StyleGAN 沿革

PGGAN → StyleGAN → pSp

Progressive GAN

(簡稱 PGGAN 或 ProGAN)

初始 GAN 的改良，發表於 2017

以【漸進式的訓練】使能夠生成高解析度的圖

傳統 GAN 的問題

- 模式崩潰 (Mode collapse)
  - 生成數據只是原始數據的子集（如：訓練數字 1~9，但最後只產生1、7、9）
  - Generator 發現產生固定類別的圖很容易騙過 Discriminator，所以偷懶只產生某些類別
- 難訓練高解析度圖片
  - 起初生成器的參數是隨機的，剛開始生成的爛圖很難騙過鑑別器，所以反向傳播時會反饋及大的梯度給生成器
  - 但參數大範圍更新有機率導致模型無法收斂，或是脫離收斂

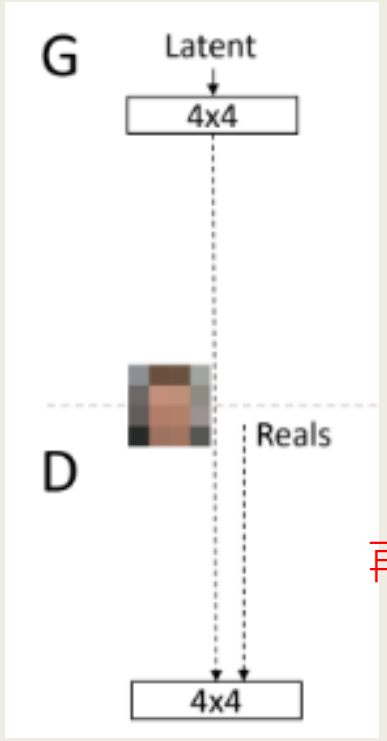
PGGAN 的改良點(1/4)

- 漸進式訓練 (Progressive Growing)
  - 從低解析度圖像開始訓練，逐漸提高到高解析度
  - 提高生成品質、降低模型崩潰可能性、降低訓練時間

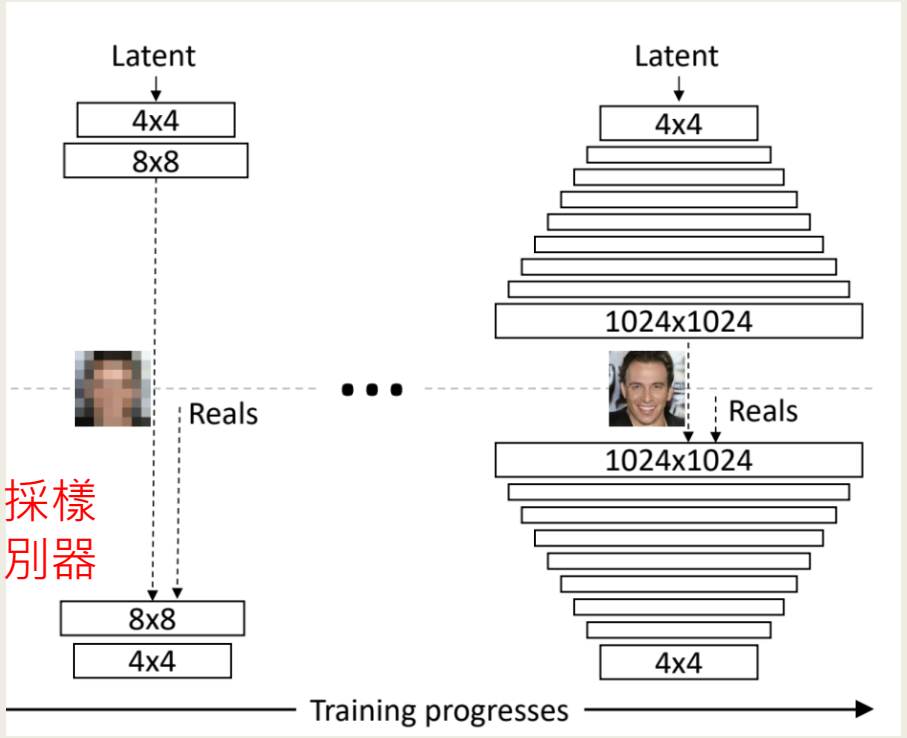
經過 4x4 生成網路產生
(512 x 4 x 4)

通道數轉為3，變成這張圖片
(3x4x4)

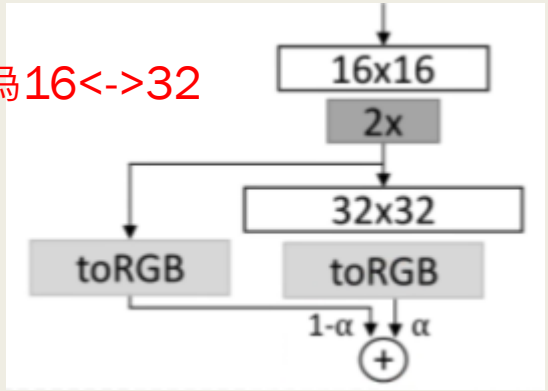
輸入影像給 4x4 的鑑別器
判定真偽



先做下採樣
再給 4x4 鑑別器

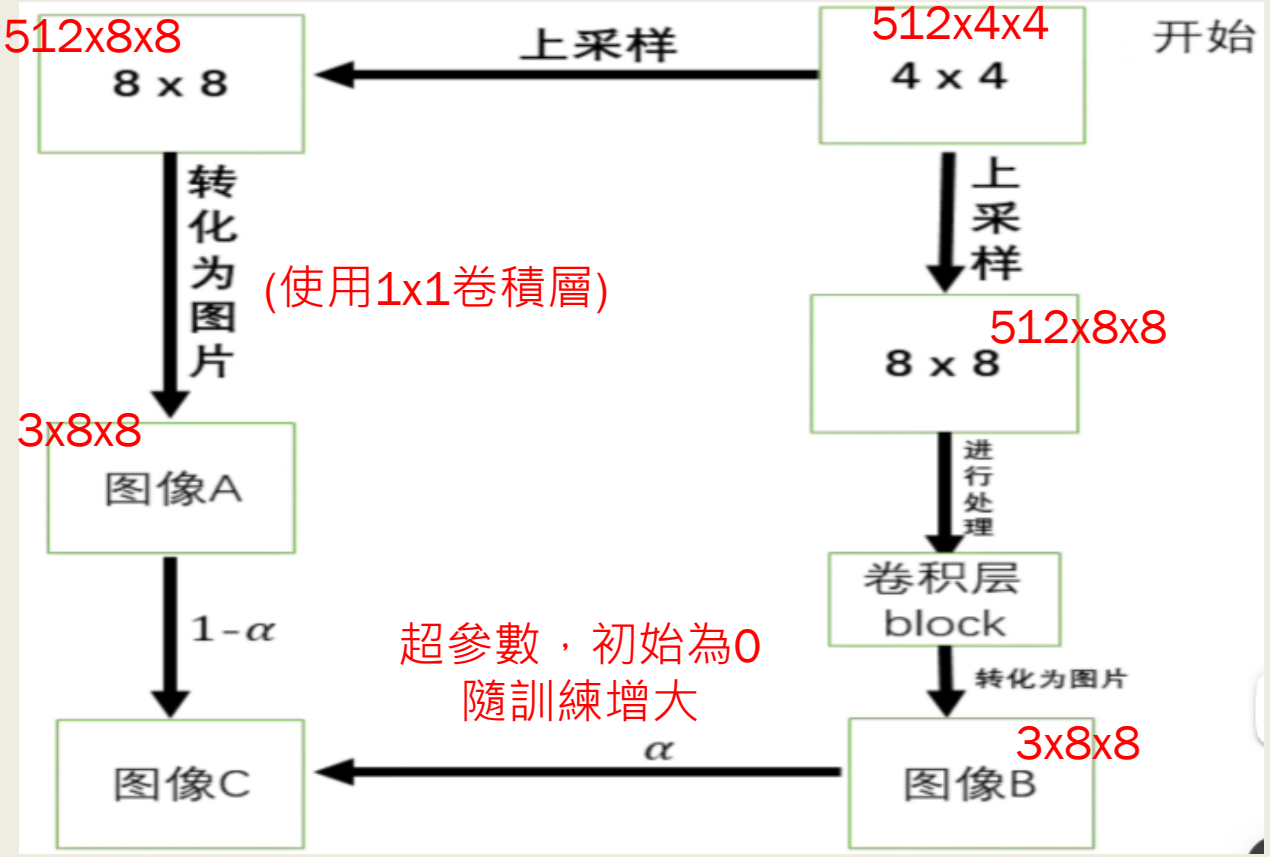


PGGAN 的改良點(1/4)_漸進式訓練



- 平滑過度(維度擴充的**緩衝**~)
  - 基本上是加權求和
  - 剛訓練8x8時更願意相信圖像A (因為圖B是經過才剛初始化的卷積來的)
  - 所以 α 初始才為 0
- 上採樣做法

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{bmatrix}$$



圖像C = 圖像A*(1-α) + 圖像B*α

PGGAN 的改良點(2/4)

- 均衡學習率 (Equalized Learning Rate)
  - 模型參數 w 從標準正態分佈中初始化，前向傳播時對 w 縮放，進行約束
    - c : 初始化標準差常數
    - i : 輸入神經元數量
    - n : 例如對於卷積層，n 為輸入通道數*卷積核高*卷積核寬

但程式碼中似乎用乘法→不知為啥

$$\hat{w}_i = w_i / c_i$$

$$c_i = \sqrt{\frac{2}{n_i}}$$

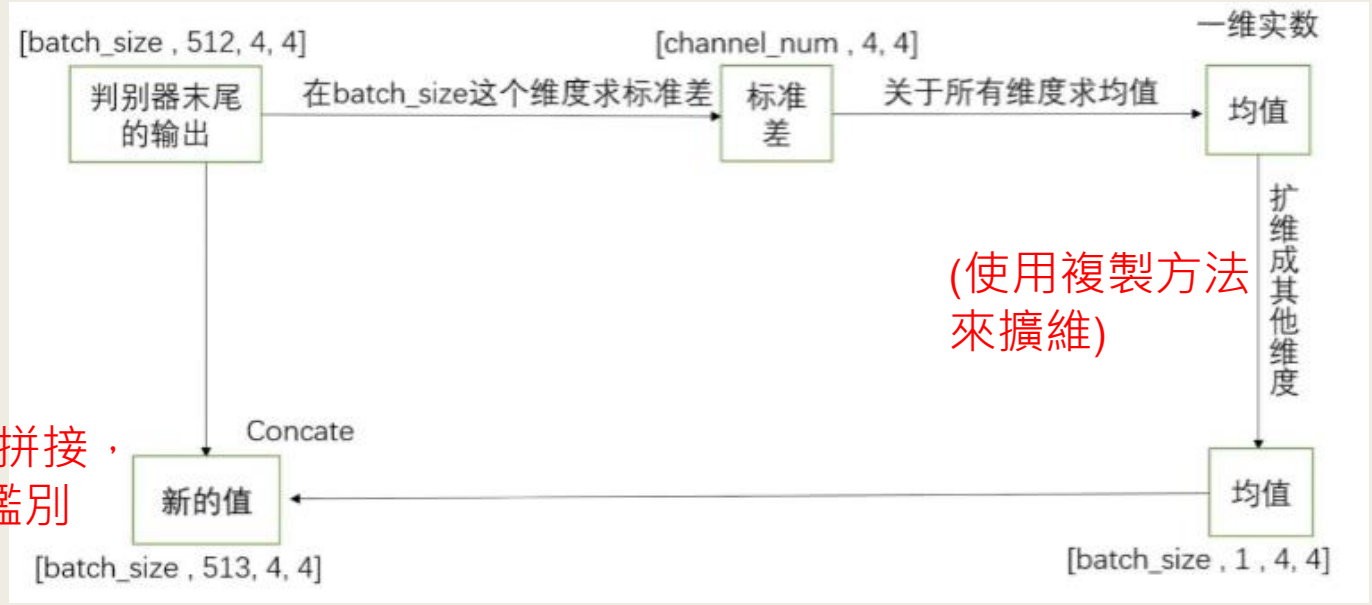
  - 平衡所有權重 w 的學習速率，不因優化做出誇張的調整
- 像素特徵向量的歸一化 (Pixel Normalization)
  - 沿著 channel 維度做歸一化
  - 避免生成器梯度爆炸

$$b_{x,y} = a_{x,y} / \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (a_{x,y}^{j})^2 + \epsilon}$$

PGGAN 的改良點(3/4)

- 在鑑別器使用使用小批量標準差 (Mini-batch standard deviation)
  - 一批算出的標準差高，表示裡面包含各種類別的影像，所以之間的差異較大
  - 鑑別器若覺得這批標準差太低，八成是假的 (真實情況應該要大)
  - 因此生成器要騙過鑑別器的話，就只能設法增加生成樣本的標準差

→增加生成樣本多樣性



在channel維度做拼接，接著將此值送給鑑別器看輸出即可

PGGAN 的改良點(4/4)

- 對鑑別器的 loss 加上一個極小的權重
  - 作者在訓練數據集 CELEBA-HQ 時提出的
  - 防止鑑別器在反向傳播時，梯度為零的情況

$$\hat{L} = L + \epsilon_{drift}\mathbb{E}_{x\sim P_r}[D(x)^2]$$

其中L、$\hat{L}$分别代表初始的损失和加上极小权重项的损失。$\epsilon_{drift} = 0.001$，P_r代表原始数据，E代表数学期望

PGGAN conclusion

- 提出四大改良
  - 漸進式訓練 (Progressive Growing)
  - 均衡學習率 (Equalized Learning Rate)
  - 像素特徵向量的歸一化 (Pixel Normalization)
  - 使用小批量標準差增加生成樣本多樣性 (minibatch)

- 主要在解決傳統 GAN 的模式崩潰、無法產生高解析度圖片的問題

StyleGAN

發表於 2018 年底

對生成的圖像進行更好的控制和理解

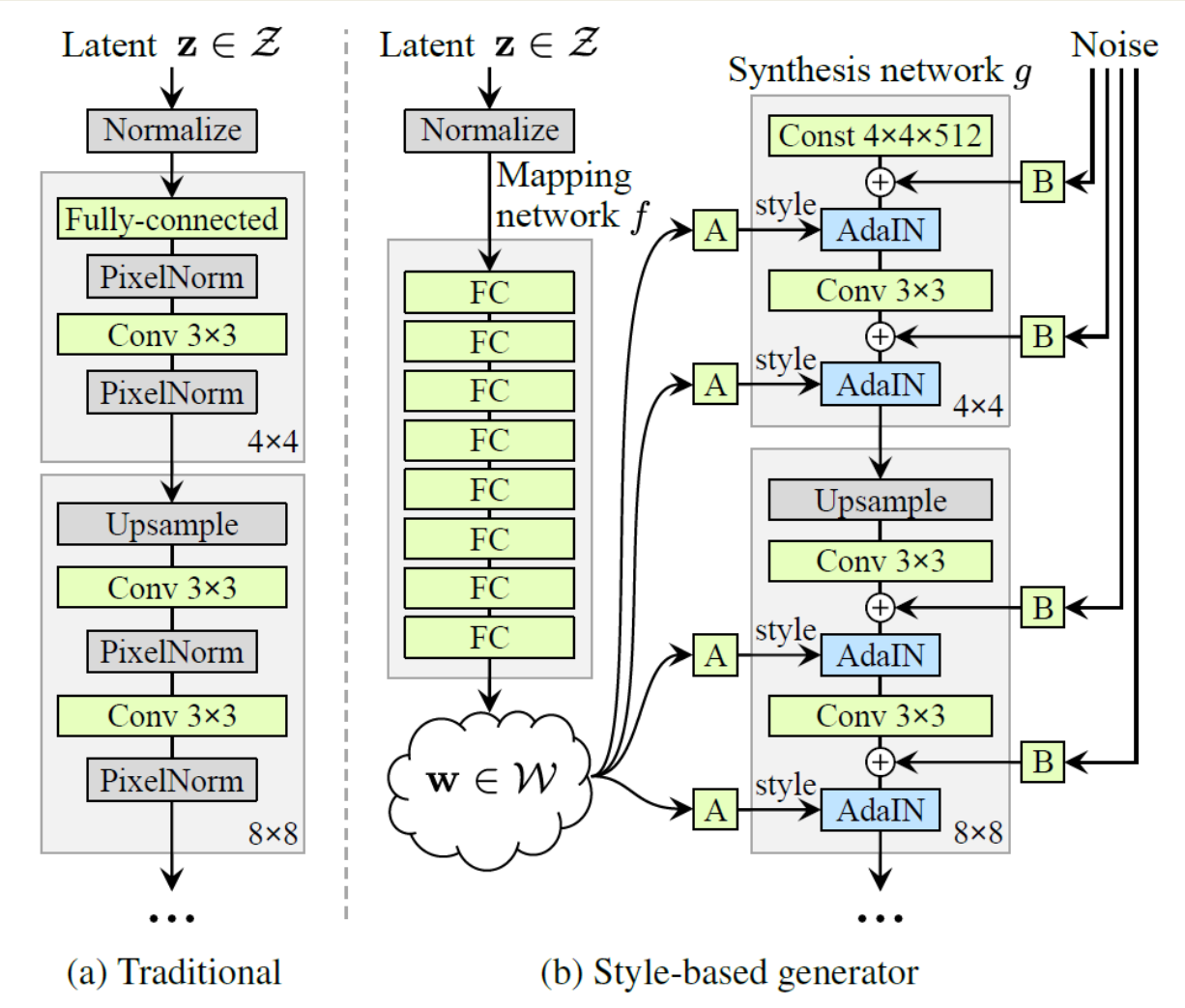
先前 GAN 的困難點

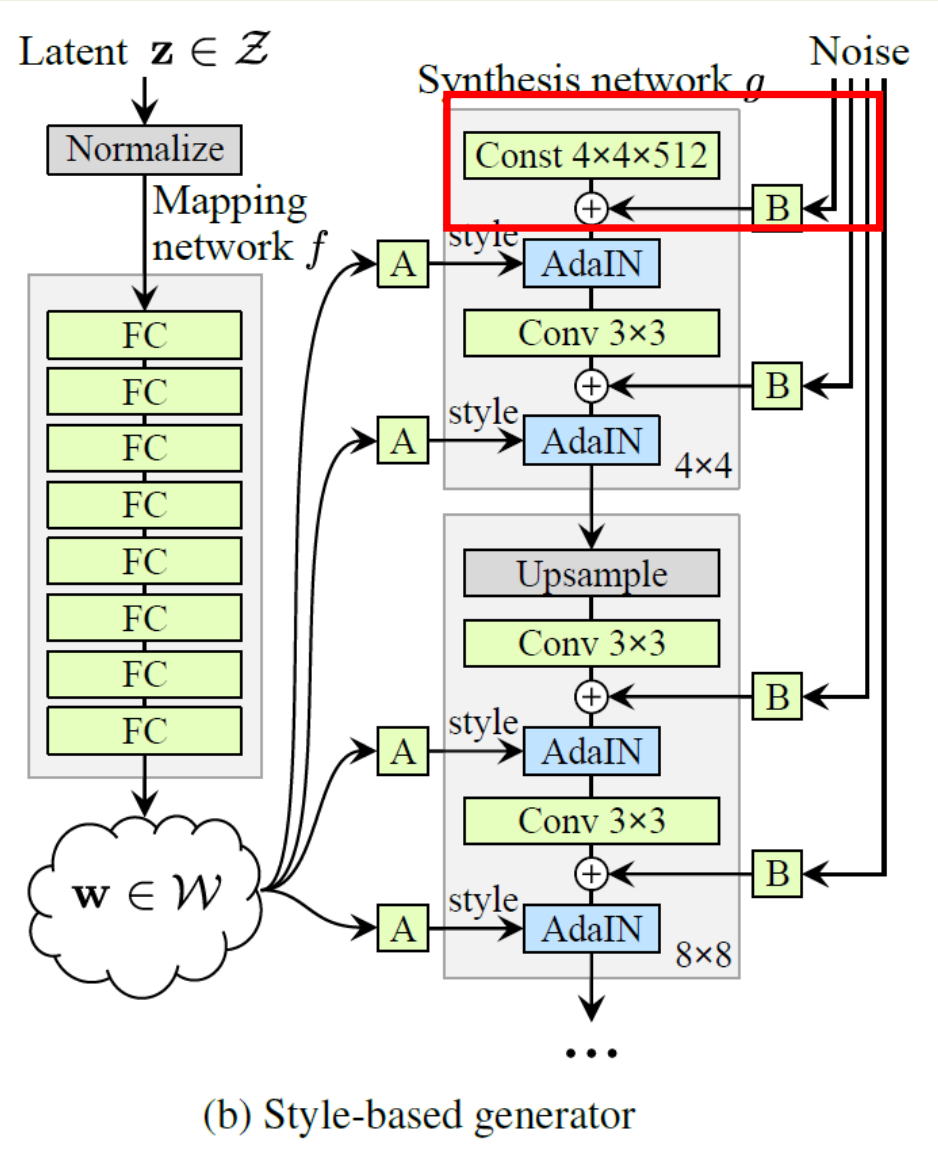
- 特徵糾纏 (entanglement)
  - 特徵之間的耦合太高，改變一個值會影響到多個其他特徵，導致生成圖像往意料之外方向變化

- 若能實現**特徵解耦**，就能實現修改一些值便修改圖像的目的

StyleGAN 對生成器的調整

- 不調整其他部分
  - 如鑑別器、loss function
- 目的在實現特徵解耦





(b) Style-based generator

- 從標準分佈中採樣出 noise
- Noise 經過仿射變換 (乘以可學習參數 B)

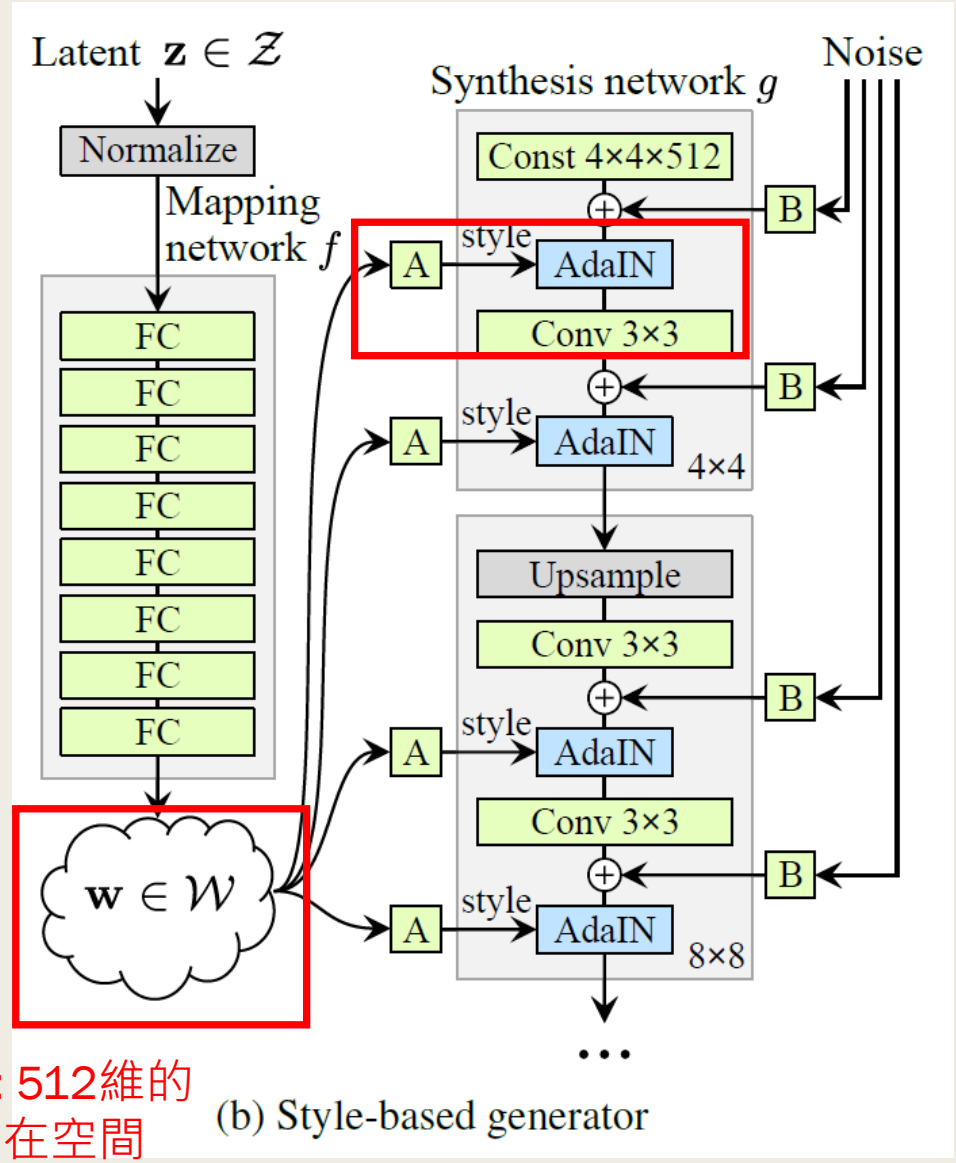
$$B = w * Noise$$

*此處w表示網路參數
不是圖中的w

- 與先前初始化的常數相加

$$x^{new} = x + B$$

Adaptive Instance Normalization (AdaIN) 中的 IN 可能造成水滴狀雜質，因此在 styleGAN2 中移除



W: 512維的潛在空間

- w 經過線性變換(A)，獲得 style

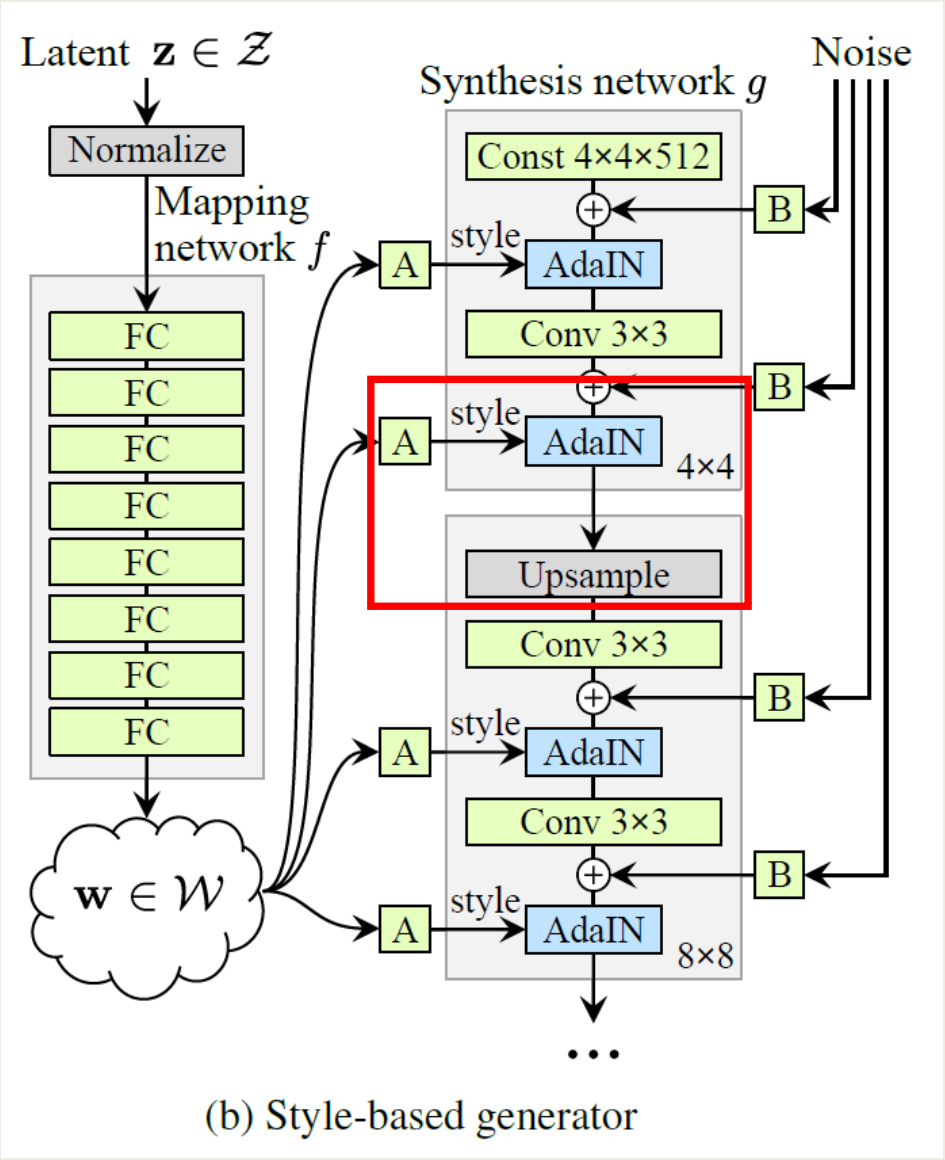
$$A = w^{*}\text{Linear}\ [512\text{->}512^{*}2]$$
$$y_s, y_b = A[:512], A[512:]$$

兩個 y 是來自圖中 w 做線性變化而來

- x^{new} 輸入進 AdaIN 模組
  - 先對 x^{new} 做 instance normalization
  - 再做縮放並加上 bias

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$$

- 再送給 3x3 的卷積層

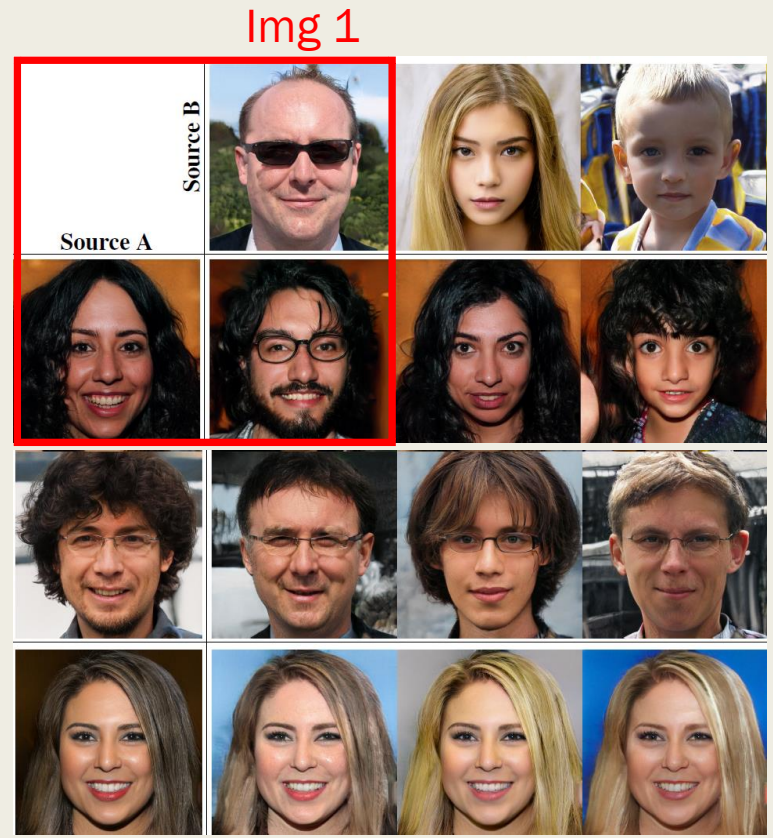


(b) Style-based generator

- 重複前面步驟
- 此處 Upsampling 與前面 PGGAN 用的上採樣做法一樣
- 這種做法真的可以實現特徵解耦?
  - 先從實驗來看結果

特徵解耦– w 的實驗觀察

- 採樣出兩個隨機噪音(latent code $z1$、$z2$)，通過 mapping network 獲得對應的$w1$、$w2$.
- 照前面步驟獲得生成圖像
  - 低解析度時，輸入 w1 (Img1風格) 到生成過程
  - 高解析度時，輸入 w2 (Img2風格) 到生成過程
- 相當於把兩種風格在不同解析度混用
- 觀察
  - 在低解析度注入w能控制性別、臉型
  - 中解析度則控制臉部特徵
  - 高解析度則控制頭髮、背景



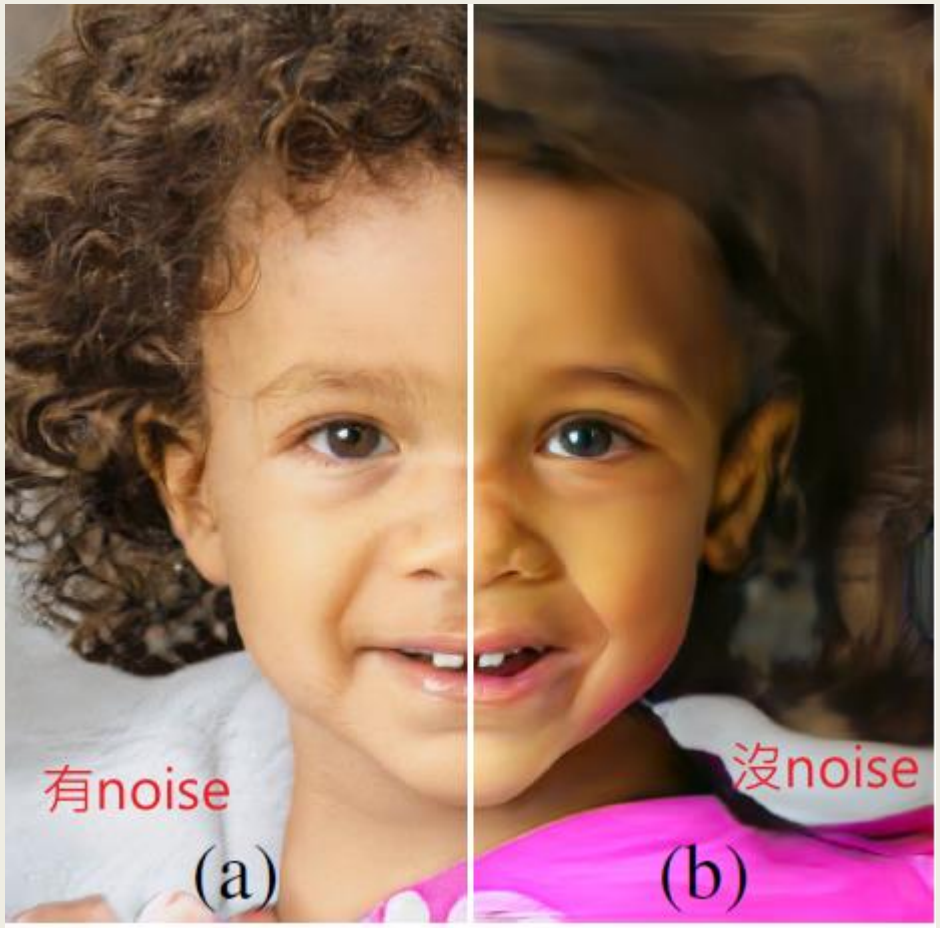
w 風格訊息如何充分解耦?

- 生成器採取 style mixing 方式訓練，強迫生成器解耦風格訊息，讓不同 level 的 style 控制不同東西，例如性別、頭髮、年齡。實現 style 的本地化。
- 訓練步驟
  - 採樣出兩個隨機噪音(latent code $z1$、$z2$)，通過映射獲得對應的$w1$、$w2$.
  - 在合成網路 g 生成圖片時，選一個臨界解析度 crossover，把網路分為低於 crossover 的低解析度部分，以及高解析度部分
  - 低解析度的注入 $w1$；高的注入 $w2$
- 假設現在低、高解析度的特徵尚未充分解耦，style mixing 相當於修改了w的部分值，會導致最後生成圖歪掉，而被鑑別器抓出來，因此生成器便會努力讓特徵解耦。

特徵隨機變化– noise 的實驗觀察

- 過去的 GAN是在一開始輸入層的 activation 中產生偽隨機數，但很難隱藏訊號的週期而導致常出現重複的 pattern。

- styleGAN 則是在每次卷積後，以 pixel 為單位添加 per-pixel noise 來解決這個問題。
  - → 添加雜訊可**優化特徵的隨機變化**，且不影響圖片整體合理性。



訓練中的其他技巧 -- Truncation trick

- 訓練數據中某些類別的數據較少 (低密度區域)，因此較得不到好的訓練，導致低密度區域的圖片生成品質不佳。因此作者希望盡量降低低密度區域圖像生成的可能 (對w空間做限制)，而用 Truncation 來收縮風格空間。
- 步驟
  - 從自己採樣出一批數據，並映射成 w
  - 算出 w 的數學期望，作為這批圖像的平均風格
  - 對於遠離平均的圖像(即低密度數據)，距離就會越遠，縮放程度越大

$$\overline{w} = \mathbf{E}_{z \sim P(z)}[f(z)]$$

收縮

$$w' = \overline{w} + \phi(w - \overline{w})$$

當前圖跟平均圖的距離

小結

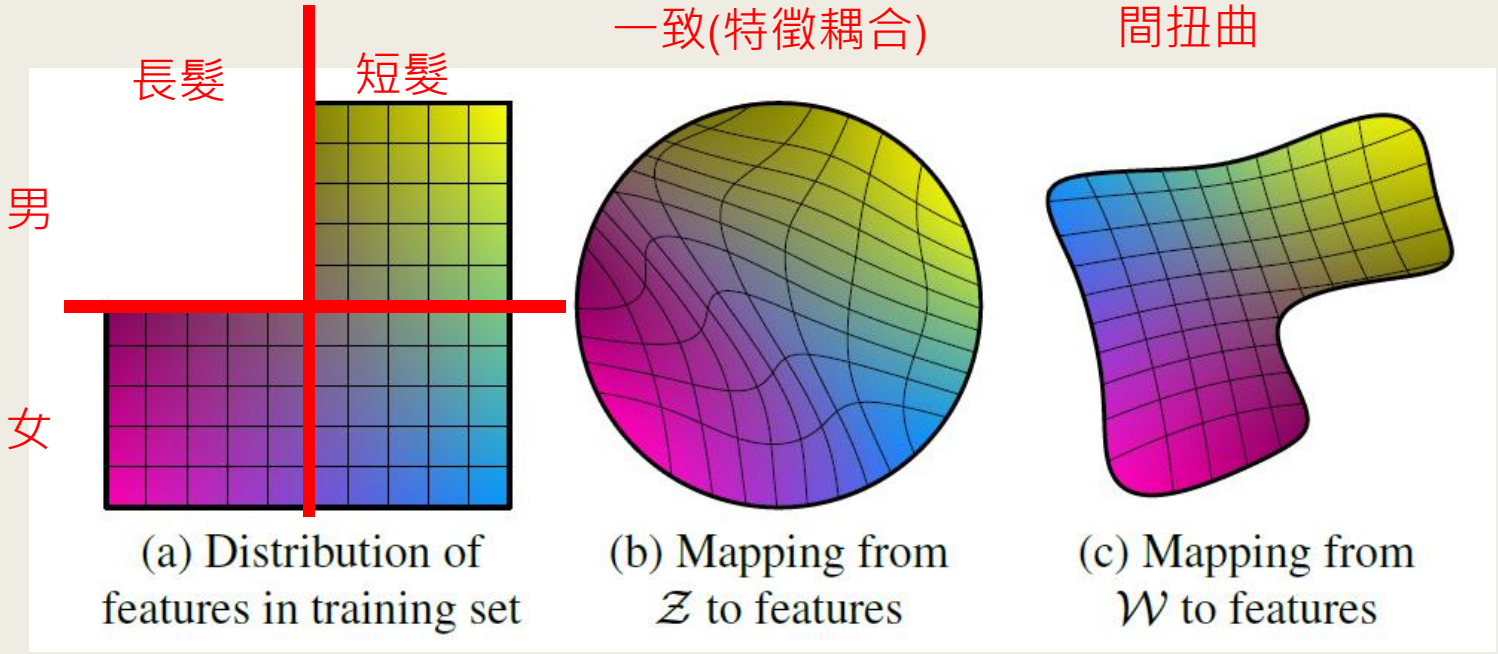
- W 控制圖像的全局風格訊息，例如臉型、頭髮、膚色
- Noise 控制局部訊息，不改變整體風格結構

為何要使用 mapping network?

- 因為 w 空間比 z 空間更符合風格訊息的機率分布！
- 論文例子：缺失長髮男數據

為避免產生長髮男，造成空間不一致(特徵耦合)

在映射過程中可直接消去長髮男組合，避免空間扭曲



然而這裡皆是猜想，需要有實驗數據的量化指標來佐證！

解耦程度量化指標

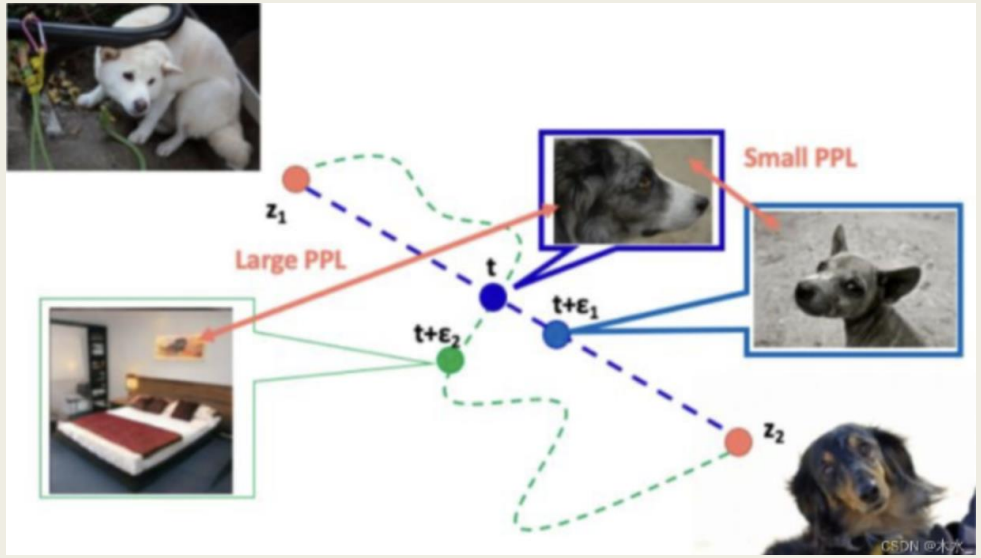
提出兩種量化解纏結的方法，不需要編碼器，任何圖像資料集和生成器都可計算的

- Perceptual path length (PPL)
- Linear separability(separability)

Perceptual path length (PPL)

- 可用於判斷生成器是否選擇最近路線
  - 比較 Z 空間生成圖與 W 空間生成圖，計算解耦程度即可知道誰比較好
  - 能夠以優化 PPL 為目標可以提升 GAN 品質，因此在 styleGAN2 中，PPL 作為正則項加入到損失函數中。

$$l_{\mathcal{Z}} = \mathbb{E}\left[\frac{1}{\epsilon^2} d\big(G(\mathrm{slerp}(\mathbf{z}_1, \mathbf{z}_2;\, t)),\; G(\mathrm{slerp}(\mathbf{z}_1, \mathbf{z}_2;\, t+\epsilon))\big)\right],$$



Linear separability(separability)

- 概念: 如果潛在空間與圖像特徵足夠解耦，則潛在空間中存在線性超平面，可以分類兩種特徵。(可分離性越小越好)

StyleGAN conclusion

- 重點
  - 使用風格遷移中的 AdaIN 模塊
  - 使用映射網路將 z 轉為 w，實現映射關係的解耦
  - 加入隨機噪音, 提升圖像細節並增加多樣性 (優化隨機性)
- 創新點
  - 加入可通用的映射網路
  - style mixing
  - 兩種不需要 encoder 的解耦量化指標 PPL、separability
  - 提出新的人臉資料集（Flickr-Faces-HQ、FFHQ）

pixel2style2pixel (pSp)

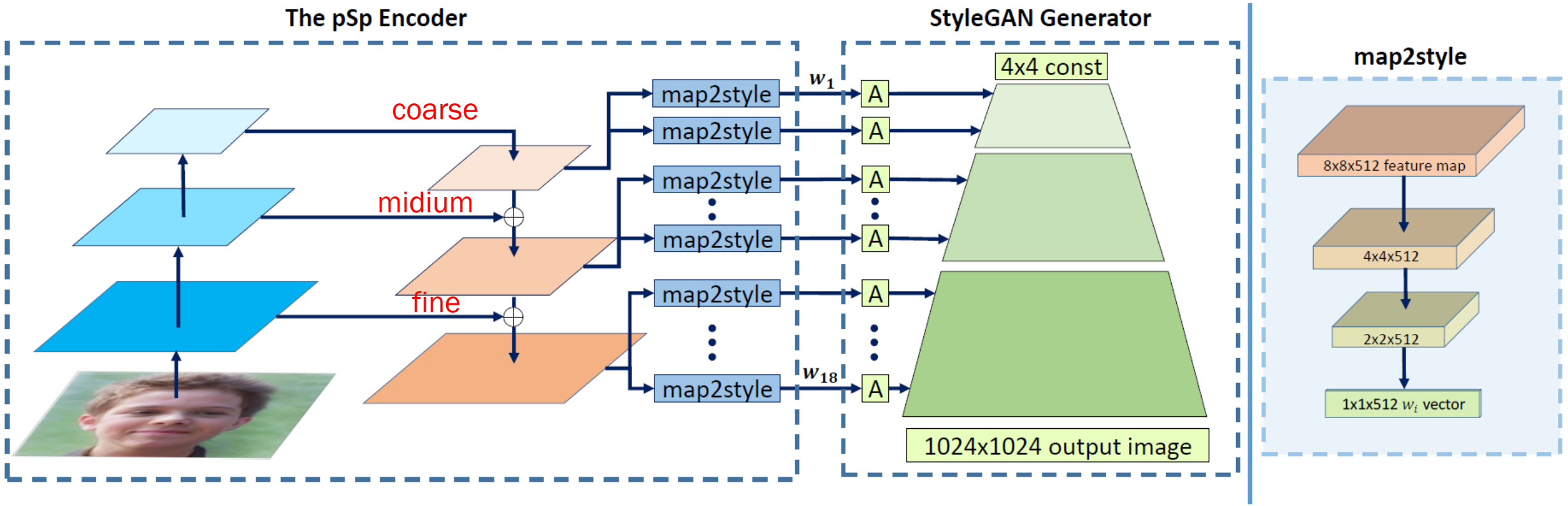
發表於 2021

以輸入圖像作為條件生成特定影像，實現 img2img

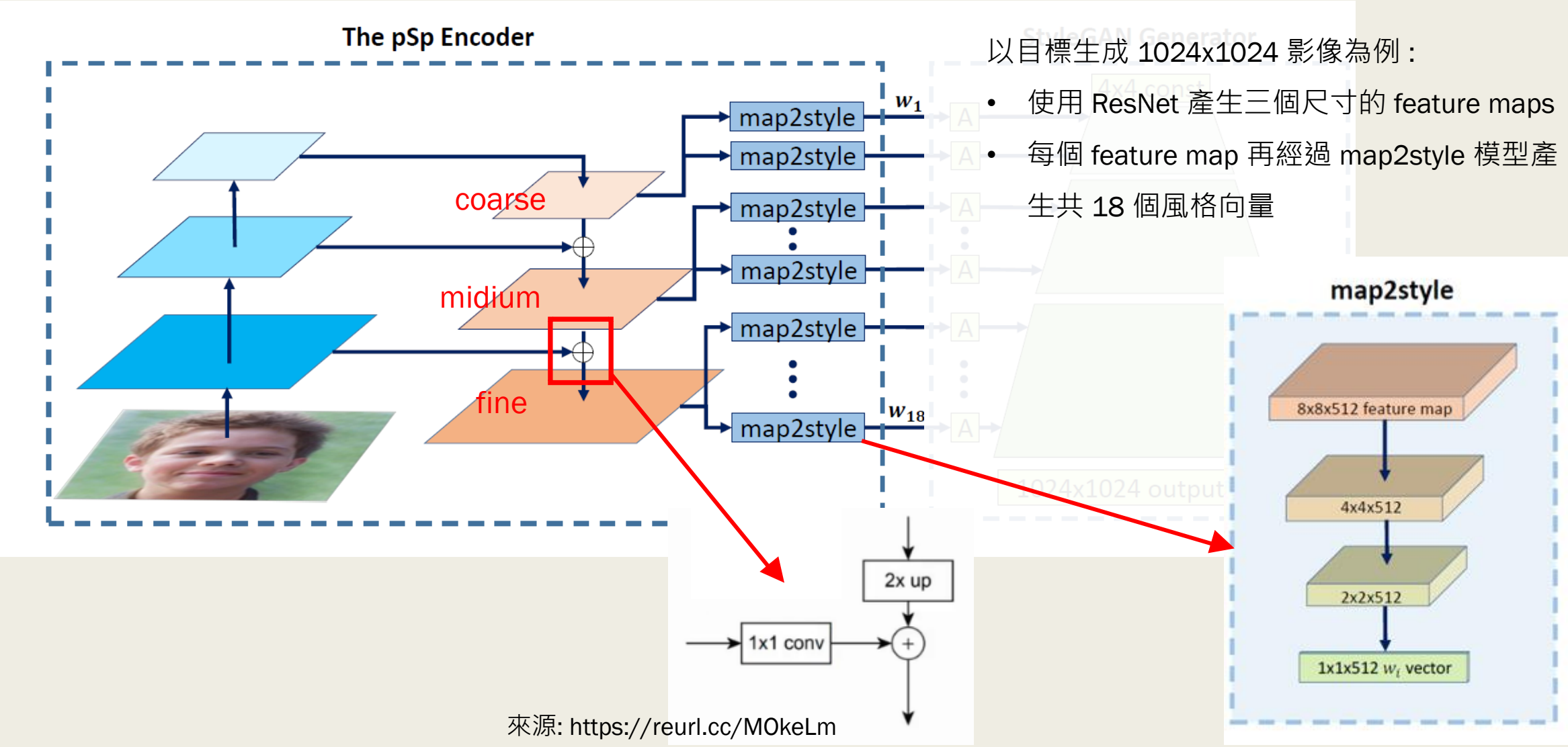
pixel2style2pixel (pSp) 特點

- 將輸入影像透過 encoder 直接轉為 latent code (style)
- 提出通用的影像翻譯(img2img) 框架

Framework


The pSp Encoder
StyleGAN Generator
map2style
coarse
midium
fine
map2style
map2style
map2style
map2style
map2style
w_1
w_{18}
A
A
A
A
A
A
4x4 const
1024x1024 output image
8x8x512 feature map
4x4x512
2x2x512
1x1x512 w_i vector

Encoder – Feature Pyramid Network(FPN)



以目標生成 1024x1024 影像為例：

- 使用 ResNet 產生三個尺寸的 feature maps
- 每個 feature map 再經過 map2style 模型產生共 18 個風格向量

來源: https://reurl.cc/MOkeLm

Style mixing

- 在 coarse 跟 medium 層面上使用圖像經 encode 輸出的 style，fine 才使用隨機變量，以保證產生的是同一個人，但可以有不同的細節變化。

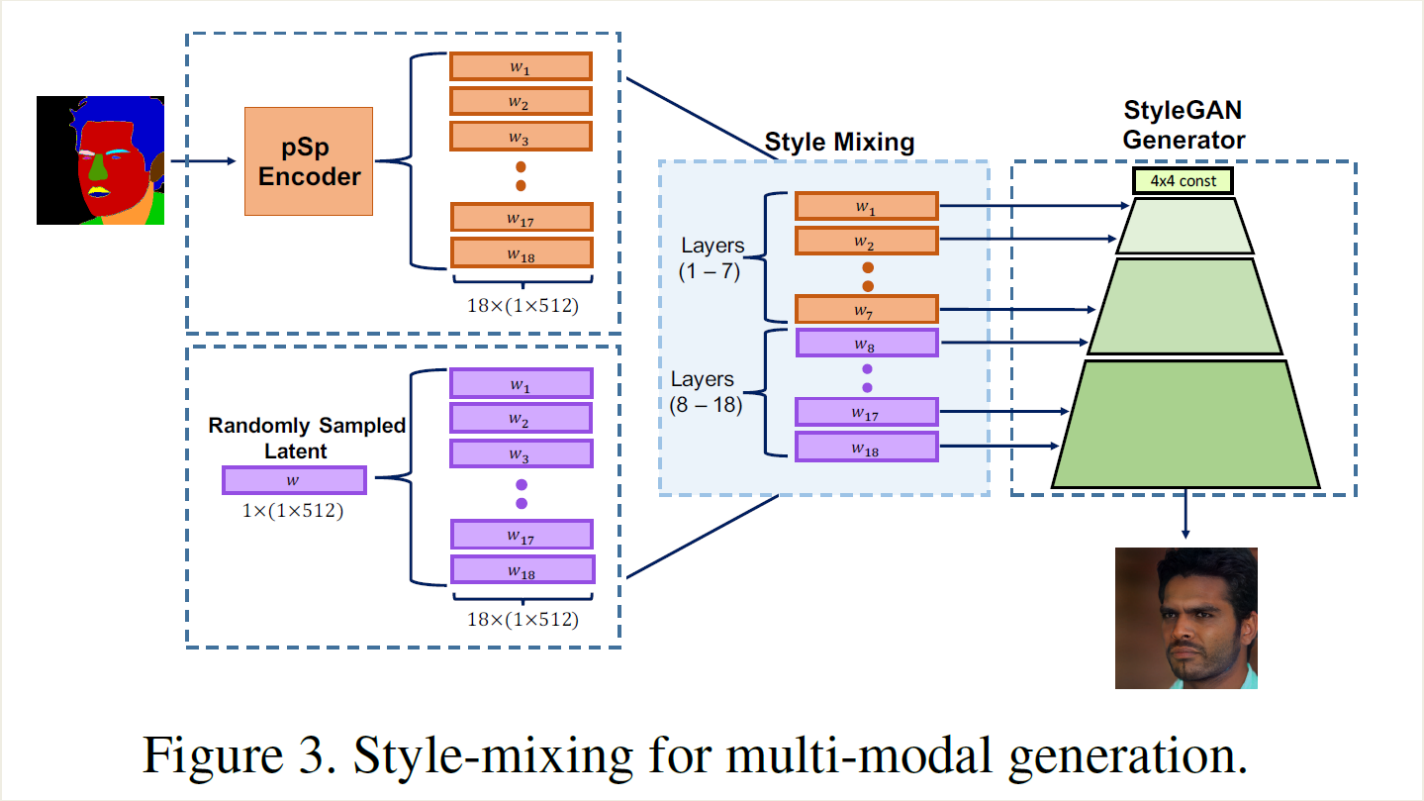


Figure 3. Style-mixing for multi-modal generation.

Loss Functions

encoder的訓練是透過多個 loss 的加權組合而成的。

$$\mathcal{L}_2(x) = \|x - pSp(x)\|_2$$

- pixel-wise L2 loss

$$\mathcal{L}_{PIPS}(x) = \|F(x) - F(pSp(x))\|_2$$

- 為了學習感知相似性，利用 LPIPS 損失(已被證明可以更好地保持影像品質)

$$\mathcal{L}_{reg}(x) = \|E(x) - \bar{w}\|_2$$

- 鼓勵編碼器輸出更接近平均latent vector的latent styke vector，另外定義的正規化損失
- 與 StyleGAN 中的截斷技巧類似，作者發現在encoder的訓練中添加這種正則化可以提高圖像品質，而不損害輸出的保真度，特別是在更模糊的任務中

$$\mathcal{L}_{ID}(x) = 1 - < R(x),\ R(pSp(x)) >$$

- 處理臉部影像編碼的特定任務時的一個常見挑戰是保存輸入身分，因此採用專用的辨識損失來測量輸出影像與其來源影像之間的餘弦相似度
- R: 預訓練的 ArcFace 模型

$$\mathcal{L}(x) = \lambda_1\mathcal{L}_2(x) + \lambda_2\mathcal{L}_{PIPS}(x) + \lambda_3\mathcal{L}_{ID}(x) + \lambda_4\mathcal{L}_{reg}(x)$$

- 最終結果。其中λ_1~λ_4是定義損失權重的常數。

pSp 應用

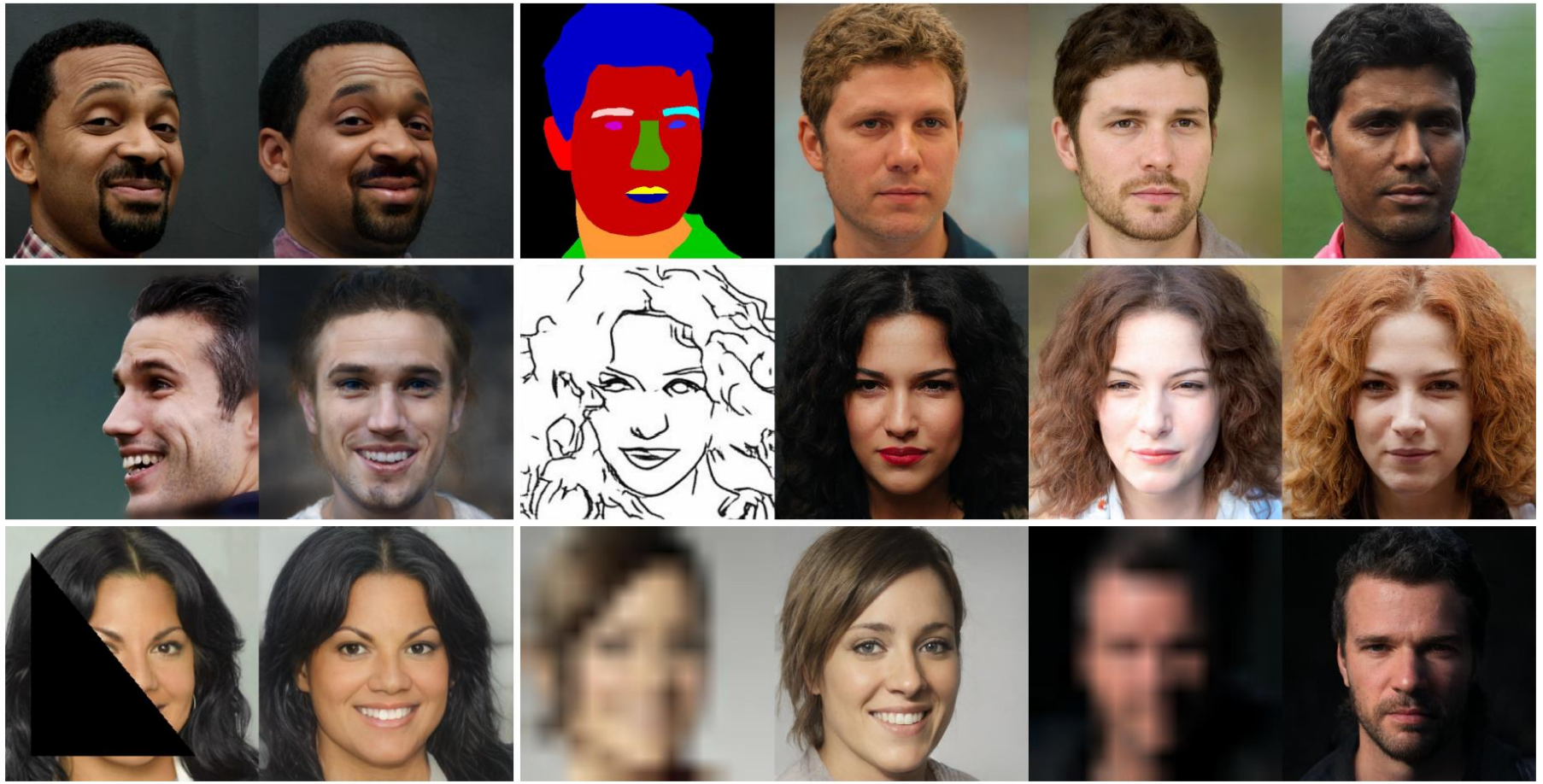
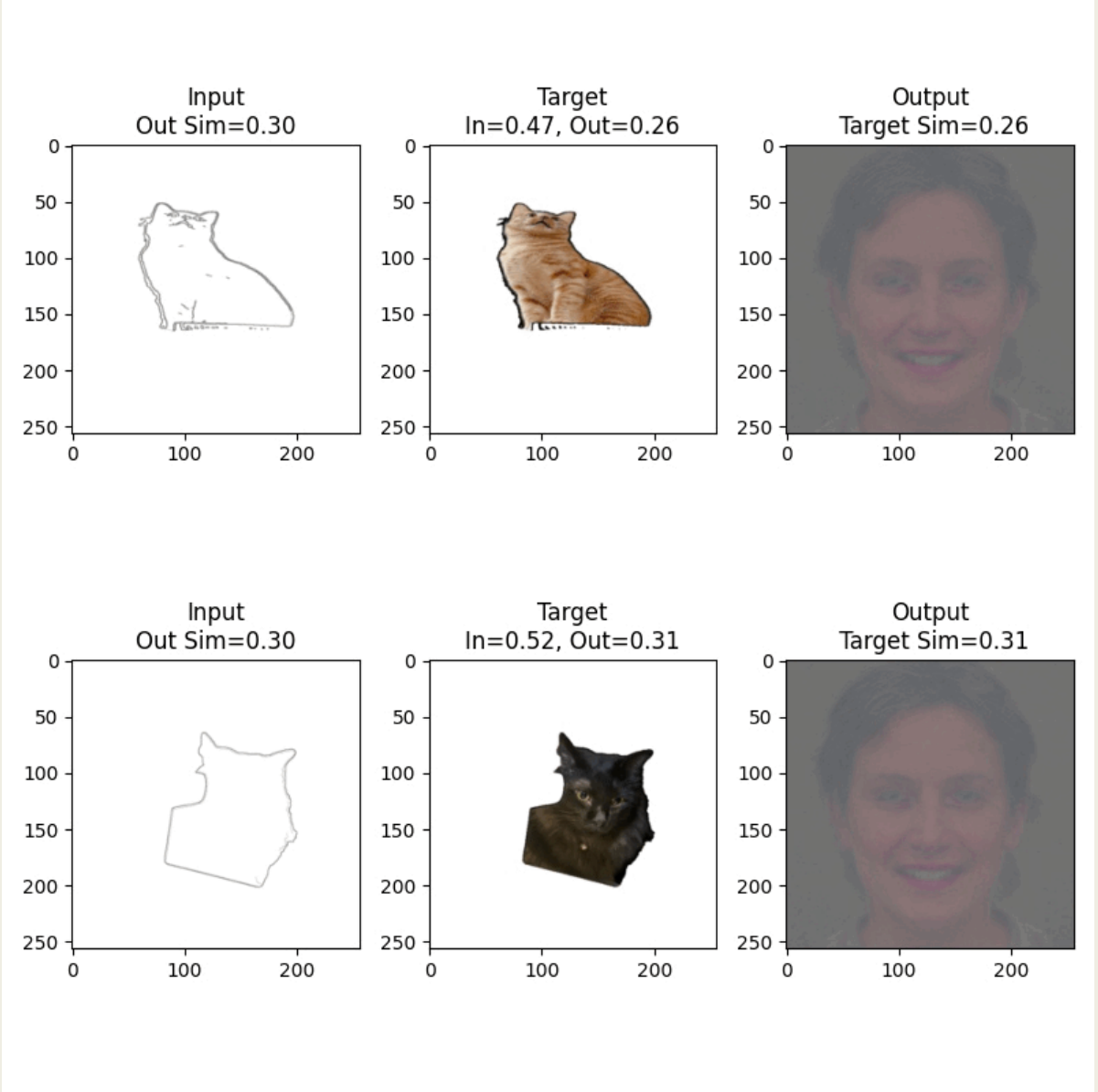


Figure 1. The proposed pixel2style2pixel framework can be used to solve a wide variety of image-to-image translation tasks. Here we show results of pSp on StyleGAN inversion, multi-modal conditional image synthesis, facial frontalization, inpainting and super-resolution.

實測

Epoch 0~147400



參考

- 伪造高清人像——PGGAN原理解析
- 图像风格混合——StyleGAN原理解析
- pixel2style2pixel（pSp）实现解读【一】

報告結束~