Bridging the Gap: Enhancing the Utility of Synthetic Data via Post-Processing Techniques

Andrea Lampis
andrea.lampis@mail.polimi.it
Eugenio Lomurno
eugenio.lomurno@polimi.it
Matteo Matteucci
matteo.matteucci@polimi.it

Department of Electronics, Information and Bioengineering
Politecnico di Milano
Via Ponzio 34/5
20133 Milan, Italy

Abstract

Acquiring and annotating suitable datasets for training deep learning models is challenging. This often results in tedious and time-consuming efforts that can hinder research progress. Generative models have emerged as a promising solution for generating synthetic datasets that can replace or augment real-world data. However, the effectiveness of synthetic data is limited by their inability to fully capture the complexity and diversity of real-world data. In this paper, we explore the use of Generative Adversarial Networks to generate synthetic datasets for training classifiers that are subsequently evaluated on real-world images. To improve the quality and diversity of the synthetic dataset, we propose three novel post-processing techniques: Dynamic Sample Filtering, Dynamic Dataset Recycle, and Expansion Trick. In addition, we introduce a pipeline called Gap Filler (GaFi), which applies these techniques in an optimal and coordinated manner to maximise classification accuracy on real-world data. Our experiments show that GaFi reduces the Classification Accuracy Score gap to an error of 2.03%, 1.78%, 3.99%, 3.33% and 2.04% on the Fashion-MNIST, CIFAR-10, CIFAR-100, CINIC-10 and DermaMNIST datasets, respectively. These results represent a new state of the art in Classification Accuracy Score and highlight the effectiveness of post-processing techniques in improving the quality of synthetic datasets.

1 Introduction

Over the last few years, deep generative models have become so powerful that they are able to produce high-quality samples that are almost indistinguishable from the real ones. With these recent developments, it is natural to ask whether these models are powerful enough to generate data that can be effectively used to train a machine learning model to perform a specific downstream task, thus completely replacing the need for real data. This would have several advantages, for example it could significantly reduce the cost and effort of data collection, or it could be helpful in cases where information cannot be shared directly for privacy or sensitivity reasons, or when the original dataset is too large and the generative model can be used as a compressed version of the real data.

© 2023. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

In light of these considerations, although the focus of these models has historically been on the perceptual quality of the data they generate, there have been attempts in recent years to formalise and quantify the usefulness of synthetic data for training. An essential contribution was made by Ravuri *et al*., who pioneered the metric called Classification Accuracy Score (CAS) [1]. Given a system for generating data, the CAS represents the accuracy performance that a classifier trained solely on its generated data is able to achieve on a test set consisting of real data. Surprisingly, despite the high perceptual quality of the data generated by the latest deep learning models, and despite the ability to generate an almost unlimited number of samples, training a model on them has been observed to lead to a lower CAS value than the accuracy of the same model trained on real data.

In this paper, we investigate if and how we can bridge the utility gap between synthetic data generated by generative models and real-world data as measure via the CAS metric. We analyse the post-processing techniques available in the literature and propose new ones to improve synthetic data quality. We then present a new post-processing pipeline, Gap Filler (GaFi), which can be applied to any generative model. GaFi combines the most effective post-processing techniques to achieve a significantly better generator, without the need to modify the model's architecture or learning technique. The contributions of our work are:

- We propose two improved post-processing techniques, namely Dynamic Sample Filtering and Dynamic Dataset Recycle, and a novel method called Expansion Trick.
- We propose the GaFi pipeline, which consists of a set of post-processing techniques suitable for any generative model to maximise the CAS achieved with its generated data.
- We demonstrate the effectiveness of the GaFi pipeline by obtaining empirical CAS results that approach the upper bound of real accuracy performance. This achievement sets a new state of the art in generating synthetic data for classification tasks.

2 Related Works

In the past decade, deep learning has seen a surge in the development of generative models that are capable of producing synthetic data with increasing similarity to real-world training data. Some of the key architectures that have contributed to this progress include Variational Autoencoders (VAEs) [2], Generative Adversarial Networks (GANs) [3], and Denoising Diffusion Probabilistic Models (DDPMs) [4]. These models have been predominantly used in the field of computer vision, particularly for image generation. To measure the perceptual quality of generated images, various metrics have been proposed, of which the Inception Score (IS) [5] and the Frécht Inception Distance (FID) [6] are the most widely used and empirically validated in the field of computer vision and image generation. Both IS and FID are based on the Inception network architecture and use statistical measures to assess the similarity between generated images and a reference dataset [7].

In addition, the creation of synthetic datasets, either to replace or to complement real-world ones, has gained increasing attention in machine learning applications. Synthetic data can offer significant advantages, such as the possibility to generate large-scale datasets with known properties, reducing the need for costly data collection and annotation, and overcoming issues related to data privacy and access. The use of generative models for synthetic data generation has found applications in several fields, including semantic segmentation [8, 9, 10, 11], optical flow estimation [12, 13, 14], human motion understanding [15, 16, 17, 18], and image classification [1, 19, 20, 21].

A fundamental contribution to the advancement of this alternative use of generative models has been made by Ravuri *et al*., which focused on evaluating the performance of GANs through a downstream classifier, introducing the Classification Accuracy Score (CAS) metric [1]. The proposed method involves training a classifier with synthetic images and testing it on a dataset of real images to assess its performance. The challenge is that, if a generative model captures in an optimal way the real data distribution, performance on the downstream task should be similar whether using the original data or synthetic ones. Unfortunately, achieving comparable performance between classifiers trained on real and synthetic data remains a challenge, despite the efforts made to bridge this gap.

One notable approach is the **Sample Filtering** technique proposed by Dat *et al*. [22], which aims to optimise the quality of the data generated. The technique uses an auxiliary classifier trained on real data to predict the labels of synthetic samples. Samples with incorrect predictions or those with low prediction confidence are discarded. In addition, the same authors propose the use of **multiple generative models** to improve the accuracy of synthetic data by better capturing the real data distribution. Another approach to address the accuracy gap is the **Dataset Smoothing** technique proposed by Besnier *et al*. [20]. This technique aims to create a diverse but gradually changing dataset by replacing only a portion of the generated training data with new samples at each epoch. Leveraging these contributions we propose improved and novel methods to reduce the CAS gap further.

3 Method

The present study proposes a comprehensive post-processing pipeline that can be applied to a broad range of generative models with the intention of improving their Classification Accuracy Score (CAS). To this end, we have meticulously surveyed the existing literature to identify the most effective techniques and subsequently adapted them to enhance their dynamicity and flexibility. In addition, we have devised and integrated novel methods into the proposed optimisation pipeline.

3.1 Post-processing Techniques

Dynamic Sample Filtering Following Dat *et al*. findings, we have re-implemented and extended their proposed technique using an adaptive approach that takes into account the dataset and the generative model in use. As demonstrated in their ablation study, varying the filtering threshold may result in a synthetic dataset of superior quality compared to static filtering [22]. To this end, we introduce Dynamic Sample Filtering, a two-step technique. Firstly, we use a classifier to predict the generated samples, thereby discarding all incorrectly classified samples. Secondly, we define a range of filtering thresholds, incrementally increasing from 0 to 0.9, and, for each threshold, we construct a standalone dataset consisting of only the correctly predicted samples with confidence greater or equal to the threshold. We generate new data until the filtered samples reach the desired amount for each synthetic dataset. Finally, we train a classifier for each dataset and determine the threshold value of the one achieving the best CAS. This technique helps eliminate low-quality images that could negatively impact the performance of the downstream classifier.

Dynamic Dataset Recycle Inspired by Besnier *et al*., we propose to extend their Dataset Smoothing technique, which has shown significant benefits in their research. Our approach, called Dynamic Dataset Recycle, differs from the original in that it replaces the entire synthetic dataset in each iteration, rather than just a portion. Our ablation studies show that

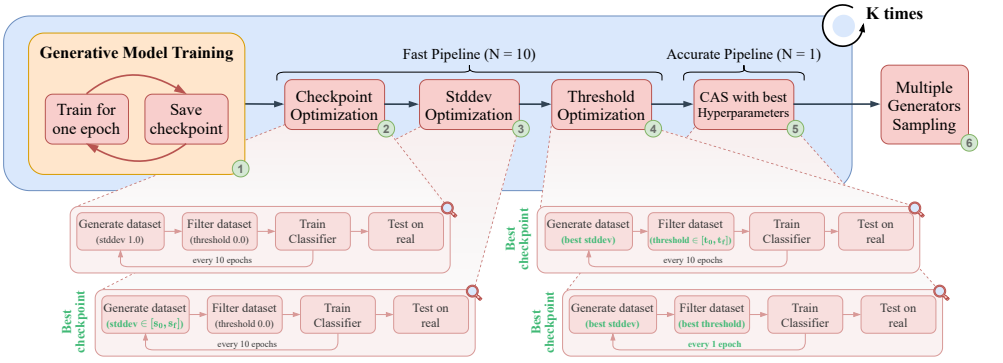

Figure 1: Overview of the Gap Filler (GaFi) pipeline.

recycling the entire dataset leads to better performance in terms of CAS. Furthermore, to address the issue of time complexity, which is proportional to the size of the generated dataset, we propose to generalise its use by recycling the entire dataset every N epochs of classifier training.

Expansion Trick We present a new method, the Expansion Trick, which works in contrast to the Truncation Trick proposed by Brock *et al.* [23]. Instead of truncating the input noise space, we expand it by sampling from a normal distribution with a higher standard deviation than that used in model training. By increasing the diversity of the input noise space, our method encourages the generative model to explore regions that were less sampled during training. This leads to more diverse and novel images, a desirable outcome in settings where diversity is prioritised over visual quality. As expected, the increased standard deviation of the input noise distribution reduces the quality of individual samples. Therefore, the Expansion Trick is most effective when combined with sample filtering techniques, to mitigate the negative effects of lower sample quality by selecting only the most relevant samples to train the classifier.

3.2 Gap Filler Pipeline

In this section, we introduce the Gap Filler (GaFi) pipeline, which combines the post-processing techniques presented in the previous section in an iterative fashion. We discuss also the importance of the correct application order and optimal hyperparameters for the techniques to work effectively and in synergy. The GaFi pipeline, depicted in Figure 1, is composed of the following sequential steps:

1. **Generative Model Training**: the initial step of the pipeline entails training a generative model and saving its checkpoints after every epoch for subsequent use. The specific type of generative model is not a constraint.
2. **Checkpoint Optimization**: in order to optimise the performance of downstream classifiers, it is necessary to choose the best model among the saved checkpoints obtained during training. To accomplish this step, we propose evaluating each checkpoint by computing the CAS, and selecting the one that yields the highest performance. At this stage, we adopt a pipeline with fixed hyperparameters: a standard deviation (*stddev*) of 1.0 and a filtering threshold (*threshold*) of 0.0, meaning only samples predicted as the wrong class are discarded, without considering the confidence of the prediction. To balance the training time of the classifier, we set the Dataset Recycle technique

parameter N to 10, which generates a new synthetic dataset every 10 epochs. This configuration is referred to as the *"Fast Pipeline"*.

3. **Stddev Optimization**: after identifying the best model checkpoint to use, we tune the hyperparameters. We start by determining the optimal standard deviation for the input noise distribution, which corresponds to the best configuration for the Expansion Trick. We achieve this by using the "Fast Pipeline" to calculate the CAS while varying the standard deviation between two predefined values, s_0 and s_f. In our experiments, we set $s_0 = 1.0$ and $s_f = 2.0$, with incremental steps of 0.05.
4. **Threshold Optimization**: once the optimal standard deviation has been determined, the next step is to find the optimal filtering threshold for the Dynamic Sample Filtering technique. We follow a similar approach as before, using the "Fast Pipeline" to compute the CAS while varying the filtering threshold between two specified values, t_0 and t_f. In our experiments, we set $t_0 = 0.0$ and $t_f = 0.9$, with increments of 0.1.
5. **CAS with best hyperparameters**: finally, with the optimal hyperparameters selected so far, we can proceed to train the classifier using the *"Accurate Pipeline"*. Here, the Dynamic Dataset Recycle technique is set to $N = 1$, allowing the use of data with a high degree of diversity to obtain the optimal classifier with respect to the generator under consideration.
6. **Multiple Generators Sampling**: the final step of the GaFi pipeline is to create multiple generative models to sample the data and train a single optimal classifier. This is achieved by repeating all the previous steps of the pipeline K times. According to Dat *et al.* [22], it is sufficient to train multiple identical generative models with different initialisation seeds on the same dataset. In this way, different aspects of the distribution of the dataset can be captured. A synthetic dataset is then created by sampling uniformly from these multiple models for each training epoch of the classifier.

4 Experimental Setup

In order to increase the transparency and reproducibility of the study, this section provides a comprehensive description of the experiments conducted and their setup. We selected the BigGAN Deep architecture [23] as the generative model for our study. To implement this architecture, we adopted the `StudioGAN` library[1] [24], which makes slight modifications to the layer layout of the generator and discriminator residual blocks. Both G and D networks are initialised using the Orthogonal Initialization [25] technique, and trained using the Adam optimizer [26] with hyperparameters $\beta_1 = 0.5$, $\beta_2 = 0.999$, and a constant learning rate of 2×10^{-4}. We also used the Exponential Moving Average (EMA) technique for the weights of G with a decay rate of 0.9999, as recommended by Brock et al. [23]. Data augmentation was limited to random horizontal flipping of the training set. We trained all models using a batch size of 192 and with 3 D steps per G step.

We have chosen to employ the ResNet-20 architecture [27] as the downstream classifier due to its well-established performance. ResNet-20's width was set to 64, and the conventional ResNet training techniques were employed. This includes training it with cross-entropy loss, using a batch size of 128, training for 100 epochs, and using the SGD optimizer with an initial learning rate of 0.1, momentum of 0.9, and weight decay of 1×10^{-4}. The learning rate was reduced by a factor of 10 at epochs 60 and 80. To augment the synthetic

[1] https://github.com/POSTECH-CVLab/PyTorch-StudioGAN/

Table 1: The results of the CAS metric obtained using the Dynamic Sample Filtering technique for each filtering threshold.

	No Filtering	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Fashion-MNIST	88.70%	89.88%	90.01%	89.59%	89.98%	89.89%	90.05%	**90.21%**	89.86%	90.12%	89.90%
CIFAR-10	87.11%	88.45%	88.95%	88.75%	88.45%	88.67%	**89.06%**	88.72%	88.96%	88.09%	88.41%
CIFAR-100	57.74%	59.13%	58.82%	**59.39%**	59.35%	59.20%	59.06%	58.79%	58.76%	57.28%	55.52%
CINIC-10	75.58%	76.70%	77.10%	77.85%	76.83%	78.08%	77.62%	77.80%	77.94%	**78.50%**	77.11%
DermaMNIST	67.48%	67.58%	67.38%	67.08%	66.88%	67.18%	**67.93%**	67.53%	66.48%	67.08%	66.38%

training set, which is a balanced dataset with the same cardinality as the real dataset, a simple form of data augmentation was used. This involves zero-padding the input image, or its horizontally flipped version, to a size of 40×40, extracting a random crop of size 32×32, and using it as the final input image.

All experiments have been conducted on five datasets, namely `Fashion-MNIST` [28], `CIFAR-10` [29], `CIFAR-100` [29], `CINIC-10` [30] and `DermaMNIST` [31]. The `Fashion-MNIST` dataset contains 60,000 28×28 greyscale training images divided into 10 classes. `CIFAR-10` and `CIFAR-100` contain 50,000 32×32 RGB training images divided into 10 and 100 categories respectively. All three datasets have a test set of 10,000 images. `CINIC-10` contains 180,000 32×32 RGB training images divided into 10 categories. Its test images are 90,000. Finally, the `DermaMNIST` dataset contains 8,010 RGB 28×28 training images divided into 7 categories and a test set size of 2,005. To make the images of the `Fashion-MNIST` dataset the same size as the other datasets, we resized them to 32×32 using zero padding. For the `DermaMNIST` dataset, on the other hand, due to its RGB nature, a 32×32 resizing using the Lanczos algorithm was chosen. The experiments have been conducted on a machine equipped with an Intel(R) Xeon(R) Gold 6238R CPU @ 2.20GHz CPU and an Nvidia Quadro RTX 6000 GPU. Training a single ResNet-20 model takes between 1 and 2.5 hours, depending on which and how many post-processing techniques are used, while training a BigGAN Deep requires around 48 hours.

5 Results

In this section, we present the results of our experiments. We start by analysing the individual impact of each proposed post-processing technique. The results of these techniques are evaluated based on the final checkpoint of the generative model, i.e. without the application of the Checkpoint Optimization step. Finally, we present the results of the whole GaFi pipeline.

Dynamic Sample Filtering As we can see from Table 1, the use of the Sample Filtering technique is beneficial for all the datasets in analysis. However, from this table it can be seen that the optimal threshold value is highly dependant on the specific dataset. For instance, the CAS for `Fashion-MNIST` and `CIFAR-10` remains almost constant for any threshold value, while for the `CIFAR-100` it is clearly visible that a higher threshold value degrades the performance of the classifier. We assume that this behaviour is due to the fact that the generators trained on the first two datasets, being easier to learn, produce images that are very faithful to the original dataset. Therefore, the classifiers pretrained on real images will have high confidence in their predictions and most of the bad images will already be removed due to incorrect labelling. In contrast, `CIFAR-100` is a much more complex dataset as it has 10 times the number of classes, causing the generated images to be more likely rejected by the pretrained classifier when a high filtering threshold is used. On average, with respect to

Table 2: The results of the CAS metric obtained using the Dynamic Dataset Recycle technique for each considered recycle frequency.

	No Recycle	N=10	N=5	N=1
Fashion-MNIST	88.70%	89.29%	89.88%	**90.16%**
CIFAR-10	87.11%	89.72%	90.25%	**90.42%**
CIFAR-100	57.74%	59.68%	60.57%	**61.38%**
CINIC-10	75.58%	79.37%	80.55%	**82.57%**
DermaMNIST	67.48%	68.03%	68.33%	**68.43%**

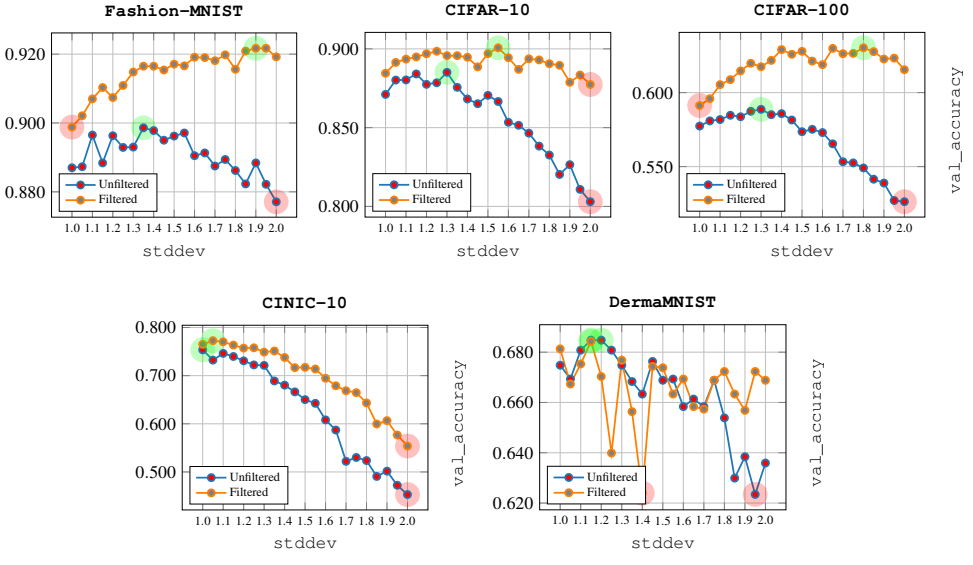

Figure 2: The results of the CAS metric obtained using the Expansion Trick technique. The plots compare unfiltered and filtered datasets (filtering threshold: 0.0).

the baseline CAS achieved, i.e. the "No Filtering" column of the table, the Dynamic Sample Filtering technique improves the CAS by 1.7%.

Dynamic Dataset Recycle Table 2 shows that the proposed dataset recycling technique significantly improves the CAS for all five datasets. The results reveal that even with a relatively soft recycling period, such as $N = 10$, there is an increase in accuracy ranging from 0.55% to 3.79%, depending on the dataset. Notably, by reducing the recycling period, i.e. generating new synthetic data more frequently during training, we can obtain an additional performance boost. The gain in accuracy is more pronounced with increasing dataset complexity, as expected, since the generative model may require more attempts before generating meaningful data, especially for those classes learnt with poor effectiveness.

Expansion Trick Figure 2 displays the impact of the Expansion Trick on the CAS. The results indicate that when the dataset is unfiltered, there is a small increase in performance when using a standard deviation slightly higher than 1, with the exception of `CINIC-10`. However, as the standard deviation increases, there is a degradation in performance. This outcome is not unexpected, as a higher standard deviation leads to more diverse images at

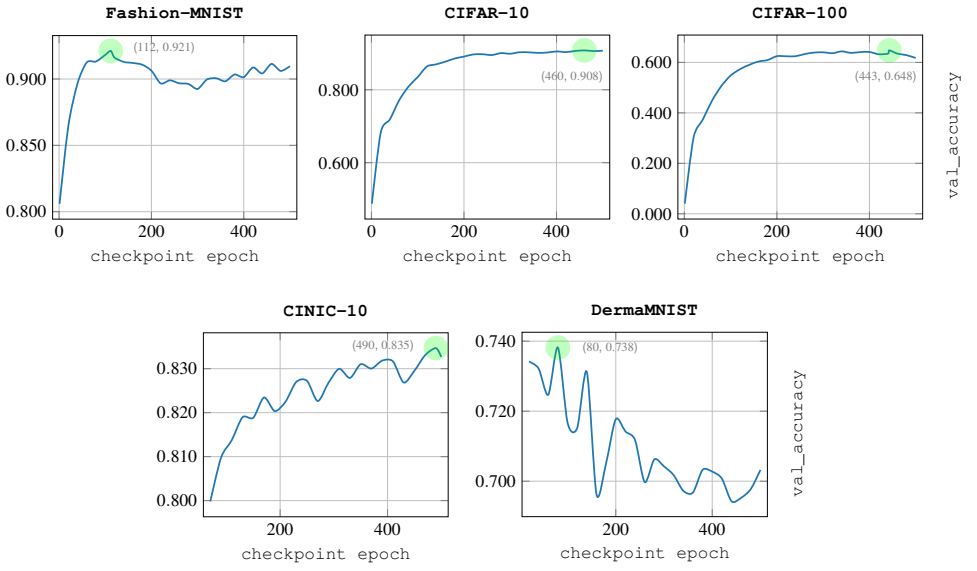

Figure 3: The results of the CAS metric obtained for each checkpoint and dataset.

the cost of image quality. As a result, beyond a certain point, the images become too degraded to be useful for training a downstream classifier. On the other hand, when using the Expansion Trick with a sample filtering technique, we can achieve significantly higher standard deviation values without compromising performance. This is because only the "good" images that meet the filtering criteria - in this case, the correctly classified ones - are kept, ensuring that the dataset is more diverse while still containing higher-quality images than the unfiltered dataset, leading to a higher classification accuracy. Our novel technique improves the CAS by 3.47%, 2.96%, 5.29%, 1.69% and 0.95% on the `Fashion-MNIST`, `CIFAR-10`, `CIFAR-100`, `CINIC-10` and `DermaMNIST` datasets, respectively. These gains demonstrate the efficacy of the Expansion Trick in enhancing the generative model's ability to produce informative samples, which in turn improves the downstream classifier's performance.

Checkpoint Optimization Figure 3 shows the evolution of the CAS metric as a function of the generative model checkpoint. Bearing in mind that each point corresponds to the complete training of a ResNet-20 classifier, the aim of this step is to identify the best checkpoint with respect to the CAS metric, even though it may be computationally expensive. Looking at the graphs, the best epochs for the datasets `Fashion-MINST` and `DermaMINST` are 112 and 80 respectively. On the other hand, for the datasets `CIFAR-10`, `CIFAR-100` and `CINIC-10` we have that the optimum was reached at epochs 460, 443 and 490 respectively. The first observation is that CAS increases with the number of epochs up to a certain point. This behaviour is consistent with that of GANs from the point of view of the perceptual quality of the generated images, which tend to collapse after a certain number of training iterations. The optimal point clearly varies with the dataset and its complexity. Given the same image size of the considered datasets, this complexity is to be understood in terms of the number of classes, image channels and the cardinality of the datasets themselves. This assertion is confirmed by the CAS convergence points in the first quarter of the available

Table 3: The optimal hyperparameters configuration and CAS performance obtained using the Accurate Pipeline.

	Checkpoint	Standard Deviation	Filtering Threshold	CAS
Fashion-MNIST	112	2.00	0.0	**94.03%**
CIFAR-10	460	1.60	0.3	**92.60%**
CIFAR-100	443	1.70	0.1	**68.92%**
CINIC-10	490	1.25	0.0	**84.37%**
DermaMNIST	80	1.30	0.4	**73.66%**

Table 4: The final results comparing the CAS obtained from the classifiers trained on generated data. The GaFi pipeline is compared with the previous state of the art, with the Synthetic Baseline and with the accuracy of the classifiers trained on real data.

#Generators		Fashion-MNIST	CIFAR-10	CIFAR-100	CINIC-10	DermaMNIST
	Real Data	96.01%	94.98%	75.64%	89.05%	77.25%
	Baseline	88.70%	87.11%	57.74%	75.58%	67.48%
1	Dat *et al.*	-	88.25%	62.22%	-	-
	GaFi (ours)	94.03%	92.60% (+4.35%)	68.92% (+6.70%)	84.37%	73.66%
2	Dat *et al.*	-	89.68%	64.33%	-	-
	GaFi (ours)	93.98%	92.74% (+3.06%)	70.22% (+5.89%)	85.42%	75.06%
4	Dat *et al.*	-	90.68%	67.22%	-	-
	GaFi (ours)	93.99%	93.02% (+2.34%)	71.75% (+4.53%)	85.62%	74.71%
6	Dat *et al.*	-	91.14%	67.56%	-	-
	GaFi (ours)	93.98%	93.20% (+2.06%)	71.95% (+4.39%)	85.72%	75.21%

epochs for the simplest datasets and in the last quarter for the others. Overall, the Checkpoint Optimisation step is critical and allows the subsequent steps in the GaFi pipeline to start from the optimal generative model.

Accurate Pipeline After determining the optimal configuration, which is summarised in Table 3, the classifier can be retrained using the "Accurate Pipeline", where the recycling period N is set to 1, regenerating the dataset at each training epoch to achieve the best possible CAS through the GaFi pipeline. It is evident that the Expansion Trick in combination with the Dynamic Sample Filtering technique played a crucial role in achieving the optimal CAS. This is supported by the shifted values towards the standard deviation of 2 in each of the configurations compared to the Expansion Trick application alone.

The final results of the experiments are presented in Table 4, which includes the accuracy scores obtained from real data (*Real Data*), the CAS of a single BigGAN Deep without post-processing techniques representing the baseline (*Baseline*), and for each number of generators considered, a comparison between our GaFi technique and the previous state-of-the-art post-processing performed by Dat *et al.* using the same generative architecture used in this work. Our approach achieves the best results, with improvements over the baseline of 5.33% for `Fashion-MNIST`, 6.09% for `CIFAR-10`, 14.21% for `CIFAR-100`, 10.14% for `CINIC-10` and 7.73% for `DermaMNIST`. Furthermore, our pipeline achieves higher accuracy even when using only one generator compared to the best configuration of Dat *et al.* with six generators. These results demonstrate that our proposed post-processing techniques, and the way they are applied in the GaFi pipeline, lead to superior classifiers trained

on more generalised and useful data.

It is worth noting that the gap between our synthetic data and the real data has narrowed significantly. Specifically, for the `Fashion-MNIST`, `CIFAR-10`, `CIFAR-100`, `CINIC-10` and `DermaMNIST` datasets, the gap with respect to the baseline has been reduced from 7.31%, 7.87% , 17.9%, 13.47% and 9.77% to 2.03%, 1.78%, 3.99%, 3.33% and 2.04%, respectively. This remarkable result demonstrates the undeniable effectiveness of post training techniques. Moreover, it implies that the use of other generative models, whether existing or forthcoming, can further reduce this gap, and it may even be possible to achieve classifiers trained on synthetic data that outperform those trained on real data. This promising prospect illustrates the great potential of our proposed approach for synthesising high-quality data.

6 Conclusion

In this study, we introduced the Gap Filler pipeline (GaFi) to enhance the Classification Accuracy Score (CAS) by proposing new and enhanced post-processing techniques for generative models. These techniques included Dynamic Sample Filtering, Dynamic Dataset Recycle, and Expansion Trick, which have been shown to be highly beneficial when applied correctly. Our experimental results demonstrated that the proposed pipeline significantly increased the CAS, resulting in a new state-of-the-art performance on the five datasets analysed. Despite our research yielding an accuracy that was slightly lower than that obtained on real data, we believe that the remaining gap raises a philosophical question about the very essence of generative modeling: whether it is possible to produce a model that can perfectly learn the distribution of real data. However, we remain optimistic that it is achievable. We acknowledge that there are challenges that need to be addressed to bridge this gap, but we are confident that once this is achieved, it would open up new avenues for research and revolutionize several fields.

Acknowledgements

This project has been supported by AI-SPRINT: AI in Secure Privacy-pReserving computINg conTinuum (European Union H2020 grant agreement No. 101016577) and FAIR: Future Artificial Intelligence Research (NextGenerationEU, PNRR-PE-AI scheme, M4C2, investment 1.3, line on Artificial Intelligence).

References

[1] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *Advances in neural information processing systems*, 32, 2019.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[5] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[8] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

[9] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21330–21340, 2022.

[10] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8300–8311, 2021.

[11] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3752–3761, 2018.

[12] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick Van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015.

[13] Yo-whan Kim, Samarth Mishra, SouYoung Jin, Rameswar Panda, Hilde Kuehne, Leonid Karlinsky, Venkatesh Saligrama, Kate Saenko, Aude Oliva, and Rogerio Feris. How transferable are video representations based on synthetic data? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 35710–35723. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/e8507db80464ced5658d16b49bd458b9-Paper-Datasets_and_Benchmarks.pdf.

[14] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10093–10102, 2021.

[15] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 109–117, 2017.

[16] Jianxin Ma, Shuai Bai, and Chang Zhou. Pretrained diffusion models for unified human motion synthesis. *arXiv preprint arXiv:2212.02837*, 2022.

[17] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *UIST '11 Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, October 2011. URL https://www.microsoft.com/en-us/research/publication/kinectfusion-real-time-3d-reconstruction-and-interaction-using-a-moving-depth-camera/.

[18] Xi Guo, Wei Wu, Dongliang Wang, Jing Su, Haisheng Su, Weihao Gan, Jian Huang, and Qin Yang. Learning video representations of human motion from synthetic data. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20165–20175, 2022. doi: 10.1109/CVPR52688.2022.01956.

[19] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *Proceedings of the European conference on computer vision (ECCV)*, pages 213–229, 2018.

[20] Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. *arXiv preprint arXiv:1911.02888*, 2019.

[21] Eugenio Lomurno, Alberto Archetti, Lorenzo Cazzella, Stefano Samele, Leonardo Di Perna, and Matteo Matteucci. Sgde: Secure generative data exchange for cross-silo federated learning. *arXiv preprint arXiv:2109.12062*, 2021.

[22] Pham Thanh Dat, Anuvabh Dutt, Denis Pellerin, and Georges Quénot. Classifier training from a generative model. In *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2019. doi: 10.1109/CBMI.2019.8877479.

[23] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[24] MinGuk Kang, Joonghyuk Shin, and Jaesik Park. StudioGAN: A Taxonomy and Benchmark of GANs for Image Synthesis. *2206.09479 (arXiv)*, 2022.

[25] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Toronto University press*, 2009.

[30] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[31] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.