An Application of GANs For Scenic Image Translations

Adam Dulloo **Alexander Hunter** **Martin Yong**

Abstract

The problem which we attempt to address in this project is that of image translation, namely translating images of daytime scenes into nighttime scenes and vice-versa. In order to achieve this, we employed a new machine learning technique called a "CycleGAN" which pits two sets of image generators and discriminators against one another. The final trained model is able to take in an image of a landscape scene and returns a somewhat realistic reimagined image of that same scene, but at the opposite time of day. The resulting generated images, whilst impressive for some inputs, often contain artefacts or do not visibly change the time of day. These problems may be resolved with a larger dataset.

1 Introduction

Suppose we were in the position of a film producer presented with a large string of photographs of potential filming sites captured during various times of the day. Further suppose that a pivotal decision must be made regarding the choice of a site for an evening shoot. As it is usually a resource intensive process of capturing photographs of each location at a specific time-interval of the evening, one might resort to informing their decision based on an imagination of what each site might resemble in the evening.

We would like to present a systematic approach to this persistent problem faced by many field professionals via the application of Generative Adversarial Networks (GANs) to scenic image-to-image translations. Devised by Ian Goodfellow et al. in 2014, GANs belong to a family of generative models that possess the capacity to perform domain translations in the absence of paired training samples, which can be costly to acquire in practice. The broad application of the GANs framework has delivered promising results across numerous domains that include image processing/generation, computer vision, video game graphics, and astronomical modelling.

Specifically, this paper aims to formalize an in-depth approach to day-to-night (or vice versa) scenic image translations using Cycle-Consistent Adversarial Networks (CycleGANs). Let daytime images be represented by domain **X** and nighttime images by domain **Y**, the primary aim of the CycleGAN is to learn the mapping functions **G : X -> Y** and **F : Y -> X**. These functions are termed as 'generators' in that given an input **x \in X / y \in Y** they respectively output an image **G(x) \in Y / F(y) in X**.

2 Dataset

2.1 Base Dataset

The base dataset used in our study consisted of 987 panoramic images set in nature; 551 daytime and 436 nighttime images.

7,066 images were scraped from DuckDuckGo's search engine and as such, not all images returned were relevant or useful in training our model. To clean the dataset, all images were manually filtered according to a set criteria, such as only including real photographs that were taken outside with the

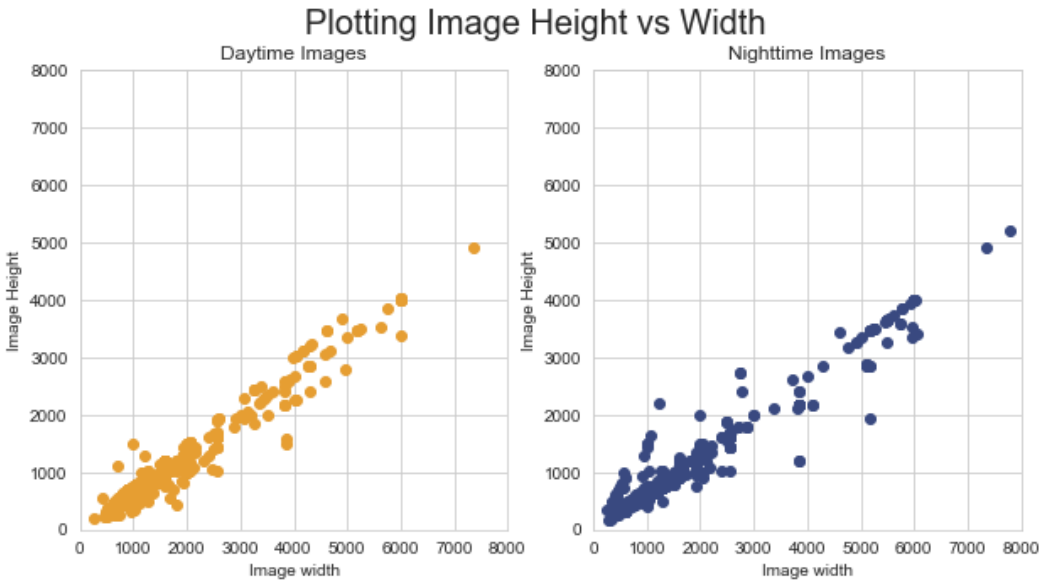
Preprint. Under review.

camera focused on the background. After reviewing the filtered dataset, it was noted that landscape shots of buildings were overrepresented in the nighttime dataset. For this reason, an additional filtering was performed to remove this overrepresented type of image as leaving it included may have adverse effects on training.

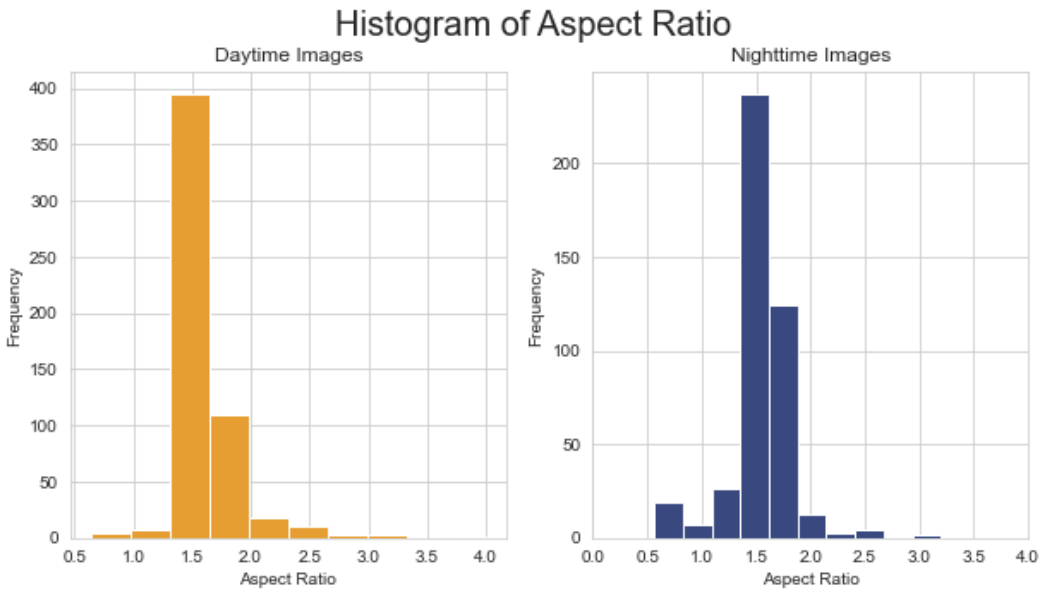
The following are typical images from the daytime dataset and nighttime dataset:

The filtered dataset was analysed to explore potential trends in the data in the following ways:

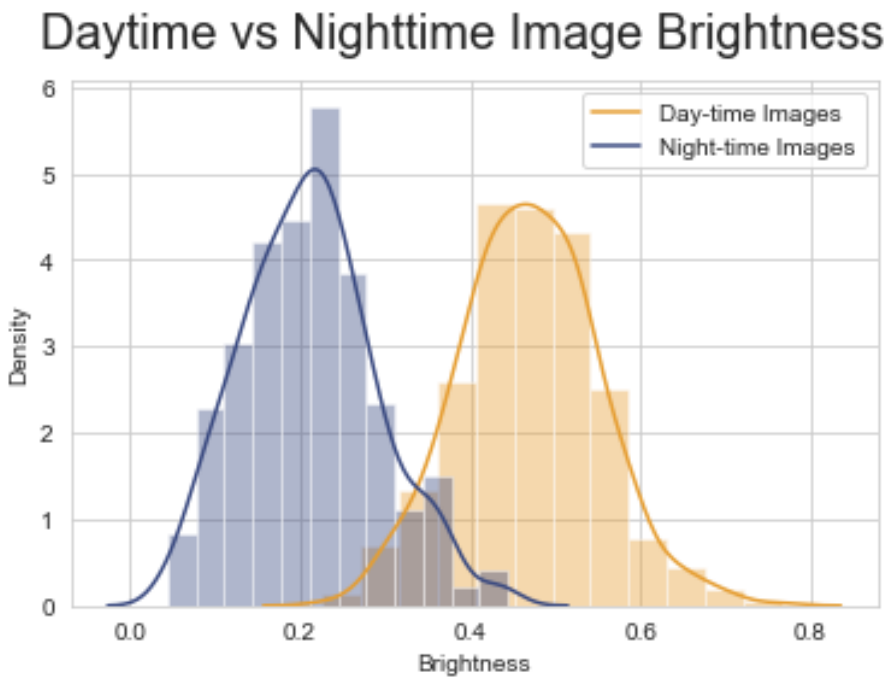
- Comparing image resolutions between daytime and nighttime images
- Comparing aspect ratios between daytime and nighttime images
- Comparing the brightness between daytime and nighttime images
- Comparing the colour channel values between daytime and nighttime images



From this figure it can be seen that daytime and nighttime images have a very similar distribution of resolutions and follow the same relation between image width and image height.

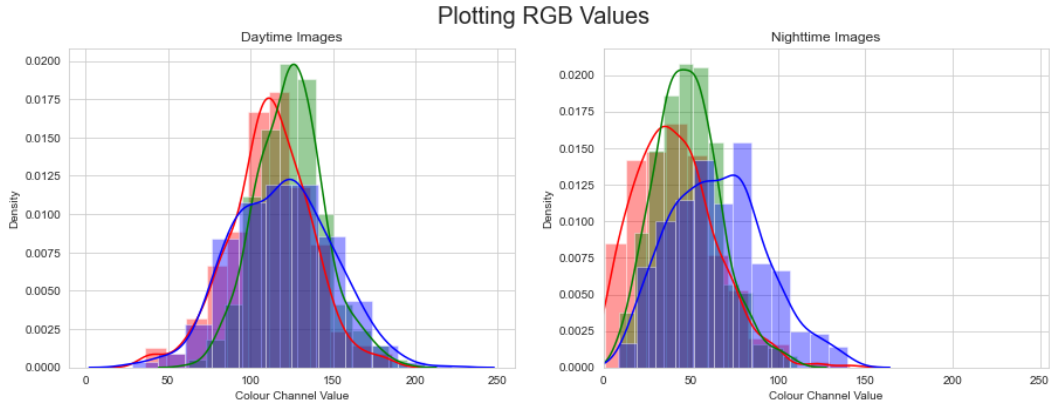


Exploring this relationship between image width and height further by analysing the aspect ratio again shows a very similar distribution.



The average brightness in both daytime images and nighttime images was calculated and plotted, where the brightness of an image is calculated by first finding the average colour, then finding the distance between the average colour and pure black and normalising such that values fall between 0 and 1. As expected there is a clear trend that daytime images have a greater brightness than nighttime images. This trend will be a key indicator to the model as to whether certain images belong to a certain time of day. For instance, if a generator attempts to produce a daytime image from a nighttime image and the brightness of the output is 0.2 or less, then the discriminator can be certain that the image is fake as the vast majority of daytime images have a higher brightness. The same is

true for a generator producing a nighttime image with a brightness larger than 0.5. The overlap in brightnesses between the two groups from 0.25 to 0.4 may prove more difficult to correctly classify and context clues such as what types of colours are present in the scene will become a useful metric.



Analysing the red, green and blue values across both data sets reveals that daytime images contain roughly the same amount of red, green and blue, whereas there is a significantly higher proportion of blue compared to red or green in nighttime images, revealing another potential source to distinguish between the two datasets.

2.2 Preprocessiing

Due to the constraint that inputs and outputs to the CycleGAN must be of a fixed, predetermined size, all images were first scaled down such that their smallest dimension was 300 pixels wide and in order to make training easier, the rgb values were normalised such that they take values between -1 and 1.

Synthetic data generation methods were employed to address the relatively small sample size of the dataset. The aggregate number of training samples were artificially increased through taking combinations of random crops and reflections of images in the base dataset, reducing the resolution of the images down to their final size of 256x256. This resulted in a training dataset of size (2644 day, 2092 night) and test dataset of size (662 day, 524 night).

3 Methods

3.1 Method

As illustrated in Figure 4.1, our CycleGAN model includes two generative networks G and F, that given training samples x and y, map input from G: X to Y and F: Y to X. The generative networks are trained indirectly via two discriminative networks Dx and Dy. The purpose of Dx is to distinguish between true images from x and translated images F(y). Likewise, Dy distinguishes between true images from y and translated images G(x).

3.2 Generator

A deconvolutional neural network provided by Tensorflow's pix2pix library was used for our generator model. This network comprises 1 input layer, 9 progressively smaller convolutional layers (downsampling), 14 residual blocks, 6 progressively larger convolutional layers (upsampling), and 1 transposed convolutional layer (deconvolution/output layer).

The generator receives an vectorised image from its input domain and subsequently returns a vectorised image from its target domain.

3.2.1 Discriminator

A convolutional neural network provided by Tensorflow's pix2pix library was used for our discriminator model. This network comprises 1 input layer, 4 progressively larger convolutional layers, and 1 2-D convolutional layer.

The discriminator is tasked with classification. It receives a vectorised image as input, either a true or generated image, and subsequently returns a 30 x 30 matrix of probabilities corresponding to the class label of the input image.

3.2.2 Optimisation/Training

GANs learn in an unsupervised manner where each generator is trained to produce an image aimed at 'fooling' their respective discriminator. During the training phase, the weights of each generative and discriminative network are updated independently in tandem to ensure that the generators produce superior images and discriminators gain greater competency in distinguishing translated images.

The full objective function incorporated by our model can be defined by the sum of 3 loss components: adversarial loss, cycle consistency loss and identity loss.

3.2.3 Cycle Consistency Loss

The use of cycle consistency losses was first promulgated by Jun et.al and is based on the applied principle of transitivity to the generator functions, where translations of an image from one domain to another and back again should deductively result in identical inputs and output. Cycle consistency losses are measured by the absolute difference between the pixel values of a 'cycled' image and its original input.

$$\mathcal{L}_{cyc}(G, F) = E_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + E_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1]$$

3.2.4 Adversarial Loss

Adversarial loss can be regarded as a quantification of a generators' ability to 'deceive' its corresponding discriminator and is defined by the cross-entropy between real and generated distributions. This objective function is used to train the generator and discriminator networks separately inline with their respective goals. Generators are updated to minimise the loss while the discriminators maximise it.

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

Where,

D(x): the discriminator D's probabilistic estimate of the real image instance 'x' being real.

Ex: the expectation over the domain of all X images.

G(z): the generator's output given an image 'z' from domain Z.

D(G(z)): the discriminator D's probabilistic estimate of the generated image instance G(z) being real.

Ez: the expectation over all generated image instances.

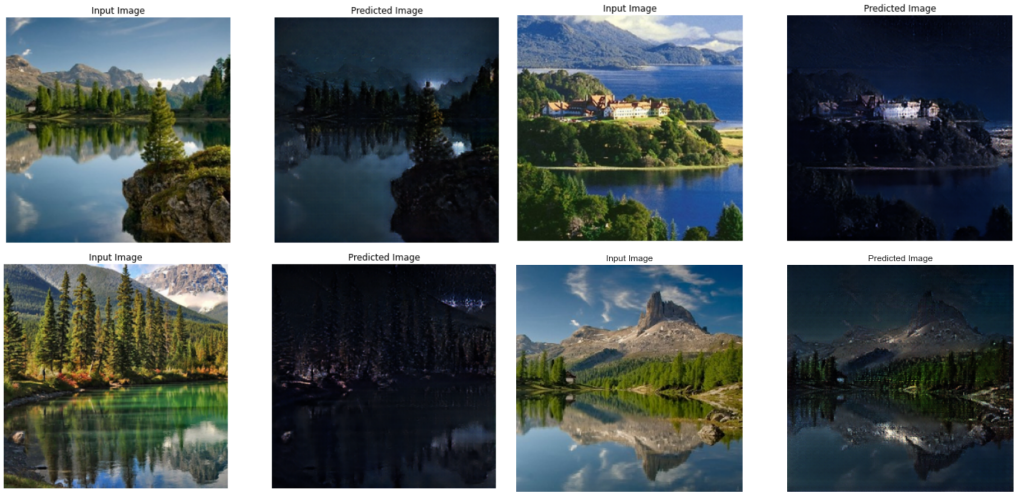
3.2.5 Identity Loss

The method of identity loss put forward by Taigman et al was chosen to further regularize the generators.The transitive intuition behind identity loss is similar to that of cycle consistency loss, in that instances of input to the generator belonging to the domain of its output should theoretically result in output being identical to the input. This objective is measured by the absolute difference between the pixel values of the input and output instances of the same domain.

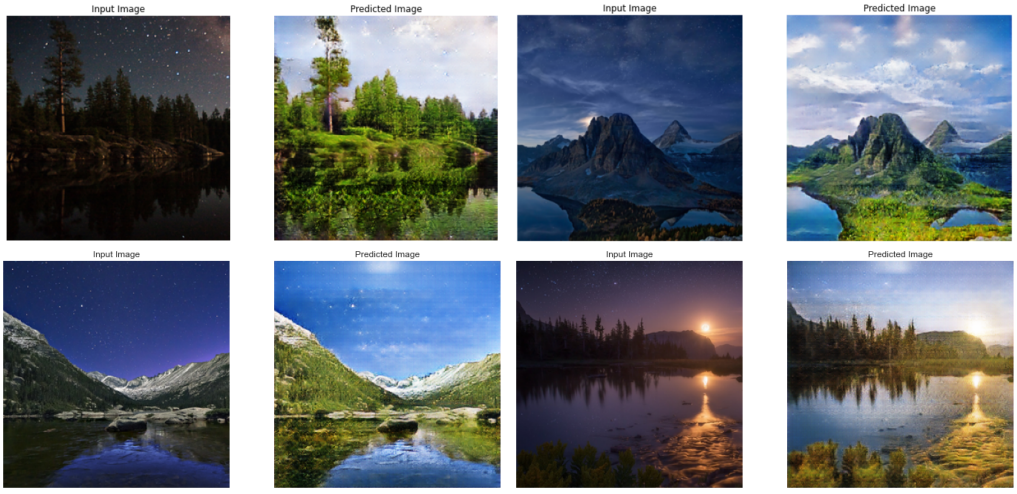
$$\mathcal{L}_{identity}(G, F) = E_{y \sim p_{data}(y)}[\|G(y) - y\|_1] + E_{x \sim p_{data}(x)}[\|F(x)) - x\|_1]$$

4 Experiments and Results

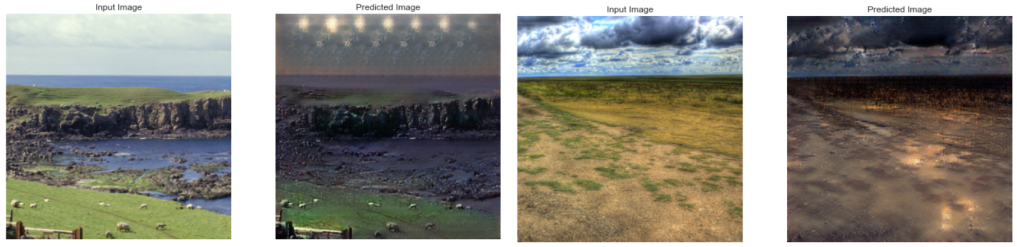
The following are some of the most impressive examples of the model's performance trained on 40 epochs when given a daytime image:



The following are some of the most impressive examples of the model's performance trained on 40 epochs when given a nighttime image:



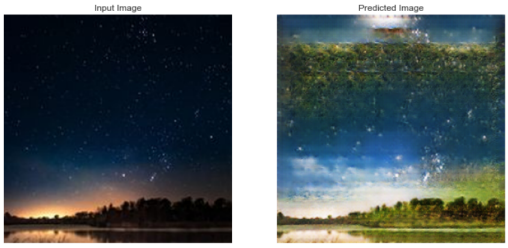
Some outputs from the model however contain artefacts and as such would not fool a human into thinking that the output is a real image such as the following:



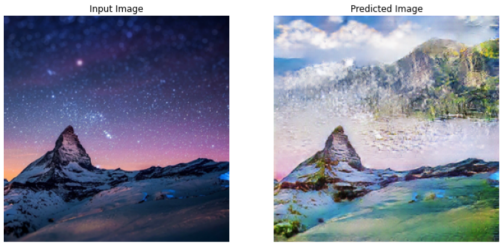
Other undesirable results include the model making no noticeable change to the input image such as:

It is possible that the generator is able to subtly manipulate some images in order to trick the discriminator, but leaving the images appearing almost identical to a human.

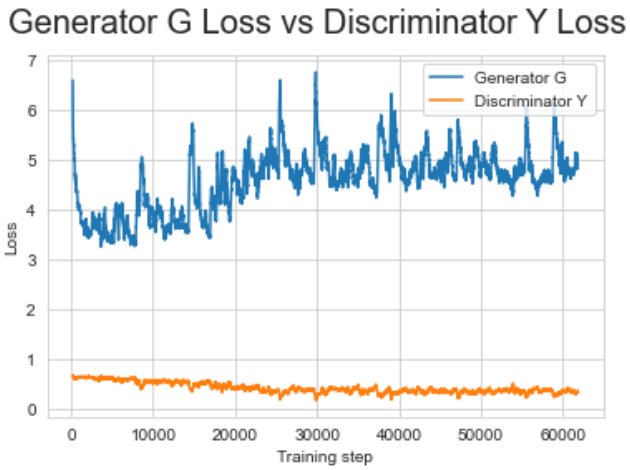
The model appears to have learnt a relation between daytime images and green foliage and as a result often adds green plant life to nighttime inputs that make no sense, such as the following:

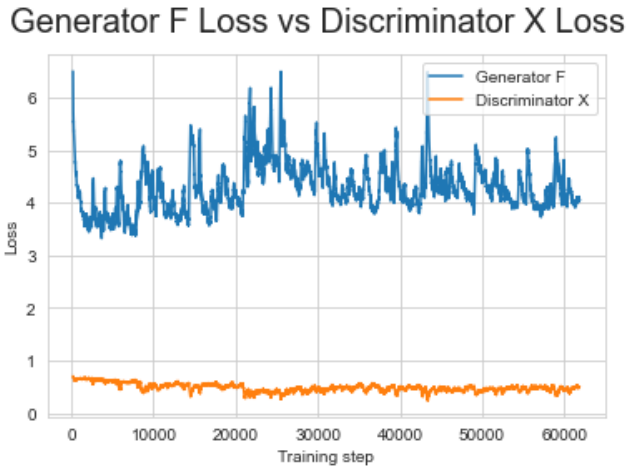


This leads to the biggest flaw in the performance of this model, namely that it's goals do not seem to directly be in line with the project goals. The goal of the CycleGAN model is to take in an image from one domain and return an image from another domain. The goal of this project however is to reimagine a particular image of a scene at another time of day. This can lead to the following situation:



The input image is from the nighttime domain and the output looks convincingly from the daytime domain, however the model has added many features to the image that clearly do not appear in the input, such as the mountain in the background. Whilst the output does indeed look like a real daytime image, it does not look like the input image if it were actually taken during the day. It is possible however that with a large enough dataset and enough training that this problem would cease to be. An output such as the one shown may produce a very low adversarial loss, fooling the discriminator, however would have a lot of potential improvement in terms of cycle consistency loss.





The loss of both generators and their respective discriminators was recorded over the course of training shown in the above figures. As the loss function of the generator is determined by the performance of the discriminator and vice-versa, the loss function can change sporadically over time with no guarantee of convergence to any particular value. This makes it difficult to quantitatively conclude that the model has improved over time from just looking at the loss values alone.

5 Discussion

Overall, the application of the CycleGAN framework to scenic image-to-image translations in the absence of paired training data has shown significant results as seen in the examples highlighted earlier in the previous section. Though we would like to draw attention to the instances where our model failed, these image translations typically contained characteristics like heavy cloud coverage and stars in the background of the image. Additionally, we note the presence of abrupt shifts in the color gradient between the foreground and background of these images. An amalgamation of both these features are likely to have confused our model. Another cause for model breakdown in these cases may be attributed to the variance and imbalanced sample sizes between the daytime and nighttime training images. The daytime training set was larger and contained significantly more foliage and forest coverage in comparison to the nighttime training set which contained more mountainous imagery and rock formations. Going forward, the first step to resolving these model generalisation errors will require a greater degree of emphasis on the data curation process, specifically in reducing the geometric/structural variance in imagery between both training sets. Furthermore, one could improve model generalisation through experimentation with alternative image normalisation/color scaling techniques and objective functions. Moreover it will generally be ideal to train the model with paired image data, if available, especially in cases where the intended translations between domains are extreme.

References

[1] Taigman et al. https://arxiv.org/abs/1611.02200

[2] Ian J. Goodfellow et al. https://arxiv.org/abs/1406.2661