Denoising Diffusion Probabilistic Models 논문 리뷰

0. Abstract

- diffusion probabilistic model을 이용한 고품질 이미지 합성 방법 제시
  - **diffusion probabilistic model:** 비평형 열역학 개념에서 영감을 받아 개발된 잠재 변수 모델의 일종
  - Markov chain을 사용하여 점진적으로 **데이터에 노이즈를 추가**하고, 이를 역으로 진행하여 원래 데이터를 복원하는 방식으로 이미지를 생성
- **주요 기법**
  - **Variational Inference**을 사용한 학습

💡 **Variational Inference**

- 복잡한 확률 분포를 근사하는 방법 중 하나로, 주로 **베이즈 추론**을 효율적으로 수행하기 위해 사용

- 복잡한 **목표 분포**를 직접 계산하지 않고, 비교적 간단한 **근사 분포** (variational distribution)를 사용하여 이를 근사하는 것

- **방법**
  - **근사 분포 선택**:

    비교적 간단한 함수군에서 근사 분포

    q(z)를 선택
    - **목표와 근사의 차이 최소화**: 근사분포 q(z)가 목표 분포 q(z|x) 와 최대한 유사하도록 최적화
    - **최적화**: q(z)의 매개변수를 조정하여
      q(z)가 q(z|x) 와 최대한 유사하도록 만들기

  - 생성된 샘플이 원본 데이터와 일치하도록 Markov chain의 전환 과정 학습
- **Denoising Score Matching**과 **Langevin Dynamics**의 연결을 강조

  ⇒ 고품질 샘플을 생성할 수 있음을 보임

💡 **Denoising Score Matching & Langevin Dynamics**

1. Denoising Score Matching

- 확률 밀도 함수의 score function을 학습하는 방법
  - score function: 확률 밀도 함수의 로그 미분 값으로, 특정 데이터 포인트에서 그 밀도 함수가 얼마나 가파르게 증가하거나 감소하는지를 나타냄

$$\text{score function} = \nabla_x \log p(x)$$

- **Desnoising**: 데이터를 학습할 때 노이즈가 추가된 데이터에서 원래의 데이터를 추정하도록 모델을 학습
- **점수 매칭**: 확률 분포의 형태를 직접적으로 추정하지 않고, 그 분포의 기울기 정보를 학습하는 방식

2. Langevin Dynamics

- 물리학에서 나온 개념으로, 확률적인 시스템에서 **확률 밀도 함수에 따라 샘플을 생성**하는 방법 중 하나
- 확률 분포에서 샘플링할 때 **점진적으로 노이즈를 추가**하면서 확률 밀도의 높은 부분을 찾아가게 하는 방식
- Stochastic Differential Equation (확률 미분 방정식)
  - Langevin Dynamics는 확률적 방정식을 사용하여 확률 분포에서 샘플을 생성
  - 랜덤한 노이즈가 추가되게 하는 과정을 반복하여 점차 안정적인 분포로 수렴하게 만들어줌

$$x_{t+1} = x_t + \epsilon \nabla_x \log p(x_t) + \sqrt{2\epsilon} z_t$$

1. Introduction

- 최근 딥러닝 기반 **생성 모델**들은 다양한 데이터 형식에서 높은 품질의 샘플을 생성해옴
  - GANs (Generative Adversarial Networks), 자율 회귀 모델(Autoregressive Models), 플로우 모델(Flows), VAEs

⇒ 이미지와 오디오 샘플 생성에서 뛰어난 성과

⇒ Energy-based Models, Score Matching 에서도 매우 높은 수준의 이미지 품질을 달성

- **Diffusion Probabilistic Model**

: Markov Chain을 이용하여 데이터에 점진적으로 노이즈를 추가한 뒤, 이를 역으로 처리해 원본 데이터와 유사한 샘플을 생성하는 모델

- **Variational Inference**을 통해 이 모델의 전환 과정을 학습하며, 모델이 데이터를 효율적으로 샘플링할 수 있도록 함

- 기존 연구에서는 확산 모델이 명확한 품질의 샘플을 생성하는 데 실패
  - **Denoising Score Matching**과 **Langevin Dynamics**을 결합하여 더 나은 결과를 얻을 수 있음을 보여줌
  - CIFAR10과 LSUN 데이터셋에서 뛰어난 성능을 기록

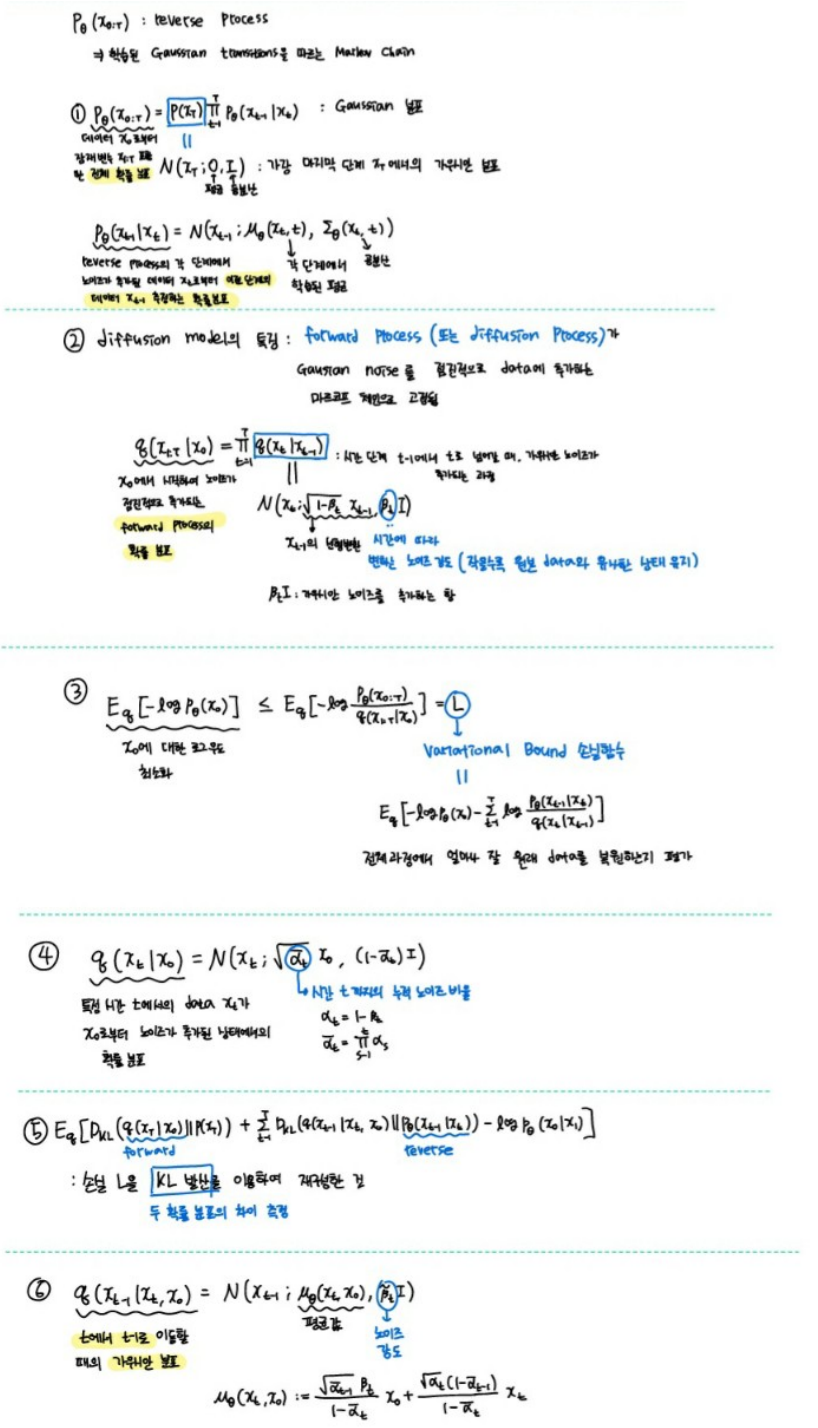
  ⇒ 확산 모델이 다른 생성 모델들보다 더 높은 품질의 이미지를 생성할 수 있음을 입증

2. Background

- Diffusion Models은 잠재 변수 모델의 한 종류로, 데이터 x0로부터 잠재변수 x1:T를 확률적으로 추출하여 모델링

1. 확산 모델의 **joint probability distribution**
2. forward process: x0에서 시작하여 노이즈가 점진적으로 추가되는 과정
3. Variational Bound 최적화
4. Forward 과정에서 xt의 분포

5. KL 발산을 이용한 손실 함수 재구성
6. q(x_{t-1} | x_t, x_0): 시간 단계 t에서 t−1로 이동할 때의 가우시안 분포



3. Diffusion models and denoising autoencoders

- 확산 모델은 제한된 잠재 변수 모델로 보일 수 있지만, 실제로는 구현에서 많은 자유를 허용

- 모델 구현: **forward process**에서 사용되는 분산 βt와 **reverse process**에서의 모델 아키텍처 및 가우시안 분포 매개변수를 선택해야함

1. Forward process and L_T

- **forward process**에서의 분산 βt가 reparameterization를 통해 학습 가능하다는 사실을 무시하고, 대신 이를 고정된 상수로 설정
- 구현에서는 approximate posterior q가 학습 가능한 매개변수를 가지지 않으며, 이에 따라 **L_T는 학습 과정 동안 상수로 유지**됨

  ⇒ 무시 가능

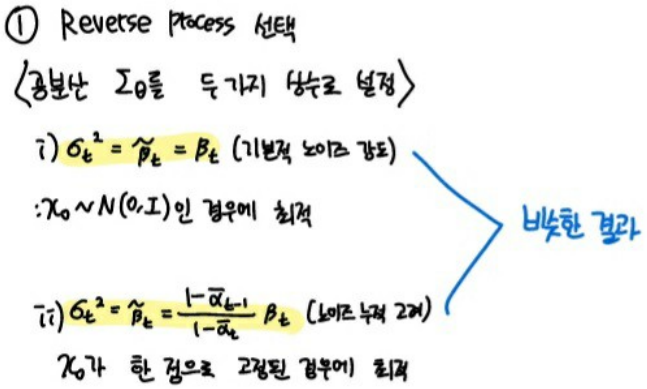
2. Reverse process and L_1:T-1

1. **Reverse Process 선택**

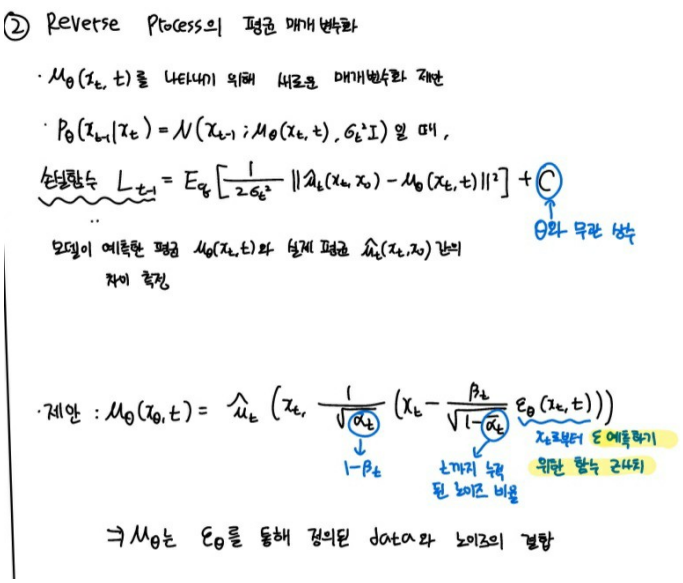
- **Reverse Process**: 각 단계에서 데이터를 이전 단계로 복원하는 과정

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

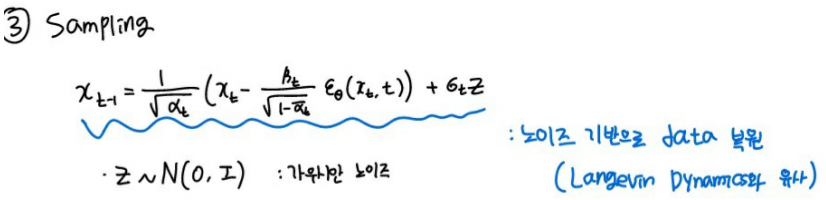
- Σθ(xt, t): 시간에 따라 변하는 상수로 설정된 가우시안 분산
- μθ(xt,t): 시간 t에서 이전 단계 t−1의 평균을 예측하는 함수



2. **평균 매개변수화**



3. 샘플링



4. 손실 함수의 간소화

$$L_t - C = \mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\right]$$

- **denoising score matching**과 유사한 형태
  - 다양한 노이즈 수준에서의 denoising score matching과 동일한 역할 수행

5. Algorithm 1

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

- reverse process에서 모델을 훈련시키는 방법

1. **반복 시작**: 훈련을 위한 과정 반복.
2. **샘플링 x0**: 데이터에서 샘플 x0 뽑기
3. **시간 t 선택**: 1에서 T까지의 시간 단계 중 하나를 무작위로 선택
4. **노이즈 추가**: 가우시안 노이즈 ϵ~N(0,I)를 추가
5. **경사 하강법**: 손실 함수의 그래디언트를 계산하여 ϵθ에 대한 경사 하강법을 수행
6. **반복**: 수렴할 때까지 반복

- 훈련 과정에서 모델은 매개변수 ϵθ를 업데이트

  ⇒ **노이즈와 데이터를 기반으로 데이터를 복원하는 역할**

6. **Algorithm 2**

- 샘플링 알고리즘: 훈련된 모델을 사용하여 새로운 데이터를 생성하는 방법

1. **초기 샘플링**: xT~N(0,I)에서 샘플링을 시작. 즉, 가장 마지막 단계에서 노이즈가 완전히 추가된 상태에서 시작
2. **역방향 진행**: t=T에서 1까지 단계별로 진행
3. **노이즈 제거**: 샘플링을 통해 xt−1을 계산. 이 과정에서 노이즈를 점차 줄이며 원래 데이터로 복원
4. **결과 반환**: 최종적으로 복원된 x0을 반환

- 샘플링 절차는 **Langevin Dynamics**와 유사한 구조로, **가우시안 노이즈를 기반으로 데이터를 점진적으로 복원하는 과정**

3. Data scaling, reverse process decoder, and L0

1. **데이터 스케일링**

- 이미지 데이터는 보통 {0,1,...,255}값의 정수로 구성됨

  ⇒ 이 값을 [-1, 1] 범위로 선형적 스케일링

- 신경망이 일관된 스케일을 가진 입력값에서 작동할 수 있도록 함
- Reverse Process가 표준 정규 분포 p(xT)에서 시작하여 일관된 스케일로 작동할 수 있게 함

2. **Reverse Process 디코더**

- Reverse Process의 마지막 단계는 독립적인 discrete decoder로 변환됨
  - 이 디코더는 가우시안 분포로부터 유도됨

$$p_\theta(\mathbf{x}_0|\mathbf{x}_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(\mathbf{x}_1, 1), \sigma_1^2)\, dx \tag{13}$$

$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases} \qquad \delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

→ D: 데이터 차원 수

→ i: 각 차원의 한 좌표

→ δ+(x) 와 δ-(x): 구간의 경계 정의

- **VAE** 디코더나 **자율 회귀 모델**에서 사용된 연속 분포와 유사

- **변분 경계가 손실 압축과 유사하게 작동하여, 샘플링 시 잡음 없이 깨끗한 출력을 얻을 수 있도록 해줌**

  ⇒ 데이터를 압축하는 과정과 데이터의 확률 분포를 근사하는 과정이 유사함

  ⇒

  손실 압축: 데이터를 **정보 손실 없이** 압축하는 과정

3. **L0**

- discrete decoder 사용하면, **log likelihood**를 계산할 때 스케일링 작업의 야코비안(Jacobian)을 로그 우도에 포함시킬 필요가 없다는 장점이 있음
- 샘플링이 끝나면 μθ(x1,1)를 노이즈 없이 표시

4. Simplified training objective

- 샘플링 품질을 향상시키고 더 간단한 구현을 위해 간소화된 variational bound를 훈련 목표로 설정

1. **간소화된 목표 도출**

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,x_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right]$$

- t는 1에서 T까지 균일하게 분포
  - L0의 경우를 포함한 모든 시간 단계에 적용됨

2. **가중치 재조정**

- 간소화된 목표: **NCSN Denoising Score Matching** 모델에서 사용된 것과 유사한 방식
  - 가중치 없이 훈련 진행
  - **Noise Conditional Score Network(NDSN)에서 사용되는 Denoising Score Matching**: 노이즈가 포함된 데이터의 점수 함수(score function)를 학습하여, 노이즈를 제거한 깨끗한 데이터를 복원하는 방법
    - **Denoising Score Matching: 노이즈가 포함된 데이터에서 점수 함수를 학습**하여, 그 데이터를 깨끗하게 복원하는 방식. 노이즈가 추가된 데이터를 사용해 **원래의 깨끗한 데이터에 대한 점수 함수를 학습**

- LT는 고정된 βt 값을 사용하므로 등장하지 X

- 가중치를 제거하여, 작은 t에 해당하는 손실 항목을 줄임으로써 네트워크가 더 어려운 노이즈 제거 작업에 집중하도록 함

3. 실험 결과

- CIFAR10 데이터셋에서 실험된 모델들의 결과

→ 여러 기존 모델들과 비교하여 더 나은 Inception Score (IS)와 **Fréchet Inception Distance (FID)** 점수를 달성

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**	
Unconditional			
Diffusion (original) [53]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
PixelIQN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87±0.12	25.32	
SNGAN [39]	8.22±0.05	21.7	
SNGAN-DDLS [4]	9.09±0.10	15.42	
StyleGAN2 + ADA (v1) [29]	**9.74** ± 0.05	3.26	
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67±0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46±0.11	**3.17**	≤ 3.75 (3.72)

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
$\tilde{\mu}$ prediction (baseline)		
L, learned diagonal $\mathbf{\Sigma}$	7.28±0.10	23.69
L, fixed isotropic $\mathbf{\Sigma}$	8.06±0.09	13.22
$\|\tilde{\mu} - \tilde{\mu}_\theta\|^2$	–	–
ϵ prediction (ours)		
L, learned diagonal $\mathbf{\Sigma}$	–	–
L, fixed isotropic $\mathbf{\Sigma}$	7.67±0.13	13.51
$\|\tilde{\epsilon} - \epsilon_\theta\|^2$ (L_{simple})	**9.46±0.11**	**3.17**

💡 **IS & FID**

1. Inception Score

- 생성된 이미지의 **다양성**과 **품질**을 평가하는 지표
- 주로 이미지 분류 모델을 사용하여 생성된 이미지의 **클래스 가능성**을 측정
- 이미지가 얼마나 **다양한 범주**에 속하는지와 각 범주에서 **확신 있게** 분류되는지를 평가

2. Fréchet Inception Distance

- 샘플의 품질을 평가하는 데 중요한 지표
- **생성된 이미지와 실제 이미지 사이의 거리**를 측정

4. Experiments

- **실험 설정**
  - **T = 1000**: 모든 실험에서 샘플링을 위해 **1000단계**를 설정

    ⇒ 이전 연구들과 동일한 샘플링 단계 수로, 모델 간 비교를 용이하게 함
  - **Forward Process Variances**: Forward 과정에서의 분산 값 β1=10^-4 에서 시작하여 βT =0.02까지 선형적으로 증가하도록 설정

    ⇒ 데이터가 [−1,1]로 스케일된 상황에서 노이즈와 신호 비율을 조정

    ⇒ 실험에서 xT에서의 신호: 노이즈 비율을 가능한 낮게 유지

- Reverse Process 구현
  - **U-Net Backbone**: Reverse Process는 **U-Net** 백본을 사용하여 구현됨
    - PixelCNN++과 유사하지만 마스크되지 않은 구조를 사용
    - 네트워크의 모든 레이어에 Group Normalization 적용하여 안정성 유지

- **파라미터 공유**: Reverse Process에서 **시간 간의 파라미터 공유**를 통해 효율성 높임
  - 네트워크에 시간 정보를 전달하기 위해 **Transformer의 사인-코사인 위치 임베딩**을 사용

- **Self-Attention**: **16×16** 크기의 피처맵에서 Self-Attention 메커니즘을 적용하여 전역적인 문맥 정보를 처리

1. Sample quality

- 실험 결과로 얻은 **Inception Score (IS)**, **Fréchet Inception Distance (FID)**, Negative Log Likelihood, NLL를 통해 모델의 성능을 평가
- **CIFAR10 데이터셋 결과**: 표 1에서, 얻은 FID 점수는 3.17로, 대부분의 기존 모델들보다 우수한 성능
  - unconditional model에서 얻은 결과로, 일부 class-conditional models보다도 더 나은 성능을 기록
  - 테스트 데이터셋과 비교한 경우에도 FID 점수가 5.24로, 보고된 많은 **학습 데이터셋 FID 점수**보다 우수한 결과

- **Algorithm 3: Sending x0**

Algorithm 3 Sending $\mathbf{x}_0$

1: Send $\mathbf{x}_T \sim q(\mathbf{x}_T|\mathbf{x}_0)$ using $p(\mathbf{x}_T)$
2: **for** $t = T-1, \ldots, 2, 1$ **do**
3: Send $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0)$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
4: **end for**
5: Send $\mathbf{x}_0$ using $p_\theta(\mathbf{x}_0|\mathbf{x}_1)$

  - 샘플링을 위한 과정
  - **Forward Process에서 노이즈가 추가된 데이터에서, Reverse Process를 통해 노이즈를 점차적으로 제거하고 x0를 복원하는 방식**

- 주어진 xT에서 시작하여 x0을 역으로 샘플링하는 절차
- xT: 노이즈가 가득 찬 데이터, x0: 원래의 깨끗한 데이터

1. **노이즈 상태에서 시작**
    - xT~q(xT ∣ x0)를 사용하여 노이즈 상태에서 샘플 xT 보냄
    - q(xT ∣ x0): Forward Process에서 노이즈가 추가된 상태
2. **역방향 진행**
    - t = T-1부터 1까지 루프를 통해 각 xt를 역으로 샘플링
    - Reverse Process는 pθ(xt, xt+1, x0)을 사용하여 샘플링
    - xt+1에서 이전단계 xt로 데이터 복원해나감
3. **최종 샘플 x0 전송**
    - x0을 pθ(x0|x1)을 통해 보냄
    - 원래 데이터를 복원하는 죄종 단계

- **Algorithm 4: Receiving x0**

Algorithm 4 Receiving

1: Receive $\mathbf{x}_T$ using $p(\mathbf{x}_T)$
2: **for** $t = T-1, \ldots, 1, 0$ **do**
3: Receive $\mathbf{x}_t$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
4: **end for**
5: **return** $\mathbf{x}_0$

- 노이즈가 있는 데이터 xT에서 출발하여, 역으로 x0 복원해나가기
- **노이즈가 포함된 데이터에서 점차적으로 노이즈를 제거하면서 원래의 데이터를 복원하는 과정**

1. **노이즈 상태에서 시작**
    - p(xT)에서 xT 받기
    - xT: 노이즈가 가득한 상태의 데이터
2. **역방향 진행**

- t = T -1부터 0까지 루프를 돌면서 xt를 역으로 수신
- pθ(xt|xt+1)를사용하여 각 단계에서 이전 단계의 데이터 복원

3. **최종 데이터 반환**
   - 모델이 예측한 x0 반환

2. Reverse process parameterization and training objective ablation

- Reverse Process에서 사용하는 parameterization와 training objective가 샘플 품질에 어떤 영향을 미치는지 분석
- **평균 μ~ 예측**: true variational bound를 기반으로 훈련될 때만 잘 작동
  - 단순화된 목표인 가중치 없는 평균 제곱 오차(MSE)를 사용할 경우 성능 저하

- **분산 학습**
  - Reverse Process에서 분산을 학습하는 방식(**매개변수화된 대각 행렬 Σθ(xt)을 variational bound에 통합**)은 **훈련이 불안정**해지고 **샘플 품질이 저하**되는 결과를 가져옴
  - 정된 분산을 사용하는 방식과 비교했을 때 성능이 더 나쁨

- **ϵ 예측(노이즈 예측)**
  - variational bound를 사용하여 고정된 분산에서 훈련할 때 평균 μ~를 예측하는 방식과 비슷한 성능을 보임
  - 단순화된 목표를 사용했을 때는 ϵ 예측 방식이 훨씬 더 나은 성능을 보임

3. Progressive coding

- **Progressive coding:** 데이터를 점진적으로 압축하는 방식
  - 데이터는 처음에 큰 그림(대략적인 정보)을 전송하고, 이후 점진적으로 더 정밀한 정보를 추가해서 데이터의 정확성을 높임
- 모델이 데이터를 생성하는 과정이 **Progressive Coding**과 유사

- Diffusion Model:
  처음에
  노이즈가 많이 포함된 데이터에서 시작하여 점진적으로 **노이즈를 제거**하면서 원래 데이터를 복원하는 과정
  ⇒ Progressive Coding에서 점진적으로 더 많은 비트를 전송하면서 왜곡을 줄여가는 것과 유사한 과정

1. **CIFAR10 모델의 코딩 길이 분석**

- **훈련 데이터**와 **테스트 데이터** 간의 차이는 1차원당 최대 0.03 비트

  ⇒ 다른 likelihood 기반 모델에서 보고된 차이와 유사

  ⇒ 모델이 overfitting 하지 않았음을 의미

- 손실 없는 코딩 길이가 에너지 기반 모델이나 score matching 모델에서 보고된 큰 추정치보다 더 좋음
- But, 다른 likelihood 기반 생성 모델에 비해 여전히 성능이 떨어짐

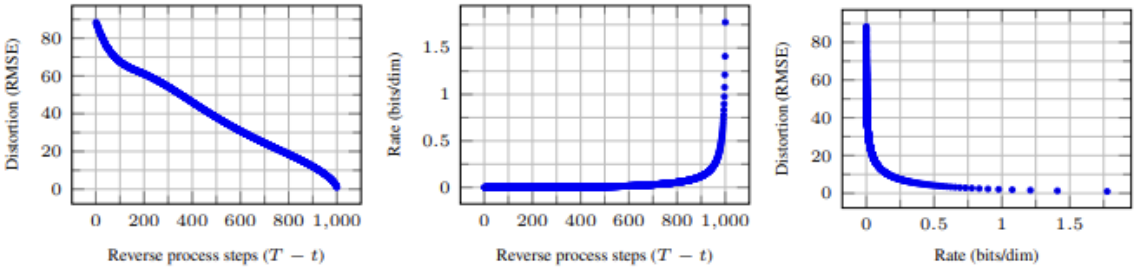
2. **손실 압축과 variational bound**

- 생성된 샘플이 lossy compression의 특성을 지님
  - variational bound에서 L1+ .. + LT를 rate(압축 속도)로, L0을 distortion(왜곡)으로 간주 가능하다는 의미
- CIFAR10에서 최고 품질의 샘플을 생성하는 모델은 **1.78 bits/dim**의 rate와 **1.97 bits/dim**의 왜곡을 가짐
  - RMSE가 0.95
  - 손실 없는 코딩 길이의 절반 이상이 이러한 인지할 수 없는 왜곡에 할당되었음을 보여줌

3. **Progressive Lossy Compression**

$$\mathbf{x}_0 \approx \hat{\mathbf{x}}_0 = \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t)\right)/\sqrt{\bar{\alpha}_t}$$

- 모델의 **rate-distortion behavior**를 더 깊이 탐구하기 위해 **progressive lossy code**를 도입
- **Rate**: 데이터를 표현하는 데 필요한 비트 수
- **Distortion:** 데이터를 복원했을 때 원본과 얼마나 차이가 있는지 나타냄
    - 데이터를 압축할수록 사용되는 비트 수(rate)는 줄어들지만, 원본 데이터와 복원된 데이터 사이의 왜곡(distortion)은 커짐
- x0 ~ q(x0)를 전송할 때마다 점진적으로 압축된 데이터를 전송
- **각 t에서 왜곡을 줄이면서 점진적으로 데이터 복원 가능**
- 실험에서는 CIFAR10 테스트셋에 대해 각 시간 t에서 왜곡이 **RMSE**로 계산되고, rate는 누적된 비트 수로 계산됨



⇒ 초기 저속 구간에서는 왜곡이 급격히 감소하다가 점차적으로 완만해짐

= 많은 비트들이 인지할 수 없는 왜곡을 줄이는 데 할당된다는 것

4. **Progressive Generation**

- **압축된 데이터를 점진적으로 복원하면서 이미지 품질을 점차적으로 높이는 방식**
- **무조건적 생성 과정**
    - 무조건적인 **progressive generation** 프로세스를 통해 데이터를 복원하는 실험을 진행
    - **Algorithm 2**에서 설명한 것과 같은 방식으로 진행됨
    - 샘플링된 x0_hat이 xt_hat에서 점차적으로 복원됨
    - **큰 시각적 특징**이 먼저 형성되며, 시간이 지날수록 더 세부적인 정보가 복원

- <CIFAR10 데이터셋에서의 **progressive generation** 과정>

- <CelebA-HQ 256 × 256 샘플에서 **같은 잠재 변수를 공유한 이미지**가 각 시간 단계에서 점진적으로 생성되는 과정>

5. **Autoregressive Decoding과의 연결**

- variational Bound를 **Autoregressive Decoding** 방식으로 재해석할 수 있음을 언급

$$L = D_{\mathrm{KL}}(q(x_T)||p(x_T)) + \mathbb{E}_q \left[\sum_{t>1} D_{\mathrm{KL}}(q(x_{t-1}|x_t)||p_\theta(x_{t-1}|x_t)) \right] + H(x_0)$$

- autoregressive 모델과 유사한 방식으로 데이터를 생성하는 과정임을 의미
- Gaussian Diffusion Model은 autoregressive 모델처럼 작동하며, 각 단계에서 데이터를 점진적으로 생성

4. Interpolation

- x0와 x0′ 사이에서 보간된 데이터 xt를 생성하고 reverse process를 사용해 해당 데이터 복원
- **노이즈가 추가된 상태**에서도 보간된 이미지를 복원할 수 있음을 보여줌

- 보간: 두 지점 사이의 중간 값을 추정하는 방법(이미 알고 있는 데이터 포인트들 사이에 **새로운 값을 생성**하는 과정)

  < 두 얼굴 이미지 사이에서 여러 값을 사용하여 중간 형태의 이미지를 생성>

Figure 8: Interpolations of CelebA-HQ 256x256 images with 500 timesteps of diffusion.

5. Related Work

1. **Diffusion Models vs 다른 모델**

- Flows, VAE(Variational Autoencoders)와 구조적으로 유사
- Diffusion model
  - 잠재변수 xT가 데이터 x0과 mutual information(상호 정보)를 거의 가지지 X

    ⇒ **노이즈가 많이 포함된 상태**에서 데이터 복원을 시작하는 독특한 구조를 반영

- **ϵ 예측 방식**: diffusion model에서 사용하는 노이즈 예측 방식은 Denoising Score Matching(DSM)의 수학적 원리와 관련 있음

  ⇒ 여러 노이즈 수준에서의 **denoising score matching**과 **Annealed Langevin Dynamics** 샘플링 사이의 유사성을 강조하며, 샘플링 방식에 중요한 통찰을 제공

2. **Variational Inference & Langevin Dynamics 훈련**

- Diffusion Model: log likelihood 직접적으로 평가 가능
- **Langevin Dynamics 샘플러**를 훈련하는 과정에서 Variational Inference를 명시적으로 사용
- infusion training, variational walkback, generative stochastic networks(GSN)
  ⇒ Markov 연쇄(체인)의 전이 연산자(transition operators)를 학습하는 방법

3. **에너지 기반 모델들과의 연결**

- **rate-distortion 곡선**을 Variational bound를 사용하여 계산

  ⇒ **annealed importance sampling**에서 왜곡 페널티를 통해 계산된 rate-distortion 곡선과 유사한 방식

💡 **<Annealed Importance Sampling (AIS) & Rate-Distortion 곡선>**

1. Annealed Importance Sampling (AIS)

- 확률 분포에서 **효율적으로 샘플링**하는 방법 중 하나
- **Importance Sampling:** 하나의 쉬운 분포에서 샘플을 뽑고, 이를 이용해 복잡한 분포에 대한 추정치를 계산하는 방법

2. **Rate-Distortion 곡선**

- **데이터를 손실적으로 압축**할 때, 얼마나 적은 비트로 표현할 수 있는지, 그리고 그 과정에서 데이터가 얼마나 왜곡될 것인지 사이의 관계를 설명하는 것
- Rate: 데이터를 압축하여 저장하거나 전송하는 데 필요한 **비트 수**
- **Distortion(왜곡)**: 압축된 데이터를 다시 복원할 때 원래 데이터와 얼마나 **다르게 복원**되는지를 나타냄

4. **Progressive Decoding**

- **Progressive Decoding** 개념: **convolutional DRAW** 및 관련 모델에서 유사하게 나타나는 개념

💡 **<DRAW(Differentiable Recurrent Attentive Writer) & Convolutional DRAW 모델>**

1. DRAW(Differentiable Recurrent Attentive Writer) 모델

- 여러 timesteps에 걸쳐 순차적으로 이미지 생성
- RNN을 사용하여 각 타임스텝에서 생성된 이미지를 업데이트
- Attention 이용: 매 타임스텝에서 이미지의 특정 부분에 집중해 그 부분을 더 자세히 생성하고, 나머지 부분은 나중에 처리

2. **Convolutional DRAW 모델**

- DRAW 모델의 순차적인 이미지 생성 방식을 CNN과 결합하여 확장한 모델

  ⇒ **공간적 관계**를 더 잘 학습하고, 더 높은 해상도의 이미지를 생성할 수 있도록 설계된 모델

6. Conclusion

1. **연구 요약**

- diffusion model을 사용해 **고품질의 이미지 샘플**을 생성할 수 있음을 보여줌
- **variational inference**, **Denoising Score Matching**, **Langevin Dynamics**, **Progressive Lossy Compression** 등의 기법들과의 연결성 발견
- diffusion model이 이미지 데이터에 대해 Inductive Bias를 가지고 있음을 확인
  - 다른 유형의 데이터 modalities에 적용하거나, 생성 모델의 다른 구성 요소로 확장할 수 있는 가능성을 가지고 있다고 평가됨

2. **Broader Impact**

- 긍정적 영향
  - diffusion model은 **데이터 압축**에 유용할 수 있음
  - 인터넷 트래픽이 증가함에 따라 데이터 액세스 가능성을 높이는 데 기여 가능
  - unlabeled data에 대한 Representation Learning에도 기여 가능

        - 분류, 강화 학습, 및 다양한 분야에 적용
    - **예술**, **사진**, **음악** 등 창의적인 분야에서 긍정적인 방식으로 사용될 가능성

- 부정적 영향
    - **딥페이크(Deepfake)** 및 가짜 이미지 생성은 정치적 목적으로 악용 가능
    - 모델이 학습된 데이터에 내재된 bias가 사회적으로 부정적인 영향을 미칠 수 있음
        - 자동화된 시스템이 인터넷에서 수집하는 대규모 데이터셋에서 bias를 제거하지 못한다면, 편향이 강화될 위험성