DIFFUSION MODELS ALREADY HAVE A SEMANTIC LATENT SPACE (ICLR2023)

Mingi Kwon, Jaeseok Jeong, Youngjung Uh

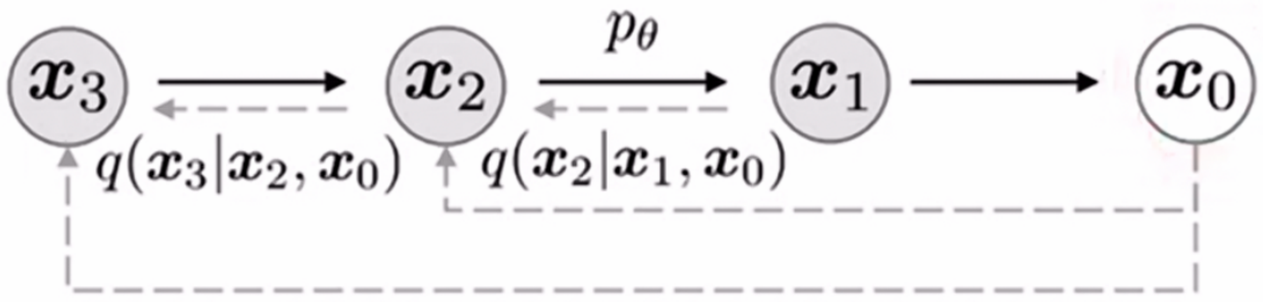
2022.10

신원식, 한상유

MIPAL aboratory
Machine Intelligence
& Pattern Analysis

Background

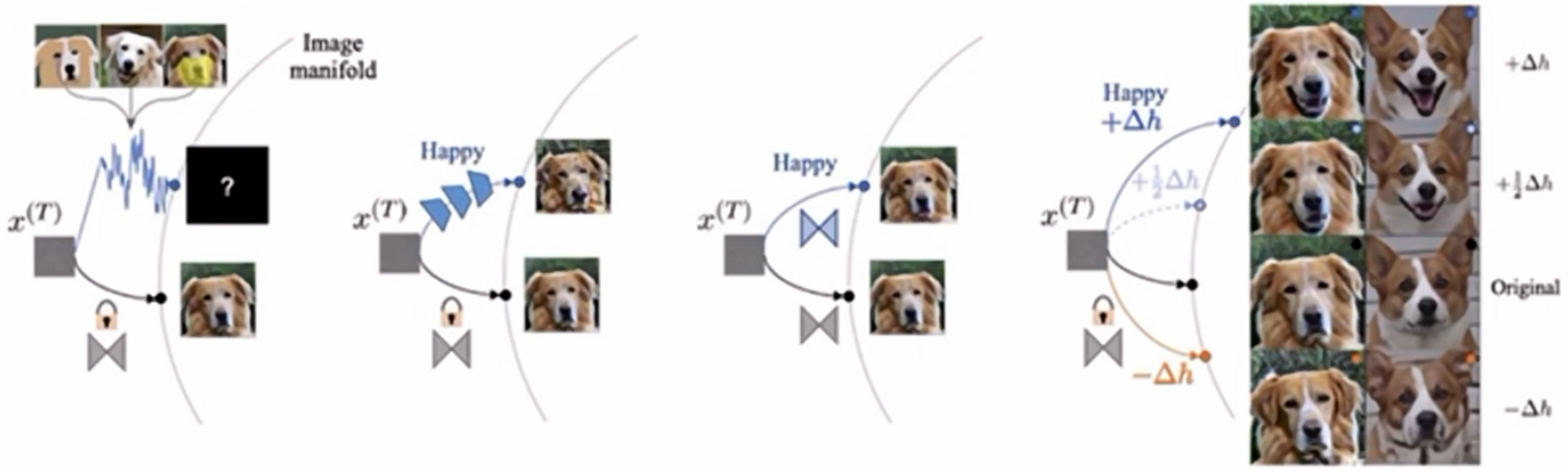
DDIM(Denoising Diffusion Implicit Model)



$$x_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_t^\theta(x_t)}{\sqrt{\alpha_t}} \right)}_{\text{"predicted } x_0\text{"}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_t^\theta(x_t)}_{\text{"direction pointing to } x_t\text{"}} + \underbrace{\sigma_t z_t}_{\text{random noise}}$$

- DDIM 모델은 x_t 시점에서 x_0(real image)를 predict한 뒤, 다시 x_(t-1) 시점의 latent를 예측.
- DDIM 모델은 random noise term의 계수를 0으로 만들어 deterministic 하게 Image generation.

Introduction



(a) Image guidance (b) Classifier guidance (c) DiffusionCLIP (d) Ours

Figure 1: **Manipulation approaches for diffusion models.** (a) Image guidance suffers ambiguity while controlling the generative process. (b) Classifier guidance requires an extra classifier, is hardly editable, degrades quality, or alters the content. (c) DiffusionCLIP requires fine-tuning the whole model. (d) Our method discovers a semantic latent space of a *frozen* diffusion model.

- Image guidance: Attribute를 포함하고 있는 이미지의 latent를 혼합하여 Generation을 진행
- Classifier Guidance: latent 를 특정 class로 classify하는 classifier를 학습하여, 해당 모델의 gradient를 Generation 과정에 주입
- DiffusionCLIP: CLIP Loss를 부여하여 Text condition으로 주어진 attribute를 학습

Problem

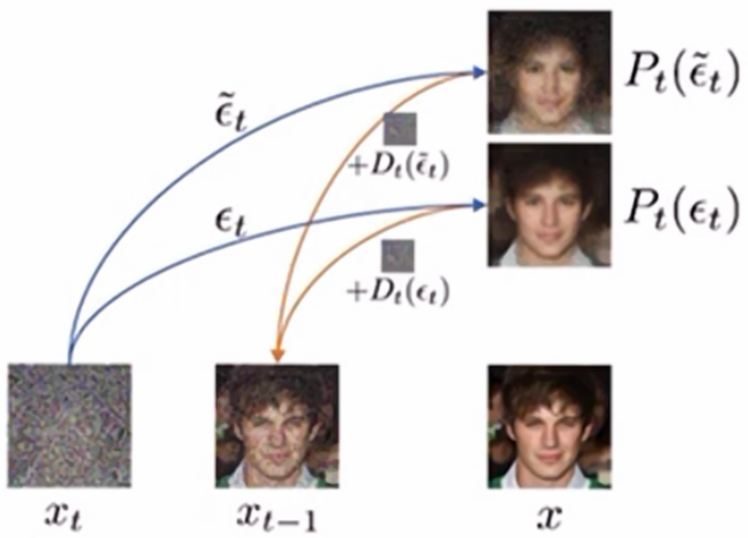


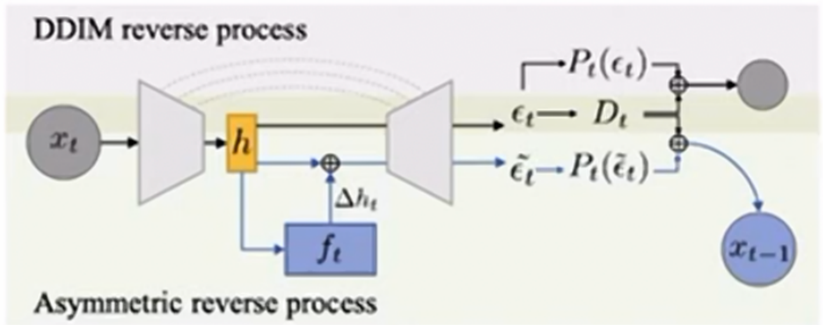
Figure 12: **Illustration of Theorem 1.** Upper blue line describes applying noise $\tilde{\epsilon}_t = \epsilon_t + \Delta\epsilon_t$ to produce $\mathbf{P}_t(\tilde{\epsilon}_t) =$ the shifted predicted x_0. However, the shift due to $\Delta\epsilon_t$ is canceled out by the shift in $\mathbf{D}_t(\tilde{\epsilon}_t)$ due to $\Delta\epsilon_t$. As a results, applying $\Delta\epsilon_t$ both on $\mathbf{P}_t$ and $\mathbf{D}_t$ brings identical outputs to the original.

Proof of **Theorem 1.** Let ϵ_t^θ be a predicted noise during the original reverse process at t and $\tilde{\epsilon}_t^\theta$ be its shifted counterpart. Then, $\Delta x_t = \tilde{x}_{t-1} - x_{t-1}$ is negligible where $\tilde{x}_{t-1} = \sqrt{\alpha_{t-1}}\,\mathbf{P}_t(\tilde{\epsilon}_t^\theta(x_t)) + \mathbf{D}_t(\tilde{\epsilon}_t^\theta(x_t))$.

- CLIP Guidance model의 경우, generate 한 이미지가 target text의 clip embedding과 가까워지도록 u-net output(epsilon)을 update
- 하지만, DDIM model에 이러한 method를 적용할 경우, x_0를 예측한 뒤, x_(t-1) 시점으로 noising을 진행하면, x_0에 적용된 attribute가 사라지는 문제 발생

Method(1)

Asymmetric Reverse Process(Asyrp)



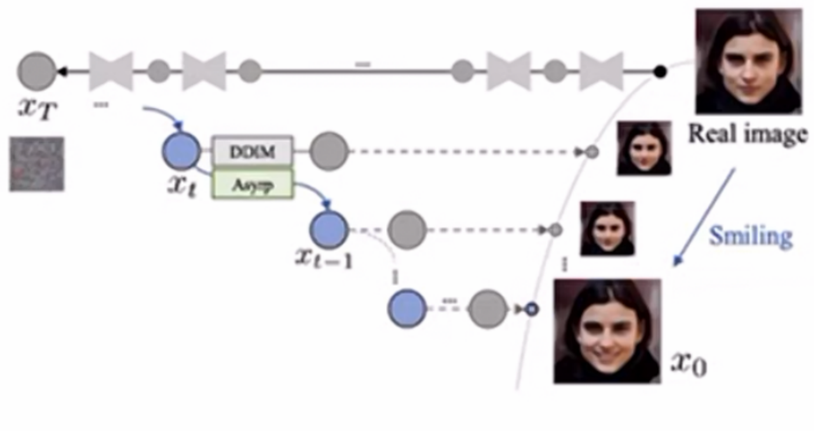


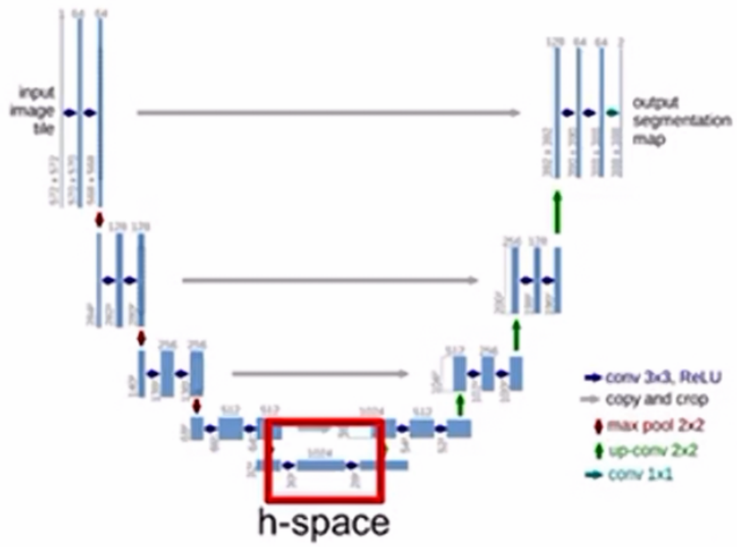
Figure 2: **Generative process of Asyrp.** The green box on the left illustrates Asyrp which only alters $\mathbf{P}_t$ while preserving $\mathbf{D}_t$ shared by DDIM. The right describes that Asyrp modifies the original reverse process toward the target attribute reflecting the change in *h-space*.

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\,\mathbf{P}_t(\tilde{\boldsymbol{\epsilon}}_t^{\theta}(\boldsymbol{x}_t)) + \mathbf{D}_t(\boldsymbol{\epsilon}_t^{\theta}(\boldsymbol{x}_t))$$

- DDIM Model 에서 x_0를 predict 할 때에는 condition을 주입하고, x_0에서 x_(t-1) 시점으로 noising을 할 때에는 pretrained model을 사용
- Backward process와 forward process에서 사용하는 모델이 asymmetric

Method(2)

h-space



$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\,\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t|\Delta\boldsymbol{h}_t)) + \mathbf{D}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)) + \sigma_t\boldsymbol{z}_t,$$

- The same $\Delta\boldsymbol{h}$ leads to the same effect on different samples.
- Linearly scaling $\Delta\boldsymbol{h}$ controls the magnitude of attribute change, even with negative scales.
- Adding multiple $\Delta\boldsymbol{h}$ manipulates the corresponding multiple attributes simultaneously.
- $\Delta\boldsymbol{h}$ preserves the quality of the resulting images without degradation.
- $\Delta\boldsymbol{h}_t$ is roughly consistent across different timesteps t.

- e_t 전체를 shift하는 것이 아니라, u-net의 가장 bottom에 해당하는 space만 shift.
- U-net의 bottom 부분은 down-block과 up-block에 비해 semantic 정보를 더 많이 포함하고 있음
- h-space 만을 manipulate 했을 경우, epsilon 전체를 manipulate 했을 경우보다 더 효율적이고, 좋은 퀄리티를 보임

Method(3)

Implicit Neural Directions

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\,\mathbf{P}_t(\boldsymbol{\epsilon}_t^{\theta}(\boldsymbol{x}_t|\Delta\boldsymbol{h}_t)) + \mathbf{D}_t(\boldsymbol{\epsilon}_t^{\theta}(\boldsymbol{x}_t)) + \sigma_t\boldsymbol{z}_t,$$

$$\mathbf{P}_t^{\text{edit}} = \mathbf{P}_t(\boldsymbol{\epsilon}_t^{\theta}(\boldsymbol{x}_t|\boldsymbol{f}_t))$$

- 모든 t 시점에 대한 h_t를 학습하는 데에는 너무 많은 iteration이 필요함
- t 시점의 h_t를 생성하는 f_t(small neural network with two 1x1 convolutional network) 를 학습하여 모든 t 시점의 h_t를 생성

Method(4)

Editing Process

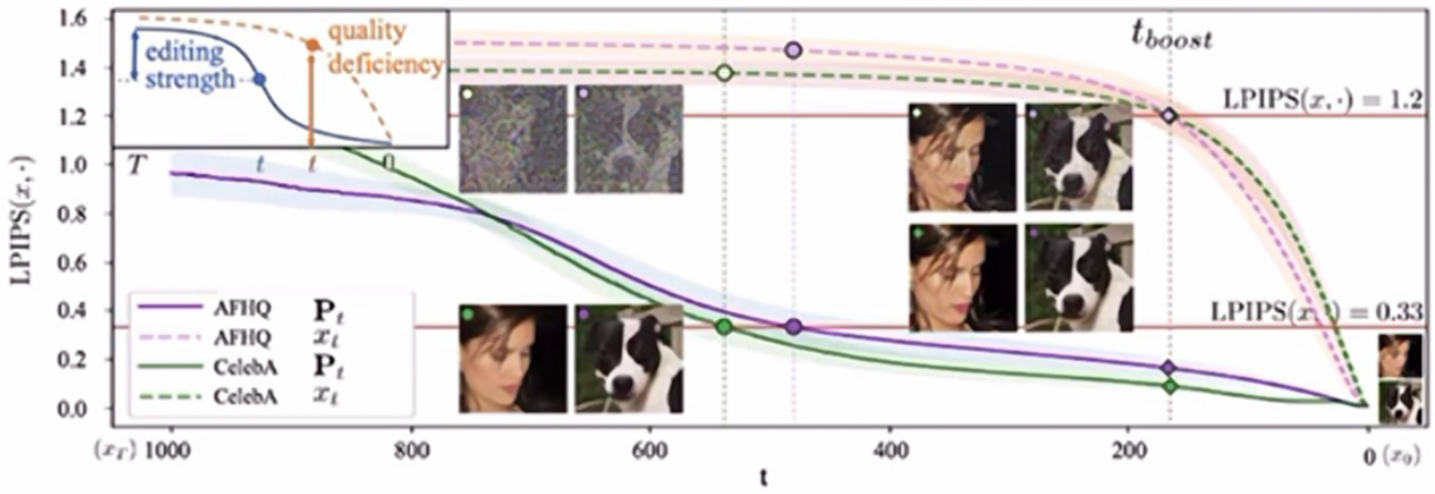


Figure 3: **Intuition for choosing the intervals for editing and quality boosting.** We choose the intervals by quantifying two measures (top left inset). The editing strength of an interval $[T, t]$ measures its perceptual difference from T until t. We set $[T, t]$ to the interval with the smallest editing strength that synthesizes $\mathbf{P}_t$ close to $\boldsymbol{x}$, i.e., $\text{LPIPS}(\boldsymbol{x}, \mathbf{P}_t) = 0.33$. Editing flexibility of an interval $[t, 0]$ measures the potential amount of changes after t. Quality deficiency at t measures the amount of noise in $\boldsymbol{x}_t$. We set $[t, 0]$ to handle large quality deficiency (i.e., $\text{LPIPS}(\boldsymbol{x}, \boldsymbol{x}_t) = 1.2$) with small editing flexibility.

- t_edit 시점까지 h-space를 manipulate함. editing strength를 최대화하는 minimum t시점을 empirical 탐색
- t_edit시점이 너무 t_0시점에 가까울 경우, image quality가 너무 떨어짐
- t_boost 시점부터 t_0 시점까지 random noise를 inject하여 image quality를 높임

Method(5)

overall generation process

$$p_\theta^{(t)}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \begin{cases} \mathcal{N}\left(\sqrt{\alpha_{t-1}}\,\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t|\boldsymbol{f}_t)) + \mathbf{D}_t, \sigma_t^2 \boldsymbol{I}\right), \ \eta = 0 & \text{if } T \geq t \geq t_{\text{edit}} \\ \mathcal{N}\left(\sqrt{\alpha_{t-1}}\,\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)) + \mathbf{D}_t, \sigma_t^2 \boldsymbol{I}\right), \ \eta = 0 & \text{if } t_{\text{edit}} > t \geq t_{\text{boost}} \\ \mathcal{N}\left(\sqrt{\alpha_{t-1}}\,\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)) + \mathbf{D}_t, \sigma_t^2 \boldsymbol{I}\right), \ \eta = 1 & \text{if } t_{\text{boost}} > t \end{cases}$$

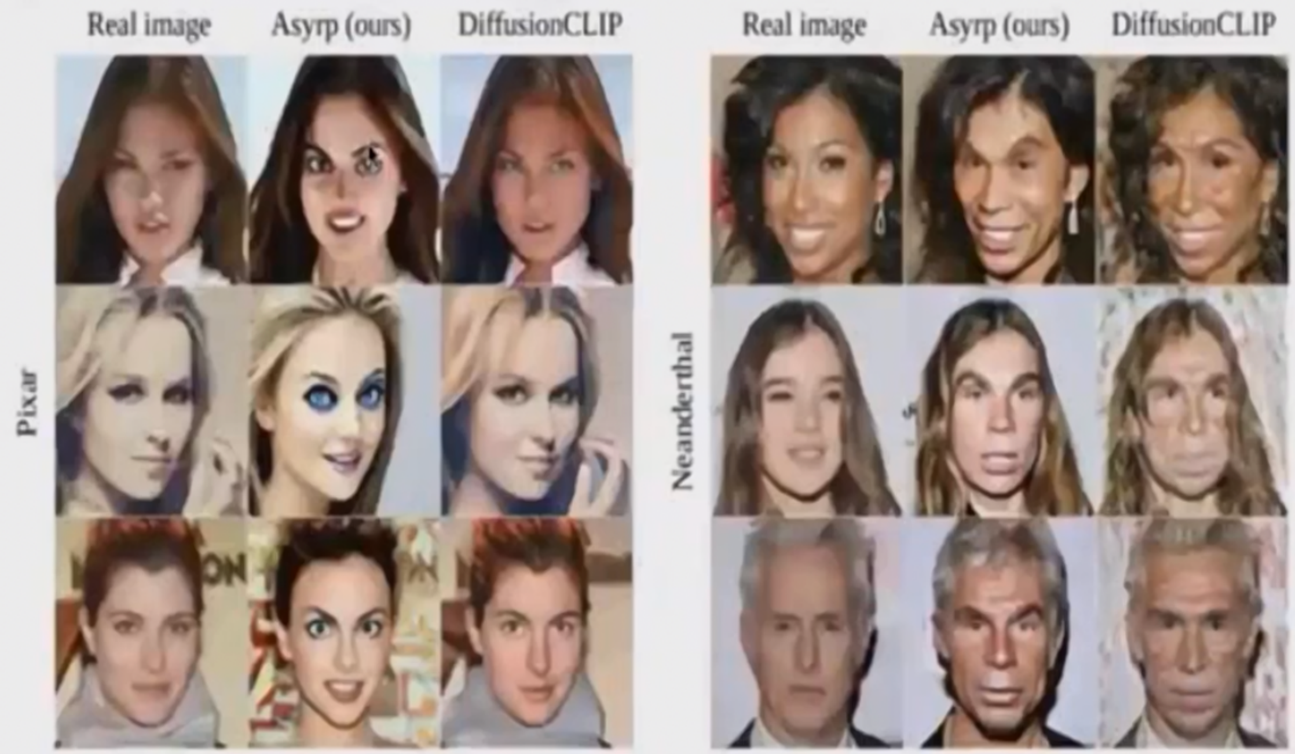
Experiments

Experiments

MIPAL aboratory
Machine Intelligence & Pattern Analysis

Experiments

- Unseen domain에서도 잘 작동



- User study에서도 DiffusionCLIP보다 더 나은 성능을 보임.

	CelebA-HQ in-domain			CelebA-HQ unseen-domain			LSUN-church			
	quality	attribute	overall	quality	attribute	overall	quality	attribute	diversity	overall
Asyrp (ours)	**98.36%**	**88.13%**	**94.92%**	**71.56%**	**59.84%**	**63.13%**	**73.19%**	**71.81%**	**87.50%**	**76.81%**
DiffusionCLIP	1.64%	11.88%	5.08%	28.44%	40.16%	36.88%	26.81%	28.19%	12.50%	23.19%

Table 1: **User study** with 80 participants. The details are described in § K.1

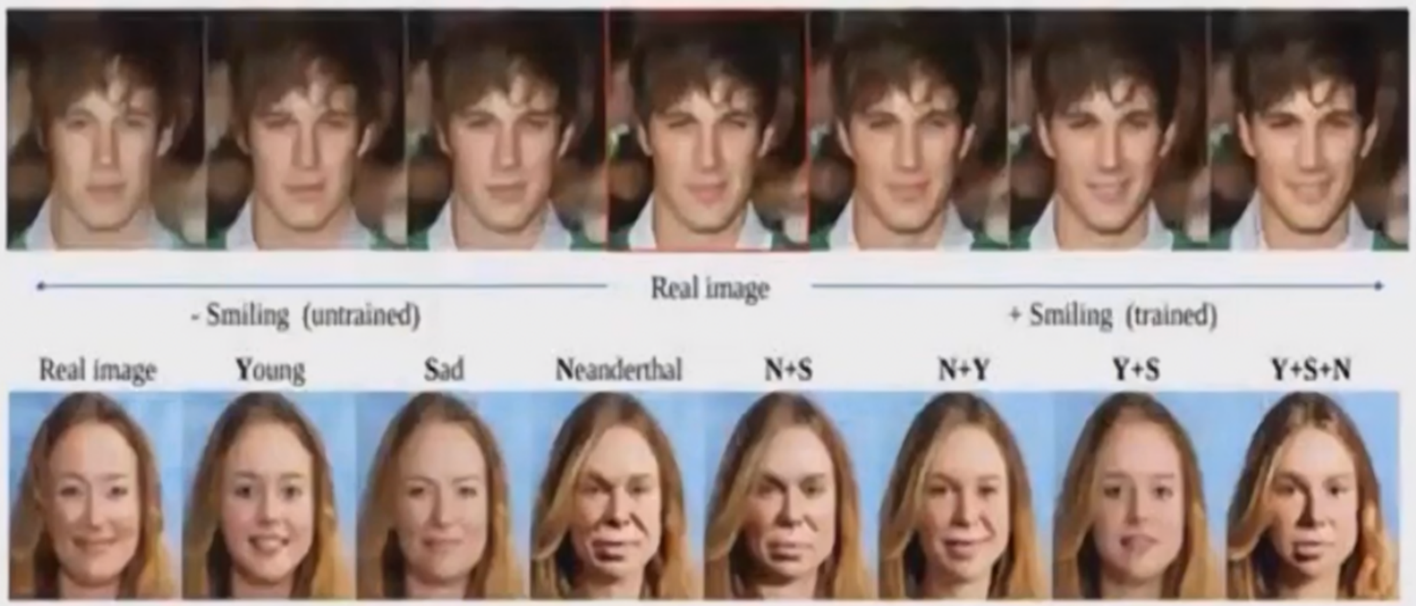
h-space analysis

- Homogeneity : 한 샘플에 대해 찾은 Δh_t가 다른 샘플에 대해서도 적용됨. 반면, $\Delta \epsilon_t$는 다른 샘플에 적용이 안됨.

(a) Result of optimization

(b) Result of adaptation to other images

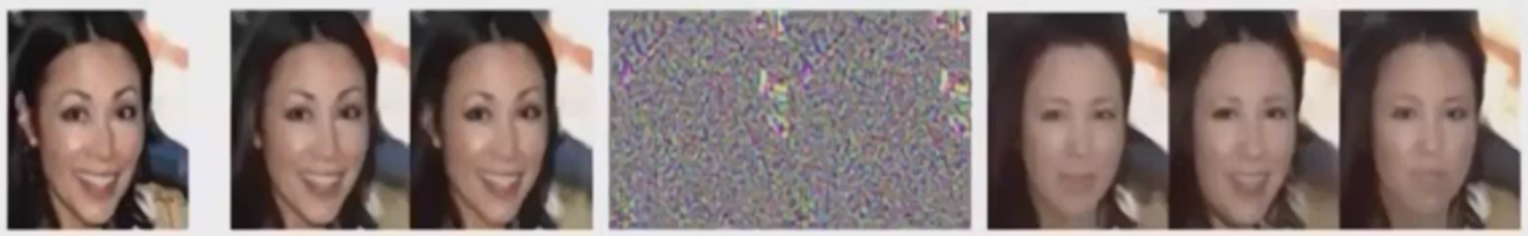
- Linearity : 찾은 Δh_t를 선형적으로 변화시키면 결과물도 선형적으로 변함. 특히, Δh_t의 반대 방향은 학습시키지 않았음에도 잘 적용됨.



h-space analysis

- Linearity : 찾아낸 Δh_t들을 Linear combination하여 생성된 특정들을 합칠 수 있음.

- Robustness : Δh_t에 랜덤한 노이즈를 주입하여도 특징을 잘 보존함. 반면, $\Delta \epsilon_t$에 노이즈를 주입할 경우는 이미지 생성이 불가능



Real image (a) Δh (b) $\Delta \epsilon$ (c) Δh ($\times 3$)