Generative AI
: Deep Generative Models

MILab Undergraduate student, Kim Taehyeon

2023. 08. 17

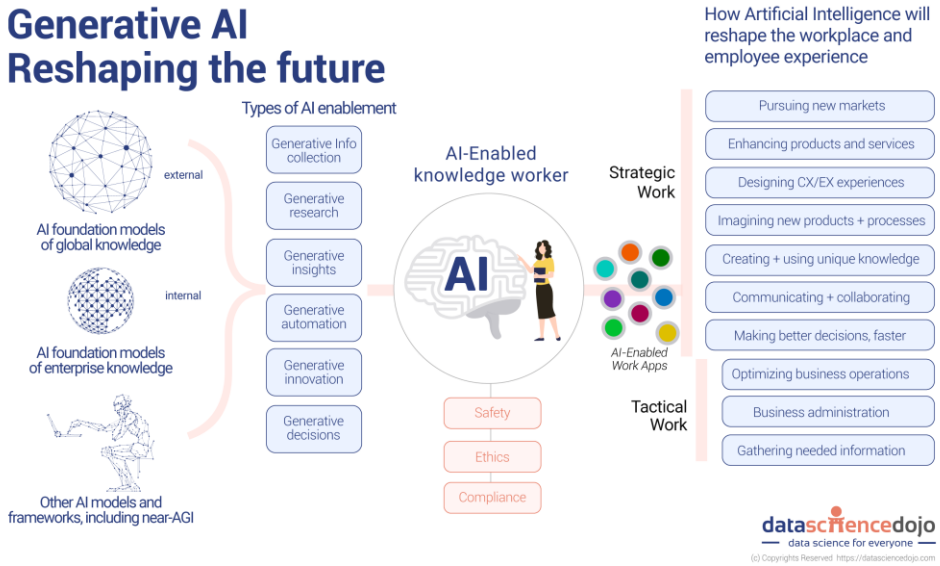
목차

1. 생성형 AI 리마인드

1. 생성형 AI 란

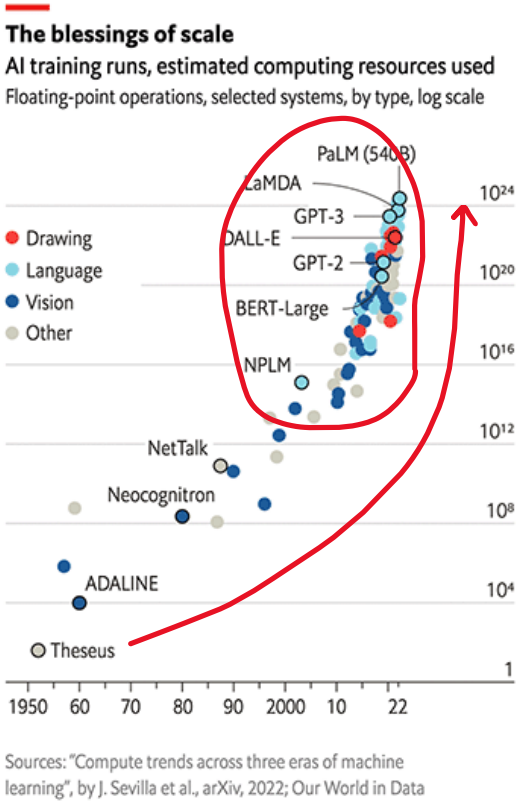
생성형 AI (Generative AI) 는 비정형 데이터와 딥러닝 모델을 사용하여 사용자 입력을 기반으로 콘텐츠를 생성하는 인공지능이다.

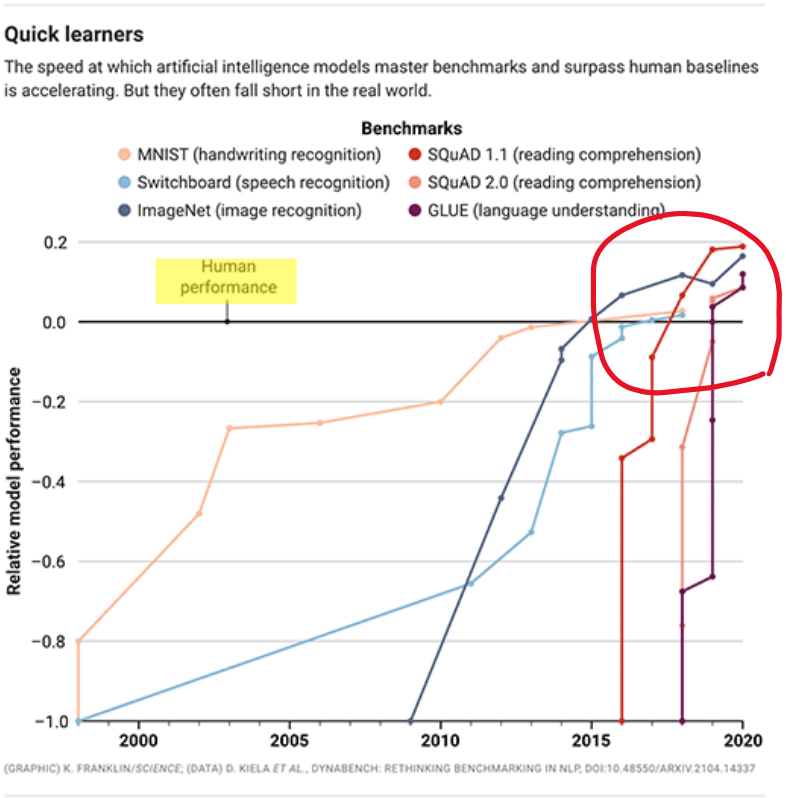
예를 들어, **ChatGPT** 에 질문을 입력하면 간단하지만 합리적이고 상세한 답변을 제공해주고, 후속 질문에 대하여 대화 초기의 내용을 기반으로 답변이 가능하다.



1. 생성형 AI 리마인드

2. 모델의 발전



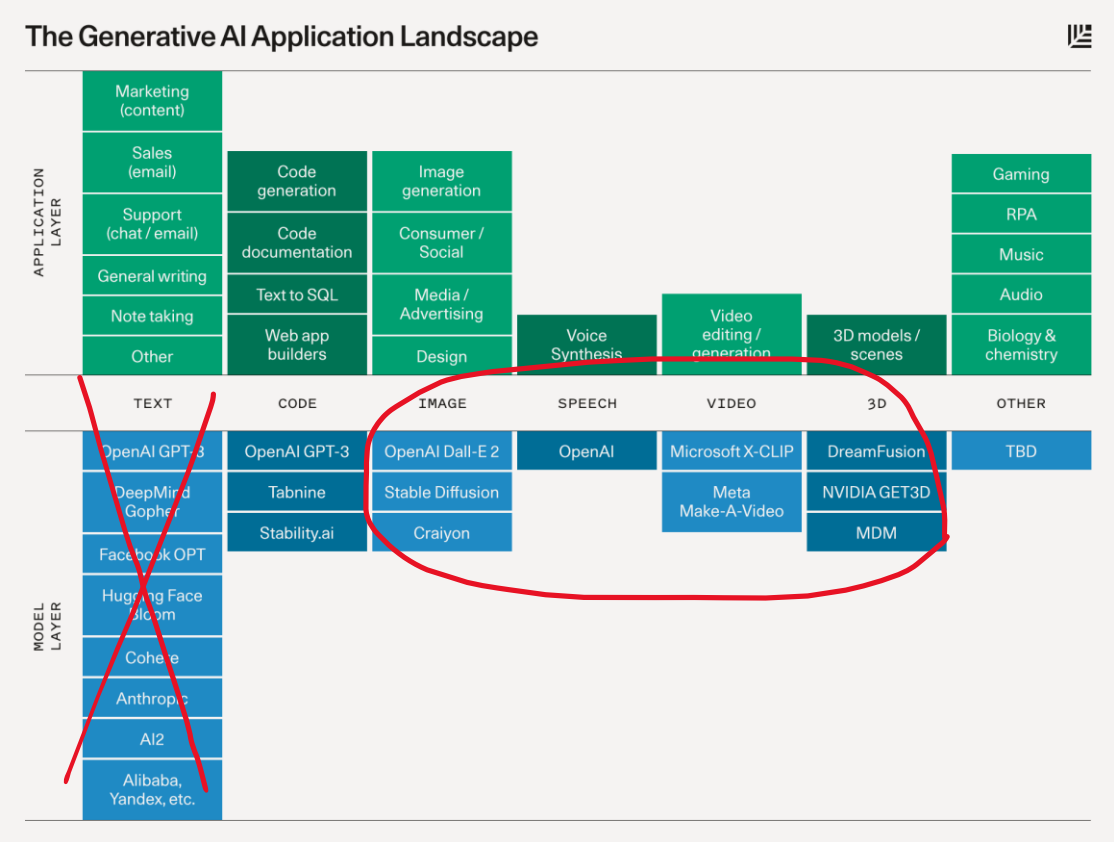


현재 방대한 정보의 양을 바탕으로 대부분 모델의 성능이 비약적으로 발전함.

이러한 상황이 생성형 AI 의 급발전이 가능하게 된 초석이 되었음.

1. 생성형 AI 리마인드

2. 모델의 발전



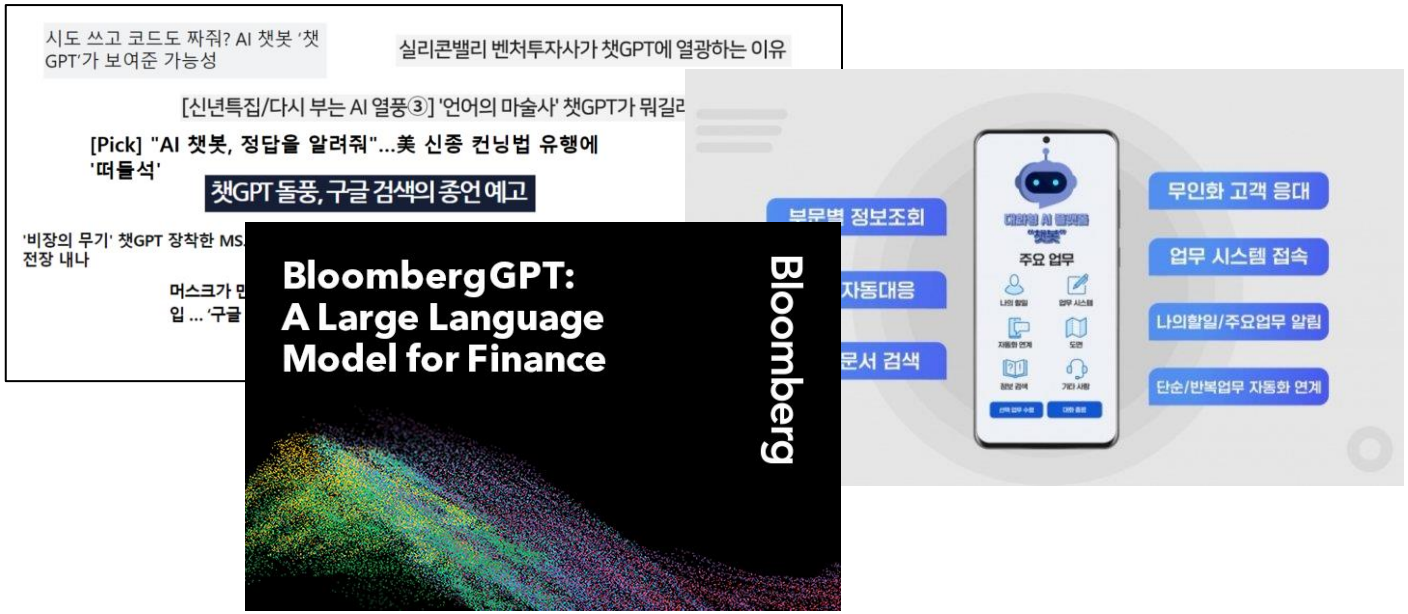
텍스트 관련 생성형 AI 은 이전 세미나에서 다루었음.

이번에는 IMAGE, SPEECH, VIDEO, 3D 에 대한 설명이 주가 될 것임.

목차

2. 생성형 AI 적용 사례

0. 산업별 활용 사례 도입부



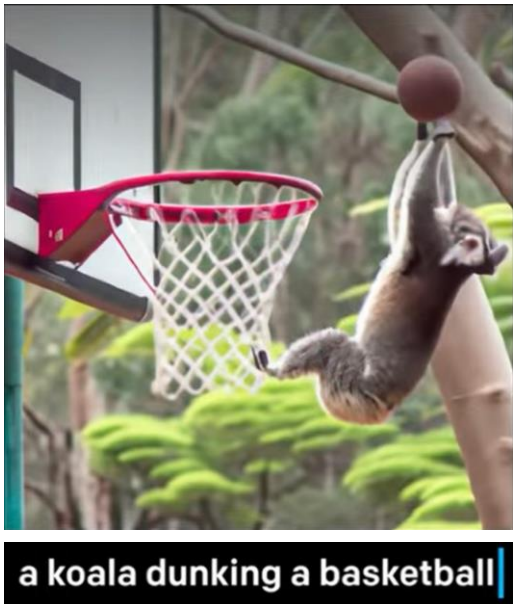
게임 업계 / 텍스트 생성 AI 를 바탕으로 게임 시나리오 등 콘텐츠 생성 자동화

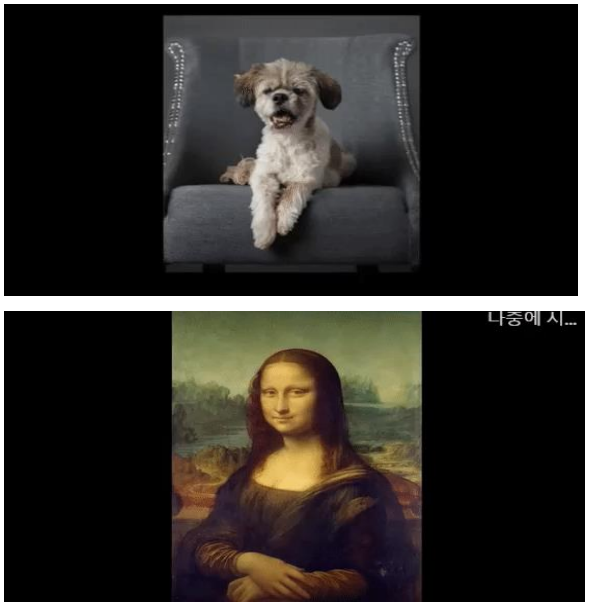
금융 업계 / 챗봇을 바탕으로 고객 서비스 에이전트, 대출 도우미 등 진행

2023년 3월 말 블룸버그에서 금융 뉴스를 학습시킨 블룸버그GPT 출시

2. 생성형 AI 적용 사례

1. **Image**, Video, 3D





이미지에 대한 생성형 AI 는 Image Generation, Design 등에 사용이 됨.

그 예시로는 OpenAI Dall-E 2, Stable Diffusion 이 있음.

위 자료는 OpenAI Dall-E 2 가 지원하는 기능임.

2. 생성형 AI 적용 사례

1. Image, **Video**, 3D

동영상의 경우, 이미지를 연속적으로 보여준 것과 동일함.

따라서, 기술 메커니즘은 이전의 Image 예시와 유사함.

Video 에서는 Video Generation 기능을 대체로 제공함.

2. 생성형 AI 적용 사례

1. Image, Video, **3D**

DreamFusion: Text-to-3D using 2D Diffusion

Ben Poole, Google Research
Ajay Jain, UC Berkeley
Jonathan T. Barron, Google Research
Ben Mildenhall, Google Research

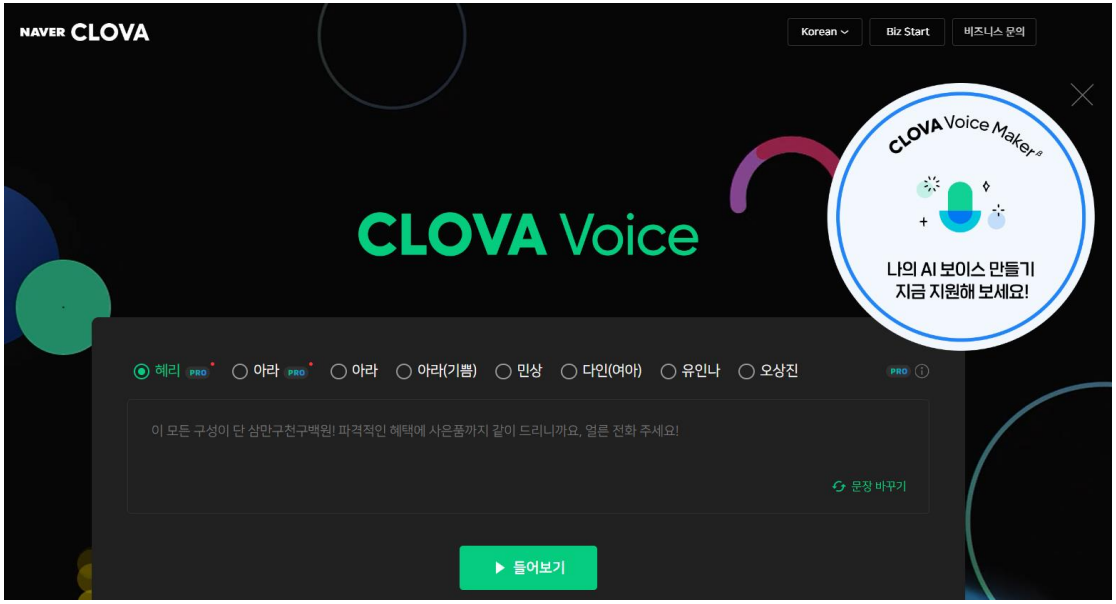
`

3D 는 모델링을 목적으로 기능을 제공함.

예시로는 DreamFusion, NVIDIA GET3D 가 있음.

2. 생성형 AI 적용 사례

2. Speech



CLOVA Voice, OpenAI 등 Speech Synthesis 영역에서도 생성형 AI 가 쓰임.

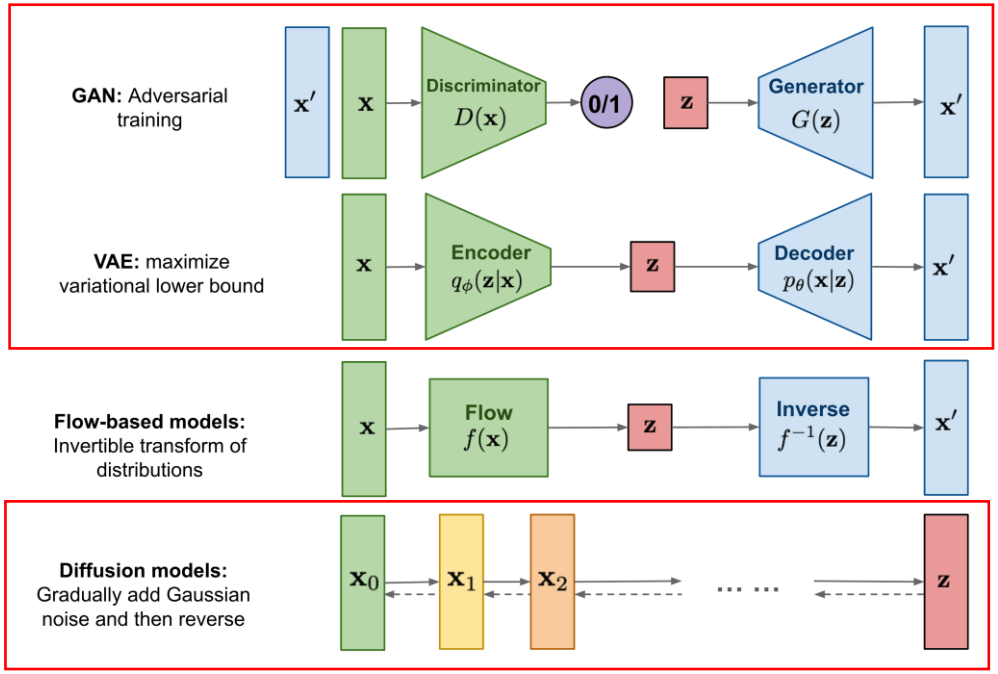
말투, 음색, 습관 등을 특징 잡아 학습한 Unit-selection TTS 과

Deep Neural Network 를 바탕으로 학습한 End-to-End TTS 모델을 제공함.

목차

3. 적용 기술 설명

0. 들어가기에 앞서..

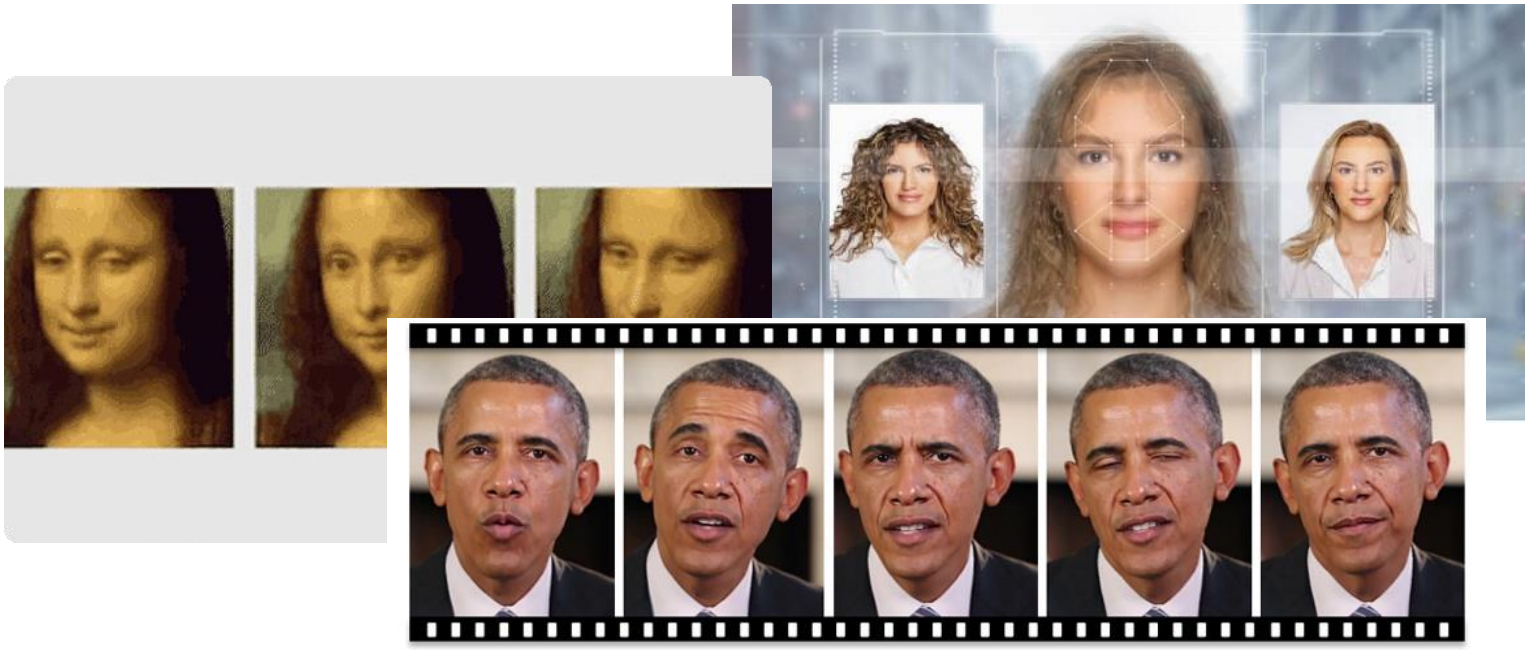


대표적인 생성형 AI 모델인 GAN 과 VAE, Diffusion 모델을 설명하고

앞에 설명한 다양한 사례들 중 Image, Video 에 대한 작동 방식을 설명할 것임.

3. 적용 기술 설명

1-1. GAN 모델 (Generative Adversarial Networks)



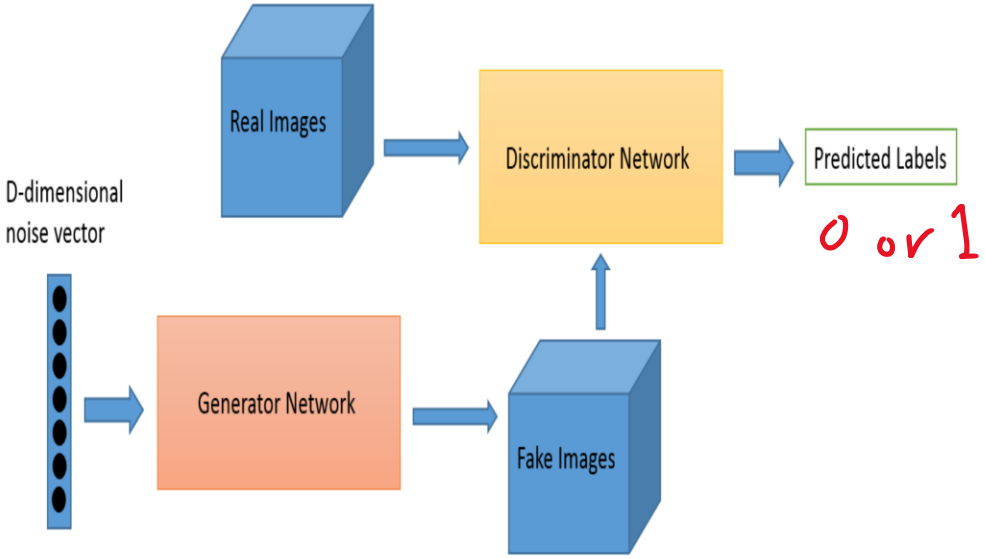
GAN 은 Generative Adversarial Networks 모델로 실제에 가까운 이미지나

사람이 쓴 것과 같은 글 등 여러 가짜 데이터들을 생성하는 모델임.

서로 다른 두 개의 네트워크를 Adversarial 하게 학습시킴.

3. 적용 기술 설명

1-1. GAN 모델 (Generative Adversarial Networks)



$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim p_{\text{data}}} \log D_{\phi}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\phi}(G_{\theta}(z)))$$

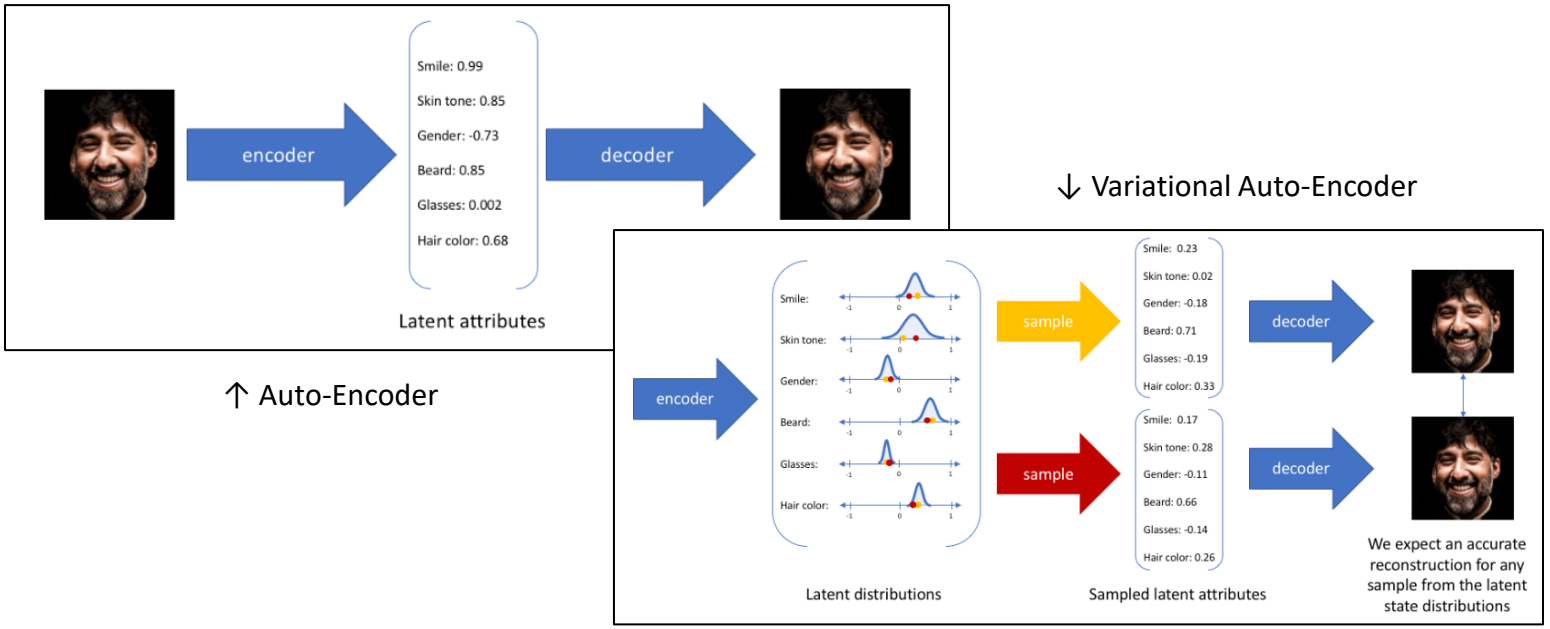
$$\max_{\phi} \mathbb{E}_{x \sim p_{\text{data}}} \log D_{\phi}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\phi}(G_{\theta}(z)))$$

간단하게, 생성기(G)는 실제 데이터와 비슷한 데이터를 만들어내도록 학습함.

판별기(D)는 실제 데이터와 G가 생성한 가짜 데이터를 구별하도록 학습됨.

3. 적용 기술 설명

1-2. VAE 모델 (Variational Auto-Encoder)



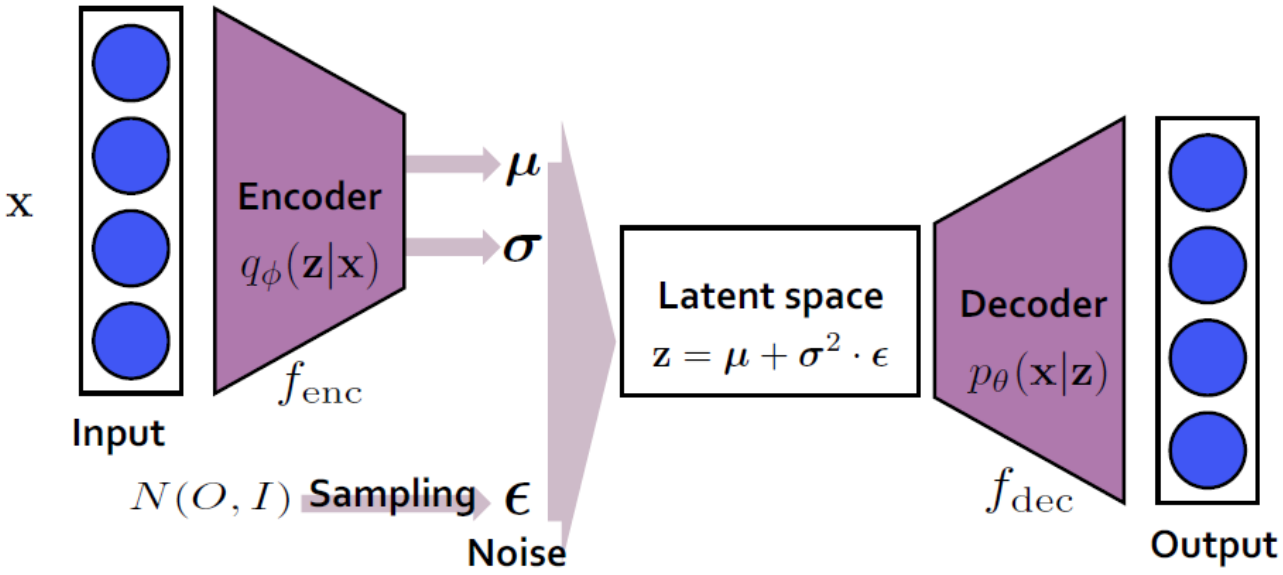
우선, **Auto-Encoder** 에서 Encoder 는 하나의 Single Value 를 출력함.

Decoder 는 이 값을 사용하여 원본 데이터를 복원하도록 학습이 됨.

반면에, VAE 모델에서 Encoder 는 Single Value 가 아닌 확률 분포를 출력함.

3. 적용 기술 설명

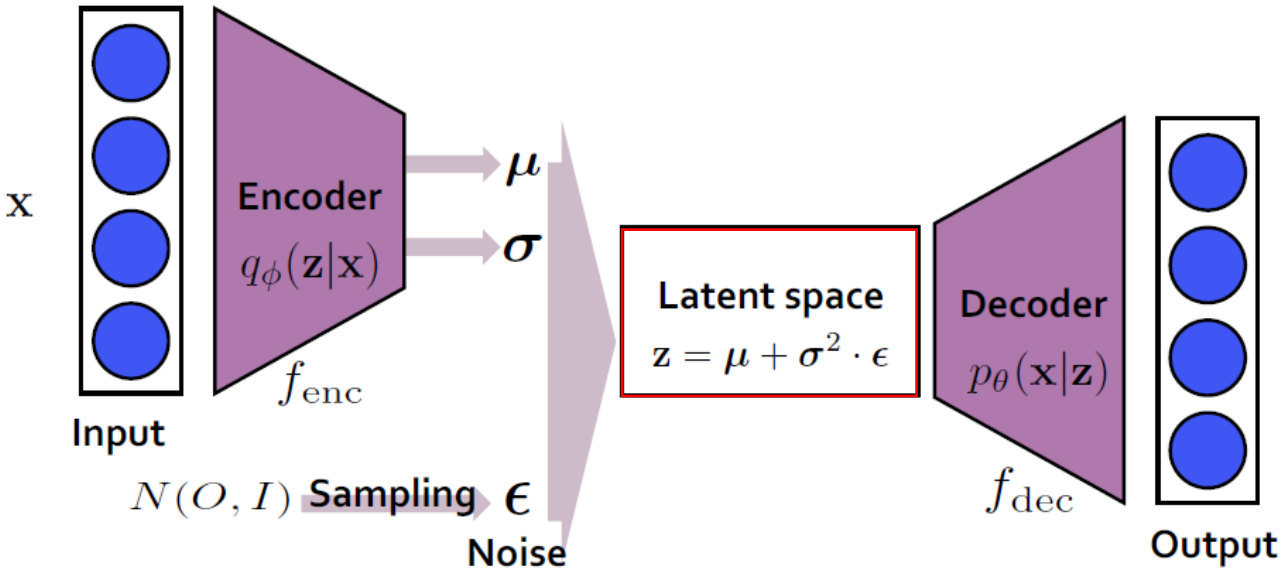
1-2. VAE 모델 (Variational Auto-Encoder)



Variational Auto-Encoder (VAE) 는 Generative Model 의 한 종류로, input 과 output 을 같게 만드는 것을 통해 encoder 와 decoder 를 활용하여 latent space 를 도출하고, 이 latent space 로부터 우리가 원하는 output 을 decoding 하는 방식으로 데이터를 생성함.

3. 적용 기술 설명

1-2. VAE 모델 (Variational Auto-Encoder) – Latent Vector

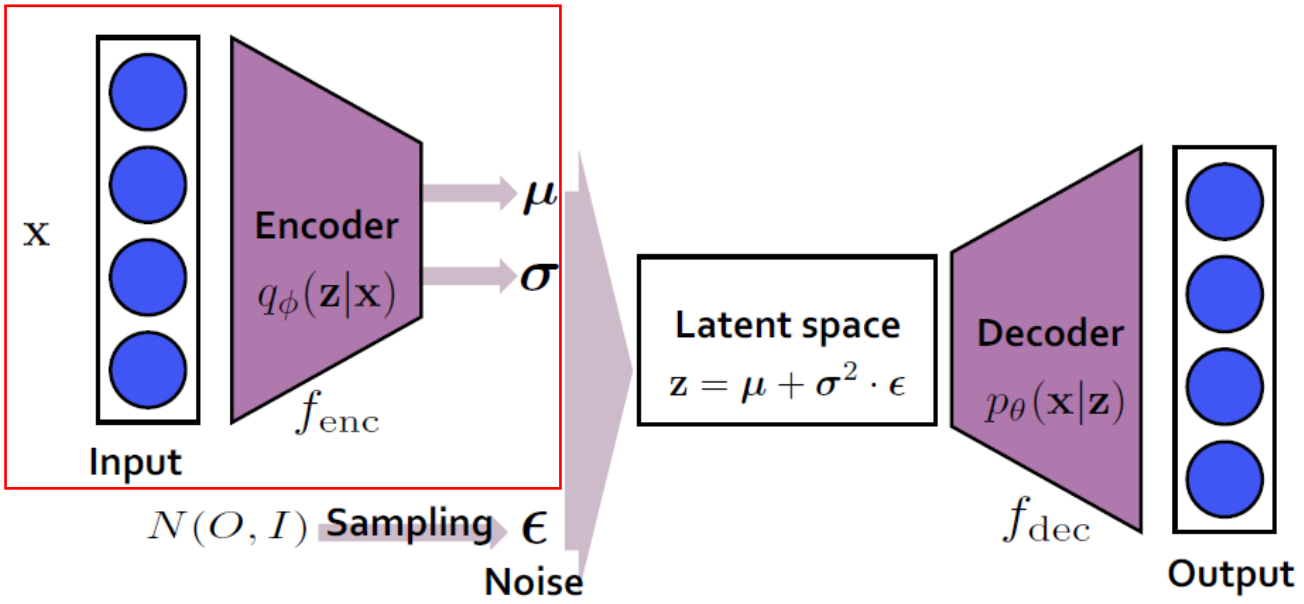


완전히 Input 과 Output 을 똑같이 만드는 Auto-Encoder 가 아닌 확률 분포를 반환하는 구조를 만들기 위하여 Latent Space 를 채택하였음.

noise epsilon 를 gaussian sampling 하여 얻고, Encoder 에서 나온 결과에서 분산을 곱하고 평균을 더해서 latent vector z 를 구해냄.

3. 적용 기술 설명

1-2. VAE 모델 (Variational Auto-Encoder) – Encoder



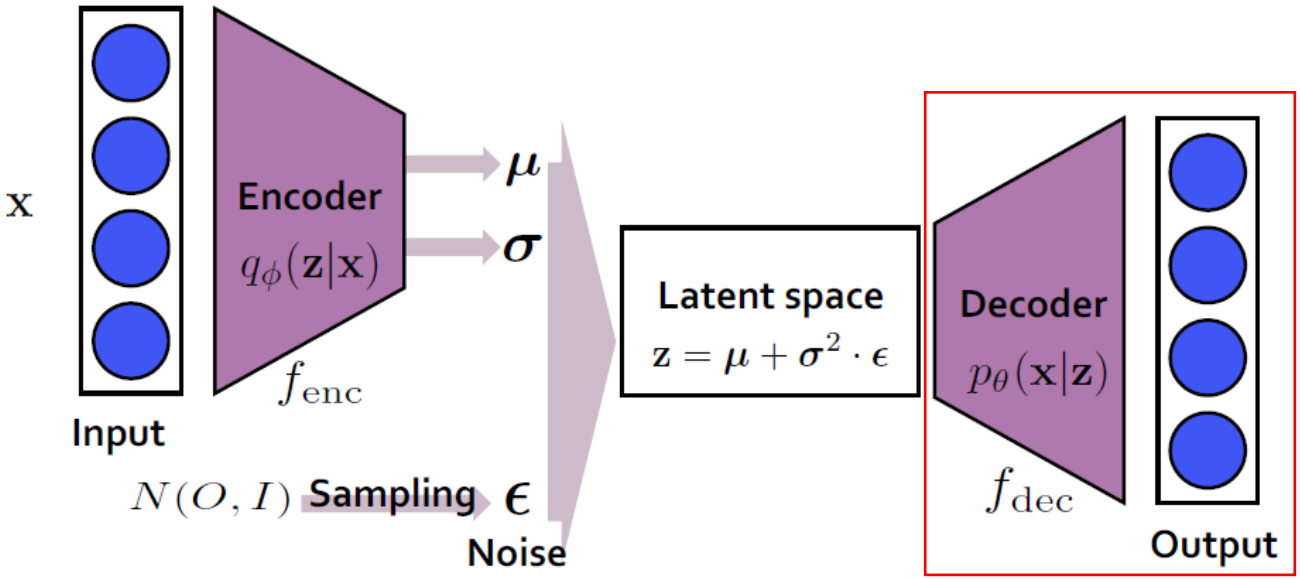
Encoder 는 input 을 latent space 로 변환하는 역할을 함.

Encoder 는 input x 데이터에 대하여 latent vector z 의 분포(=q(z|x))를 추정함.

q(z|x) 에 대한 분포를 잘 구해내기 위하여 파라미터를 잘 구하여야 한다.

3. 적용 기술 설명

1-2. VAE 모델 (Variational Auto-Encoder) – Decoder



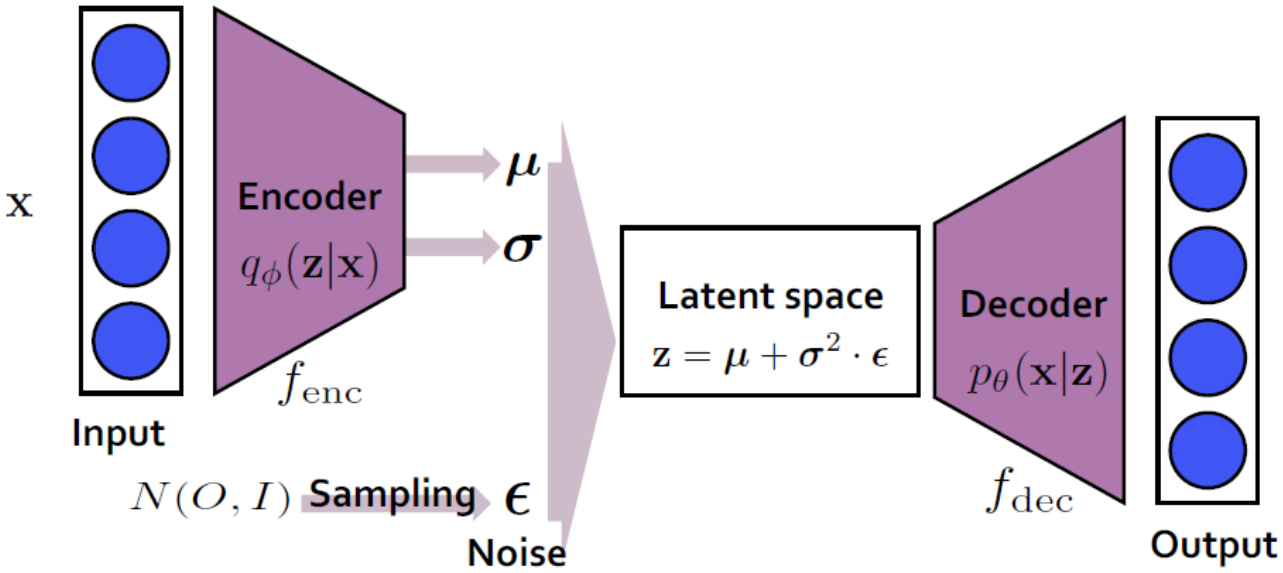
Decoder 는 Encoder 와 반대로 latent space 를 input 으로 변환하는 역할을 함.

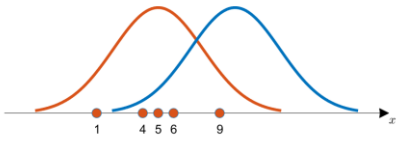
Latent vector z 가 주어졌을 때, output x 가 나오도록 추정함.

따라서, Decoder 가 Generative Model 의 역할을 진행함.

3. 적용 기술 설명

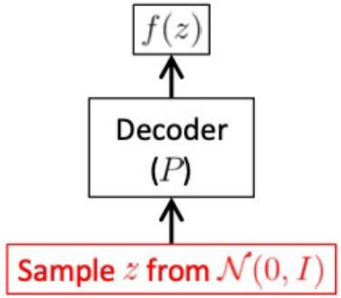
1-2. VAE 모델 (Variational Auto-Encoder) – ELBO, Data Generation





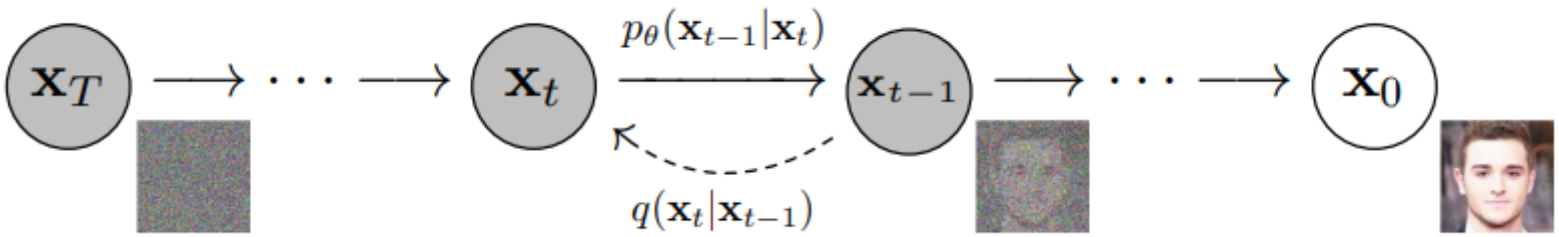
Maximum Likelihood

$\log p_\theta(\mathbf{x})$



3. 적용 기술 설명

1-3. Diffusion 모델



P(x0) = ?

Diffusion Model 은 noise(=Xt) 로부터 data(=X0) 를 복원해내는 모델임.

Forward Process 는 본래의 data 에서 noise 로 변화하는 과정임.

Reverse Process 는 noise 에서 본래의 data 로 가는 과정으로 데이터를 생성함.

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

목차

3. 적용 기술 설명

2-1. Text2Image

(a) a tapir made of accordion. a tapir with the texture of an accordion.

(b) an illustration of a baby hedgehog in a christmas sweater walking a dog

(c) a neon sign that reads "backprop". a neon sign that reads "backprop". backprop neon sign

(d) the exact same cat on the top as a sketch on the bottom

Dall-E 가 그려낸 다양한 이미지들임.

상황별 세부 정보 추론 / 개체별 각 속성 및 개체 간 관계와 공간 관계 파악

Ex) "코끼리-코, 고슴도치-가시, …", 이미지 캡션에 있어서 연관성 매칭하여 인지

3. 적용 기술 설명

2-1. Text2Image

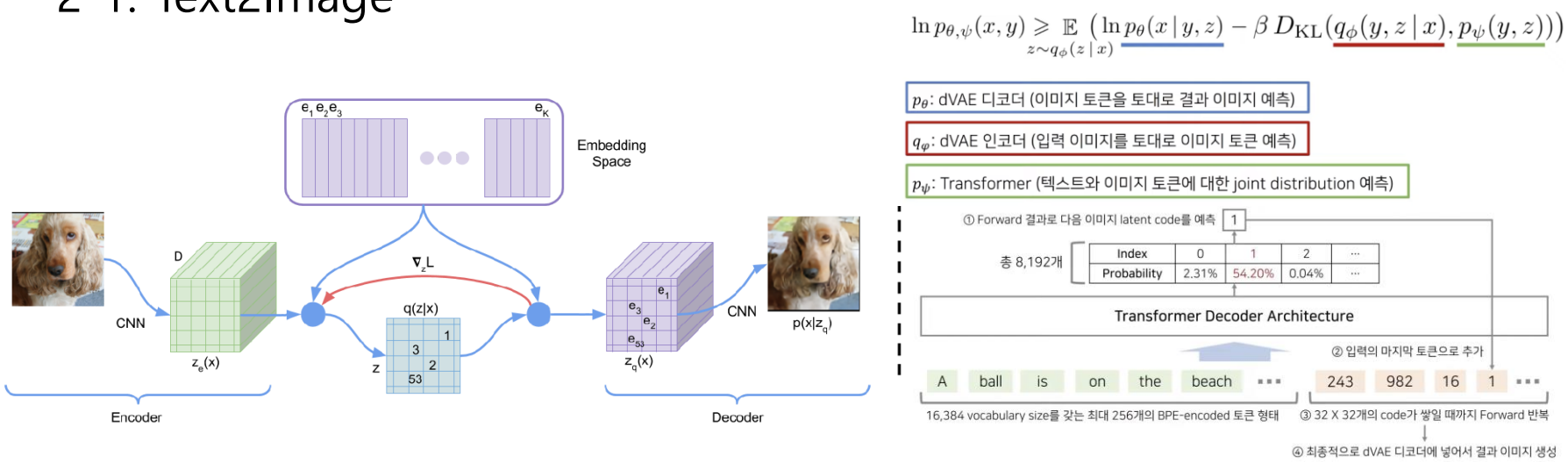


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

전체적으로는 이미지 인식 기술과 자연어 처리를 함께 사용함.

일단, Variational Auto-Encoder 모델을 바탕으로 전체적인 확률 분포를 학습함.

VAE Encoder -> Transformer Decoder -> VAE Decoder 순으로 학습.

3. 적용 기술 설명

2-1. Text2Image

<MS-COCO captioning task / https://cocodataset.org/#captions-2015>

Dall-E 는 넷 상 이미지(픽셀)와 자연어 캡션을 대량 사용하는 방식으로 제작되었음.

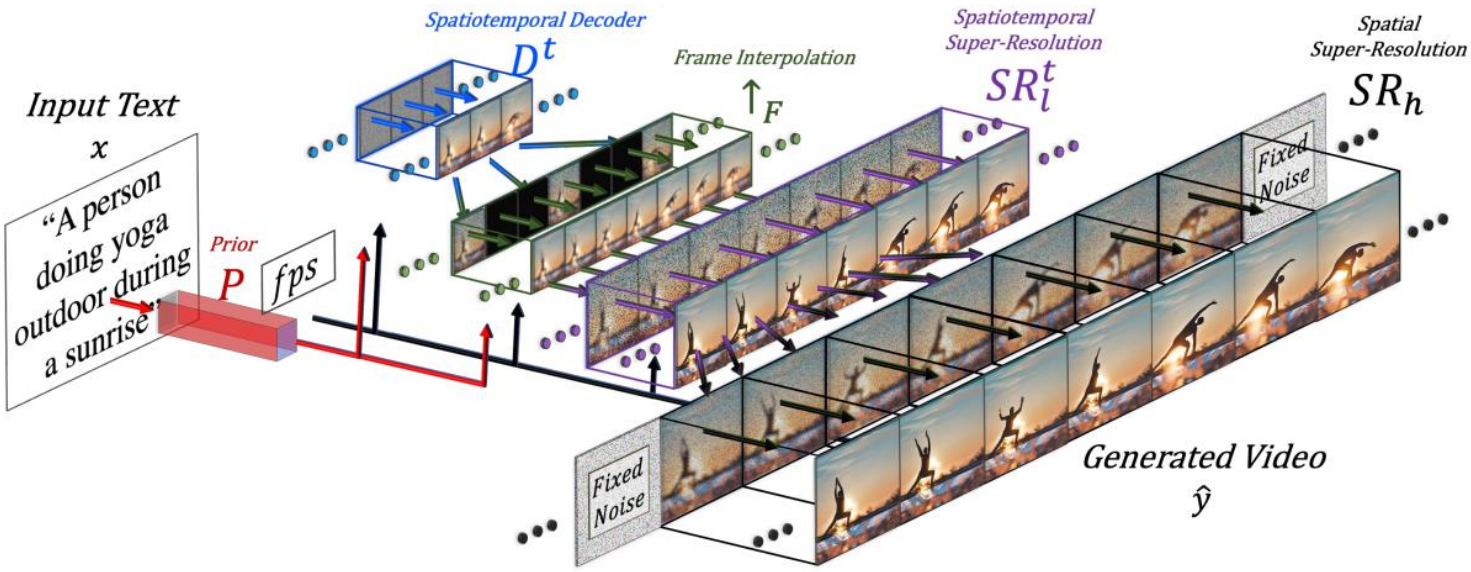
VAE 와 Transformer 모델로 상관 없는 이미지들 간에 관계를 파악하여 조합함.

GPT-3 제로샷 추론 기능을 시각 영역으로 확장하였음.

3. 적용 기술 설명

2-2. Text2Video

$$\hat{y}_t = \mathrm{SR}_h \circ \mathrm{SR}_l^t \circ \uparrow_F \circ \mathrm{D}^t \circ \mathrm{P} \circ (\hat{x}, \mathrm{C}_x(x)),$$



동영상 제작에 있어서, 학습된 디코더를 바탕으로 몇 개의 이미지 생성 후에

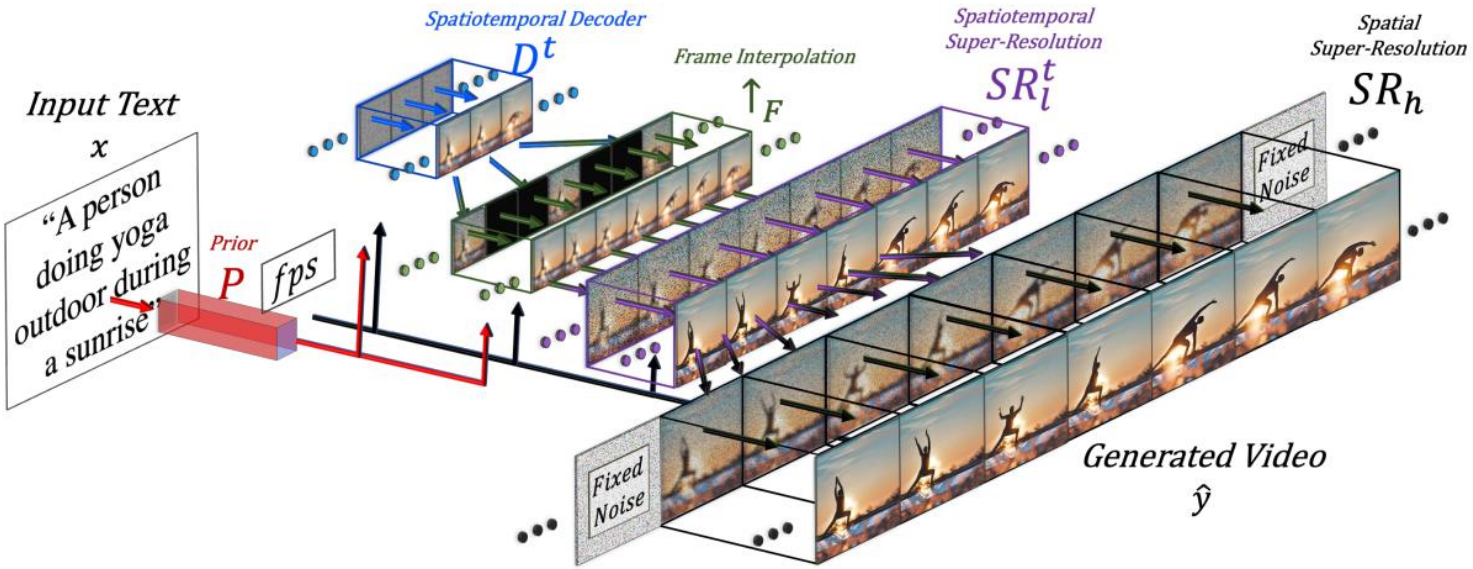
Frame Interpolation 를 바탕으로 자연스럽게 프레임들을 생성한다.

그 이후에, Super-Resolution 을 바탕으로 고품질의 영상을 제작한다.

3. 적용 기술 설명

2-2. Text2Video

$$\hat{y}_t = \mathrm{SR}_h \circ \mathrm{SR}_l^t \circ \uparrow_F \circ \mathrm{D}^t \circ \mathrm{P} \circ (\hat{x}, \mathrm{C}_x(x)),$$



(i) TEXT-TO-IMAGE MODEL

(ii) SPATIOTEMPORAL LAYERS

  (i) PSEUDO-3D CONVOLUTIONAL LAYERS

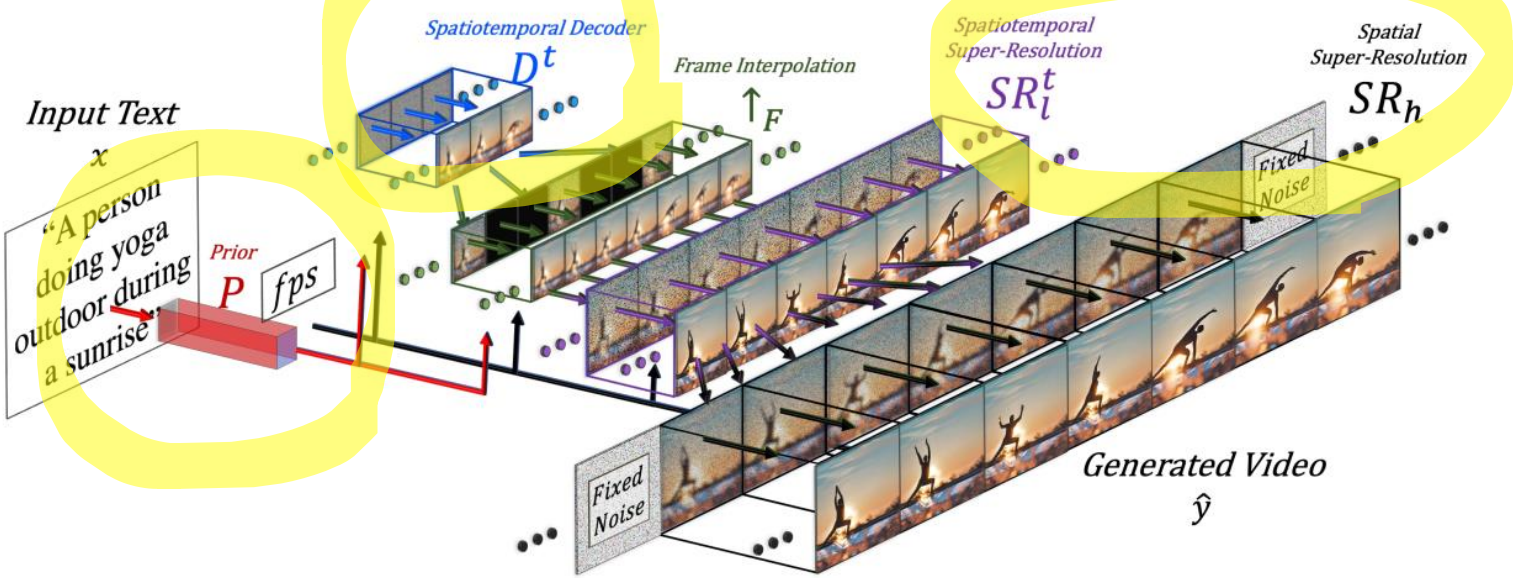
  (ii) PSEUDO-3D ATTENTION LAYERS

(iii) FRAME INTERPOLATION NETWORK

3. 적용 기술 설명

2-2. Text2Video

$$\hat{y}_t = \mathrm{SR}_h \circ \mathrm{SR}_l^t \circ \uparrow_F \circ \mathrm{D}^t \circ \mathrm{P} \circ (\hat{x}, \mathrm{C}_x(x)),$$



(i) TEXT-TO-IMAGE MODEL

(ii) SPATIOTEMPORAL LAYERS

- (i) PSEUDO-3D CONVOLUTIONAL LAYERS
- (ii) PSEUDO-3D ATTENTION LAYERS

(iii) FRAME INTERPOLATION NETWORK

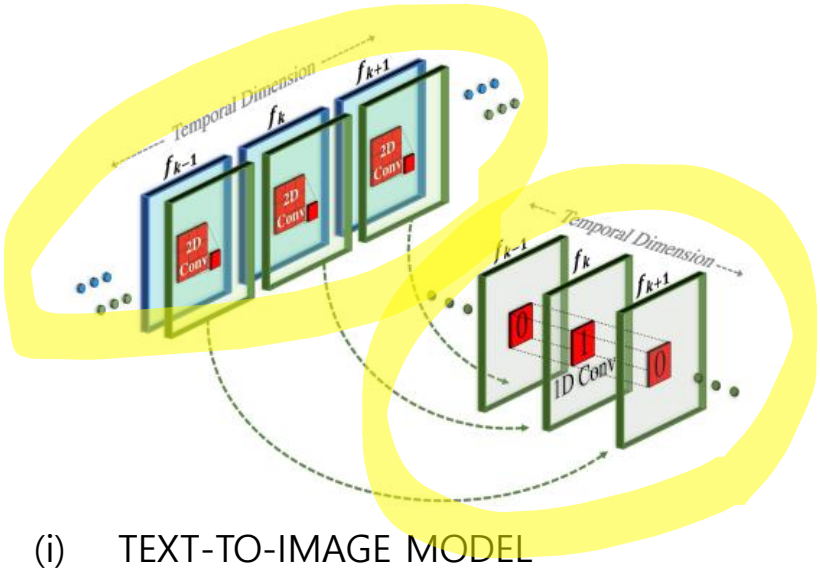
(i) A prior network P
(ii) A decoder network D
(iii) Two super-resolution networks SRl, SRh

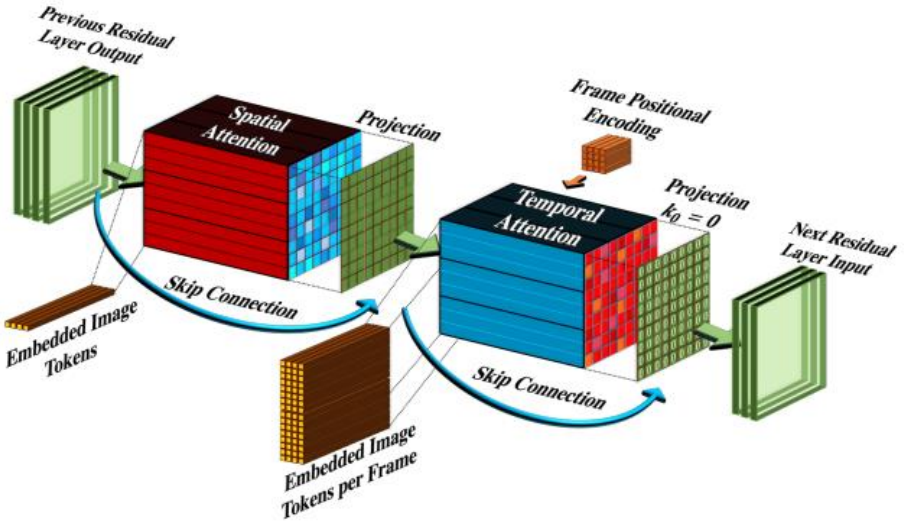
3. 적용 기술 설명

2-2. Text2Video

Given an input tensor $h \in \mathbb{R}^{B \times C \times F \times H \times W}$, where B, C, F, H, W are the batch, channels, frames, height, and width dimensions respectively, the Pseudo-3D convolutional layer is defined as:

$$Conv_{P3D}(h) := Conv_{1D}(Conv_{2D}(h) \circ T) \circ T, \quad (2)$$





(i) TEXT-TO-IMAGE MODEL

(ii) SPATIOTEMPORAL LAYERS

- (i) PSEUDO-3D CONVOLUTIONAL LAYERS
- (ii) PSEUDO-3D ATTENTION LAYERS

(iii) FRAME INTERPOLATION NETWORK

1. 공간 차원 데이터로
2D conv layer 구성

2. 공간 차원 데이터를 바탕으로
시간 차원을 1D conv layer 로 구성

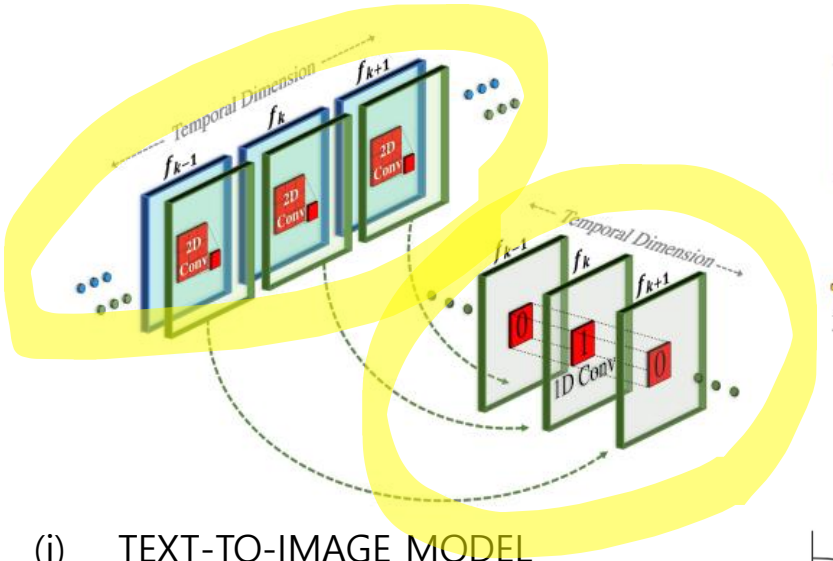
3. PSEUDO-3D CONVOLUTIONAL LAYERS

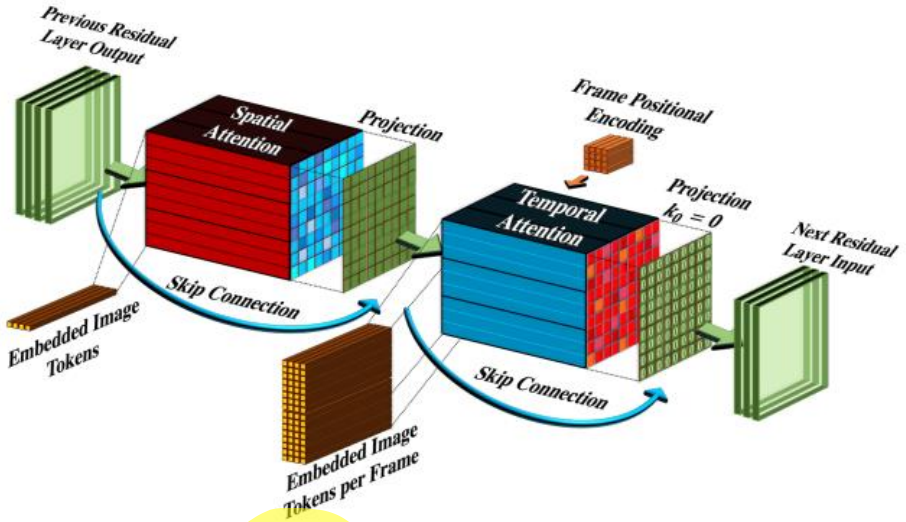
3. 적용 기술 설명

2-2. Text2Video

Given an input tensor $h \in \mathbb{R}^{B \times C \times F \times H \times W}$, where B, C, F, H, W are the batch, channels, frames, height, and width dimensions respectively, the Pseudo-3D convolutional layer is defined as:

$$Conv_{P3D}(h) := Conv_{1D}(Conv_{2D}(h) \circ T) \circ T, \quad (2)$$





(i) TEXT-TO-IMAGE MODEL

(ii) SPATIOTEMPORAL LAYERS

(i) PSEUDO-3D CONVOLUTIONAL L

(ii) PSEUDO-3D ATTENTION LAYERS

(iii) FRAME INTERPOLATION NETWORK

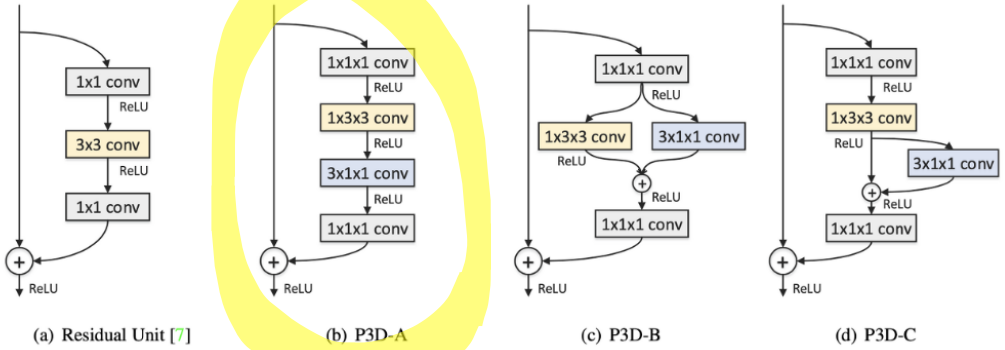


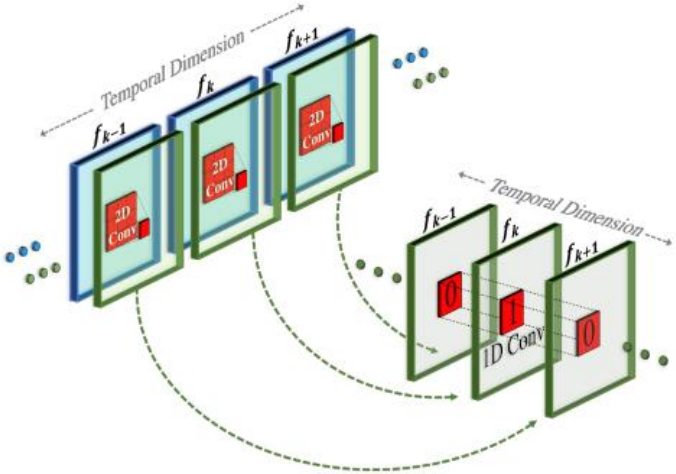
Figure 3. Bottleneck building blocks of Residual Unit and our Pseudo-3D.

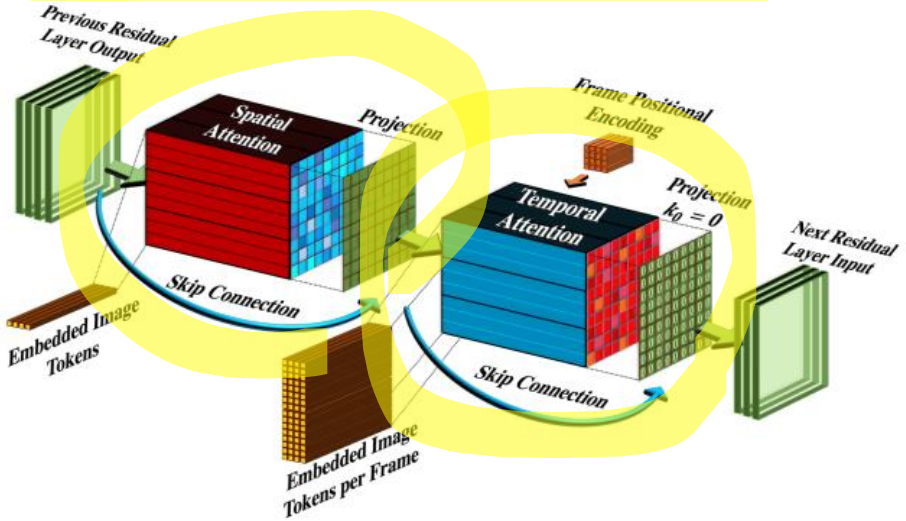
3. 적용 기술 설명

2-2. Text2Video

flattens the spatial dimension into $h' \in R^{B \times C \times F \times HW}$. $unflatten$ is defined as the inverse matrix operator. The Pseudo-3D attention layer therefore is therefore defined as:

$$ATTN_{P3D}(h) = unflatten(ATTN_{1D}(ATTN_{2D}(flatten(h)) \circ T) \circ T). \quad (3)$$





(i) TEXT-TO-IMAGE MODEL

(ii) SPATIOTEMPORAL LAYERS

- (i) PSEUDO-3D CONVOLUTIONAL LAYERS
- (ii) PSEUDO-3D ATTENTION LAYERS

(iii) FRAME INTERPOLATION NETWORK

1. 공간 차원 데이터로
2D attn layer 구성

2. 공간 차원 데이터를 바탕으로
시간 차원을 1D attn layer 로 구성

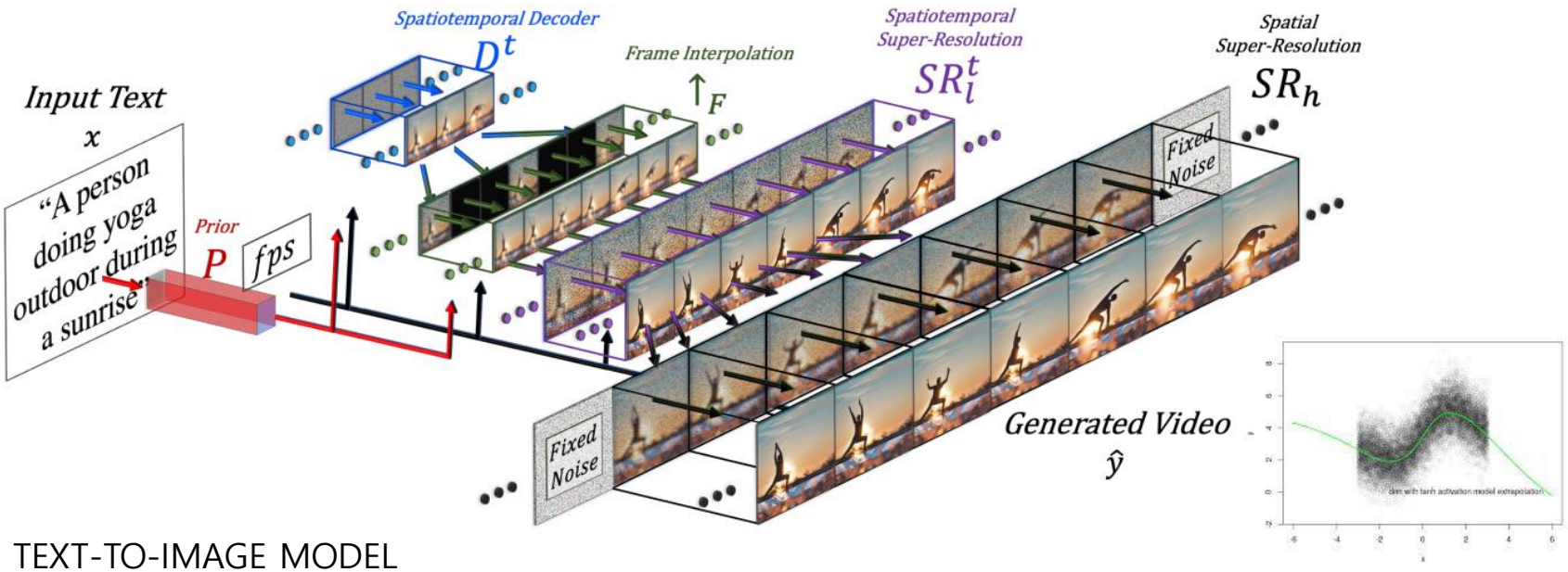
3. Feature, Text Information 추가

4. PSEUDO-3D ATTENTION LAYERS

3. 적용 기술 설명

2-2. Text2Video

$$\hat{y}_t = \mathrm{SR}_h \circ \mathrm{SR}_l^t \circ \uparrow_F \circ \mathrm{D}^t \circ \mathrm{P} \circ (\hat{x}, \mathrm{C}_x(x)),$$



(i) TEXT-TO-IMAGE MODEL

(ii) SPATIOTEMPORAL LAYERS

(i) PSEUDO-3D CONVOLUTIONAL LAYERS

(ii) PSEUDO-3D ATTENTION LAYERS

(iii) FRAME INTERPOLATION NETWORK

- Interpolation / Extrapolation Network 를 바탕으로 학습
- Zero-Padding 된 Data 에 대하여 Interpolation Task 학습

Thank you!

MILab Undergraduate student, Kim Taehyeon

2023. 08. 17

Reference

- https://biz.chosun.com/it-science/ict/2023/06/26/KJGNAQIFTBCELAUEOYFXWJUTX4/
- https://www.2e.co.kr/news/articleView.html?idxno=302661
- https://m.mk.co.kr/luxmen/view.php?sc=42600117&year=2023&no=469376
- http://www.2e.co.kr/news/articleView.html?idxno=302651
- https://economist.co.kr/article/view/ecn202307130052
- https://www.aitimes.com/news/articleView.html?idxno=135460
- https://learnopencv.com/mastering-dall-e-2/
- https://medium.com/@turc.raluca/fine-tuning-dall-e-mini-craiyon-to-generate-blogpost-images-32903cc7aa52
- https://makeavideo.studio/
- https://process-mining.tistory.com/182
- https://arxiv.org/pdf/2209.14792.pdf
- https://arxiv.org/pdf/2102.12092.pdf
- https://www.cs.cmu.edu/~awb/papers/IEEE2002/allthetime/node1.html
- https://dreamfusion3d.github.io/
- https://pseudo-lab.github.io/Tutorial-Book/chapters/GAN/Ch1-Introduction.html
- https://process-mining.tistory.com/161
- https://kdeon.tistory.com/58