Generative Models from the perspective of Continual Learning*

Timothée Lesort[†1,2], Hugo Caselles-Dupré[†1,3], Michael Garcia-Ortiz[3], Jean-François Goudou[2], and David Filliat[1]

[1]Flowers Laboratory (ENSTA ParisTech & INRIA)
[2]Thales, Theresis Laboratory
[3]Softbank Robotics Europe

Abstract

Which generative model is the most suitable for Continual Learning? This paper aims at evaluating and comparing generative models on disjoint sequential image generation tasks. We investigate how several models learn and forget, considering various strategies: rehearsal, regularization, generative replay and fine-tuning. We used two quantitative metrics to estimate the generation quality and memory ability. We experiment with sequential tasks on three commonly used benchmarks for Continual Learning (MNIST, Fashion MNIST). We found that among all models, the original GAN performs best and among Continual Learning strategies, generative replay outperforms all other methods.

1 Introduction

Learning in a continual fashion is a key aspect for cognitive development among biological species (Fagot and Cook, 2006). In Machine Learning, such learning scenario has been formalized as a Continual Learning (CL) setting (Srivastava et al., 2013; Nguyen et al., 2017). Continual algorithms should be able to learn from a data distribution that change over time without forgetting crucial information.

In this paper, we conduct a comparative study of generative models with different CL strategies. We experiment on ten sequential disjoint tasks, using commonly used benchmarks for CL: MNIST (LeCun et al., 1998), Fashion MNIST (Xiao et al., 2017).

We evaluate several generative models from two frameworks: Variational Auto-Encoders and Generative Adversarial Networks. We compare results on approaches inspired from classification CL settings: *finetuning*, *rehearsal*, *regularization* and *generative replay*. *Generative replay* consists in using generated samples to maintain knowledge from previous tasks. We evaluate the generative models with two quantitative metrics, Fréchet Inception Distance (Heusel et al., 2017) and Fitting Capacity (Lesort et al., 2018), as well as visualization. Also, we discuss the data availability and scalability of CL strategies. Our contributions are:

- Evaluating a wide range of generative models in a Continual Learning setting.
- Highlight success/failure modes of combinations of generative models and CL approaches.
- Comparing, in a CL setting, two evaluation metrics of generative models.

*Extended version is currently under review for ICLR 2019.
†Equal contribution.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

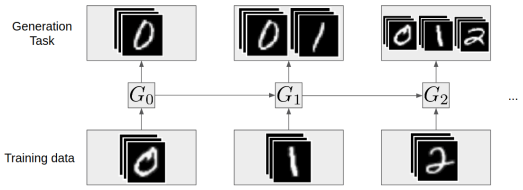


Figure 1: The disjoint setting considered. At task i the training set includes images belonging to category i, and the task is to generate samples from all previously seen categories.

2 Related work

Continual Learning has mainly been developed for discriminative models. Previously proposed approaches can be classified into four main methods. (1) *Rehearsal* methods keep samples from previous tasks (Rebuffi et al., 2017; Nguyen et al., 2017). (2) *Regularization* constrains weight updates in order to maintain knowledge from previous tasks and thus avoid forgetting. Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) has become the standard method for this type of regularization. (3) *Dynamic architectures* are also used to maintain past knowledge and learn new information (Rusu et al., 2016; Li and Hoiem, 2016; Fernando et al., 2017). (4) *Generative replay* (Shin et al., 2017; Venkatesan et al., 2017) uses a generative model to produce samples from previous tasks.

Discriminative and generative models do not share the same learning objective and architecture. For this reason, CL strategies for discriminative models are usually not directly applicable to generative models. CL in the context of generative models remains largely unexplored compared to CL for discriminative models. Existing approaches are either reserved to VAEs (Nguyen et al., 2017; Ramapuram et al., 2017; Achille et al., 2018) or GANs (Seff et al., 2017). A different approach, applicable to both adversarial and variational frameworks, is Generative Replay. It uses two generative models: one which acts as a memory, capable of generating all past tasks, and one that learns to generate data from all past tasks and the current task. It has mainly been used as a method for Continual Learning of discriminative models (Shin et al., 2017; Venkatesan et al., 2017; Shah et al., 2018). Recently, Wu et al. (2018) have developed a similar approach called Memory Replay GANs.

3 Approach

Typical previous work on Continual Learning for generative models focus on presenting a novel CL technique and comparing it to previous approaches, on one type of generative model (e.g. GAN or VAE). On the contrary, we focus on searching for the best generative model and CL strategy association. For now, empirical evaluation remain the only way to find the best performing combinations. Hence, we compare several existing CL strategies on a wide variety of generative models with the objective of finding the most suited generative model for Continual Learning. We use two evaluation metrics.

The Fréchet Inception Distance (FID) (Heusel et al., 2017) is a commonly used metric for evaluating generative models. It is designed to improve on the Inception Score (IS) (Salimans et al., 2016) which has many intrinsic shortcomings, as well as additional problems when used on a dataset different than ImageNet (Barratt and Sharma, 2018). FID circumvent these issues by comparing the statistics of generated samples to real samples, instead of evaluating generated samples directly.

A different approach is to use labeled generated samples from a generator G (GAN or VAE) to train a classifier and evaluate it afterwards on real data (Lesort et al., 2018). This evaluation, called Fitting Capacity of G, is the test accuracy of a classifier trained with G's samples.

For all the progress made in quantitative metrics for evaluating generative models (Borji, 2018), qualitative evaluation remains a widely used and informative method. While visualizing samples provides a instantaneous detection of failure, it does not provide a way to compare two well-performing models. It is not a rigorous evaluation and it may be misleading when evaluating sample variability.

4 Experimental setup

4.1 Datasets, tasks, metrics and models

Our main experiments use 10 sequential tasks created using the MNIST, Fashion MNIST dataset. For each dataset, we define 10 sequential tasks, one task corresponds to learning to generate a new class and all the previous ones (See Fig. 1 for an example on MNIST). Both evaluations, FID and Fitting Capacity of generative models, are computed at the end of each task on the full test set.

We use 6 different generative models. We experiment with the original and conditional version of GANs (Goodfellow et al., 2014) and VAEs (Kingma and Welling, 2013). We also added WGAN (Arjovsky et al., 2017) and a variant of it WGAN-GP (Gulrajani et al., 2017), as they are commonly used baselines that supposedly improve upon the original GAN.

4.2 Strategies for continual learning

We focus on strategies that are usable in both the variational and adversarial frameworks. We first use a vanilla Rehearsal method, where we keep a 10 samples of each observed task, and add those samples to the training set of the current generative model. We also experiment with EWC. We apply the method described by Seff et al. (2017) for GANs, i.e. the penalty is applied only on the generator's weights (even though they specificy it only works in the conditional case), and for VAEs the penalty is applied on all weights. The last method is Generative Replay, described in Section 2. Generative replay is a dual-model approach where a "frozen" generative model G_{t-1} is used to sample from previously learned distributions and a "current" generative model G_t is used to learn the current distribution and G_{t-1} distribution. When a task is over, the G_{t-1} is replaced by a copy of G_t , and learning can continue. We also experiment with 3 baselines: Fine-tuning, Upperbound Data, for which one generative model is trained on joint data from all past tasks, and Upperbound Model, for which one separate generator is trained for each task. Our experimental setup is detailed in Appendix C. Code is available online [3]

5 Results

Our main results are displayed in Fig. 2. A well performing model should increase its Fitting Capacity and decrease its FID. We observe a strong correlation between both evaluation (see an example on GAN for MNIST in Appendix Fig. 8 and F for full results). The best combination we found is Generative Replay + GAN with a mean Fitting Capacity of 95.81% on MNIST and 81.52% on Fashion MNIST. The relative performance of each CL method on GAN can be analyzed class by class in Fig. 9. We observe that, for the adversarial framework, Generative Replay outperforms other approaches by a significant margin. However, for the variational framework, the Rehearsal approach was the best performing. The Rehearsal approach worked quite well but is unsatisfactory for CGAN and WGAN-GP. Indeed, the Fitting Capacity is lower than the accuracy of a classifier trained on 10 samples per classes (see Fig. 4a and 4b in annex). In our setting, EWC is not able to overcome catastrophic forgetting and performs as well as the naive Fine-tuning baseline which is contradictory with the results of Seff et al. (2017) who found EWC successful in a slightly different setting. We replicated their result in a setting where there are two classes by tasks (see Appendix H for details), showing the strong effect of task definition.

Our results do not give a clear distinction between conditional and unconditional models. However, adversarial methods perform significantly better than variational methods. GANs variants are able to produce better, sharper quality and variety of samples, as observed in Fig. 13 and 14 in Appendix J. Hence, adversarial methods seem more viable for CL. We can link the accuracy from 4a and 4b to the Fitting Capacity results. As an example, we can estimate that GAN with Generative Replay is equivalent for both datasets to a memory of approximately 100 samples per class.

[3] Github link: `https://github.com/TLESORT/Generative_Continual_Learning`

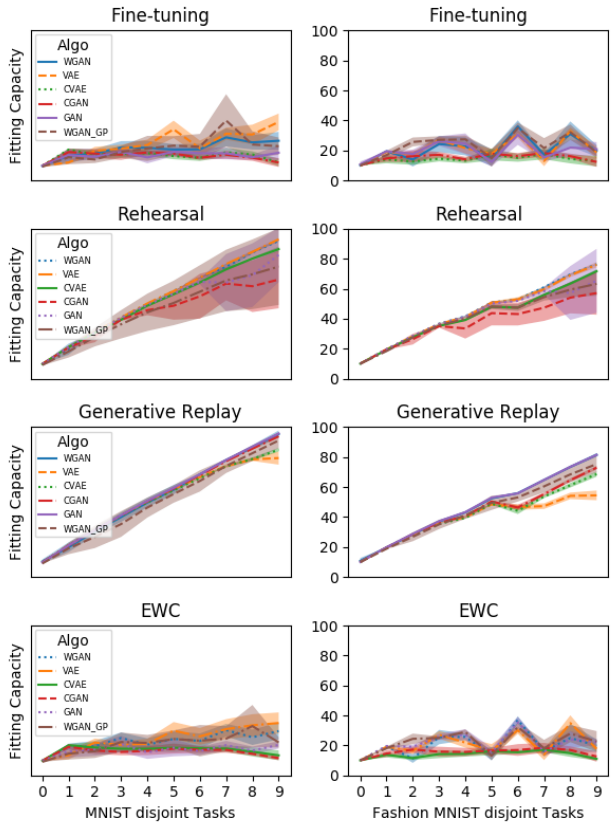


Figure 2: Means and standard deviations over 8 seeds of Fitting Capacity metric evaluation of VAE, CVAE, GAN, CGAN and WGAN. The four considered CL strategies are: Fine Tuning, Generative Replay, Rehearsal and EWC. The setting is 10 disjoint tasks on MNIST and Fashion MNIST.

6 Discussion and future work

Besides the quantitative results and visual evaluation of the generated samples, the evaluated strategies have, by design, specific characteristics relevant to CL that we discuss here. Rehearsal violates the data availability assumption, often required in CL scenarios, by recording part of the samples. Plus, the risk of overfitting is high when only few samples represent a task. EWC and Generative Replay respect this assumption. EWC does not add any computational overload during training, but it comes at the cost of computing the Fisher information matrix, and storing its values as well as a copy of previous parameters. The memory needed for EWC to save information from the past is twice the size of the model which may be expensive in comparison to rehearsal methods. Nevertheless, with Rehearsal and Generative Replay, the model has more and more samples to learn from at each new task, which makes training more costly. A natural line of future work is to experiment with a more challenging dataset. We present preliminary results with CIFAR10 (Krizhevsky et al., 2009) in Appendix E. We use WGAN_GP with Generative Replay as it worked best on a single generation task on CIFAR10. In a CL scenario, results show that the model accumulates generation errors as tasks are sequentially presented, which ultimately results in blurry indistinguishable samples.

7 Conclusion and future work

In this paper, we experimented with the viability and effectiveness of generative models on Continual Learning (CL) settings. Our experiments indicate that on MNIST and Fashion MNIST, the original GAN combined to the Generative Replay method is particularly effective. This method avoids catastrophic forgetting by using the generator as a memory to sample from the previous tasks and hence maintain past knowledge.

References

Alessandro Achille, Tom Eccles, Loic Matthey, Christopher P Burgess, Nick Watters, Alexander Lerchner, and Irina Higgins. 2018. Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies. arXiv preprint arXiv:1808.06508 (2018).

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017).

Shane Barratt and Rishi Sharma. 2018. A Note on the Inception Score. arXiv preprint arXiv:1801.01973 (2018).

Ali Borji. 2018. Pros and Cons of GAN Evaluation Measures. arXiv preprint arXiv:1802.03446 (2018).

Joël Fagot and Robert G Cook. 2006. Evidence for large long-term memory capacities in baboons and pigeons and its implications for learning and the evolution of cognition. Proceedings of the National Academy of Sciences 103, 46 (2006), 17564–17567.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. 2017. PathNet: Evolution Channels Gradient Descent in Super Neural Networks. CoRR abs/1701.08734 (2017). arXiv:1701.08734 `http://arxiv.org/abs/1701.08734`

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Advances in neural information processing systems. 2672–2680.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In Advances in Neural Information Processing Systems. 5767–5777.

Xu He and Herbert Jaeger. 2018. Overcoming Catastrophic Interference using Conceptor-Aided Backpropagation. In International Conference on Learning Representations. `https://openreview.net/forum?id=B1al7jg0b`

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in Neural Information Processing Systems. 6626–6637.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences (2017), 201611835.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2009. CIFAR-10 (Canadian Institute for Advanced Research). (2009). `http://www.cs.toronto.edu/~kriz/cifar.html`

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

Timothée Lesort, Jean-François Goudou, and David Filliat. 2018. Training Discriminative Models to Evaluate Generative Ones. arXiv preprint arXiv:1806.10840 (2018).

Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. 2017. Alice: Towards understanding adversarial learning for joint distribution matching. In Advances in Neural Information Processing Systems. 5495–5503.

Z. Li and D. Hoiem. 2016. Learning without Forgetting. ArXiv e-prints (June 2016). arXiv:cs.CV/1606.09282

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. 2017. Variational continual learning. arXiv preprint arXiv:1710.10628 (2017).

Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis. 2017. Lifelong Generative Modeling. arXiv preprint arXiv:1705.09847 (2017).

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning.

A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. 2016. Progressive Neural Networks. ArXiv e-prints (June 2016). arXiv:cs.LG/1606.04671

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. CoRR abs/1606.03498 (2016).

Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. 2017. Continual learning in generative adversarial nets. arXiv preprint arXiv:1705.08395 (2017).

Haseeb Shah, Khurram Javed, and Faisal Shafait. 2018. Distillation Techniques for Pseudo-rehearsal Based Incremental Learning. arXiv preprint arXiv:1807.02799 (2018).

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In Advances in Neural Information Processing Systems. 2990–2999.

Rupesh K Srivastava, Jonathan Masci, Sohrob Kazerounian, Faustino Gomez, and Jürgen Schmidhuber. 2013. Compete to Compute. In Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2310–2318. http://papers.nips.cc/paper/5059-compete-to-compute.pdf

Ragav Venkatesan, Hemanth Venkateswara, Sethuraman Panchanathan, and Baoxin Li. 2017. A Strategy for an Uncompromising Incremental Learner. arXiv preprint arXiv:1705.00744 (2017).

Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, and Bogdan Raducanu. 2018. Memory Replay GANs: learning to generate images from new categories without forgetting. arXiv preprint arXiv:1809.02058 (2018).

Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. (2017). arXiv:cs.LG/1708.07747

A Samples at each step

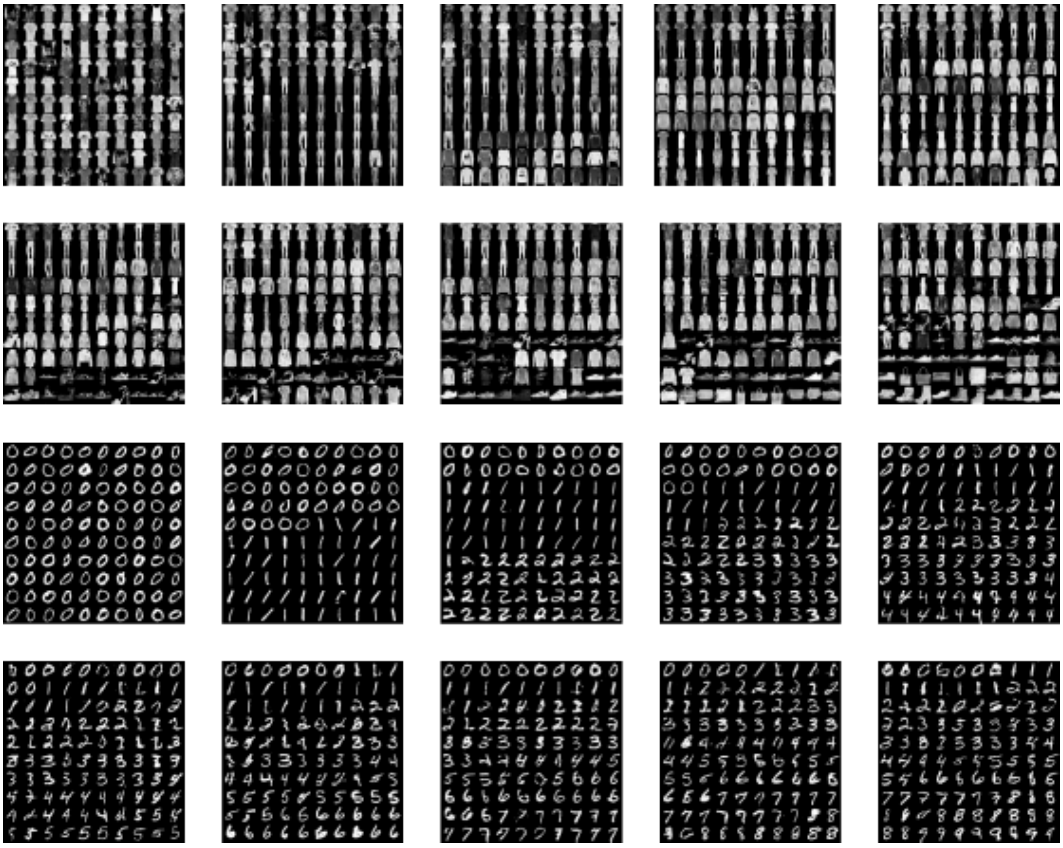
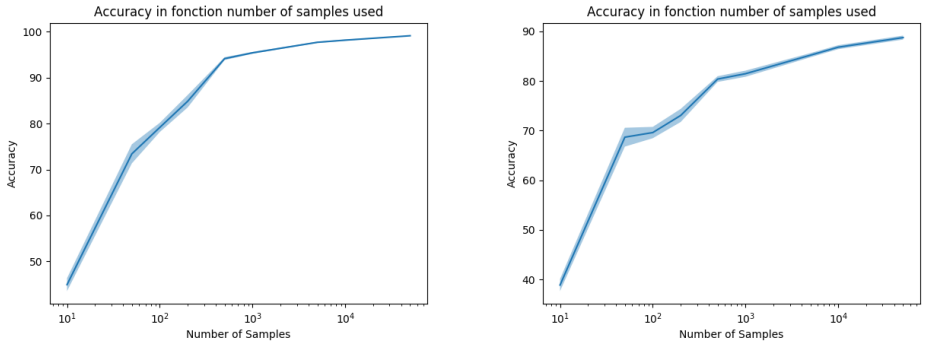


Figure 3: Samples of a well performing solution : GAN + Generative Replay for each step in a sequence of 10 tasks with MNIST and Fashion MNIST.

B Classifiers performances



(a) MNIST Classifier accuracy

(b) fashion-MNIST Classifier accuracy

Figure 4: Test set classification accuracy as a function of number of training samples, on MNIST.

Fig. 4a and 4b make possible to estimate the number of samples needed to solve the full task. Furthermore by comparing against the fitting capacity we can estimate how many different images of the dataset a generator can produce.

C Experimental setup

Our experiments are done on 8 seeds with 50 epochs per tasks for MNIST and Fashion MNIST using Adam (Kingma and Ba, 2014) for optimization (for hyper-parameter settings, see Appendix I).

Rehearsal: We balance the resulting dataset by copying the saved samples so that each class has the same number of samples. The number of samples selected, here 10, is motivated by the results in Fig. 4a and 4b, where we show that 10 samples per class is enough to get a satisfactory but not maximal validation accuracy for a classification task on MNIST and Fashion MNIST. As the Fitting Capacity share the same test set, we can compare the original accuracy with 10 samples per task to the final fitting capacity. A higher Fitting capacity show that the memory prevents catastrophic forgetting. Equal Fitting Capacity means overfitting of the saved samples and lower Fitting Capacity means that the generator failed to even memorize these samples.

EWC : As tasks are sequentially presented, we choose to update the diagonal of the Fisher information matrix by cumulatively adding the new one to the previous one.

Metrics, FID: Heusel et al. (2017) propose using the Fréchet distance between two multivariate Gaussians:

$$FID = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}), \tag{1}$$

where the statistics (μ_r, Σ_r) and (μ_g, Σ_g) are the activations of a specific layer of a discriminative neural network trained on ImageNet, for real and generated samples respectively. A lower FID correspond to more similar real and generated samples as measured by the distance between their activation distributions. Originally the activation should be taken from a given layer of a given Inception-v3 instance, however this setting can be adapted with another classifier in order to compare a set of models with each other (Li et al., 2017; Lesort et al., 2018).

Metrics, Fitting capacity: It measures the generator's ability to train a classifier that generalize well on a testing set, i.e the generator's ability to fit the distribution of the testing set. This method aims at evaluating generative models on complex characteristics of data and not only on their features distribution. In the original paper, the authors annotated samples by generating them conditionally, either with a conditional model or by using one unconditional model for each class. In this paper, we also use an adaptation of the Fitting Capacity where data from unconditional models are labelled by an expert network trained on the dataset.

D Corollary results

Catastrophic forgetting can easily be visualized in Fig.9. However the Fitting Capacity of Fine-tuning and EWC in Table 1 is higher than expected for unconditional models. As the generator is only able to produce samples from the last task, the Fitting capacity should be near 10%. This is a downside of using an expert for annotation before computing the Fitting Capacity. Fuzzy samples can be wrongly annotated, which can artificially increase the labels variability and thus the Fitting Capacity of low performing models, e.g., VAE with Fine-tuning. However, this results stay lower than the Fitting Capacity of well performing models.

Incidentally, an important side result is that the Fitting capacity of conditional generative models is comparable to results of Continual Learning classification. Our best performance in this setting is with CGAN: 94.7% on MNIST and 75.44% on Fashion MNIST . In a similar setting with 2 sequential tasks, which is arguably easier than our setting (one with digits from 0,1,2,3,4 and another with 5,6,7,8,9), He and Jaeger (2018) achieve a performance of 94.91%. This shows that using generative models for CL could be a competitive tool in a classification scenario. It is worth noting that we did not compare our results of unconditional models Fitting Capacity with classification state of the art. Indeed, in this case, the Fitting capacity is based on an annotation from an expert not trained in a continual setting. The comparison would then not be fair.

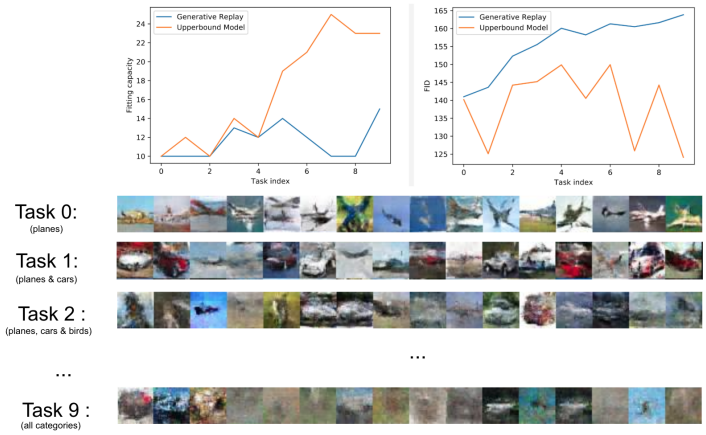


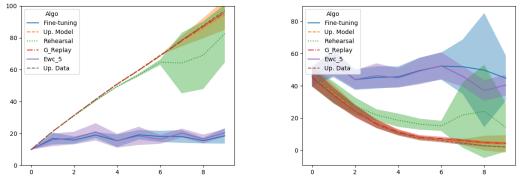
Figure 5: Fitting capacity and FID score of Generative Replay applied to WGAN_GP, on CIFAR10. Samples of task 0 (planes), task 1 (planes and cars), and the final task 9 are presented below.

E CIFAR10 results

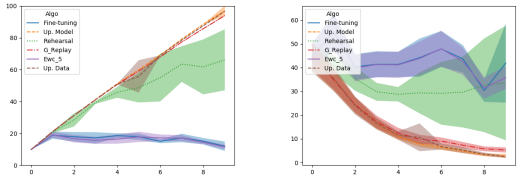
In this experiment, we selected the best performing CL method on MNIST and Fashion MNIST, Generative Replay, and tested it on the more challenging CIFAR10 dataset. The setting remains the same, one task for each category, for which the aim is to avoid forgetting of previously seen categories. We selected WGAN-GP because it produced the most satisfying samples on CIFAR10 (see Fig. 15 in Appendix J).

Results are provided in Fig. 5. Since the dataset is composed of real-life images, the generation task is much harder to complete. As seen in Task 0, the generator is able to produce images that roughly resemble samples of the category, here planes. As tasks are presented, minor generation errors accumulated and snowballed into the result in task 9: samples are blurry and categories are indistinguishable. As a consequence, the FID and Fitting Capacity scores do not improve at each task. We also trained the same model separately on each task, and while the result is visually satisfactory, the quantitative metrics show that generation quality is not excellent. This negative result shows that training a generative model on a sequential task scenario does not reduce to successfully training a generative model on all data or each category, and that state-of-the-art generative models struggle on real-life image datasets like CIFAR10. Designing a CL strategy for these type of datasets remains a challenge.

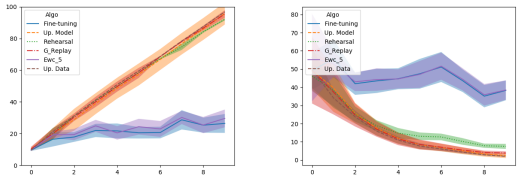
F Additional figures



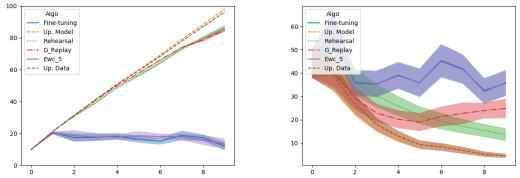
(a) Fitting Capacity GAN (b) FID GAN



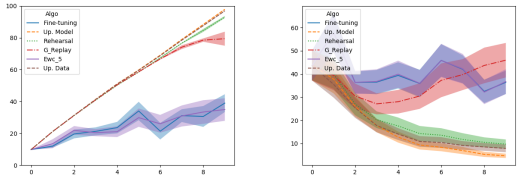
(c) Fitting Capacity CGAN (d) FID CGAN



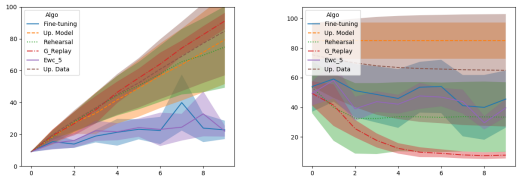
(e) Fitting Capacity WGAN (f) FID WGAN



(g) Fitting Capacity CVAE (h) FID CVAE



(i) Fitting Capacity VAE (j) FID VAE

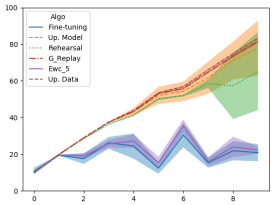


(k) F. Capacity WGAN-GP (l) FID WGAN-GP

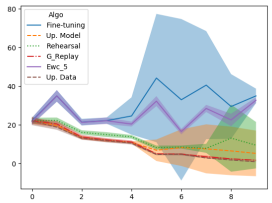
Figure 6: Comparison of the Fitting Capacity and FID results on MNIST.

Table 1: Mean and standard deviations for Fitting Capacity (in %) metric evaluation on last task of 10 disjoint task setting, on MNIST and Fashion MNIST, over 8 seeds.

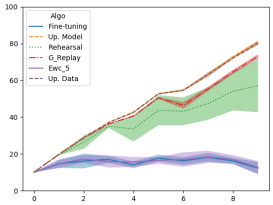
Strategy	Dataset	GAN	CGAN	WGAN	WGAN-GP	VAE	CVAE
Fine-tuning	MNIST	18.43±4.85	11.93±2.97	23.17±5.66	22.79±5.75	38.98±5.57	11.96±2.56
EWC	-	20.34±2.39	11.53±1.42	29.57±5.59	22.00±3.39	34.93±7.06	13.37±3.28
Rehearsal	-	82.69±18.21	66.14±19.2	92.05±0.64	74.79±25.25	92.99±0.64	86.47±1.69
Generative Replay	-	**95.81**±0.31	93.89±0.35	95.41±2.41	91.12±5.09	79.38±4.40	84.95±1.24
Upperbound Model	-	94.50±9.51	96.84±3.22	95.72±6.93	79.41±27.85	97.82±0.17	97.89±0.12
Upperbound Data	-	97.10±0.13	96.65±0.21	96.76±0.29	84.79±27.76	96.88±0.27	96.17±0.19
Fine-tuning	Fashion MNIST	20.82±4.69	12.30±3.33	19.68±3.92	18.75±2.58	18.60±4.24	12.82±3.55
EWC	-	22.22±2.03	12.58±3.48	19.81±4.18	22.63±6.91	17.70±1.83	11.00±1.16
Rehearsal	-	65.34±21.3	57.12±14.4	76.32±0.33	63.28±7.9	76.03±1.77	71.73±1.29
Generative Replay	-	**81.52**±0.87	72.98±1.22	81.50±1.26	75.37±5.49	54.49±3.24	68.70±1.71
Upperbound Model	-	77.93±15.07	80.96±0.69	73.20±5.63	65.5±2.69	78.64±1.36	79.15±0.96
Upperbound Data	-	83.27±0.41	80.09±0.94	83.29±0.52	81.5±0.50	80.21±0.79	79.51±0.55



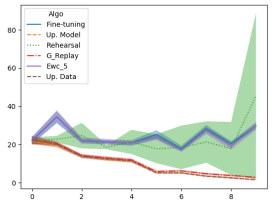
(a) Fitting Capacity GAN



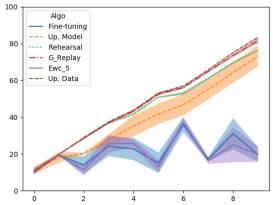
(b) FID GAN



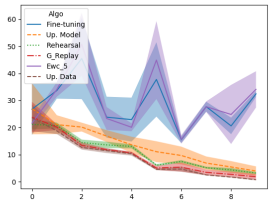
(c) Fitting Capacity CGAN



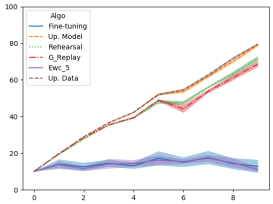
(d) FID CGAN



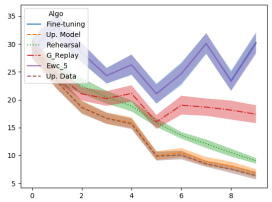
(e) Fitting Capacity WGAN



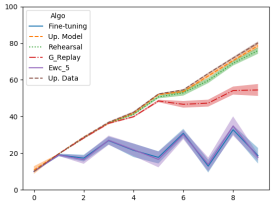
(f) FID WGAN



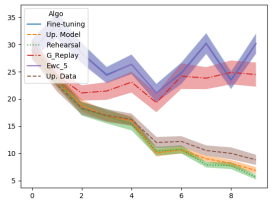
(g) Fitting Capacity CVAE



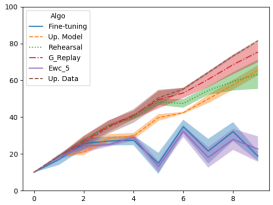
(h) FID CVAE



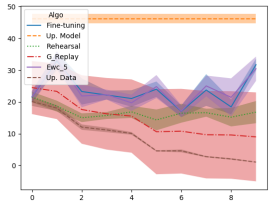
(i) Fitting Capacity VAE



(j) FID VAE



(k) Fitting Capacity WGAN-GP



(l) FID WGAN-GP

Figure 7: Comparison of the Fitting Capacity and FID results on Fashion MNIST.

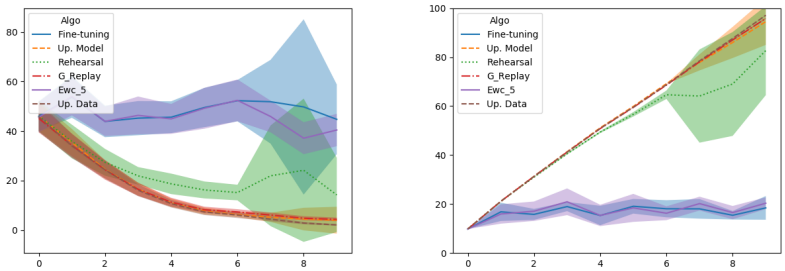
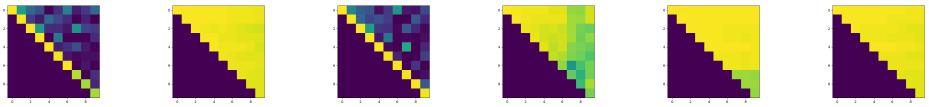


Figure 8: Comparison, averaged over 8 seeds, between FID results(left, lower is better) and Fitting Capacity results (right, higher is better) with GAN trained on MNIST.



(a) Fine-tuning (b) Gen. Replay (c) EWC (d) Rehearsal (e) Up. Model (f) Up. Data

Figure 9: Fitting Capacity results for GAN on MNIST. Each square is the accuracy on one class for one task. Abscissa is the task index (left: 0 , right: 9) and orderly is the class index (top: 0, down: 9). The accuracy is proportional to the color (dark blue : 0%, yellow 100%)

G Comparison with (Wu et al., 2018)

Table 2: Our results using the metric proposed by Wu et al. (2018). Rehearsal, even thought suffers from mode collapse, performs as good as Generative Replay, which visually produce better samples.

Strategy	Dataset	CVAE	CGAN
Rehearsal	Mnist	99.86%	95.72%
Generative Replay	-	99.70%	99.26%
Ewc	-	10.78%	10.54%
Baseline	-	10.70%	10.52%
Rehearsal	Fashion	94.42%	92.36%
Generative Replay	-	88.64%	89.98%
Ewc	-	10.62%	10.50%
Baseline	-	10.68%	10.60%

H Reproduction of results in (Seff et al., 2017)

Figure 10: CGAN augmented with EWC. MNIST samples after 5 sequential tasks of 2 digits each. Catastrophic forgetting in avoided.

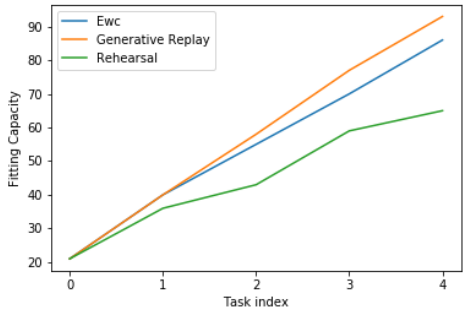


Figure 11: CGAN results with EWC, Rehearsal and Generative Replay, on 5 sequential tasks of 2 digits each. EWC performs well, compared to the results obtained on a 10 sequential task setting.

Table 3: Hyperparameters for MNIST and Fashion MNIST all models (all CL strategies have the same training hyper parameters)

Model Datasets	Epochs	Lr	n_critic	beta1	beta2	Batch	lambda	clipping value
GAN	50	2e-4	1	5e-1	0.999	64	-	-
CGAN	50	2e-4	1	5e-1	0.999	64	-	-
VAE	50	2e-4	1	5e-1	0.999	64	-	-
CVAE	50	2e-4	1	5e-1	0.999	64	-	-
WGAN	50	2e-4	2	5e-1	0.999	64	-	0.01
WGAN_GP	50	2e-4	2	5e-1	0.999	64	0.25	-
Classifier	50	0.5	-	5e-1	0.999	64	-	-

I Hyperparameters

J Samples

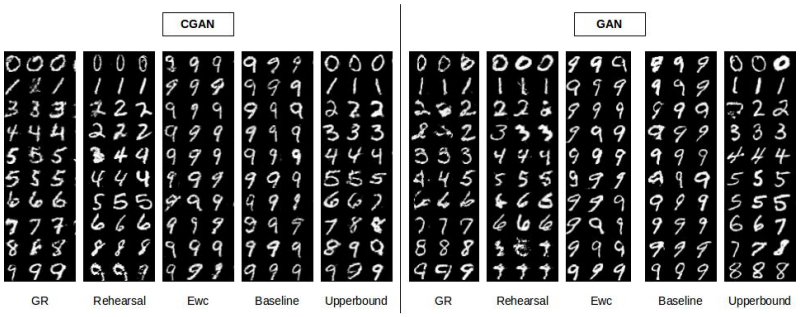


Figure 12: Samples from GAN and Conditional-GAN for each Continual Learning strategy. Upperbound refers to Upperbound Model.

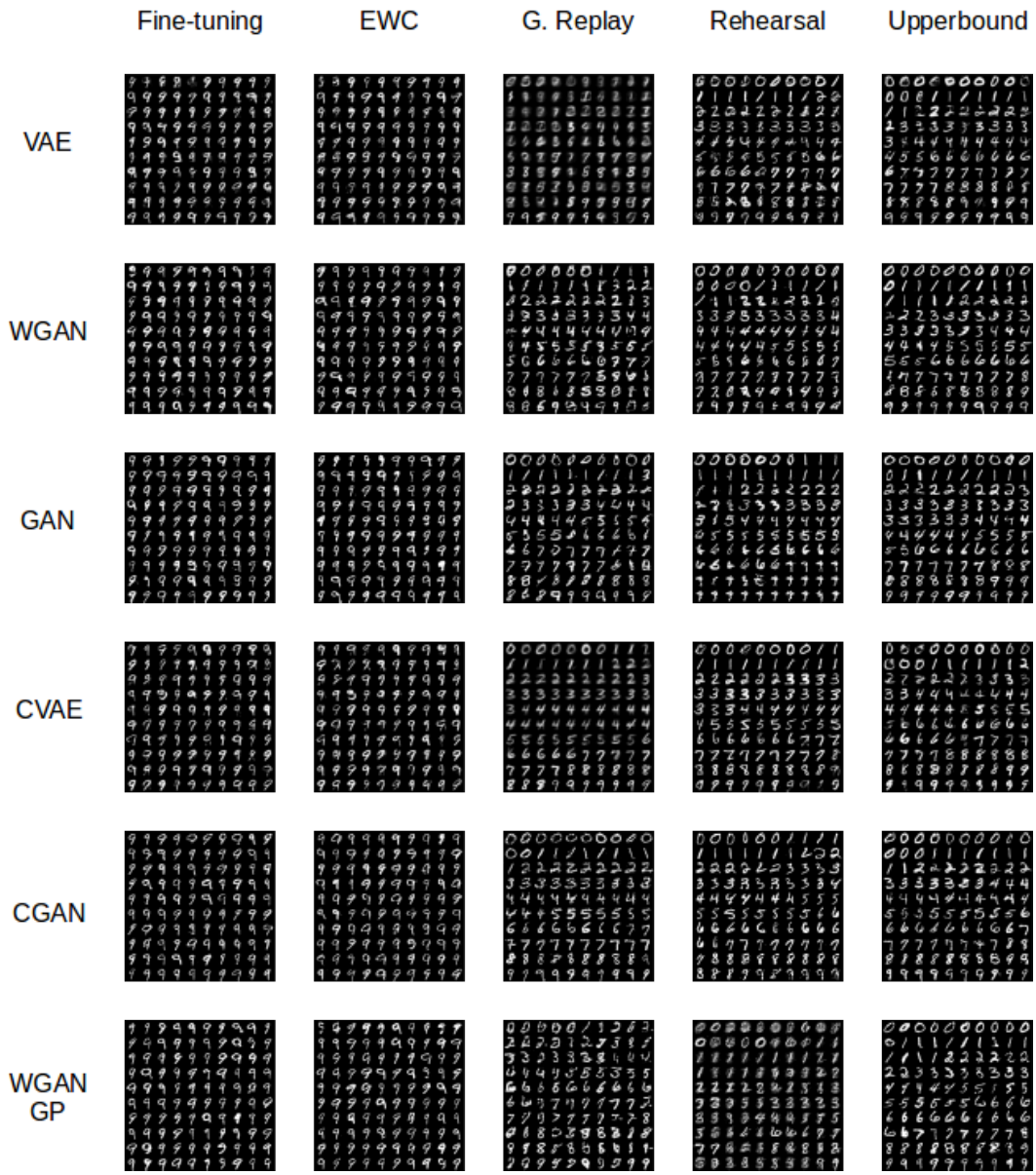


Figure 13: MNIST samples for each generative model and each Continual Learning strategy, at the end of training on 10 sequential tasks. The goal is to produce samples from all categories.

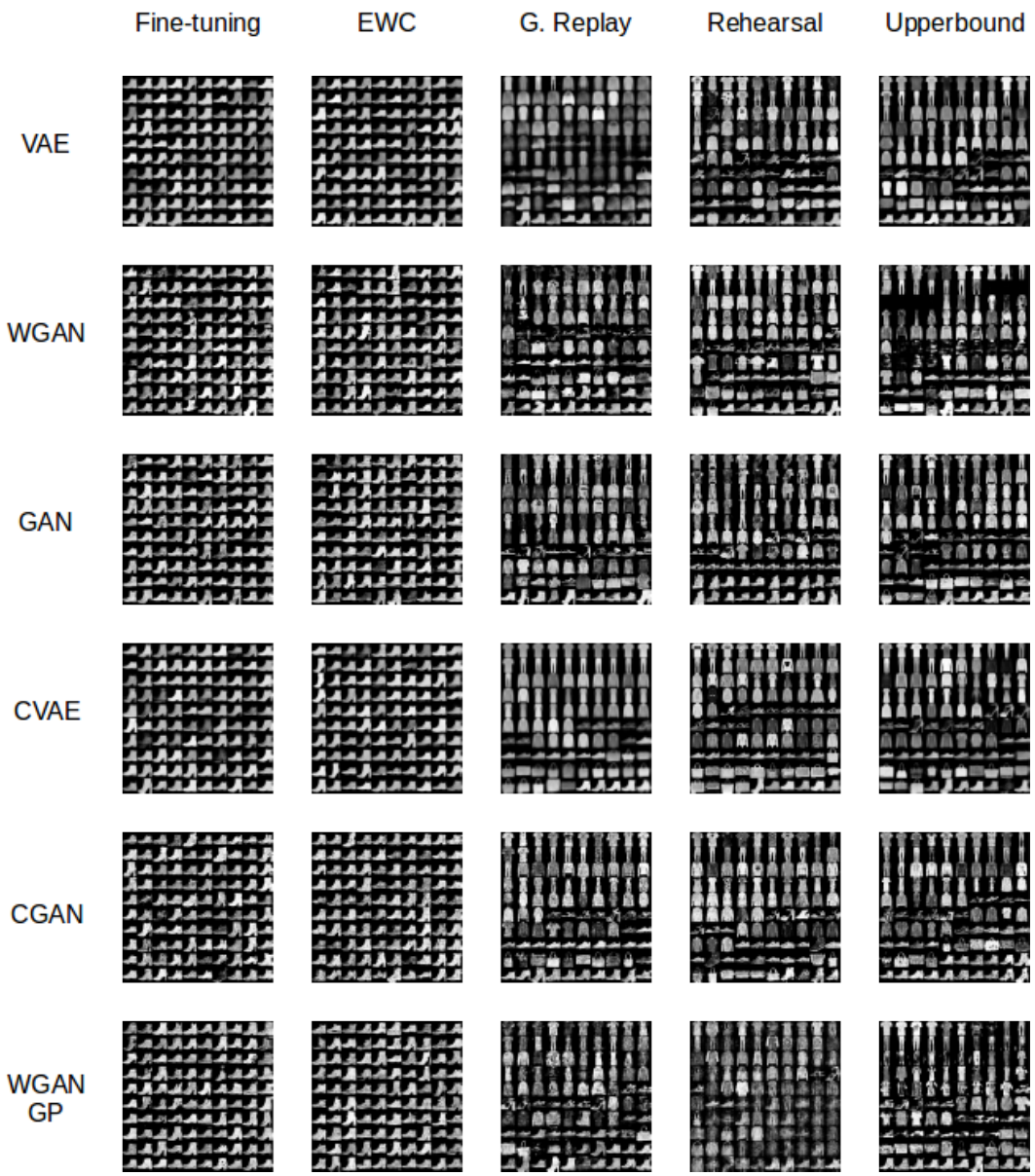


Figure 14: Fashion MNIST samples for each generative model and each Continual Learning strategy, at the end of training on 10 sequential tasks. The goal is to produce samples from all categories.

Figure 15: WGAN-GP samples on CIFAR10, with on training for each separate category. The implementation we used is available here: `https://github.com/caogang/wgan-gp`. Classes, from 0 to 9, are planes, cars, birds, cats, deers, dogs, frogs, horses, ships and trucks.