生成式 AI：文字與圖像生成的原理與實務

11. 文字生成圖像 AI 的原理及用 Fooocus 實作

蔡炎龍

政治大學應用數學系

01.

讓文字和圖像的意涵拉近的 CLIP

CLIP

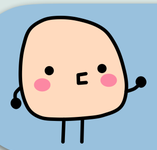
Contrastive **L**anguage-**I**mage **P**retraining

CLIP

越像越好

Alec Radford et al. (OpenAI), “Learning Transferable Visual Models From Natural Language Supervision,” 2021.

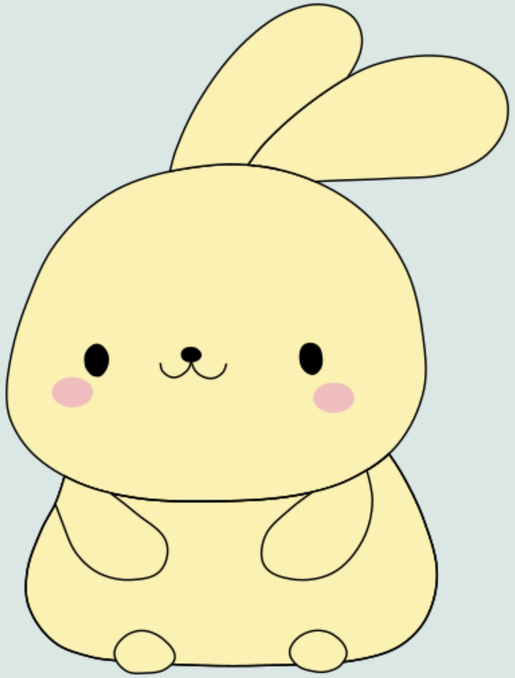
https://arxiv.org/abs/2103.00020



OpenCLIP

雖然 OpenAI 有開放 CLIP 的模型, 但訓練好的參數並沒有開放。於是有了 OpenCLIP 計畫, 提供開放參數給大家下載。

Stable Diffusion 2.x 開始使用 OpenCLIP, 是用 LAION-5B 數據集訓練的。



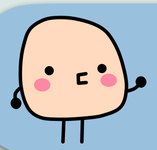
Christoph Schuhmann et al. (LAION), "Reproducible scaling laws for contrastive language-image learning," 2022.

https://arxiv.org/abs/2210.08402

LAION-5B

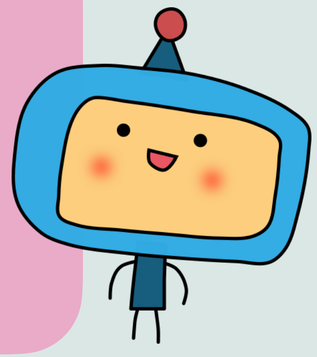
Christoph Schuhmann et al. (LAION), "LAION-5B: An open large-scale dataset for training next generation image-text models," NeurIPS 2022.

https://arxiv.org/abs/2210.08402



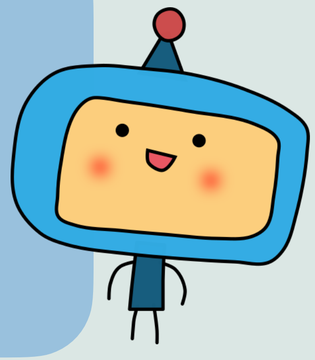
Stable Diffusion 理解文字的機制

SD 1.x



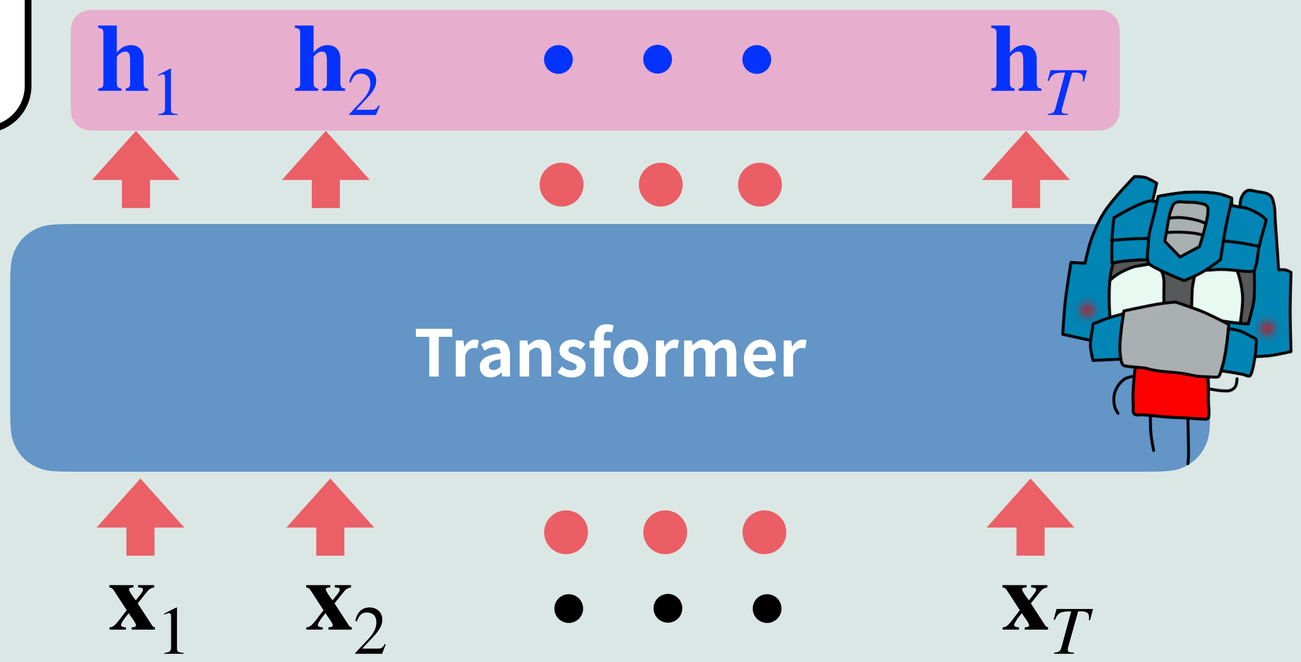
使用 CLIP

SD 2.x

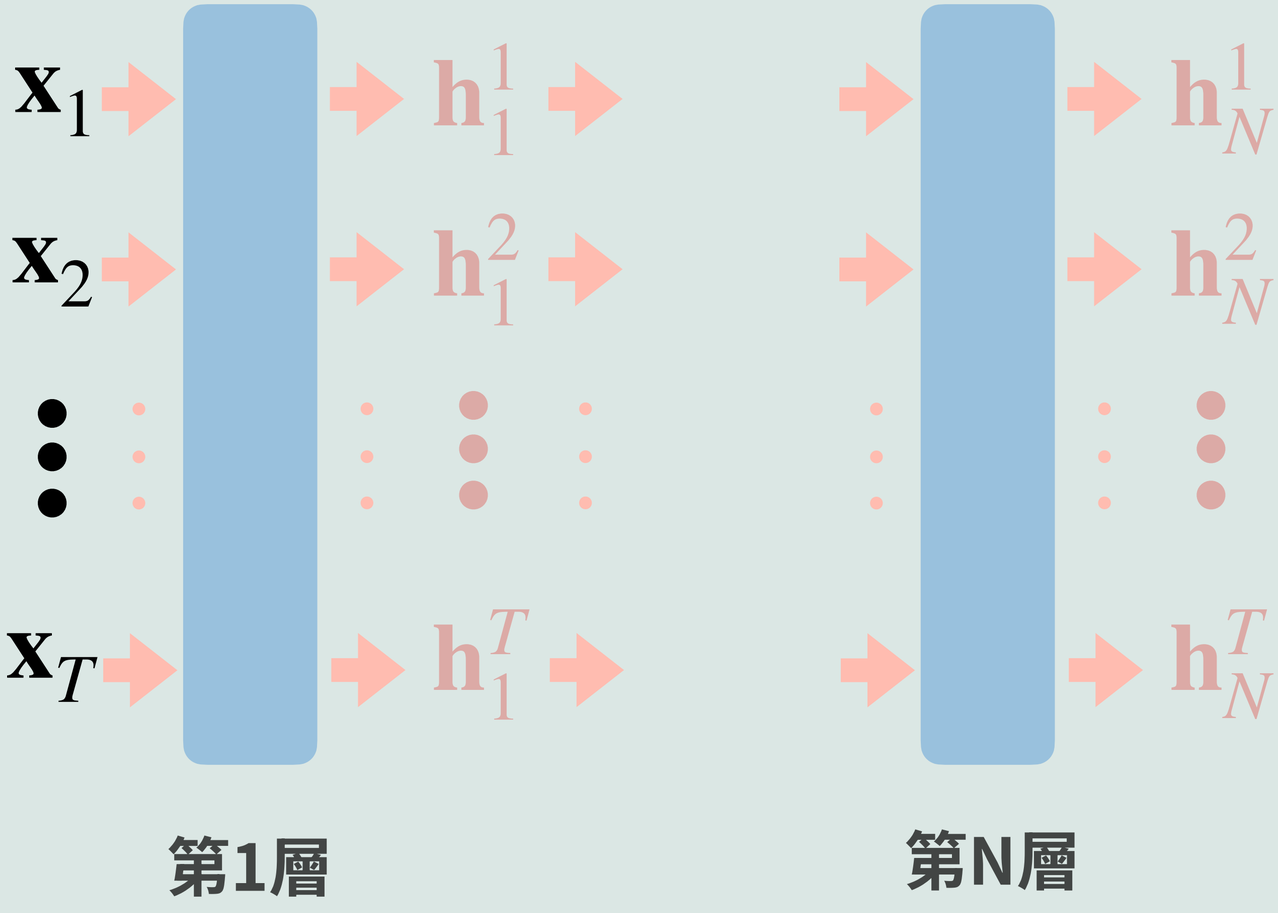


使用 OpenCLIP

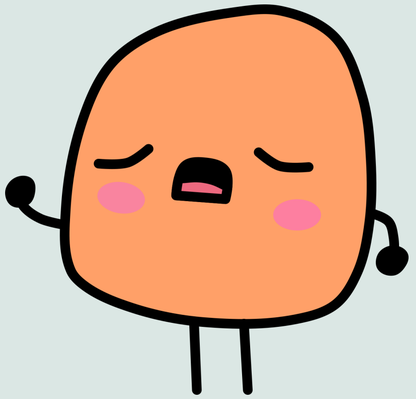
Transformers 模型是這個樣子



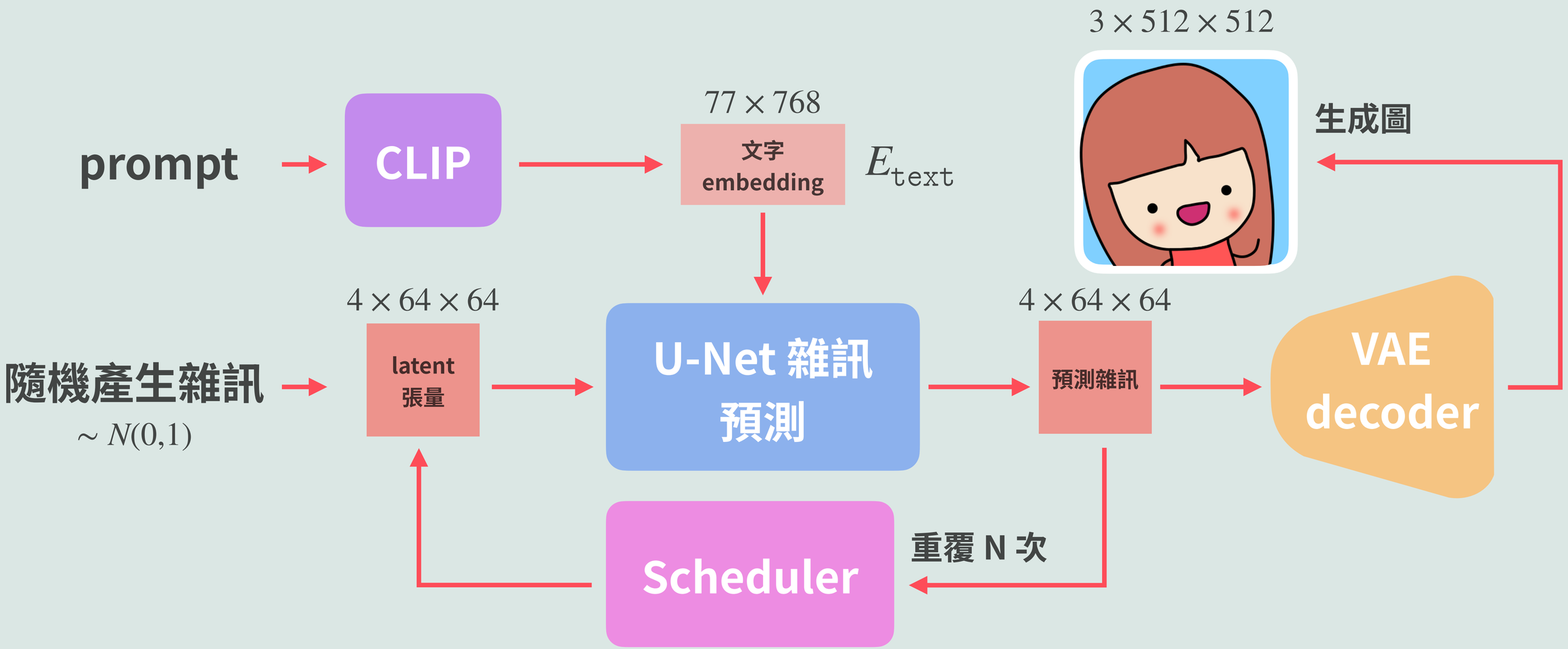
CLIP 有 12 層 transformers

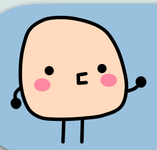


總之, transformers 和其他神經網路一樣, 將原向量轉換成另一組向量, 可以視為電腦對輸入的「抽象理解」, 或是原來數據的特徵代表向量。



複習 Stable Diffusion 架構圖





如之前說過, 先用 VAE 找圖的 latent tensor

$3 \times 512 \times 512$

VAE encoder

$4 \times 64 \times 64$

latent 張量

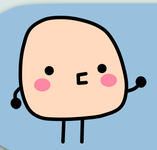
VAE decoder

$3 \times 512 \times 512$

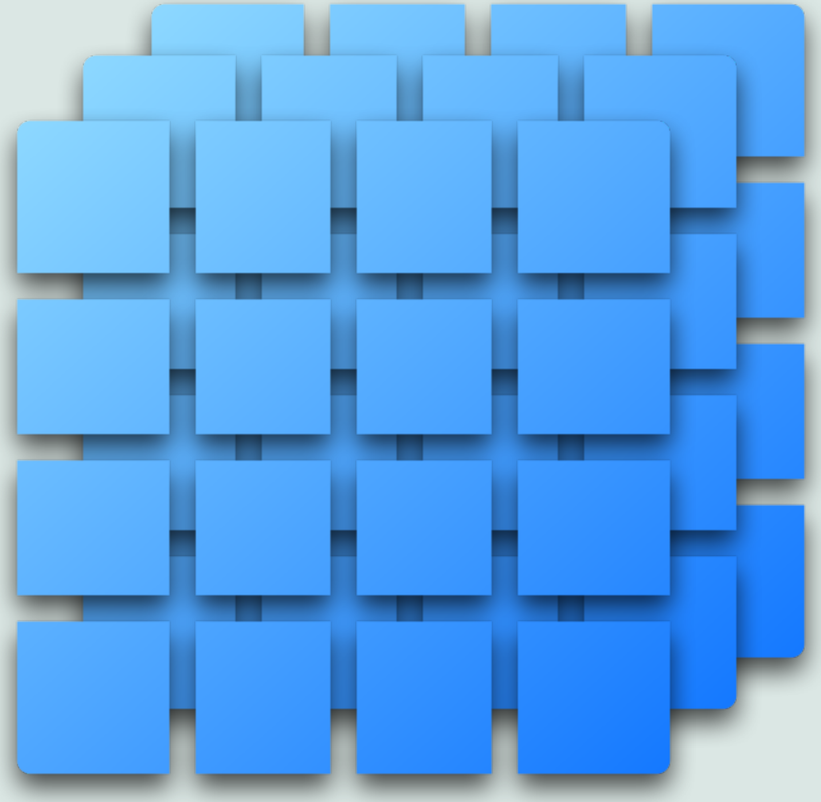
因為 R, G, B 所以有 3 個通道

為什麼是 4 個通道?

CNN 一個 filter (特徵截取器) 會生出一個計分板



有幾個 filter, 就有幾個計分板



然後「加到」我們隨機生成的那些 noise

latent vector

$$z = z_T + y$$

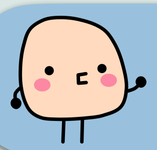
z_T: $4 \times 4 \times 3$

y: 77×768

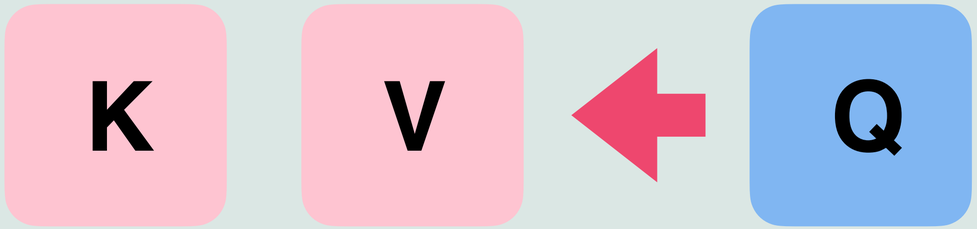
目前的計分板

代表文字意思的

是不是和 StyleGAN 很像? 這裡「加」也不一定是真的加...



標準「融合」通常是用 transformer...

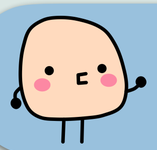


y

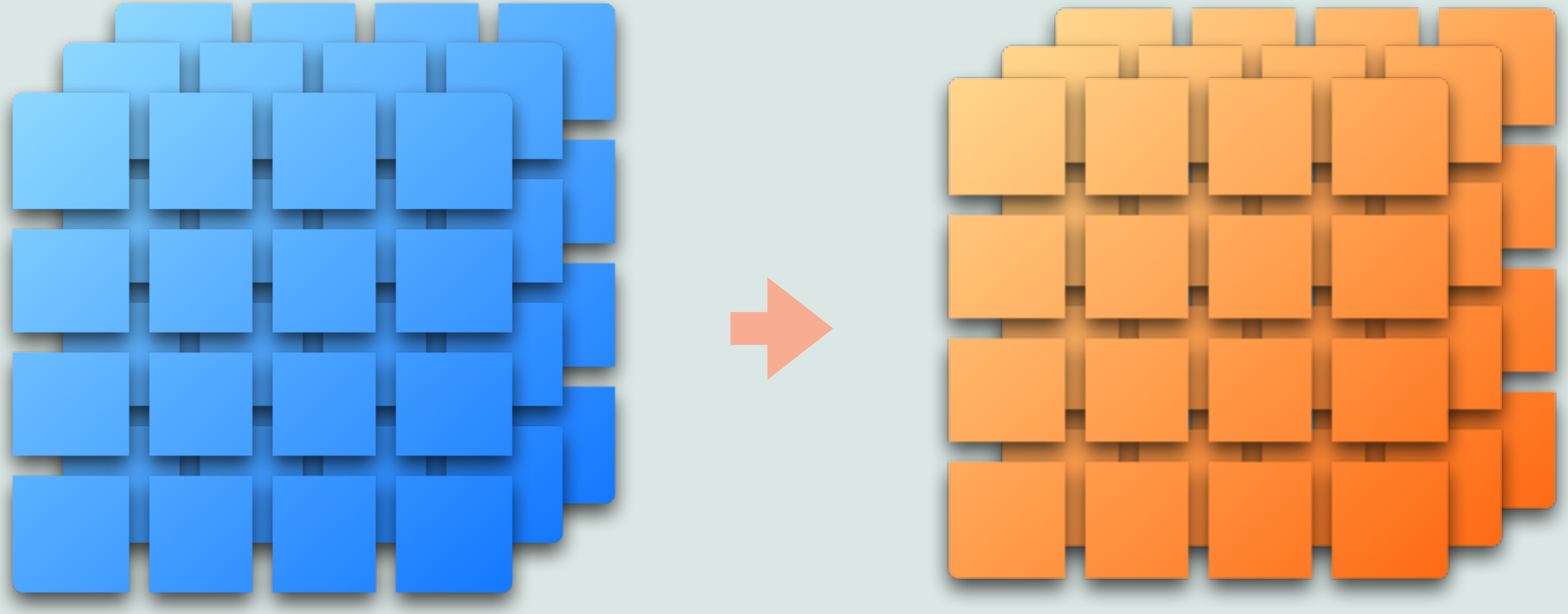
z_T

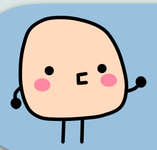
K, V 是文字這邊來的

Q 是隨機生出原始 latent vector 來的

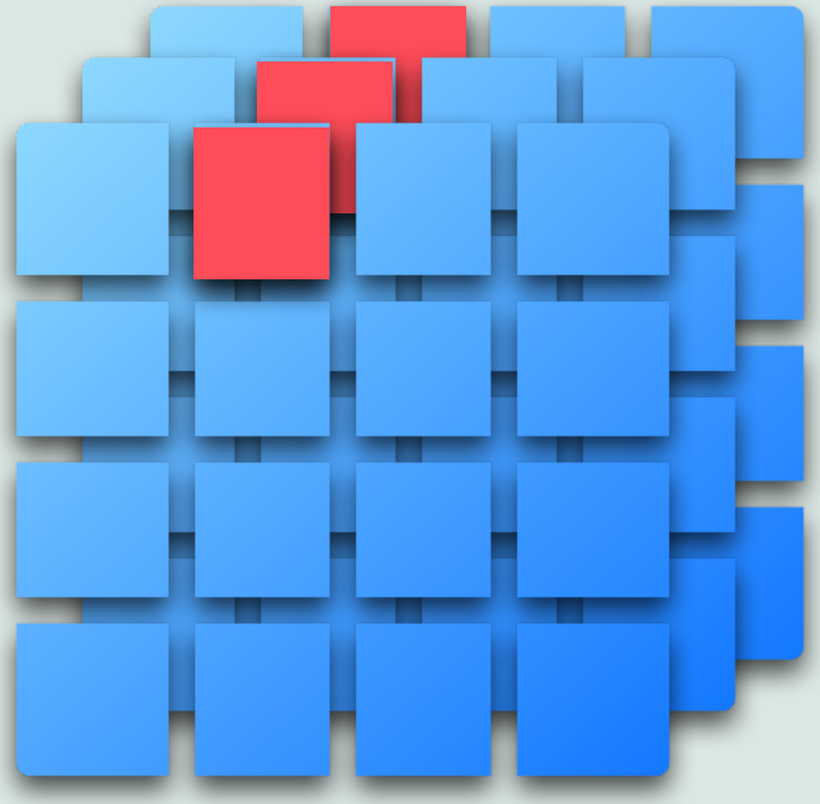


目標是輸出要和原大小一樣!





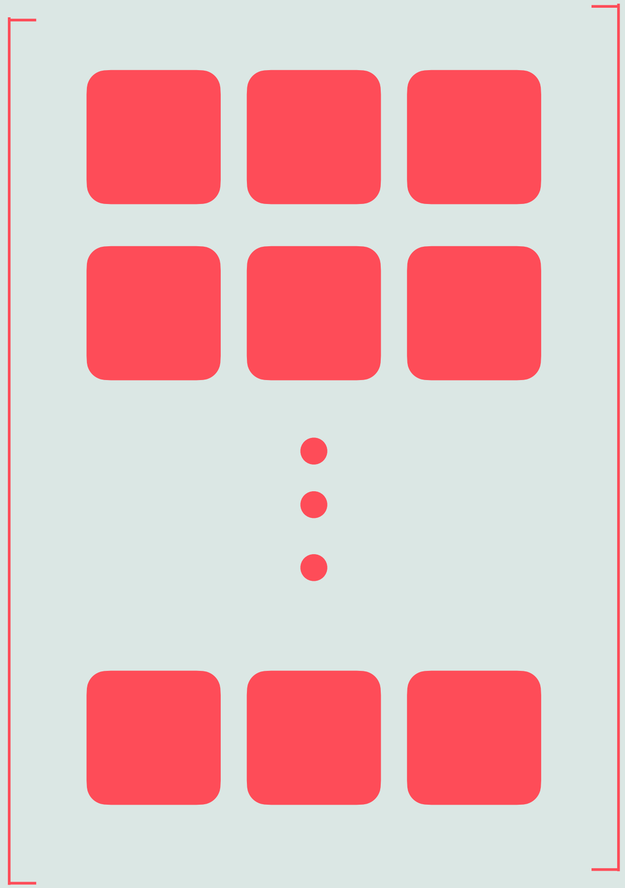
一個 query 向量長這樣



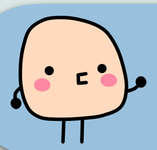
本例共有 16 個

Q 矩陣

$$Q =$$



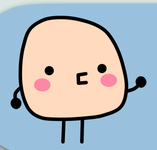
16×3



文字的 embedding 大小是固定的

$$E_{\texttt{text}} = \begin{bmatrix} \blacksquare & \blacksquare & \cdots & \blacksquare \\ \blacksquare & \blacksquare & \cdots & \blacksquare \\ & & \ddots & \\ \blacksquare & \blacksquare & \cdots & \blacksquare \end{bmatrix}_{77 \times 768}$$

可以想成有 77 個字。



會轉成和 Q 一樣維度的 K, V

E_{text}

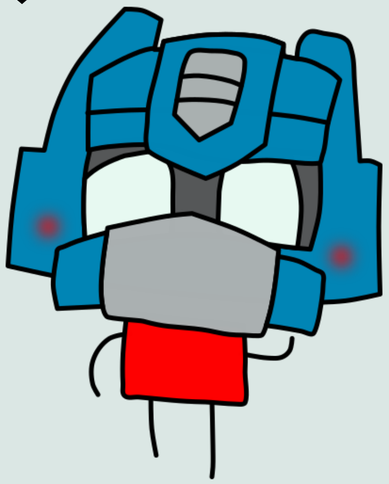
K

77×3

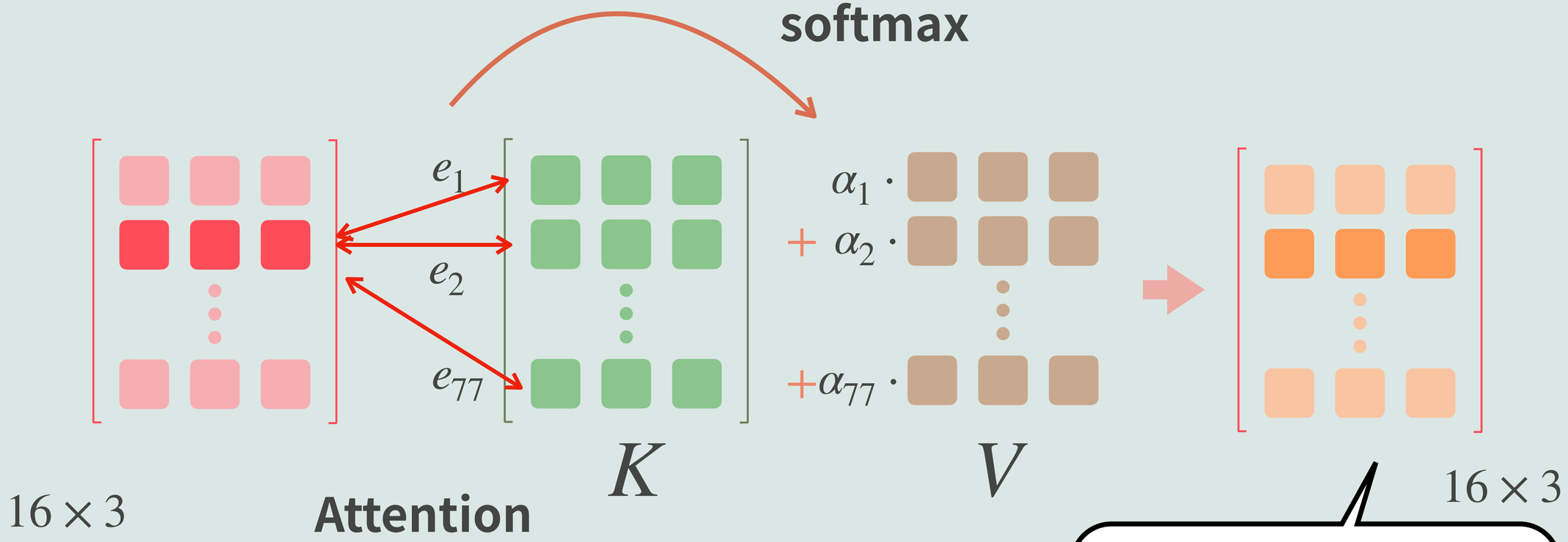
V

77×3

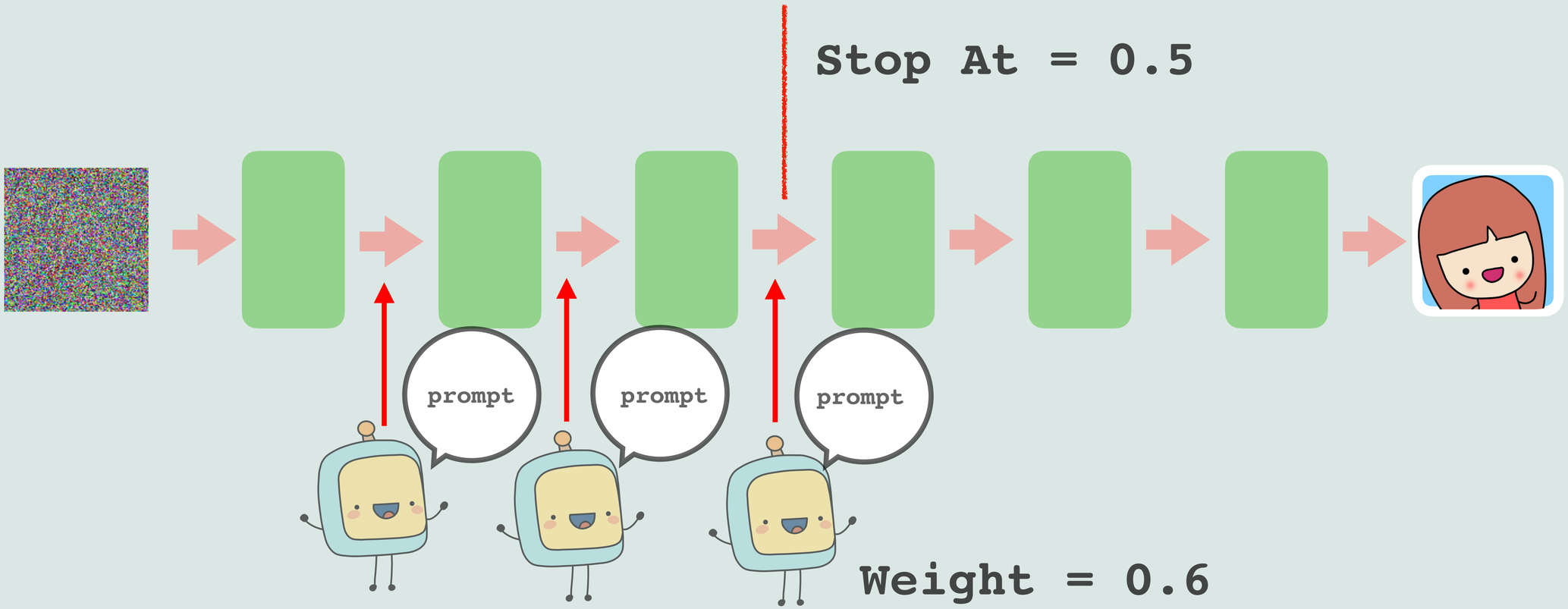
準備做 transformer。



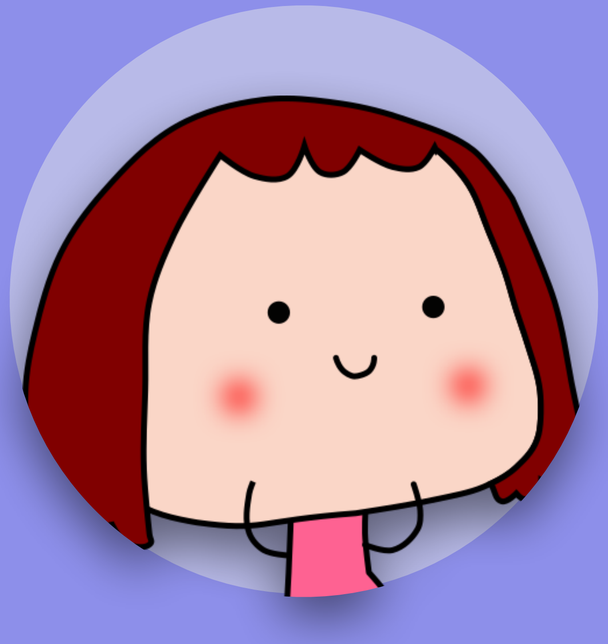
混進文字意思, 輸出原大小的「圖」



解碼 (生成) 最主要是用 U-Net



我們的「想法」至少有兩個參數可以調整

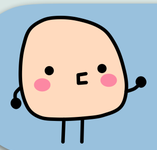


02.

不要 A 圖 — 排程器

Sampler

(Scheduler)

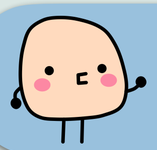


數列

$$a_1, a_2, \cdots, a_n, \cdots$$

$$\lim_{n \to \infty} a_n = ?$$

回想微積分的美好時光, 有一刻我們好在意一個數列, 也就是一串的數字, 是收斂還是發散的。



收斂和發散

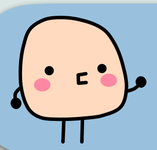
收斂的例子

9, 4, 8, 7, 6, 7, 2, 4, 3, 2, 1, 1, 1, 1, 1, ...

發散的例子

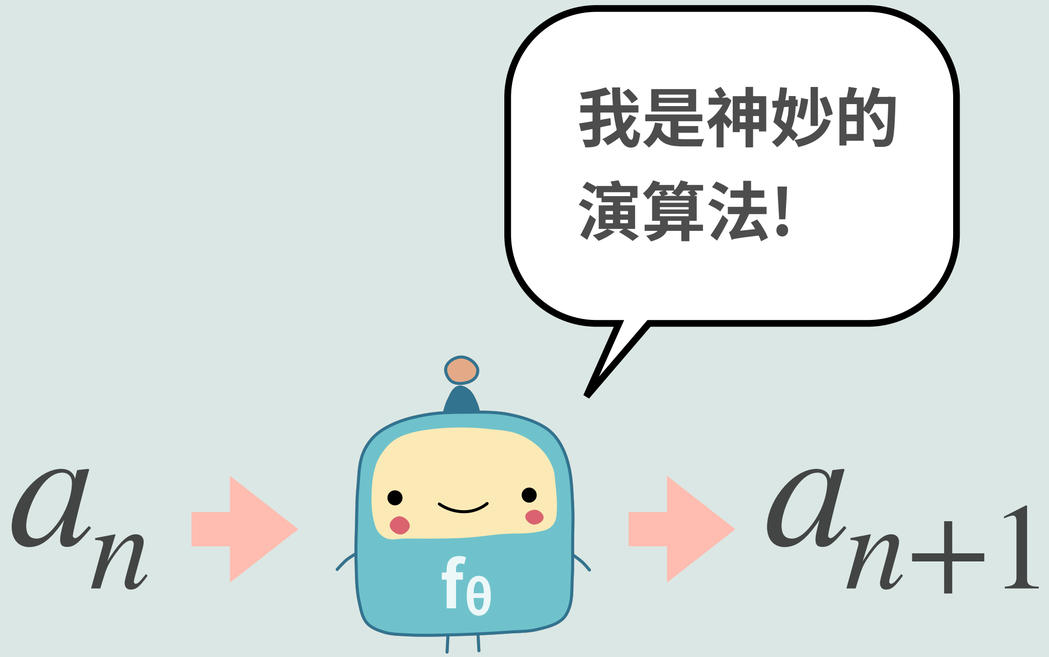
3, 7, 1, -1, 1, -1, 1, -1, ...

1, 2, 4, 8, 16, 32, ...

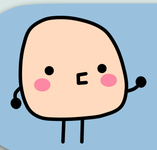


這其實是分析的標準手法!

我們想知道一個東西的答案, 但並不知怎麼立刻算出來。不過我們可以一開始先亂猜一個 a_1, 然後找個神妙的算法去調整一點點, 調整一點點, 最後能得到一個正確答案。



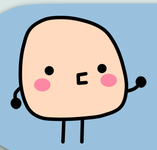
一般來說, 神妙演算法找出的數列起碼要收斂!



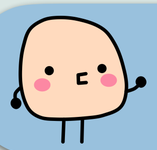
當然, 不一定要是數字!

$$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n, \cdots$$

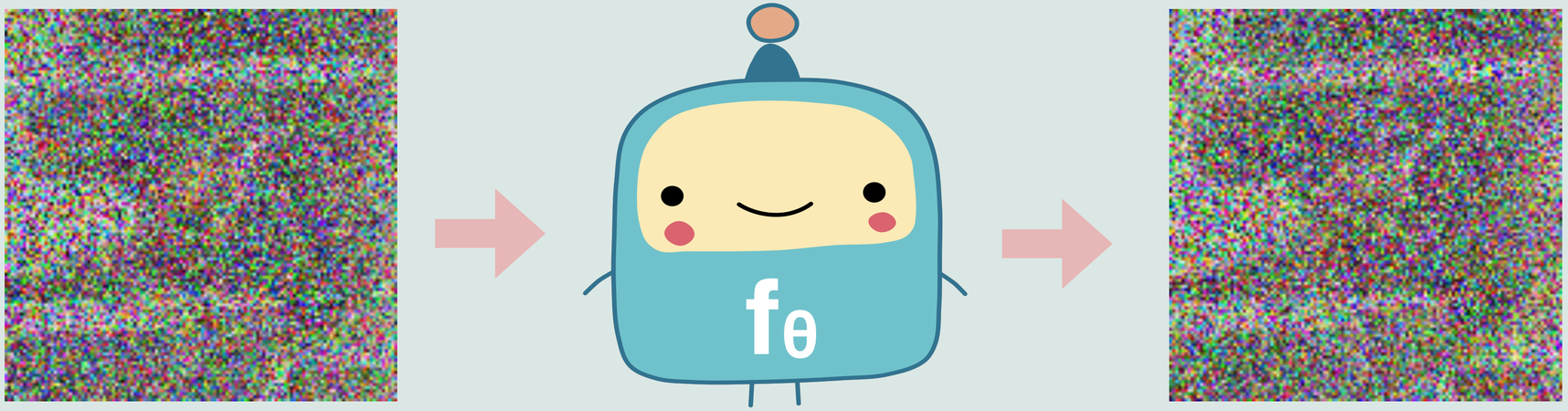
一串向量, 或是更一般的 tensors, 我們都可以考慮是不是收斂的。



相信至此, 大家都在想老師是不是弄錯了課程?



其實我們 Denoise 的過程就是在做這件事啊!



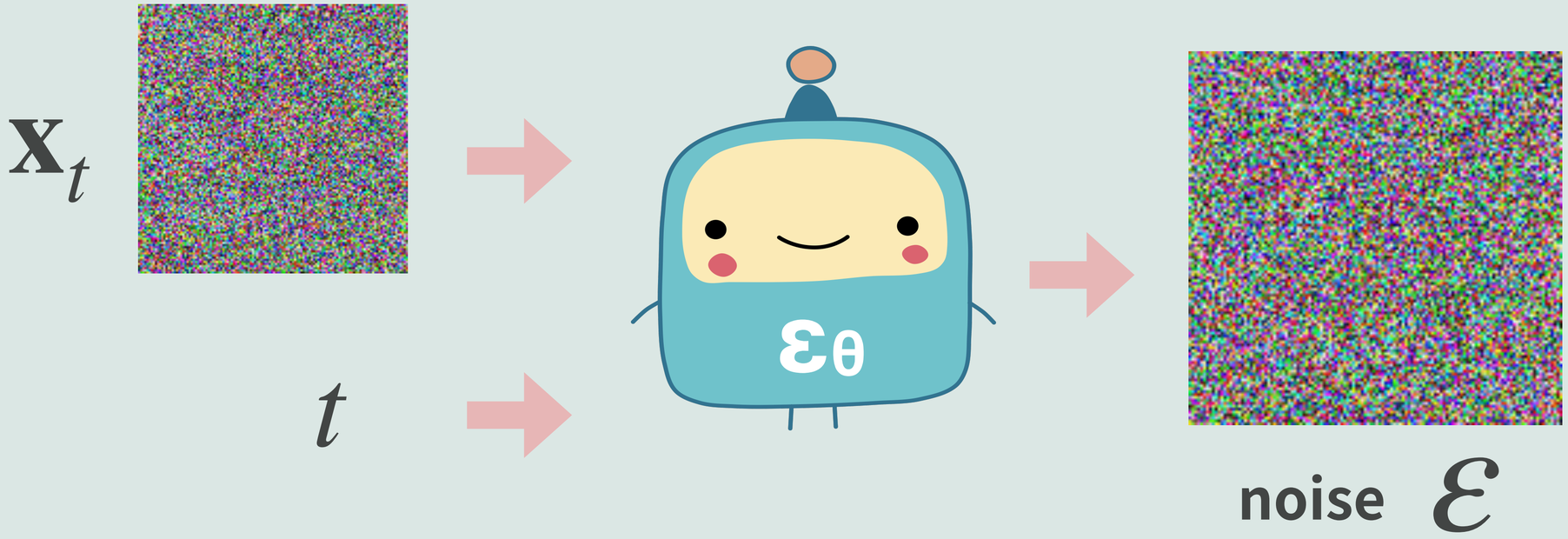
$\mathbf{x}_t$

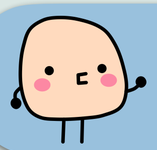
$\mathbf{x}_{t-1}$

Scheduler (Sampler)

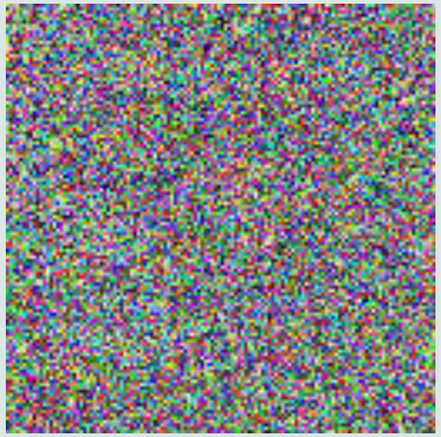
我們加上 noise 做了 1000 步達成, 但生圖總不希望再 1000 步才還原。因此有不同的 scheduler (sampler) 來加速。

詳細說我們學的其實是 noise



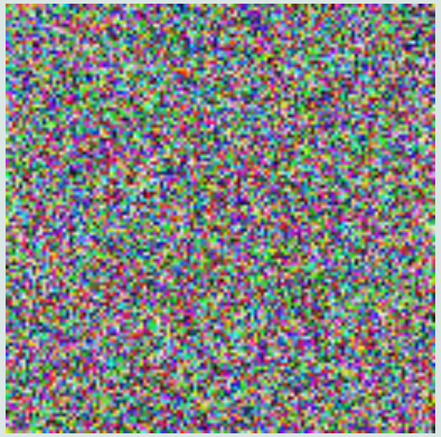


幻想中我們會得到...



$\mathbf{x}_t$

-

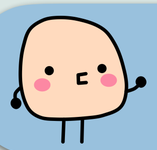


ε

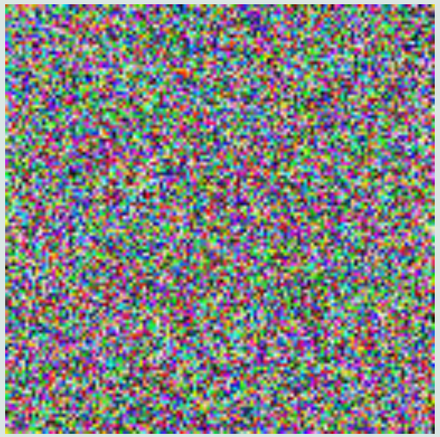
=

$\mathbf{x}_0$

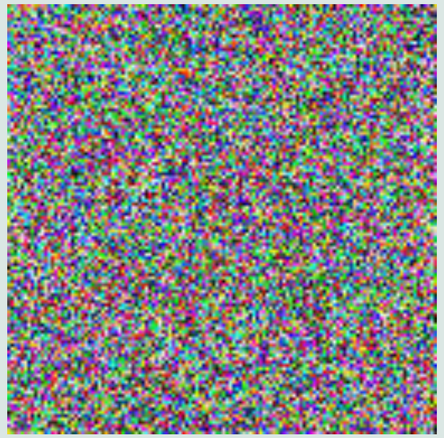
但真的想太多, 沒這麼厲害



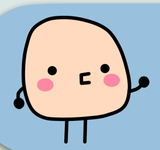
但可以估出某個時間點加的 noise



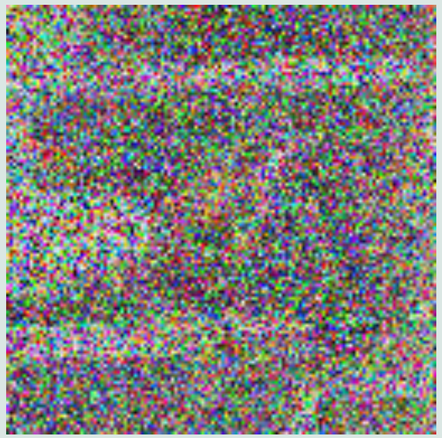
ε



ε_{t-1}

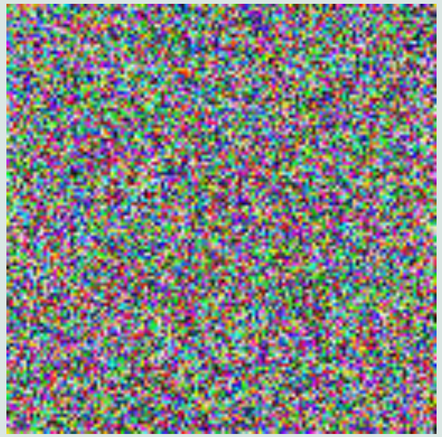


於是可以算出 $\mathbf{x}_{t-1}$



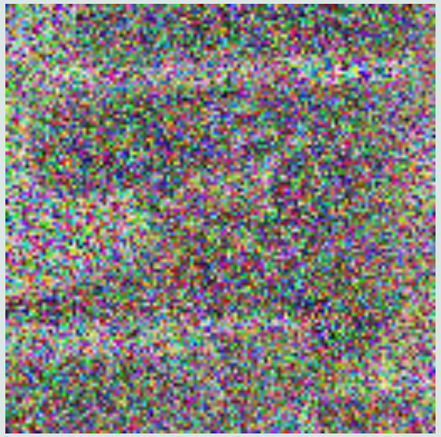
$\mathbf{x}_t$

$-$

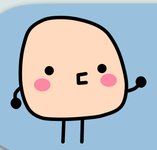


ε_{t-1}

$=$

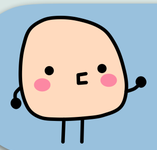


$\mathbf{x}_{t-1}$

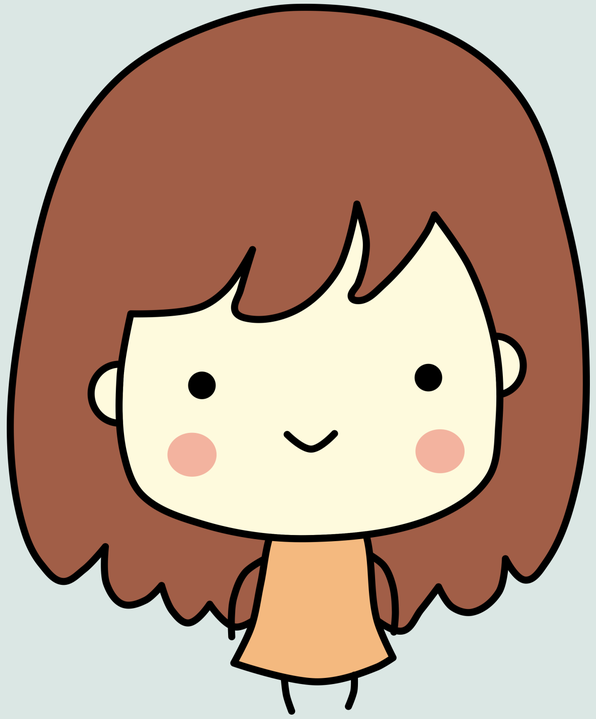


當然要估算任何時間段也都可以

$$\varepsilon_{t-1} + \varepsilon_{t-2} + \cdots + \varepsilon_{t-k}$$



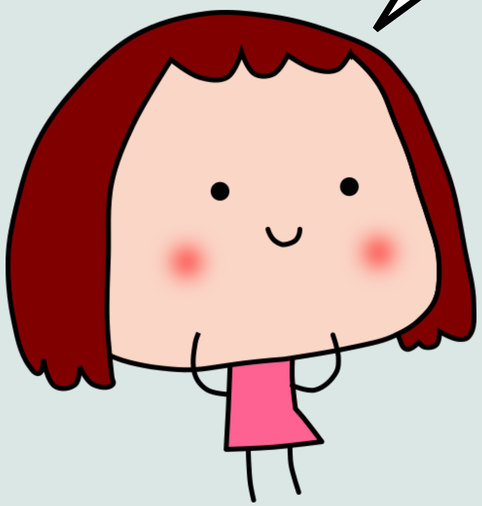
也就是可以估出...



所以我們不一定要走 1,000 步生出最終圖像!

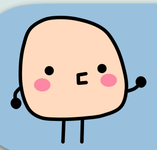
$$\mathbf{x}_t \rightarrow \mathbf{x}_{t-k}$$

幻想中, 算圖步數越多次, 圖就越美



step 20

step 100

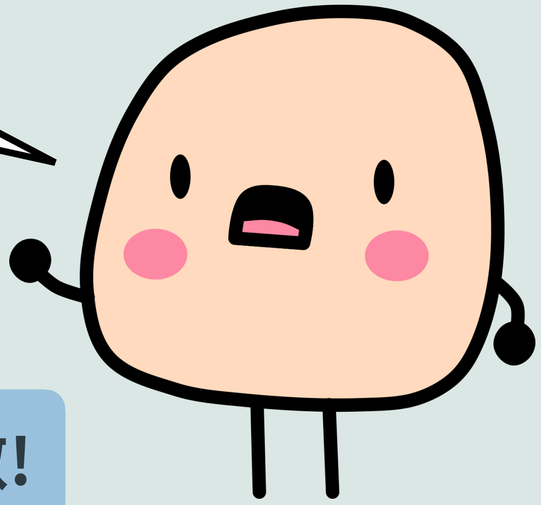


要注意 A 圖系的演算法!

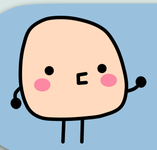
Ancestral schedulers

- Euler a
- DPM2 a
- DMP2 a Karras
- DPM++ 2S a
- DPM++ 2S a Karras

這些 a 系列的演算法,
在降噪過程中會透過
一些隨機 noise 技巧,
加速降噪過程。

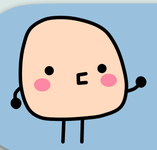


問題就是, 基本上不太會收斂!



Euler a 範例展示

我們來看一個用 Euler Ancestral scheduler 的範例。



開始比較可怕一點

step 5

這有點靈異片的感覺...

step 10

某種藝術?

step 15

開始有點樣子

第 20 步看來可以了, 但...

step 20

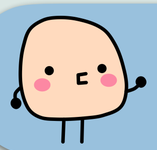
相當好, 皮衣看來不是真皮, 但不殺生很好

step 25

為什麼突然換了件衣服!?

step 30

又換回皮衣, 但人物反而沒那麼清晰, 還換了杯飲料



接下來我們快轉...

step 45

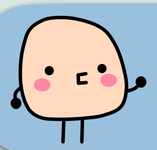
接下來都換牛仔外套, 而且好像覺得位子不好還換了位子

step 85

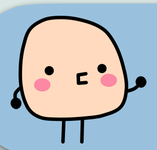
覺得太熱乾脆剪個頭髮, 而且換到室內

step 100

拍太久有點不耐的表情了, 我們就示範到這裡



其實不只 A 系列有隨機性

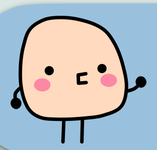


收斂系的推薦

- DPM++ 2M Karras: 20–30 步
- UniPC: 20-30 步

【強烈推薦延申閱讀】

https://stable-diffusion-art.com/samplers



非收斂系的推薦

- DPM++ SDE Karras: 8–12 步 (這是比較慢的演算法)
- DDIM: 10-15 步

【強烈推薦延申閱讀】

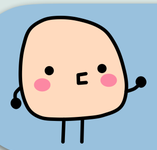
https://stable-diffusion-art.com/samplers

03.

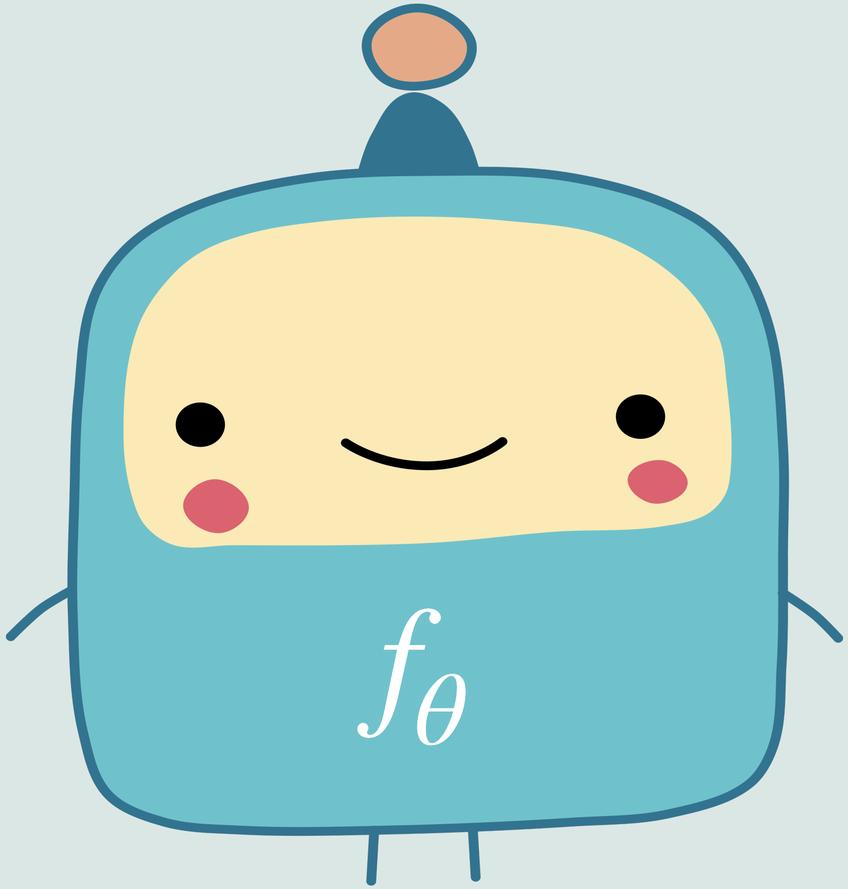
LoRA

LoRA

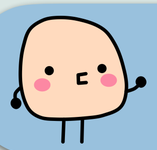
Low-**R**ank **A**daptation



一個訓練好的神經網路模型

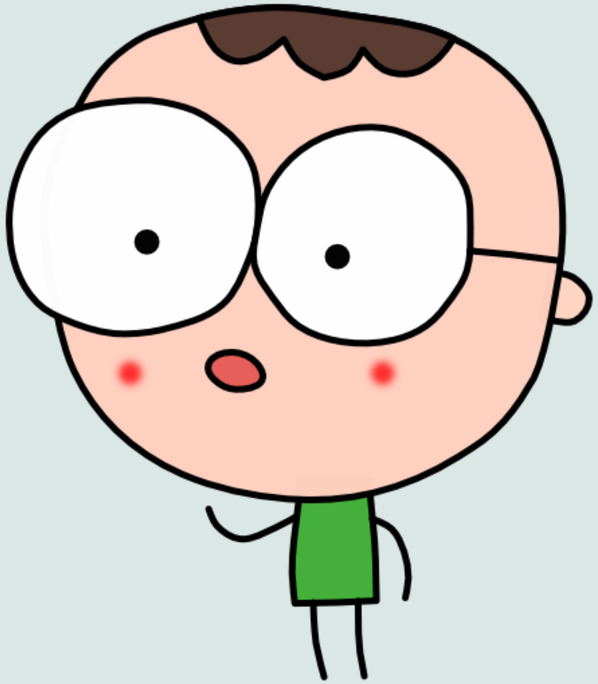


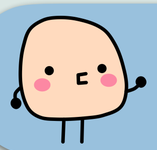
Stable Diffusion 好是好, 但原始的版本 (我們稱為**預訓練模型**) 的版本有些地方還是不盡如人意。所以我們常常會希望拿我們自己的數據再**微調**一下。



但常有幾個問題

- 現在 AI 都軍備競賽, 模型很大。我們的電腦可能沒辦法訓練, 或要訓練很久。
- 我們資料量通常小很多, 會不會這些數據破壞原來的學習成果?

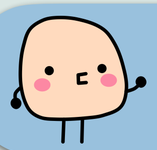




把 Stable Diffusion 的參數寫成一個矩陣

假設我們 Stable Diffusion 的參數是 $m \times n$ 的矩陣。

$$W$$

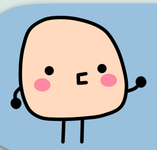


我們會「凍結」原來的 W

$$W + \Delta W$$

凍結

只調整這個

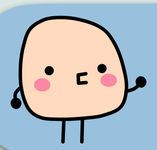


LoRA 的魔術來了

所以 k 選小一點的
數字就好了！

$$\Delta W = A \cdot B$$

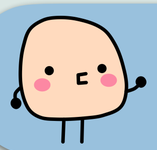
$m \times n$　　$m \times k$　　$k \times n$



這是 Microsoft 的論文

Edward J. Hu et al. (Microsoft), “LoRA: Low-Rank Adaptation of Large Language Models,” 2021.

https://arxiv.org/abs/2106.09685



哪裡可以找到人家訓練好的 LoRA 呢?

有名的模型、LoRA 等網站

https://civitai.com/

Hugging Face 上的 LoRA

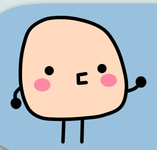
https://huggingface.co/models?other=lora

LoRA 使用時是這樣!

直接融入原本模型, 好像新的模型一樣!

可以調強度, 比如 0.7

$$W + \alpha \times \Delta W$$

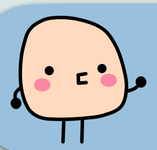


在有支援時的使用

調整混入權重

<lora:LoRA的檔案名稱:0.7>

在 prompt 加入。

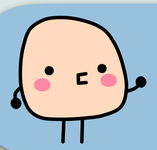


Checkpoint 是什麼?

$$W + 0.7 \times \Delta W$$

checkpoint

一個 checkpoint 是一組完整的模型參數, 比如我們把某個 LoRA 混入原來參數中, 新的參數就是一個 checkpoint。

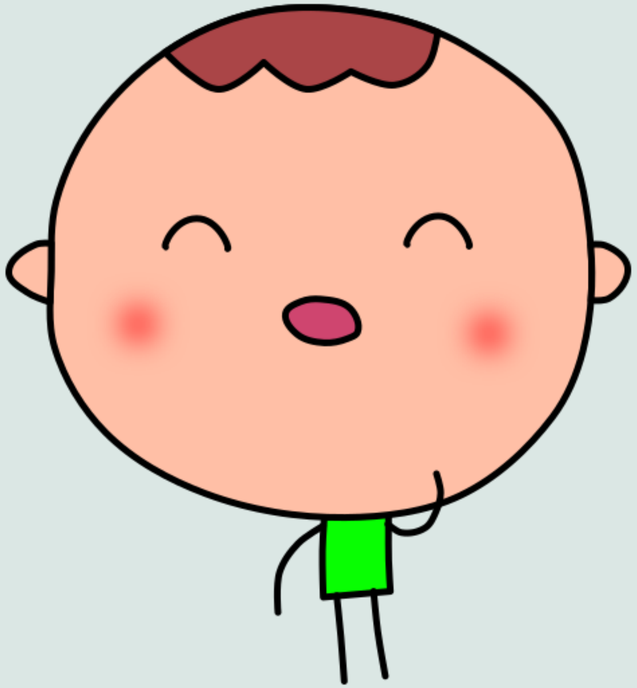


Stable Diffusion 存模型的檔案格式

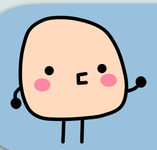
.ckpt

.safetensors

Diffusers 一個模型是一個資料夾



因此只有 .ckpt/.safetensors 時需要轉換 diffusers 才能用



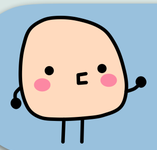
LoRA 存檔格式

.ckpt

.safetensors

04.

簡單易用圖像生成 Fooocus



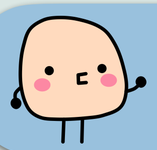
Stable Diffusion 的新 Web UI

目標是像 Midjourney 一樣
簡單的 Stable Diffusion。

Fooocus

fooocus github

https://github.com/lllyasviel/Fooocus



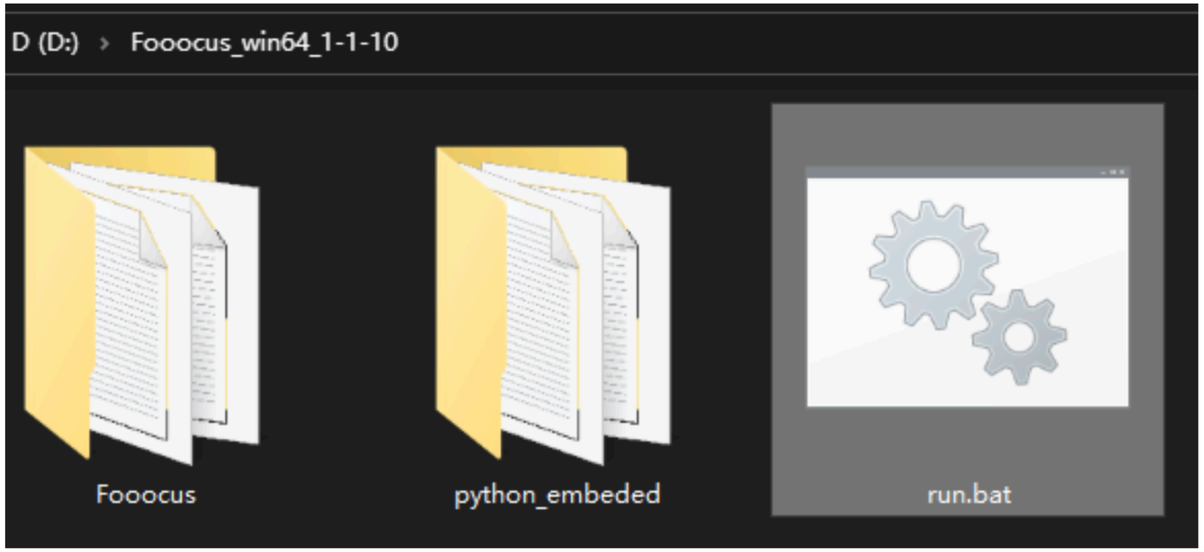
安裝: Windows 篇

Windows

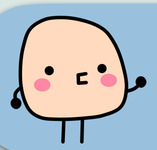
You can directly download Fooocus with:

>>> Click here to download <<<

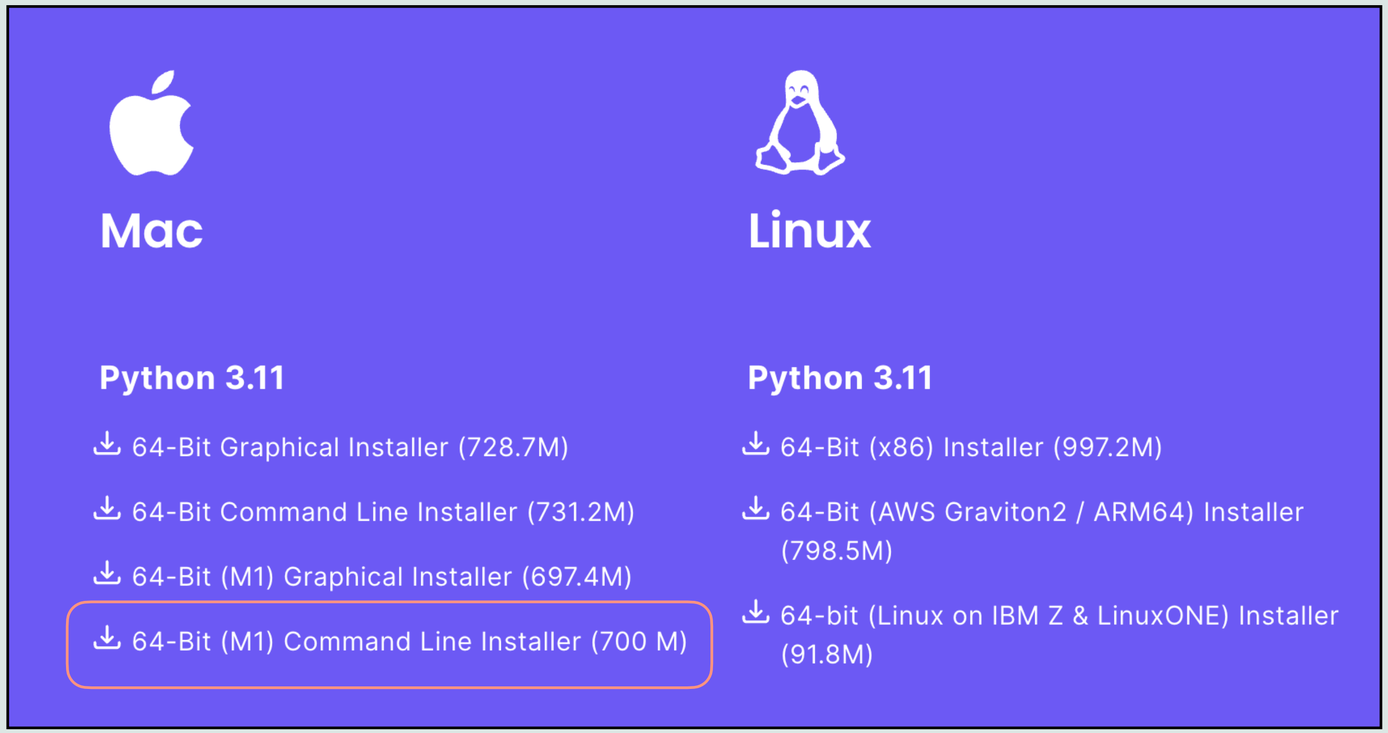
After you download the file, please uncompress it and then run the "run.bat".



找到下載, 解壓縮到未來想放的資料夾中, 找到 **run.bat** 執行。

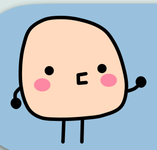


安裝: Mac/Linux 篇 — 安裝 Anaconda



https://www.anaconda.com/download

到 Anaconda 下載區, 找到適合自己平台的下載。注意 Mac M1/M2 等系列, 要選對版本。

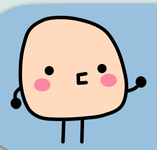


安裝: Mac/Linux 篇 — 安裝 Anaconda

進入下載那個檔案夾

```
> cd ~/Downloads
> sh Anaconda3-2024.xxoo
```

剛下載下來的檔案



安裝: Mac/Linux 篇 — 安裝 Fooocus

選擇在家目錄安裝

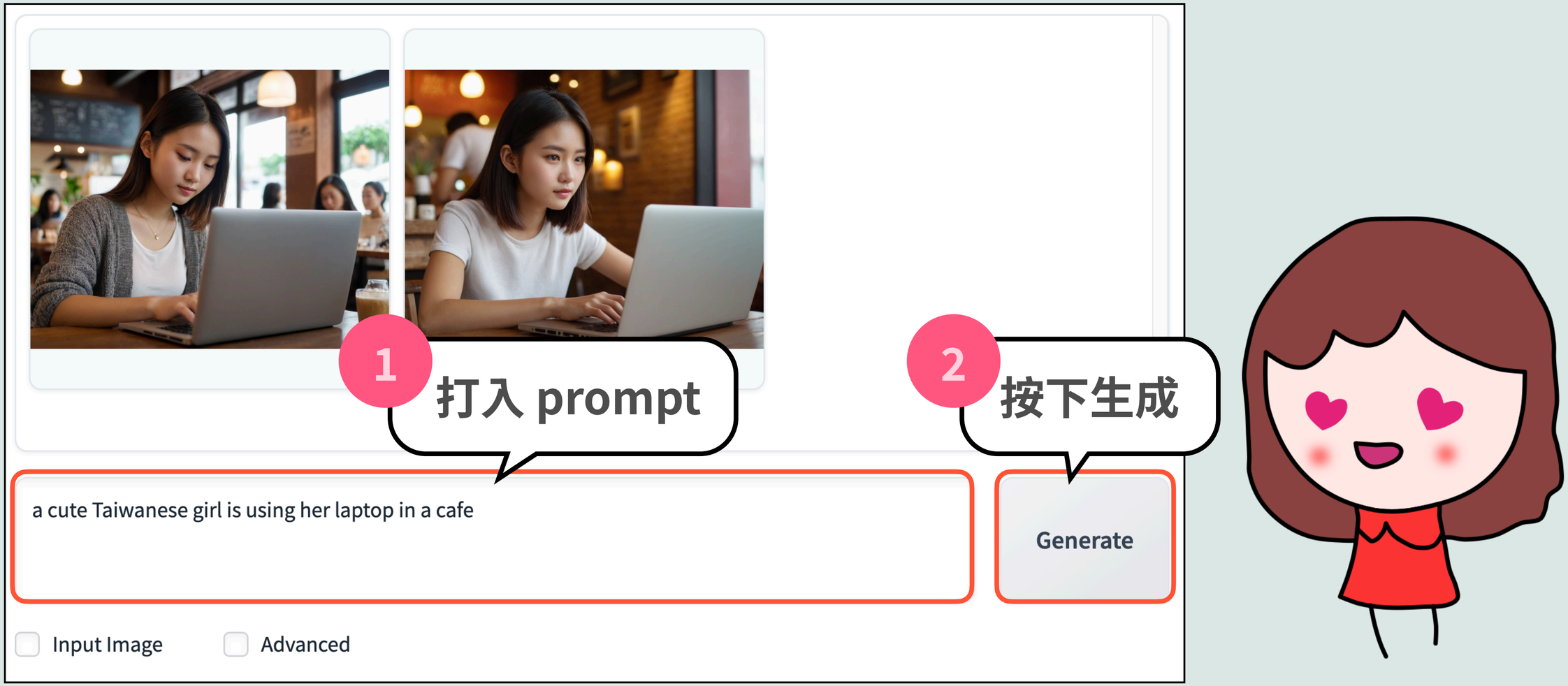
這會建一個叫 `fooocus` 的虛擬環境

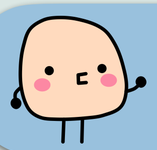
```
> cd
> git clone https://github.com/lllyasviel/Fooocus.git
> cd Fooocus
> conda env create -f environment.yaml
> conda activate fooocus
> pip install -r requirements_versions.txt
```

05.

Fooocus 基本使用

介面非常簡單!

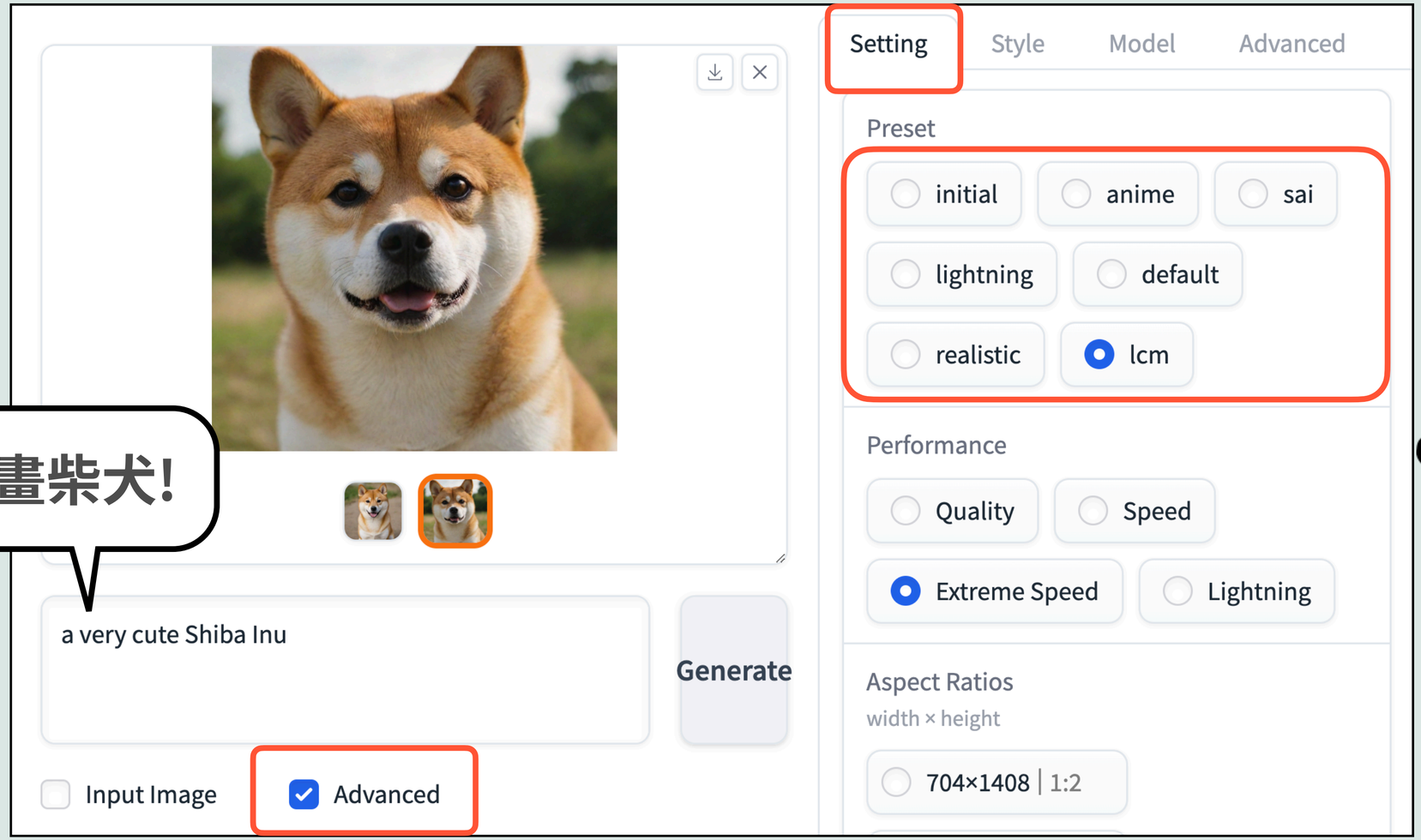

1
打入 prompt
2
按下生成
a cute Taiwanese girl is using her laptop in a cafe
Generate
Input Image
Advanced



預設就這樣的水準!

不需要下 negative prompt 哦 (因為 Fooocus 幫你下了)。

Advanced 下可以選不同 Preset



不同 Preset 設定下的結果

initial

anime

sai

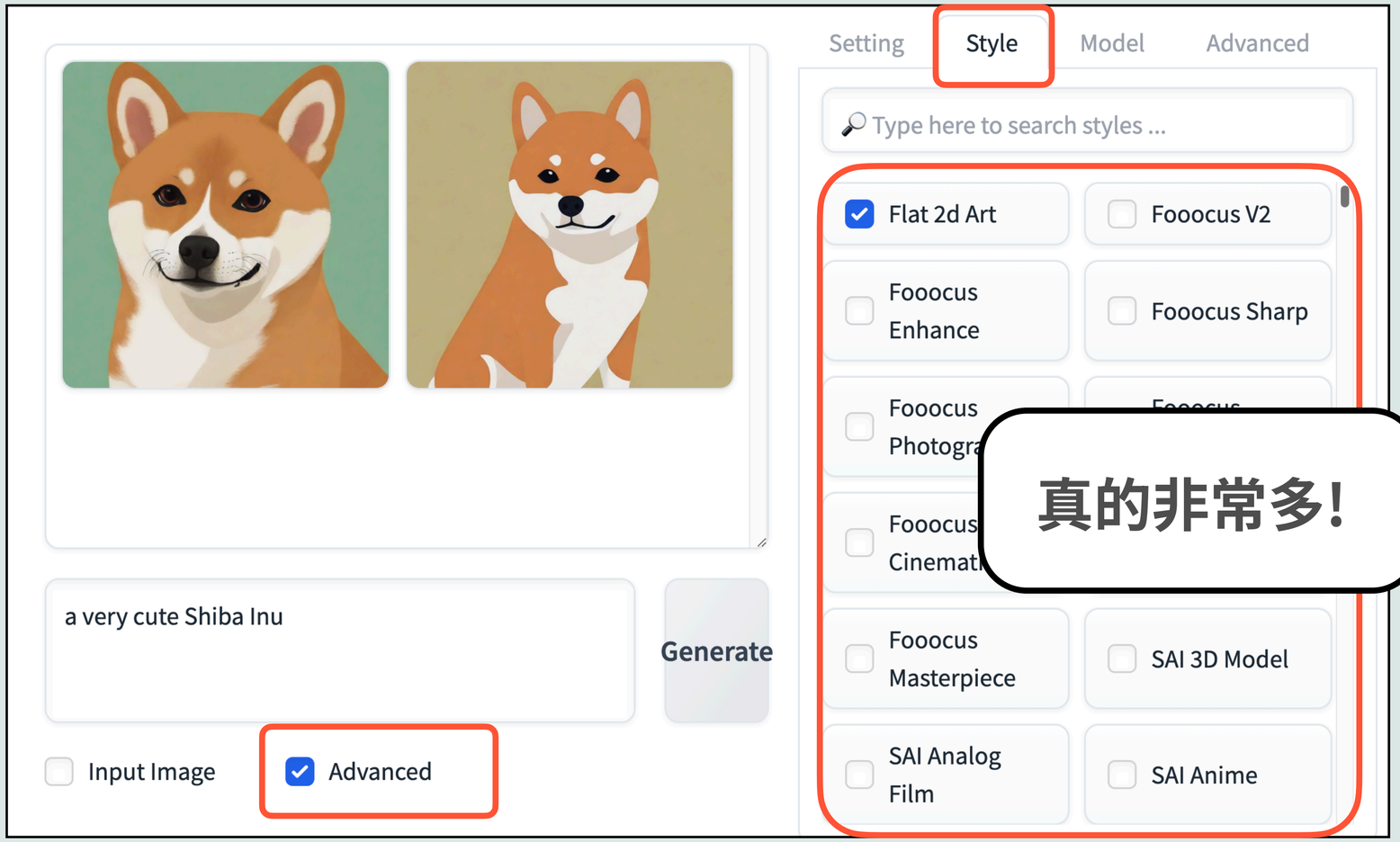
ligtning

default

realistic

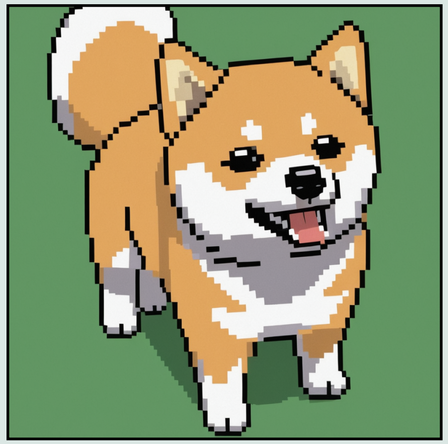
lcm

可以選不同風格 Style



不同 Style 呈現的樣貌 (只有一小部份)

Flat 2D Art



SAI Pixel Art

Mk Color Sketchnote

Papercraft Flat Papercut

SAI Origami

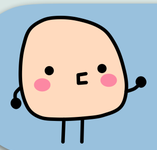
MK Tlingit Art

Game Pokeman

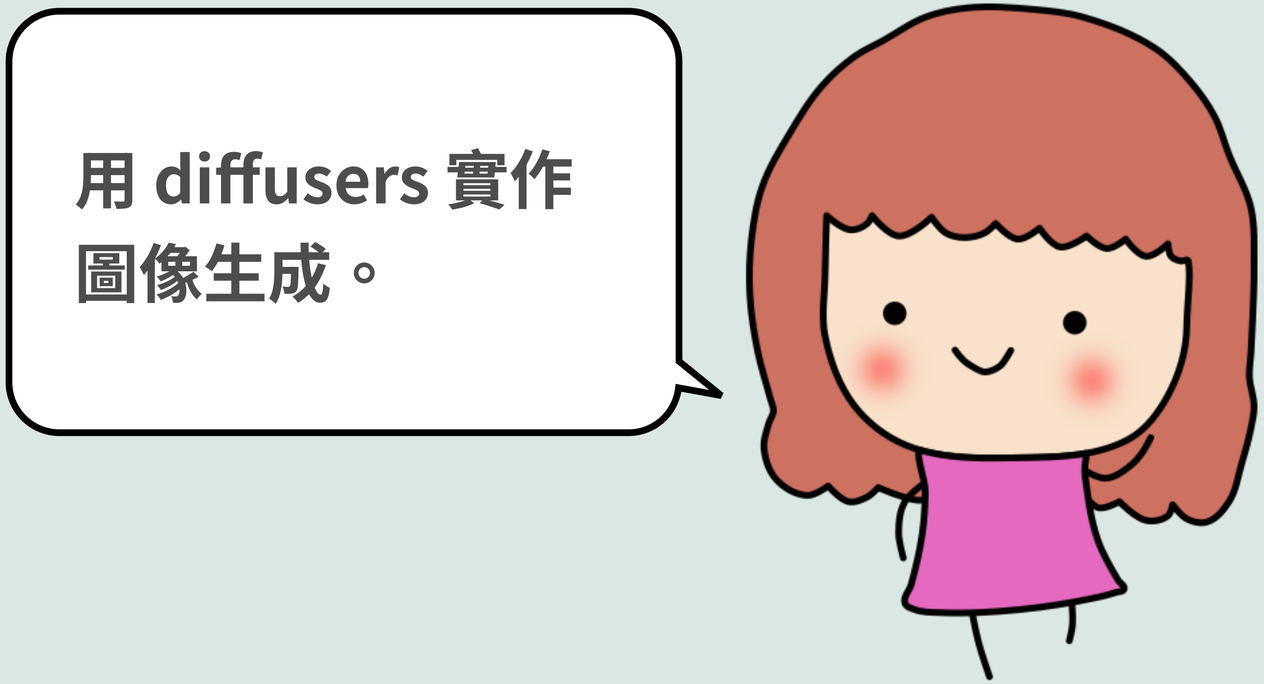
MRE Sumi E Detailed

06.

【附錄】用 diffusers 實作圖像生成



Stable Diffusion



https://yenlung.me/AI08

flat 2d vector art

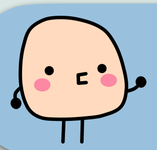
very cute (chibi girl), simple flat 2d, vector art, minimalist, white background

icon 設計

collection of icon designs for (animals), flat 2d, simple vector art

系統調校

我們來說明一下, 在 Colab 或是以後在自己電腦上一些細部的調校。



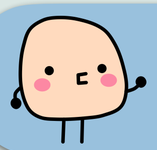
為什麼是 fp16 呢?

運算資源少的時候

```
pipe = StableDiffusionPipeline.from_pretrained("runwayml/stable-diffusion-v1-5",
        torch_dtype=torch.float16)

pipe = pipe.to("cuda")
```

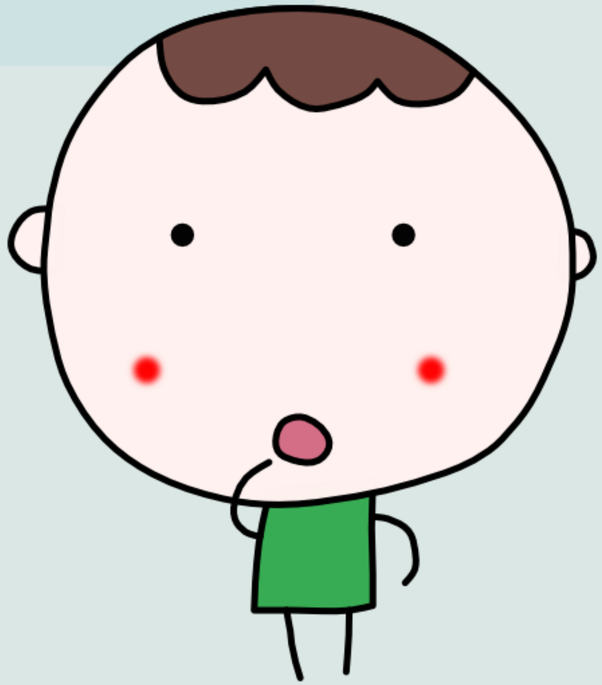
使用 GPU 計算, 用 Mac 請改 "mps"

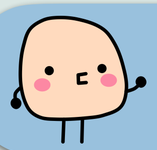


小於 64G 的 Mac/VRAM 小的時候

```
pipe.enable_attention_slicing()
```

VRAM 小的時候, 這樣可以切開來算, 省計憶體。這裡不針對 Mac, 而是 Mac VRAM/RAM 是共用的, 所以記憶體很容易被吃掉。不過你買了 96G RAM 的 Mac 當然就不用擔心了...

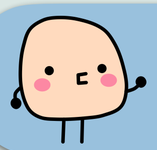




讓 GPU 休息

```
pipe.enable_model_cpu_offload()
```

GPU 在執行完一個任務中間應該有個「休息」時間, 讓 GPU 自動休息。

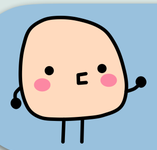


Mac 需要暖機!

```
prompt = 'a happy rabbit'

_ = pipe(prompt, num_inference_steps=1)
```

事到如今, 還是沒人知道為什麼。但
Mac 需要做一次「暖機」動作才會正常
運作。這讓 Mac 隨便畫就好。



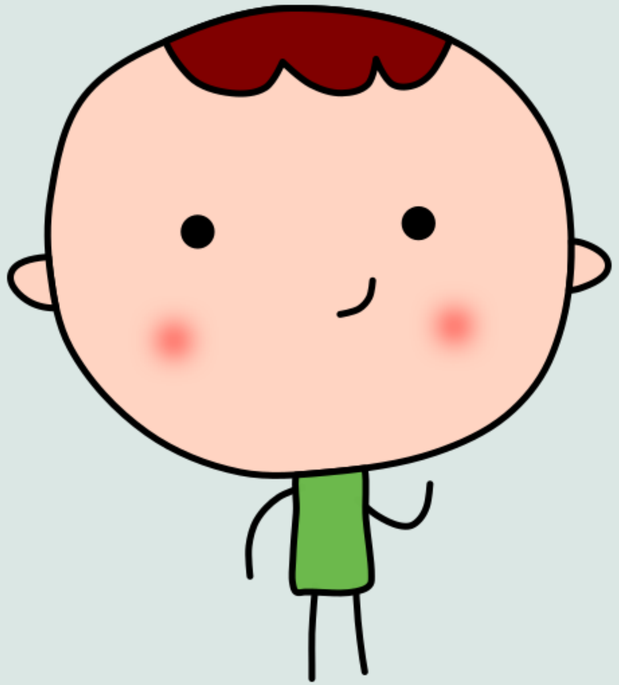
我們來做個有趣的練習

主題

動作

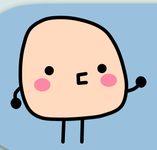
風格

政大新生書院的導師立威老師，
想了個有趣的方法讓大家試著畫
不同的東西。



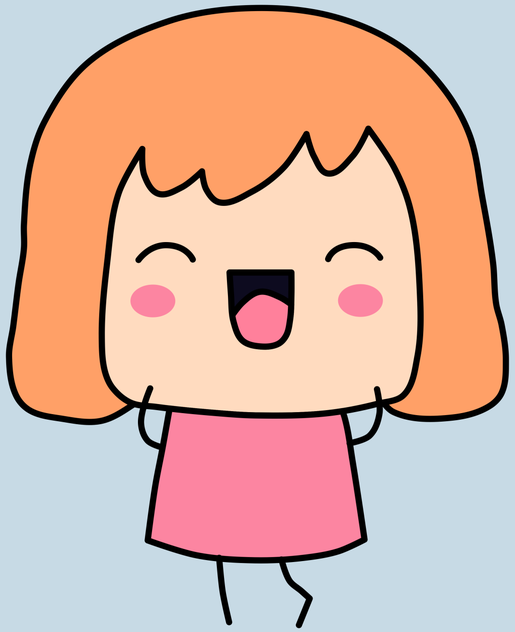
讓電腦自己產生天馬行空的想法!

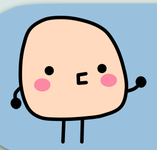
a dog rinding a bicycle, Joan Miro style



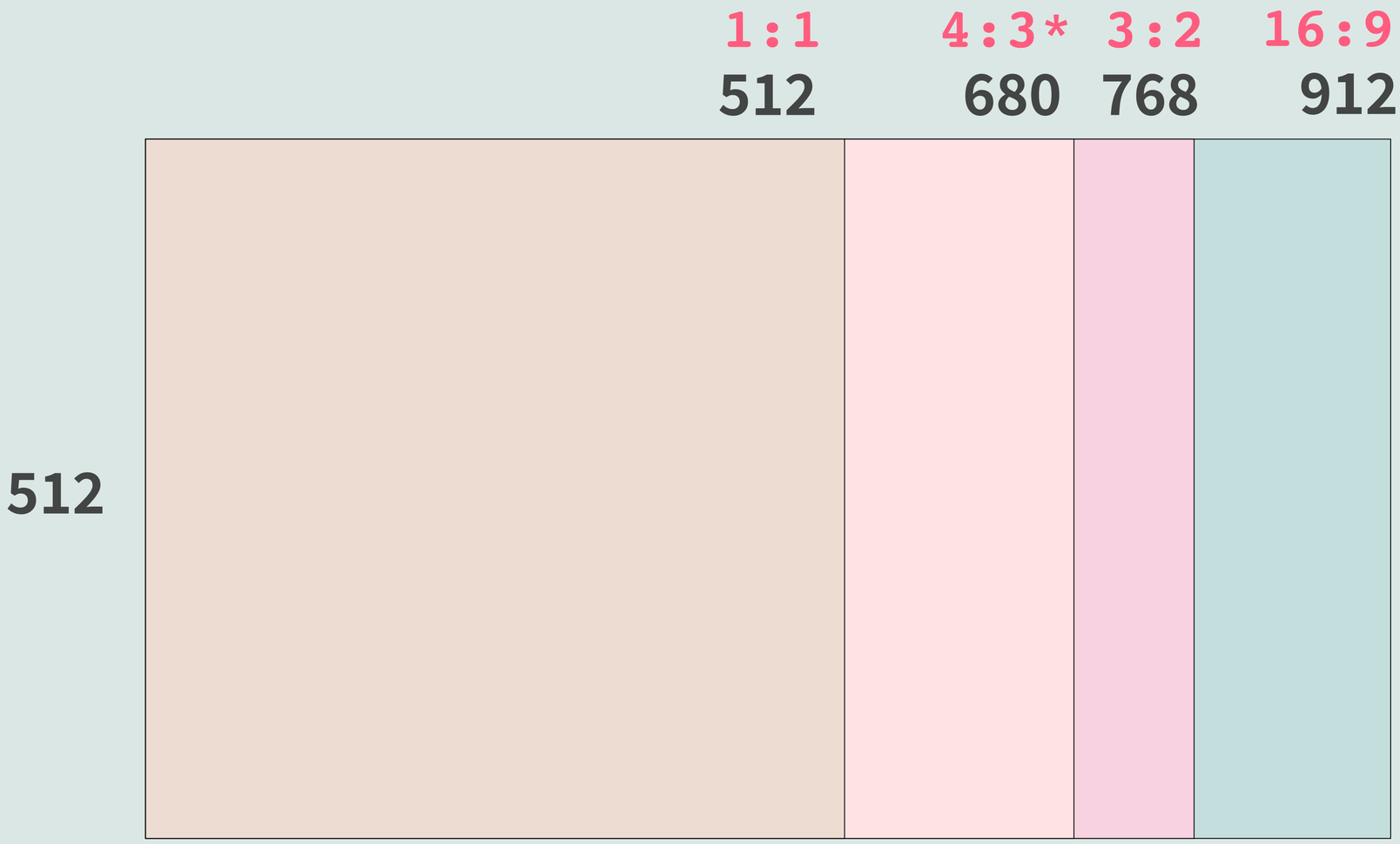
常用 pipeline 的參數

- ☑ `prompt:` 咒語
- ☑ `negative_prompt:` 反向咒語
- ☑ `width:` 寬 (預設 512)
- ☑ `height:` 高 (預設 512)
- ☑ `num_images_per_prompt:` 生幾張圖
- ☑ `num_inference_steps:` 幾步生成 (預設 50)
- ☑ `guidance_scale:` GFC scale, 越高越符合 prompt (預設 7.5)
- ☑ `generator:` 生成器設定 (如 ramdom seed)



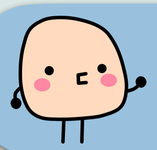


關於圖的大小



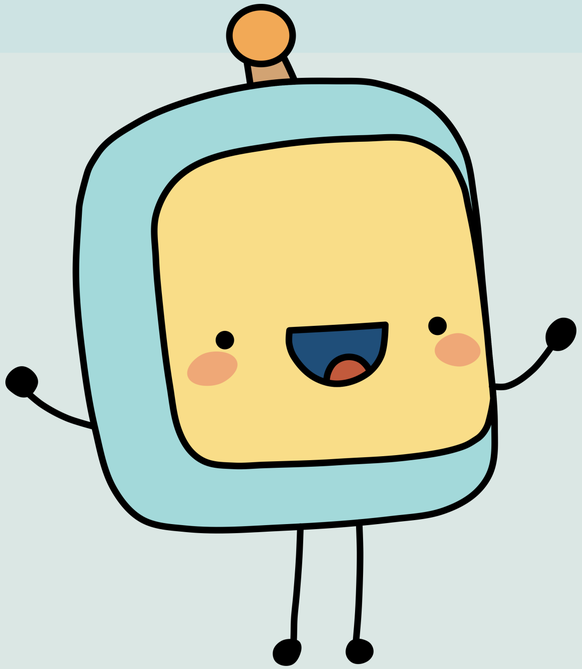
Stable Diffusion 建議一邊是 512, 另一邊仍是 8 的倍數。

* 只是近似



亂數種子是控圖的關鍵!

```
torch.Generator(device="cpu").manual_seed(r)
```



設定 generator 用某個固定的亂數種子。

https://hackmd.io/@yenlung/random_seed

幾個可以試試的 SD 1.5 模型

stablediffusionapi/sdvn5-3dcutewave

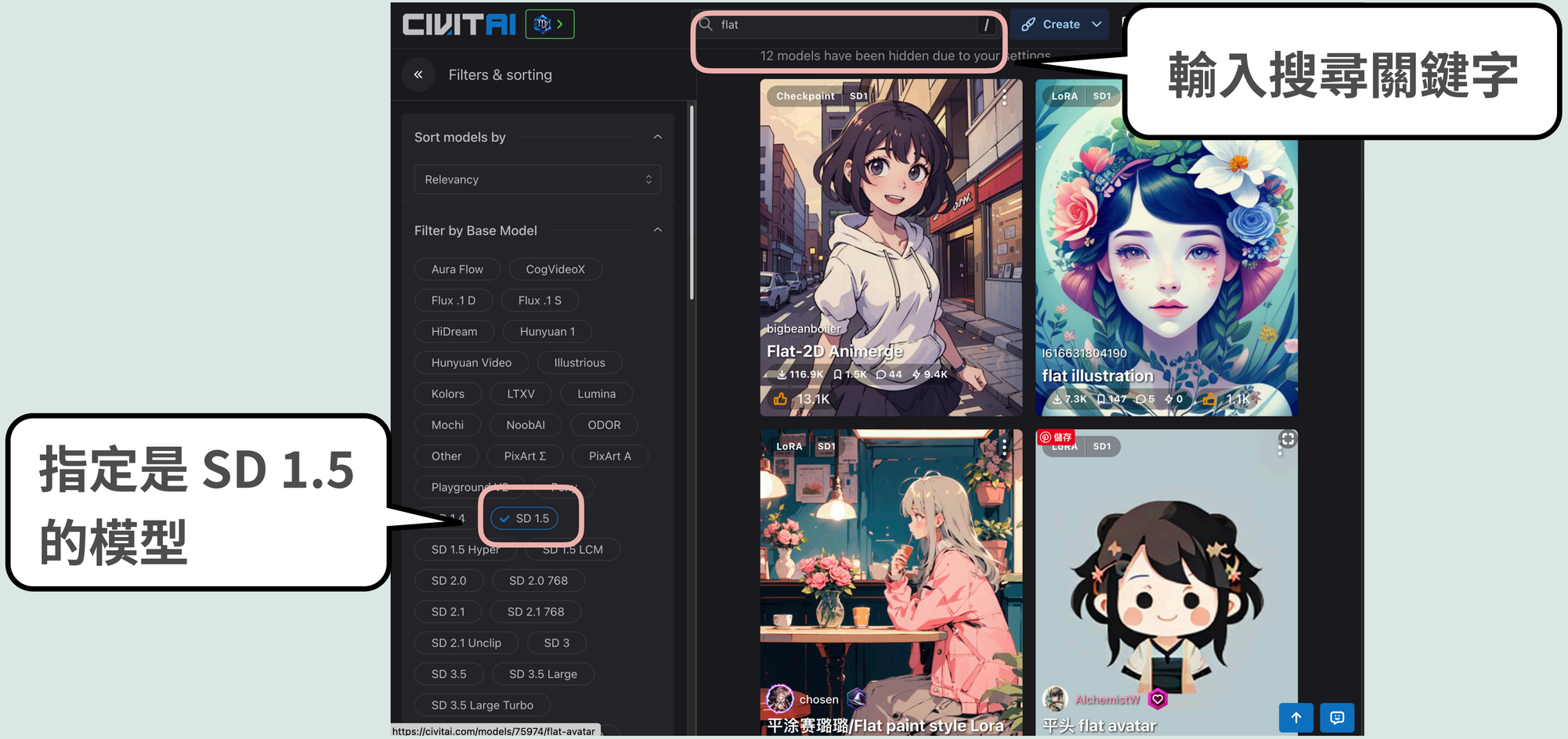
Lykon/dreamshaper-8

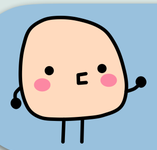
stablediffusionapi/chilloutmix

SG161222/Realistic_Vision_V6.0_B1_noVAE

danbrown/RevAnimated-v1-2-2

找模型最簡單可能是去 CivitAI 上找





打造自己的圖像生成 Web App!

MajicMIX v6 互動圖像生成器

歡迎使用！輸入提示詞、選擇設定，立即生成你的寫實風格作品！

Prompt

ice cream sundae

加強 Prompt

加強內容

masterpiece, ultra high quality, intricate skin details, cinematic lighting

使用 Negative Prompt

Negative Prompt 內容

bad anatomy, blurry, disfigured, poorly drawn hands, extra fingers, mutated hands, low quality, worst quality

自訂 Random Seed

指定 seed (選填)

42

高度 Height

512

寬度 Width

512

生成步數 (Steps)

20

10 50

生成張數

4

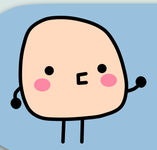
1 4

開始生成！

使用的 Random Seeds

使用的 random seeds: [2398942625, 2398942626, 2398942627, 2398942628]

https://yenlung.me/AI08g



作業: 使用 Fooocus 來創作!

* 先用英文概念 (可用 LLM 來翻譯), 提供給 Fooocus 來創作。
* 先用任何語言模型, 寫個你要生出圖的概念。然後用 Gemini Pro 或其他 LLM, 擴充更詳細的概念, 並且翻成英文。再畫一次。
* 比較兩個作品的差異。