IFT6135 - Assignment 4 (GANs)

Valentin Thomas, Rémi Le Priol, Salem Lahlou
Friday April 20th

I - Generating faces with different architectures

In all these experiments, we used the non-saturating GAN with gradient penalty (regularization factor $\lambda = 0.5$ unless specified otherwise). We update the generator and the discriminator once alternatively.

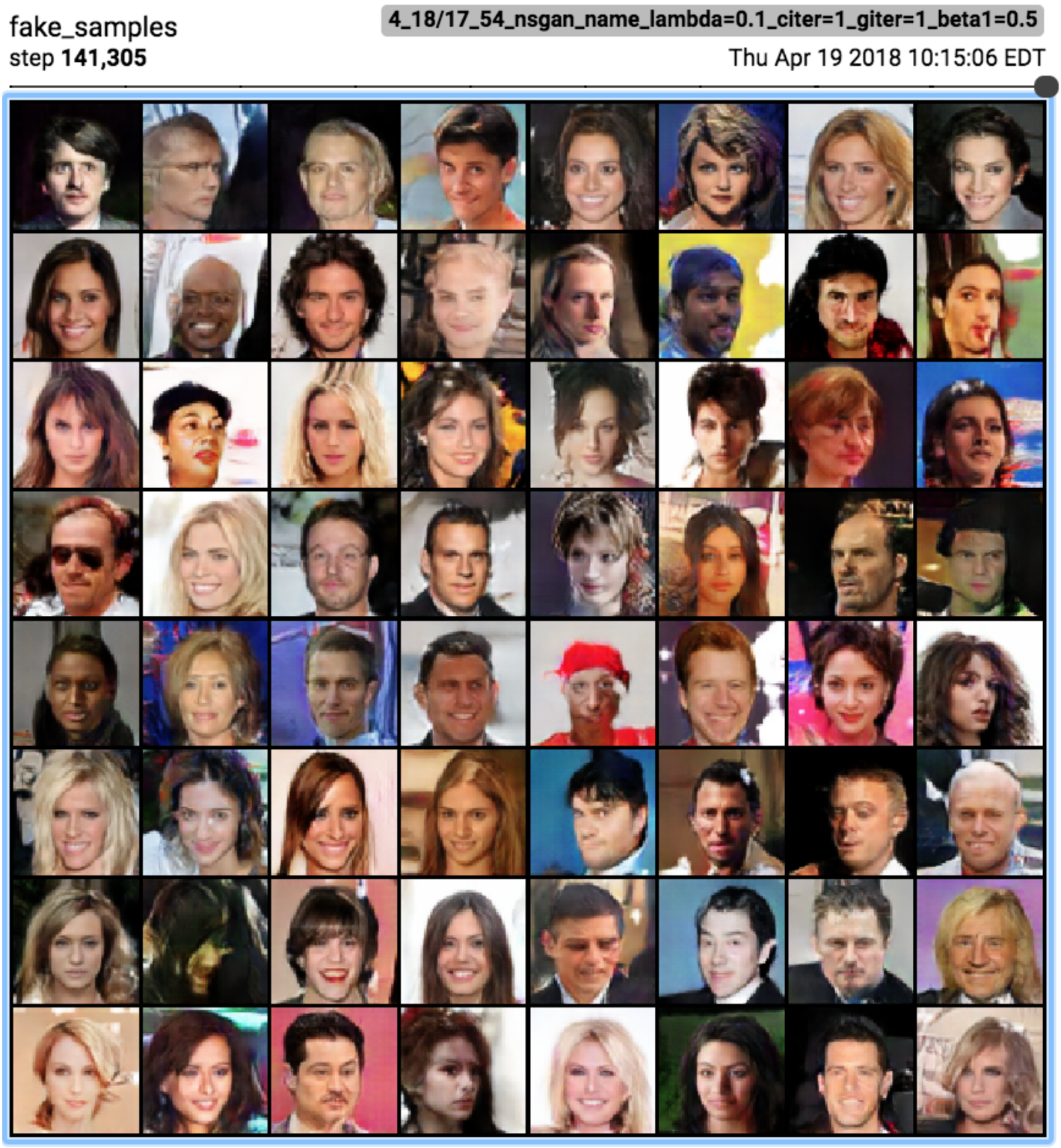
By gradient penalty here we mean $\lambda \cdot \mathbb{E}_{p,q}[||\nabla_x f(x)||^2]$ where f is the critic for a WGAN or the logit in the case of regular GANs. We do not use the interpolation between real and fakes and the -1 to drive the gradient norm to 1 as done in Improved training of WGAN (https://arxiv.org/pdf/1704.00028.pdf) as we did not find it helping.

In the images and plots below, a step corresponds to an iteration of the algorithm on a minibatch of size 64.

(a) Deconvolution

We use the following layer architecture.

```
nn.ConvTranspose2d(channels_in, channels_out, 4, 2, 1, bias=False),
nn.BatchNorm2d(channels_out),
nn.ReLU(True),
```

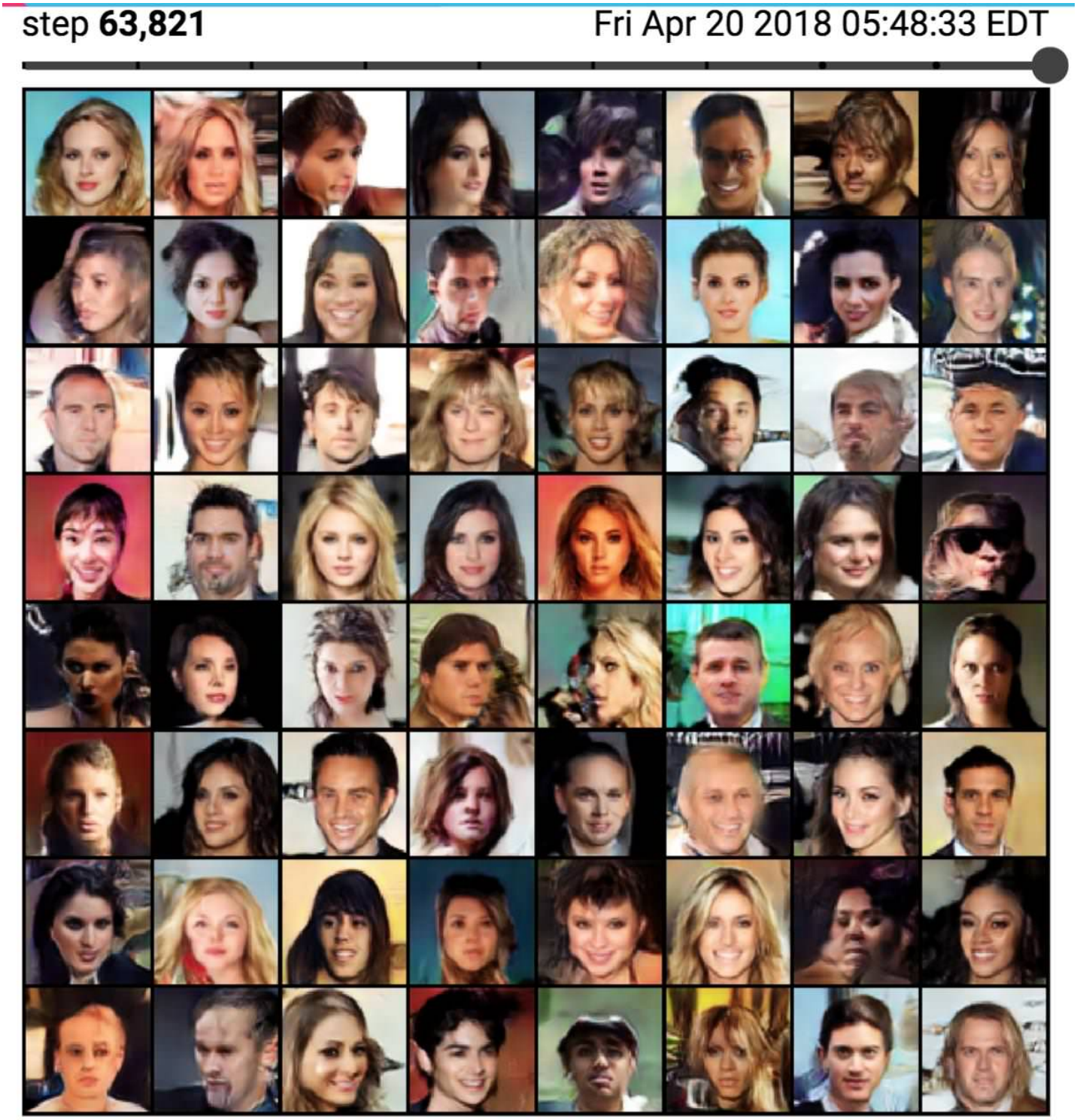


(b) Nearest-Neighbor Upsampling

We use the following layer architecture.

```
        nn.Upsample(scale_factor=2, mode='nearest'),
        nn.Conv2d(channels_in, channels_out, 5, padding=2, bias=False),
        nn.BatchNorm2d(channels_out),
        nn.ReLU(True),
```

We report some samples generated:



(c) Bilinear Upsampling

Code: same as above with `mode='bilinear'`.
We report some samples below.

As explained by Vincent Dumoulin in his Distill article https://distill.pub/2016/deconv-checkerboard/ (https://distill.pub/2016/deconv-checkerboard/), upsampling with deconvolution (eg strided convolution) tends to produce checkerboard artifacts. We do observe that some samples generated with deconvolution tend to be more pixelated than those produced with upsampling.

In the following, we reports results produced with nearest neighbor interpolation.

II - WGAN Variant

We compare our previous model (Non saturating GAN with a Wasserstein GAN).
The WGAN objective can be written as follows

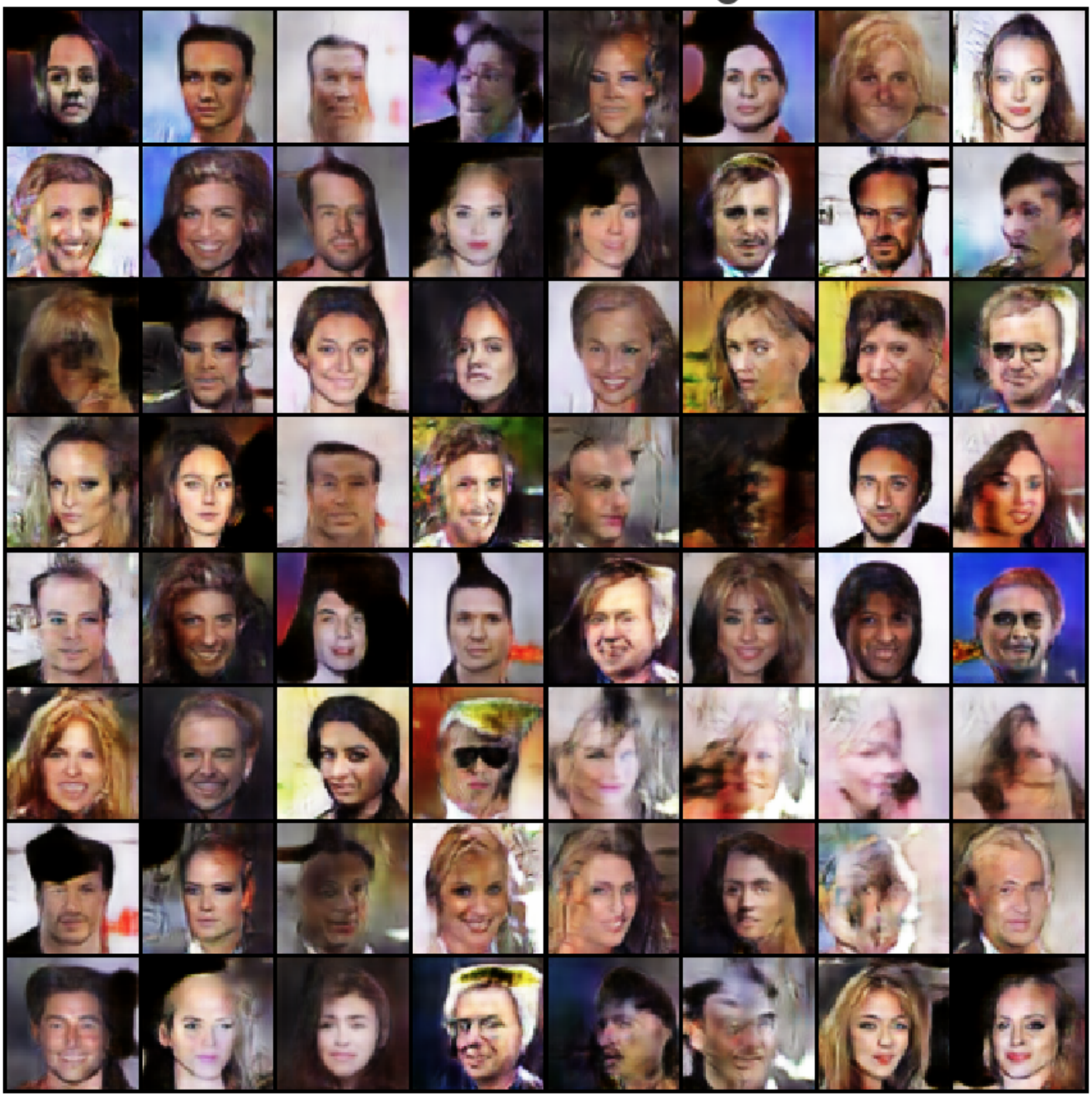
$$\min_{G} \max_{f \in \text{k-Lipschitz}} = \mathbb{E}_{x \sim p}[f(x)] - \mathbb{E}_{z \sim \mathcal{N}(0,1)}[f(G(z))]$$

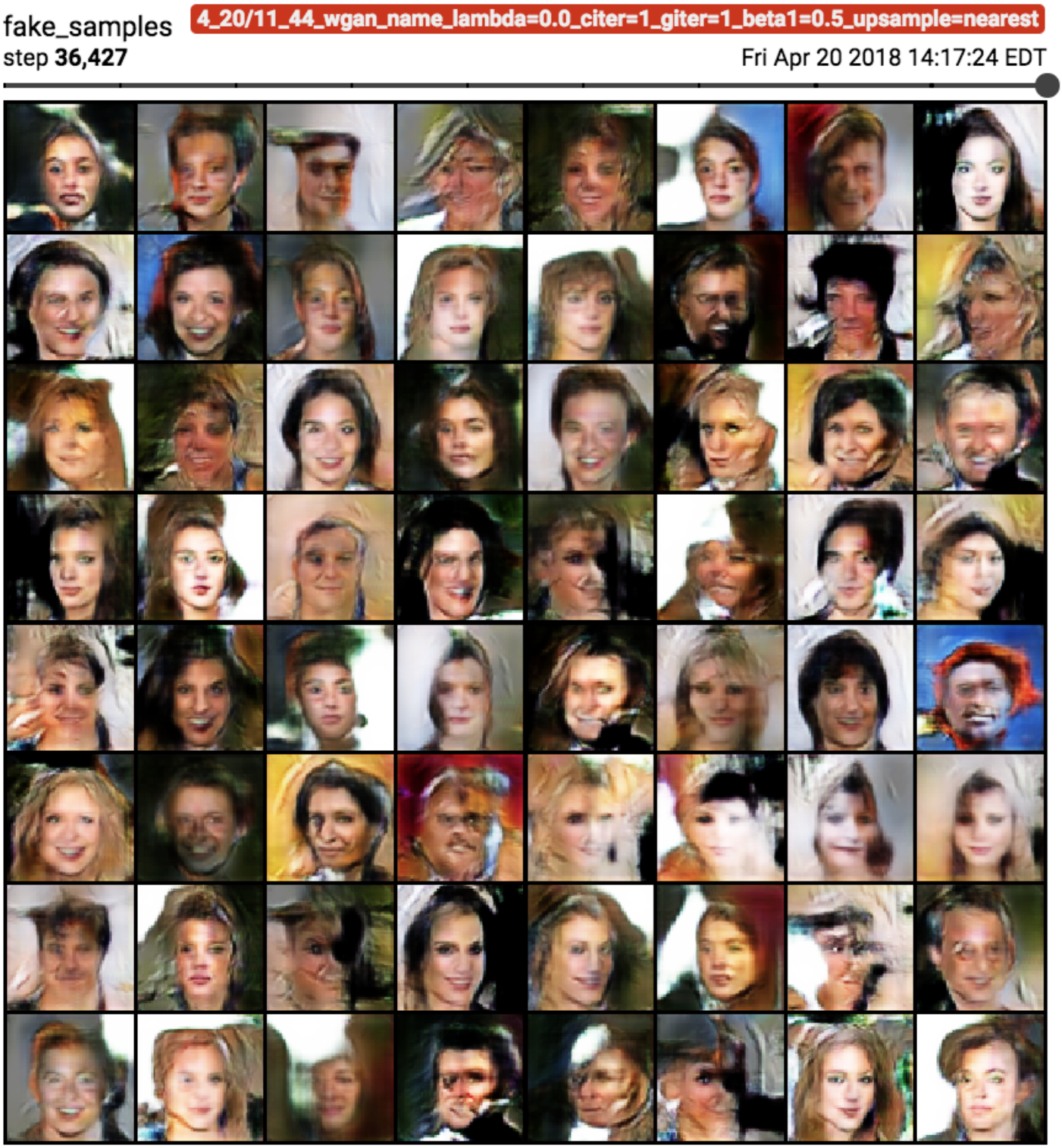
This stems from the Kantorovich-Rubinstein duality (extension of Lagrangian duality to infinite dimensional objects) applied to the definition of the Wassertein distance.

As suggested in the assignment, we enforce the Lipschitz constraint in a hard way by clipping the weights of f. For the sake of comparison, we do not use gradient penalty for this model.

II - 1 Nearest, clip = 0.05, 1 critic iter

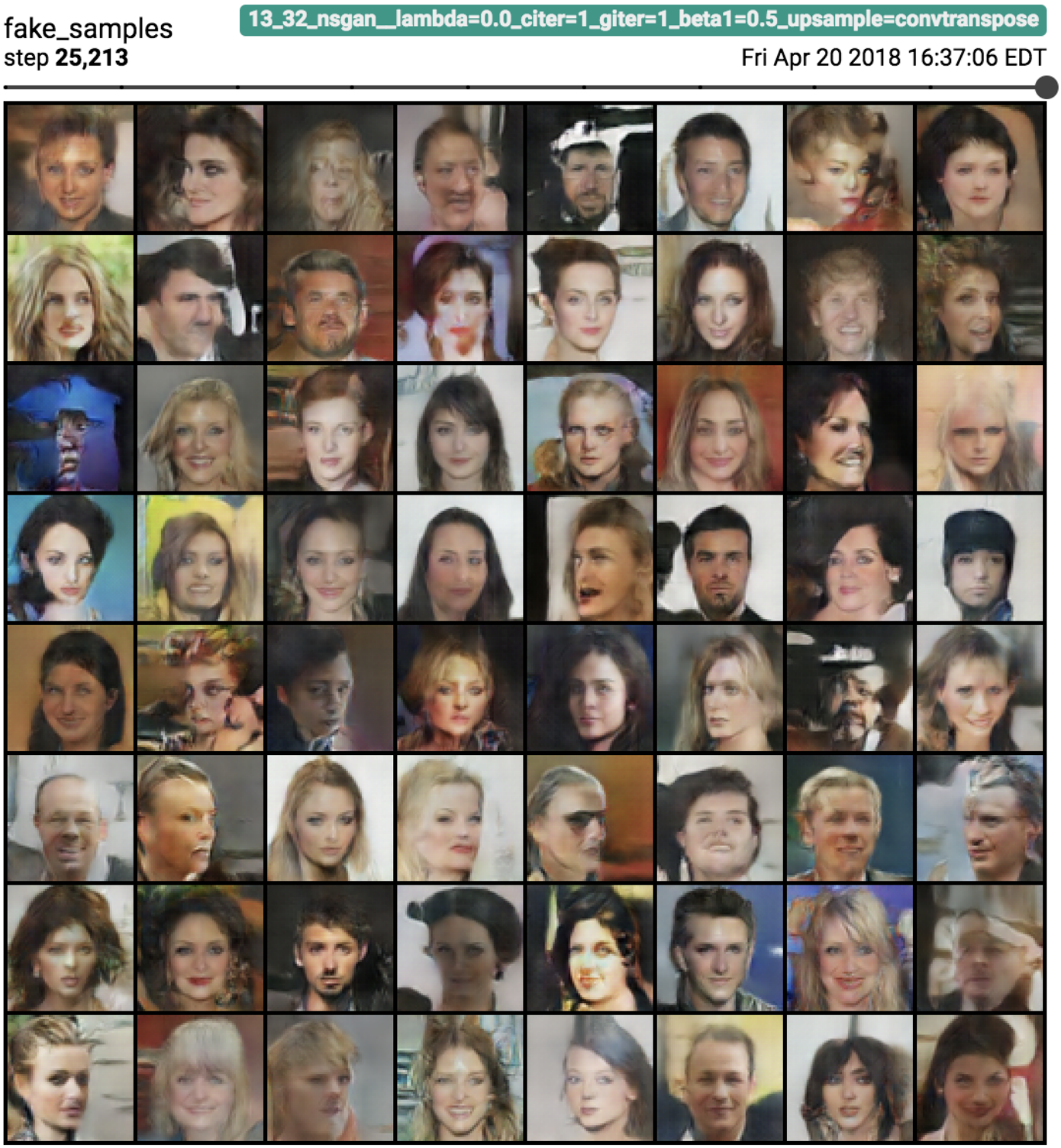
fake_samples 4_20/11_44_wgan_name_lambda=0.0_citer=1_giter=1_beta1=0.5_upsample=nearest
step **25,013** Fri Apr 20 2018 13:29:29 EDT





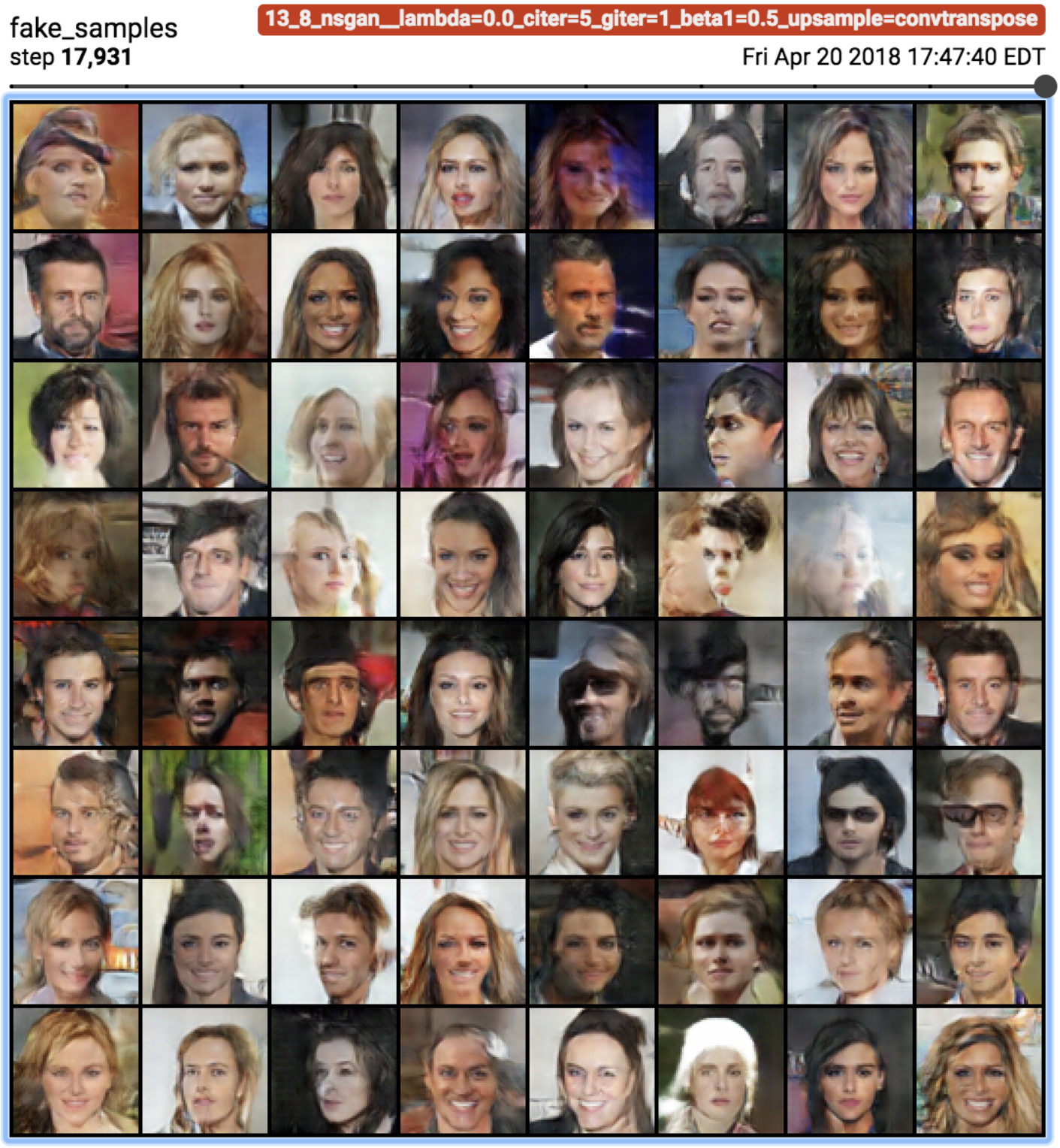
Samples a bit later in training, we seem to have a bit of mode collapse which will disappear i na few steps.

II - 2 DeConv, clip = 0.05, 1 critic iter



II - 3 DeConv, clip = 0.05, 5 critic iter

=



It seems like using 5 critic iterations requires stronger weight clipping. In the original paper (hhttps://arxiv.org/pdf/1701.07875.pdf), they advise to use a weight clipping in $[-0.01, 0.01]$when using 5 critic iterations.

We also observed that with 5 critic iterations, the discriminator's accuracy is near-perfect and the generator fails to make the induced distribution approach the real data generating distribution.

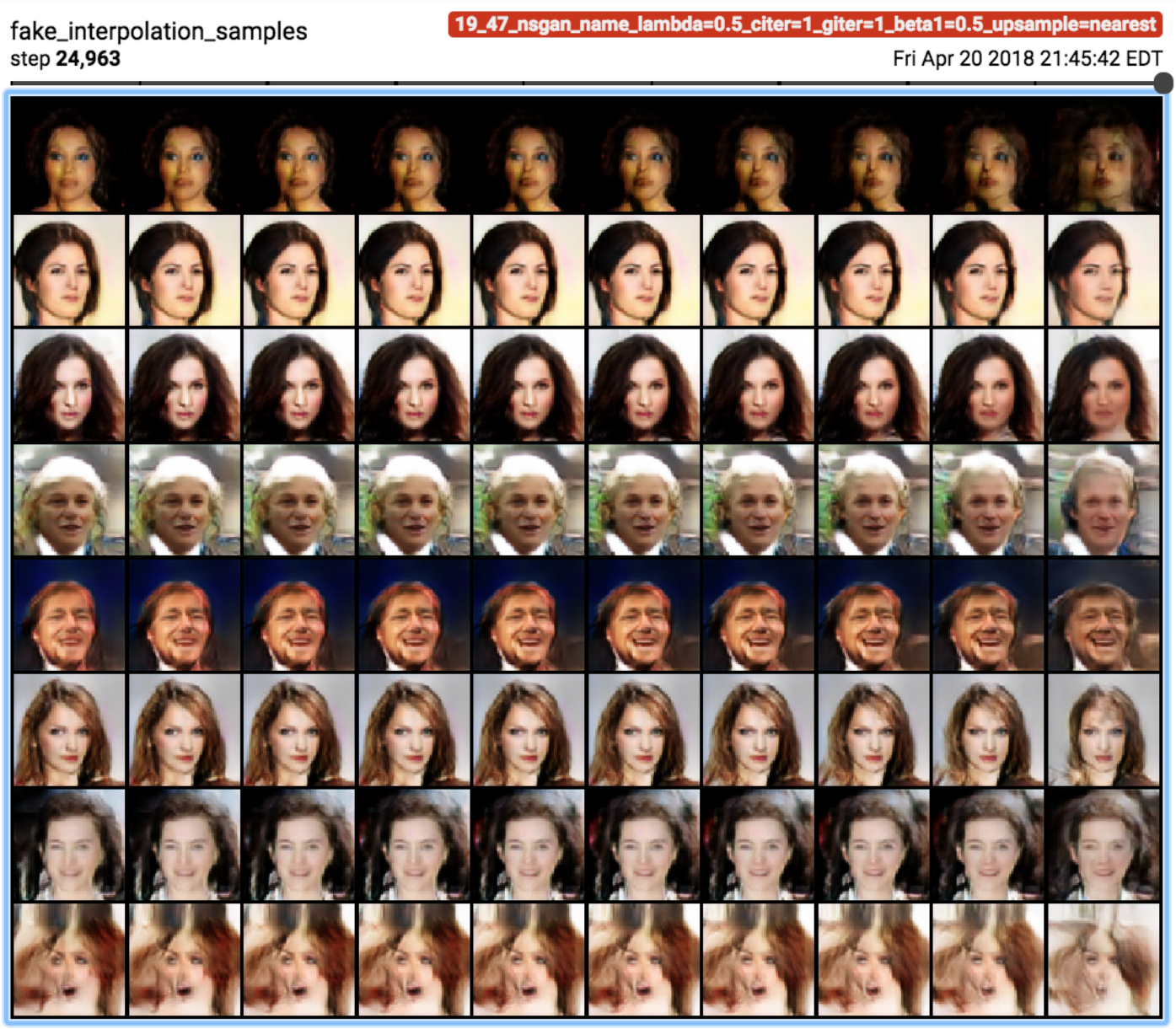
We still observe mode collapse as there are some patterns that do appear in most images.

Gradient penalty seems to play an important role then given the quality of the images we obtaines using W-GAN compared to the previous ones.

III - Qualitative Evaluations

(a) We observe as a general rule of thumb that models trained with the gradient penalty tend to have good results visually. The specific form of the loss used (NLL vs difference of expectations) does not seem to be the main influencing factor.
It seems that the nearest upsampling performs really well, but the ConvTranspose model seems also to be competitive. The best looking samples are probably the ones presented in the first section (NS-GAN with GP), they were also run on more steps than other variants.

(b)



Interpolation in latent space. On each row i, coordinate i of a random noise z was slided from -4 to $+4$. While we do see some slight changes in images, it is not easy to find a specific feature a specific coordinate is changing.

(c) We report interpolations in the latent space (left) and in the sample space (right).

Interpolation in the image space yields non-face images. The main reason for this is that the faces are not necessarily in the same position in the image, and interpolating between two different face attributes (e.g. nose and eye) just makes it look like a blurry superposition of contours.

From a geometrical perspective, interpolation in the latent space should follow the data manifold. On contrary interpolation in the sample space capture points that are far from the manifold.

IV - Quantitative Evaluations

In addition to the Inception Score, we Implemented the Mode Score, based on the Inception_v3 model, pretrained on imagenet.

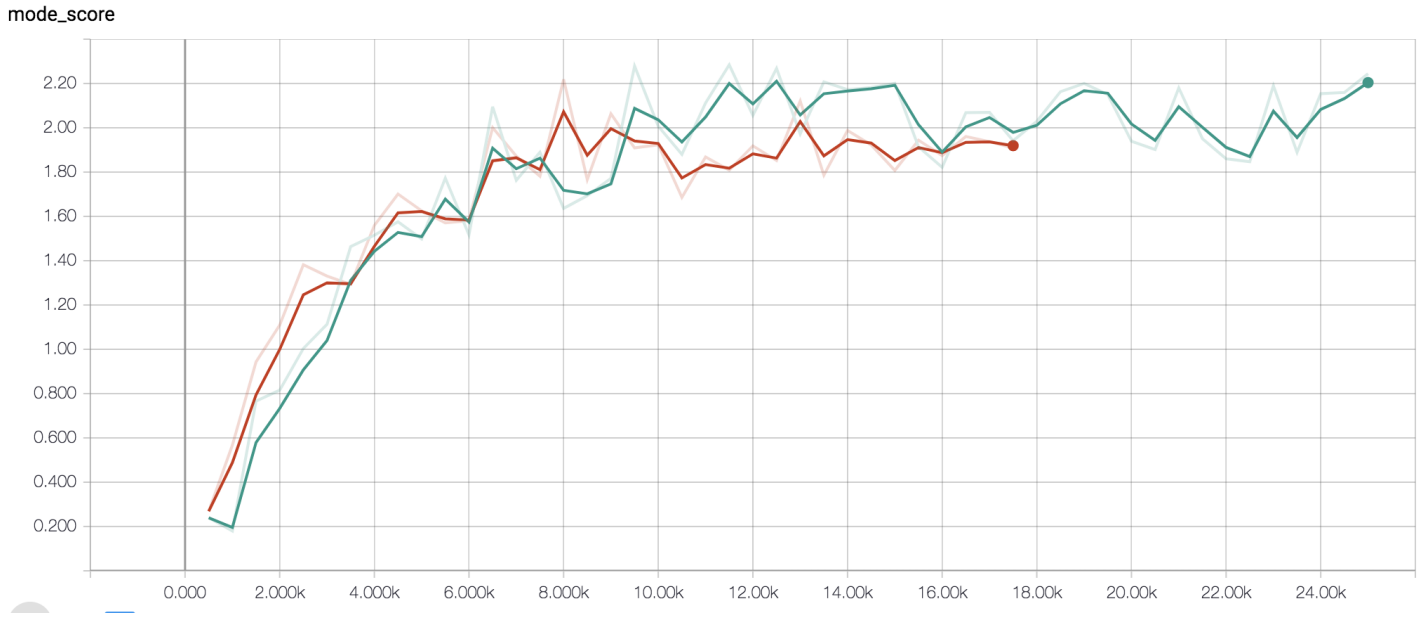
- Inception score : $e^{\mathbb{E}_{x\sim\mathbb{P}_g}[KL(p_{\mathcal{M}}(y|x)||p_{\mathcal{M}}(y))]}$
- Mode score : $e^{\mathbb{E}_{x\sim\mathbb{P}_g}[KL(p_{\mathcal{M}}(y|x)||p_{\mathcal{M}}(y^*))]-KL(p_{\mathcal{M}}(y)||p_{\mathcal{M}}(y^*))}$
  where $p_{\mathcal{M}}(y|x)$ is the distribution of labels predicted by Inception_v3 with input x and $P_{\mathcal{M}}(y) = \int_x p_{\mathcal{M}}(y|x)d\mathbb{P}_g \quad , \quad P_{\mathcal{M}}(y^*) = \int_x p_{\mathcal{M}}(y|x)d\mathbb{P}_r$

The main difference between the two scores is that the Mode Score can capture the dissimilarity between the real data distribution and the distribution induced by the generator, thanks to the $KL(p_{\mathcal{M}}(y)||p_{\mathcal{M}}(y^*))$term

Actually, both these metrics do not work very well on the CelebA dataset because the labels in imagenet are not sufficient to characterize every mode of the training data.

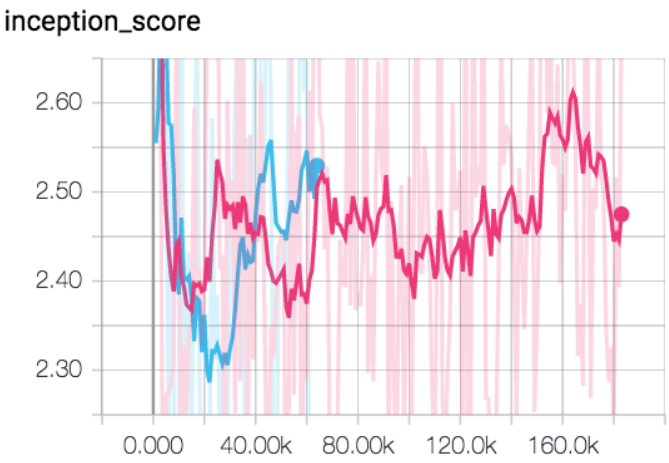
Example : Mode score for W-GAN

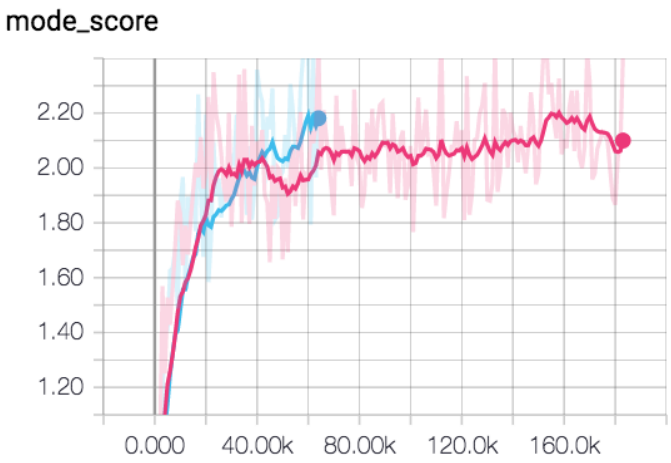
Here are the mode scores obtained for a W-GAN with a conv-transpose architecture. In green, it's with 1 critic iter. In red-ish, it's with 5 critic iter. The x-axis represents the number of steps.



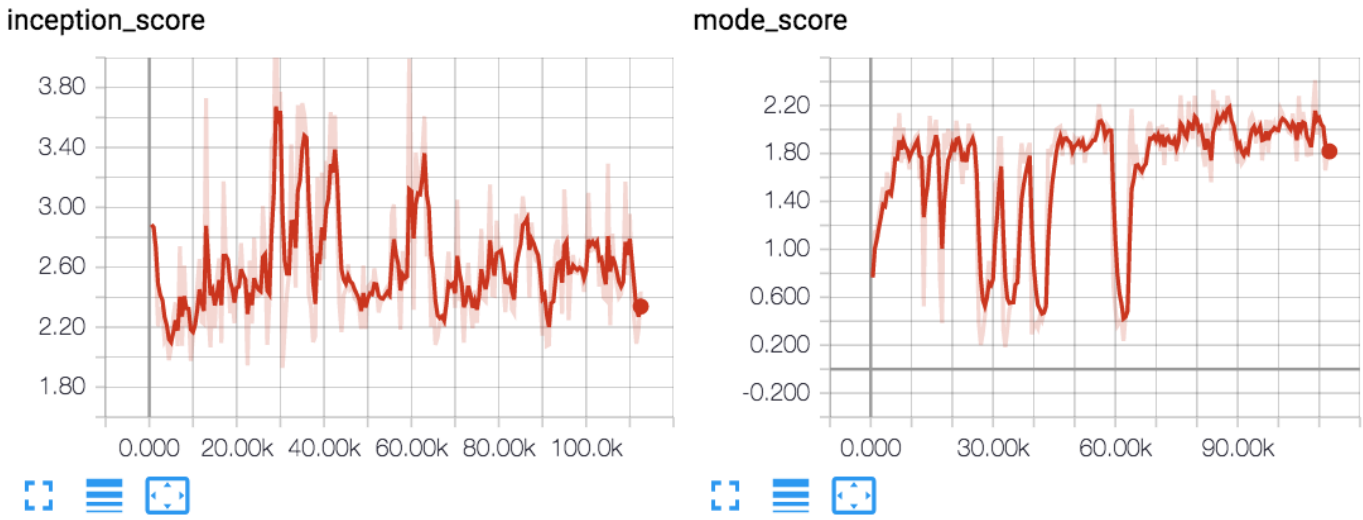
Example : Inception and Mode Scores for NS-GAN

For the Non Saturating GAN, we report the inception and mode scores for 2 different architectures. In pink, we use bilinear upsampling, and in light blue, nearest neighbor upsampling.





Example with mode collapse on a WGAN



This example here is for a WGAN, we report both inception and mode score. While the inception score can be difficult to interpret, we found the mode score more closely related to the visual quality of the samples.
Here, each drop in mode score corresponds to a clear mode collapse in the generator.

We do believe however than none of these metrics are particularly well adapted to the celebA dataset as it is a low resulation dataset of faces while inception was trained on a higher resulution imagenet.