GAI_HW6 313511022 陳威云

一、 訓練架構

1. Dataset Loader

   在 load 進資料時，參照範例程式有將 10%的 caption 設成空白，讓模型學到怪物大致平均是甚麼樣子的，用於後續生成。

2. Training Function

   我的 unet 設計如下:

```
unet = UNet2DConditionModel(
    sample_size=32,  # size in latent space after VAE, e.g. 256 // 8
    in_channels=4,
    out_channels=4,
    layers_per_block=2,
    block_out_channels=(160, 320, 640, 1280, 1280),
    down_block_types=(
        "CrossAttnDownBlock2D",
        "CrossAttnDownBlock2D",
        "CrossAttnDownBlock2D",
        "CrossAttnDownBlock2D",
        "DownBlock2D",  # optional: no attention at deepest
    ),
    up_block_types=(
        "UpBlock2D",  # optional: no attention
        "CrossAttnUpBlock2D",
        "CrossAttnUpBlock2D",
        "CrossAttnUpBlock2D",
        "CrossAttnUpBlock2D",
    ),
    cross_attention_dim=512,
).to(device)
```

使用了五層，並將 batch size 調到 128，幾乎吃滿 GPU 記憶體，並訓練 30 epochs，留下 loss 最低的模型。

二、 圖片生成

在圖片生成時，我使用 PNDMScheduler，雖然生成時間長，但效果比較好，下列實驗接選用 test.json 前 36 個資料並在 local 端跑測試實驗，因沒有 image 的真正答案，故不考慮 CLIP image to image 的分數，並分成兩種生成方式:

1. 隨機生成

   生成圖片的起點為隨機向量，並使用 Classifier-free guidance 的方式生成，實驗結果如下:

   guidance_scale 固定為 10:

scheduler.set_timesteps	FID	CLIP image to text
60	261	0.267
70	246.938	0.266
80	250.745	0.275
90	260.452	0.271
100	263.843	0.271

最終 timesteps 選用 70。

scheduler.set_timesteps 固定為 70:

guidence	FID	CLIP image to text
2.5	279.112	0.243
5	260.56	0.269
7.5	249.839	0.273
10	246.938	0.267
12.5	264.16	0.272

最終 guidence 選用 10。

整體最終上傳結果成績為:

"CLIP Image-Text Score": 0.2788,

"CLIP Image-Image Score": 0.746,

"FID": 109.689

2. RAG 生成

生成前先用該 text_prompt 的 embedding 去找到 training description 中最相近的描述，並把該描述的圖片轉成 embedding，當作生成的起點，並由於直接用 unet 生成會去掉太多原本圖片該有的資訊而導致模糊，所以先加上幾層雜訊再去雜訊反而有更好的結果:

scheduler.set_timesteps 固定為 70:

加幾次雜訊	FID	CLIP image to text
20	317.4	0.243
30	283.021	0.255
40	260.616	0.269
50	234.656	0.272
60	222.27	0.276

最終加 60 次雜訊。

整體最終上傳結果成績為:

"CLIP Image-Text Score": 0.286,

"CLIP Image-Image Score": 0.806,

"FID": 63.059

因此選用方法二。