Collaborative Sampling for Image Inpainting

Thevie Mortiniera, Master in Data Sciences

École Polytechnique Fédérale de Lausanne

Dated: January 20, 2020

***Abstract*—Semantic Image Inpainting is a challenging task where large missing regions have to be filled with realistic content, not necessarily the ground truth, based on the available visual data. While many sophisticated algorithm-based software were previously used to replace lost or corrupted part of the data, deep generative models like GANs have proven their ability to produce highly realistic images. Yet, their training remain quite unstable, impacting significantly the quality of the generated images. In this work, we use the recently proposed Collaborative Sampling[1] scheme to determine its usefulness in improving the quality of the inpainting results obtained from a DCGAN model. Experimental results on the FASHION-MNIST dataset [2] showed the potential of this method to able to effectively improve qualitatively the generated outputs, proposing more realistic inpainted results.**

I. Introduction

Semantic image inpainting is a challenging process of restorative conservation where damaged, deteriorated, or missing parts of an artwork are reconstructed with realistic content but not necessarily the ground-truth, which itself may be one of many possibilities.
Inpainting originally find its roots in physical artwork, such as painting and sculpture where traditional inpainting is performed by a trained art conservator. At the end of the 20th century, the process evolved to digital media to include images or video interpolation by the use of computer software that relies on sophisticated algorithms to replace lost or corrupted parts of the image data.
Nevertheless, image being one of the most common forms of information, editing an image without any traces poses a problem to the public trust and confidence, which has led to an increase in the demand for automated tools to detect and extract the real image among other ones.

With the exponential spread of data available to everyone and the increasing popularity of deep learning, recent models now achieve high pixel quality outperforming traditional state-of-the-art softwares. Recent works focused on GANs which are a powerful class of deep generative models for realistic image generation. However, although several works attempted to improve GAN training, it still remains a open area of research.

In this work, we focus on the method proposed by Yeh et al. for semantic image inpainting[3] which is a GAN-based model using backpropagation to the input to predict meaningful content for the corrupted images. While experimental results demonstrated its superior performance on challenging image inpainting examples in comparison to state-of-the-art methods at that time, like low rank (LR) [4], PatchMatch(PM) [5] or Context Encoders(CE) [6]; the authors specified that the prediction performance strongly relies on the generative model as well as the training procedure, i.e, GANs are known to be quite unstable.
In this regard, we would like to experiment with Collaborative Sampling in Generative Adversarial Networks [1] to determine its usefulness in improving the quality of the inpainted results of a GAN model thanks to gradients feedback provided by a shaped discriminator to refine the generated samples.

In our experiment, we used the DCGAN model architecture from Radford et al. [32]. Most of the code base relies on a combination of open source tensorflow implementations of semantic image inpainting from Cheng-Bin Jin [7] and Yuejiang Liu et al. [8].

We evaluate our method on the Fashion-MNIST dataset [2]. It consists of a training set of 60,000 examples and a test set of 10,000. Each example is a 28x28 grayscale image resized to 64x64 to fit the DCGAN

model architecture. We use it as a benchmark for our new method in place of the original MNIST dataset (LeCun et al.1998), as intended by its authors.

II. Semantic Image Inpainting with Deep Generative Models

We present here the main details regarding the method proposed by Yeh et al. for semantic image inpainting [3].

Given a corrupted image as input, i.e, modelled by an image on which is applied a centered mask removing 25% of the original image content, we would like to be able to fill the hole with some realistic content.

Given a trained deep generative model, the aim is to search for the closest encoding of the corrupted image in the latent image manifold by means of backpropagation to the input using the context and prior losses. The context loss $\mathcal{L}_c$ constrains the generated image given the input corrupted image y and the hole mask M while the prior loss $\mathcal{L}_p$ penalizes unrealistic images.

Once an encoding $\hat{z}$ is found and $G(\hat{z})$ produced, the inpainting result is obtained by overlaying the uncorrupted pixels from the input image.

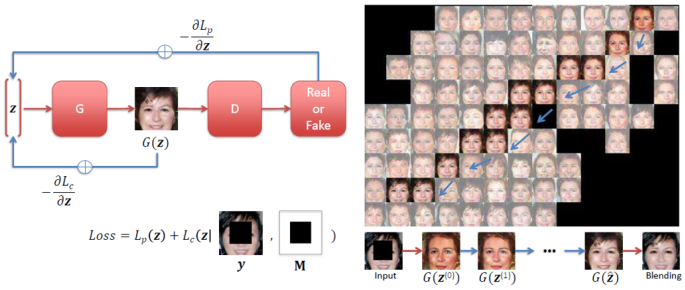

Fig. 1. Semantic image inpainting framework . Image borrowed from[3]

A. Weighted contextual loss

In order to keep the same context as in the input image, we need to make sure that the known pixel locations in the input image y are similar to the one in the generated image $G(z)$. Hence, we need to penalize the generator for not generating a similar image for the known pixels locations.

In the paper, it is defined as a weighted version of the ℓ_1-norm between the generated sample $G(z)$ and the non corrupted portion of the input image y, paying more attention to the missing region close to the hole. It is given by :

$$\mathcal{L}_c(z|y, M) = \|W \odot (G(z) - y)\|_1$$

With W being the importance weighting term and $\odot$ denoting the element-wise multiplication. In the ideal case, we would like the contextual loss to be really close to 0 to make sure that the known pixels are correctly generated.

B. Prior or perceptual loss

To recover an image that seems real, we need to make sure that the discriminator perceives it as real. Because the discriminator is trained to differentiate generated images from real images, the prior loss was chosen to be identical to the GAN loss for training the discriminator D, which is :

$$\mathcal{L}_p(z) = \lambda log(1 - D(G(z)))$$

With λ being a hyper-parameter that controls the importance of the contextual loss relatively to the prior loss

C. Inpainting

Finally, $\hat{z}$ is obtained by minimizing the combination of the perceptual and contextual loss.

$$\mathcal{L}(z) = \mathcal{L}_{contextual}(z) + \lambda \mathcal{L}_{prior}(z)$$

$$\hat{z} = \arg\min_z \mathcal{L}(z)$$

Then, the reconstructed image fills in the missing values of the corrupted input image y with $G(\hat{z})$:

$$x_{reconstructed} = M \odot y + (1 - M) \odot G(\hat{z})$$

With M being the mask modelling the missing region.

III. Collaborative Sampling in Generative Adversarial Networks

The authors noticed that most standard practices in GANs discard the discriminator during sampling. In this regard, they take advantage of valuable information learned by the discriminator regarding the data distribution to propose a collaborative sampling scheme between the generator and the discriminator to improve data generation.[1]. Guided by the discriminator, this approach refines the generated samples through gradient-based updates at a particular layer of the generator, shifting the generator distribution closer to the real data distribution.

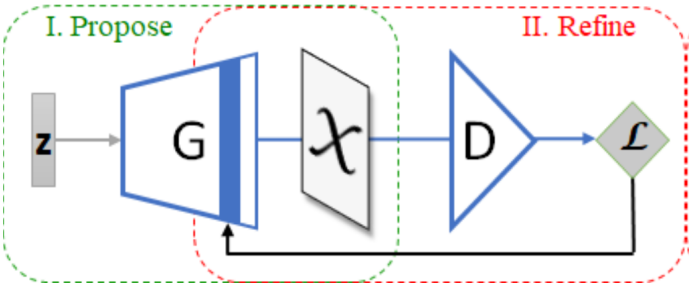

Fig. 2. Collaborative Sampling Scheme. Image borrowed from [1]

A. GAN's minimax game

The generator's role is to generate a realistic sample $G(z)$ given a latent vector z sampled from a given distribution p_z whereas the discriminator's role is to output a number corresponding to an accepting probability or confidence which tell whether a given sample comes from the generator distribution p_g or the real data distribution p_r.

When training a GAN model, we want to optimize the following expression :

$$\min_G \max_D \mathbb{E}_{x\sim p_r}[log(D(x))] + \mathbb{E}_{z\sim p_z}[1 - log(D(g(z)))]$$

The optimal solution leads to a generator capable of modelling the real data distribution p_r. When the generator and the discriminator reach an equilibrium, the generated samples are no longer distinguishable from the real samples and we have :

$$D^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)} = \frac{1}{2}$$

However it is unlikely to be reached as GAN's training is highly unstable due to the minimax problem. Most of the time, when a GAN training converges, it often ends up in local Nash equilibrium that is associated with mode collapse. The proposed method bypass training issues by directly improving image generation during sampling.

B. Collaborative Sampling

Given a generator G and a discriminator D, this method performs a single proposal step and multiple sample refinement steps. For a given latent vector z, the generator output a sample x drawn from its learned distribution p_g and the discriminator outputs a real-valued scalar $D(x)$ indicating the probability of x to be real. Subsequently, the discriminator provides gradients, with respect to the activation maps of the proposed samples, back to a particular layer of the generator and gradient-based updates are performed iteratively. The algorithm stopping criterion η is either chosen in deterministic manner to the median of the discriminator outputs for samples from the real distribution when *sample quality* matters or in a probabilistic manner by stopping the sample refinement process at each step with a positive probability when *sample diversity* is particularly at interest. In this work, we are more interested in sample quality.

C. Discriminator shaping

In the paper, the authors highlighted that the discriminator obtained from standard training may misclassify a poorly refined sample as real and thus fail to suggest further improvements. Indeed, during training, the discriminator's role remains to solely discriminate fake sample from real ones which makes it prone to overfitting the generator's distribution and less robust to unexplored regions.

In order to resolve this issue, they designed a discriminator shaping method which aim to strengthen the discriminator to effectively guide the refining procedure. Given the trained generator and discriminator, this method fine-tune the discriminator using refined samples from the Collaborative Sampling method by optimizing the following objective function :

$$\mathcal{L}_\mathcal{D} = -\mathbb{E}_{x\sim p_r}[log(D(x))] + \mathbb{E}_{x'\sim p_c}[1 - log(D(x'))]$$

where x is a sample drawn from the real distribution, x' a refined sample and p_c the refined data distribution obtained from the collaborative sampling scheme.

This should gradually expands the coverage of the model distribution to enforces the discriminator to generalize and better collaborate with the generator for sample refinement.

More details on its implementation and mathematical details can be found here[1].

IV. Experiments and Observations

A. Settings

The model was trained for 24000 iterations (roughly 25 epochs on this data set) with a batch size of 64 and a learning rate of 0.0002 for the Adam optimizer, on a NVIDIA Tesla P100 GPU on Google Colab. The architecture and other parameters remain the same as in the DCGAN model architecture from Radford et al. [9]

Furthermore, backpropagation to the input was run for 1000 iterations with $\lambda = 0.0001$ for the prior loss importance hyperparameter.

B. Assessment of the validity of baseline methods

For consistency, we wanted to experiment the baseline semantic image inpainting method and the collaborative sampling scheme without the inpainting process to make sure that we were actually able to recover the missing region and improve the generated samples quality. As in the paper, the generated samples are refined at the 2nd layer of the generator for a maximum of 50 steps with a step size of 0.1. Results are displayed in figure 3.

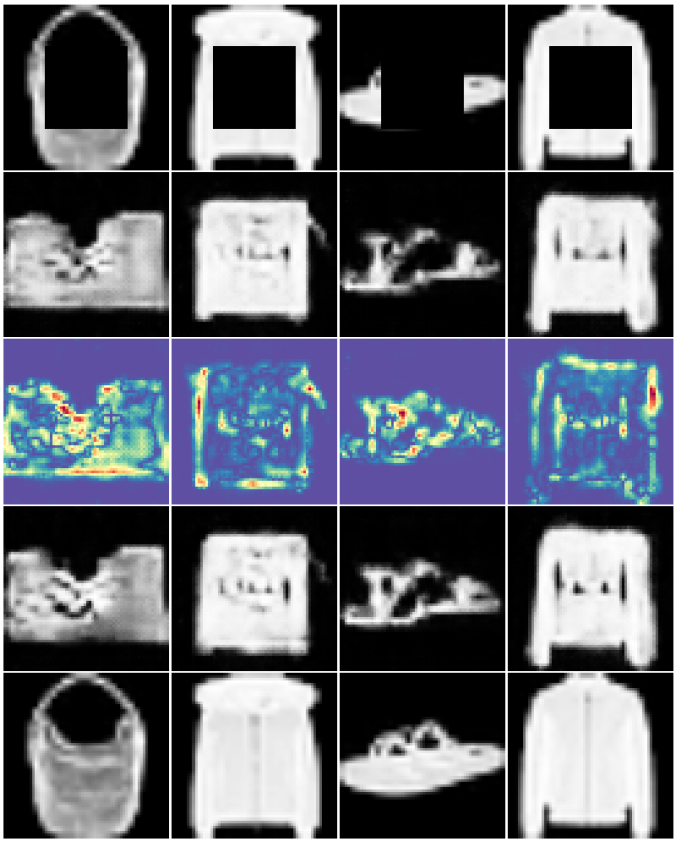

Fig. 3. Collaborative Sampling vs standard DCGAN model. *(Top) Input data with masked region (second row) output of DCGAN model (third row) Heatmap highlighting visual differences between the output of the generator in the 2nd row and the refined results in the fourth row. The closer to the red, the higher the differences (fourth row) refined samples after applying collaborative sampling and 2 epochs of discriminator shaping (bottom) Original images.*

We also wanted to highlight the effect of discriminator shaping which is essential to make sure that the discriminator is able to effectively guide the refinement process. Results are displayed in figure 4. More test samples are displayed in annexe in figures 11, 12, 13 and 14

As expected, the proposed framework is able to improve generated sample quality as the generated images looks more realistic than the ones given by the output of the generator. However, it seems that the regions for which the adjustments are made are quite random and arbitratry. Unfortunately, for our specific inpainting task, we need to find a way to guide the discriminator and the generator to focus on improving mainly the corrupted region to assess its usefulness. Indeed, if the collaborative scheme propose improvements on an area which will be overlayed by the context image, the improvement won't be visible, making it useless. In a sense, we need to make the discriminator aware of the region we actually want to improve.

V. Proposed method and Observations

In the original Collaborative Sampling scheme, the discriminator provides gradient feedback from the generator loss to adjust the activation input of the refinement layer l :

$$\mathcal{L}_G(z) = log(1 - D(G(z)))$$

In order to target the missing region, we need to make the discriminator aware of the inpainting process by making it look at the inpainted image rather than the output of the generator. Following this intuition, we modify the loss as follow and expect changes to occur only or mostly inside the missing region :

$$\mathcal{L}_G(z) = log(1 - D(y * M + G(z) * (1 - M)))$$

Furthermore, the refined samples x_c fed to the discriminator D for shaping are not anymore entirely drawn from the collaborative distribution p_c as in the original paper. Here, we feed inpainted refined samples to the discriminator instead. We hope this strategy will allow the discriminator to guide more effectively the generator during the refinement process in proposing more realistic inpainted samples. Results are shown in figure 5 and more test samples are provided in annexe figures 15, 16, 17 and 14 .

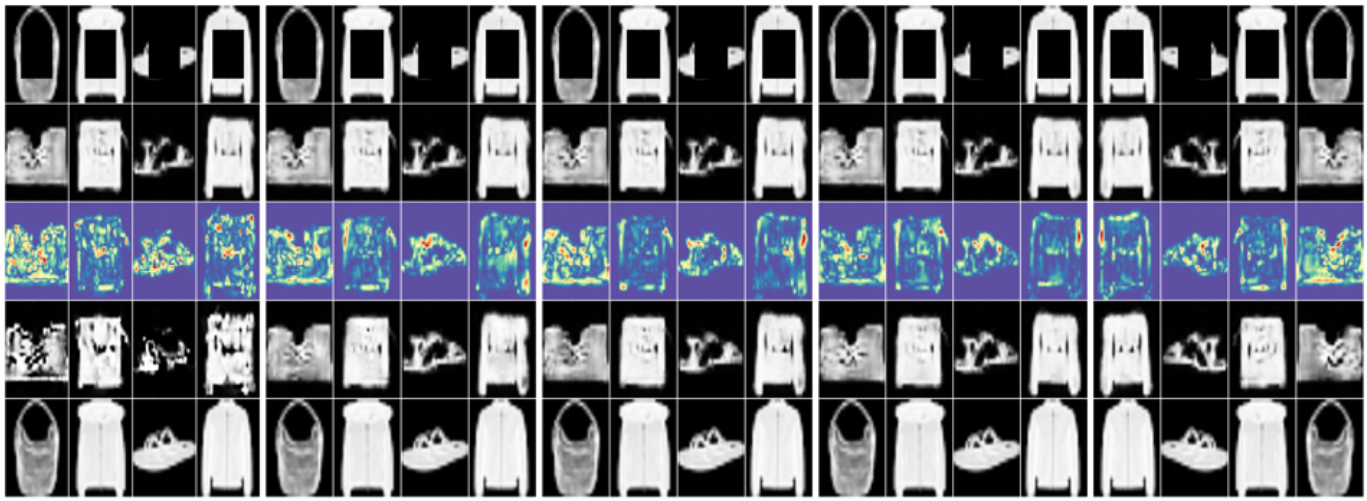

Fig. 4. Effect of collaborative sampling at different stages of discriminator shaping. From left to right : without discriminator shaping, after 0.1 epoch of discriminator shaping, 0.5 epoch, 1 epoch, 1.5 and 2 epochs. *(Top) Input data with masked region (second row)output of DCGAN model (third row) Heatmap highlighting visual differences between the output of the generator in the 2nd row and the refined results in the fourth row. The closer to the red, the higher the differences (fourth row) refined samples after applying collaborative sampling at a certain stage of discriminator shaping (bottom) Original images.*

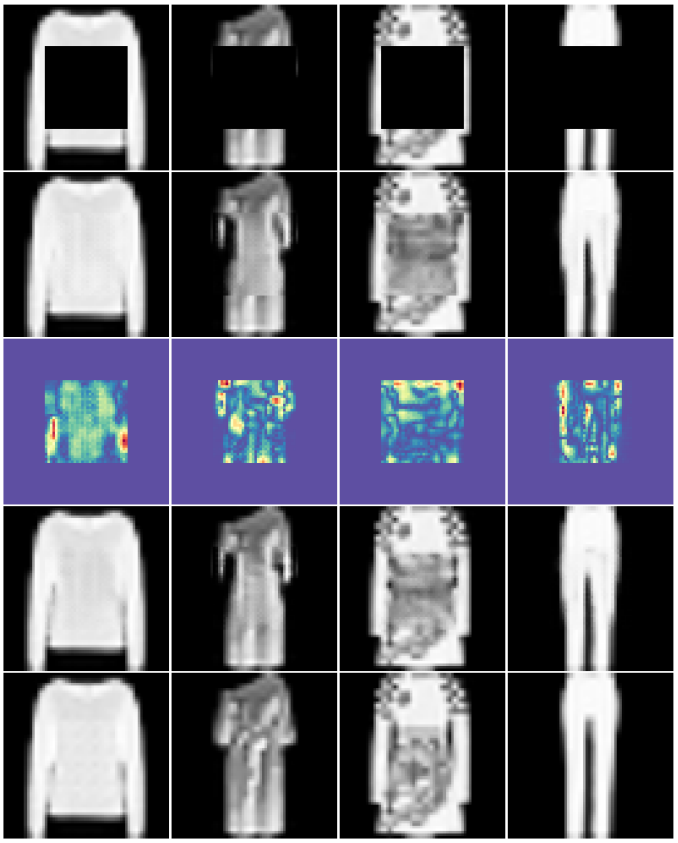

Fig. 5. Collaborative image inpainting vs DCGAN model. *(Top) Input data with masked region (second row) Output of the generator surrounded by its context image (third row) Heatmap highlighting visual differences between the inpainted output of the generator in the 2nd row and the refined results in the fourth row. The closer to the red, the higher the differences (fourth row) refined samples surrounded by their context image after two epochs of discriminator shaping (bottom) Original images.*

In figure 6, we show the necessary importance of discriminator shaping for inpainting new in this collaborative sampling scheme, as we obtain results of higher visual quality with further iterations.

VI. Discussion

Although the presented results are qualitatively superior than the original inpainting method, because of the toy dataset used, generalizing those results may be too presomptuous. It would be better to have further confirmation by experimenting this method on high quality images from CelebA dataset or CIFAR-10 datasets for example.

Furthermore, while qualitative results are useful and necessary, quantitative measures like PSNR (Peak-Signal-to-noise-ratio) or IS (Inception Score) to assess this method ability in proposing higher quality images than the the original would be more rigorous.

However, as highlighted in the beginning, the aim is to fill the missing region with realistic content and not necessary the ground truth. Experimental results obtained from [3] and other similar works, showed counterintuitive and conflicting results using PSNR and SSIM. While a reconstructed image clearly showed better perceptual quality between an artifact-free generated sample with another with visible ones, the PSNR value was lower for the latter, suggesting lower image quality. The actual problem was that the model generated a completely different hair style from the ground truth,

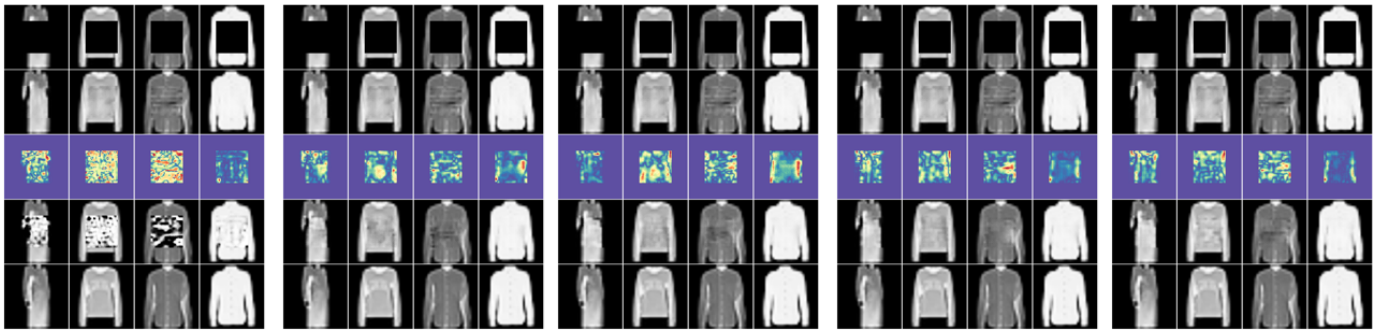

Fig. 6. Effect of collaborative image inpainting at different stages of discriminator shaping. From left to right : without discriminator shaping, after 0.1 epoch of discriminator shaping, 0.5 epoch, 1 epoch, 1.5 and 2 epochs. *(Top) Input data with masked region (second row) Output of the generator surrounded by its context image (third row) Heatmap highlighting visual differences between the inpainted output of the generator in the 2nd row and the refined results in the fourth row. The closer to the red, the higher the differences (fourth row) refined samples after applying collaborative sampling and discriminator shaping surrounded by its context image (bottom) Original images.*

even though it was more realistic than the other generated sample.

Finally, the proposed method provides higher quality inpainted results but at the expense of further iterations for discriminator shaping and storage to compute the closest encoding latent vectors offline.

VII. Conclusion

Throughout this project, we ran multiples experiments to determine how to integrate the Collaborative Sampling scheme into a pre-existing GAN model to tackle and improve the semantic image inpainting problem. Superior qualitative results led us to confirm this method potential to help in obtaining high quality results as this method is scalable to any GAN model. In a future work, provided more time at our disposal, it would have been interesting to investigate with higher quality images and provide numerical results to strengthen our results.

VIII. Acknowledgements

This paper has been made possible with the help of Professor Alexandre ALAHI and the VITA LAB team. I am very grateful to my supervisor, Yuejiang LIU, for its support and valuable feedbacks and advices throughout this project.

References

[1] Y. Liu, P. A. Kothari, and A. Alahi. (2019) Collaborative sampling in generative adversarial networks. [Online]. Available: https://arxiv.org/abs/1902.00813

[2] R. V. Han Xiao, Kashif Rasul. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. [Online]. Available: https://arxiv.org/abs/1708.07747

[3] C.-B. Jin, "semantic-image-inpainting," https://arxiv.org/abs/1607.07539, 2018.

[4] Y. H. . D. Z. . J. Y. . X. L. . X. He. (2013) Fast and accurate matrix completion via truncated nuclear norm regularization. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6389682

[5] C. B. E. S. A. F. D. B. Goldman. (2009) Patchmatch: a randomized correspondence algorithm for structural image editing. [Online]. Available: https://dl.acm.org/doi/10.1145/1531326.1531330

[6] T. D. A. A. E. Deepak Pathak; Philipp Krahenbuhl; Jeff Donahue. (2016) Context encoders: Feature learning by inpainting. [Online]. Available: https://arxiv.org/abs/1604.07379

[7] C.-B. Jin, "semantic-image-inpainting," https://github.com/ChengBinJin/semantic-image-inpainting, 2018, commit xxxxxxx.

[8] Y. Liu, P. A. Kothari, and A. Alahi. (2019) Collaborative sampling in generative adversarial networks. https://github.com/vita-epfl/collaborative-gan-sampling.

[9] L. M. A. Radford and S. Chintala. (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. [Online]. Available: https://arxiv.org/abs/1511.06434

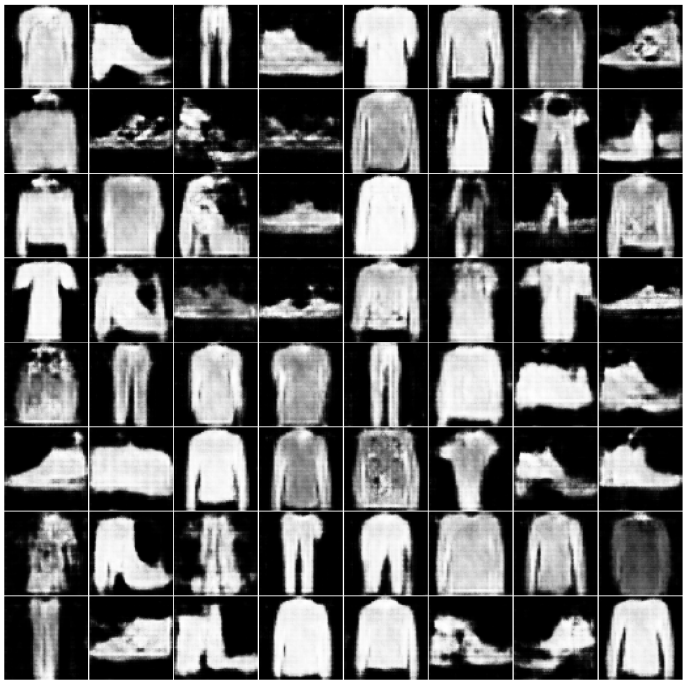

Fig. 7. Samples of the generator of the DCGAN model during training at epoch 1 for FASHION-MNIST

Fig. 9. Samples of the generator of the DCGAN model during training at epoch 15 for FASHION-MNIST

Fig. 8. Samples of the generator of the DCGAN model during training at epoch 5 for FASHION-MNIST

Fig. 10. Samples of the generator of the DCGAN model during training at epoch 25 for FASHION-MNIST

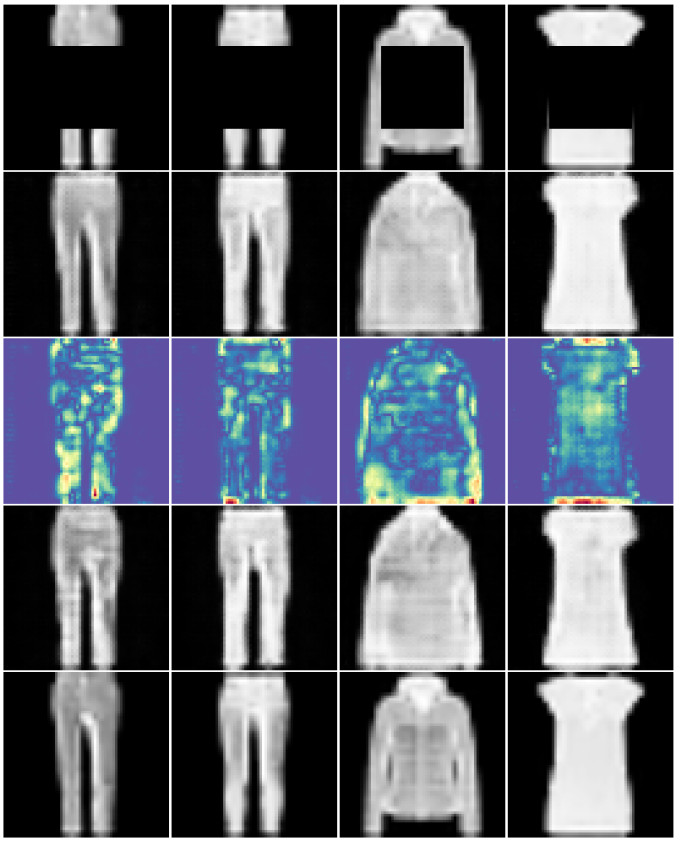

Fig. 11. Collaborative Sampling after 2 epoch of discriminator shaping vs DCGAN model. *(Top) Input data with masked region (second row) output of DCGAN model (third row) Heatmap highlighting visual differences between the output of the generator in the 2nd row and the refined results in the fourth row. The closer to the red, the higher the differences (fourth row) refined samples after applying collaborative sampling and 2 epochs of discriminator shaping (bottom) Original images.*

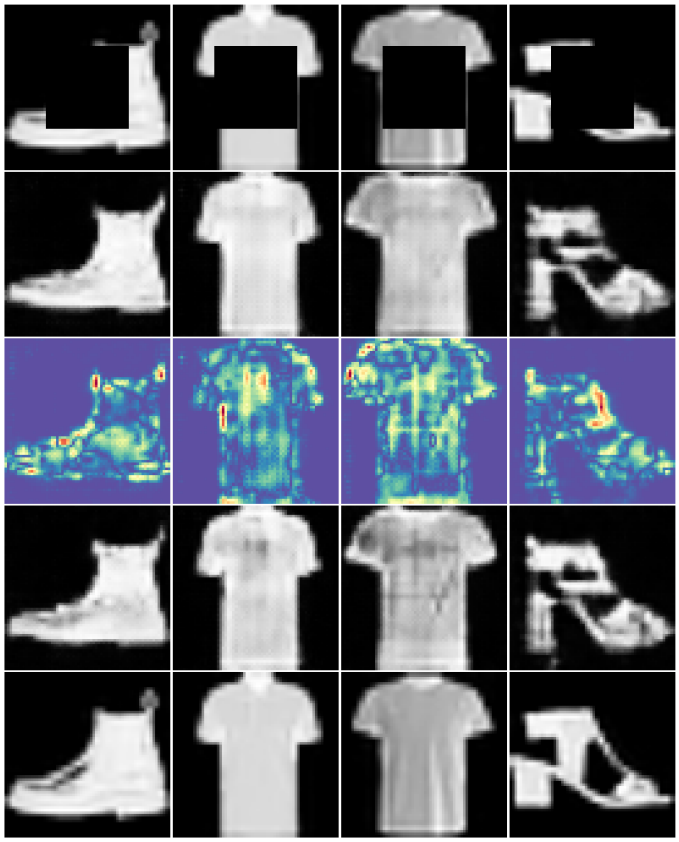

Fig. 12. Collaborative Sampling after 2 epoch of discriminator shaping vs DCGAN model. *(Top) Input data with masked region (second row) output of DCGAN model (third row) Heatmap highlighting visual differences between the output of the generator in the 2nd row and the refined results in the fourth row. The closer to the red, the higher the differences (fourth row) refined samples after applying collaborative sampling and 2 epochs of discriminator shaping (bottom) Original images.*

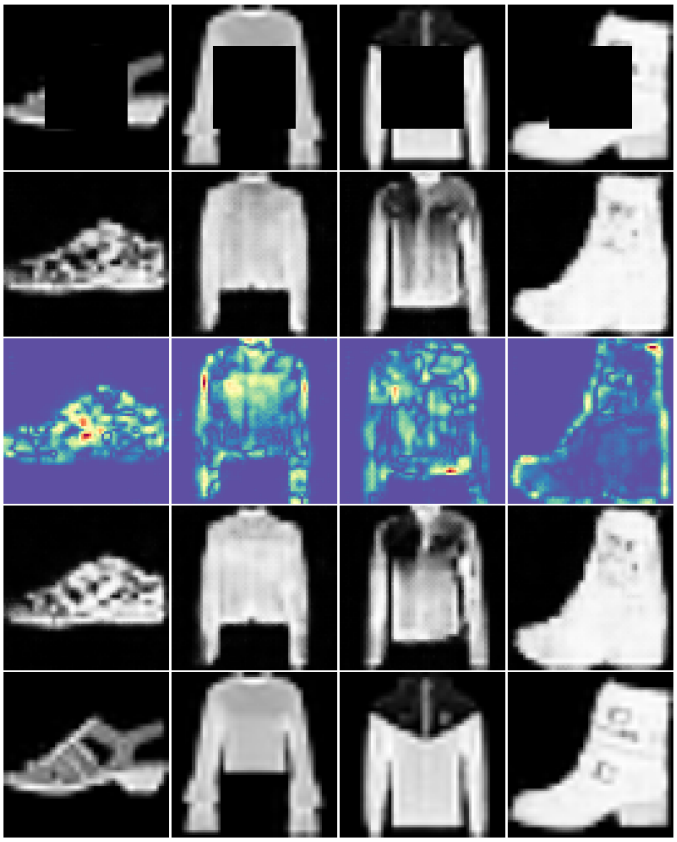

Fig. 13. Collaborative Sampling after 2 epoch of discriminator shaping vs DCGAN model. *(Top) Input data with masked region (second row) output of DCGAN model (third row) Heatmap highlighting visual differences between the output of the generator in the 2nd row and the refined results in the fourth row. The closer to the red, the higher the differences (fourth row) refined samples after applying collaborative sampling and 2 epochs of discriminator shaping (bottom) Original images.*

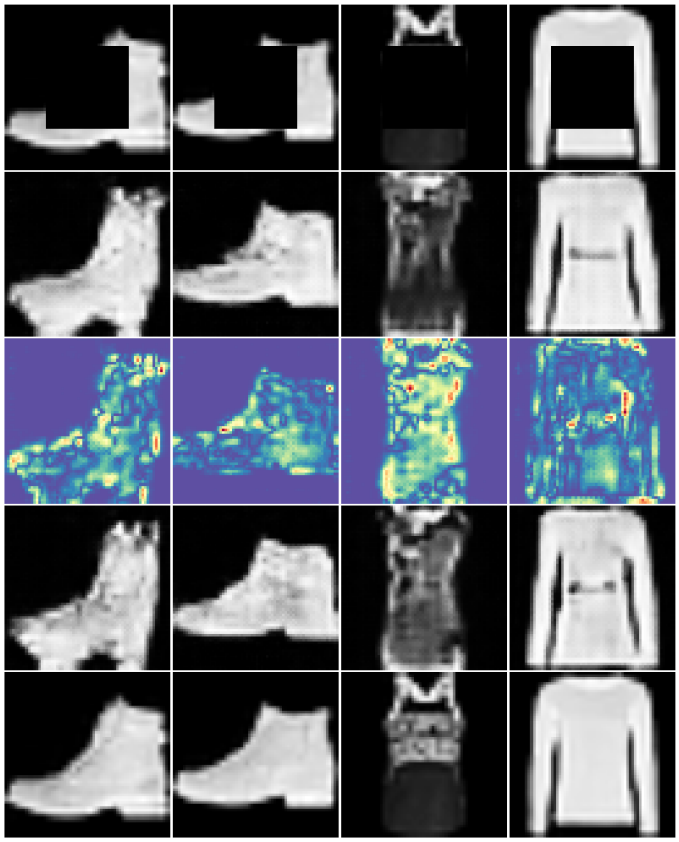

Fig. 14. Collaborative Sampling after 2 epoch of discriminator shaping vs DCGAN model. *(Top) Input data with masked region (second row) output of DCGAN model (third row) Heatmap highlighting visual differences between the output of the generator in the 2nd row and the refined results in the fourth row. The closer to the red, the higher the differences (fourth row) refined samples after applying collaborative sampling and 2 epochs of discriminator shaping (bottom) Original images.*

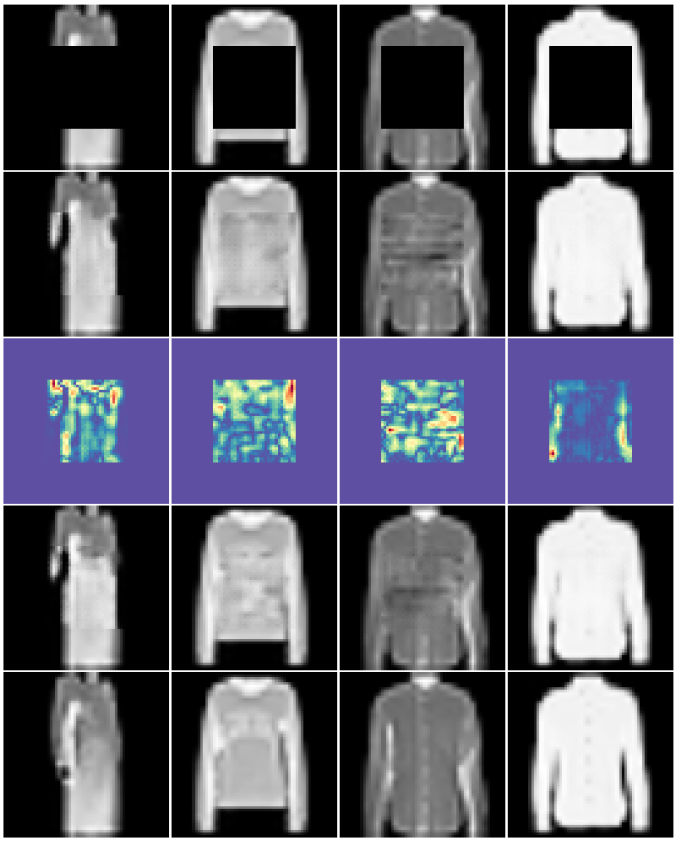

Fig. 15. Collaborative image inpainting after 2 epoch of discriminator shaping vs DCGAN inpainting

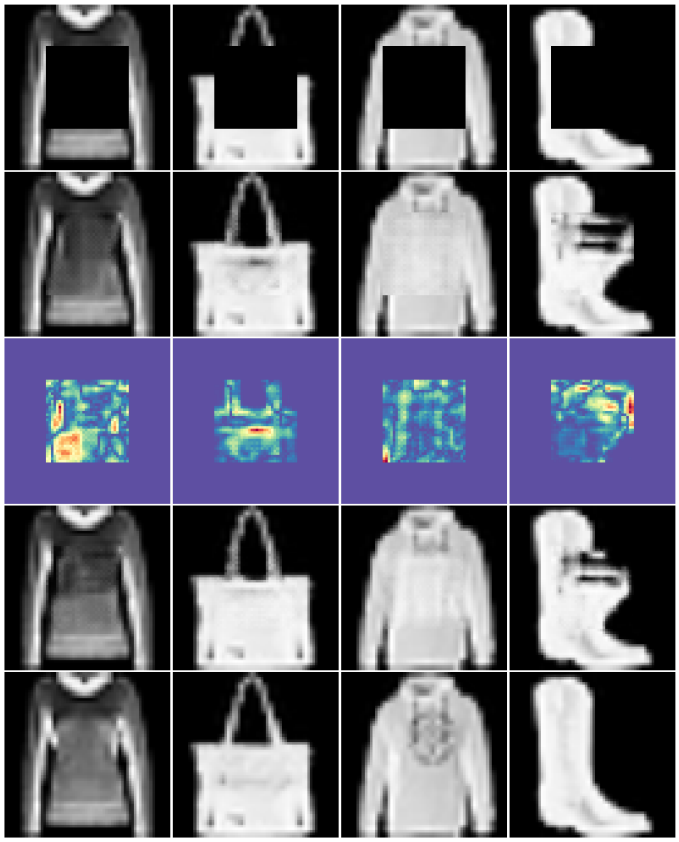

Fig. 16. Collaborative image inpainting after 2 epoch of discriminator shaping vs DCGAN inpainting

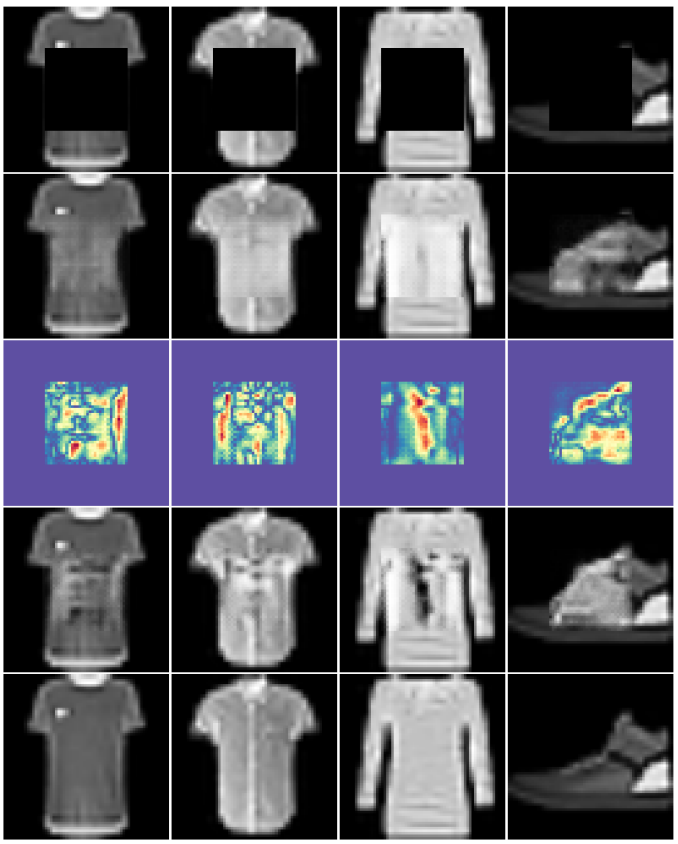

Fig. 17. Collaborative image inpainting after 2 epoch of discriminator shaping vs DCGAN inpainting

Fig. 18. Collaborative image inpainting after 2 epoch of discriminator shaping vs DCGAN inpainting