Supplementary Material
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

GAN Architecture

We employed the BigGAN Deep architecture as our generative model [1]. Specifically, we utilized the implementation from the *StudioGAN* library[1], which introduces slight modifications to the layout of residual blocks in both the generator and discriminator [2]. In the generator's *G* block, rather than dropping channels, we upsample the residual path and apply a *Conv1x1* operation to ensure consistent channel numbers between the residual and non-residual paths. Similarly, in the discriminator's *D* block, we first pass the residual path through a *Conv1x1* layer to obtain the correct channel output and then downsample it. For a visual comparison of the original BigGAN Deep blocks and the StudioGAN implementation, refer to Figure 1.

All models underwent training for 500 epochs, employing a batch size of 192 and utilizing 3 discriminator (*D*) steps per generator (*G*) step. While our interpretation of "N *D* steps per *G* step" differs slightly from its original formulation, our experimental findings have shown that it leads to superior results. Specifically, instead of dividing the batch (size 192) into N sub-batches (size 64) and training the discriminator on each sub-batch, we perform N iterations of discriminator training on the complete batch (size 192) (refer to Figure 2). We conjecture that this modification may yield improved outcomes on datasets smaller than ImageNet, which is why we refrain from adopting the larger batch sizes (2048) proposed by the authors of BigGAN, who trained the model on ImageNet. Lastly, our training set only employed random horizontal flips as the sole form of data augmentation.

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

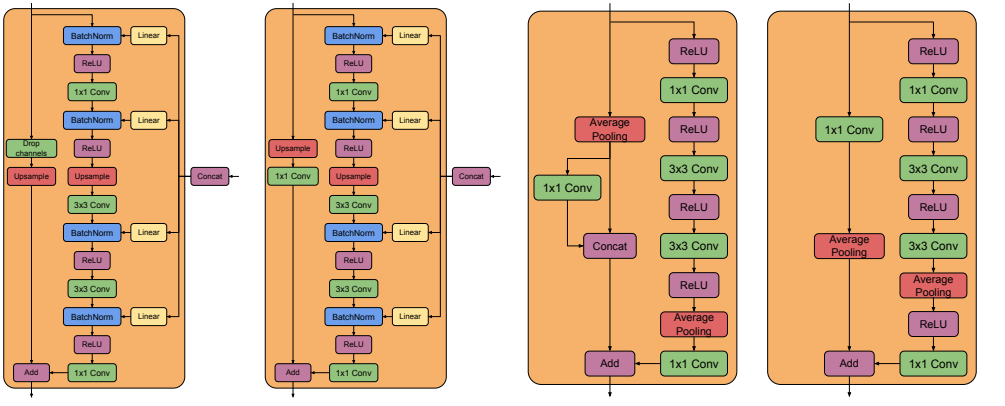

Figure 1: BigGAN Deep blocks architectures. From left to right, the legacy Generator block, the StudioGAN Generator block (used in our work), the legacy Discriminator block, the StudioGAN Discriminator block (used in our work).

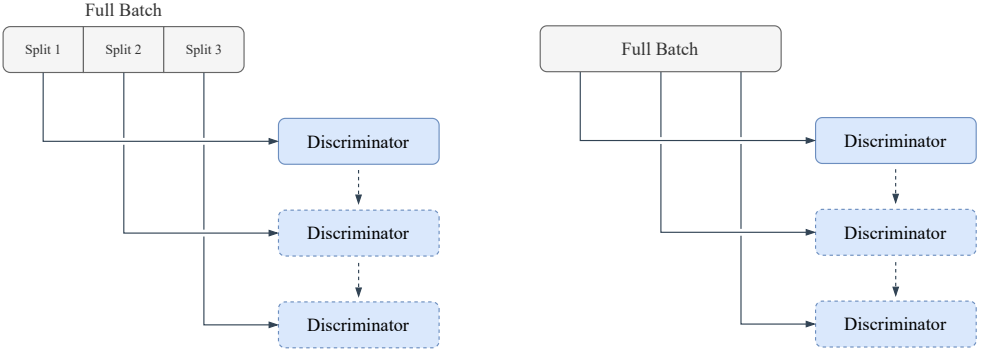

Figure 2: Discrimination Steps - Legacy Vs Ours.

Dynamic Sample Filtering

The relationship between the filtering threshold of the Dynamic Sample Filtering technique and the discarded image count, leading to a balanced dataset with the same cardinality of the real training set, is illustrated in Figures 3, 4, 5, 6, and 7. Notably, the results demonstrate that for the `Fashion-MNIST` [3] and `CIFAR-10` [2] datasets, the discarded image count remains relatively stable until a high threshold value is surpassed, while for `CIFAR-100` [2], it starts to increase exponentially even at relatively low threshold values. The paper's findings emphasize the significance of the Dynamic Sample Filtering technique in enhancing the Classification Accuracy Score (CAS) [5]. However, it is crucial to cautiously validate the threshold value to avoid performance degradation. In cases where the optimal value of this parameter has not yet been determined, we recommend utilizing a value of 0.0.

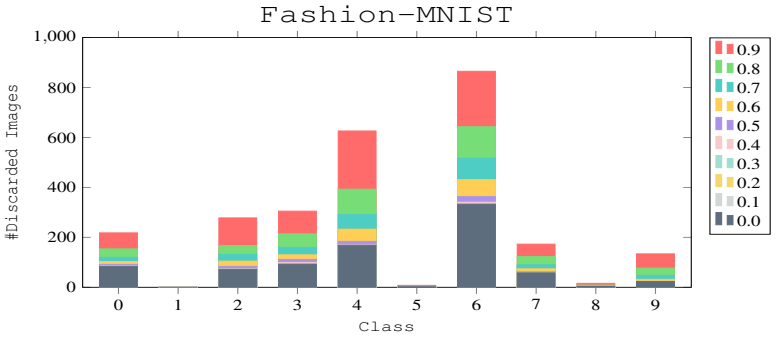

Figure 3: The relationship between the filtering threshold and the number of images discarded for the Fashion-MNIST dataset.

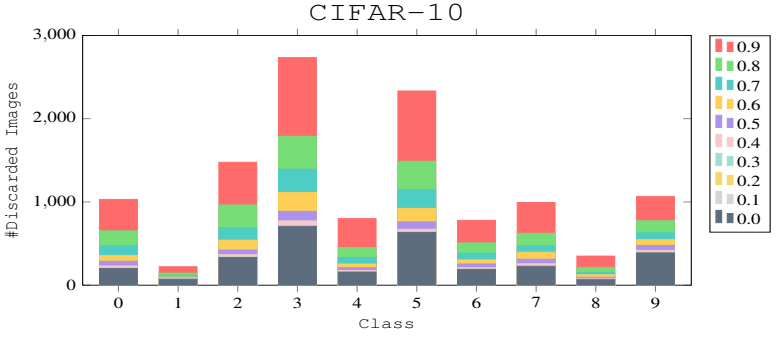

Figure 4: The relationship between the filtering threshold and the number of images discarded for the CIFAR-10 dataset.

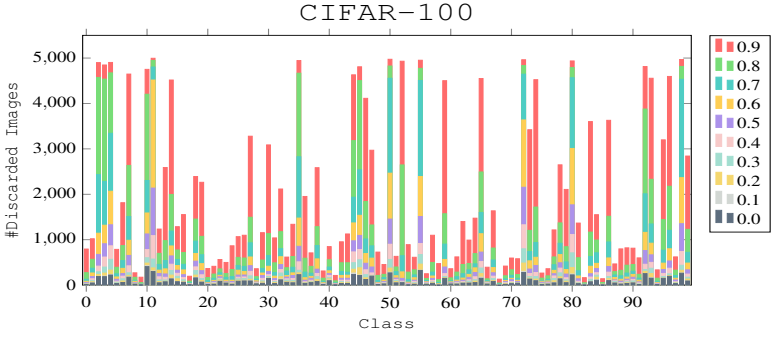

Figure 5: The relationship between the filtering threshold and the number of images discarded for the CIFAR-100 dataset.

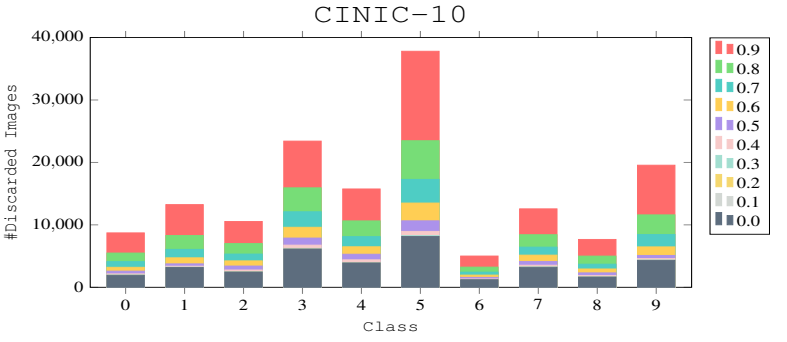

Figure 6: The relationship between the filtering threshold and the number of images discarded for the CINIC-10 dataset.

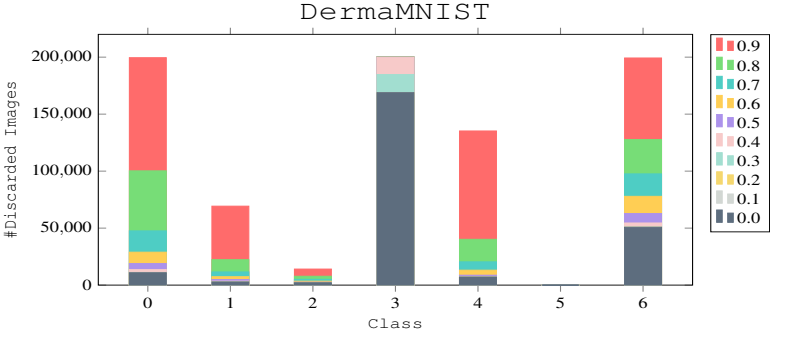

Figure 7: The relationship between the filtering threshold and the number of images discarded for the DermaMNIST dataset.

Expansion Trick

The Expansion Trick involves expanding the input noise space instead of truncating it. This is achieved by sampling from a normal distribution with a higher standard deviation than that used during model training. By broadening the diversity of the input noise space, our approach encourages the generative model to explore underrepresented regions encountered less frequently during training. Consequently, it facilitates the generation of more diverse and novel images, a desirable outcome in scenarios prioritizing diversity over visual fidelity. However, as anticipated, the increased standard deviation of the input noise distribution adversely impacts the quality of individual samples, as shown in Figure 8. Hence, the effectiveness of the Expansion Trick is enhanced when employed in conjunction with sample filtering techniques. This helps mitigate the negative effects of reduced sample quality by selecting only the most pertinent samples for training the classifier.

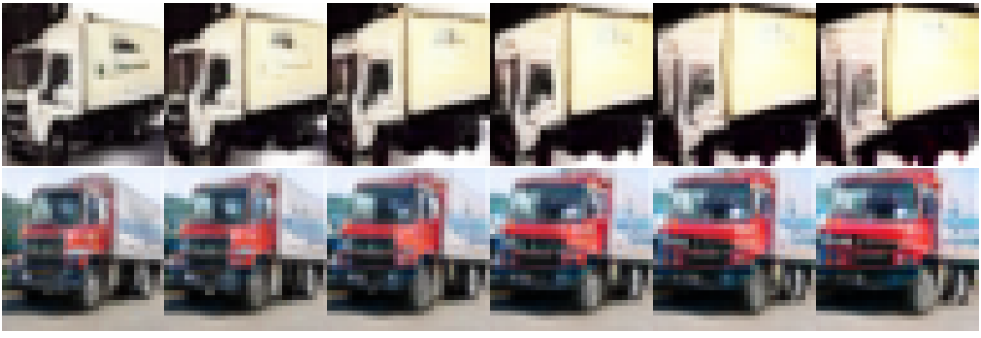

Figure 8: Images sampled for class label “Truck”, with standard deviation ranging from 1.0 to 2.0 with increments of 0.2 (fixed seed). Top: image for which a higher stddev degrades the quality, so it will most likely be filtered. Bottom: image for which a higher stddev increases diversity without reducing quality.

Evaluation Metrics

We investigated the potential correlation between the Classification Accuracy Score (CAS) and commonly utilized evaluation metrics for assessing generative models. Our analysis includes the CAS trends in relation to each specific metric, with "training checkpoints" referring to EMA versions of the generator saved at specific epochs. The CAS was compared against the Inception Score (IS) [6], Fréchet Inception Distance (FID) [7], and Kernel Inception Distance (KID) [8]. However, as evidenced in Figures 9 and 10, no apparent correlation was found between these metrics and the CAS. This lack of correlation can be attributed to the limitations of traditional metrics in capturing all facets of sample quality relevant to classification tasks. For instance, a sample with a low FID score may still be classified incorrectly, indicating its limited usefulness for downstream applications. Likewise, the CAS may not fully encompass the diversity and richness of generated samples that are significant for other objectives, such as artistic image synthesis.

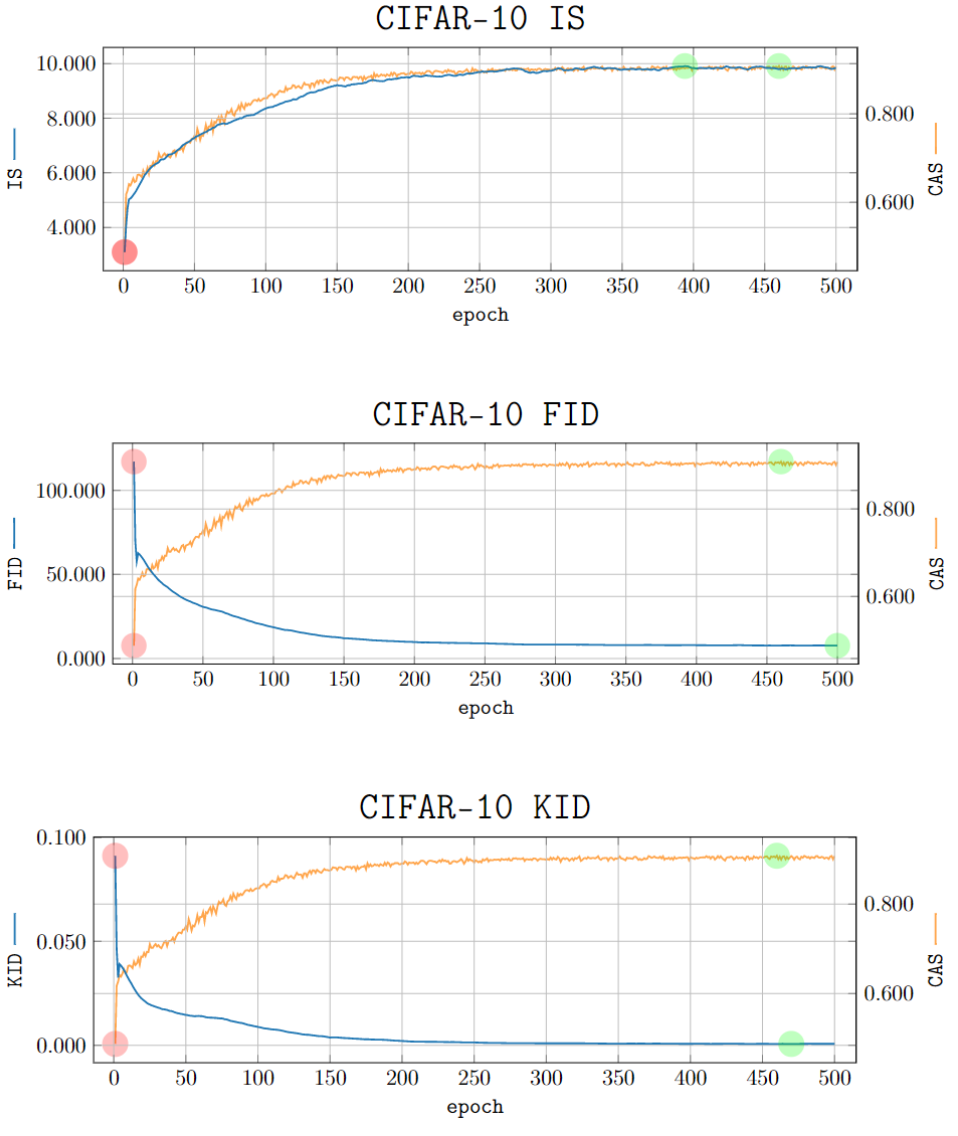

Figure 9: The comparison of the Classification Accuracy Score (CAS) with Inception Score (IS), Fréchet Inception Distance (FID), and Kernel Inception Distance (KID) for each checkpoint (CIFAR-10 dataset).

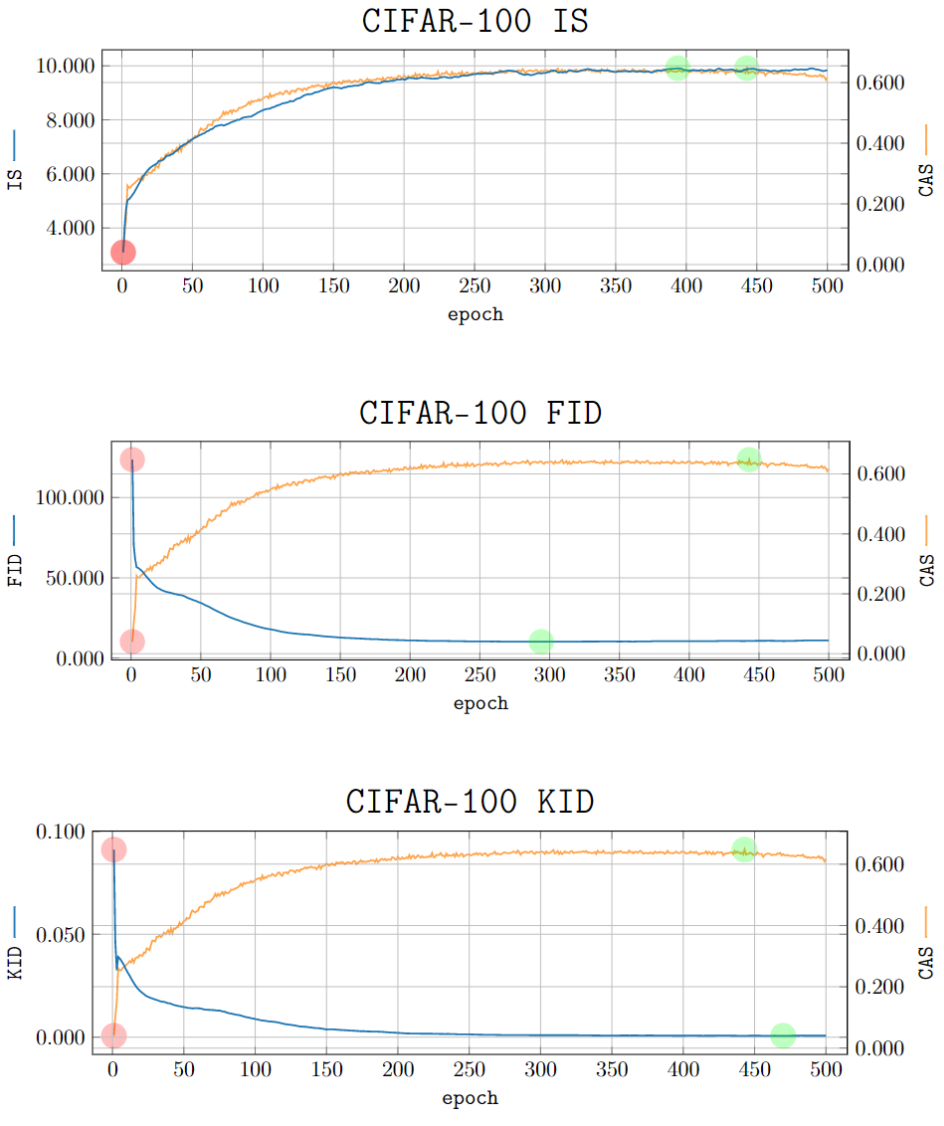

Figure 10: The comparison of the Classification Accuracy Score (CAS) with Inception Score (IS), Fréchet Inception Distance (FID), and Kernel Inception Distance (KID) for each checkpoint (CIFAR-100 dataset).

Datasets t-SNE Embeddings

Figures 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 present t-SNE embeddings of all classes for the `Fashion-MNIST`, `CIFAR-10`, `CIFAR-100` (10 classes), `CINIC-10` [9] and `DermaMNIST` [10] datasets. These t-SNE visualizations are generated using the embeddings of these images in the feature space of the baseline classifier, specifically the ResNet-20 employed in all our experiments. The feature space of a CNN classifier refers to the output of the network's convolutional component. For our experiments utilizing ResNet-20, the feature space corresponds to the output of the last AveragePooling2D layer prior to the fully connected layers. This layer produces a tensor comprising high-level features learned by the network from the input images. These features serve as a condensed representation of the input images, capturing the most relevant patterns and structures for the classification task. By leveraging the feature space as the foundation for t-SNE embeddings, we visualize the distribution of images in a lower-dimensional space that reflects the similarities and dissimilarities among their high-level features. This approach allows us to gain insights into how the classifier clusters images from different classes and assess the quality of the learned features in terms of their discriminative power.

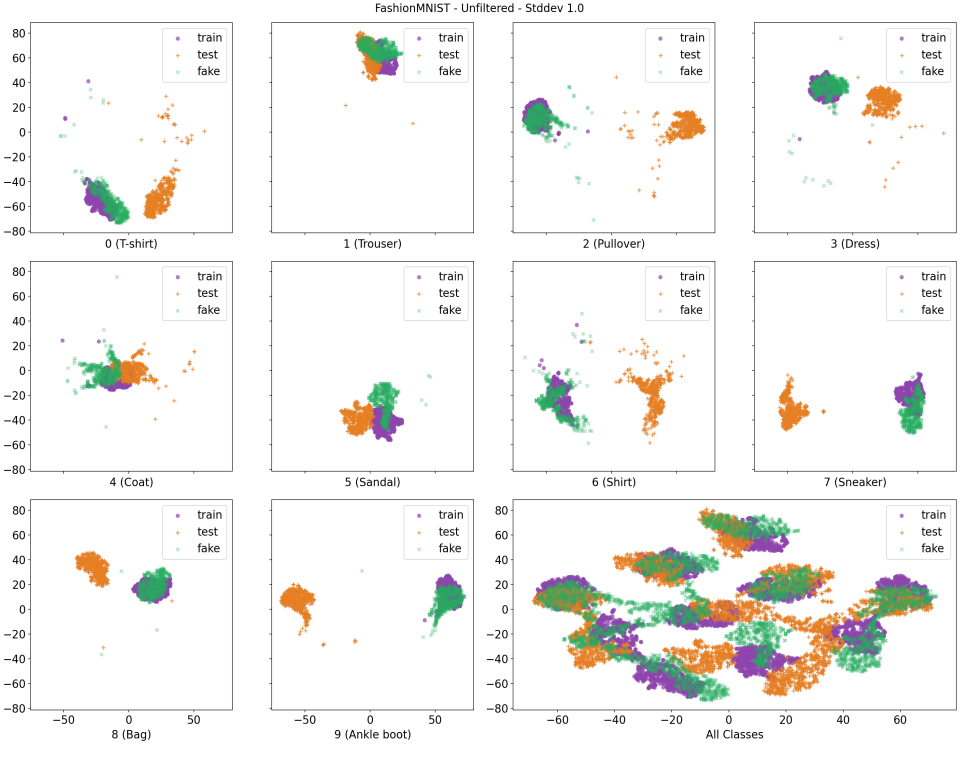

Figure 11: t-SNE embedding of images from *Fashion-MNIST* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. No post-processing techniques have been applied.

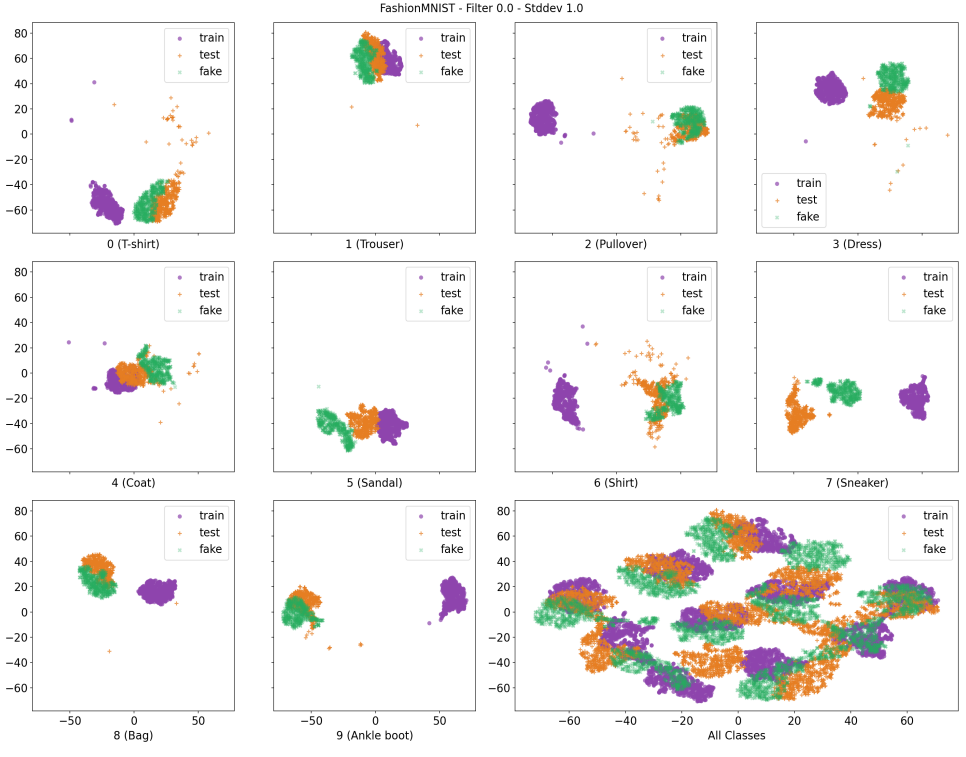

Figure 12: t-SNE embedding of images from *Fashion-MNIST* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. Dynamic Sample Filtering has been applied with a threshold of 0.0.

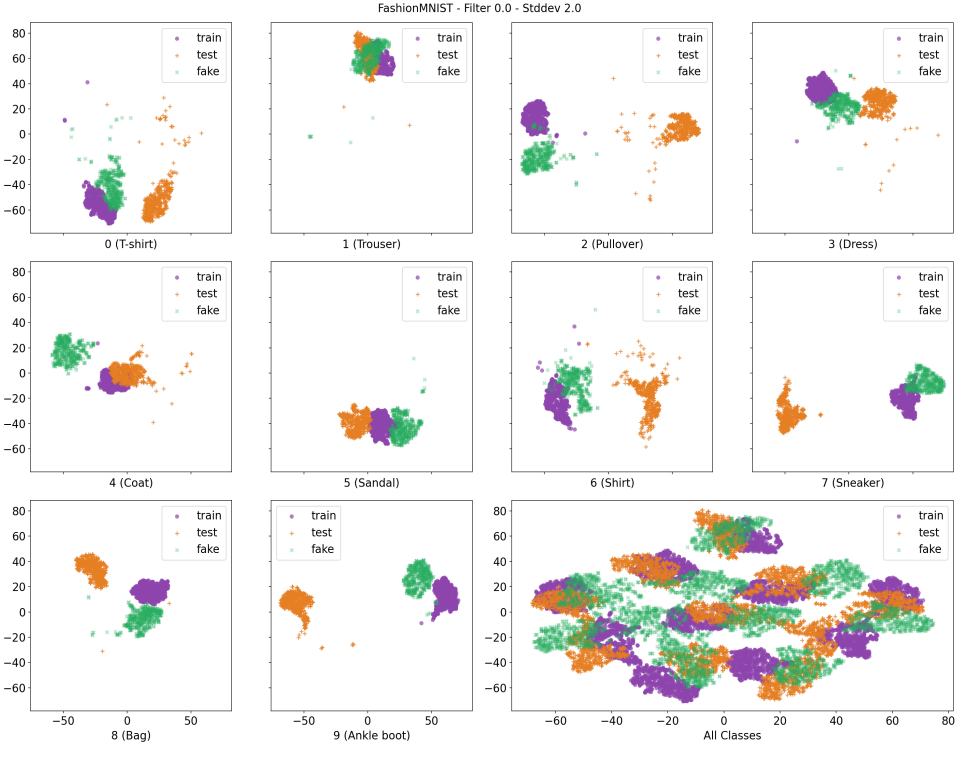

Figure 13: t-SNE embedding of images from *Fashion-MNIST* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. Dynamic Sample Filtering and Expansion Trick have been applied with the optimal hyperparameters.

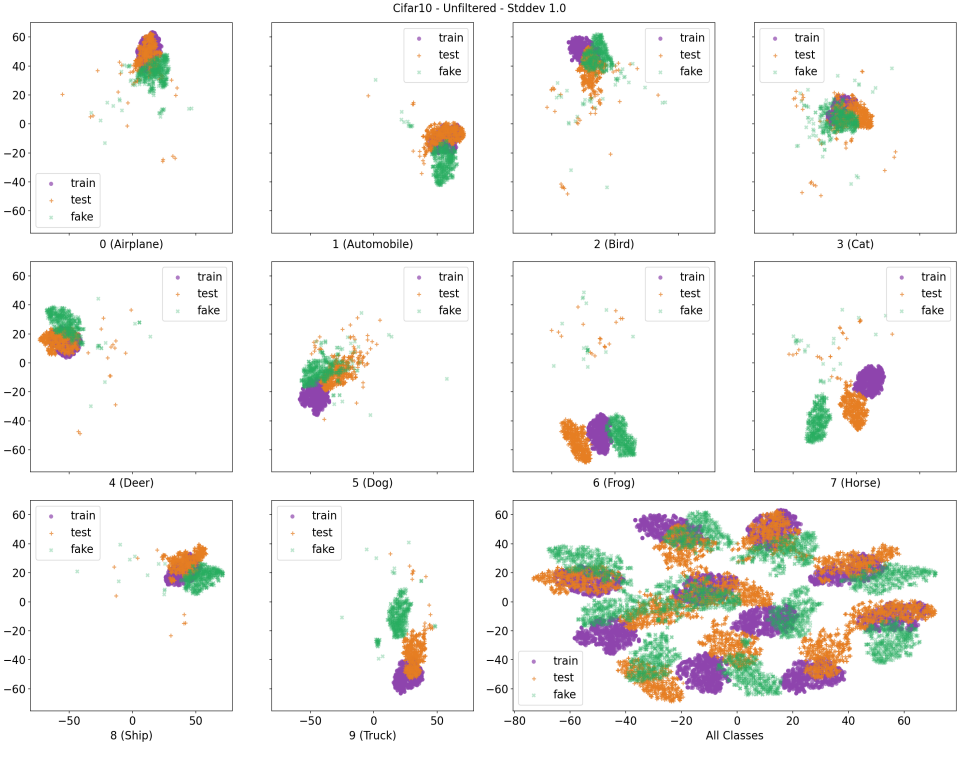

Figure 14: t-SNE embedding of images from *CIFAR-10* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. No post-processing techniques have been applied.

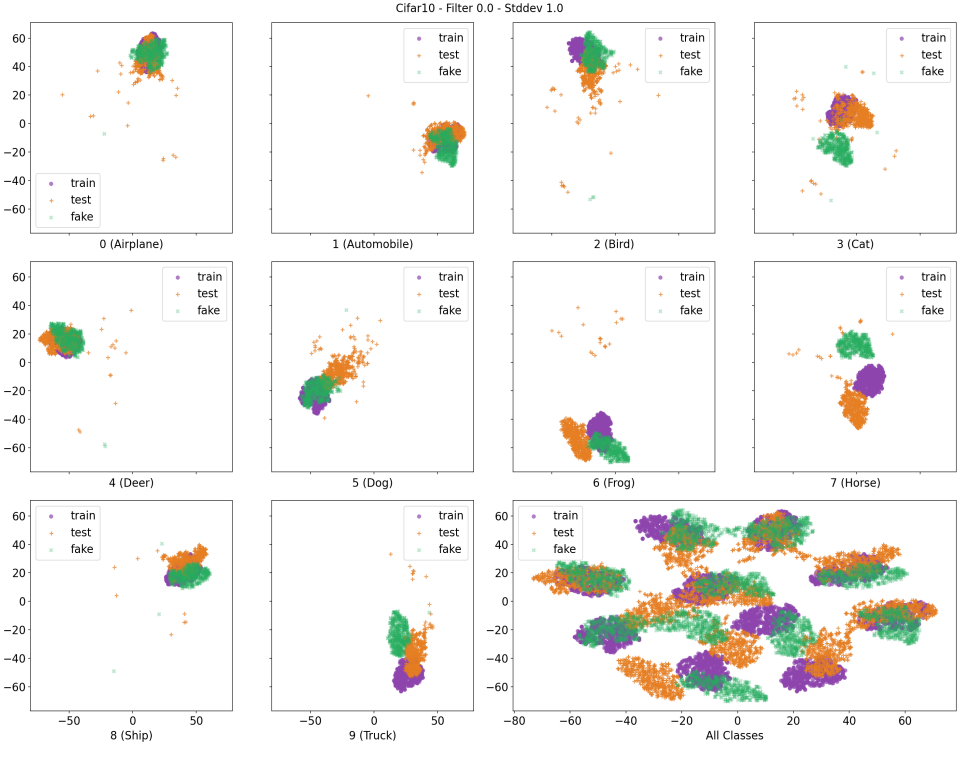

Figure 15: t-SNE embedding of images from *CIFAR-10* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. Dynamic Sample Filtering has been applied with a threshold of 0.0.

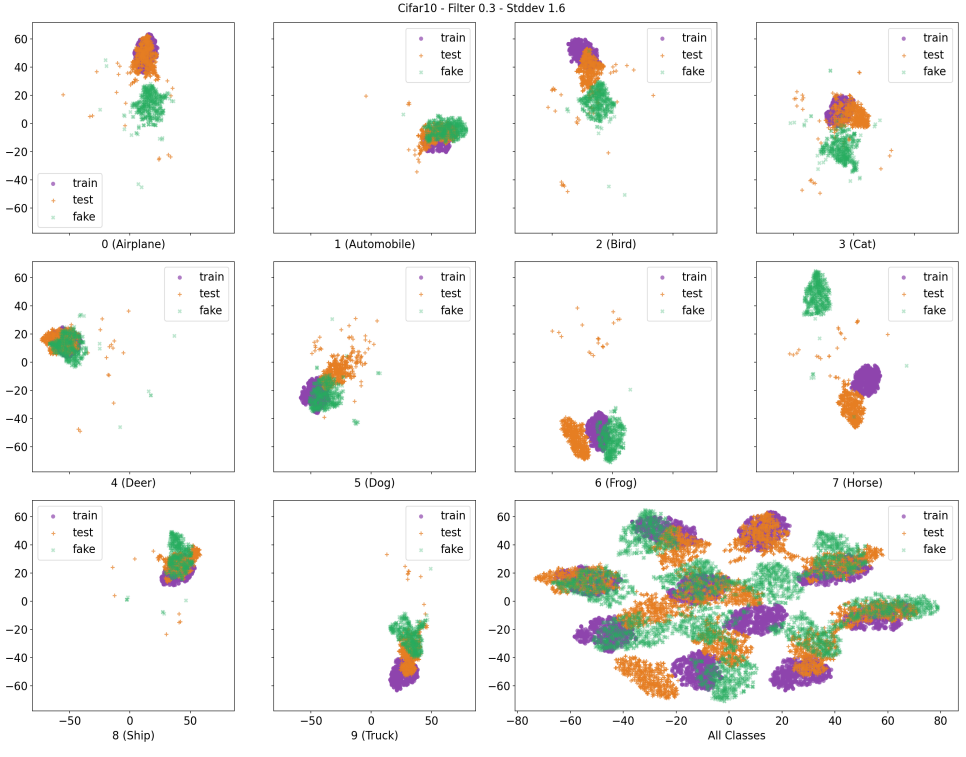

Figure 16: t-SNE embedding of images from *CIFAR-10* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. Dynamic Sample Filtering and Expansion Trick have been applied with the optimal hyperparameters.

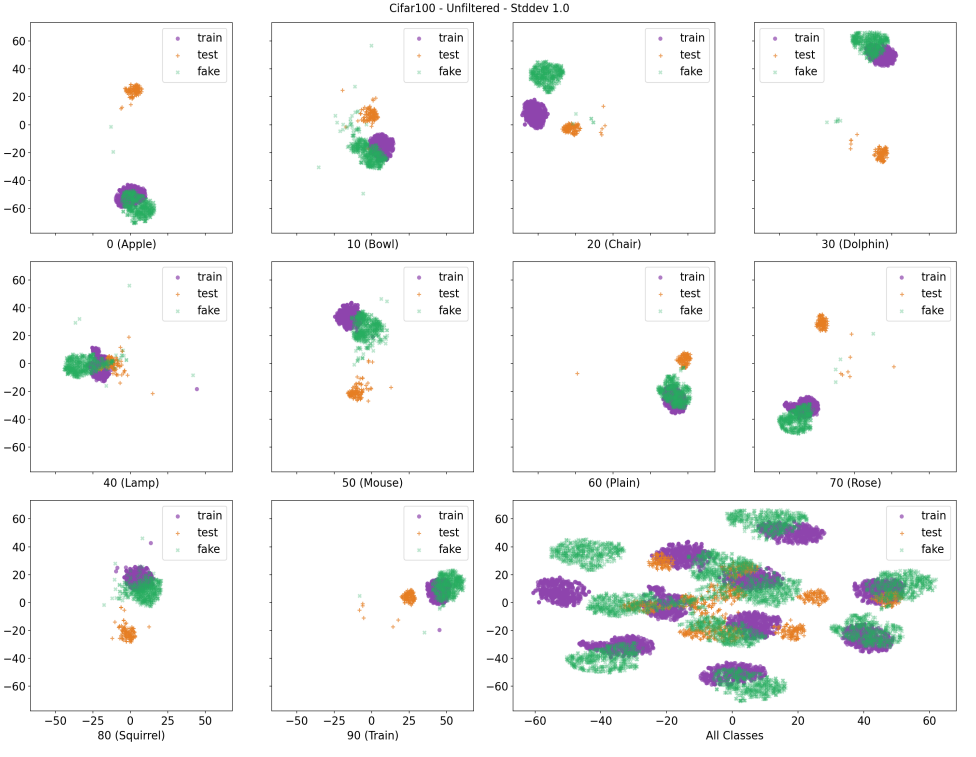

Figure 17: t-SNE embedding of images from *CIFAR-100* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 and 100 from the train and test sets respectively, along with 500 images each generated with BigGAN. No post-processing techniques have been applied.

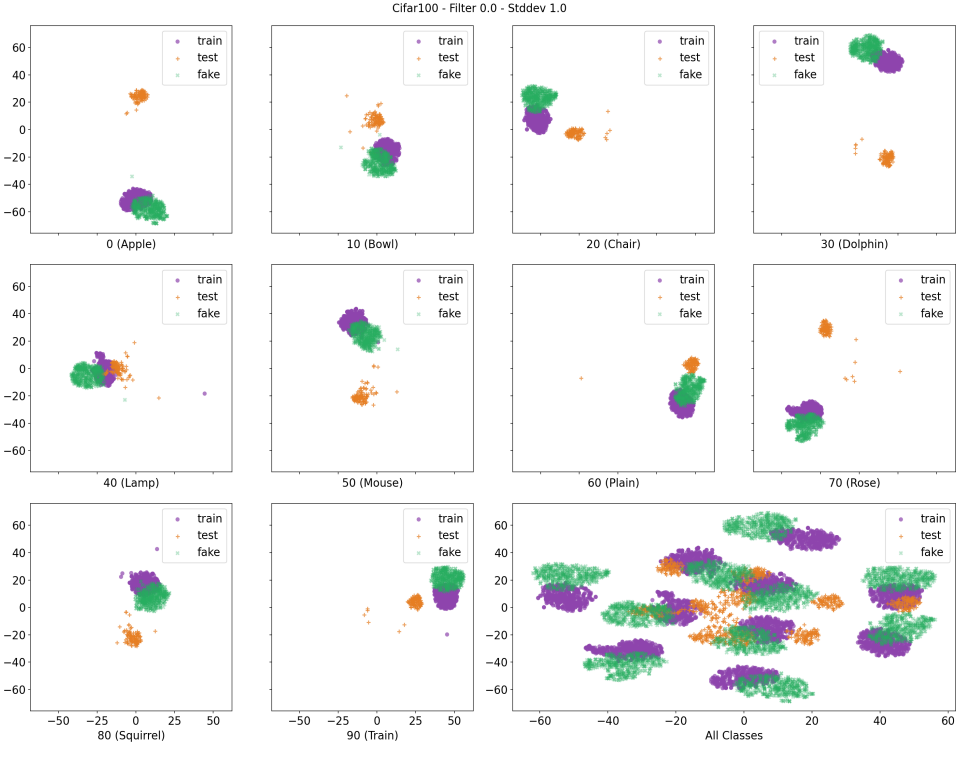

Figure 18: t-SNE embedding of images from *CIFAR-100* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 and 100 images from the train and test sets respectively, along with 500 images each generated with BigGAN. Dynamic Sample Filtering has been applied with a threshold of 0.0.

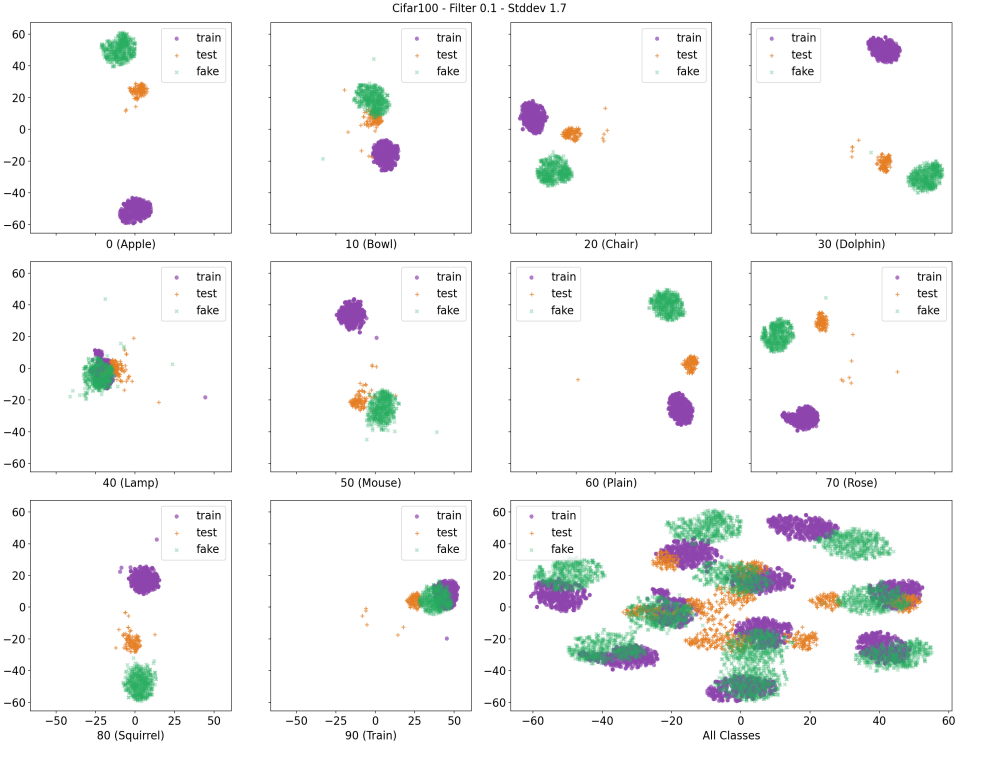

Figure 19: t-SNE embedding of images from *CIFAR-100* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 and 100 images from the train and test sets respectively, along with 500 images each generated with BigGAN. Dynamic Sample Filtering and Expansion Trick have been applied with the optimal hyperparameters.

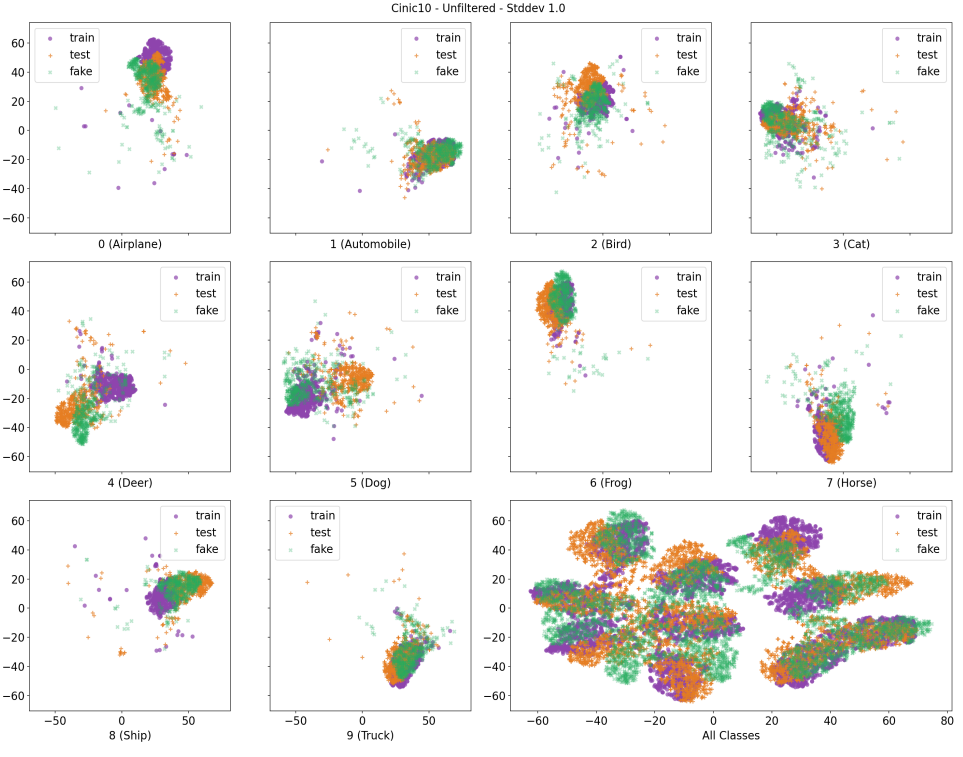

Figure 20: t-SNE embedding of images from *CINIC-10* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. No post-processing techniques have been applied.

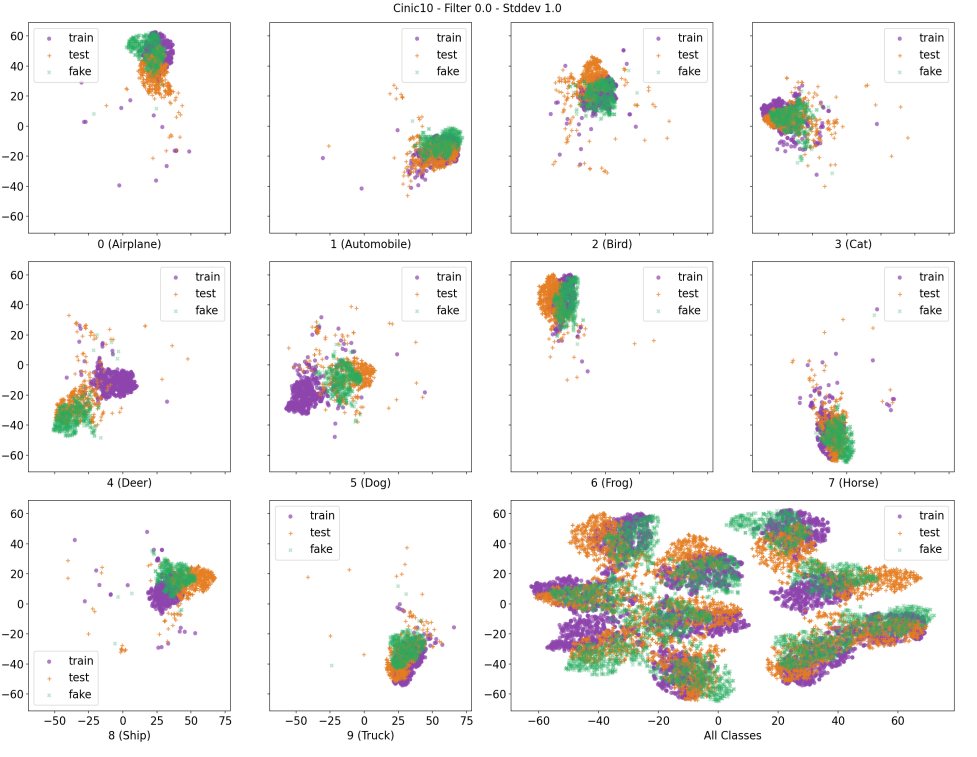

Figure 21: t-SNE embedding of images from *CINIC-10* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. Dynamic Sample Filtering has been applied with a threshold of 0.0.

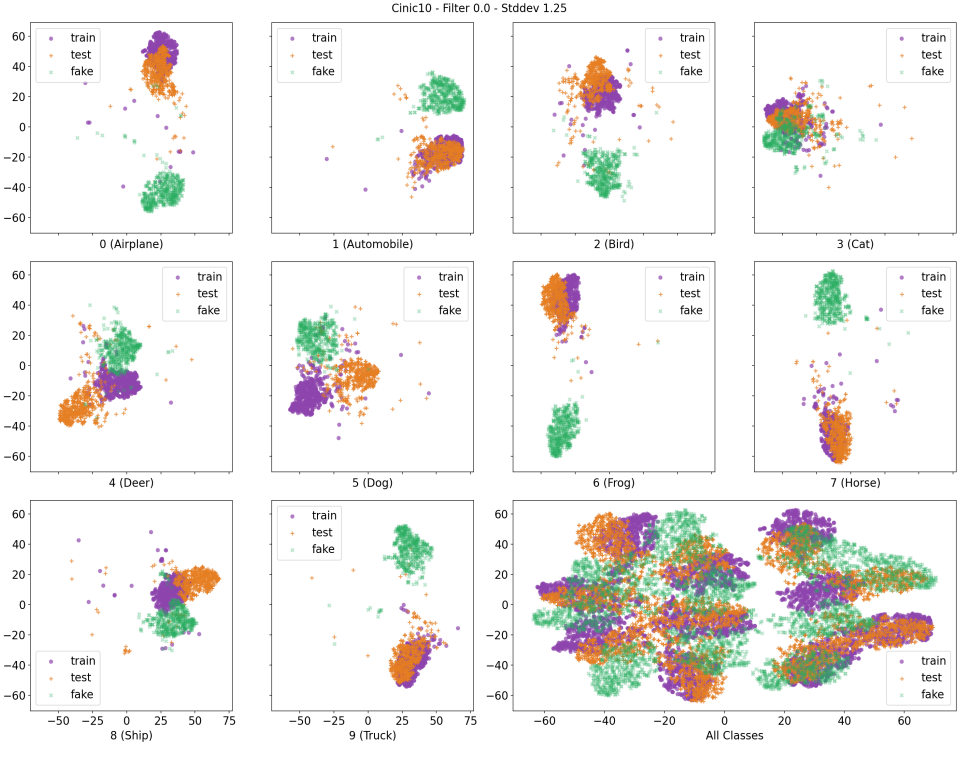

Figure 22: t-SNE embedding of images from *CINIC-10* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use 500 images from both the train and test sets, along with 500 images each generated with BigGAN. Dynamic Sample Filtering and Expansion Trick have been applied with the optimal hyperparameters.

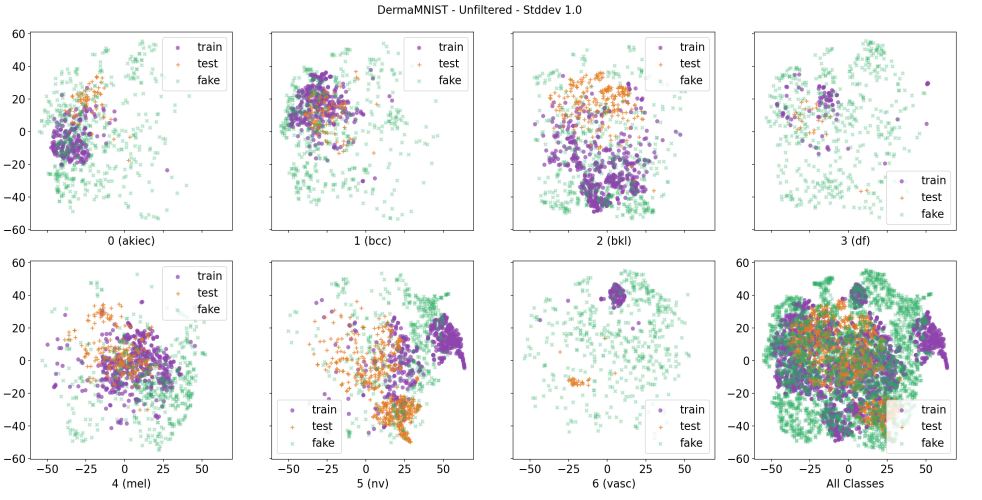

Figure 23: t-SNE embedding of images from *DermaMNIST* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use at most 500 and 100 images from the train and test sets respectively, along with 500 images each generated with BigGAN. No post-processing techniques have been applied.

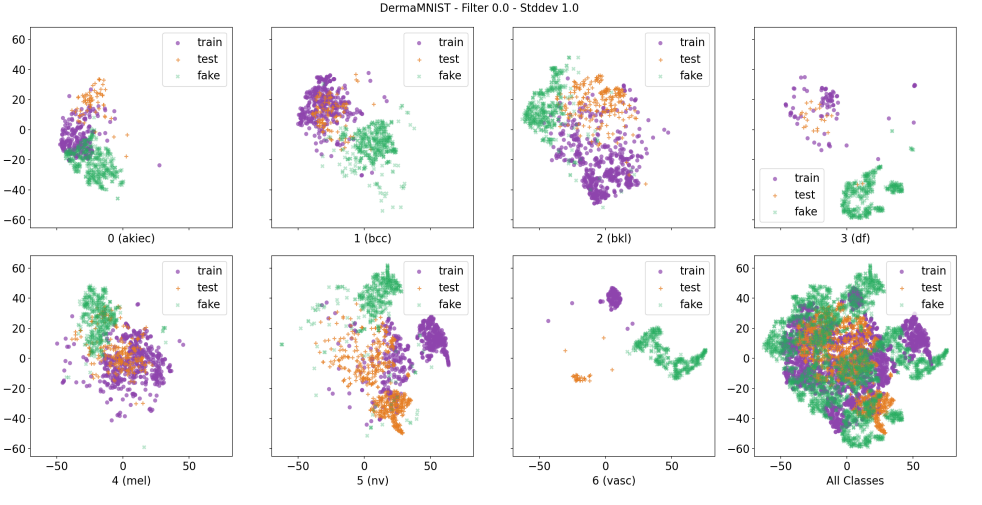

Figure 24: t-SNE embedding of images from *DermaMNIST* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use at most 500 and 100 images from the train and test sets respectively, along with 500 images each generated with BigGAN. Dynamic Sample Filtering has been applied with a threshold of 0.0.

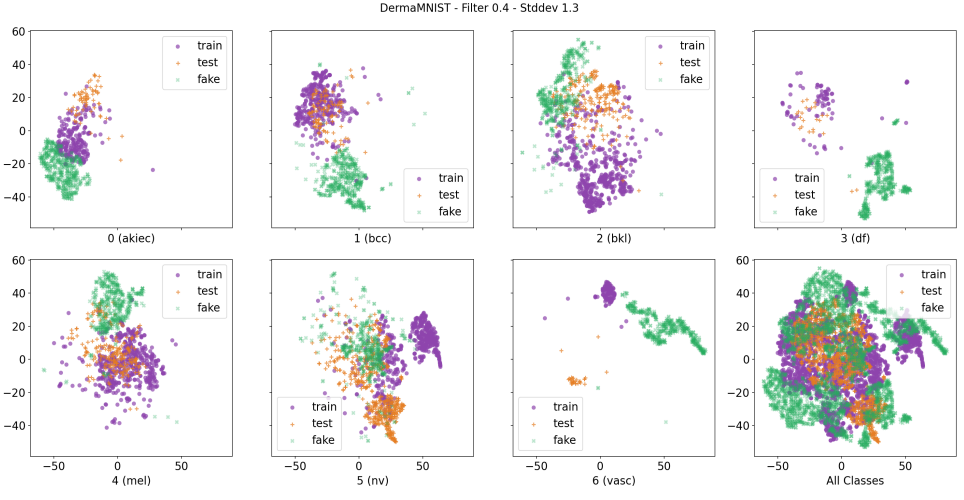

Figure 25: t-SNE embedding of images from *DermaMNIST* classes, embedded in the feature space of the baseline ResNet-20 classifier. We use at most 500 and 100 images from the train and test sets respectively, along with 500 images each generated with BigGAN. Dynamic Sample Filtering and Expansion Trick have been applied with the optimal hyperparameters.

References

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

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

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

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

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

[6] 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.

[7] 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.

[8] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[9] 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.

[10] 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.