Image-to-Image Schrodinger Bridge for Deblurring task

Prepared by Master's Student:
Alexander Zaytsev

December 21, 2024

1 Problem statement

Image restoration is a crucial challenge in computer vision. Traditional solutions are often ill-posed, and are typically solved using modern data-driven approaches with conditional generation. However, mapping between clean and degraded images can be formulated as an inverse problem that restores the underlying clean signal from the degraded measurement, based on diffusion priors. One way to solve this problem is to start the generative process directly from degraded images, and build a bridge between the clean and degraded data distributions. In this paper, we consider the Image-to-Image Schrödinger Bridge (I2SB) approach and apply it to the task of image deblurring.

All results can be found on github.

github.com/AlZayts/Image-to-Image-Schrodinger-Bridge-for-debluring-task/blob/main/I2ISB.ipynb

2 Brief about math behind it

Firstly, we need to define general Schrödinger Bridge (1932) formulation. Let $X_0 \sim p_{\mathcal{A}}(x), X_1 \sim p_{\mathcal{B}}(x)$ then considering the following forward and backward SDEs:

$$\begin{aligned} \mathrm{d}X_t &= \left[f_t + \beta_t \nabla \log \Psi\left(X_t, t\right)\right] \mathrm{d}t + \sqrt{\beta_t}\, \mathrm{d}W_t \\ \mathrm{d}X_t &= \left[f_t - \beta_t \nabla \log \hat{\Psi}\left(X_t, t\right)\right] \mathrm{d}t + \sqrt{\beta_t}\, \mathrm{d}\bar{W}_t \end{aligned} \tag{1}$$

The functions $\Psi, \widehat{\Psi}$ are time-varying energy potentials that solve the following coupled PDE's.

$$\begin{cases} \dfrac{\partial \Psi(x,t)}{\partial t} = -\nabla \Psi^{\top} f - \dfrac{1}{2}\beta \Delta \Psi \\ \dfrac{\partial \widehat{\Psi}(x,t)}{\partial t} = -\nabla \cdot (\widehat{\Psi} f) + \dfrac{1}{2}\beta \Delta \widehat{\Psi} \end{cases} \tag{2}$$
$$\text{s.t. } \Psi(x,0)\widehat{\Psi}(x,0) = p_{\mathcal{A}}(x), \Psi(x,1)\widehat{\Psi}(x,1) = p_{\mathcal{B}}(x)$$

So, the initial formulation is challenging to implement, as we need $\Psi, \widehat{\Psi}$, which are not easily obtained. Therefore, the task is to simplify the initial model in such a way

that we can avoid solving coupled partial differential equations.From the condition at $t = 0$ and $t = 1$, the multiplication of Ψ and $\widehat{\Psi}$ is the density. Let's assume that each of these variables is also a density. Then, $\nabla \log \Psi$ and $\nabla \log \widehat{\Psi}$ are the score functions for the following stochastic differential equations:

$$\begin{aligned} \mathrm{d}X_t &= f_t\left(X_t\right)\mathrm{d}t + \sqrt{\beta_t}\ \mathrm{d}W_t, \quad X_0 \sim \widehat{\Psi}(\cdot, 0) \\ \mathrm{d}X_t &= f_t\left(X_t\right)\mathrm{d}t + \sqrt{\beta_t}\ \mathrm{d}\bar{W}_t, \quad X_1 \sim \Psi(\cdot, 1) \end{aligned} \tag{3}$$

However, we still cannot sample and code the model.

Let's use Nelson's duality (Nelson, 2020), that is, $q(\cdot, t) = \Psi(\cdot, t)\widehat{\Psi}(\cdot, t)$, where $q(, t)$ is the marginal density of some process of (1). This is more of an assumption than a result, as I did not find it in Nelson's article. Next thing it to condition Nelson duality on X_0 and X_1

$$q\left(X_t \mid X_0, X_1\right) = \Psi\left(X_t, t \mid X_0\right)\widehat{\Psi}\left(X_t, t \mid X_1\right)$$

So, Ψ and $\widehat{\Psi}$ are both normal. This is intuitive and can be shown by fixing the starting point for each equation and then seeing that each point in time adds some independent normal random variable. Therefore, both variables are normal and can be calculated as such.

$$\begin{aligned} q\left(X_t \mid X_0, X_1\right) &= \Psi\left(X_t, t \mid X_0\right)\widehat{\Psi}\left(X_t, t \mid X_1\right) = \\ &= \exp\left(-\frac{1}{2}\left(\frac{\|X_t - X_0\|^2}{\sigma_t^2} + \frac{\|X_t - X_1\|^2}{\bar{\sigma}_t^2}\right)\right) = \\ &= \mathcal{N}\left(X_t; \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \sigma_t^2}X_0 + \frac{\sigma_t^2}{\bar{\sigma}_t^2 + \sigma_t^2}X_1, \frac{\sigma_t^2\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \sigma_t^2} \cdot I\right), \end{aligned}$$

where $\sigma_t^2 := \int_0^t \beta_\tau \mathrm{d}\tau$ and $\bar{\sigma}_t^2 := \int_t^1 \beta_\tau \mathrm{d}\tau$ are analytic marginal variances of the SDEs when $f = 0$.

So, we have the density of the processes (1). This means that we have a distribution of X_t at any point if we know X_0 and X_1. This density is useful for training purposes. Now, we need to find the density for the generation process of X_n, so we should change the conditioning of X_1 on the condition of X_{n+1}. The authors prove by induction that when $f = 0$, this is true.

$$p\left(X_n \mid X_0, X_{n+1}\right) = \mathcal{N}\left(X_n; \frac{\alpha_n^2}{\alpha_n^2 + \sigma_n^2}X_0 + \frac{\sigma_n^2}{\alpha_n^2 + \sigma_n^2}X_{n+1}, \frac{\sigma_n^2\alpha_n^2}{\alpha_n^2 + \sigma_n^2} \cdot I\right),$$

where $\alpha_n^2 := \int_{t_n}^{t_{n+1}} \beta_\tau \mathrm{d}\tau$.

3 Training and generation algorithm

Train. In the previous section, we learned how to get the distribution of X_t for any time t, given the start and end points $q(X_t|X_0, X1)$. Therefore, to train the model, we need to sample data from the dataset (X_0) and time (t), and use the corrupt function ($corrupt(X0)$) to obtain X_1. Then, we can use the learning function to predict the noise.

$$\left\| \epsilon\left(X_t, t; \theta\right) - \frac{X_t - X_0}{\sigma_t} \right\|_2^2 \to \min_\theta$$

Algorithm 1 Training

Input: clean $p_{\mathcal{A}}(\cdot)$ and degraded $p_{\mathcal{B}}(\cdot|X_0)$ datasets
repeat
  $t \sim \mathcal{U}([0,1])$, $X_0 \sim p_{\mathcal{A}}(X_0)$, $X_1 \sim p_{\mathcal{B}}(X_1|X_0)$
  $X_t \sim q(X_t|X_0, X_1)$ according to (11)
  Take gradient descent step on $\epsilon(X_t, t; \theta)$ using (12)
until converges

Algorithm 2 Generation

Input: $X_N \sim p_{\mathcal{B}}(X_N)$, trained $\epsilon(\cdot, \cdot; \theta)$
for $n = N$ **to** 1 **do**
  Predict X_0^ϵ using $\epsilon(X_n, t_n; \theta)$
  $X_{n-1} \sim p(X_{n-1}|X_0^\epsilon, X_n)$ according to DDPM (4)
end for
return X_0

where $\epsilon\left(X_t, t; \theta\right)$ is a neural net.

Generation. For each generation, we use a basic scheme that involves N steps, starting with a corrupt sample X_N. For each step, we predict $X_0^\epsilon = \sigma_t * \epsilon(X_t, t; \theta)$. We then sample the previous step X_{n-1} from the conditional distribution $p(X_{n-1}|X_0^\epsilon, X_n)$.

4 Methodology

4.1 Task Definition

The goal of this study is to assess the performance of the I2SB model in addressing the issue of image deblurring. The task involves removing a Gaussian blur from images with a kernel size of 7 and standard deviation of 2.

Fig. 1 illustrates examples of blurred and ground truth images used in our study.

Figure 1: Example of blurred image

4.2 Dataset and Preprocessing

To conduct the experiment, we utilize the CelebA HQ dataset, which contains 28,000 training images and 2,000 testing images. The images are resized from 256x256 to 64x64 pixels to reduce computational overhead. No other preprocessing was performed. The I2SB model is used for the experiment. We evaluate its performance using LPIPS and FID.

4.3 Model and scheduling

We reparameterize $\epsilon(X_t, t; \theta)$ using CUNet from the ngushchin repository, with noise and embedding channels set to 128 and a base factor of 80, resulting in a total number of 27

million parameters. We use a symmetric quadratic noise scheduler for sampling, and by default, the discrete timestep is set to 100 for generation.

To train the model, we used 50 epochs, with a batch size of 32, starting with a learning rate of 3e-4 and decreasing it to 1e-5 by epoch. Each epoch took approximately 6 minutes on an T4 GPU, so the overall training time was around 5 hours.

Figure 2: Samples from dataset

5 Results

In order to check how model works we used baseline as blurred samples.

Metric	Baseline	I2SB
LPIPS	0.299	0.049
FID	571.90	157.29

Table 1: Comparison of metrics

So, by LPIPS and FID results are much better than baseline, so it showing that sanity check is complete.

So, based on the LPIPS and FID scores, the results are much better than the baseline, indicating that the sanity check has been completed. However, there are still some issues with the results, as the generated images sometimes have a very high FID and are far from a human-like face, and the images are still blurry and not smooth, causing the FID to increase. Interestingly, despite these issues, a significant number of images turn out to be quite good-looking. Also, we conducted an experiment on out-of-distribution generation in Fig. 7. The results were promising, although there is still room for improvement. You can see that the generated images have better facial features than the blurred images, and they have a certain similarity to the original images.

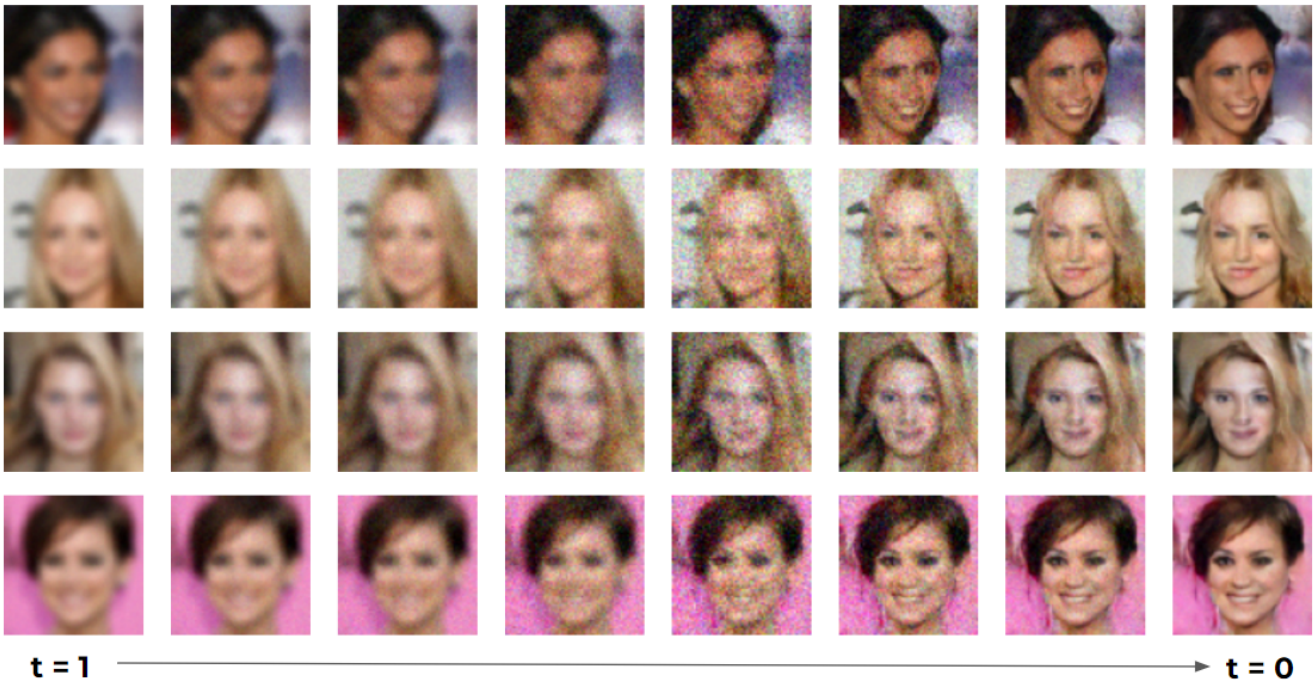

Figure 3: I2ISB results: trajectory of generation

6 Conclusion

In this work I used Image-to-Image Schrodinger Bridge for difficult deblurring task, that a lot of information of original photo is lost. Advantage of this method that it has simple algorithm for training and generation. Also, it is quite efficient, as you need to sampling from two distribution to achieve result. But to dot that, a lot of simplification of model were used, what has direct influence on a result. For example, setting $f = 0$. One way to improve results is to take neural net with more parameters (27M were used) and also take more timesteps (100 were used). Also, other metrics should be used that include sharpness result, so it will be possible to compare how sharp resulting generating in comparison to different levels ob blur.

We also conducted an experiment using out of distribution generation. The generated images were not as sharp as the original ones, but they still improved the overall performance compared to the blurred images.

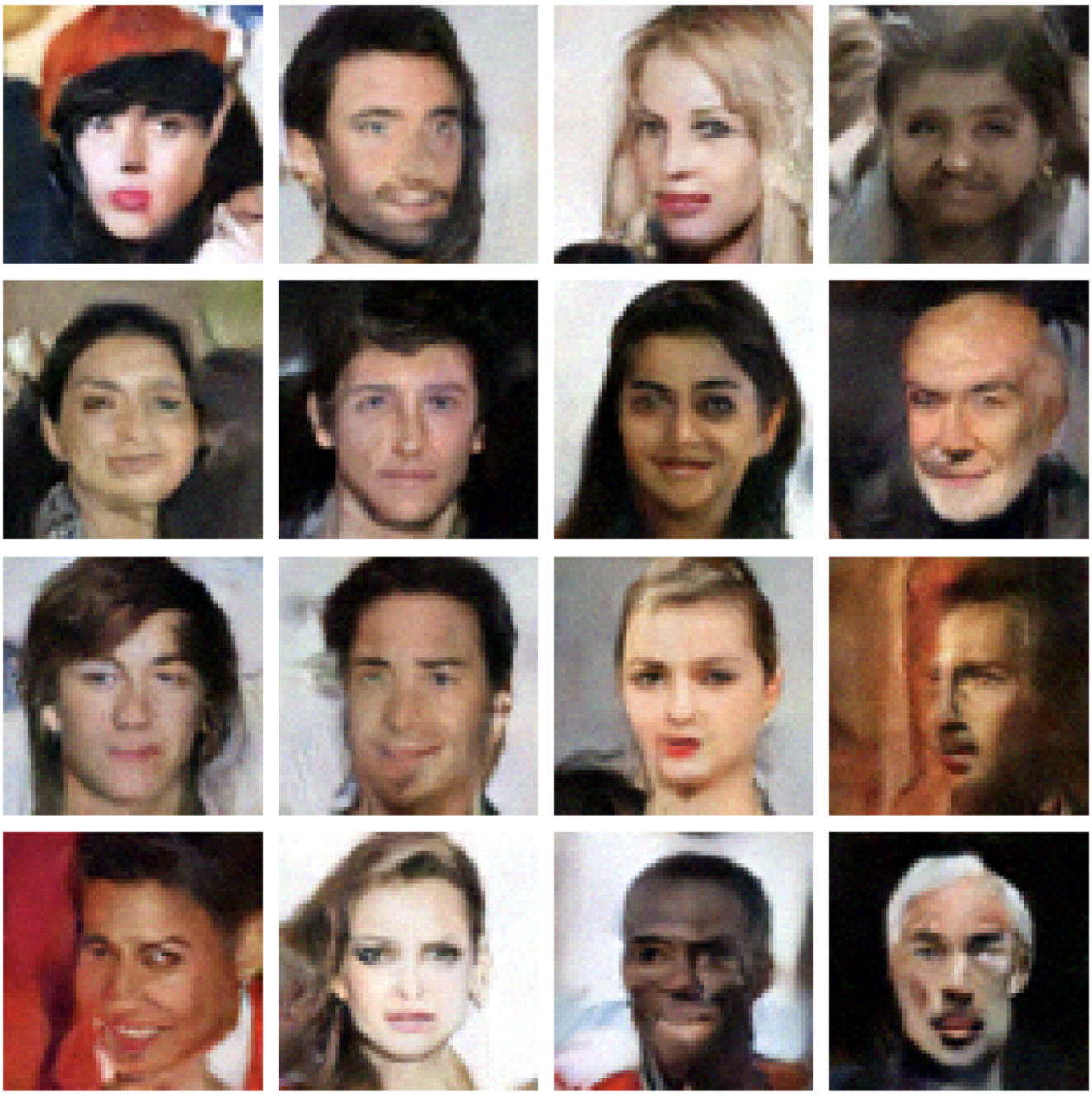

Figure 4: Results of deblurring №1

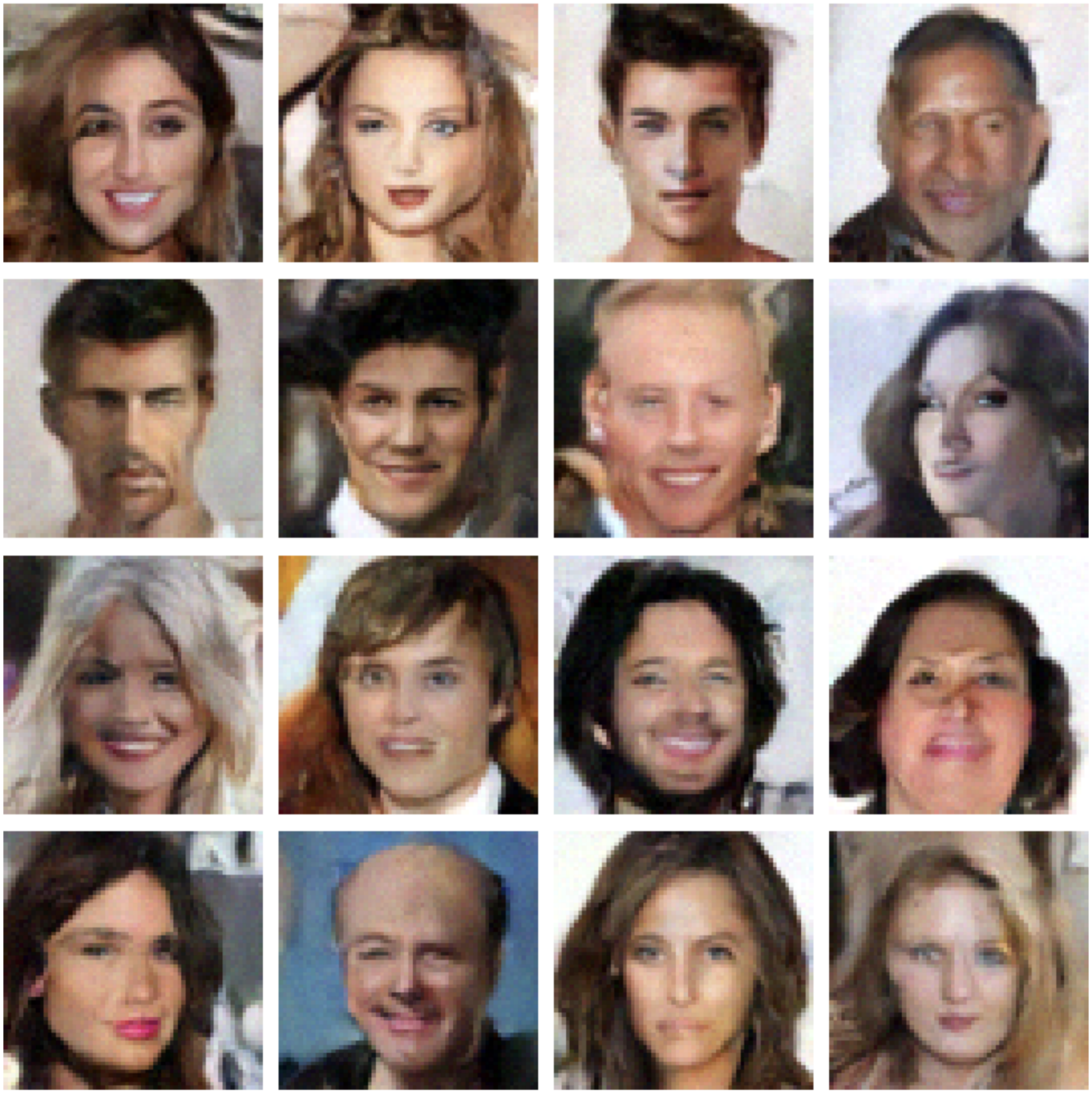

Figure 5: Results of deblurring №2

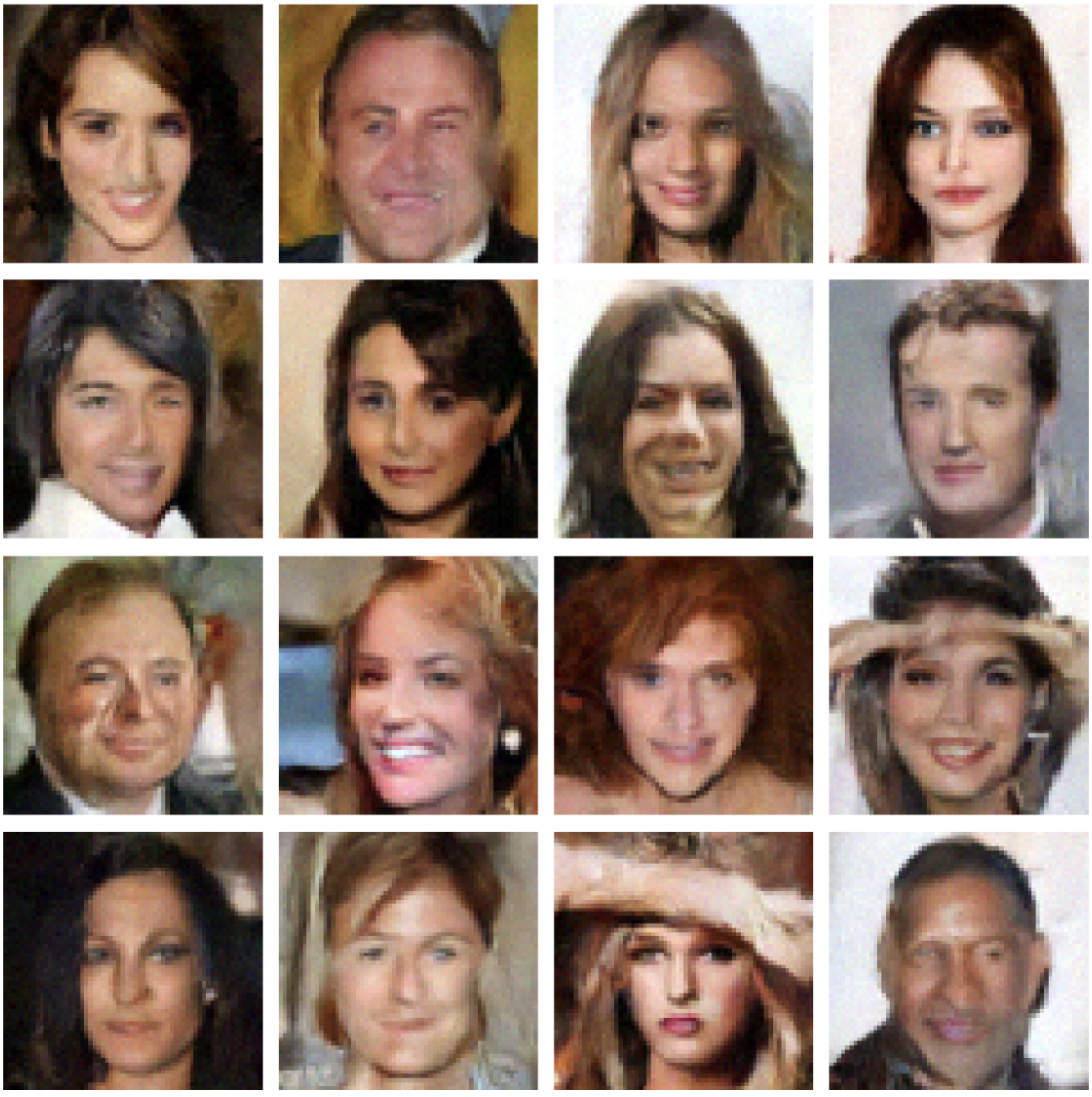

Figure 6: Results of deblurring №3

Figure 7: Out of distribution results