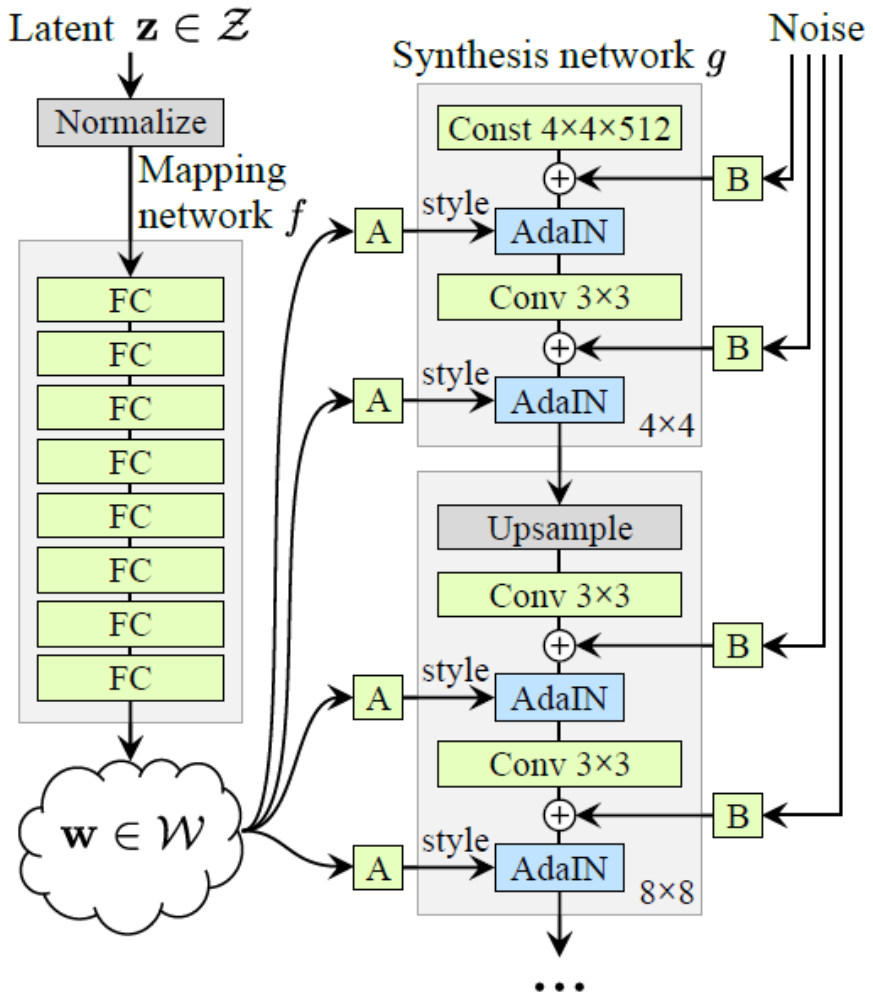
[Kim21-CVPR] Exploiting Spatial Dimensions of Latent in GAN for Real-time Image Editing

Jiin Kim

Abstract



StyleGAN

Abstract

- Latent vectors 로 이미지를 제어할 수 있다. 그러나, GAN 을 사용해 실제 이미지를 편집하는 것은 두 가지 어려움이 있다.

    1. 실제 이미지를 latent vectors 로 만들 때 **시간**이 걸린다.
    2. Encoder 를 통과시켜 만든 **embedding 이 부정확**할 수 있다.

Abstract

- **StyleMapGAN** 을 제안한다.
  - 중간 latent space 가 spatial dimensions 를 가진다.
  - 공간적으로 변형이 가능한 모듈(Spatially variant modulation)이 AdaIN 을 대체한다.
- StyleMapGAN 은 GANs 의 특성을 유지한 채로 **embedding 을 더 정확하게** 만든다.

1. Introduction

Introduction

- GAN 이 image 를 latent code 로 mapping 을 잘 못하기 때문에 real images 를 조정하는 것은 어렵다.
- Real images 를 조정하는 다양한 연구 중에서, 실용적인 접근법은 **추가적인 encoder** 를 학습하는 것이다. 추가적인 encoder 는 이미지를 그에 상응하는 latent code 로 만든다.

Introduction

- 그러나 이 방법에도 **detail 이 없는 이미지**를 만든다는 단점이 존재한다.
- 본 논문은 이런 단점이 latent space 에서 **spatial dimensions** 이 없기 때문이라고 생각했다.
- Spatial dimensions 이 없는 전통적인 Encoder 는 이미지의 local semantics 를 vector 로 압축하고, 이미지를 reconstruct 하기 어렵게 만든다.

Introduction

- 본 논문은 문제의 해결책으로 **StyleMapGAN** 을 제안한다.
  - Vector-based latent representation 을 학습하는 대신에, 명시적인 spatial dimensions 를 가진 tensor 를 사용한다. Spatial dimensions 를 사용해서, GANs 이 이미지의 **local semantics 를 latent space 로** 쉽게 encode 할 수 있도록 한다.
  - 본 논문은 또한 stylemap 의 matching 위치를 조정해서 이미지의 **특정 부분**을 편집할 수 있게 한다.

Introduction

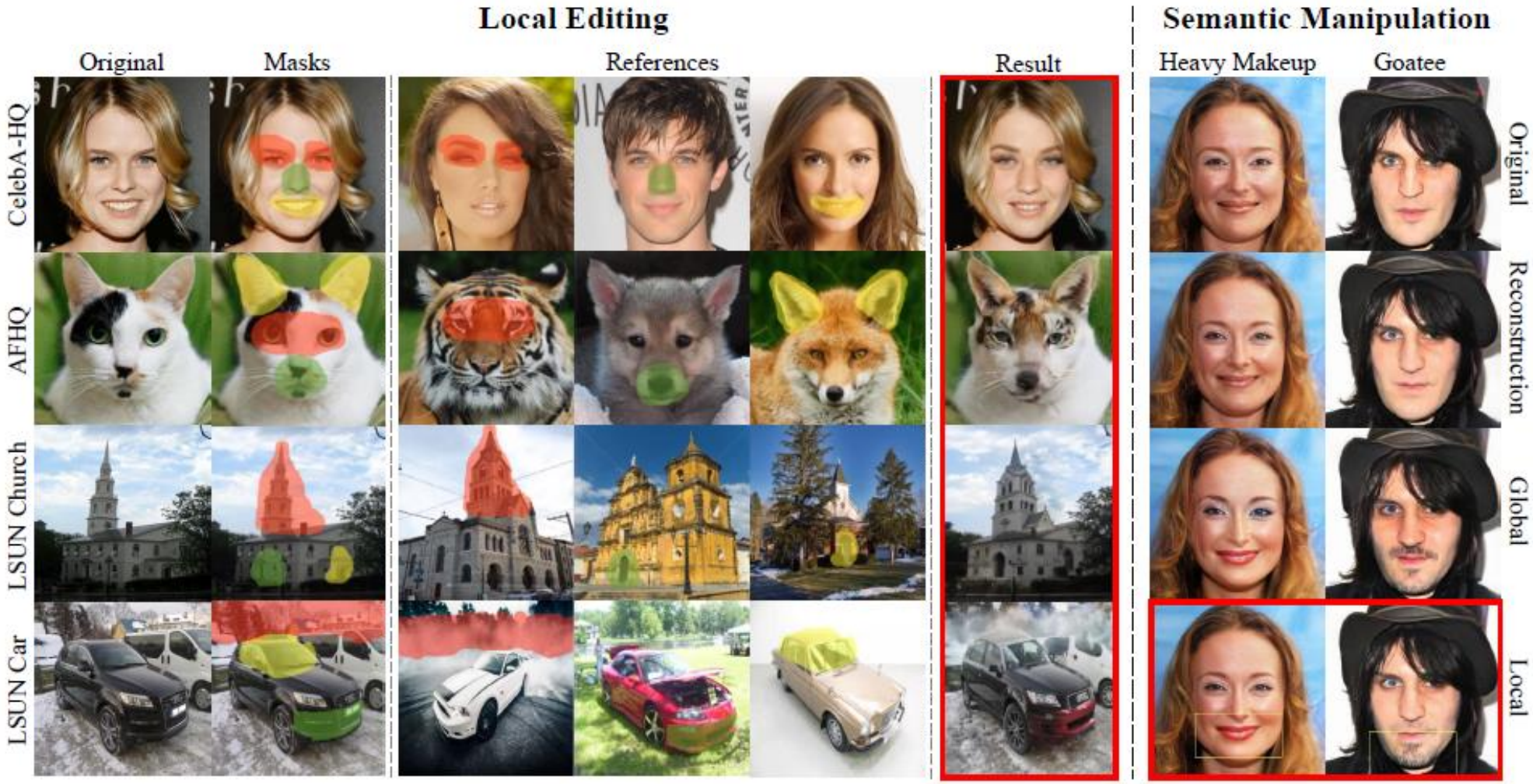
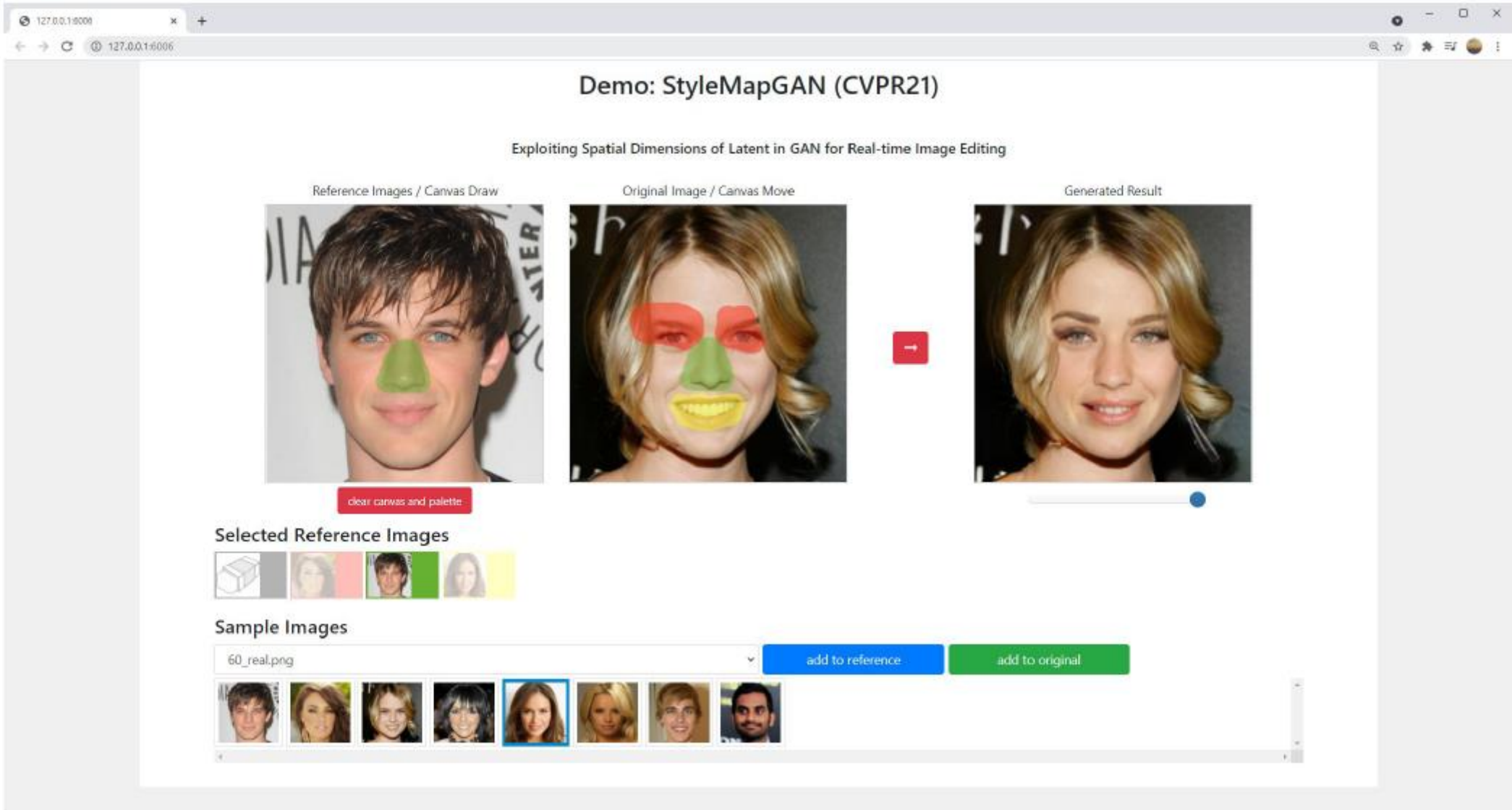


Figure 1: Various image editing results on multiple datasets. Local editing mixes multiple parts of reference images with the original image. Unlike other methods (*Global* case), ours can do semantic manipulation locally (Yellow box, *Local* case).

Introduction



2. Related work

Related work

- GAN 을 사용한 editing 방법은 세 가지로 정리할 수 있다.
  - Optimization-based editing methods
  - Learning-based editing methods
  - Local editing methods

3. StyleMapGAN

StyleMapGAN

- 본 논문은 spatial dimensions 를 가진 stylemap 을 사용하는 **StyleMapGAN** 을 제안한다.
- 본 논문의 **목표**
  - 실시간으로 encoder 를 통해 이미지를 latent space 로 정확하게 투영하기

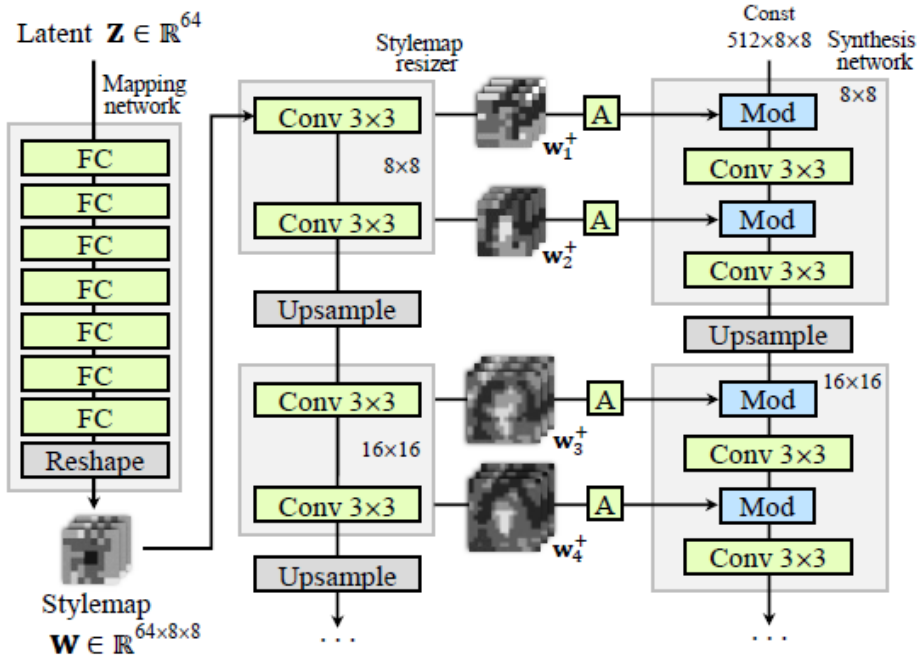
  → Encoder 는 이미지를 stylemap 으로 embed 하고, 이 방법은 optimization-based methods 보다 더 정확하게 이미지를 reconstruct 한다.

  - Latent space 에서 이미지를 지역적으로 조정하기

  → stylemap 에서 부분적인 변화는 이미지에서 local editing 을 이끈다.

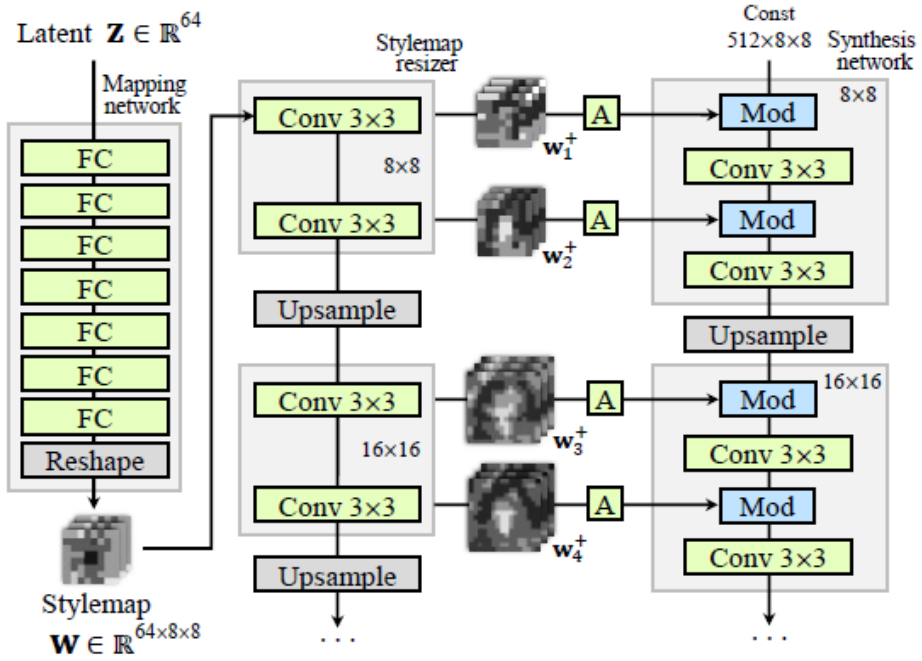
3.1. Stylemap-based generator

Stylemap-based generator



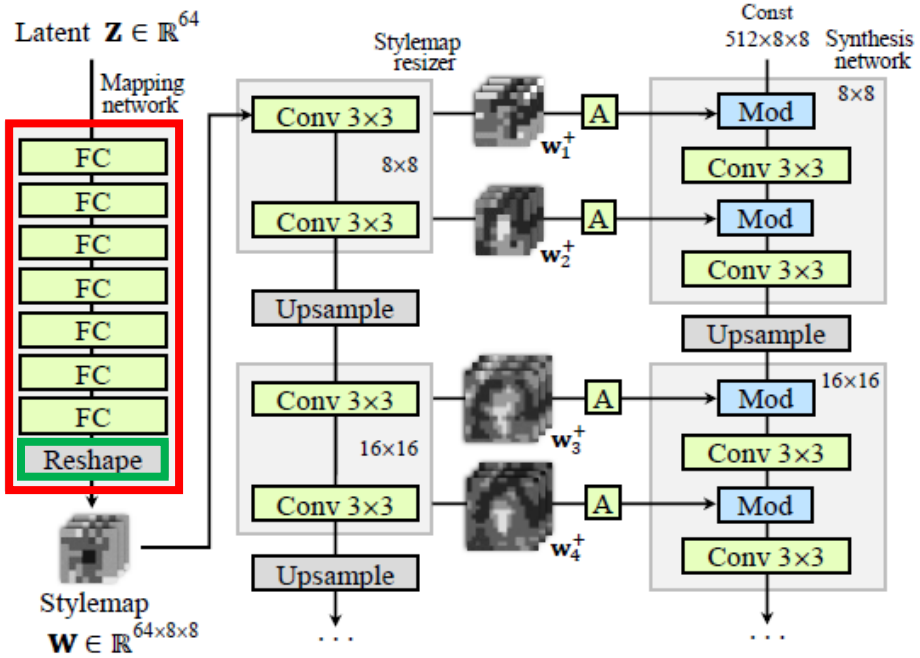
- StyleMapGAN Generator
  - 합성 네트워크에서 각 feature 의 spatial resolution 에 맞도록 stylemap w 는 합성곱층을 통해 w^+ 로 크기가 조정된다.
  - "A" 는 학습된 affine transform 이다. Spatial modulation parameters (γ 와 β)를 제공한다.
  - "Mod" 는 element-wise multiplication & addition 으로 구성된 modulation 이다.
  - 합성 네트워크의 입력으로 학습된 **상수 텐서**가 들어간다.
  - 결과 이미지의 style 은 resized stylemaps 에 의해 조정된다.

Stylemap-based generator



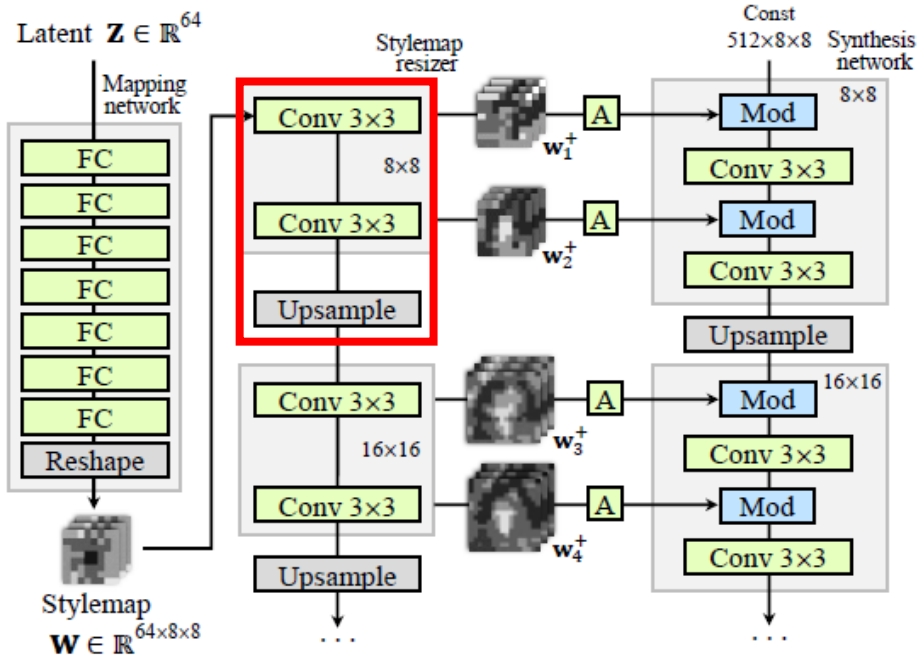
- 본 논문은 stylemap 을 **spatial dimensions** 를 사용해서 만든다.
- 이것은 추론 단계에서 (1) **real image 의 projection** 을 더 효과적으로 할 뿐만 아니라 (2) **local editing** 도 가능하게 한다.

Stylemap-based generator



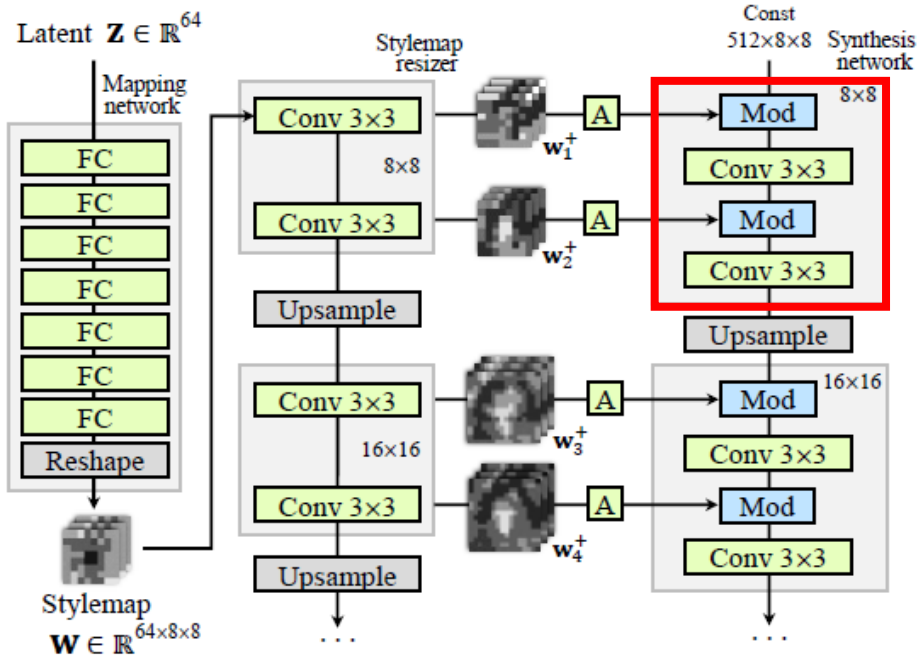
- **Mapping network** 는 공간적으로 다양한 affine parameters 의 입력을 만들기 위해 네트워크 끝에 reshape layer 를 가진다.

Stylemap-based generator



- **Synthesis network** 에서 feature maps 는 output image 에 가까워질 수록 크기가 커진다.
- 그래서 stylemaps 의 해상도를 feature maps 와 맞출 수 있도록 **Stylemap resizer** 를 도입했다.
- Stylemap resizer 는 더 구체적이고 구조화된 styles 를 전달하기 위해 학습된 convolutions 와 함께 stylemap 을 resize 하고 transform 한다.

Stylemap-based generator

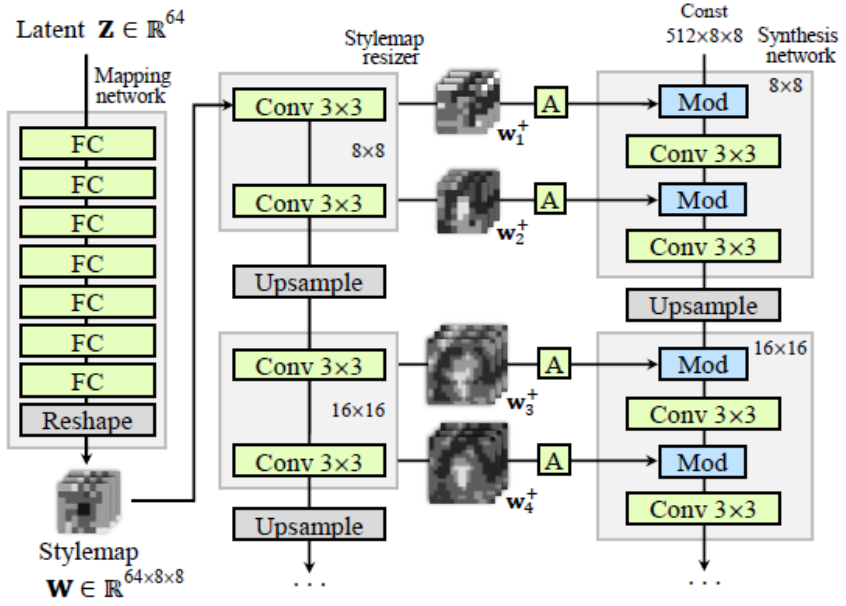


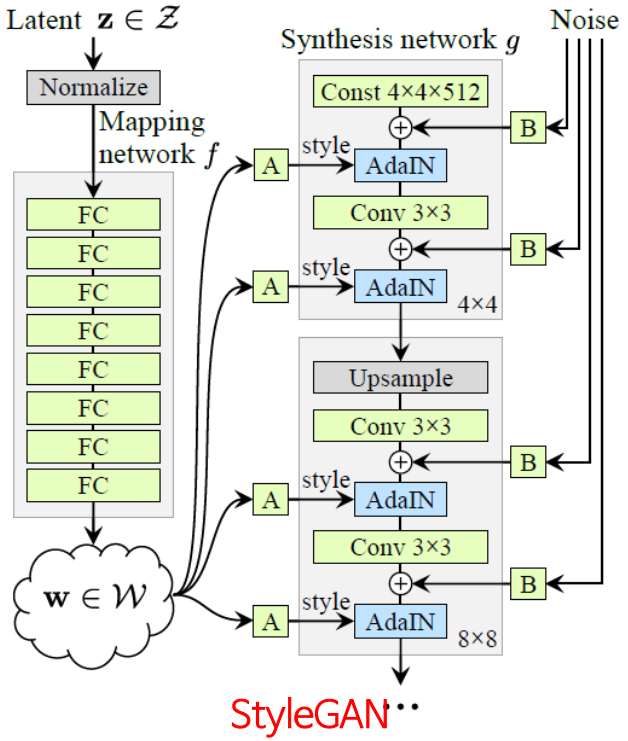
- Modulation 연산을 위해 학습된 affine transform 이 파라미터 (γ, β)를 만든다.
- Synthesis network 의 i-th 레이어에서 modulation 연산은 다음과 같다.

$$h_{i+1} = \left(\gamma_i \otimes \frac{h_i - \mu_i}{\sigma_i}\right) \oplus \beta_i \qquad (1)$$

- μ : mean, σ : variation
- γ, β : modulation parameters

Stylemap-based generator

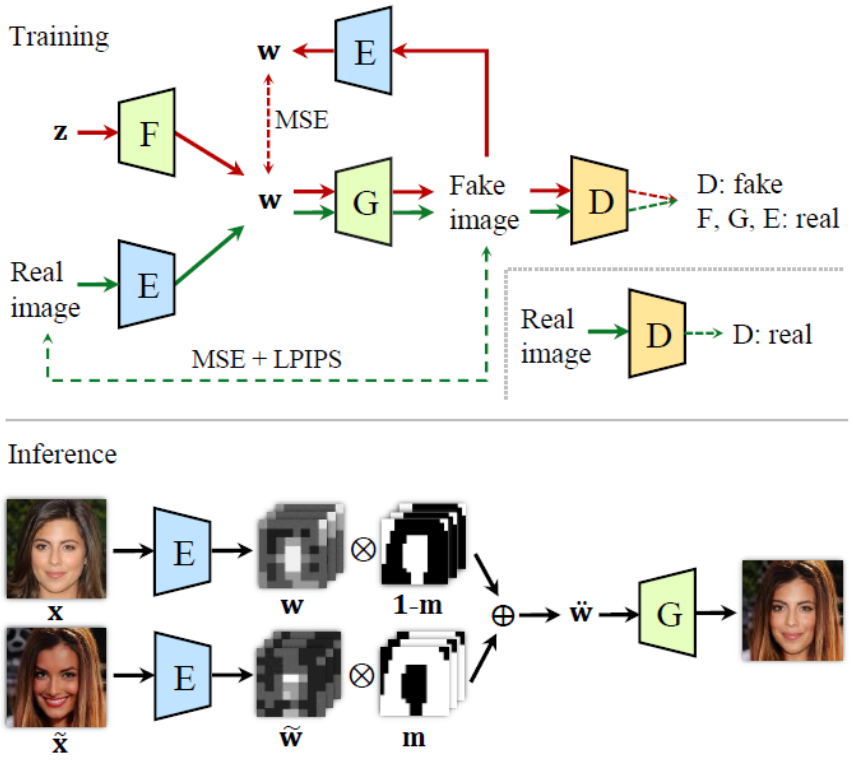




- 본 논문은 StyleGAN 에서 공간적으로 다양한 입력을 만드는 데에 사용되는 **per-pixel noise 를 제거**했다.
- 본 논문의 stylemap 은 이미 공간적으로 다양한 입력 제공하고, 하나의 입력(Const)만으로 projection 과 editing 을 더 간단하게 만들기 때문이다.

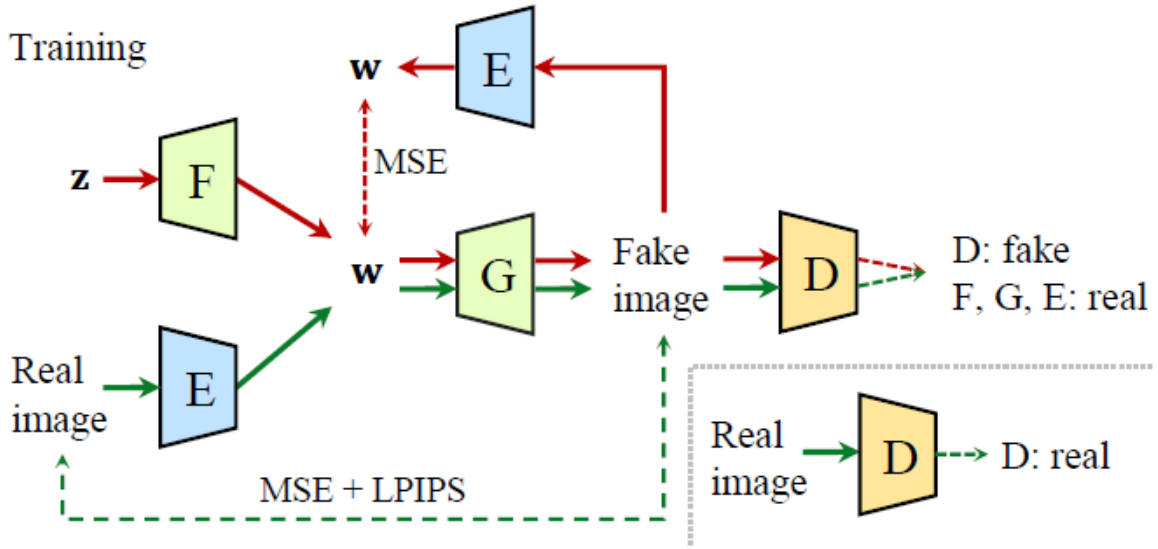
3.2. Training procedure and losses

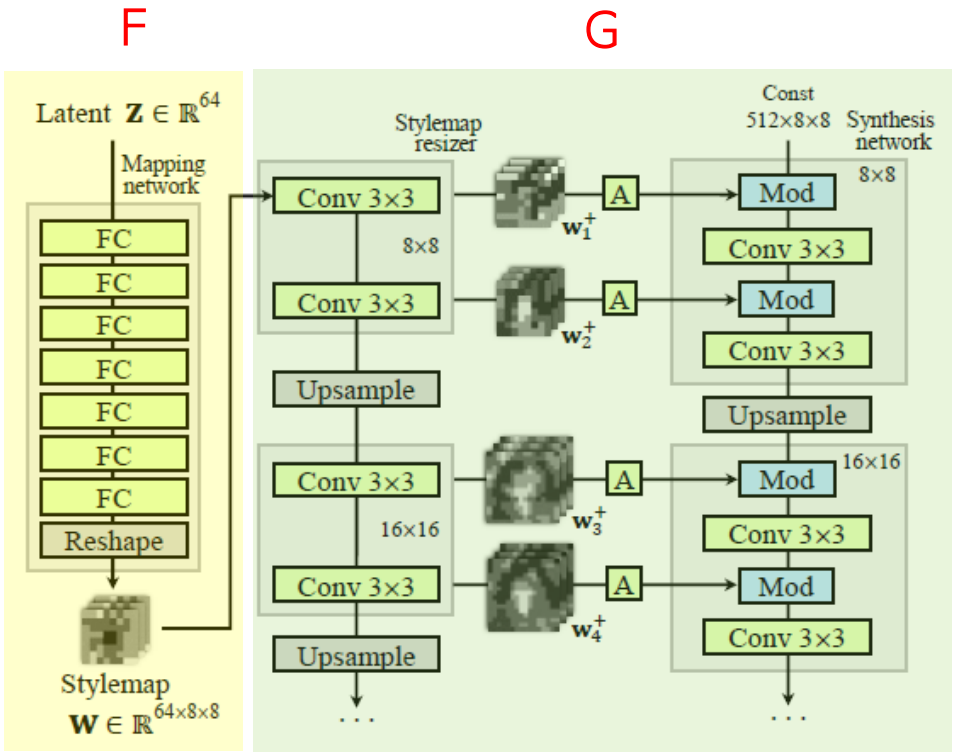
Training procedure and losses



- 첫 번째 그림은 전반적인 training 과정을 나타낸다. 초록색, 빨간색 화살표는 과정의 흐름을 나타내고, 점선은 손실 함수를 나타낸다.
- 두 번째 그림은 stylemap 에서 local editing 하는 방법을 나타낸다.

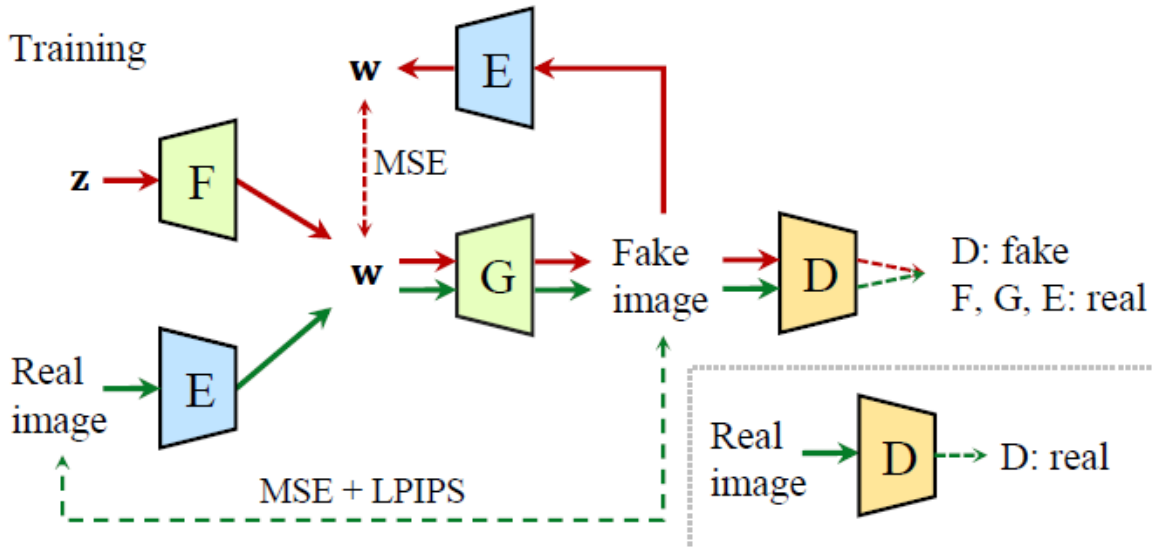
Training procedure and losses





- F : mapping network
- G : synthesis network with stylemap resizer
- E : encdoer (minibatch discrimination 이 없는 것만 제외하고 D와 유사하다)
- D : discriminator (StyleGAN2 와 동일하다)

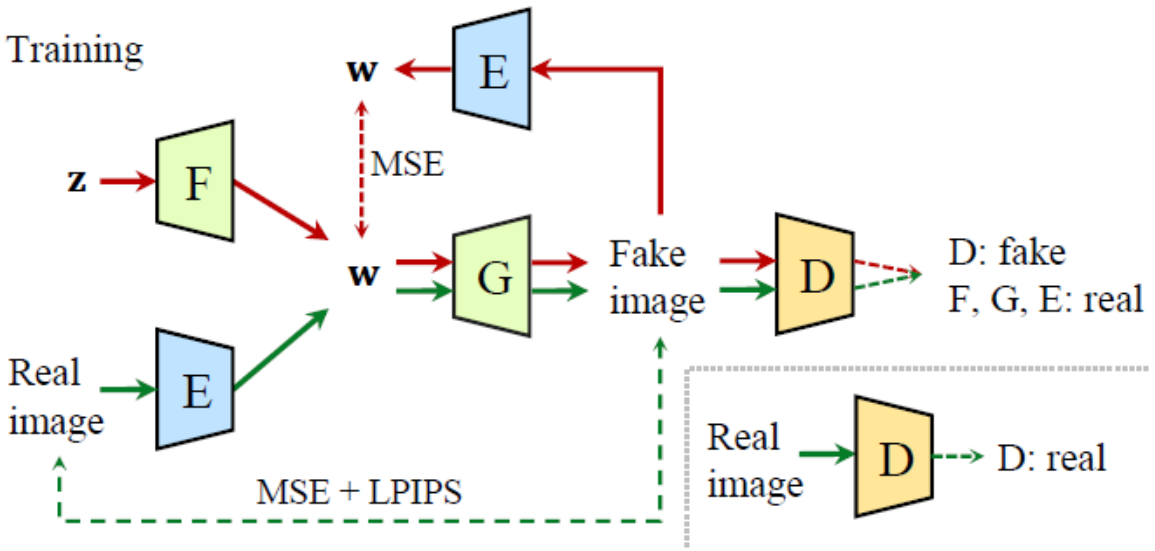
Training procedure and losses



Loss	G	D	E
Adversarial loss [16]	✓	✓	
R_1 regularization [40]		✓	
Latent reconstruction			✓
Image reconstruction	✓		✓
Perceptual loss [61]	✓		✓
Domain-guided loss [62]	✓	✓	✓

- 모든 네트워크들은 여러가지 loss 를 사용해서 함께 학습된다.
- G 와 E 는 pixel-level 과 perceptual-level 의 관점에서 real images 를 reconstruct 하기 위해 학습된다.
- G(F) 가 z 로부터 이미지를 합성할 때, E 도 MSE 를 사용해서 stylemap 을 reconstruct 하는 것을 시도한다.

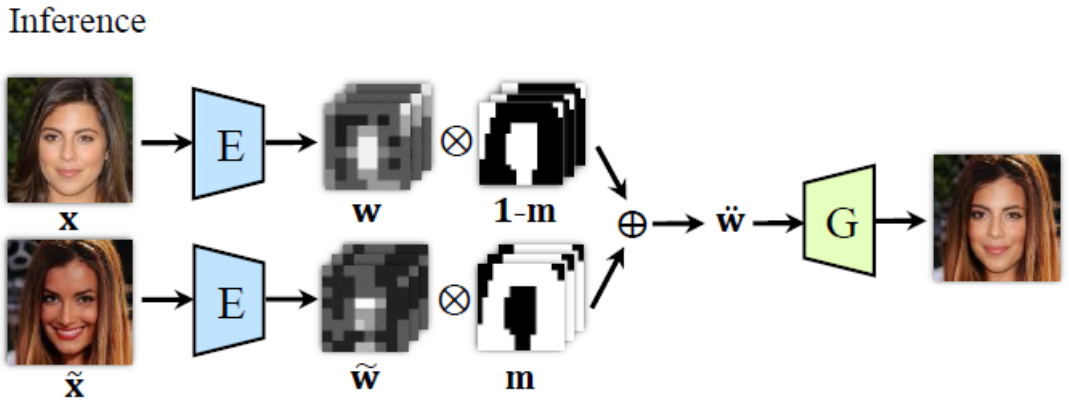
Training procedure and losses



- D 는 real images 와 가우시안 분포에서 생성된 fake images 를 분류한다.
- E 는 D 와 경쟁을 해서 더 현실적인 이미지를 reconstruct 하기 위해 노력하고, 이미지 editing 을 위해 projected stylemap 을 더 안정적으로 만든다.

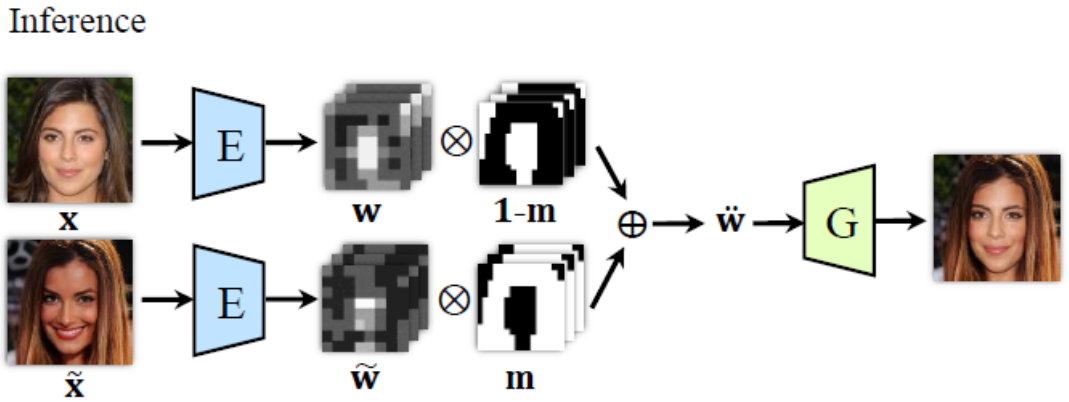
3.3. Local editing

Local editing



- **Local editing** 의 목표는 mask 를 사용해서 reference image 의 일부를 original image 로 이식하는 것이다.
- **Mask** 는 semantic segmentation 을 사용해서 interactive editing 또는 label-based editing 을 할 수 있다.

Local editing



- 본 논문은 stylemaps w 와 $\tilde{w}$ 를 만들기 위해 original image 와 reference image 를 encoder 를 통과시켜 project 한다.
- $\ddot{w}$ 는 다음의 식을 통해 만들어진다. $$\ddot{\mathbf{w}} = \mathbf{m} \otimes \tilde{\mathbf{w}} \oplus (1 - \mathbf{m}) \otimes \mathbf{w} \qquad (2)$$
- 세부적으로 조정하기 위해 w^+ space 에서 stylemaps 를 blend 하지만, simplicity 를 위해 w space 에서 blending 을 설명한다.

4. Experiments

Experiments

Method	Style resolution	Runtime (s)	CelebA-HQ MSE	CelebA-HQ LPIPS	CelebA-HQ FID	AFHQ MSE	AFHQ LPIPS	AFHQ FID
StyleGAN2	1×1	0.030	0.089	0.428	4.97	0.139	0.539	8.59
StyleMapGAN	4×4	0.085	0.062	0.351	**4.03**	0.070	0.394	14.82
StyleMapGAN	8×8	0.082	0.023	0.237	4.72	0.037	0.304	11.10
StyleMapGAN	16×16	0.078	0.010	0.146	4.71	0.016	0.183	**6.71**
StyleMapGAN	32×32	0.074	**0.004**	**0.076**	7.18	**0.006**	**0.090**	7.87

Table 2: Comparison of reconstruction and generation quality across different resolutions of the stylemap. The higher resolution helps accurate reconstruction, validating the effectiveness of stylemap. We observe that 8×8 stylemap already provides accurate enough reconstruction and accuracy gain, and afterward, improvements get visually negligible. Although FID varies differently across datasets, possibly due to the different contextual relationships between locations for generation, the stylemap does not seriously harm the images' quality; rather, it is even better in some configurations. Using our encoder and generator, total inference time is less than 0.1s with almost perfectly reconstructed images. Although StyleGAN2 with our encoder is faster than StyleMapGAN, but it suffers from poorly reconstructed images (second column).

Experiments

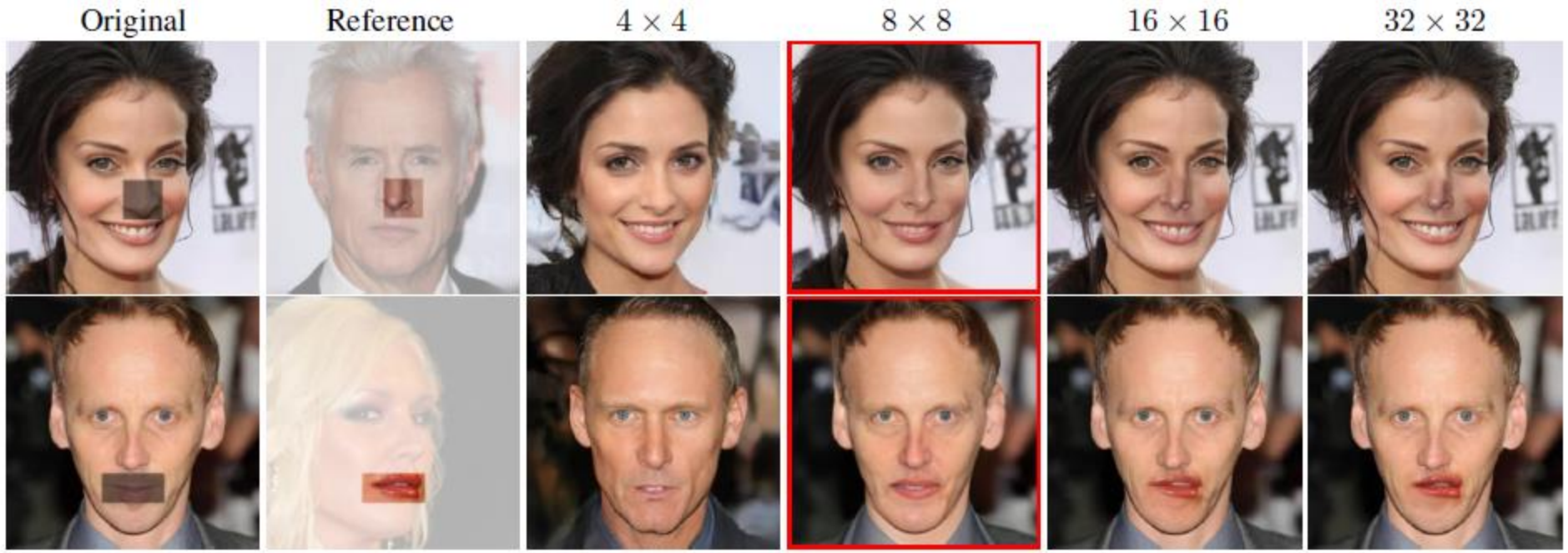
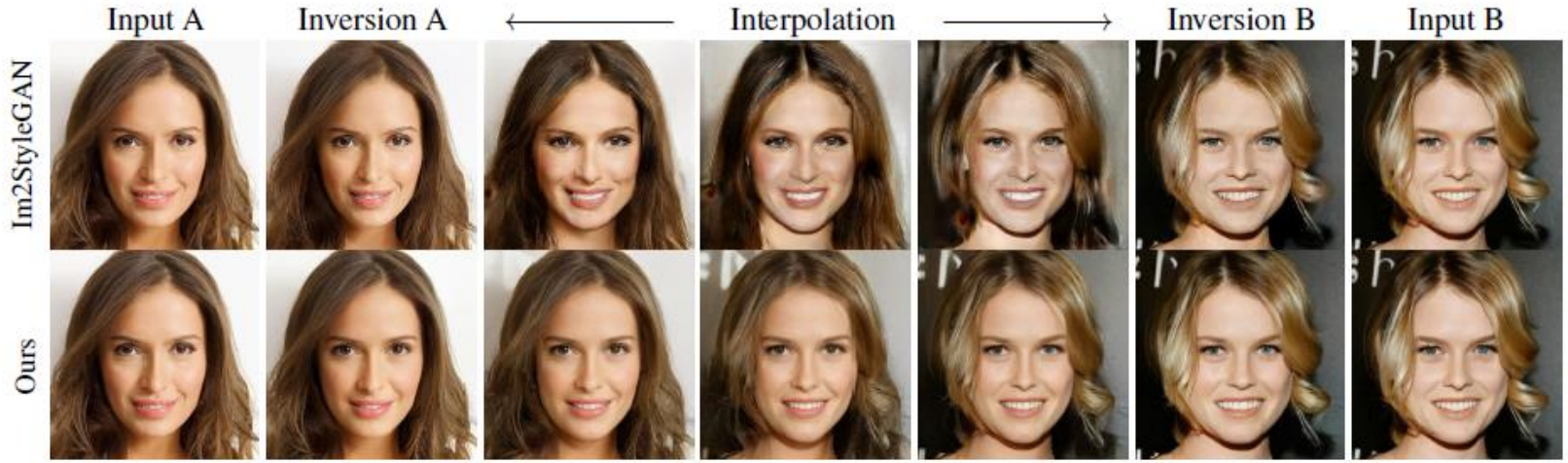


Figure 5: Local editing comparison across different resolutions of the stylemap. Regions to be discarded are faded on the original and the reference images. 4×4 suffers from the poor reconstruction. Resolutions greater than or equal to 16×16 result in too heterogeneous images. 8×8 resolution shows acceptable reconstruction and natural integration. Note that our method works well even in the case that the mask locates improperly as shown in the reference image of the first row.

Experiments



Method	Runtime (s)	CelebA-HQ MSE	CelebA-HQ LPIPS	CelebA-HQ FID_{lerp}	AFHQ MSE	AFHQ LPIPS	AFHQ FID_{lerp}
StyleGAN2 [26]	80.4	0.079	0.247	30.30	0.091	0.288	**13.87**
Image2StyleGAN [1]	192.5	**0.009**	**0.203**	23.68	**0.018**	**0.282**	40.80
Structured Noise [3]	64.4	0.097	0.256	27.96	0.144	0.332	34.99
In-DomainGAN [62]	6.8	0.052	0.340	**22.05**	0.077	0.414	17.54
SEAN [65]	0.146	0.064	0.334	30.29	N/A	N/A	N/A
StyleMapGAN (Ours, 8×8)	**0.082**	**0.024**	**0.242**	**9.97**	**0.037**	**0.304**	**12.42**

Table 3: Comparison with the baselines for real image projection. Runtime covers the end-to-end interval of projection and generation in seconds. FID_{lerp} measures the quality of the images interpolated on the style space as a proxy for the potential quality of the manipulated images. Our method allows real-time manipulation of real images while achieving the best reconstruction accuracy and the best quality of the interpolated images. Although Image2StyleGAN produces the smallest reconstruction error, it suffers from minutes of runtime and poor interpolation quality, which are not suitable for practical editing. Its flaws can be found in the figure: rugged details in overall images, especially in teeth. SEAN is not applicable to AFHQ because it requires segmentation masks for training which are not available. The horizontal line between methods separates optimization-based methods and encoder-based methods.

Experiments

Figure 6: Local editing comparison on CelebA-HQ. The first two baselines [3, 11] even fail to preserve the untouched region. In-DomainGAN loses a lot of the original image's identity and poorly blends the two images, leaking colors to faces, hair, or background, respectively. SEAN locally transfers coarse structure and color but significantly loses details. Ours seamlessly transplants the target region from the reference to the original.

Experiments

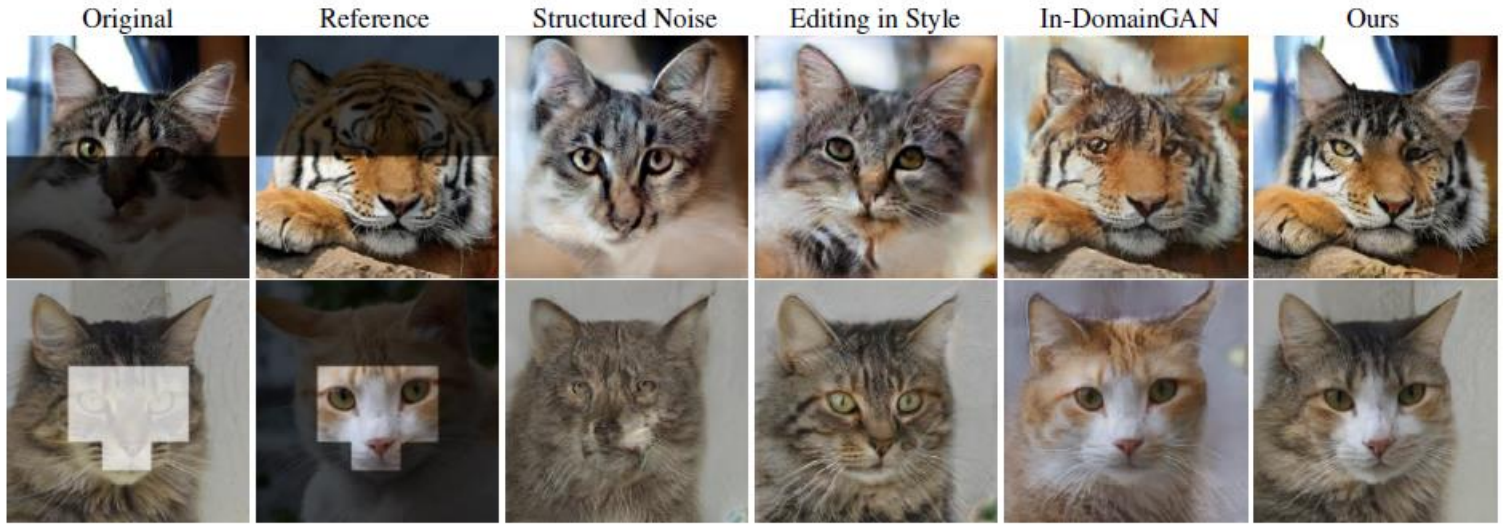


Figure 7: Local editing comparison on AFHQ. Each row blends the two images with horizontal and custom masks, respectively. Our method seamlessly composes two species with well-preserved details resulting in novel creatures, while others tend to lean towards one species.

Method	**Runtime (s)**	**CelebA-HQ**			**AFHQ**		
		AP	MSE_{src}	MSE_{ref}	AP	MSE_{src}	MSE_{ref}
Structured Noise [3]	64.4	99.16	0.105	0.395	99.88	0.137	0.444
Editing in Style [11]	55.6	98.34	0.094	0.321	99.52	0.130	0.417
In-DomainGAN [62]	6.8	98.72	0.164	**0.015**	99.59	0.172	**0.028**
SEAN [65]	0.155	90.41	0.067	0.141	N/A	N/A	N/A
StyleMapGAN (Ours, 8×8)	**0.099**	**83.60**	**0.039**	**0.105**	**98.66**	**0.050**	**0.050**

Table 4: Comparison with the baselines for local image editing. Average precision (AP) is measured with the binary classifier trained on real and fake images [58]. Low AP shows our edited images are more indistinguishable from real images than other baselines. Low MSE_{src} and MSE_{ref} imply that our model preserves the identity of the original image and brings the characteristics of the reference image well, respectively. Our method outperforms in all metrics except MSE_{ref} in In-DomainGAN. In-DomainGAN uses masked optimization, which only optimizes the target mask so that the identity of the original image has a great loss as shown in Figures 6 and 7.

Experiments

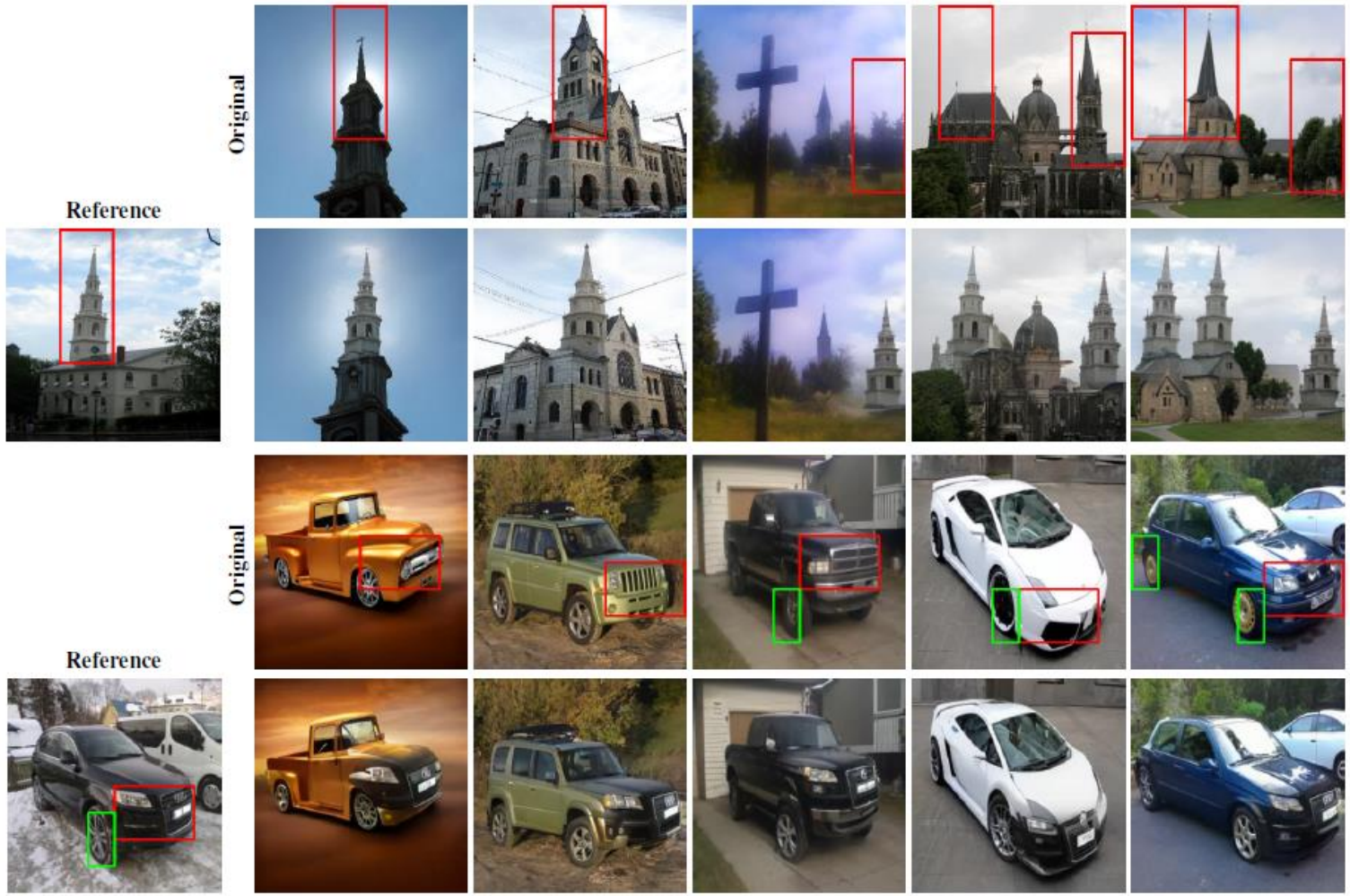


Figure 8: Examples of unaligned transplantation. StyleMapGAN allows composing arbitrary number of any regions. The size and pose of the tower, bumper and wheels are automatically adjusted regarding the surroundings. The masks are specified on 8×8 grid and the stylemaps are blended on w space. The first row shows an example of copying one area of the reference image into multiple areas of the original images. The second row shows another example of copying two areas of the reference image. Our method can transplant the arbitrary number and size of areas of reference images.

Thank you for listening