

Writing Session - eXplainable AI

생성형 모델 eXplainable AI

20212549

김채원

20192780

유광열

20212568

이서연

CONTENTS

01

연구 주제

02

연구 방향

03

진행 상황

04

연구 계획

05

연구 의의

연구 주제

Generative AI + Explainable AI (XAI)

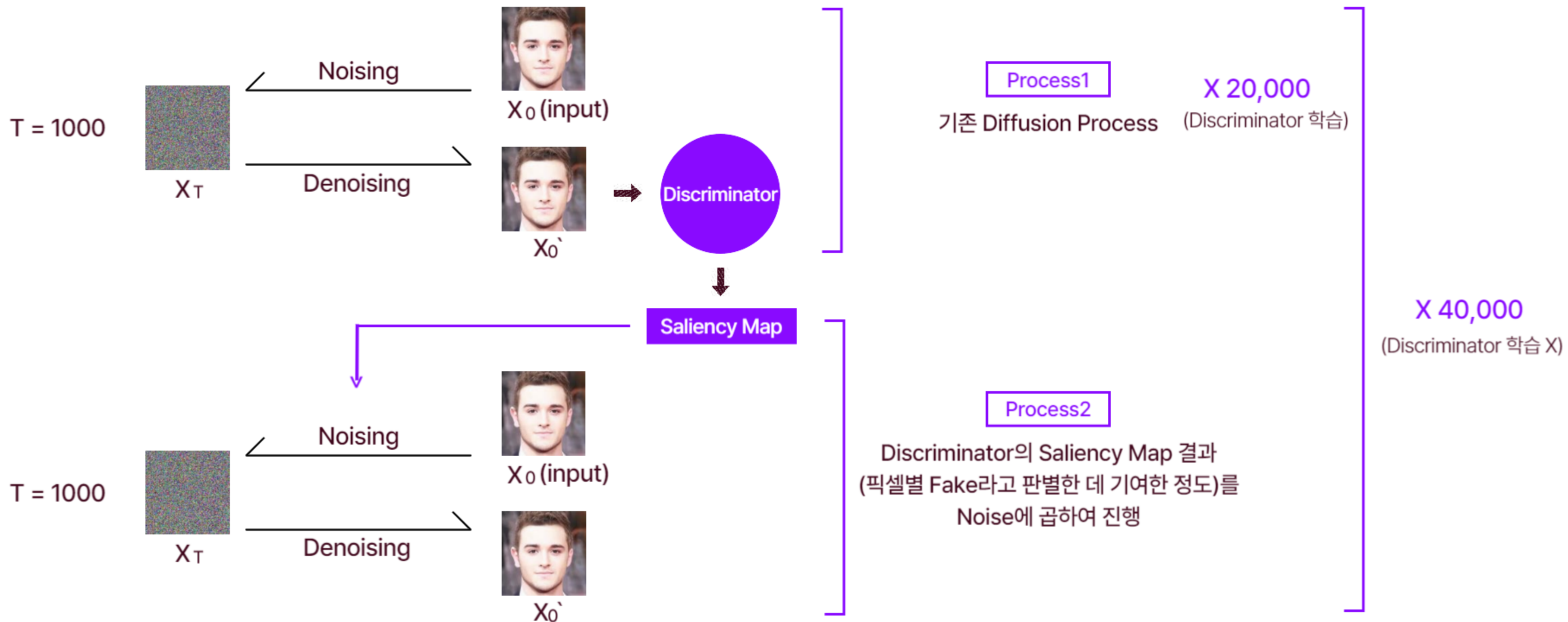
Diffusion에 XAI 기술을 적용하여 모델의 의사 결정을 이해하고 성능을 향상시킴



Discriminator를 Diffusion에 추가하고 해당 부분에 XAI 기술을 적용한 후,
이를 모델에 다시 반영하여 성능을 향상

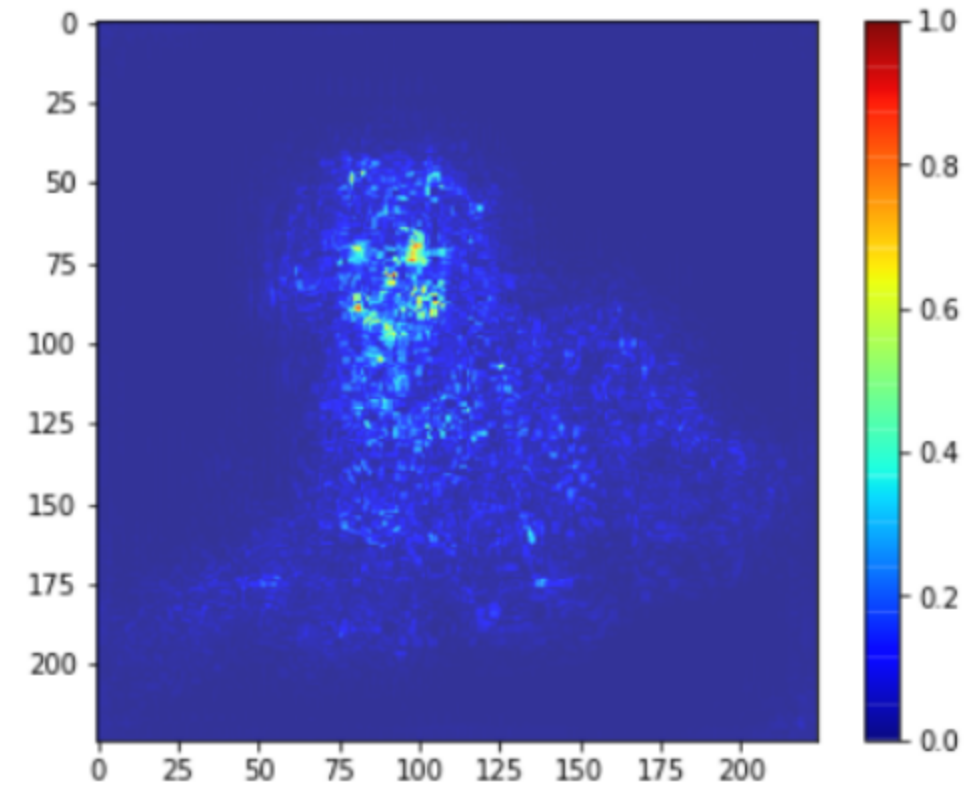
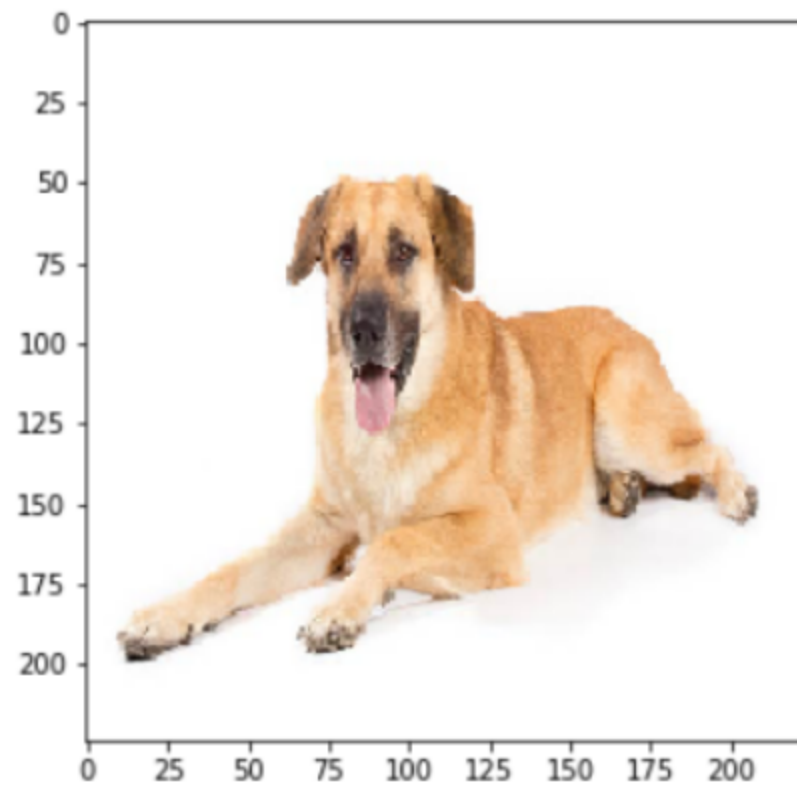
Pipeline

1 Cycle - 100,000 Process



Saliency Map?

이미지에서 어떤 부분에 집중하는지 살펴보는 방법으로,
각 Input 픽셀에 대해서 구하고자 하는 Class의 Output의 Gradient를 계산



즉 위의 사진에서 색깔이 밝을 수록, 모델이 해당 사진을 강아지라고 판단하는데 큰 도움을 줬다는 의미

Details

Process

: 한번 Noising 거치고 Denoising으로 복원하는 과정

[Process1]

기존 Diffusion Process

[Process2]

Noising 과정에서, 학습이 더 필요한 영역에 Noise를 더 많이 추가
→ 해당 부분에 집중하여 학습할 것으로 기대

Cycle

- Discriminator를 추가한 Process1로 Diffusion 모델 및 Discriminator 학습 (x 2만)
- Process1을 거친 후, 만들어진 이미지를 Discriminator에 넣어 Discriminator가 Fake라고 판단한 결과에 대한 각 픽셀의 기여도(Saliency Map)를 추출한 후, Process2에서 이미지에 Noise를 추가할 때 Noise에 해당 기여도를 곱한 값으로 추가 (x 4만)

⇒ Process1 x 2만 + (Process1 + Process2) X 4만 = 10만 Process



총 5 Cycle (500,000 Process) 진행

* 기존 DDPM 논문에서 학습한 Process 횟수와 동일하게 설정

Add XAI info to Noise : **MAKE Attention MAP**

Discriminator에서 도출된 Saliency MAP 정보를
DDPM의 Noise Layer에 순차적으로 Attention 정보로 입력

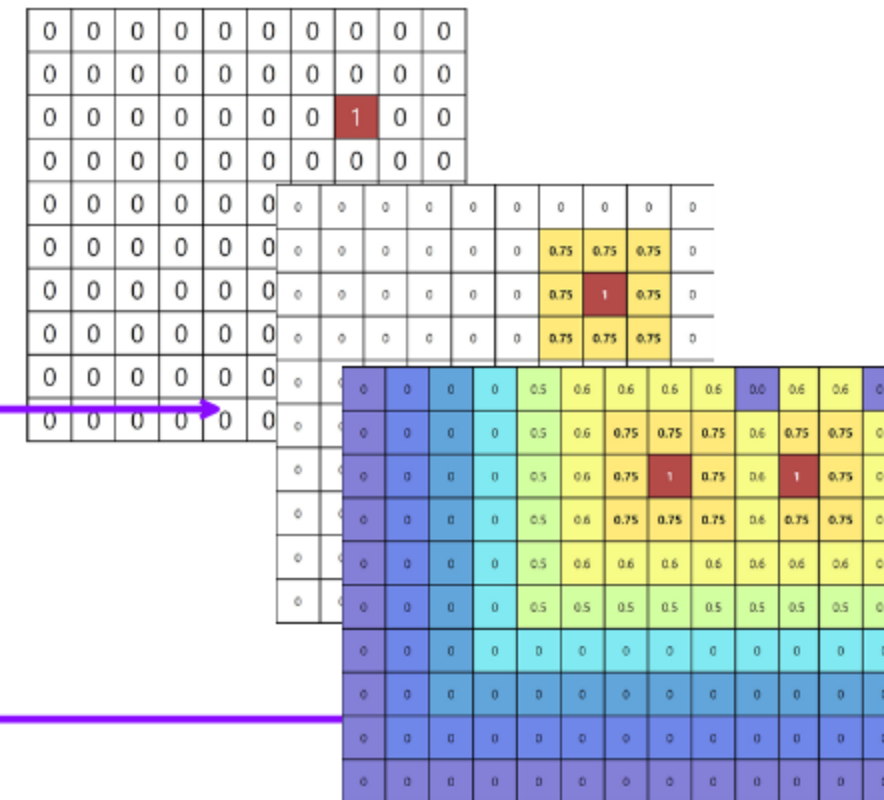
- Discriminator에서 추출된 Saliency MAP을 Hierarchical하게 변형함
- Hierarchical하게 변형 된 Saliency MAP정보를 바탕으로 Attention MAP을 생성



Discriminator Saliency MAP



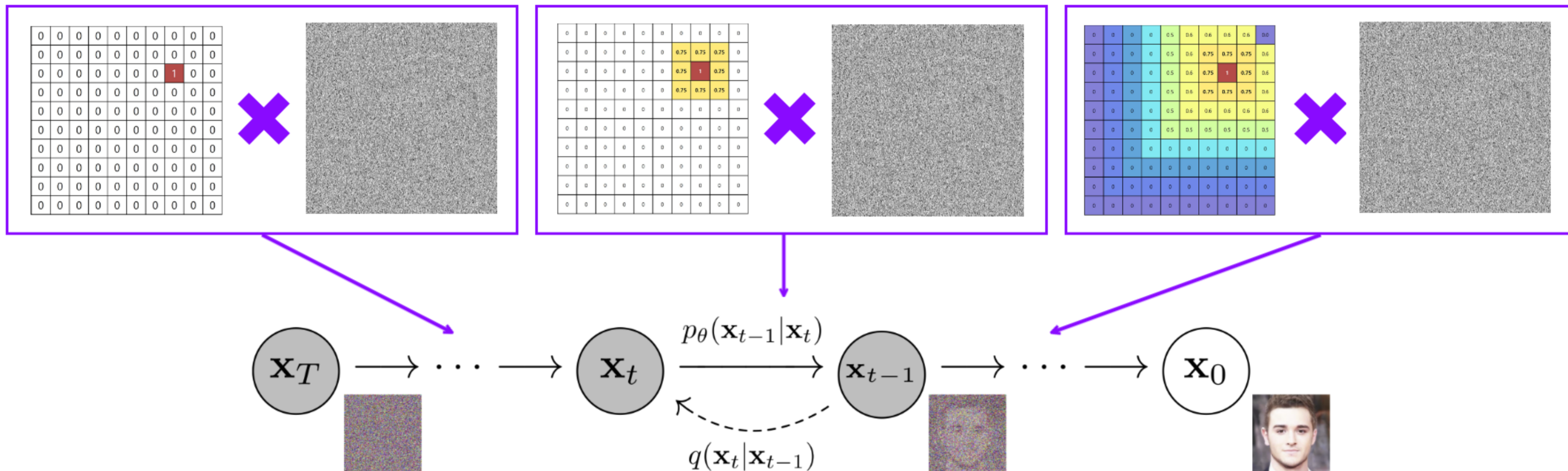
Hierarchical Saliency MAP



Make Attention MAP

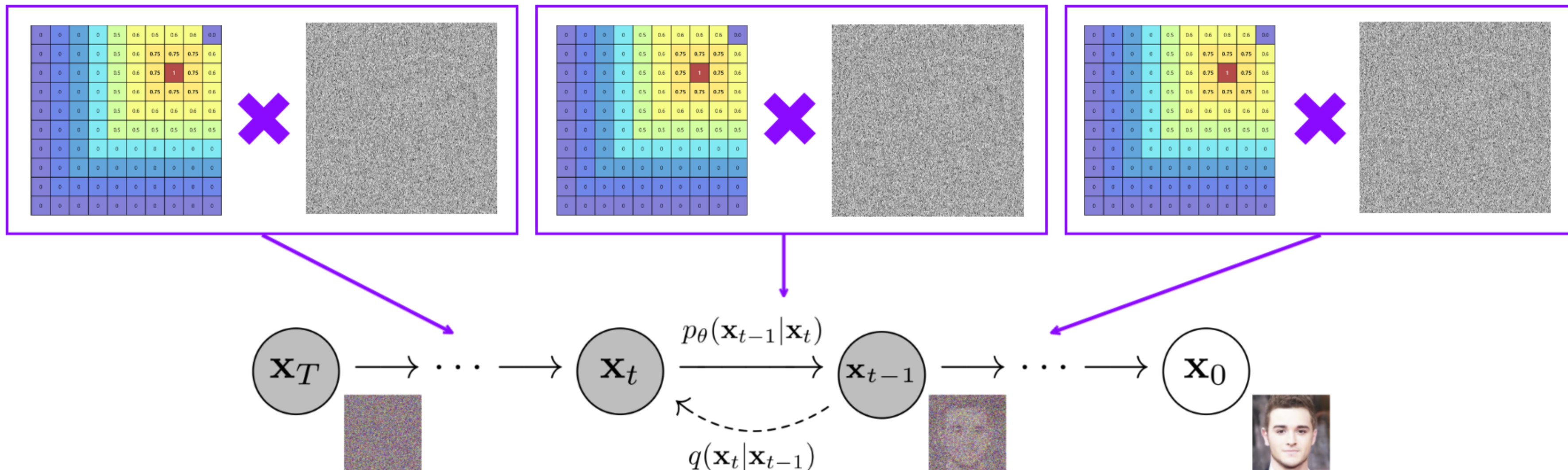
Add XAI info to Noise : Layer Hierarchical Info

Discriminator에서 도출된 Saliency MAP 정보를
DDPM의 Noise Layer에 순차적으로 Attention 정보로 입력



Add XAI info to Noise : Layer SAME Info

Discriminator에서 나온 Saliency MAP 정보를
DDPM의 Noise Layer에 같은 Attention 정보로 입력



진행 상황

DDPM 학습 진행

```
class RandomOrLearnedSinusoidalPosEmb(nn.Module):
    """ following @crowsonkb 's lead with random (learned optional) sinusoidal pos emb """
    """ https://github.com/crowsonkb/v-diffusion-jax/blob/master/diffusion/models/danbooru_128.py#L8 """

    def __init__(self, dim, is_random = False):
        super().__init__()
        assert divisible_by(dim, 2)
        half_dim = dim // 2
        self.weights = nn.Parameter(torch.randn(half_dim), requires_grad = not is_random)

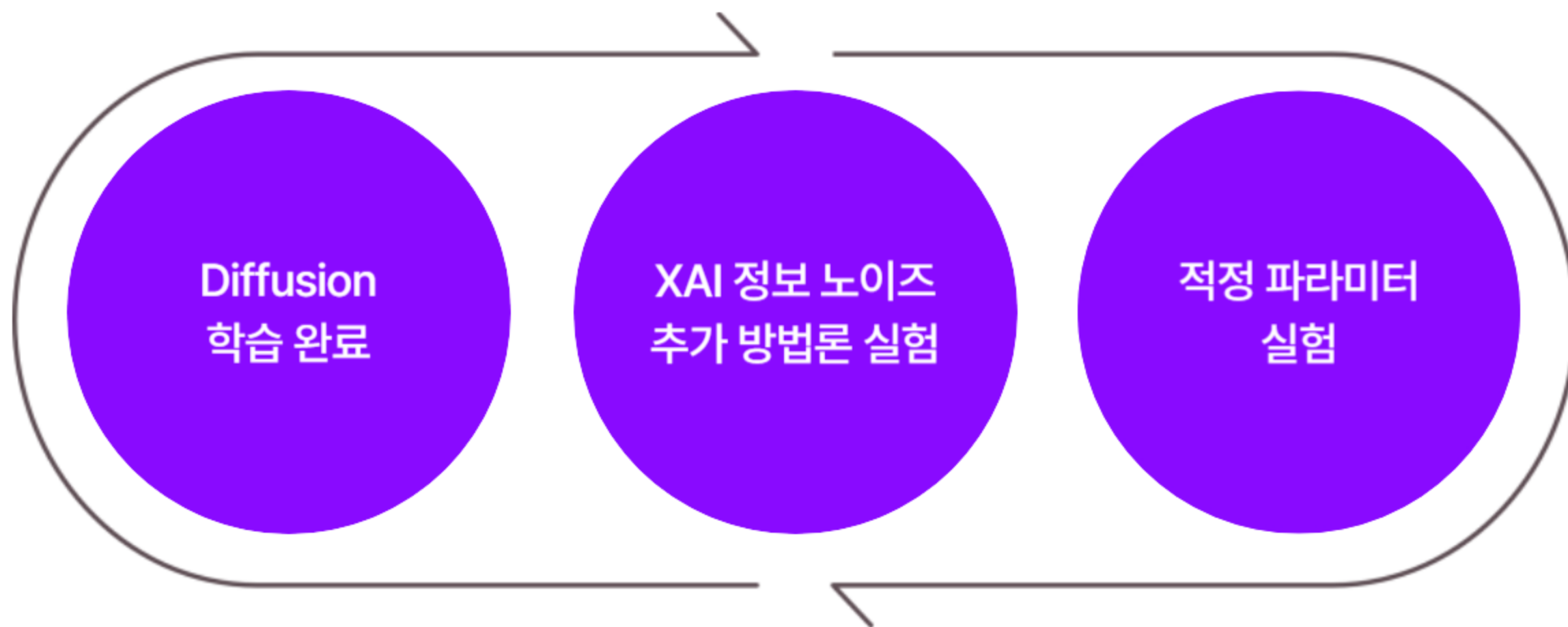
    def forward(self, x):
        x = rearrange(x, 'b -> b 1')
        freqs = x * rearrange(self.weights, 'd -> 1 d') * 2 * math.pi
        fouriered = torch.cat((freqs.sin(), freqs.cos()), dim = -1)
        fouriered = torch.cat((x, fouriered), dim = -1)
        return fouriered
```

Dataset: CelebA-HQ 256

원활한 실험 결과 비교를 위해 기존 DDPM 논문에서
사용한 데이터셋을 사용하여 학습

*현재까지 CIFAR-10, CIFAR-100에 대해서는 학습 가능 여부 확인

연구 계획



- Diffusion에 Discriminator를 적용한 후, CelebA-HQ 256으로 학습 진행
- 앞서 설명한 3가지 방법론 중 가장 효율적인 방법론으로 선택
- Cycle 및 Process 수와 같은 파라미터를 실험적으로 결정

연구 의의

Diffusion 모델에 eXplainbe AI 적용

XAI를 통해
모델을 이해할 수 있는
기회 제공 가능

- Diffusion에 Discriminator를 적용
- Discriminator에 XAI 기술을 적용하여 Saliency MAP을 통한 Generative model 모델 이해
- 모델 이해를 통한 Gerative model에 정보 전달 및 의사 결정 반영

XAI를 통해
생성 모델의
성능 향상을 기대

- Discriminator에서 추출한 Saliency map을 attentim MAP으로 변환
- 해당 attention MAP을 Hierarchical하게 Noise에 반영
- 해당 attention mAP을 모든 Noise에 반영
- 이를 통한 Generative model 성능 향상 기대

Writing Session - eXplainable AI

Q & A

2024. 02. 06