2022 하계 세미나

Recent trends in diffusion models

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
이창헌

Outline

- Background
  - Diffusion models
- Papers
  - Palette
  - CDM
  - Text-conditional diffusion models
    - GLIDE
    - DALL-E2
    - Imagen
  - RePaint
- Real-life application

Background

- DDPM diffusion models[1]
  - Forward process
    - GT에 점진적으로 gaussian noise를 더해 나가 T time step안에 GT image x_0를 white Gaussian noise(random noise) $x_T \sim N(0,1)$로 만드는 것이 목표임

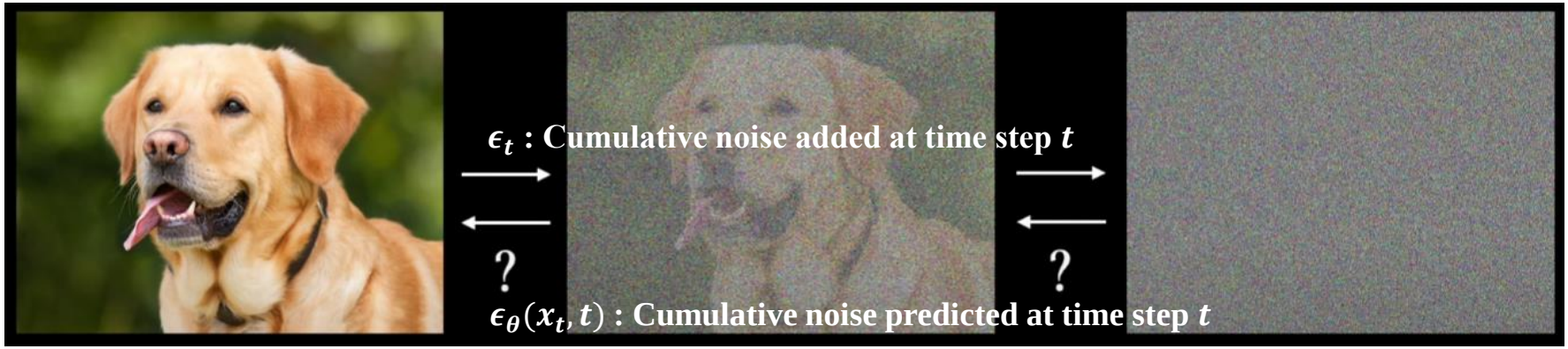
$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t} * I, \qquad I \sim N(0,1)$$

x_0 : 원본 이미지
x_t : noised 이미지
β_t : noise level

  - Reverse process
    - 현재 time step t의 noised image x_t 에 더해진 cumulative noise ϵ를 neural network를 사용해서 $\epsilon_\theta(x_t, t)$ 으로 예측함

$$x_{t-1} = N(\mu_\theta(x_t, t), \sum_\theta(x_t, t))$$

T번 반복 → $\hat{x}_0$ 생성 가능



x_0 : Ground truth　　x_t : Noised image at time step t　　x_T : White Gaussian noise

Background

- DDPM diffusion models[1]
  - Forward process

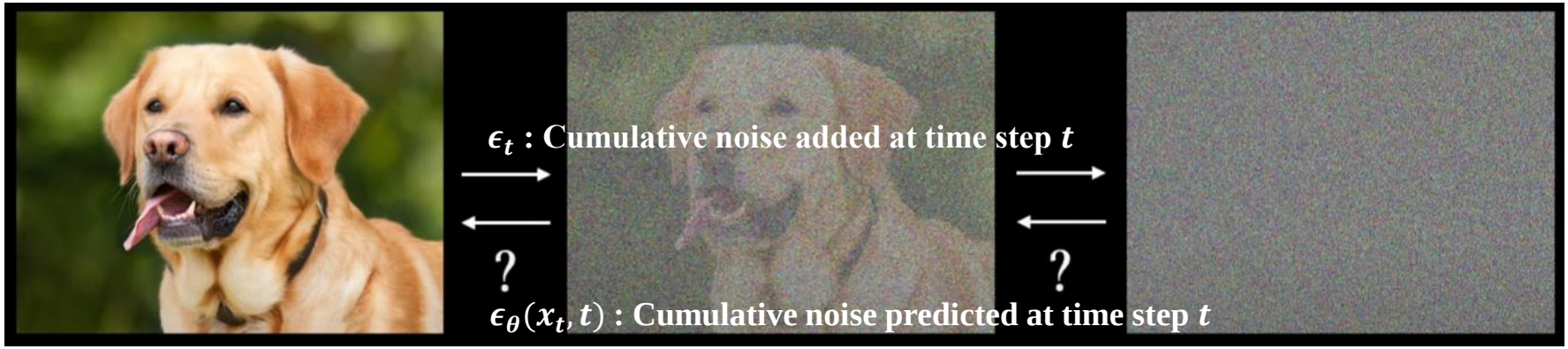
Markov property로 인해서 어떤 time step이던 x_0로부터 cumulative noise를 구할 수 있음

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} * I, \qquad I \sim N(0,1)$$

  - Reverse process
    - x_t 에 더해진 cumulative noise ϵ_0를 $\epsilon_\theta(x_t, t)$ 으로 예측함
  - Loss
    - Time step t에 대한 실제 cumulative noise와 prediction에 L2 loss function 사용

$$L_t = \|\epsilon_t - \epsilon_\theta(x_t, t)\|^2$$

ϵ_t : 실제 cumulative noise
ϵ_θ: Predicted cumulative noise
x_t : noised 이미지
t : time step



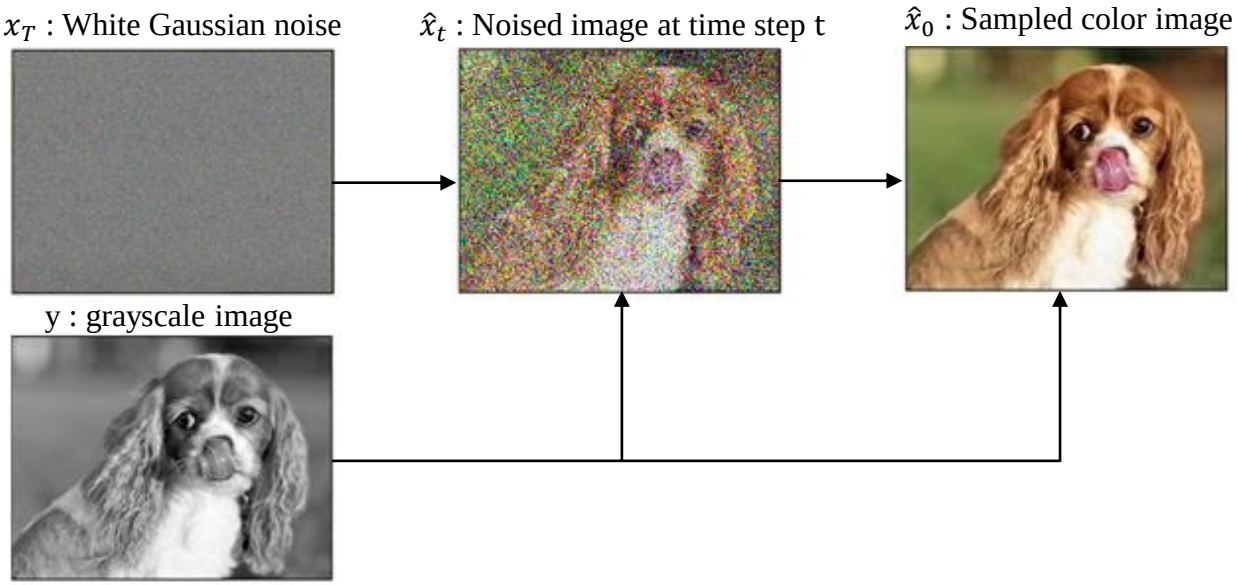
x_0 : Ground truth x_t : Noised image at time step t x_T : White Gaussian noise

Papers

- Palette[1]
  - Image-to-image translation에 범용적으로 적용 가능한 conditional diffusion model을 제안함
    - Palette는 4가지의 task에서 기존 SOTA를 능가하는 성능을 발휘함

4개의 task : colorization, inpainting, uncropping, JPEG restoration

  - Conditional diffusion models*
    - 예) Colorization task에서 train, inference 방법
      - Training : Neural network $\epsilon_\theta(x_t, y, t)$를 L2 loss로 학습함
      - Inference : $\hat{x}_{t-1} = N(\mu_\theta(\hat{x}_t, y, t), \sum_\theta(\hat{x}_t, y, t))$ 를 t번 반복해서 $\hat{x}_0$를 생성함



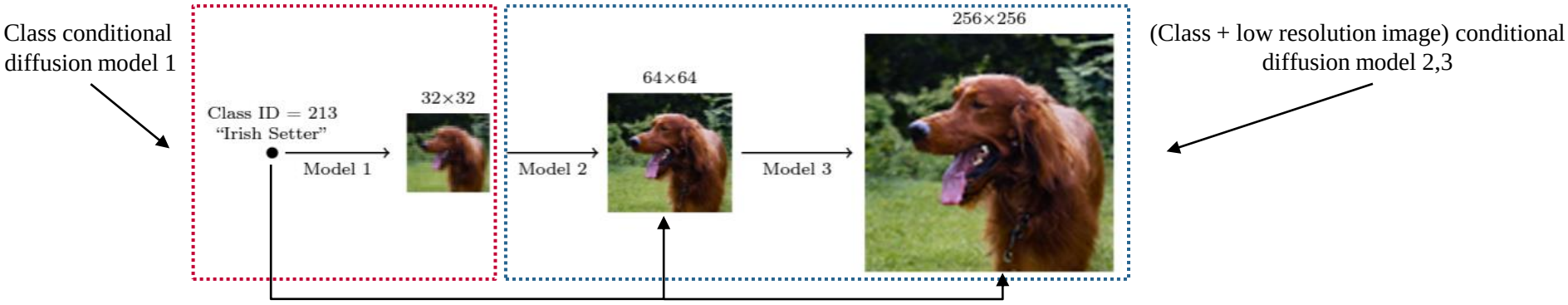
y : grayscale image

x_0 : color image

x_t : noised color image

Papers

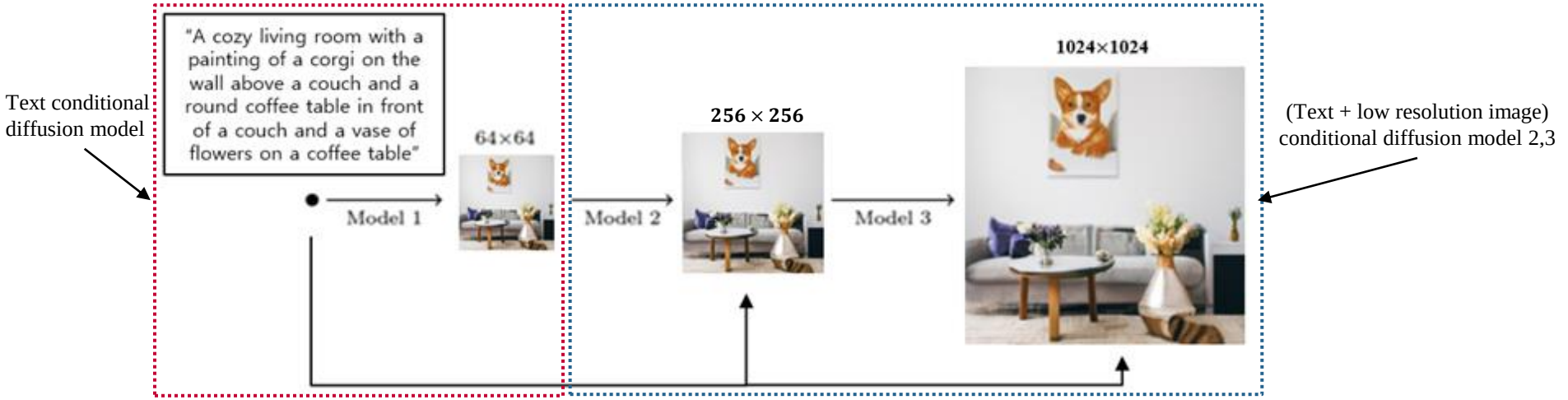
- CDM – Cascaded Diffusion Models for High Fidelity Image Generation[1]
  - Cascaded framework*
    - Diffusion model 뒤에 한 개 이상의 super-resolution diffusion model을 적용하는 cascaded 구조를 제안함
      - 생성되는 영상의 해상도를 점차 키워 나가는 방식의 framework
  - Class-conditional ImageNet generation task에서 기존 GAN, VAE 기반의 방법을 능가하는 SOTA 성능 달성함
    - Class conditional diffusion model : $\epsilon_\theta(x_t, c, t)$
    - SR diffusion model : $\epsilon_\theta(x_t, c, z, t)$



<Cascaded framework를 사용한 데이터 생성 구조>

Papers

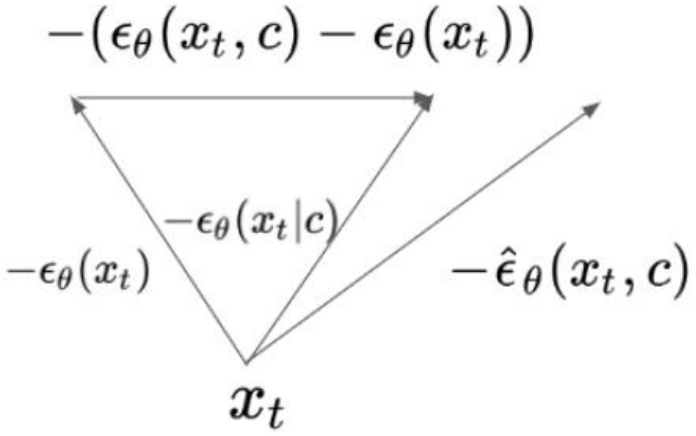
- GLIDE[1]
  - Text Conditional diffusion models[1]
    - GLIDE는 classifier-free guidance 방법으로 text를 condition으로 사용함



<Cascaded framework를 사용한 GLIDE의 데이터 생성 구조>

Papers

- GLIDE[1]
  - Scalable conditional diffusion model을 제안함*
    - Unconditional diffusion models – Baseline DDPM
      - $\epsilon_\theta(x_t, t)$
    - Conditional diffusion models – Palette, CDM
      - $\epsilon_\theta(x_t, c, t)$
    - Scalable conditional diffusion model - GLIDE
      - $\hat{\epsilon}_\theta(x_t, c, t) = \epsilon_\theta(x_t, t) + s(\epsilon_\theta(x_t, c, t) - \epsilon_\theta(x_t, t))$



<Guidance scale을 사용한 scalable conditioning 구조>

✓하나의 모델로 unconditional $\epsilon_\theta(x_t, c, t)$, unconditional $\epsilon_\theta(x_t, t)$ 모두 학습함
- 학습 시에 일정 확률로 null label을 condition으로 택하는 학습 기법 사용

✓Guidance scale s로 diversity, fidelity trade-off를 조절할 수 있음
- 높은 guidance scale에서는 reverse diffusion process에서 발생하는 diversity가 줄어드나, prompt와 유사도가 높아지며 fidelity가 상승함

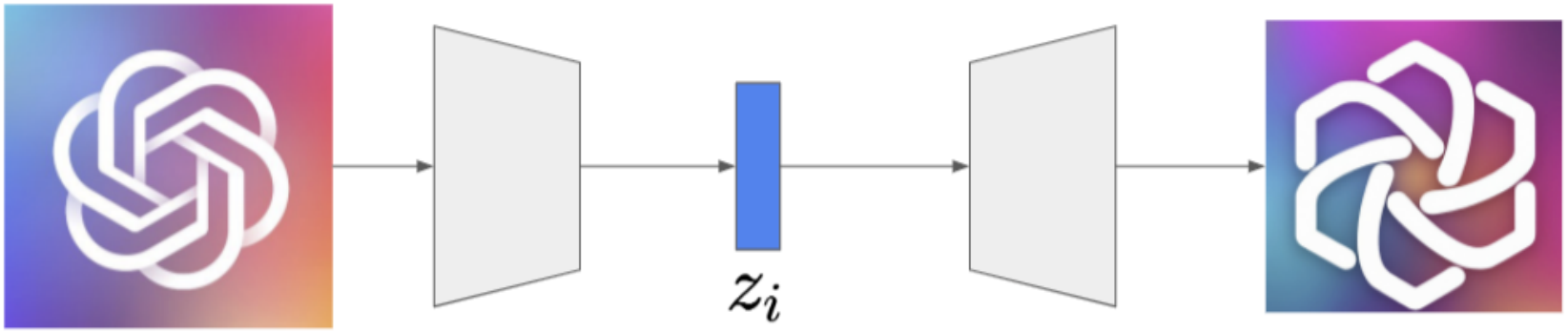
ϵ_θ : diffusion model
c : text embedding
x_t : noised data
s : guidance scale 상수

Papers

- DALL-E2[1]
  - Hierarchical conditional diffusion model을 제안함
    - GLIDE - direct text conditioning

      $\hat{\epsilon}_\theta(x_t, c, t) = \epsilon_\theta(x_t, t) + s\big(\epsilon_\theta(x_t, c, t) - \epsilon_\theta(x_t, t)\big)$, c : text
    - DALL-E2 - Hierarchical text conditioning

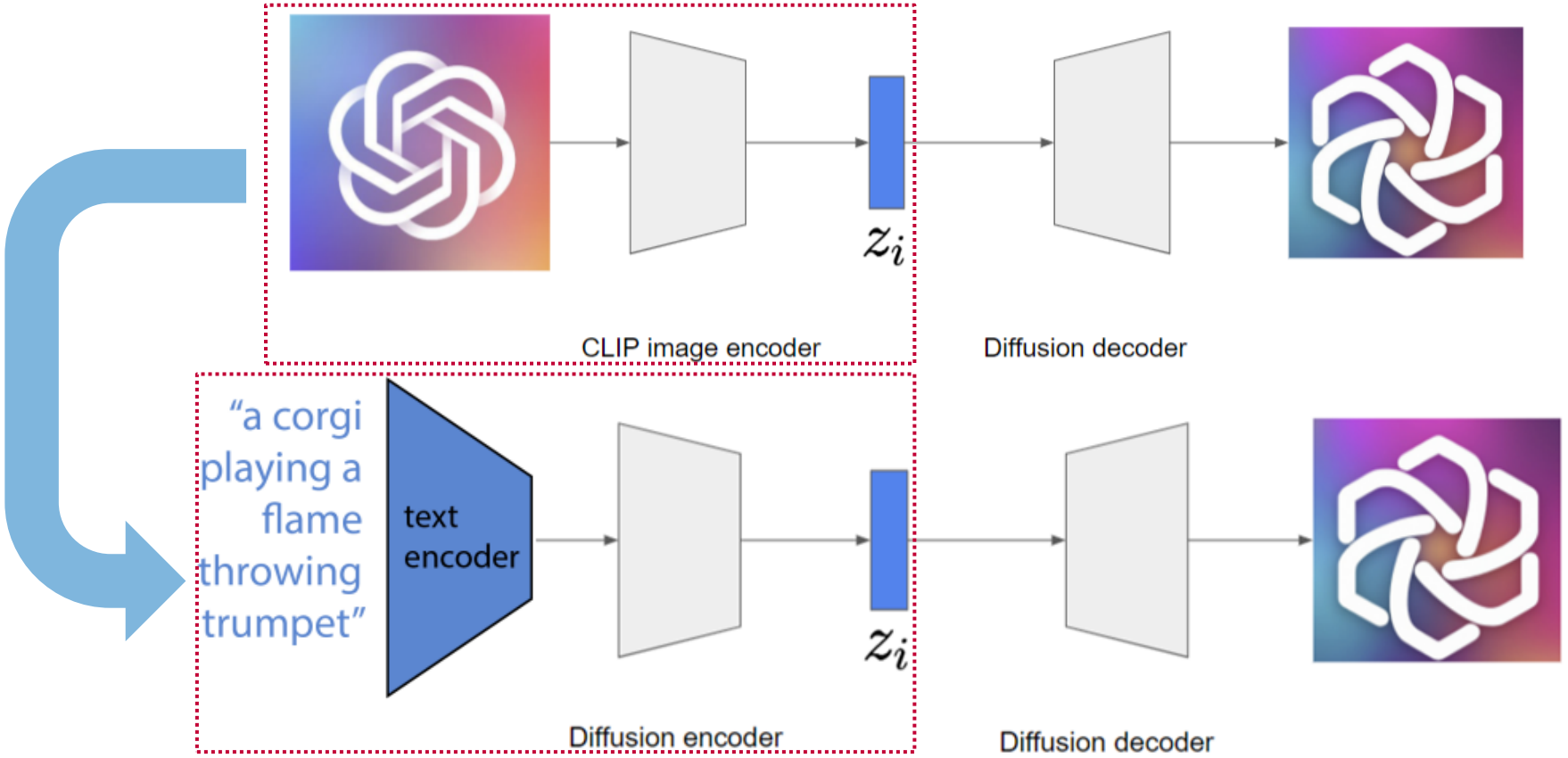
      $\hat{\epsilon}_\theta(x_t, z_i, t) = \epsilon_\theta(x_t, t) + s\big(\epsilon_\theta(x_t, z_i, t) - \epsilon_\theta(x_t, t)\big)$, z_i : CLIP image embedding
  - Decoder diffusion model
    - Decoder는 GLIDE와 완전히 동일한 diffusion model을 사용함



<Decoder diffusion model의 구조>

Papers

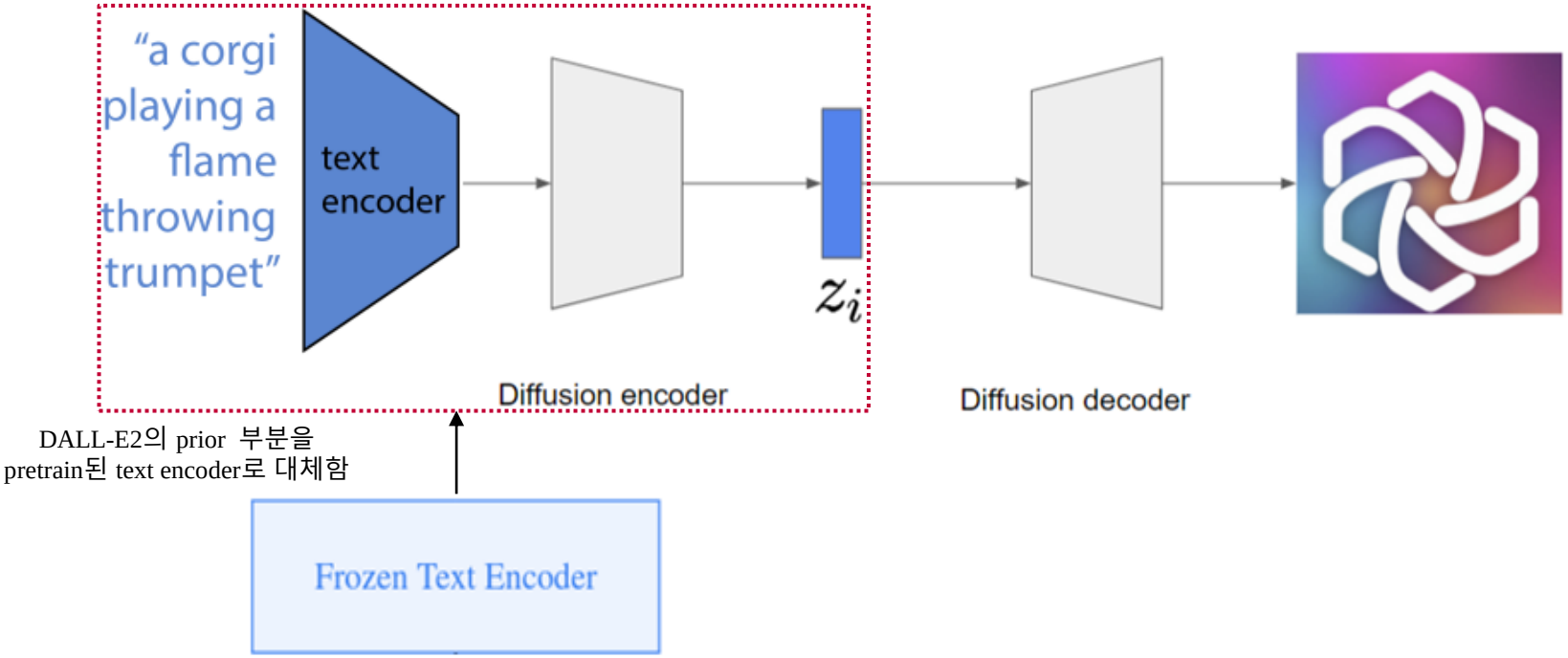
- DALL-E2[1]
  - Prior(encoder) diffusion model
    - 주어진 text에 해당하는 CLIP image embedding을 만드는 diffusion model



<실제 DALL-E2의 구조>

Papers

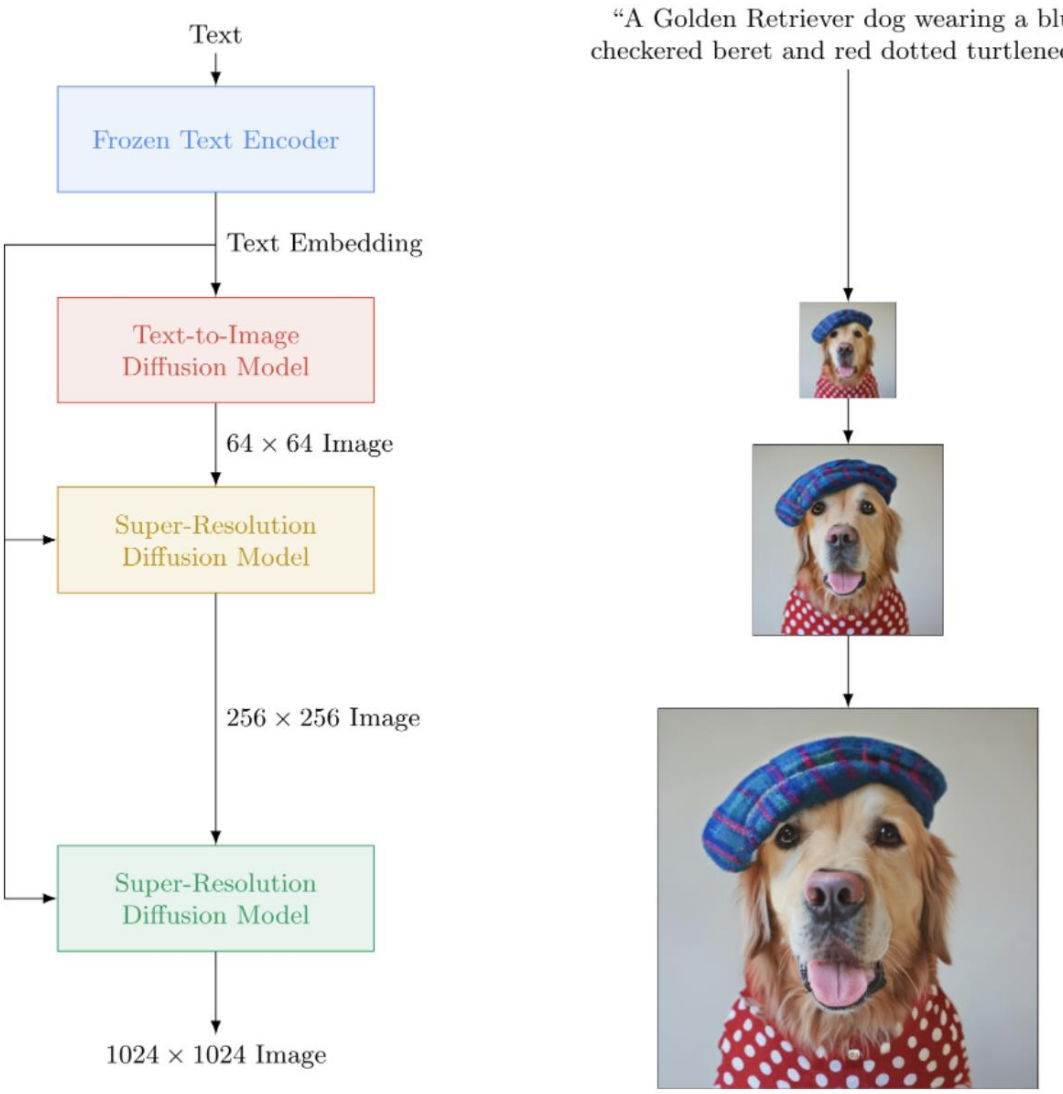
- Imagen[1]
  - DALL-E2보다 데이터 생성 프레임워크를 단순화하면서도 더 높은 성능 달성
    - Text dataset으로 pretrain된 language model을 prompt의 text encoder로 사용함



<DALLE-2, Imagen 구조 비교>

Papers

- Imagen[1]



<실제 Imagen의 구조>

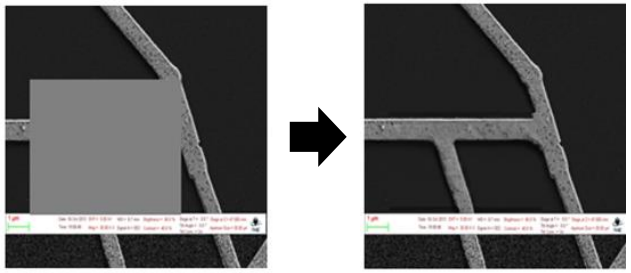
Papers

- Imagen[1]
  - Dynamic thresholding*
    - $\hat{\epsilon}_\theta(x_t, c, t) = \epsilon_\theta(x_t, t) + s(\epsilon_\theta(x_t, c, t) - \epsilon_\theta(x_t, t))$
      - Condition을 적용하기 위해 $s(\epsilon_\theta(x_t, c, t) - \epsilon_\theta(x_t, t))$를 더하면서 training data의 bound인 range[-1,1]를 벗어남
        - ✓$\hat{x}_t$가 range[-1,1]의 bound를 벗어날 경우 train-test mismatch으로 인해서 생성 이미지의 품질이 떨어짐
        - ✓따라서 range[-1,1]로 normalization, clipping하는 과정이 반드시 필요함

<Threshold 사용 유무에 따른 이미지 품질 비교>

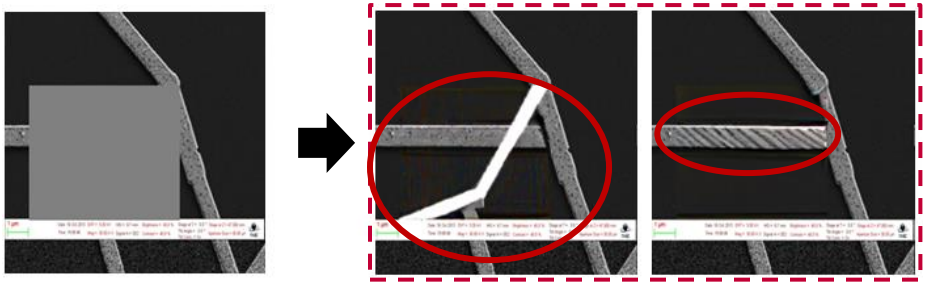
Papers

- Imagen[1]
  - Dynamic thresholding*
    - 생성되는 이미지의 품질을 높이기 위해 제안된 새로운 sampling 기법
    - Artifact를 방지하면서 높은 guidance scale 값을 사용해서 train data와 동떨어진 data를 생성 가능함
  - Static thresholding*
    - 기존 GLIDE, DALL-E2의 sampling 기법
    - Diffusion model의 reverse step마다 생성 이미지 x_{t-1}를 range$[-1, 1]$으로 clipping함
    - Guidance scale을 큰 값으로 설정해서 생성되는 결함의 강도를 높이면 sampling되는 영상에 artifact가 생성됨



< Guidance scale = 4, Static thresholding 적용 >

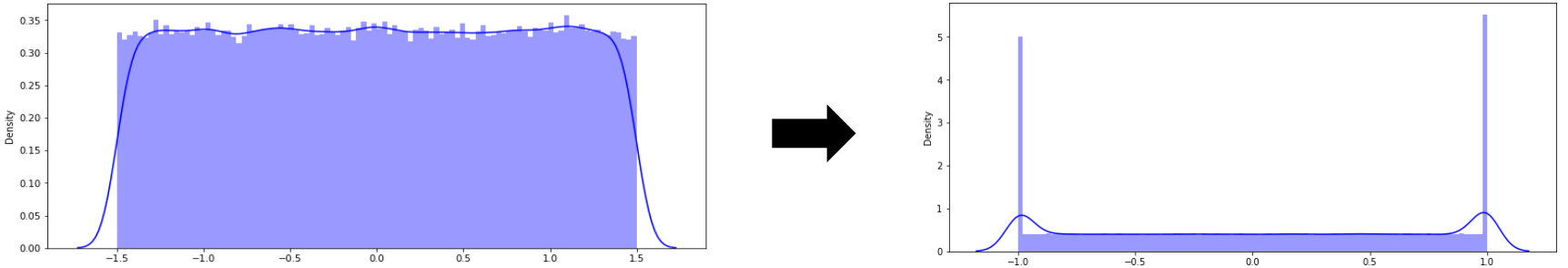
Artifact 발생



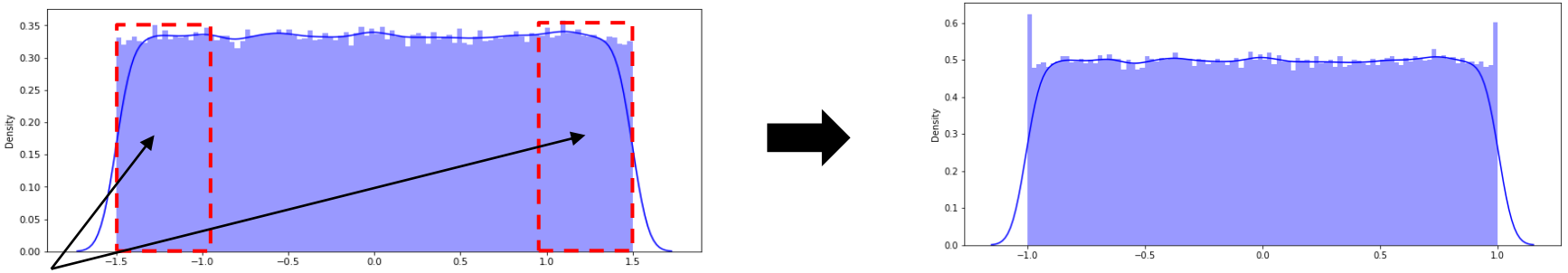
< Guidance scale = 100, Static thresholding 적용 >

Papers

- Imagen[1]
  - Dynamic thresholding*
    - x_t의 값을 순서대로 나열하여 s=0.995보다 큰 값을 s로 clipping 후 s로 나눔
      - ∴ 0.995~1.000사이의 값은 0.995로 clipping



(a) Noise data x_t <Static thresholding 사용 예시> (b) Noise data x_t + guidance weight

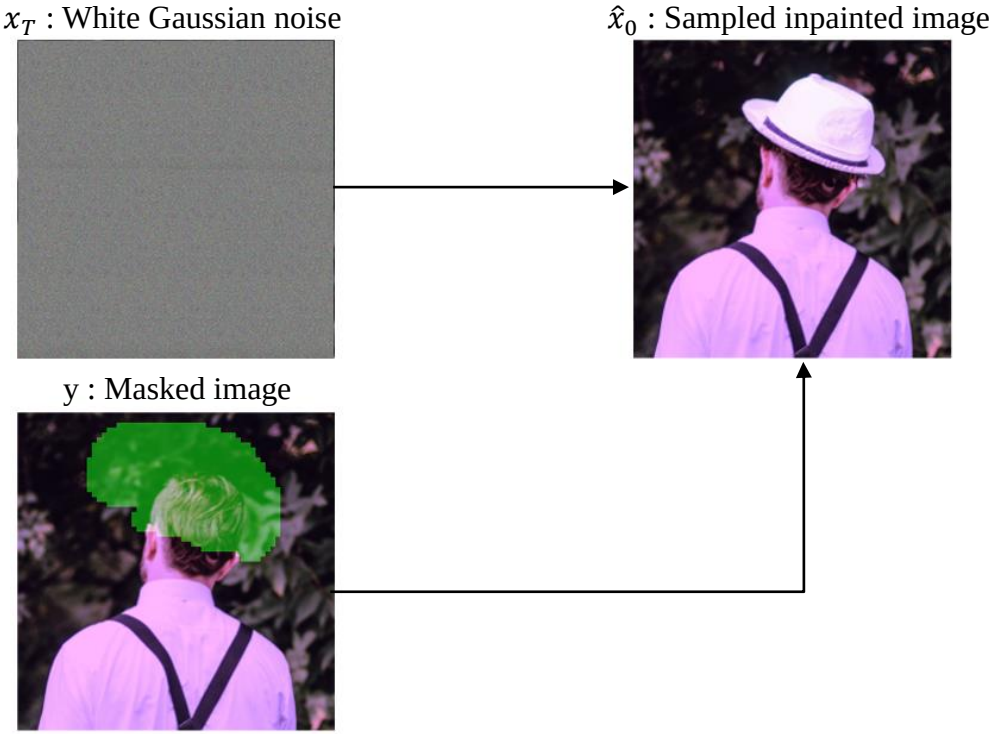


기존 $\hat{x}_t$에서 -0.995보다 작은 값은 -1로
0.995보다 큰 값은 1로 clipping되며 나머지
값들은 0.995로 나눠지며 안으로 향하게 됨

<Dynamic thresholding 사용 예시>

Papers

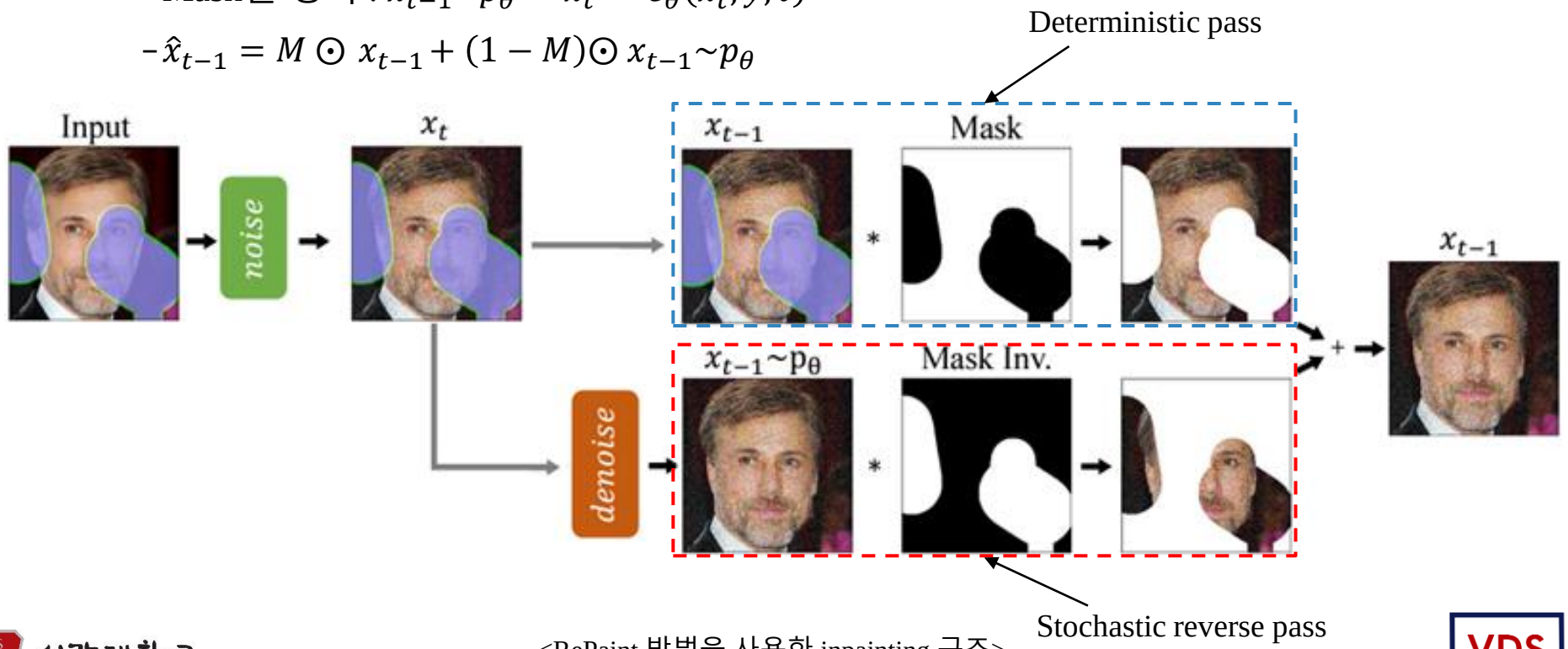
- RePaint[1]
  - Inpainting과 관계없이 학습된 DDPM 모델을 사용해서 reverse diffusion process 단계에서 inpainting을 수행하는 방법을 제안함
    - 일반적인 inpainting 모델 : Masked 이미지를 input으로 받아서 mask 영역에 GT와 유사한 부분을 생성하는 모델을 학습함



<Palette의 inpainting 구조>

Papers

- RePaint[1]
  - Inpainting과 관계없이 학습된 DDPM 모델을 사용해서 reverse diffusion process 단계에서 inpainting을 수행하는 방법을 제안함
    - Mask되지 않은 영역 : $x_{t-1} = \sqrt{1-\beta_{t-1}}x_t + \sqrt{\beta_{t-1}} * I$
    - Mask된 영역 : $x_{t-1} \sim p_\theta = x_t - \epsilon_\theta(x_t, y, t)$
    - $\hat{x}_{t-1} = M \odot x_{t-1} + (1-M) \odot x_{t-1} \sim p_\theta$



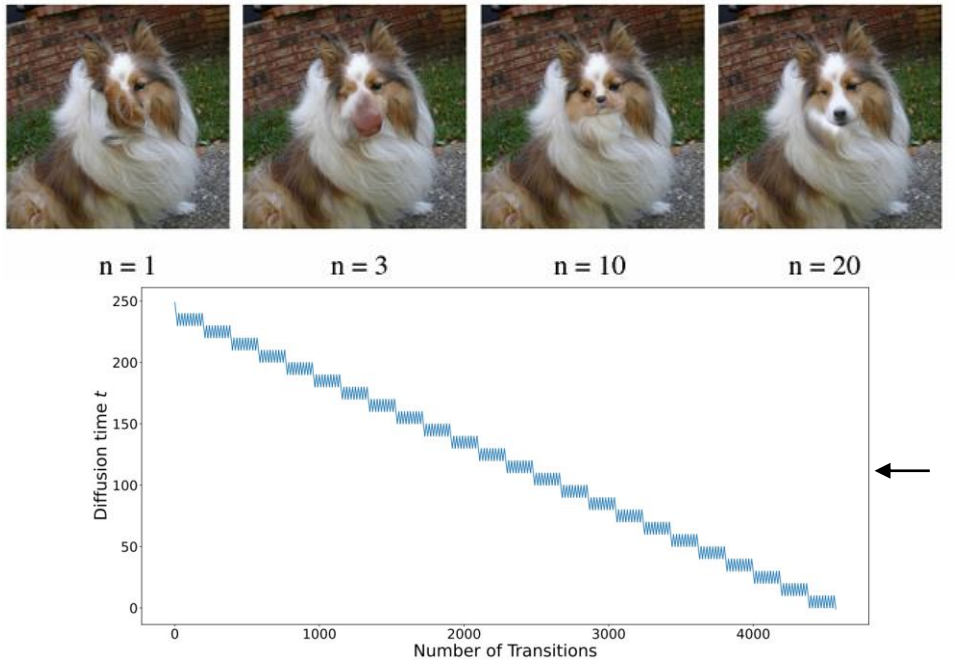
<RePaint 방법을 사용한 inpainting 구조>

Papers

- RePaint[1]
  - Resampling*
    - Reverse process에서 harmonization을 수행할 수 있는 sampling 기법 제안
    - Mask 영역과 mask inverse 영역을 합치는 과정에서 content가 일치하지 않는 문제가 발생
      - RePaint는 reverse pass에 forward pass를 섞어주는 resampling 기법으로 이를 해결함
        - ✓일반적인 diffusion model은 랜덤 노이즈 x_t부터 생성 데이터 x_0까지 reverse diffusion process를 반복함
        - ✓Resampling 기법은 reverse pass로 $x_t \rightarrow x_{t-1}$를 수행하고 다시 forward pass를 적용해서 $x_{t-1} \rightarrow x_t$를 구하는 과정을 resampling 횟수 n번만큼 반복함
          - Diffusion step = 3일 때 기존 sampling 방법은 $x_3 \rightarrow x_2 \rightarrow x_1 \rightarrow x_0$
          - Resampling 횟수 n=2일 때는 $x_3 \rightarrow x_2 \rightarrow x_3 \rightarrow x_2 \rightarrow x_1 \rightarrow x_2 \rightarrow x_1 \rightarrow x_0$

Papers

- RePaint[1]
  - Resampling*
    - 기존 diffusion model의 sampling 방법은 resampling step n = 1과 동일함
    - Resampling step n = 10이상으로 주었을 때 harmonization이 잘된 이미지가 생성됨
      - Resampling을 사용하면 inference에 필요한 diffusion step이 늘어남
        - ✓수행 속도와 이미지 품질의 trade-off가 발생



n=10, diffusion step이 250일 때,
inference에 총 4570 step이 소요됨

<RePaint 논문 실험 결과>

Real-life application

- Framework 선정하기
  - Palette

Task	Diffusion algorithm	Backbone network	Condition type	Condition term	Thresholding	Sampling process
Image-to-image translation	**DDPM**	**Unet**	Unconditional	Class label	None	**Reverse pass only**
	DDIM		**Conditional**	**Image**	**Static**	
	Elucidating DDPM	Efficient Unet	Scalable conditional	Text	Dynamic	Resampling

  - GLIDE

Task	Diffusion algorithm	Backbone network	Condition type	Condition term	Thresholding	Sampling process
Text-guided image generation	**DDPM**	**Unet**	Unconditional	Class label	None	**Reverse pass only**
	DDIM		Conditional	Image	**Static**	
	Elucidating DDPM	Efficient Unet	**Scalable conditional**	**Text**	Dynamic	Resampling

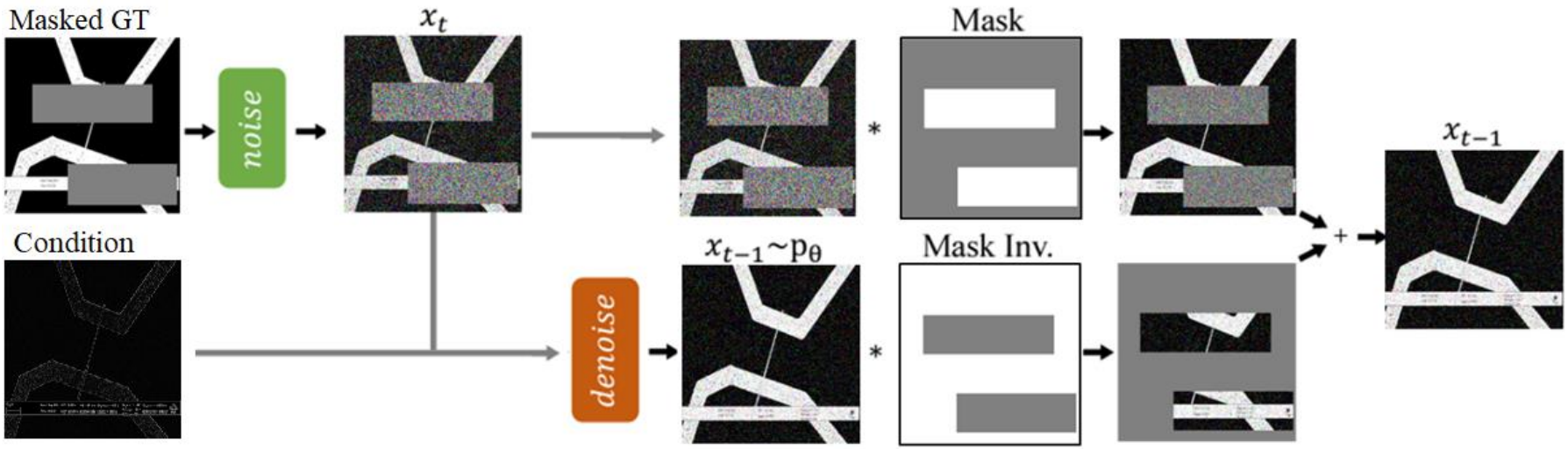
  - Imagen

Task	Diffusion algorithm	Backbone network	Condition type	Condition term	Thresholding	Sampling process
Text-guided image generation	**DDPM**	Unet	Unconditional	Class label	None	**Reverse pass only**
	DDIM		Conditional	Image	Static	
	Elucidating DDPM	**Efficient Unet**	**Scalable conditional**	**Text**	**Dynamic**	Resampling

Real-life application

- Framework 선정하기
  - Custom usage

Task	Diffusion algorithm	Backbone network	Condition type	Condition term	Thresholding	Sampling process
Defect image generation	**DDPM**	Unet	Unconditional	Class label	None	**Reverse pass only**
	DDIM		Conditional	**Image**	Static	
	Elucidating DDPM	**Efficient Unet**	**Scalable conditional**	Text	**Dynamic**	**Resampling**

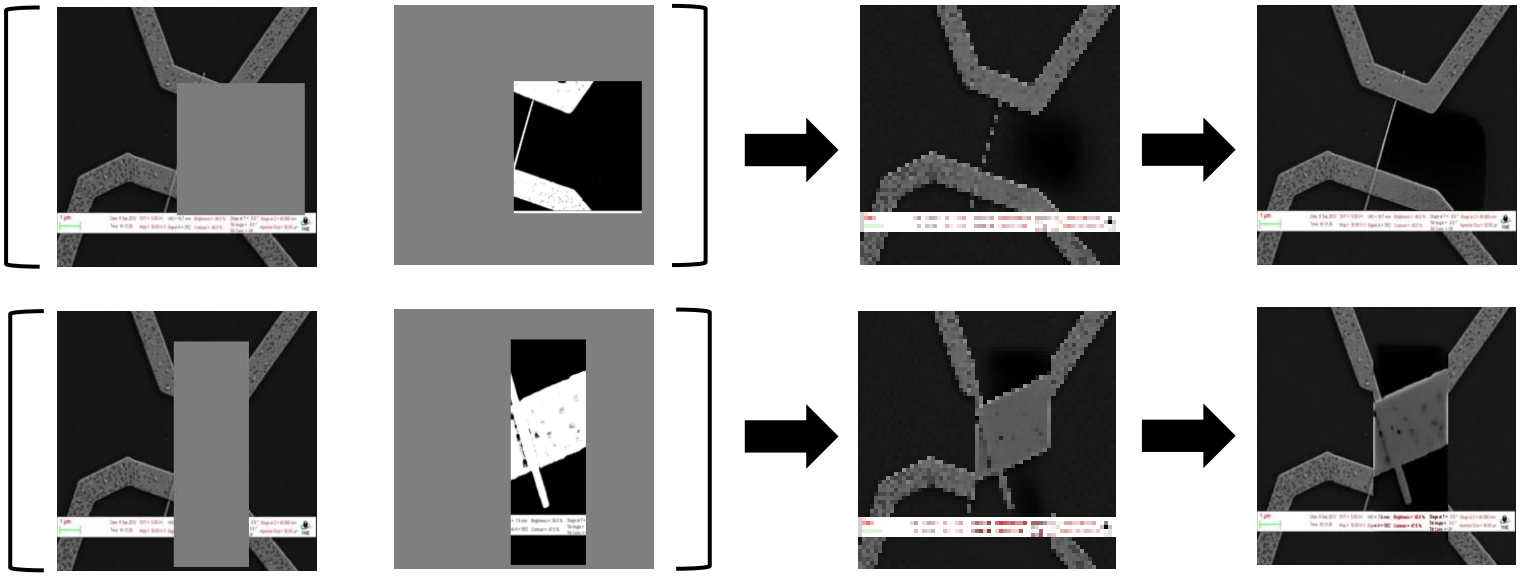


<Custom diffusion model's inpainting framework>

Real-life application

- Framework 선정하기
  - Custom usage

Task	Diffusion algorithm	Backbone network	Condition type	Condition term	Thresholding	Sampling process
Defect image generation	**DDPM**	Unet	Unconditional	Class label	None	**Reverse pass only**
	DDIM		Conditional	**Image**	Static	
	Elucidating DDPM	**Efficient Unet**	**Scalable conditional**	Text	**Dynamic**	**Resampling**

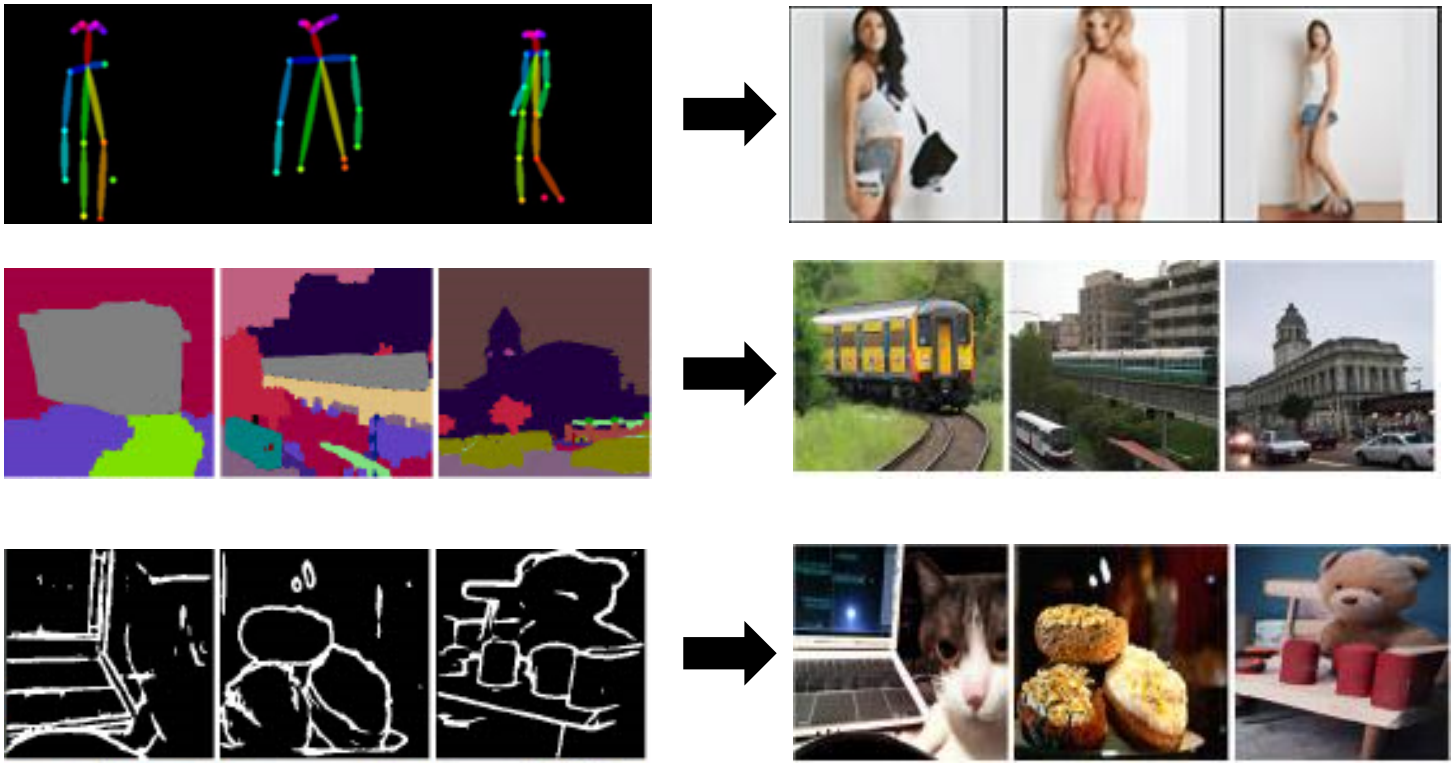


x_0 : Masked GT y : Condition image $\hat{x}_0$: Inpainted image 64×64 Inpainted image 256×256

<Custom diffusion model's 결함 데이터 생성 결과>

Real-life application

- What's next?



y : Condition 이미지

$\hat{x}_0$: 생성 이미지

Thank you! ☺