Introduction

In this lab we will play with DDPM. We will choose our model, and our own noise schedule.

Implementation details

```
down_block_types=(
    "DownBlock2D",  # a
    "DownBlock2D",
    "DownBlock2D",
    "DownBlock2D",
    "AttnDownBlock2D",
    "DownBlock2D",
),
up_block_types=(
    "UpBlock2D",  # a r
    "AttnUpBlock2D",  #
    "UpBlock2D",
    "UpBlock2D",
    "UpBlock2D",
    "UpBlock2D",
```

I chose the hugging face diffuser UNet2DModel [1]. The down blocks I used are like the figure to the left. There are two layers per block and the channels are 128,128,256,256,512,512 from top to bottom.

The noise schedule is cosine_cap_v2 from hugging face's DDPMScheduler and starts from beta 0.0001 to 0.02. The total steps T is set to 1000 for training.

I first resized the image to 64x64 to feed into the hugging face diffuser. Then, if I want to do evaluation, I have to first normalize the image to mean (0.5,0.5,0.5) and std (0.5,0.5,0.5) to feed into the evaluator.

For the class_label I used a linear layer to project the one-hot class encoding to the same dimension as the time embedding. Then I add the resulting vector with the time embedding.

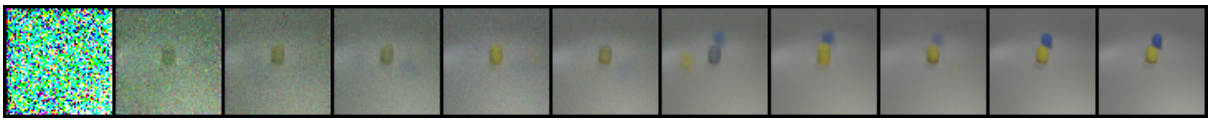
The loss function is the same as the original DDPM paper.

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

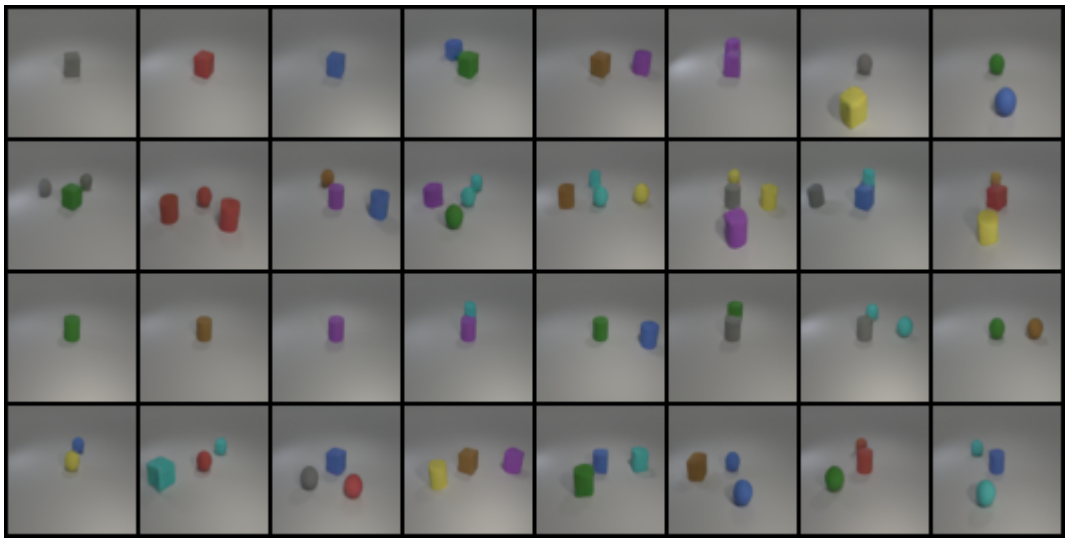
Hyperparameters

I use a learning rate of 1e-5, AdamW as the optimizer. I trained for a total of 50 epochs.

Results and discussion



progressive generation

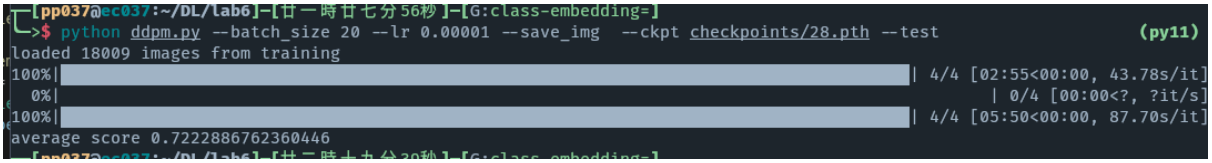


test.json

```
┌─[pp037@ec037:~/DL/lab6]-[廿一時十八分50秒]-[G:class-embedding=]
└─>$ python ddpm.py --batch_size 20 --lr 0.00001 --save_img  --ckpt checkpoints/28.pth --test                (py11)
loaded 18009 images from training
100%|                                                                                  | 4/4 [02:56<00:00, 44.09s/it]
100%|                                                                                  | 4/4 [05:51<00:00, 87.91s/it]
average score 0.6864548494983278
┌─[pp037@ec037:~/DL/lab6]-[廿一時廿七分56秒]-[G:class-embedding=]
└─>$                                                                                                          (py11)
```

test.json accuracy

new_test.json



new_test.json accuracy

I originally fed the whole training image without resizing and got either a completely black image or a color-shifted image. Later someone told me I should resize the image first before feeding the image into the diffuser. I spent a lot of time wondering what I did wrong. I tried changing the model size, since the training time is unacceptable, changing to linear schedule, cosine schedule. Removing a couple of layers or reducing a couple of channels still produces a weird colored image. I even tried the DDIMScheduler but that is not a pure version of the original DDPM so I abandoned it.

Reference

[1] https://huggingface.co/docs/diffusers/api/models/unet2d