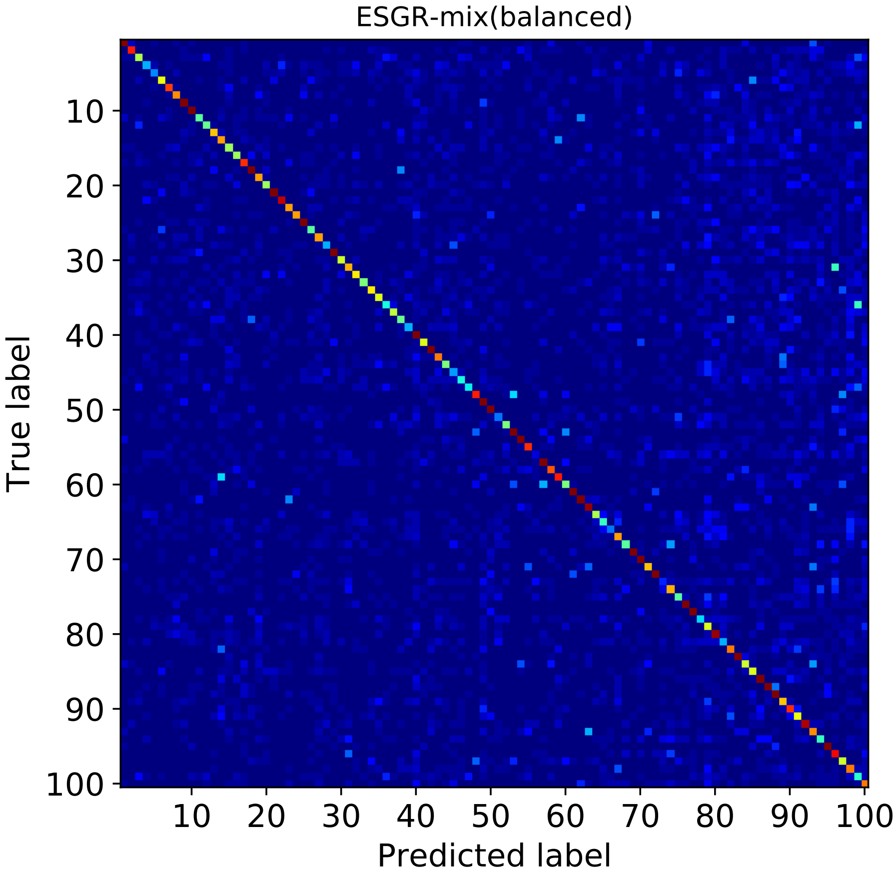

Figure 7: Visualization of the confusion matrix of the balanced version of ESGR-*mix* under the 10-class adding setting on CIFAR-100. Similar to ESGR-*gens* (Figure 6), a test sample may be predicted as each of the seen classes equally.

is indeed the main cause of the forgetting effect. Also, from the visualization of the confusion matrices of ESGR-*gens* (balanced) and ESGR-*mix* (balanced) (Figure 6 and 7), it can be easily observed that there is not a distinct bar in the right of the confusion matrix any more, indicating that a sample can be predicted as each of the seen classes equally and it doesn't have a "imbalance" problem.

References

[1] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[2] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[3] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pages 614–629. Springer, 2016.

[4] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

[5] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5533–5542, 2017.

[6] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2994–3003, 2017.