



HAL
open science

Deep Online Storage-Free Learning on Unordered Image Streams

Andrey Besedin, Pierre Blanchart, Michel Crucianu, Marin Ferecatu

► **To cite this version:**

Andrey Besedin, Pierre Blanchart, Michel Crucianu, Marin Ferecatu. Deep Online Storage-Free Learning on Unordered Image Streams. Springer; Anna Monreale, Carlos Alzate, Michael Kamp, Yamuna Krishnamurthy, Daniel Paurat, Moamar Sayed-Mouchaweh, Albert Bifet, João Gama, Rita P. Ribeiro. ECML PKDD: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Sep 2018, Dublin, Ireland. 967, pp.103-112, 2019, Communications in Computer and Information Science - ECML PKDD 2018 Workshops. 10.1007/978-3-030-14880-5_9. hal-02454302

HAL Id: hal-02454302

<https://hal.science/hal-02454302>

Submitted on 13 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

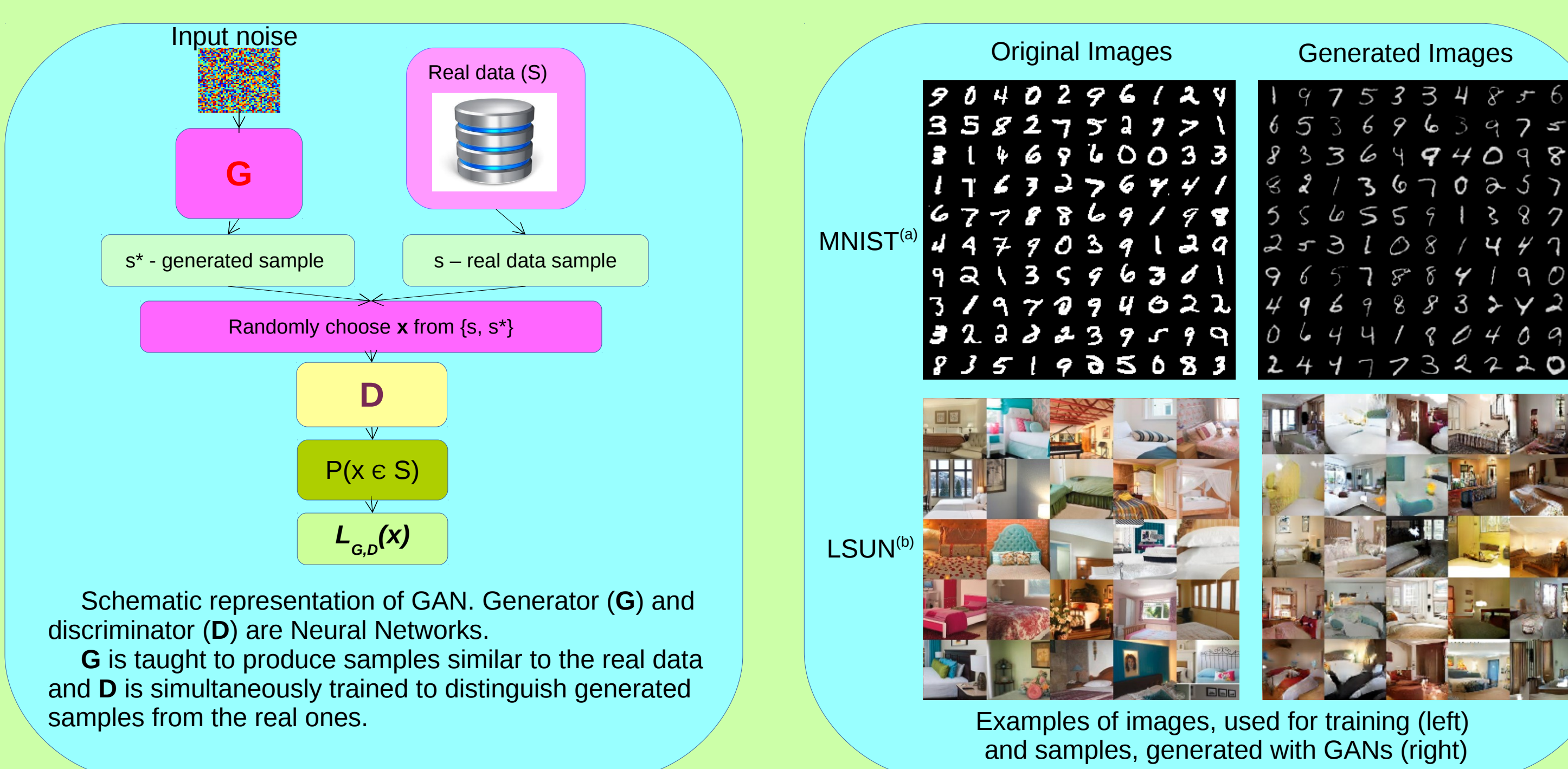
Deep Online Storage-Free Learning on Unordered Image Streams

BACKGROUND

Deep learning on data streams vs static learning, application for the classification task

| Static data | Stream data |
|---|--|
| <p>The whole dataset is stored on the disc and is available at any moment.</p> | <p>Data come continuously and have to be processed in real time.</p> |
| Deep learning on static dataset | Deep learning on data streams |
| <ol style="list-style-type: none"> 1) Dataset is of a limited size and exact training time per epoch can always be estimate. 2) Sizes of data classes are usually similar or can be equalized before training 3) The whole dataset is available for learning and retraining at any moment 4) All the data classes are available at every training epoch. Moreover, training is based on gradient-based methods, usually need multiple training epochs 5) Number of classes together with labels for each data sample are provided before training. | <ol style="list-style-type: none"> 1) Data comes with high speed → Need real time information extraction and learning. 2) Classes don't have to be equal → Want to learn well on classes of different sizes. 3) Streams are potentially of a very big (infinite) size → Want to avoid storing historical data. 4) At some point several data classes can never appear again in the stream → Need to avoid catastrophic forgetting of already learned information. 5) Data classes, never seen by the learning system before, can appear → Need a model, able to adapt to new classes |

Generative Adversarial Networks (GAN) [1]



Minimax game between G and D is based on loss function:

$$\min_G \max_D L_{G,D}(x) = E_{x \sim p_s(x)} [\log D(x)] + E_{s \sim p_s(s^*)} [\log(1 - D(G(s^*)))]$$

Deep Convolutional GANs [2] – difference from classic GANs:

- All pooling layers are replaced by strided convolutions, which allows the network learn its own down-sampling
- Batch normalization [3] is used to avoid poor initialization problems.
- No fully-connected hidden layers
- Better suited for image datasets

(a) Dataset is available at <http://yann.lecun.com/exdb/mnist/>
 (b) Dataset is available at <http://lsun.cs.princeton.edu/2016/>

- 1) Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.
- 2) Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

Ideas in development and perspectives

- Move forward more efficient learning → Better performance with less training examples (one-shot learning, non-gradient optimization methods, ...)
- Define and implement sub-space constrains when learning new data classes to avoid catastrophic forgetting without reusing historical information at all.
- Study the "separability" of Neural Networks activations, depending on data type/class, to introduce activation scheduling inside networks → more efficient learning with less forgetting

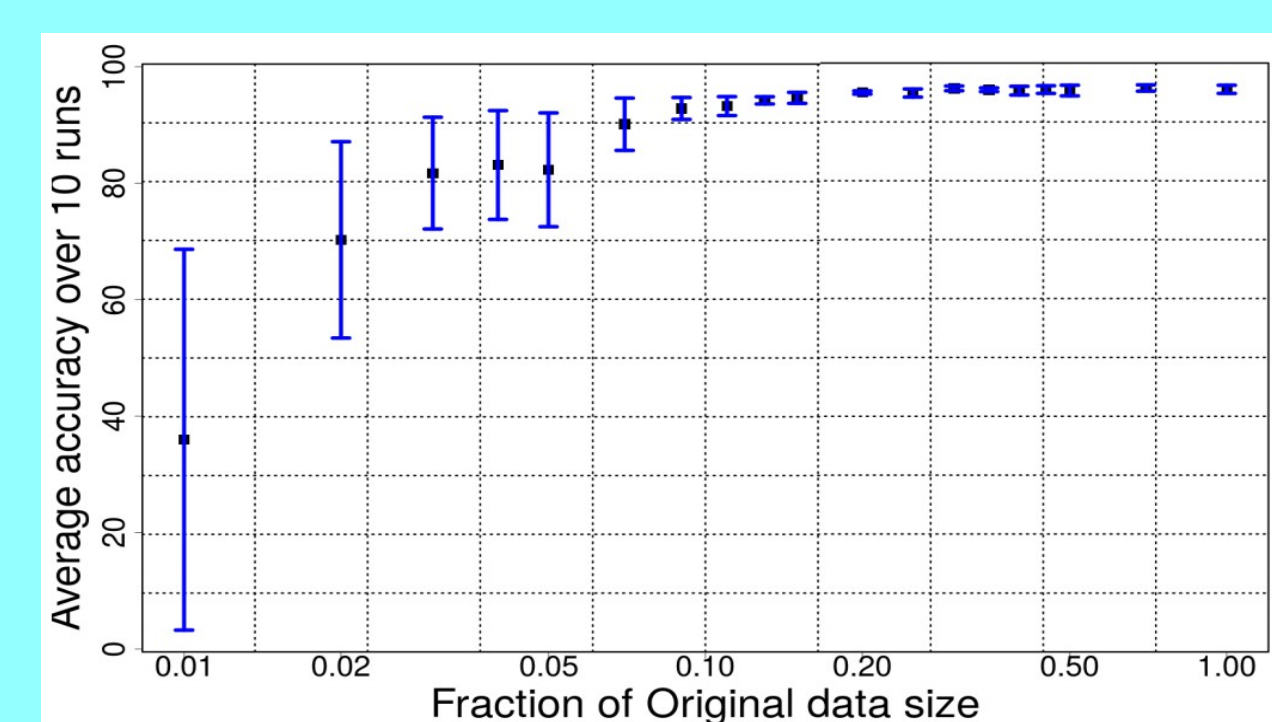
Learning classification models on streams with no storage using GANs

Batch learning on generated data (MNIST dataset)

Goal: To check the ability of generative models to produce data that represent well the initial dataset before passing to the online scenario

Method:

- Train one generator per data class
- Use generators to recreate synthetic dataset of the size, proportional to the initial one
- Train batch classifier on generated dataset
- Test obtained classifier on the original validation set



Results:

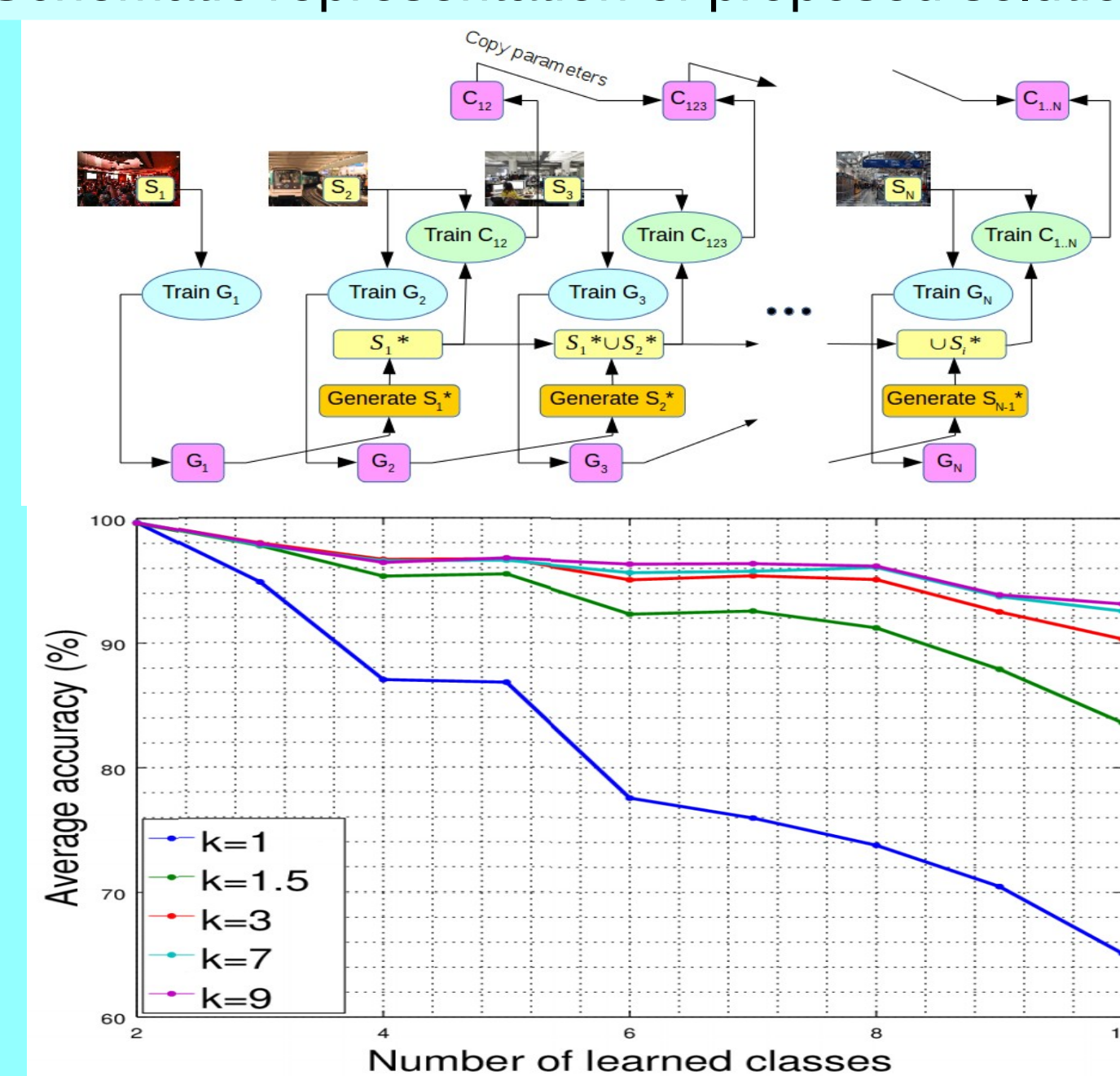
- We get to stable accuracy with the size of generated set of 20% of the initial training set
- Lose >4% in accuracy comparing to the original data usage – the price to pay for not storing the data
- Using more advanced generative models should reduce the loss

Adaptive learning on incremental data stream

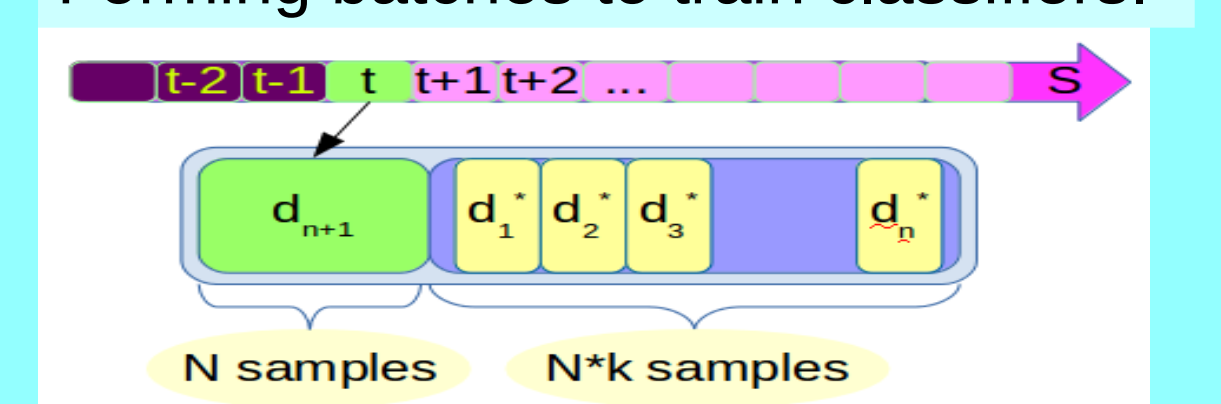
Goal:

- Use generative models in classification scenario on data stream with data arriving incrementally class-by-class to remove the necessity of storing historical data while not introducing forgetting
- Check the influence of the amount of generated data on classification accuracy

Schematic representation of proposed solution:



Forming batches to train classifiers.



- Small values of k provoke forgetting, previously learned classes get crushed
- ~94% classification accuracy with big enough k in completely online settings (99.6% in batch mode with the same net architecture)
- Very big k = a lot of data to generate, can become a problem for the datasets with 100+ classes

Online learning in realistic stream scenario on complex dataset (LSUN)

Data classes are mixed and only partially presented in the stream at every time point

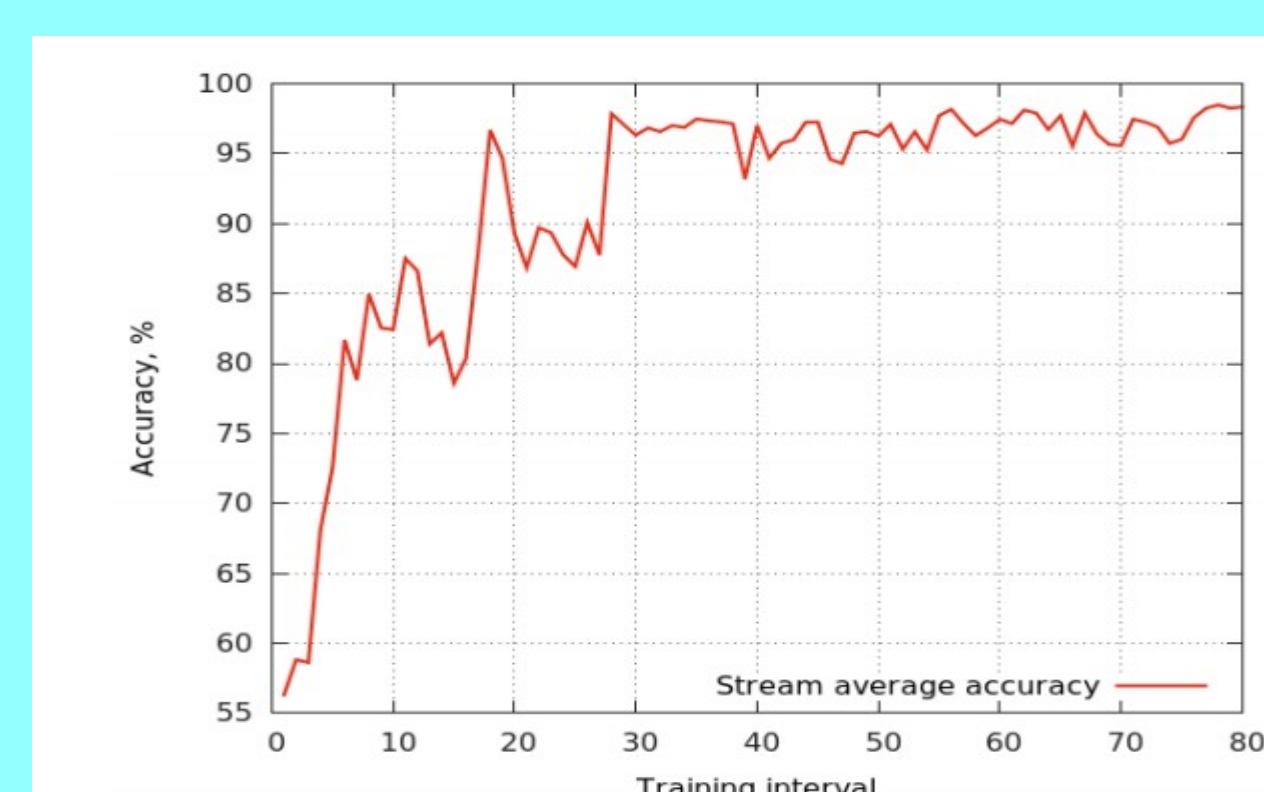
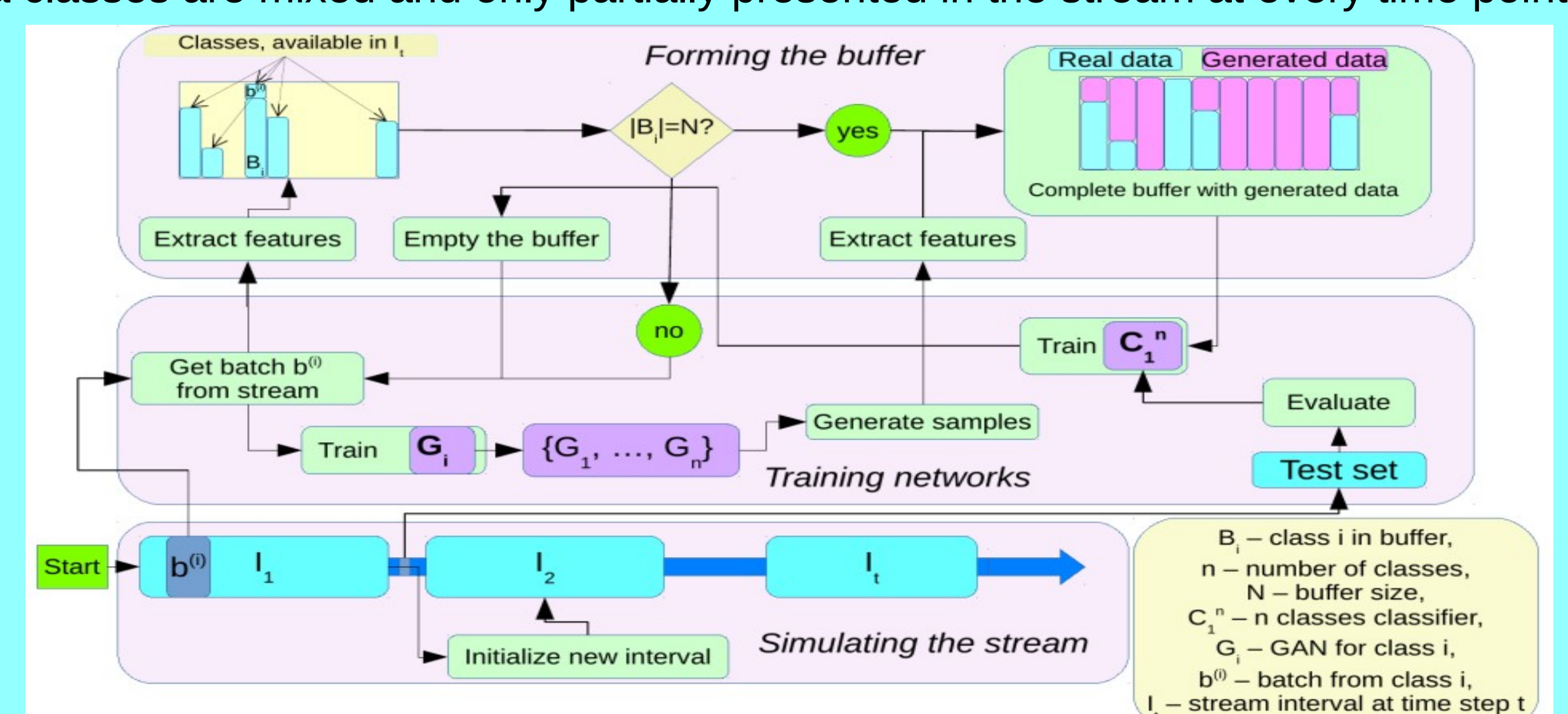


Fig. 3: Classification accuracy during online stream training for MNIST dataset

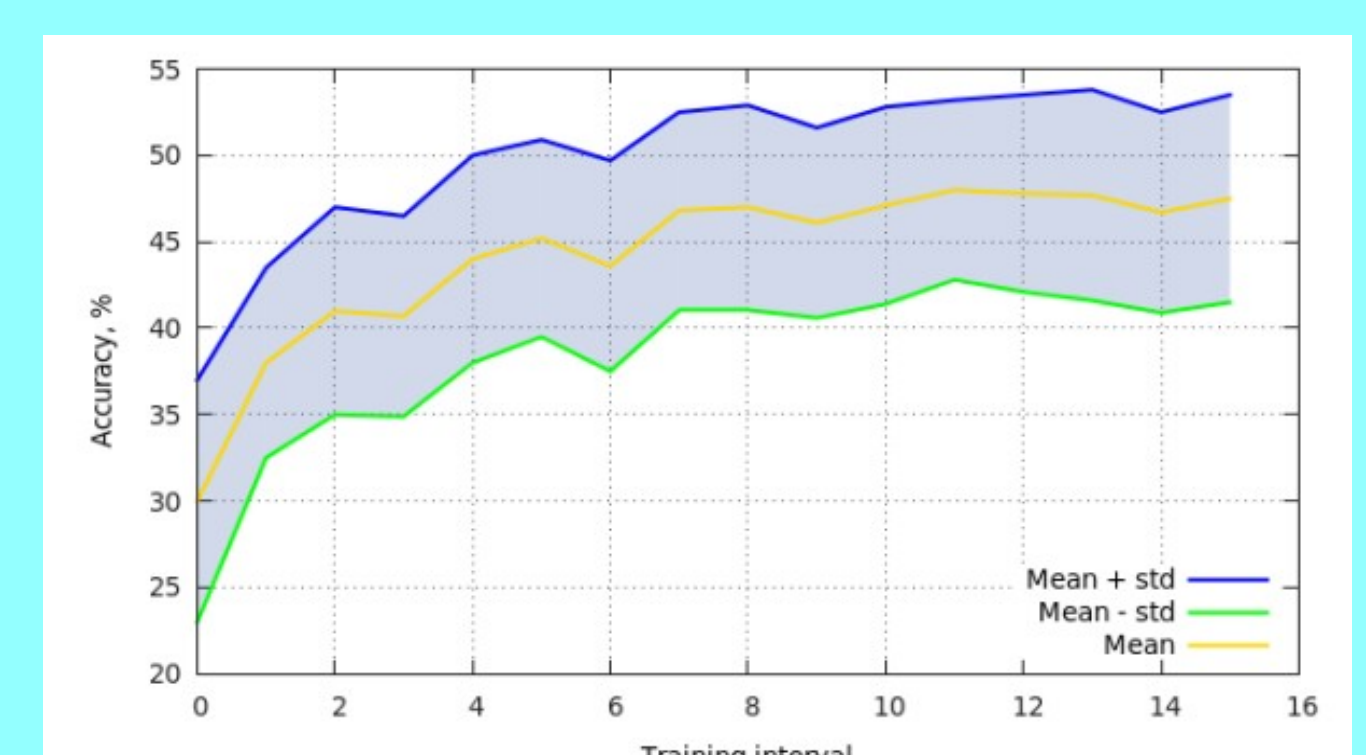


Fig. 4: Classification accuracy during online stream training for LSUN dataset. Each point on the graph corresponds to the average value over 80 training intervals