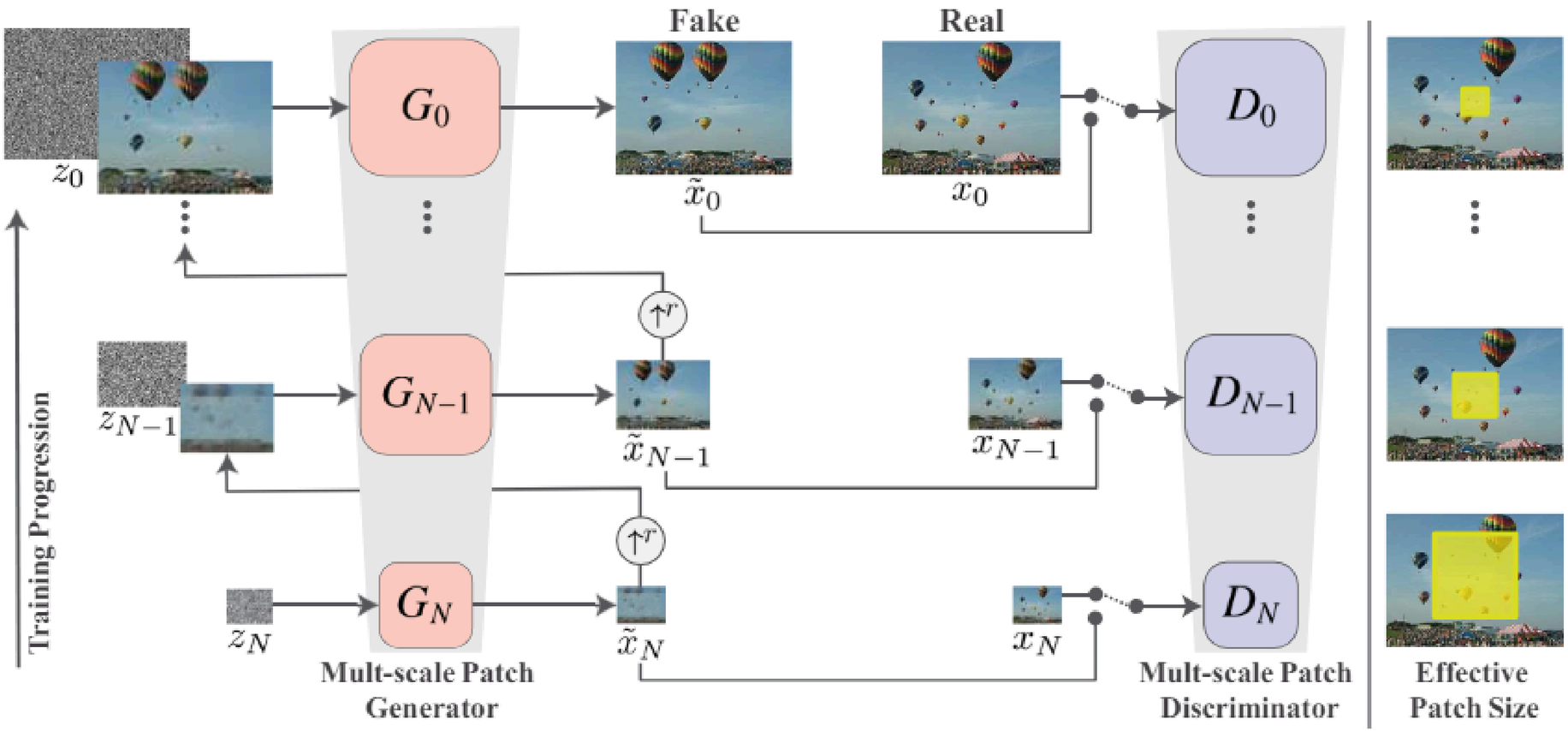
SinGAN:
Learning a Generative Model from a Single Natural Image

Elaborated by:
Haithem DAGHMOURA
Kais ZAIRI
Nourelhouda KLICH

Introduction

- Replication of the original SinGAN
- Hyperparameter and Padding Investigation
- Alternative Loss Functions for the discriminator

SinGAN architecture



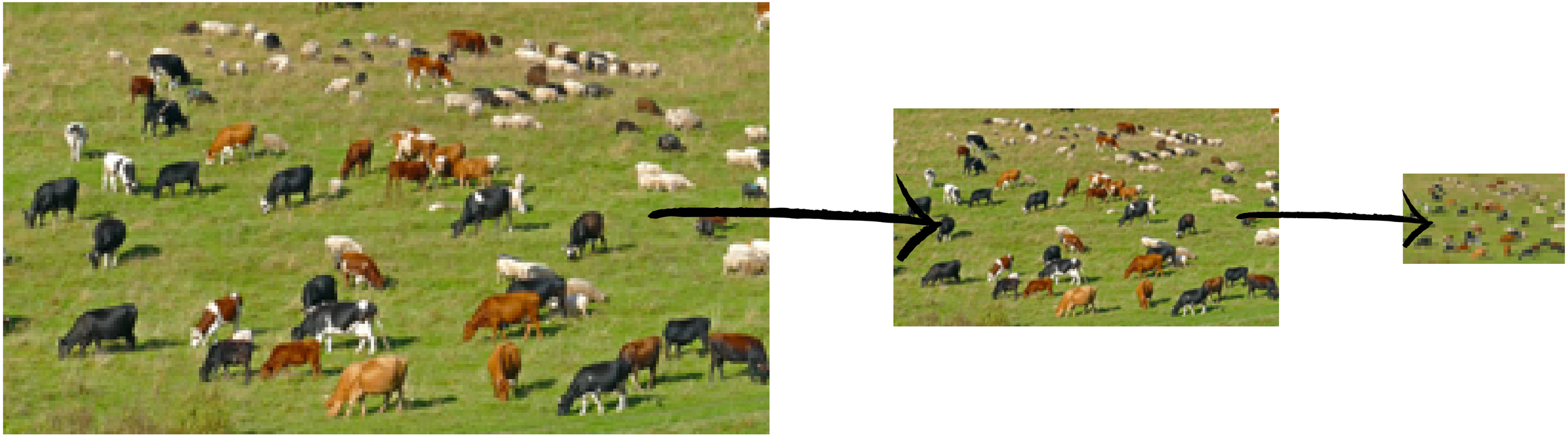
Pre-Processing for SinGAN

-Normalization: Standardize data to a range of values in [-1,1].

-Resizing: Adjust image dimensions accurately.

-Scale Computation: Determine optimal scales and factors for training.

Resizing

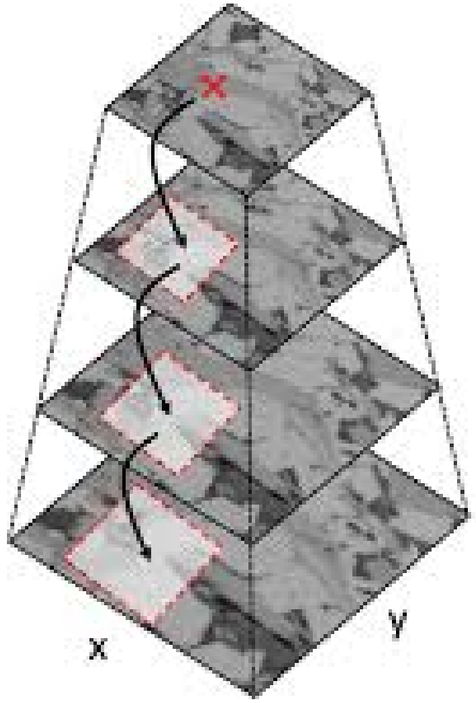


original size

resize with different factors

Fixing scales and sizes

- compute number of scales with a given scale factor
- compute the scaling factor for a given number of scales



Training

Objectives

- Combined objective: $$\min_{G_n} \max_{D_n} L_{adv}(G_n, D_n) + \alpha L_{rec}(G_n)$$

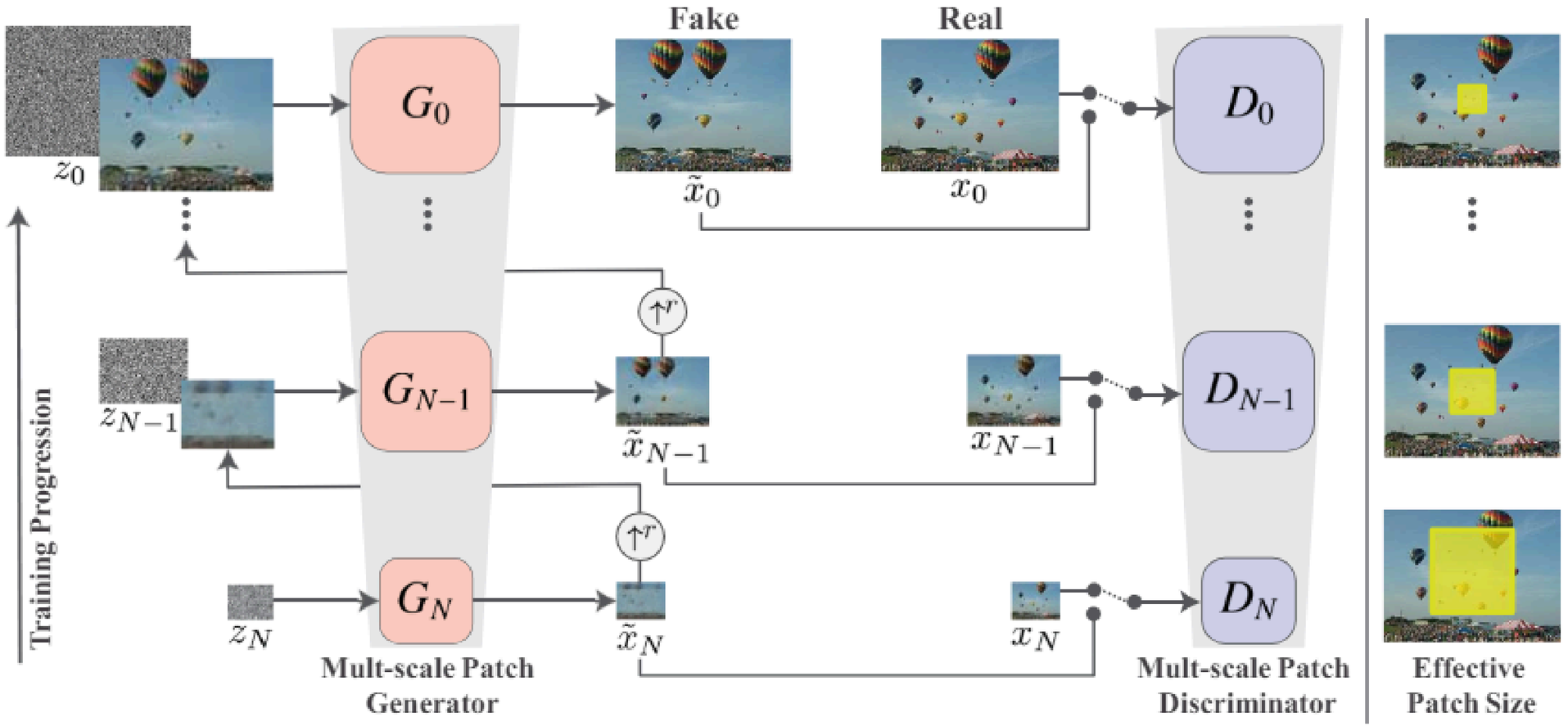
- Adversarial Loss: $$L_{adv} = -\mathbb{E}[D_n(x_n)] + \mathbb{E}[D_n(\tilde{x}_n)] + \lambda \mathbb{E}[(\|\nabla D_n(x)\|_2 - 1)^2]$$

- Reconstruction Loss:

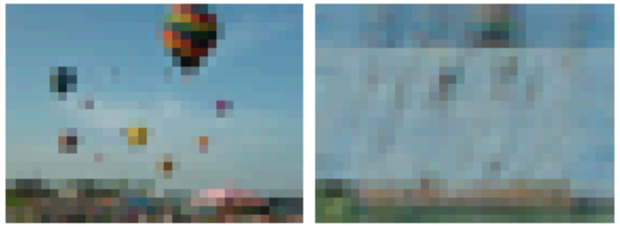
For finest scale N: $$L_{rec} = \|G_N(z^*) - x_N\|_2^2$$

For finer scales n<N: $$L_{rec} = \|G_n(0, (\tilde{x}_{n+1})\uparrow_r) - x_n\|_2^2$$

Training Procedure

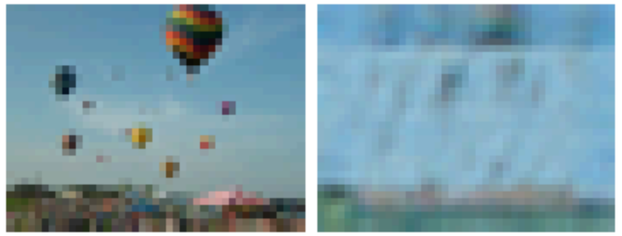


Training results



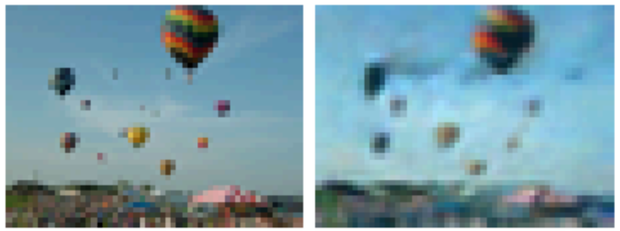
Real Fake

(a) Images at Scale 0



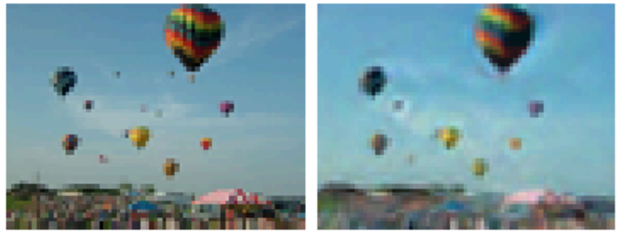
Real Fake

(b) Images at Scale 1



Real Fake

(c) Images at Scale 2



Real Fake

(d) Images at Scale 3

Real Fake

(e) Images at Scale 4

Real Fake

(f) Images at Scale 5

Real Fake

(g) Images at Scale 6

Real Fake

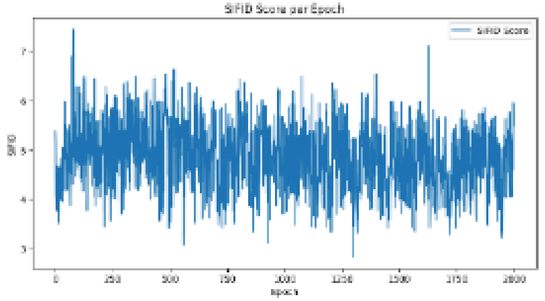
(h) Images at Scale 7

Real Fake

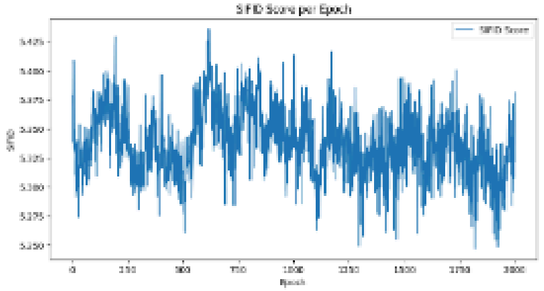
(i) Images at Scale 8

Training results

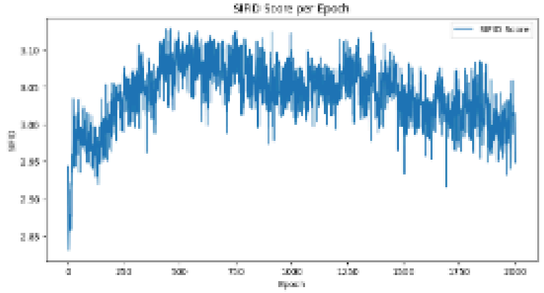
SIFID evaluation



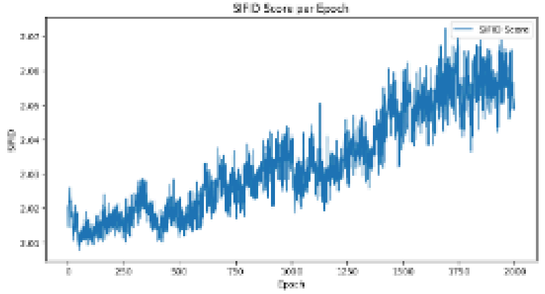
(a) scale 0



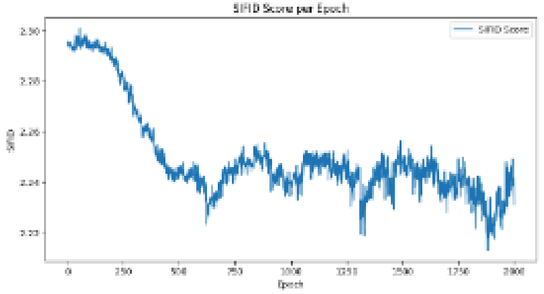
(b) scale 1



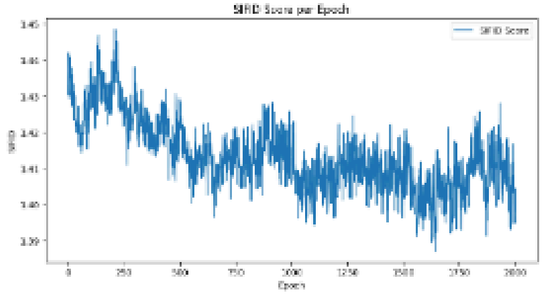
(c) scale 2



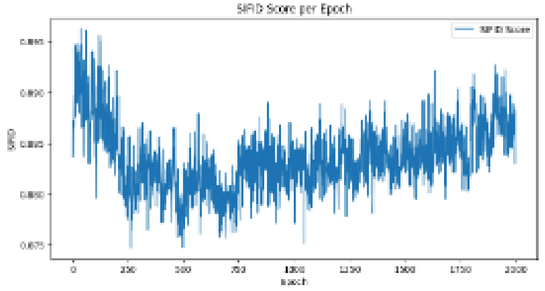
(d) scale 3



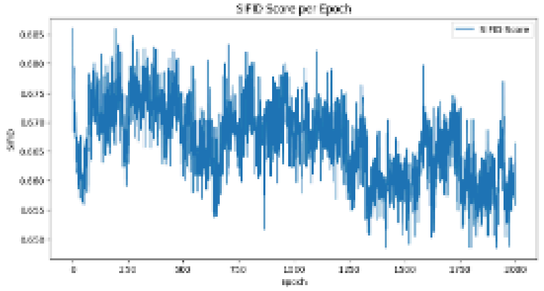
(e) scale 4



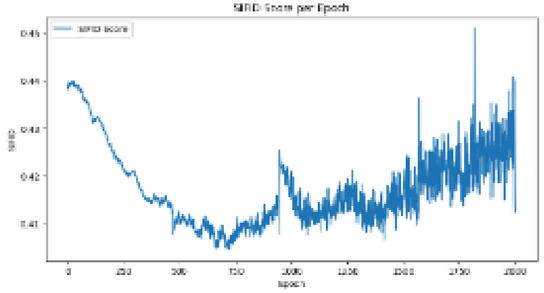
(f) scale 5



(g) scale 6

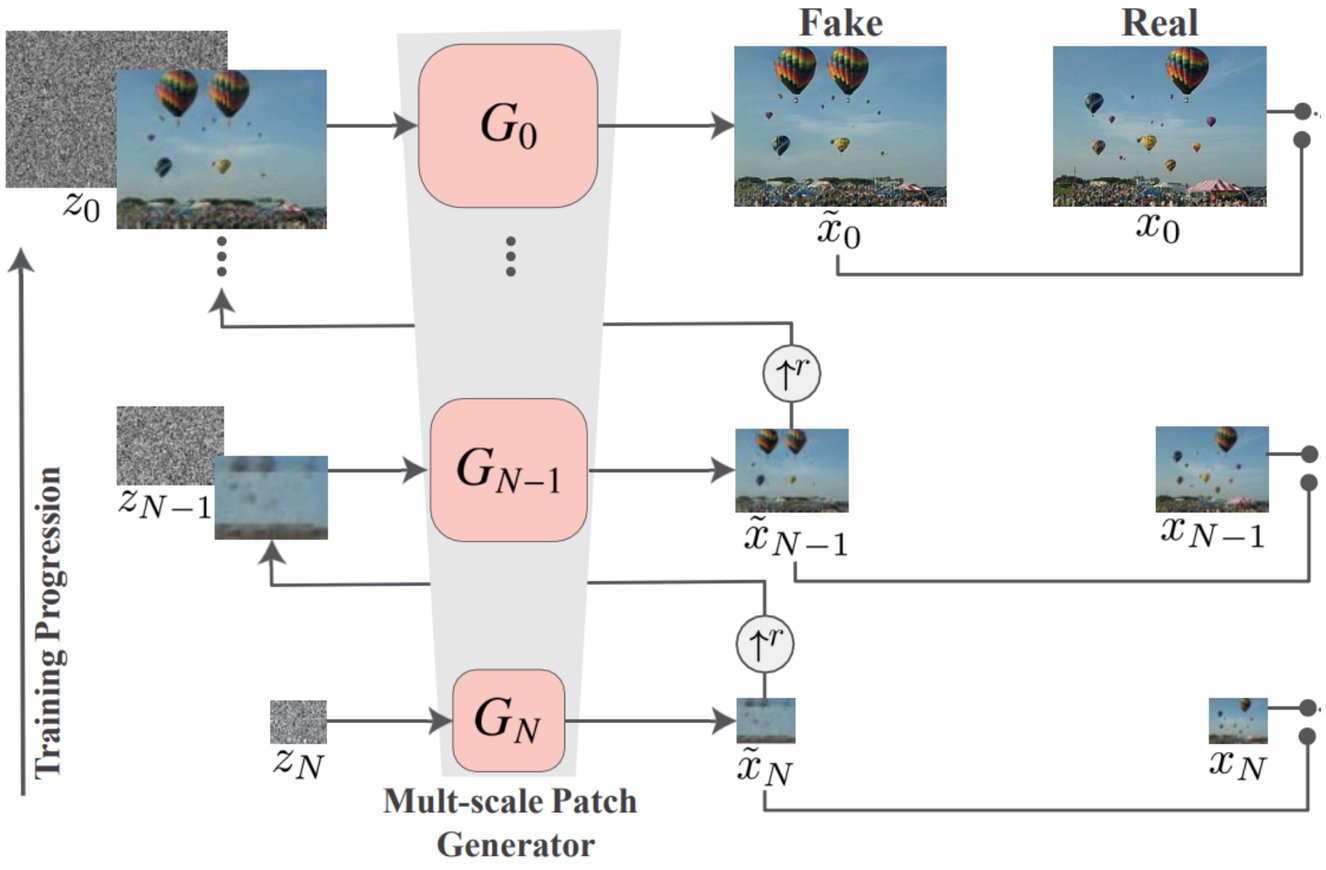


(h) scale 7

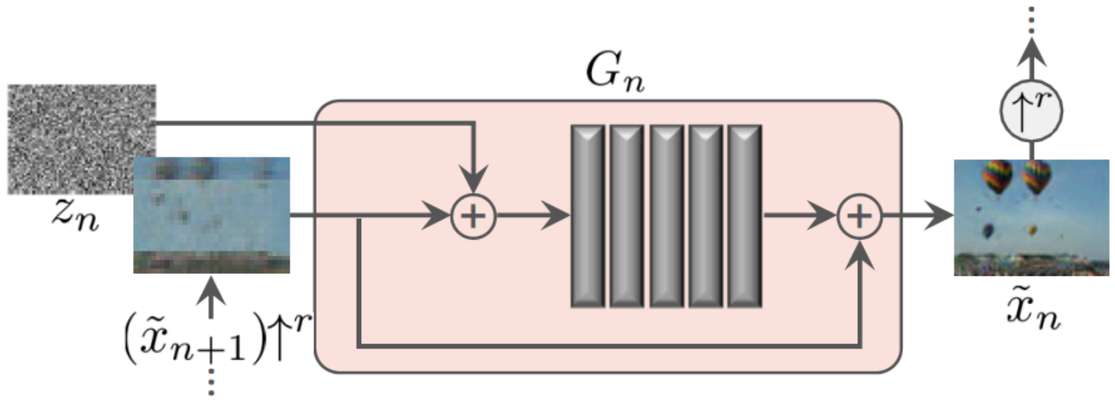


(i) scale 8

Generation



Generation pipeline



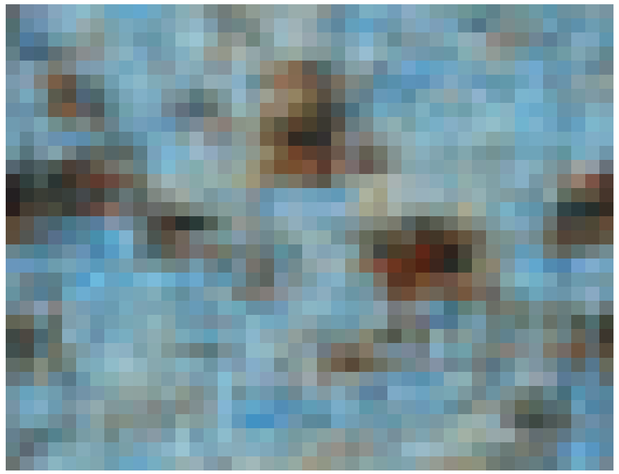
Single scale generation

Generation

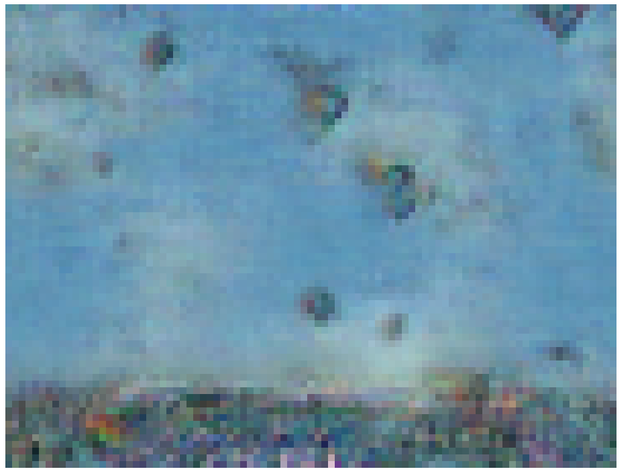
With original implementation

With our implementation

Change the boundary conditions



Scale 1



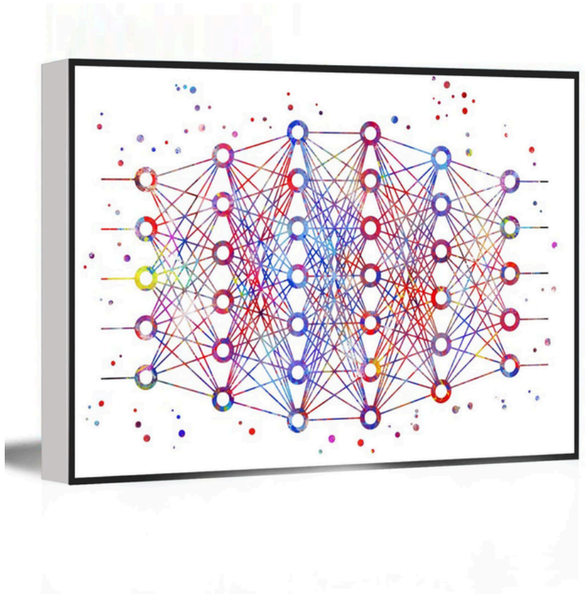
Scale 4

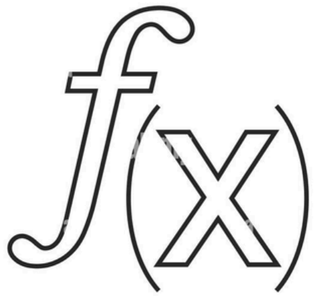
Scale 6

Scale 7

Results after changing Zero-padding by Circular-padding

Training with substitute discriminators





Frechet Distance Loss

This loss is written as :

$$d^2(\mu_r, \Sigma_r, \mu_f, \Sigma_f) = \|\mu_r - \mu_f\|^2 + \mathrm{Tr}(\Sigma_r + \Sigma_f - 2(\Sigma_r\Sigma_f)^{1/2})$$

Where

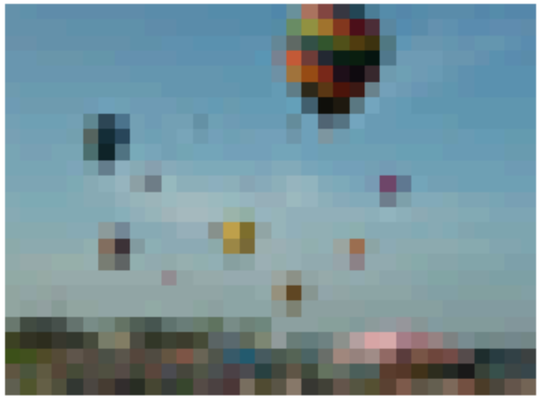
μ_r is the mean vector of patchs from real image

μ_f is the mean vector of patchs from fake image

$\sum_r$ is the covariance matrix of patchs from real image

$\sum_f$ is the covariance matrix of patchs from fake image

Frechet Distance Loss

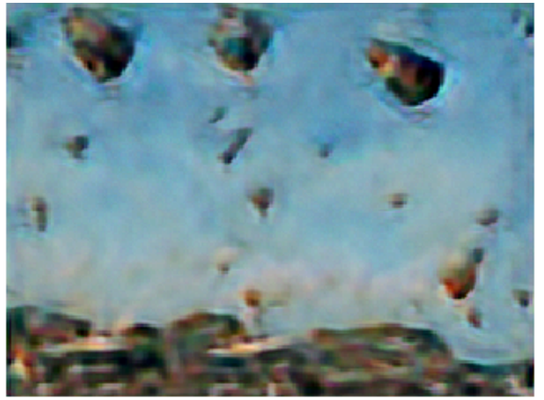


(a) Original image

(b) fake image image

Scale 1

(a) Original image

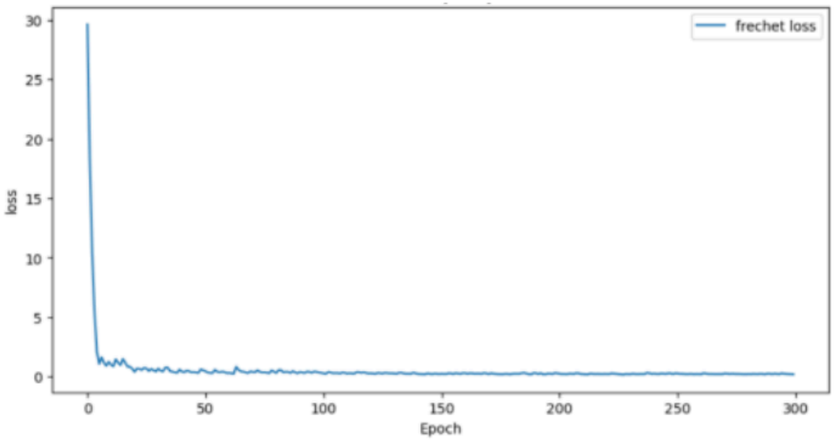


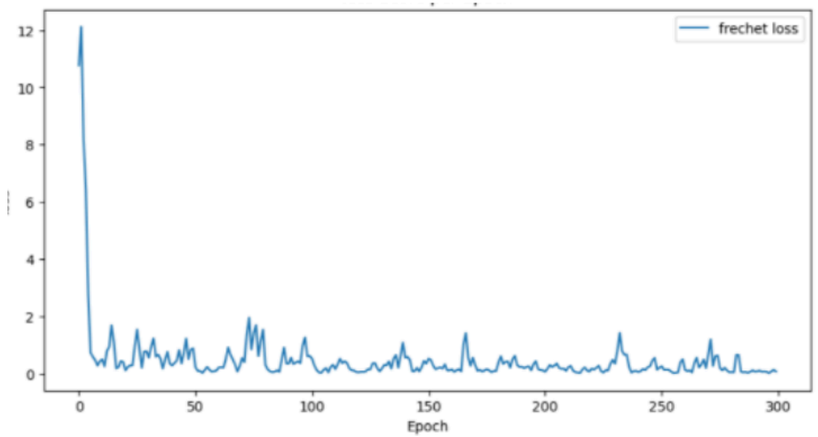
(b) fake image image

Scale 7

Generated images with frechet Loss against real images in different scales

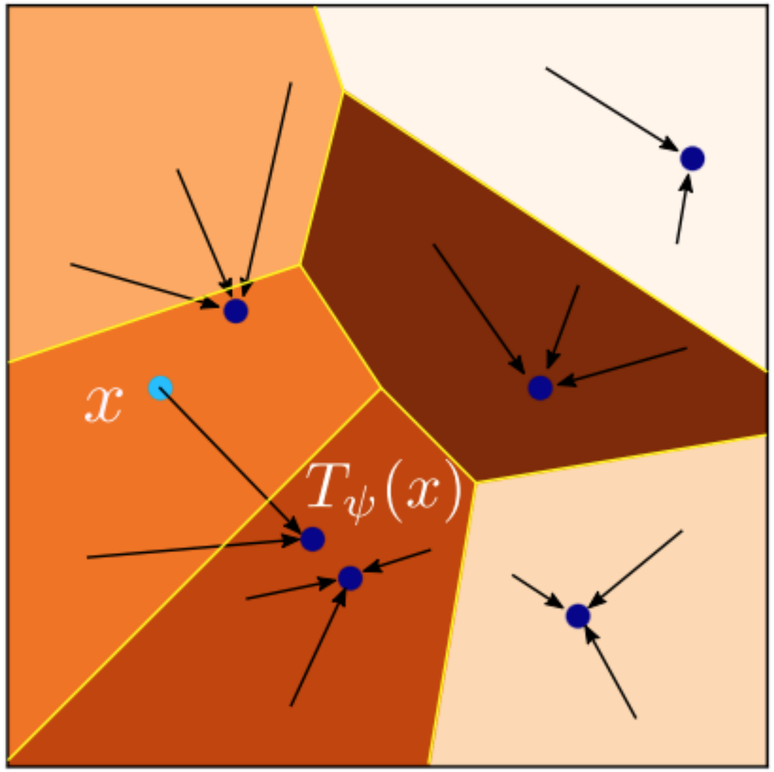
Frechet Distance Loss





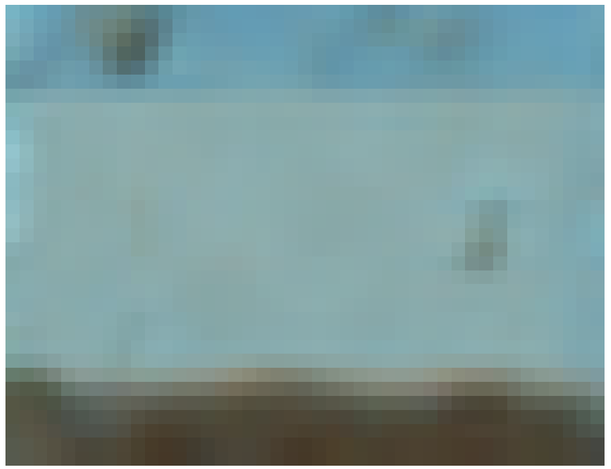
Evolution Loss in scale 1 and 8

Nearest Neighbor Patch Loss



$$NNPL(x, y) = \frac{1}{N_x} \sum_{i=1}^{N_x} \min_{j \in \{1,\ldots,N_y\}} ||x_i - y_j||^2$$

Nearest Neighbor Patch Loss



Scale 1

Scale 4

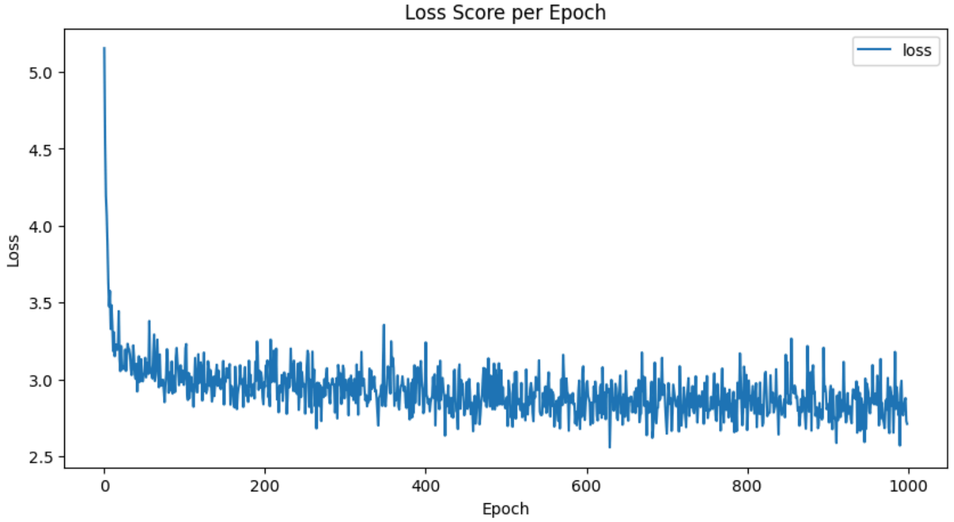
Scale 7

Scale 8

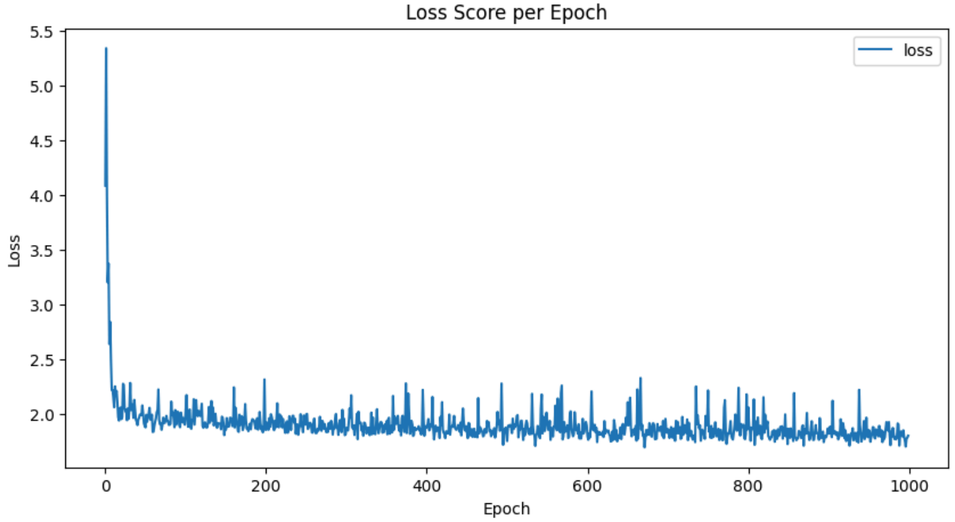
Generated images with NNPL Loss in different scales

Nearest Neighbor Patch Loss

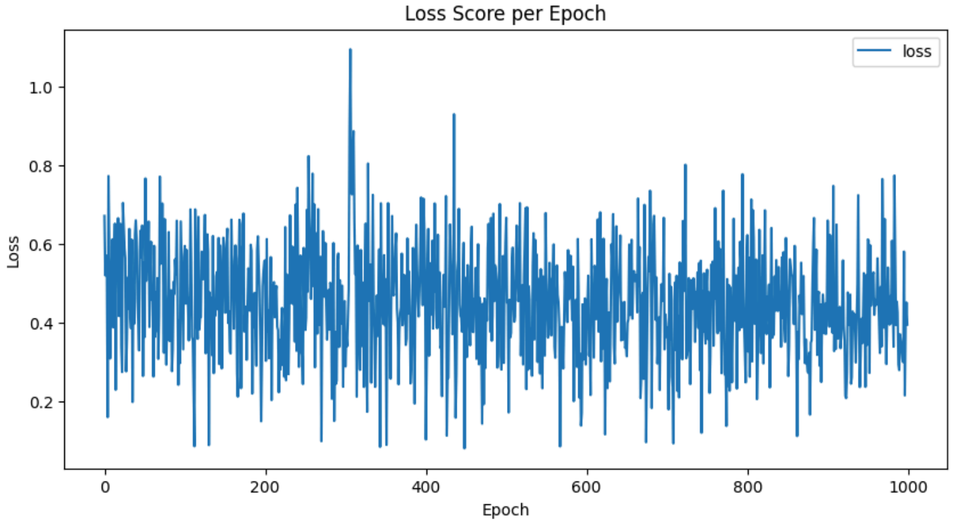
Scale 1



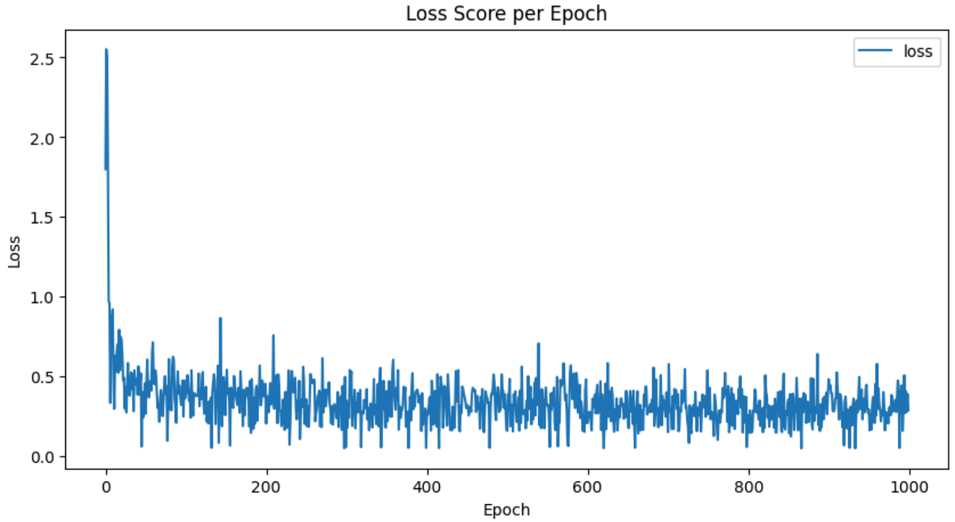
Scale 4



Scale 7

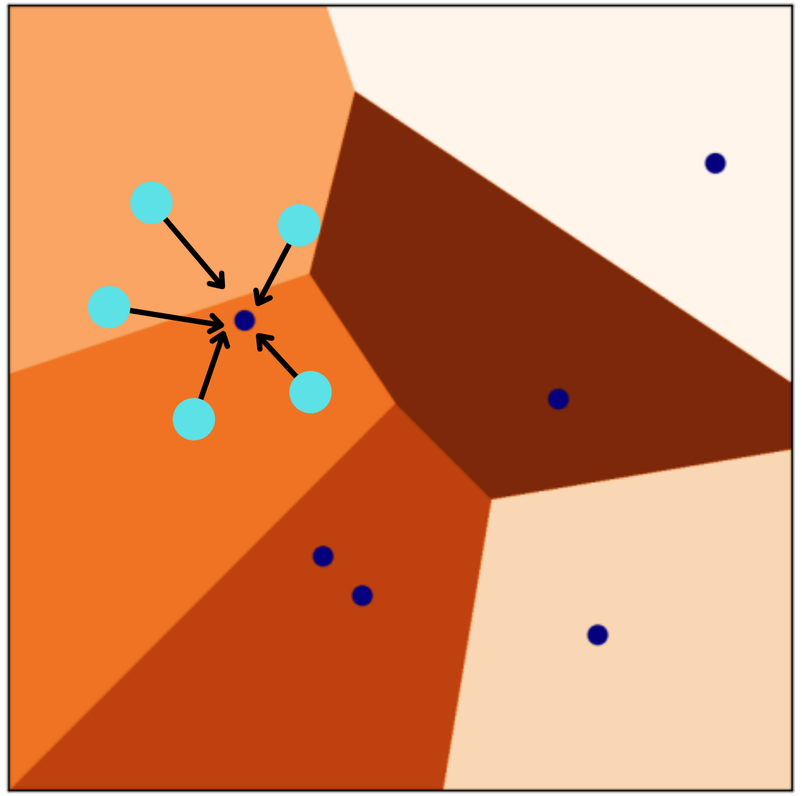


Scale 8



Evolution of NNPL Loss in different scales

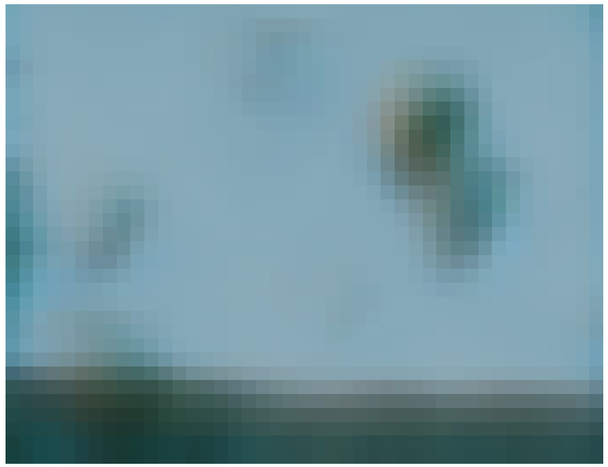
Aligned Nearest Neighbor Patch Loss



$$x' = \phi(x) = \Sigma_y^{1/2} \Sigma_x^{-1/2} (x - \mu_x) + \mu_y$$

$$ANNPL(x, y) = NNPL(x', y)$$

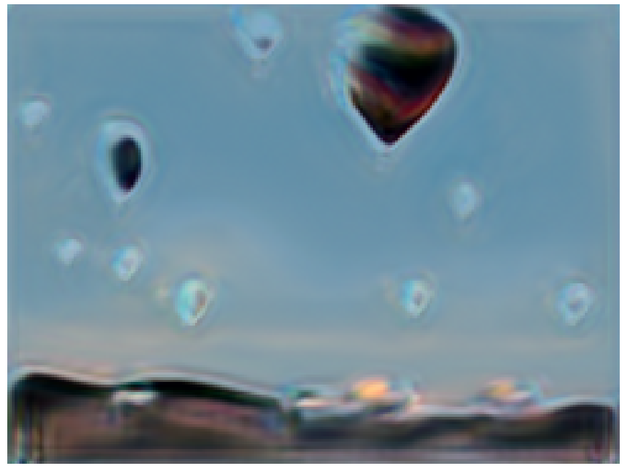
Aligned Nearest Neighbor Patch Loss



Scale 1

Scale 4

Scale 6

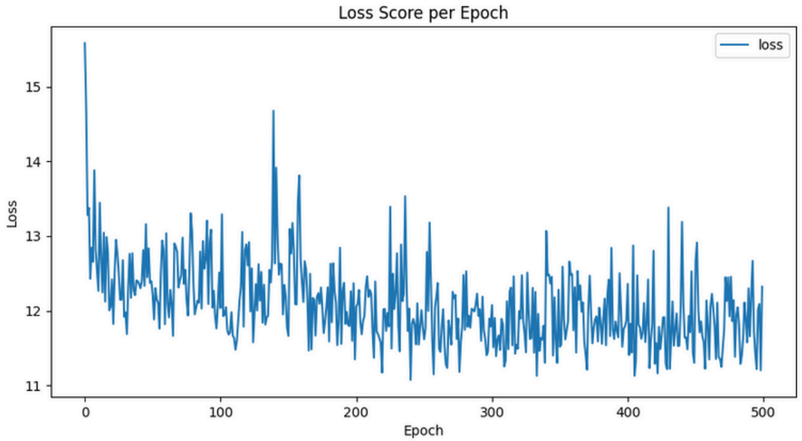


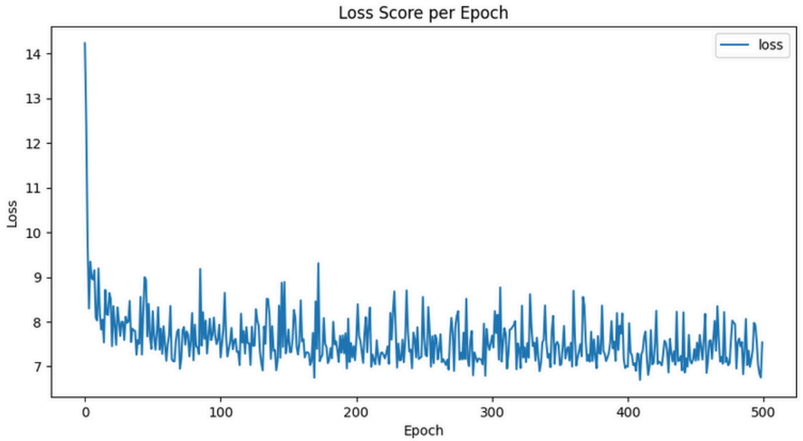
Scale 7

Generated images with ANNPL Loss in different scales

Aligned Nearest Neighbor Patch Loss

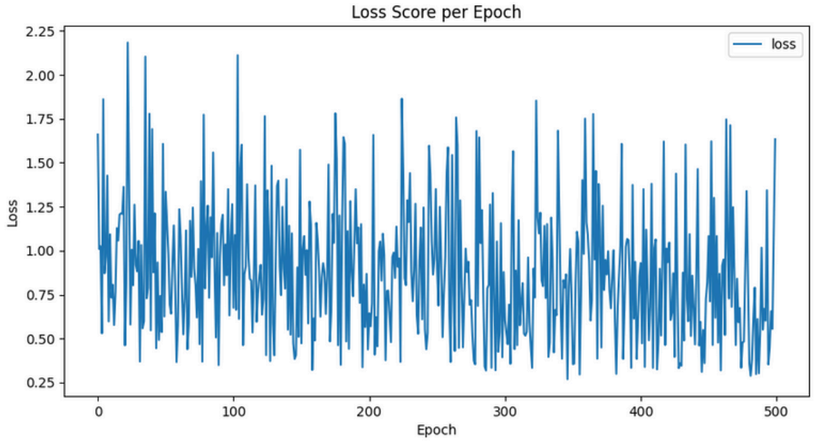
Scale 1

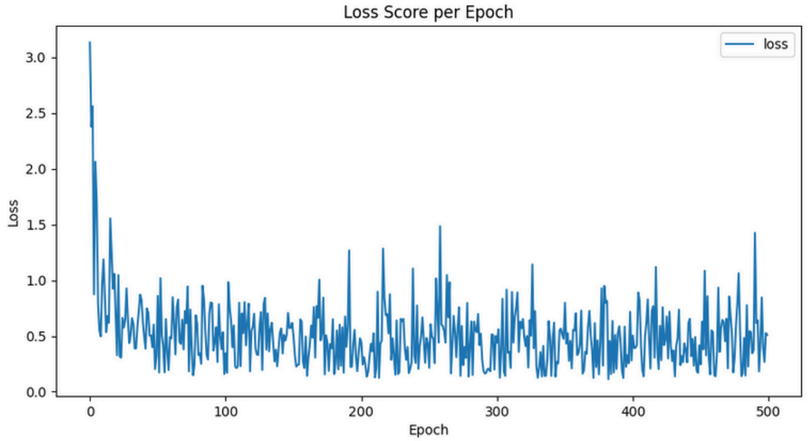




Scale 4

Scale 7





Scale 8

Evolution of ANNPL Loss in different scales

Conclusion