Conditional Generation for Inverse Problems and Class/Text-Based Conditioning

Adam Imdieke (imdie022@umn.edu)

Abstract—This report contains the implementation and results for several inverse problems and conditional generation methods for modern diffusion models. Section 1 covers: "Posterior Samplers for Inverse Problems", Section 2 covers: "Classifier & Classifier-Free Guidance", and Section 3 covers: "Text-Based Conditioning with Stable Diffusion". Each section contains implementation details, results, and analysis/discussion of the findings.

A. Posterior Samplers for Inverse Problems

for each of the following tasks, I got results for three images from each of the two datastes, for each of the four inverse problems. I will only be including one image from each dataset for each task in the report, but the full results are included in the zip file submission. Please refer to the figures for the images corresponding to each method and dataset. I will report the PSNR, SSIM, and LPIPS for the images shown in the report, but the full results across all three images are also included in the zip file submission.

I used my lab's server to generate the images for this task, which has an NVIDIA A6000, with 1TB ram and 128 CPU cores.

1) A: Predict $\hat{x_0}$ and sample ddim: I implemened the DDIM from HW3, the results are shown in Figure 1. The images generated look qualitatively similar to those from HW3, and the model for each dataset creates images from that distrobution.

2) B: Implement and compare posterior samplers: I found that an ILVR weight of 0.8 worked well during initial tests, so I used that value for all tasks. For this task I inlcuded the results from CelebA-HQ and ImageNet datasets in Figures 2 and 3.

The CelebA model had an average performance of:

- Time: 50s
- Weight: 0.8
- SRx4:
  PSNR: 30.97
  SSIM: 0.880
  LPIPS: 0.0729
- SRx8:
  PSNR: 26.44
  SSIM: 0.7464
  LPIPS: 0.1243
- 80% random inpainting:
  PSNR: 20.86
  SSIM: 0.5541
  LPIPS: 0.4891
- 128x128 box inpainting:
  PSNR: 20.33
  SSIM: 0.8244
  LPIPS: 0.1231

The ImageNet model had an average performance of:

- Time: 176s
- Weight: 0.8
- SRx4:
  PSNR: 25.33
  SSIM: 0.735
  LPIPS: 0.2653
- SRx8:
  PSNR: 22.10
  SSIM: 0.5647
  LPIPS: 0.2642
- 80% random inpainting:
  PSNR: 15.95
  SSIM: 0.1889
  LPIPS: 0.9146
- 128x128 box inpainting:
  PSNR: 17.23
  SSIM: 0.7826
  LPIPS: 0.2218

The ILVR method performed the best out of all the methods I implemented. If I had to guess, this is because the direct projection step makes the best use of the information from the measurement, and keeping the diffusion model step separate allows it to not be biased by the measurement.

Across the four tasks, the celeba model looked the best visually, and had higher PSNR and SSIM values. The majority of the results I will be talking about qualitatively are from the CelebA model, as the ImageNet model were noiser on average.

For the super-resolution tasks, The model did quite admerably, with the SRx4 task looking very close to the original image, and the SRx8 suprisingly close given how visually different the low-res input was. For the random inpainting task, The noise was still present in the output, but it was noticably cleaner than the input and the noise was around the edges. For the box inpainting task, while the face that was inpainted did not look like Bryan Cranston, I could believe that it was a face. Given the amount of missing information, it is reasonable that the model would not be able to reconstruct Bryan Cranston exactly.

Because of the amount of missing information in the last two tasks, it could be argued that there exist many plausable images that could fit the compresed measurement, so the model should not reasonalby be expected

Fig. 1. Unconditional samples from CelebA-HQ (top row) and ImageNet (bottom row) pretrained ADM models using DDIM sampling with 1000 steps.

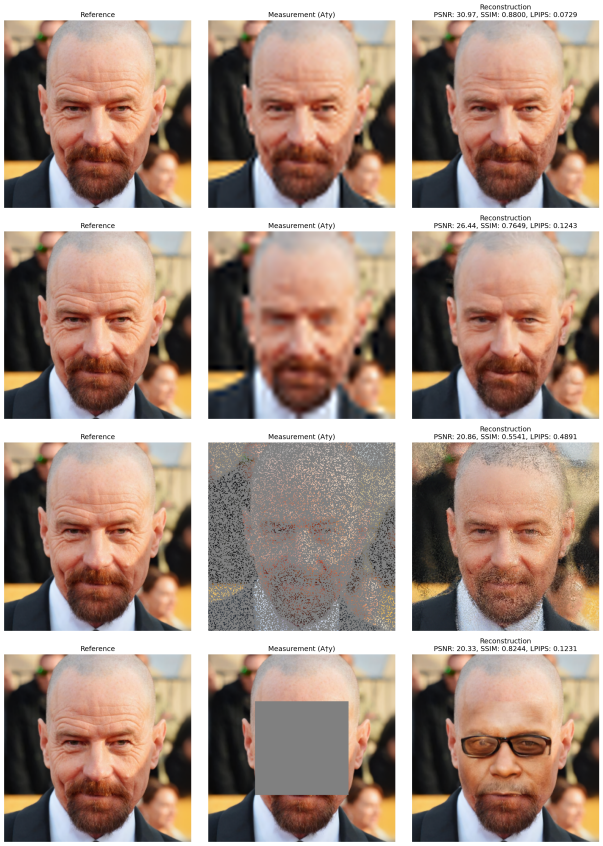

Fig. 2. ILVR Reconstructions on CelebA-HQ for SRx4, SRx8, 80% random inpainting, and 128x128 box inpainting.

Fig. 3. ILVR Reconstructions on ImageNet for SRx4, SRx8, 80% random inpainting, and 128x128 box inpainting.

to reconstruct the original image exactly. However, the model still did a good job of creating images that were consistent with the measurements and looked realistic.

3) C: Manifold Constrained Gradient (MCG): MCG produced some very sharp results for the super resolution tasks, as seen in Figures 4 and 5, but this also resulted in very notable artivacts. These are much worse than in the ILVR results, and the PSNR and SSIM values reflect this, especially for the SRx4 task on the ImageNet model. This is likely because the gradient step pushes the image to be more consistent with the measurement, but might create non realistic images.

For the 80% random inpainting task, the result was much cleaner, with no residual noise, and that is reflected

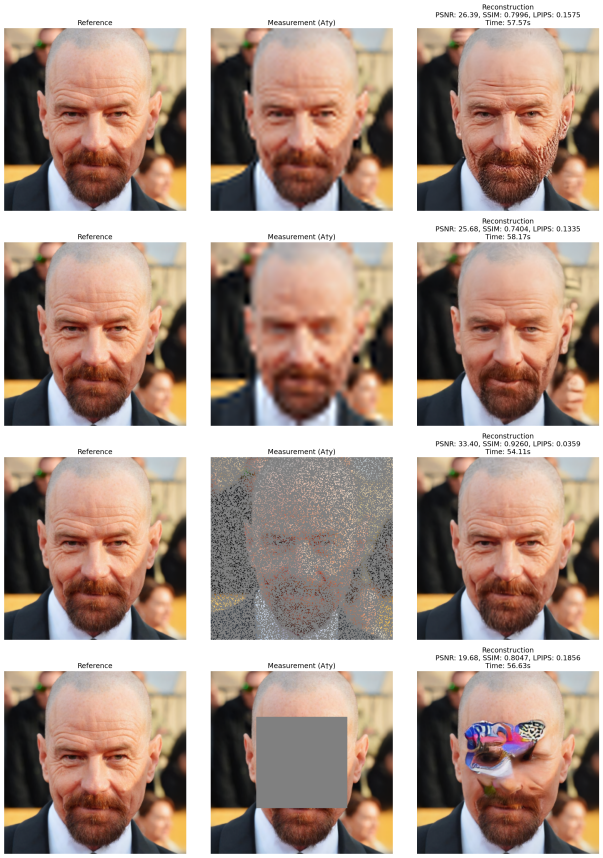

Fig. 4. MCG Reconstructions on CelebA-HQ for SRx4, SRx8, 80% random inpainting, and 128x128 box inpainting.

Fig. 5. MCG Reconstructions on ImageNet for SRx4, SRx8, 80% random inpainting, and 128x128 box inpainting.

in the PSNR and SSIM values that were higher for both models, although ImageNet had some artifacting.

For the box inpainting task, the model failed on both datasetsl, and reconstructed the idea of a face, but it was had a rainbow texture and did not look realistic at all. Image net was not better on this task. It is likely that the gradient step pushed the image away from the data manifold, and the projection step was not able to recover a realistic image.

I think that it might have been a bit slower, but I did not notice a significant difference in speed compared to ILVR on my machine.

- Time: 55s
- Weight: 0.5
- SRx4:
  PSNR: 19.68
  SSIM: 0.8047
  LPIPS: 0.1846
- SRx8:
  PSNR: 25.68
  SSIM: 0.7404
  LPIPS: 0.1335
- 80% random inpainting:
  PSNR: 33.40
  SSIM: 0.9260
  LPIPS: 0.0359
- 128x128 box inpainting:
  PSNR: 19.68
  SSIM:0.8047
  LPIPS: 0.1856

- Time: 178s
- Weight: 0.5
- SRx4:
  PSNR: 12.75
  SSIM: 0.3567
  LPIPS: 0.6224
- SRx8:
  PSNR: 14.46
  SSIM: 0.3693
  LPIPS: 0.4716
- 80% random inpainting:
  PSNR: 23.98
  SSIM: 0.7838
  LPIPS: 0.2242
- 128x128 box inpainting:
  PSNR: 14.11
  SSIM: 0.7472
  LPIPS: 0.3034

4) D: Denoising Diffusion Null-Space Model (DDNM): This model was much faster because it only used 100 sampling steps, but the results were proportionally worse than the previous two methods. The results are shown in

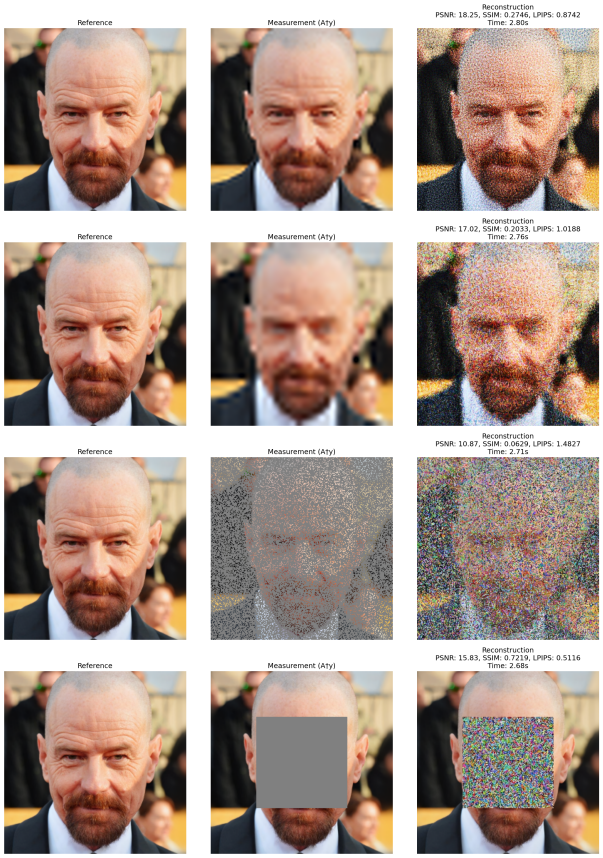

Fig. 6. DDNM Reconstructions on CelebA-HQ for SRx4, SRx8, 80% random inpainting, and 128x128 box inpainting.

Fig. 7. DDNM Reconstructions on ImageNet for SRx4, SRx8, 80% random inpainting, and 128x128 box inpainting.

Figures 6 and 7. Comparing the PSNR and SSIM values to ILVR and MCG, they are significantly lower quality and I would not use this method if quality is a priority.

The super-resolution tasks were very noisy, and had a wavey texture to them. While the box inpainting task was able to make the face and origonal image brighter, there was still a lot of noise and very few details. For some reason the random inpainting task was the worst, with almost pure noise in the box.

One explination for this would be that with only 100 steps, the model is not able to denoise the image enough, resulting in a noisy output. It is also possible I did not find a good weight for the projection step, but I tried a range of values and none produced good results.

- Time: 3s
- Weight: 1.0
- SRx4:
  PSNR: 18.25
  SSIM: 0.2746
  LPIPS: 0.8742
- SRx8:
  PSNR: 17.02
  SSIM: 0.2033
  LPIPS: 1.0188
- 80% random inpainting:
  PSNR: 10.87
  SSIM: 0.0629
  LPIPS: 14.827
- 128x128 box inpainting:
  PSNR: 15.83
  SSIM: 0.7219
  LPIPS: 0.5116
- Time: 9s
- Weight: 1.0
- SRx4:
  PSNR: 13.33
  SSIM: 0.3296
  LPIPS: 0.8221
- SRx8:
  PSNR: 12.76
  SSIM: 0.2855
  LPIPS: 0.9485
- 80% random inpainting:
  PSNR: 8.03
  SSIM: 0.0324
  LPIPS: 1.4114
- 128x128 box inpainting:
  PSNR: 13.24
  SSIM: 0.7202
  LPIPS: 0.5390

5) E: Noisy measurements and Diffusion Posterior Sampling (DPS): Refer to the appendix for the figures for the

Fig. 8. DPS Reconstructions on CelebA-HQ for SRx4, SRx8, 80% random inpainting, and 128x128 box inpainting with noisy measurements.

Fig. 9. DPS Reconstructions on ImageNet for SRx4, SRx8, 80% random inpainting, and 128x128 box inpainting with noisy measurements.

noisy ILVR, MCG, and DDNM results (20, 19, 21, 22). The genral trend is that the noisy measurements resulted in worse reconstructions across all three methods. We only use the CelebA-HQ dataset for this task. The settings for the three methods were the same as before, with only the addition of noise to the measurements.

For the spesific numbers for the noise statistics, please refer to the figures. We find that the PSNR seems universlly higher, with one exception on the ILVR at 80% random inpainting task. This seems to indicate that the noise makes it harder to reconstruct the image accurately. However, the images are still reasonable for the super-resolution tasks, but the box inpainting task has less realistic noise patterns inside the box.

6) F: Diffusion Posterior Sampling (DPS) for noisy measurements: This method produced a lot of wav ey artifacts in the super-resolution tasks for both CelebA-HQ and ImageNet. Otherwise, the face is very crisp and believeable in CelebA, whreeas the ImageNet model struggled on super-resolution.

For the random inpainting task, the model removed a lot of the noise, but there was still some in the background of both models. For the ImageNet model, this introduced some artifacts.

The box inpainting task very bad compared to the first method, where there was sturctured rainbow noise in the box, and it also created noisy patterns in the rest of the image. Image net was a little better on this task, but still had a lot of noise. This is reflected in the PSNR and SSIM values, which were much lower than the original MCG method.

One reason why this may be is that without the projection step, the gradient step may push the image off the data manifold, resulting in noisy images. This is more pronounced given I couldn't find a good weight that produced good results.

- Time: 55s
- Weight: 1.0
- SRx4:
  PSNR: 22.94
  SSIM: 0.7031
  LPIPS: 0.2730
- SRx8:
  PSNR: 26.43
  SSIM: 0.7592
  LPIPS: 0.1143
- 80% random inpainting:
  PSNR: 19.02
  SSIM: 0.7389
  LPIPS: 0.3173
- 128x128 box inpainting:
  PSNR: 11.92

SSIM: 0.3863
LPIPS: 0.5397

- Time: 177s
- Weight: 1.0
- SRx4:
  PSNR: 13.22
  SSIM: 0.3813
  LPIPS: 0.4515
- SRx8:
  PSNR: 16.65
  SSIM: 0.4041
  LPIPS: 0.3862
- 80% random inpainting:
  PSNR: 19.34
  SSIM: 0.5619
  LPIPS: 0.4727
- 128x128 box inpainting:
  PSNR: 13.69
  SSIM: 0.5530
  LPIPS: 0.4243

I. Conditioning Diffusion

A. Implementation and Results

I implemented both methods.. The samples used guidances of $\omega \in \{0.0, 1.0, 3.0, 5.0, 10.0\}$ for 1000 sampling steps. The results for CG are shown in Figure 10 and for CFG in Figure 11.

The accuracies are shown belwon in Table I along with the intra-class diversity scores.

Scale	CG Acc.	CG Div.	CFG Acc.	CFG Div.
0.0	0.080	0.605	0.080	0.605
1.0	0.580	0.504	0.800	0.417
3.0	0.880	0.400	1.000	0.199
5.0	0.930	0.364	1.000	0.146
10.0	0.980	0.331	1.000	0.109

TABLE I
Accuracy and diversity for CG and CFG across different guidance scales.

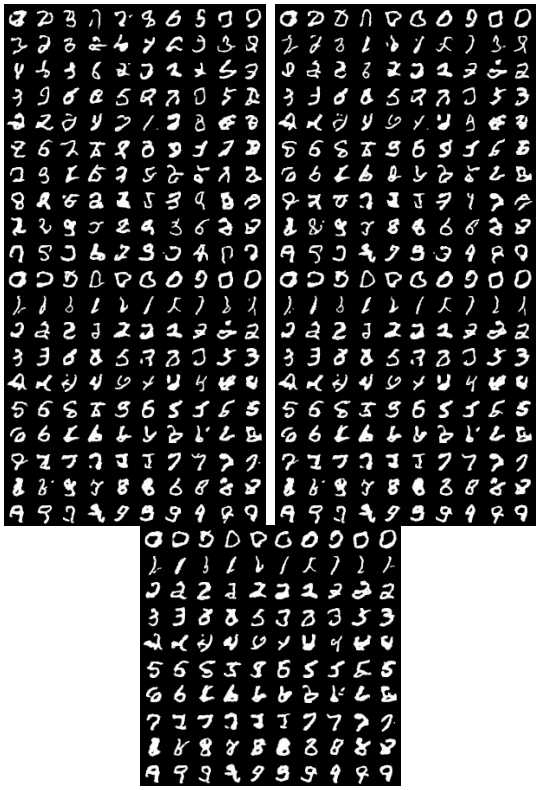

Fig. 10. Classifier Guidance (CG) results. From left to right, top to bottom: $\omega_{CG} = 0.0, 1.0, 3.0, 5.0, 10.0$. Each grid shows 10 samples per digit class (rows 0-9).

1) Reflections: From the results, The CFG got higher accuracies at lower guidance scales compared to CG. However, this came at the cost of diversity, as the CFG samples had lower intra-class diversity scores, especially at higher guidance scales.

Both models had the trend of increasing accuracy and decreasing diversity as the guidance scale increased. This makes sense beause a higher guidance makes the model move twards the class manifold, potentially reducing the entropy available for diversity. The higher the proprity of the class, the less important the original noise was influencing the sample.

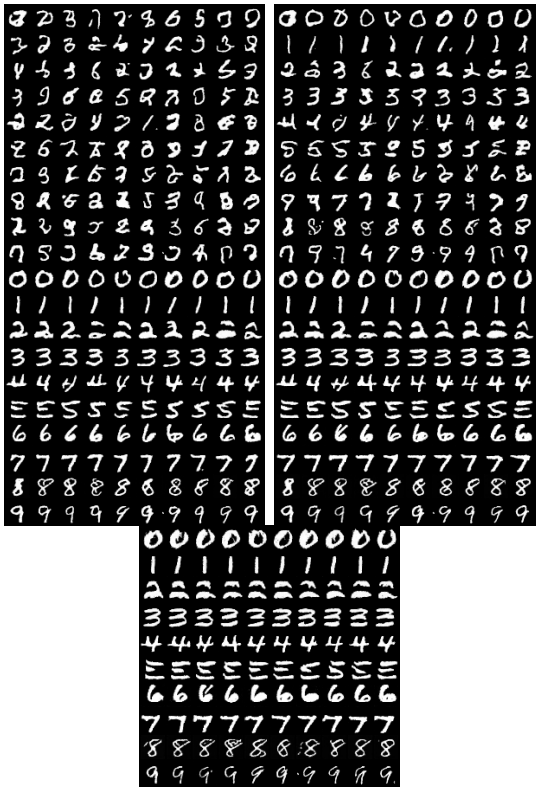

Fig. 11. Classifier-Free Guidance (CFG) results. From left to right, top to bottom: $\omega_{CFG} = 0.0, 1.0, 3.0, 5.0, 10.0$. Each grid shows 10 samples per digit class (rows 0-9).

CFG is the most stable, but all the digits look very similar to each other. However, if I had to choose, CG 10.0 looks the most diverse and accurate given all the configurations.

The fundamental difference between these methods is that CG has external classification that pushes the sample twards the class, wheras the CFG uses the internal knowledge of the model to steer the sample. This is better for many modern applications as text is more diverse and does not have discrete classes, so CG would be difficult to implement.

II. Text to image Generation

A. (A)No Prompt

For the no-prompt generation, I used the same model as in part B, with the parameters $\omega CFG = 0.0$, num steps $= 50$, and $eta = 0.0$, with a fixed seed at 42. The resulting image is shown below.

Fig. 12. No Prompt Image Generation

B. (B-C) 5X3 prompts- Manual evaluation

I implemented a pipeline that will read from a set of prompts, and then use them to generate the images from the diffusers StableDiffusionPipeline module, with the $\omega CFG = 10$, num steps $= 50$, $eta = 0.0$, and a fixed seed at 42. The prompts for each image are included in the caption of the figures.

- Space
  - Simple: Human: 5/10, CLIP: 28.49
  - Medium: Human: 8/10, CLIP: 32.65
  - Detailed: Human: 9/10, CLIP: 33.75
- Ocean
  - Simple: Human: 9/10, CLIP: 31.39
  - Medium: Human: 6/10, CLIP: 32.93
  - Detailed: Human: 5/10, CLIP: 30.34
- Castle
  - Simple: Human: 8/10, CLIP: 29.36
  - Medium: Human: 7/10, CLIP: 31.98
  - Detailed: Human: 9/10, CLIP: 30.38
- Cyberpunk
  - Simple: Human: 8/10, CLIP: 33.70
  - Medium: Human: 8/10, CLIP: 34.95
  - Detailed: Human: 7/10, CLIP: 29.07
- Cat-Bird
  - Simple: Human: 2/10, CLIP: 25.97
  - Medium: Human: 2/10, CLIP: 27.57
  - Detailed: Human: 6/10, CLIP: 37.96

Fig. 13. Space. Left: Simple - "An astronaut in space" (Human: 5/10, CLIP: 28.49). Center: Medium - "An astronaut in a spacesuit floating above Earth" (Human: 8/10, CLIP: 32.65). Right: Detailed - "A single astronaut in a shiny white spacesuit drifting serenely against the stars in the sky. There is a silent planet below with swirling clouds and blue oceans, with their ship orbiting in the distance" (Human: 9/10, CLIP: 33.75).

In almost all cases, the most detailed prompts are the best looking images. While the short prompts also did well, the longer prompts alligned better with what I was expecting from the model. I think that in most cases, if you don't know the prompt, the short and long prompts produce images of similar fidelity, which is expected as the model is trained to approach the image manifold similarly, even without a prompt as seen in part A.

The CLIP and the human scores are pretty well alligned in terms of relitive changes. While the scale of the measurements was not alligned well, when the human score

Fig. 14. Ocean. Left: Simple - "A coral reef" (Human: 9/10, CLIP: 31.39). Center: Medium - "A colorful coral reef with tropical fish of and sunlight filtering through the water" (Human: 6/10, CLIP: 32.93). Right: Detailed - "An underwater coral reef that has all sorts of life, with many fish and sharks swimming around. It has bright corals of all colors and shapes, with sunlight filtering through the clear blue water from above." (Human: 5/10, CLIP: 30.34).

Fig. 15. Castle. Left: Simple - "A medieval castle" (Human: 8/10, CLIP: 29.36). Center: Medium - "A lively medieval castle surrounded by a moat and lush greenery" (Human: 7/10, CLIP: 31.98). Right: Detailed - "A beautyful german day, with a large castle made of stone, with a few vines climbing up the spires. The vilage around the castle is full of life, with people walking around the market." (Human: 9/10, CLIP: 30.38).

changes, the clip score will usually also have a similar change in score, at least in terms of magnitude.

C. (D) Negative Prompts

I chose the prompt: "An astronaut in a spacesuit floating above Earth" with the negative prompt: "blob, blurry, low-definition, water, clouds, fingers" because the original image had a odd blob on the plannet, and I wanted to see if it could remove the fingers. The results are shown in figure 18.

CFG Scale vs CLIP Score results:

- CFG 0.0: CLIP Score 32.8%
- CFG 2.0: CLIP Score 33.9%
- CFG 5.0: CLIP Score 33.6%
- CFG 8.0: CLIP Score 35.6%
- CFG 12.0: CLIP Score 33.7%
- CFG 15.0: CLIP Score 33.0%

At CFG 0, the image is very noisy and low quality, and it does not look like it is on the image manifold. As the CFG increases, the astronaut looks more realistic, where at 5 it looks the best. At 5, there is no longer any water on the plannet, and the fingers are gone as was in the

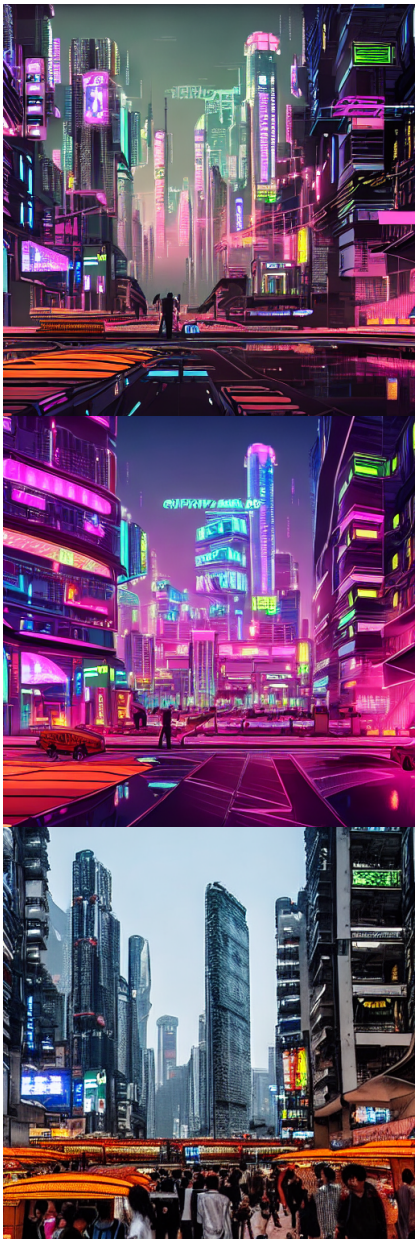

Fig. 16. Cyberpunk. Left: Simple - "A cyberpunk city" (Human: 8/10, CLIP: 33.70). Center: Medium - "A distopian cyberpunk city, with neon lights and flying cars." (Human: 8/10, CLIP: 34.95). Right: Detailed - "A breathtaking cyberpunk megacity that has bustling streets filled with people and vendors. The skyline has many towering skyscrapers, and there are futuristic flying cars." (Human: 7/10, CLIP: 29.07).

negative prompt. At higher CFG, the image becomes a bit degenerate, where the saturation increases, where at 15 the planet is no longer round. This makes sense because at high CFG, there is too much emphasis on the prompt, potentially driving the image off the manifold.

Fig. 17. Cat-Bird. Left: Simple - "A cat with a bird body" (Human: 2/10, CLIP: 25.97). Center: Medium - "A chimera with the body of a cat, wings of a bird." (Human: 2/10, CLIP: 27.57). Right: Detailed - "A beautiful chimera creature that has the body of a maine coon cat, with large majestic wings of an eagle." (Human: 6/10, CLIP: 37.96).

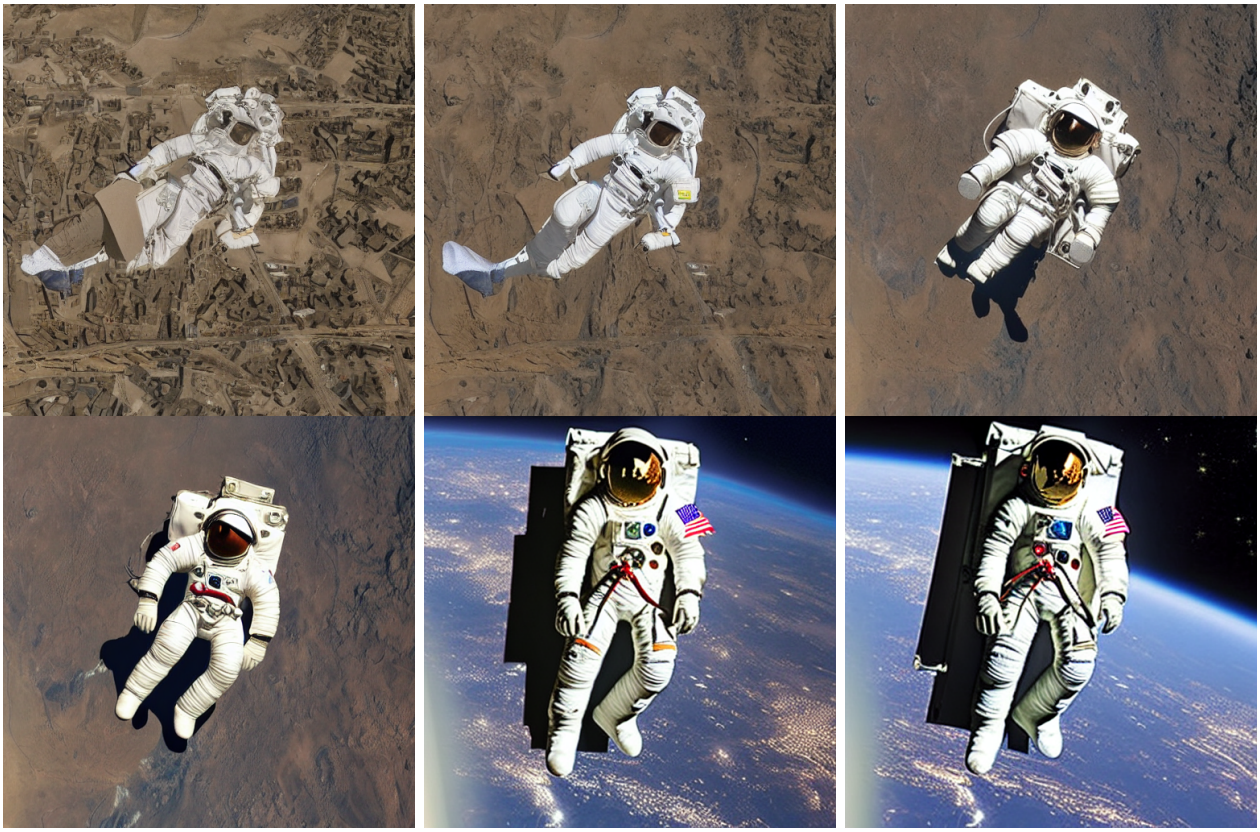

Fig. 18. Negative Prompt Results for "An astronaut in a spacesuit floating above Earth" with various CFG scales. From top left to bottom right: CFG 0.0 (CLIP: 32.8%), CFG 2.0 (CLIP: 33.9%), CFG 5.0 (CLIP: 33.6%), CFG 8.0 (CLIP: 35.6%), CFG 12.0 (CLIP: 33.7%), CFG 15.0 (CLIP: 33.0%).

Fig. 19. Noisy measurement reconstructions on CelebA-HQ for ILVR , MCG, and DDNM for SRx8.

III. Appendix

Fig. 20. Noisy measurement reconstructions on CelebA-HQ for ILVR , MCG, and DDNM for SRx4.

Fig. 21. Noisy measurement reconstructions on CelebA-HQ for ILVR , MCG, and DDNM for 80% random inpainting.

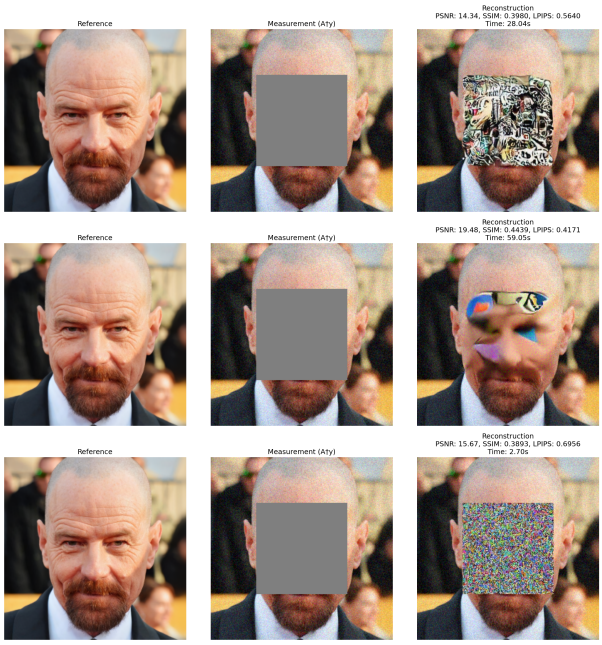

Fig. 22. Noisy measurement reconstructions on CelebA-HQ for ILVR , MCG, and DDNM for 128x128 box inpainting.