# Stable Diffusion

November 8, 2022

```
[ ]: # !pip install diffusers==0.3.0
```

```
[ ]: # !pip3 install torch torchvision torchaudio --extra-index-url https://download.
       ↪pytorch.org/whl/cu117
```

```
[ ]: # !pip install torch
```

```
[1]: from diffusers import DDPMPipeline
```

```
[2]: image_pipe = DDPMPipeline.from_pretrained("google/ddpm-celebahq-256")
```

```
     Fetching 10 files:   0%|          | 0/10 [00:00<?, ?it/s]
```

```
[3]: image_pipe.to("cpu")
```

```
[3]: DDPMPipeline {
       "_class_name": "DDPMPipeline",
       "_diffusers_version": "0.3.0",
       "scheduler": [
         "diffusers",
         "DDPMScheduler"
       ],
       "unet": [
         "diffusers",
         "UNet2DModel"
       ]
     }
```

```
[4]: images = image_pipe().images
```

```
       0%|          | 0/1000 [00:00<?, ?it/s]
```

```
[5]: images[0]
```

```
[5]:
```

[6]:

```
from diffusers import UNet2DModel
```

[7]:

```
repo_id = "google/ddpm-church-256"
model = UNet2DModel.from_pretrained(repo_id)
```

```
Downloading:   0%|          | 0.00/790 [00:00<?, ?B/s]

C:\Users\Richard\anaconda3\lib\site-
packages\huggingface_hub\file_download.py:123: UserWarning: `huggingface_hub`
cache-system uses symlinks by default to efficiently store duplicated files but
your machine does not support them in
C:\Users\Richard\.cache\huggingface\diffusers. Caching files will still work but
in a degraded version that might require more space on your disk. This warning
can be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environment
variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-
to-cache#limitations.
To support symlinks on Windows, you either need to activate Developer Mode or to
run Python as an administrator. In order to see activate developer mode, see
this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-
your-device-for-development
  warnings.warn(message)

Downloading:   0%|          | 0.00/455M [00:00<?, ?B/s]
```

[8]:

```
model
```

```
[8]: UNet2DModel(
       (conv_in): Conv2d(3, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
       (time_proj): Timesteps()
       (time_embedding): TimestepEmbedding(
         (linear_1): Linear(in_features=128, out_features=512, bias=True)
         (act): SiLU()
         (linear_2): Linear(in_features=512, out_features=512, bias=True)
       )
       (down_blocks): ModuleList(
         (0): DownBlock2D(
           (resnets): ModuleList(
             (0): ResnetBlock2D(
               (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
               (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
     padding=(1, 1))
               (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
               (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
               (dropout): Dropout(p=0.0, inplace=False)
               (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
     padding=(1, 1))
               (nonlinearity): SiLU()
             )
             (1): ResnetBlock2D(
               (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
               (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
     padding=(1, 1))
               (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
               (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
               (dropout): Dropout(p=0.0, inplace=False)
               (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
     padding=(1, 1))
               (nonlinearity): SiLU()
             )
           )
           (downsamplers): ModuleList(
             (0): Downsample2D(
               (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2))
             )
           )
         )
         (1): DownBlock2D(
           (resnets): ModuleList(
             (0): ResnetBlock2D(
               (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
               (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
     padding=(1, 1))
               (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
```

```
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
          (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
        )
      )
      (downsamplers): ModuleList(
        (0): Downsample2D(
          (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2))
        )
      )
    )
    (2): DownBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
          (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
```

```
          (nonlinearity): SiLU()
        )
      )
      (downsamplers): ModuleList(
        (0): Downsample2D(
          (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2))
        )
      )
    )
    (3): DownBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
        )
      )
      (downsamplers): ModuleList(
        (0): Downsample2D(
          (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2))
        )
      )
    )
    (4): AttnDownBlock2D(
      (attentions): ModuleList(
        (0): AttentionBlock(
          (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
          (query): Linear(in_features=512, out_features=512, bias=True)
          (key): Linear(in_features=512, out_features=512, bias=True)
          (value): Linear(in_features=512, out_features=512, bias=True)
          (proj_attn): Linear(in_features=512, out_features=512, bias=True)
        )
        (1): AttentionBlock(
          (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
          (query): Linear(in_features=512, out_features=512, bias=True)
          (key): Linear(in_features=512, out_features=512, bias=True)
          (value): Linear(in_features=512, out_features=512, bias=True)
          (proj_attn): Linear(in_features=512, out_features=512, bias=True)
        )
      )
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 512, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
          (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
        )
      )
      (downsamplers): ModuleList(
        (0): Downsample2D(
          (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2))
        )
      )
    )
    (5): DownBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
          (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
```

```
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
          (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
        )
      )
    )
  )
  (up_blocks): ModuleList(
    (0): UpBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
          (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
          (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
          (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
        )
      )
      (upsamplers): ModuleList(
        (0): Upsample2D(
          (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        )
      )
    )
    (1): AttnUpBlock2D(
      (attentions): ModuleList(
        (0): AttentionBlock(
          (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
          (query): Linear(in_features=512, out_features=512, bias=True)
          (key): Linear(in_features=512, out_features=512, bias=True)
          (value): Linear(in_features=512, out_features=512, bias=True)
          (proj_attn): Linear(in_features=512, out_features=512, bias=True)
        )
        (1): AttentionBlock(
          (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
          (query): Linear(in_features=512, out_features=512, bias=True)
          (key): Linear(in_features=512, out_features=512, bias=True)
          (value): Linear(in_features=512, out_features=512, bias=True)
          (proj_attn): Linear(in_features=512, out_features=512, bias=True)
        )
        (2): AttentionBlock(
          (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
          (query): Linear(in_features=512, out_features=512, bias=True)
          (key): Linear(in_features=512, out_features=512, bias=True)
          (value): Linear(in_features=512, out_features=512, bias=True)
          (proj_attn): Linear(in_features=512, out_features=512, bias=True)
        )
      )
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
          (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
          (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 768, eps=1e-06, affine=True)
          (conv1): Conv2d(768, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
          (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(768, 512, kernel_size=(1, 1), stride=(1, 1))
        )
      )
      (upsamplers): ModuleList(
        (0): Upsample2D(
          (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        )
      )
    )
    (2): UpBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 768, eps=1e-06, affine=True)
          (conv1): Conv2d(768, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(768, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
          (conv1): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
          (conv1): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        )
      )
      (upsamplers): ModuleList(
        (0): Upsample2D(
          (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        )
      )
    )
    (3): UpBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
          (conv1): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
          (conv1): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 384, eps=1e-06, affine=True)
          (conv1): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(384, 256, kernel_size=(1, 1), stride=(1, 1))
        )
      )
      (upsamplers): ModuleList(
        (0): Upsample2D(
          (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        )
      )
    )
    (4): UpBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 384, eps=1e-06, affine=True)
          (conv1): Conv2d(384, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
```

```
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(384, 128, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
      )
      (upsamplers): ModuleList(
        (0): Upsample2D(
          (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        )
      )
    )
    (5): UpBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
      )
    )
  )
  (mid_block): UNetMidBlock2D(
    (attentions): ModuleList(
      (0): AttentionBlock(
        (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
        (query): Linear(in_features=512, out_features=512, bias=True)
        (key): Linear(in_features=512, out_features=512, bias=True)
        (value): Linear(in_features=512, out_features=512, bias=True)
        (proj_attn): Linear(in_features=512, out_features=512, bias=True)
      )
    )
    (resnets): ModuleList(
      (0): ResnetBlock2D(
        (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
        (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1))
        (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
        (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
        (dropout): Dropout(p=0.0, inplace=False)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1))
        (nonlinearity): SiLU()
      )
      (1): ResnetBlock2D(
        (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
        (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1))
        (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
        (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
        (dropout): Dropout(p=0.0, inplace=False)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1))
        (nonlinearity): SiLU()
      )
    )
  )
  (conv_norm_out): GroupNorm(32, 128, eps=1e-06, affine=True)
  (conv_act): SiLU()
  (conv_out): Conv2d(128, 3, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
)
```

[9]:
```
model.config
```

[9]:
```
FrozenDict([('sample_size', 256),
            ('in_channels', 3),
            ('out_channels', 3),
            ('center_input_sample', False),
            ('time_embedding_type', 'positional'),
            ('freq_shift', 1),
            ('flip_sin_to_cos', False),
            ('down_block_types',
             ['DownBlock2D',
              'DownBlock2D',
              'DownBlock2D',
              'DownBlock2D',
              'AttnDownBlock2D',
              'DownBlock2D']),
            ('up_block_types',
             ['UpBlock2D',
              'AttnUpBlock2D',
              'UpBlock2D',
              'UpBlock2D',
              'UpBlock2D',
              'UpBlock2D']),
            ('block_out_channels', [128, 128, 256, 256, 512, 512]),
            ('layers_per_block', 2),
            ('mid_block_scale_factor', 1),
            ('downsample_padding', 0),
            ('act_fn', 'silu'),
            ('attention_head_dim', None),
            ('norm_num_groups', 32),
            ('norm_eps', 1e-06),
            ('_class_name', 'UNet2DModel'),
            ('_diffusers_version', '0.3.0'),
            ('_name_or_path', 'google/ddpm-church-256')])
```

[10]:
```python
model_random = UNet2DModel(**model.config)
```

[11]:
```python
model_random.save_pretrained("my_model")
```

[13]:
```python
model_random = UNet2DModel.from_pretrained("my_model")
```

[14]:
```python
import torch

torch.manual_seed(0)

noisy_sample = torch.randn(
    1, model.config.in_channels, model.config.sample_size, model.config.
 ↪sample_size
)
noisy_sample.shape
```

[14]:
```
torch.Size([1, 3, 256, 256])
```

[15]:
```python
with torch.no_grad():
    noisy_residual = model(sample=noisy_sample, timestep=2).sample
```

[16]:
```python
noisy_residual.shape
```

[16]:
```
torch.Size([1, 3, 256, 256])
```

[17]:
```python
from diffusers import DDPMScheduler

scheduler = DDPMScheduler.from_config(repo_id)
```

```
Downloading:   0%|          | 0.00/256 [00:00<?, ?B/s]
```

[18]:
```python
scheduler.config
```

```
[18]: FrozenDict([('num_train_timesteps', 1000),
                  ('beta_start', 0.0001),
                  ('beta_end', 0.02),
                  ('beta_schedule', 'linear'),
                  ('trained_betas', None),
                  ('variance_type', 'fixed_small'),
                  ('clip_sample', True),
                  ('_class_name', 'DDPMScheduler'),
                  ('_diffusers_version', '0.3.0')])
```

```
[19]: scheduler.save_config("my_scheduler")
      new_scheduler = DDPMScheduler.from_config("my_scheduler")
```

```
[20]: less_noisy_sample = scheduler.step(
          model_output=noisy_residual, timestep=2, sample=noisy_sample
      ).prev_sample
      less_noisy_sample.shape
```

```
[20]: torch.Size([1, 3, 256, 256])
```

```
[21]: import PIL.Image
      import numpy as np

      def display_sample(sample, i):
          image_processed = sample.cpu().permute(0, 2, 3, 1)
          image_processed = (image_processed + 1.0) * 127.5
          image_processed = image_processed.numpy().astype(np.uint8)

          image_pil = PIL.Image.fromarray(image_processed[0])
          display(f"Image at step {i}")
          display(image_pil)
```

```
[22]: model.to("cpu")
```

```
[22]: UNet2DModel(
        (conv_in): Conv2d(3, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (time_proj): Timesteps()
        (time_embedding): TimestepEmbedding(
          (linear_1): Linear(in_features=128, out_features=512, bias=True)
          (act): SiLU()
          (linear_2): Linear(in_features=512, out_features=512, bias=True)
        )
        (down_blocks): ModuleList(
          (0): DownBlock2D(
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
                (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
                (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
                (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
                (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
            )
            (downsamplers): ModuleList(
              (0): Downsample2D(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2))
              )
            )
          )
          (1): DownBlock2D(
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
                (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
                (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
                (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
                (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
            )
            (downsamplers): ModuleList(
              (0): Downsample2D(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2))
              )
            )
          )
          (2): DownBlock2D(
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 128, eps=1e-06, affine=True)
                (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
                (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1))
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
                (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
                (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
            )
            (downsamplers): ModuleList(
              (0): Downsample2D(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2))
              )
            )
          )
          (3): DownBlock2D(
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
                (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
                (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
                (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
                (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
            )
            (downsamplers): ModuleList(
              (0): Downsample2D(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2))
              )
            )
          )
          (4): AttnDownBlock2D(
            (attentions): ModuleList(
              (0): AttentionBlock(
                (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
                (query): Linear(in_features=512, out_features=512, bias=True)
                (key): Linear(in_features=512, out_features=512, bias=True)
                (value): Linear(in_features=512, out_features=512, bias=True)
                (proj_attn): Linear(in_features=512, out_features=512, bias=True)
              )
              (1): AttentionBlock(
                (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
                (query): Linear(in_features=512, out_features=512, bias=True)
                (key): Linear(in_features=512, out_features=512, bias=True)
                (value): Linear(in_features=512, out_features=512, bias=True)
                (proj_attn): Linear(in_features=512, out_features=512, bias=True)
              )
            )
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
                (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(256, 512, kernel_size=(1, 1), stride=(1, 1))
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
                (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
            )
            (downsamplers): ModuleList(
              (0): Downsample2D(
                (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2))
              )
            )
          )
          (5): DownBlock2D(
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
                (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
                (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
              )
            )
          )
        )
        (up_blocks): ModuleList(
          (0): UpBlock2D(
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
                (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
                (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
              )
              (2): ResnetBlock2D(
                (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
                (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
              )
            )
            (upsamplers): ModuleList(
              (0): Upsample2D(
                (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
              )
            )
          )
          (1): AttnUpBlock2D(
            (attentions): ModuleList(
              (0): AttentionBlock(
                (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
                (query): Linear(in_features=512, out_features=512, bias=True)
                (key): Linear(in_features=512, out_features=512, bias=True)
                (value): Linear(in_features=512, out_features=512, bias=True)
                (proj_attn): Linear(in_features=512, out_features=512, bias=True)
              )
              (1): AttentionBlock(
                (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
                (query): Linear(in_features=512, out_features=512, bias=True)
                (key): Linear(in_features=512, out_features=512, bias=True)
                (value): Linear(in_features=512, out_features=512, bias=True)
                (proj_attn): Linear(in_features=512, out_features=512, bias=True)
              )
              (2): AttentionBlock(
                (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
                (query): Linear(in_features=512, out_features=512, bias=True)
                (key): Linear(in_features=512, out_features=512, bias=True)
                (value): Linear(in_features=512, out_features=512, bias=True)
                (proj_attn): Linear(in_features=512, out_features=512, bias=True)
              )
            )
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
                (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 1024, eps=1e-06, affine=True)
                (conv1): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
              )
              (2): ResnetBlock2D(
                (norm1): GroupNorm(32, 768, eps=1e-06, affine=True)
                (conv1): Conv2d(768, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
                (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(768, 512, kernel_size=(1, 1), stride=(1, 1))
              )
            )
            (upsamplers): ModuleList(
              (0): Upsample2D(
                (conv): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
              )
            )
          )
          (2): UpBlock2D(
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 768, eps=1e-06, affine=True)
                (conv1): Conv2d(768, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
                (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(768, 256, kernel_size=(1, 1), stride=(1, 1))
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
                (conv1): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
                (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
              )
              (2): ResnetBlock2D(
                (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
                (conv1): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
                (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
              )
            )
            (upsamplers): ModuleList(
              (0): Upsample2D(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
              )
            )
          )
          (3): UpBlock2D(
            (resnets): ModuleList(
              (0): ResnetBlock2D(
                (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
                (conv1): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
                (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
                (dropout): Dropout(p=0.0, inplace=False)
                (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
                (nonlinearity): SiLU()
                (conv_shortcut): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
              )
              (1): ResnetBlock2D(
                (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
                (conv1): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1),
      padding=(1, 1))
```

```
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 384, eps=1e-06, affine=True)
          (conv1): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=256, bias=True)
          (norm2): GroupNorm(32, 256, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(384, 256, kernel_size=(1, 1), stride=(1, 1))
        )
      )
      (upsamplers): ModuleList(
        (0): Upsample2D(
          (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        )
      )
    )
    (4): UpBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 384, eps=1e-06, affine=True)
          (conv1): Conv2d(384, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(384, 128, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
      )
      (upsamplers): ModuleList(
        (0): Upsample2D(
          (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        )
      )
    )
    (5): UpBlock2D(
      (resnets): ModuleList(
        (0): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ResnetBlock2D(
          (norm1): GroupNorm(32, 256, eps=1e-06, affine=True)
          (conv1): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (time_emb_proj): Linear(in_features=512, out_features=128, bias=True)
          (norm2): GroupNorm(32, 128, eps=1e-06, affine=True)
          (dropout): Dropout(p=0.0, inplace=False)
          (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
          (nonlinearity): SiLU()
          (conv_shortcut): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        )
      )
    )
  )
  (mid_block): UNetMidBlock2D(
    (attentions): ModuleList(
      (0): AttentionBlock(
        (group_norm): GroupNorm(32, 512, eps=1e-06, affine=True)
        (query): Linear(in_features=512, out_features=512, bias=True)
        (key): Linear(in_features=512, out_features=512, bias=True)
        (value): Linear(in_features=512, out_features=512, bias=True)
        (proj_attn): Linear(in_features=512, out_features=512, bias=True)
      )
    )
    (resnets): ModuleList(
      (0): ResnetBlock2D(
        (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
        (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1))
        (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
        (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
        (dropout): Dropout(p=0.0, inplace=False)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1))
        (nonlinearity): SiLU()
      )
      (1): ResnetBlock2D(
        (norm1): GroupNorm(32, 512, eps=1e-06, affine=True)
        (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1))
```

```
        (time_emb_proj): Linear(in_features=512, out_features=512, bias=True)
        (norm2): GroupNorm(32, 512, eps=1e-06, affine=True)
        (dropout): Dropout(p=0.0, inplace=False)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1))
        (nonlinearity): SiLU()
      )
    )
  )
  (conv_norm_out): GroupNorm(32, 128, eps=1e-06, affine=True)
  (conv_act): SiLU()
  (conv_out): Conv2d(128, 3, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
)
```

[23]:

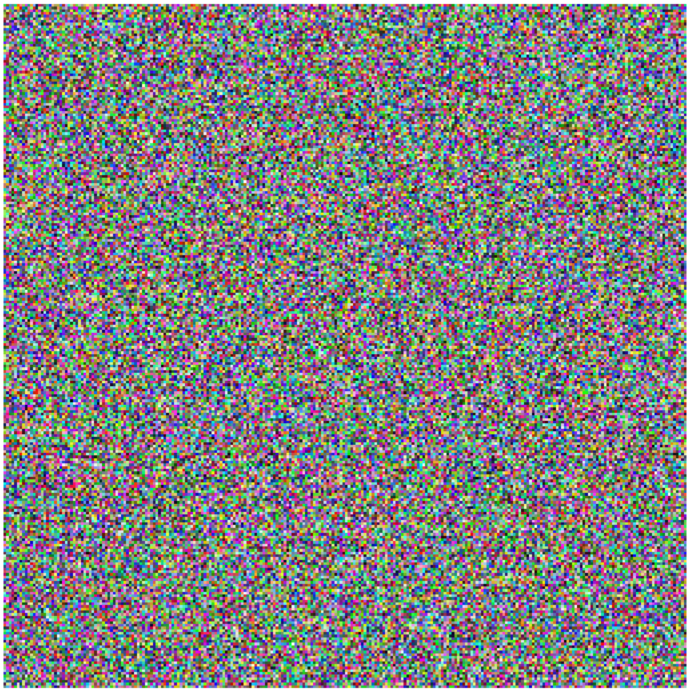
```python
noisy_sample = noisy_sample.to("cpu")
```

[24]:

```python
import tqdm

sample = noisy_sample

for i, t in enumerate(tqdm.tqdm(scheduler.timesteps)):
  # 1. predict noise residual
  with torch.no_grad():
      residual = model(sample, t).sample

  # 2. compute less noisy image and set x_t -> x_t-1
  sample = scheduler.step(residual, t, sample).prev_sample

  # 3. optionally look at image
  if (i + 1) % 50 == 0:
      display_sample(sample, i + 1)
```

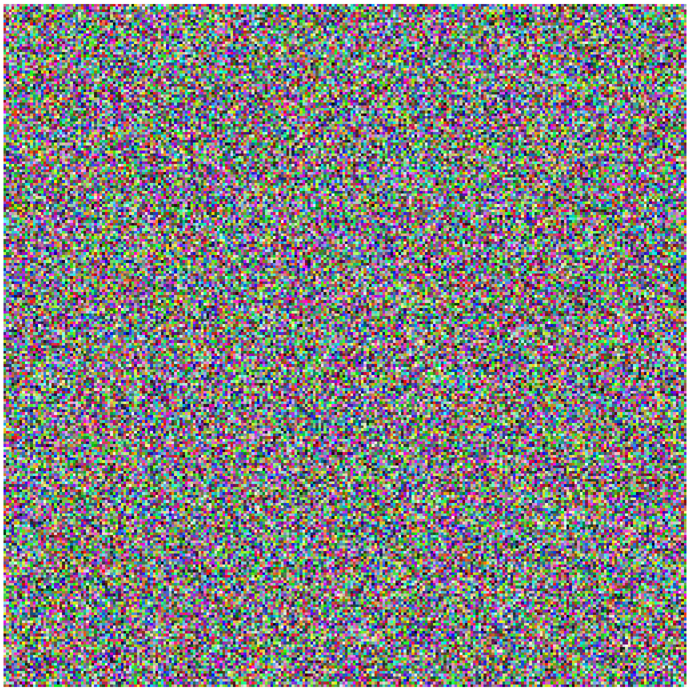
```
  5%|
| 49/1000 [02:04<40:34,  2.56s/it]
```

```
'Image at step 50'
```



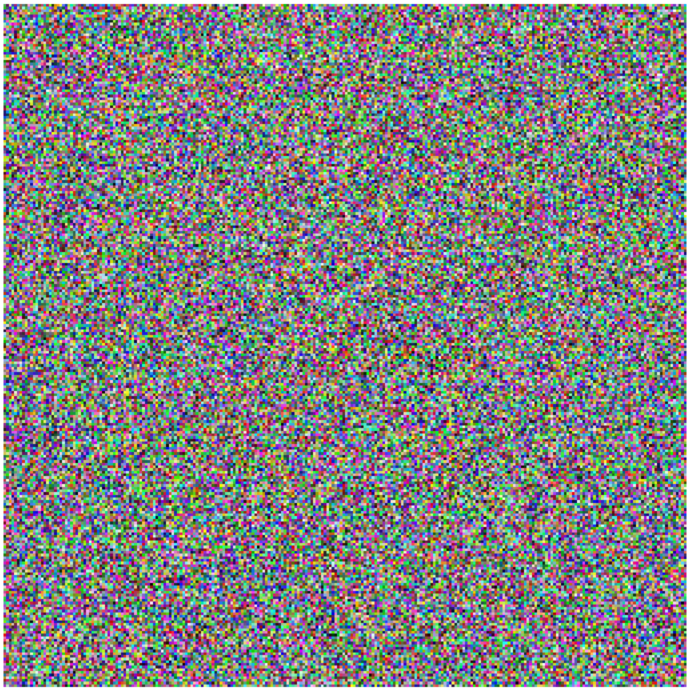
```
 10%|
| 99/1000 [04:10<38:05,  2.54s/it]
```

```
'Image at step 100'
```



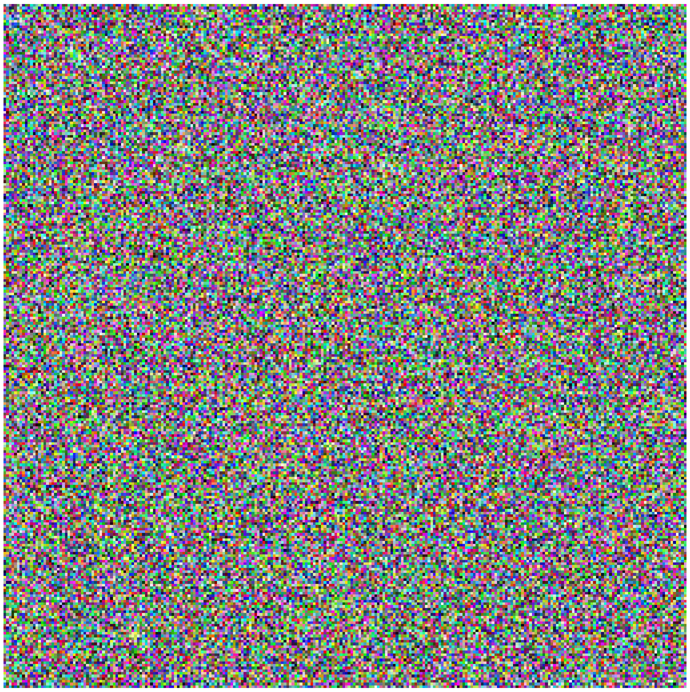
```
 15%|
| 149/1000 [06:18<35:35,  2.51s/it]
```

```
'Image at step 150'
```



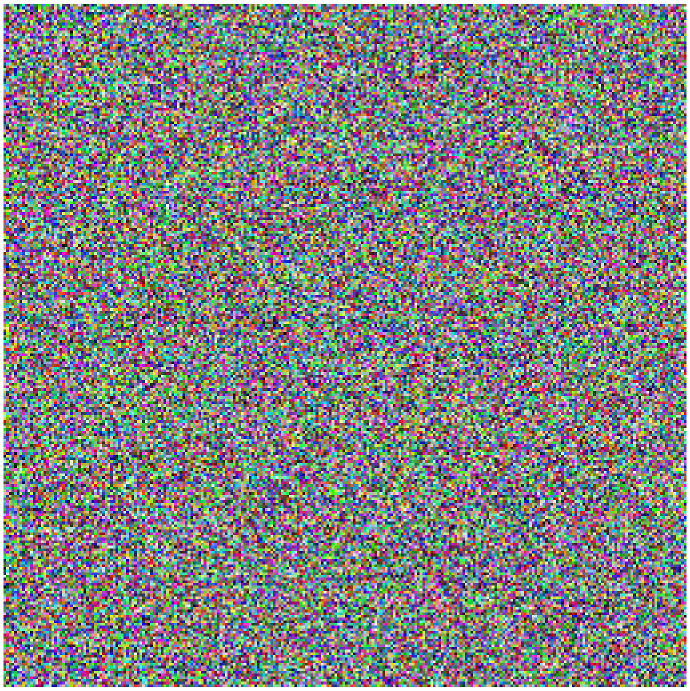
```
 20%|
| 199/1000 [08:39<37:01,  2.77s/it]
```

```
'Image at step 200'
```



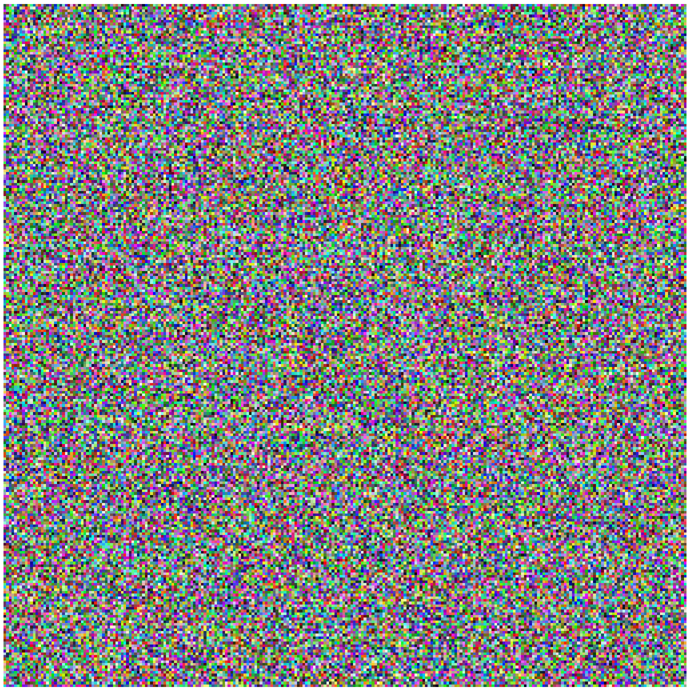
```
 25%|
| 249/1000 [10:59<33:02,  2.64s/it]
```

```
'Image at step 250'
```



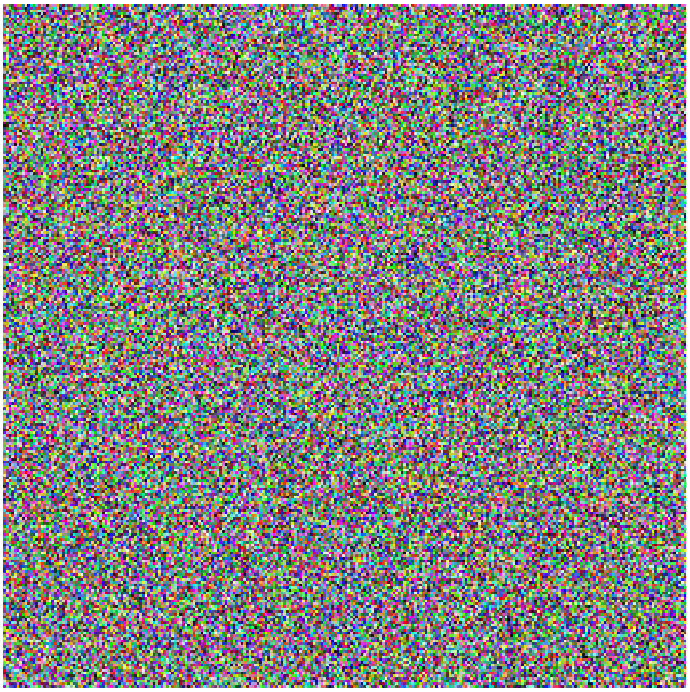
```
 30%|
| 299/1000 [13:21<32:51,  2.81s/it]
```

```
'Image at step 300'
```

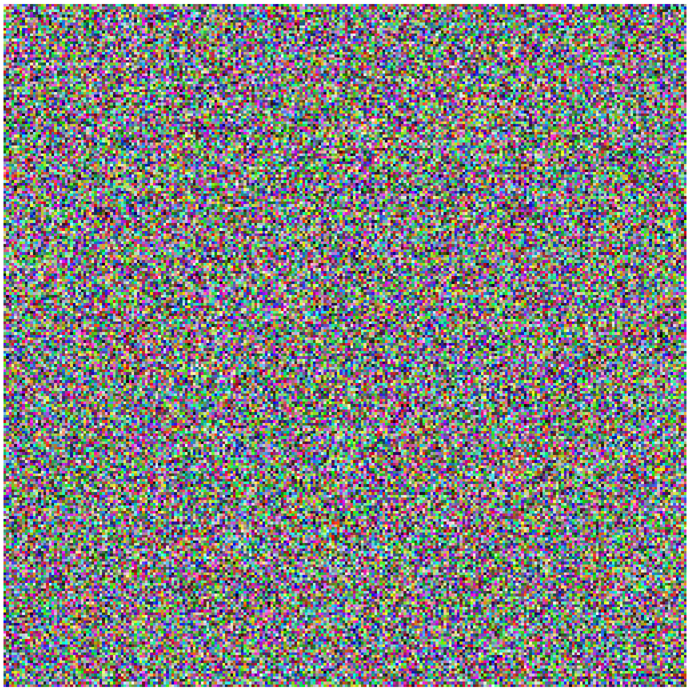


```
 35%|
| 349/1000 [15:37<30:14,  2.79s/it]
```

```
'Image at step 350'
```

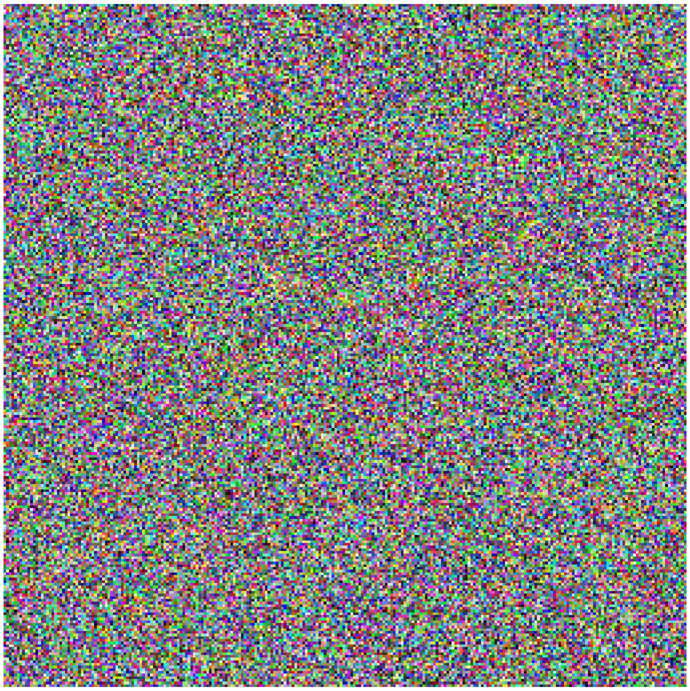


```
 40%|
| 399/1000 [17:51<25:42,  2.57s/it]

'Image at step 400'
```



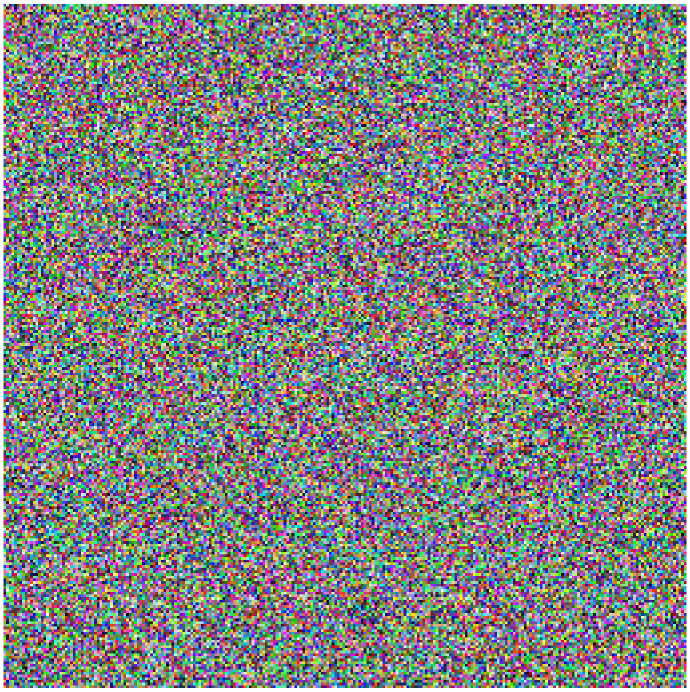
```
 45%|
| 449/1000 [20:10<25:09,  2.74s/it]
```

'Image at step 450'

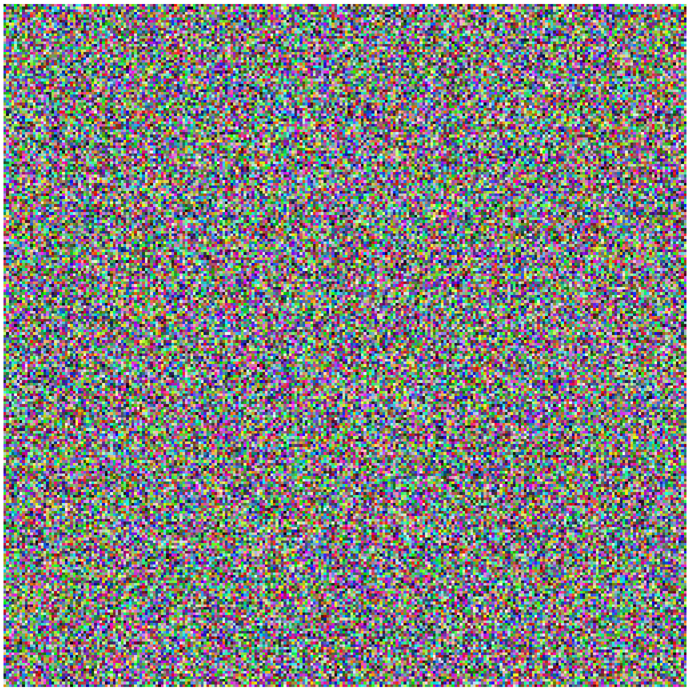


```
 50%|
| 499/1000 [22:29<22:17,  2.67s/it]
```

'Image at step 500'

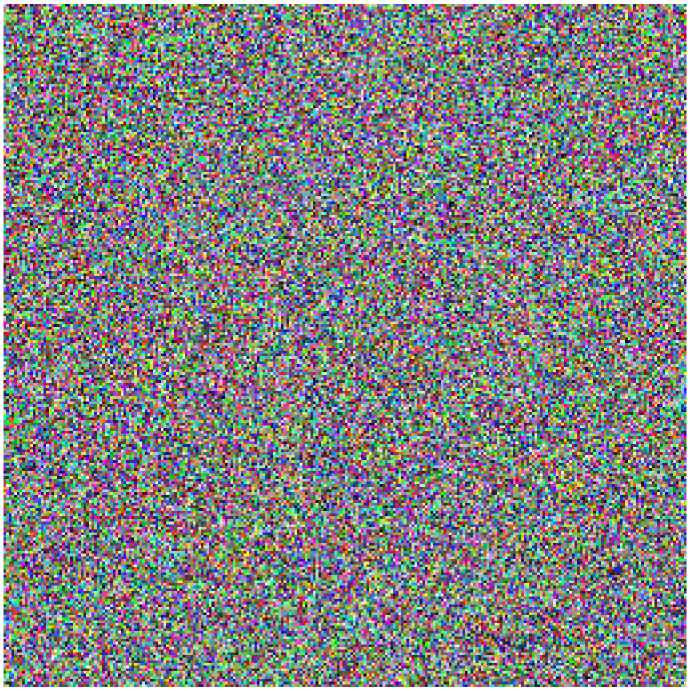


```
 55%|
| 549/1000 [24:46<20:53,  2.78s/it]

'Image at step 550'
```



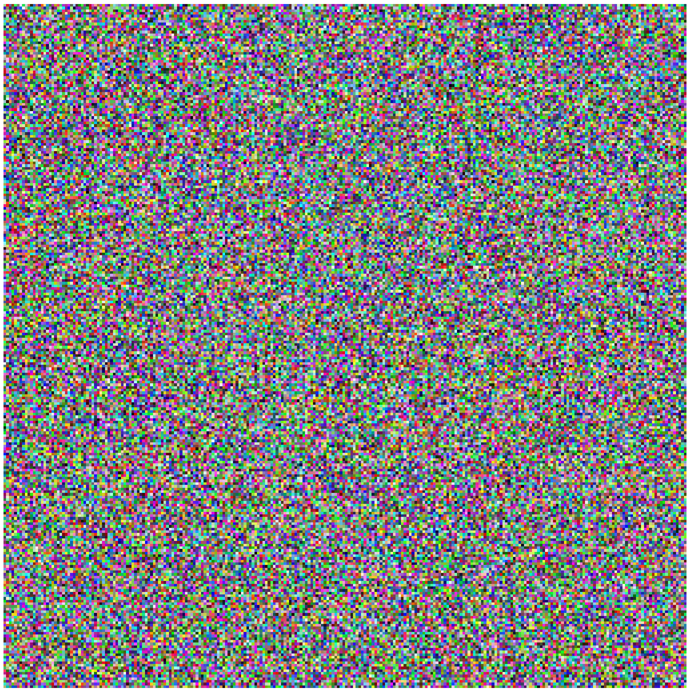
```
 60%|
| 599/1000 [26:55<19:10,  2.87s/it]
```

```
'Image at step 600'
```



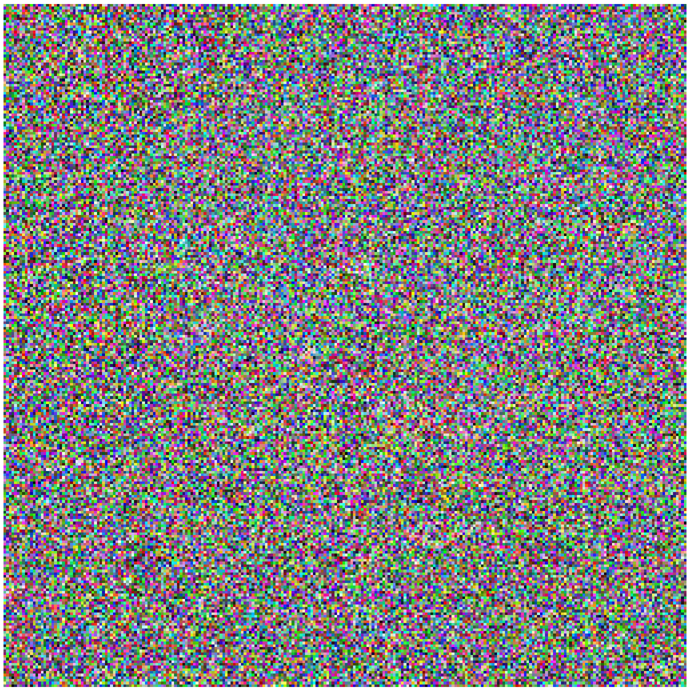
```
 65%|
| 649/1000 [29:11<16:14,  2.78s/it]
```

```
'Image at step 650'
```



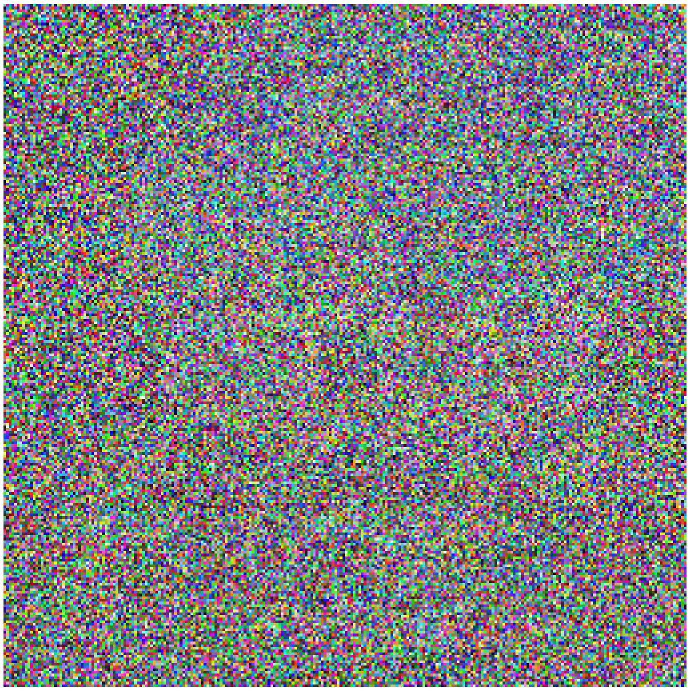
```
 70%|
| 699/1000 [31:32<13:58,  2.79s/it]
```

```
'Image at step 700'
```



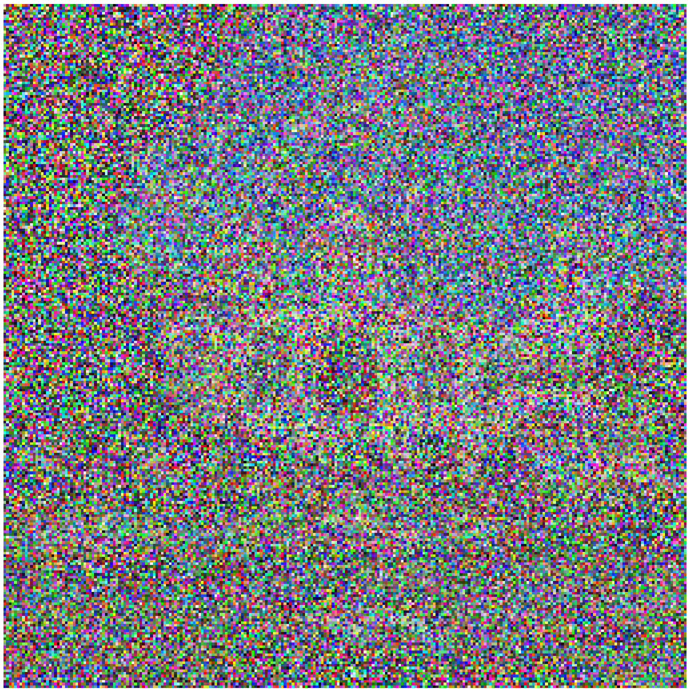
```
 75%|
| 749/1000 [33:58<11:54,  2.85s/it]
```

```
'Image at step 750'
```



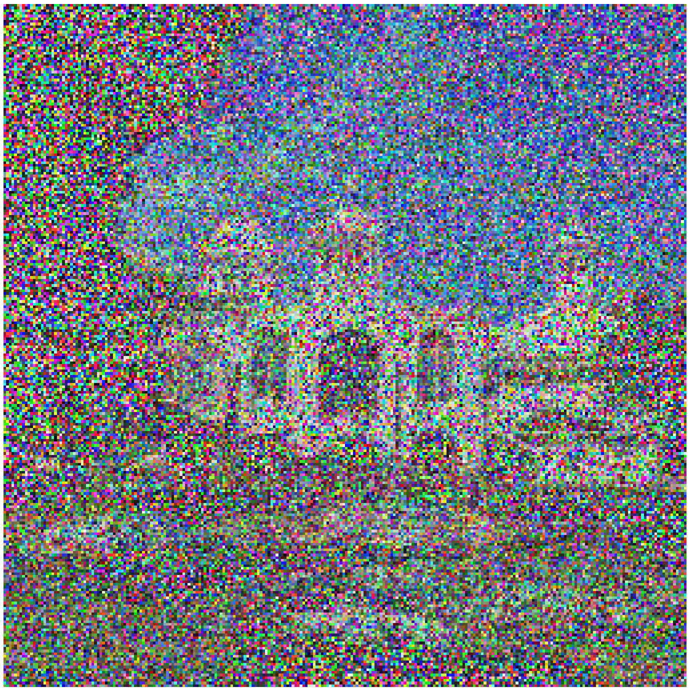
```
 80%|
| 799/1000 [36:26<09:50,  2.94s/it]
```

```
'Image at step 800'
```



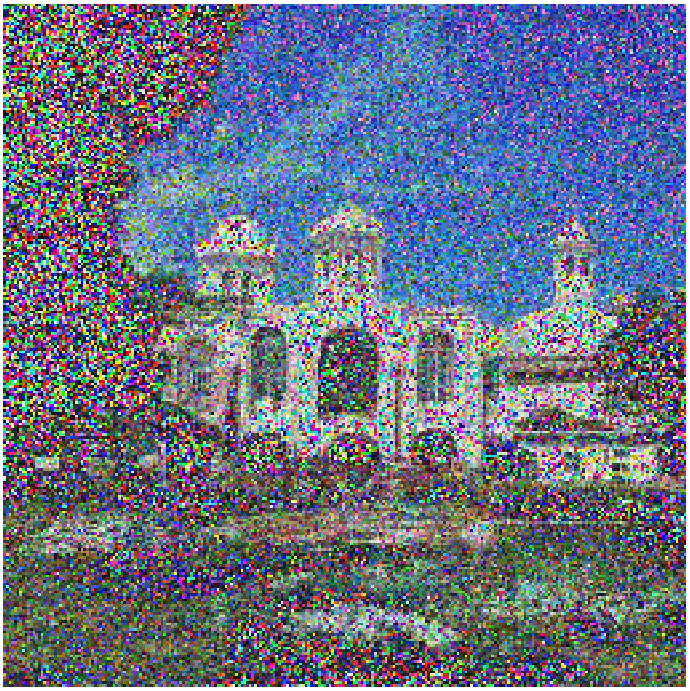
```
 85%|
| 849/1000 [38:48<06:47,  2.70s/it]
```

```
'Image at step 850'
```



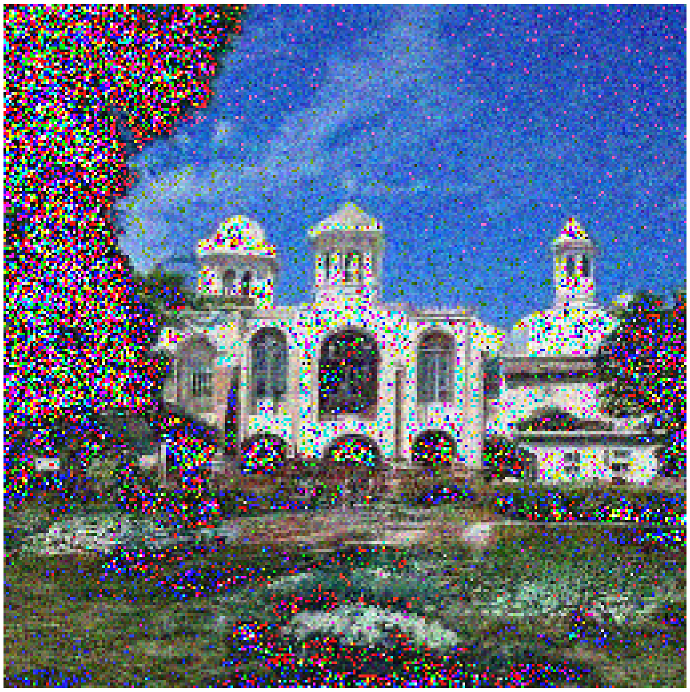
```
 90%|
| 899/1000 [41:09<05:02,  2.99s/it]
```

```
'Image at step 900'
```



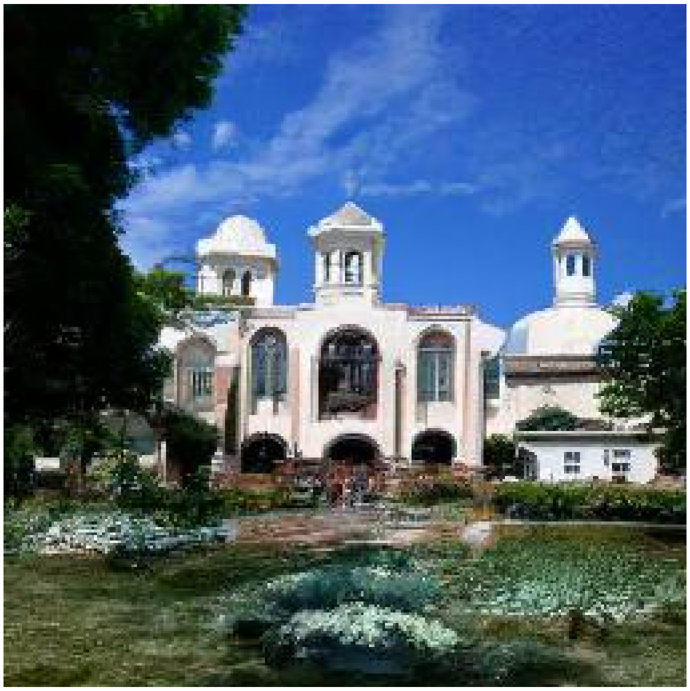
```
 95%|
| 949/1000 [43:22<02:10,  2.55s/it]
```

```
'Image at step 950'
```



```
100%|
  | 999/1000 [45:32<00:02,  2.65s/it]
```

```
'Image at step 1000'
```



```
100%|
  | 1000/1000 [45:35<00:00,  2.74s/it]
```

```
[ ]:
```