Denoising Diffusion Probabilistic Models

Abstract & Introduction

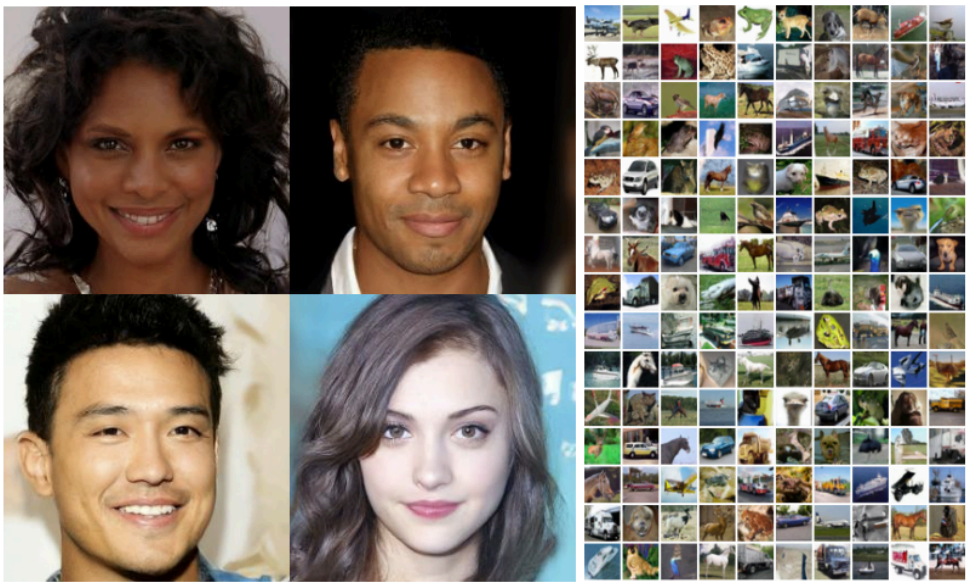

Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

최근 GAN, VAE, Autoregressive 모델 등 다양한 생성 모델이 발전함에 따라 이미지 생성 분야에서도 높은 품질의 결과물이 등장함.

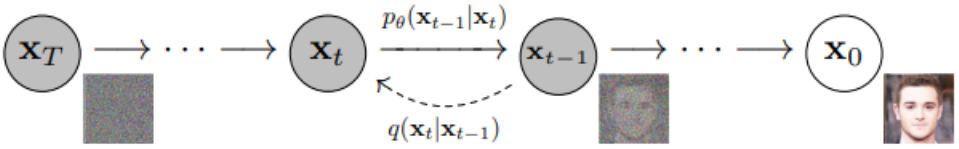

Figure 2: The directed graphical model considered in this work.

각 모델은 장단점이 뚜렷하며, 특히 GAN은 사실적인 이미지를 생성하는 데 강점을 보이나 훈련이 불안정하고 모드 붕괴 문제가 있음.

반면 VAE는 학습 안정성은 높지만, 샘플 품질이 낮은 경우가 많음.

이러한 배경 속에서 Diffusion Probabilistic Models (DPMs)는 흥미로운 대안으로 제시됨.

이 모델은 데이터를 점진적으로 노이즈화한 후, 반대로 복원하는 과정을 통해 샘플을 생성함.

그러나 기존에는 샘플 품질이 낮고, 효율이 떨어진다는 평가가 있었음.

본 논문에서는 이러한 편견을 깨고, 고품질 이미지 생성이 가능한 새로운 Diffusion 모델(Denoising Diffusion Probabilistic Model, DDPM)을 제안함.

모델은 단순하지만 강력한 학습 구조를 가지며, 기존 모델 대비 우수한 성능을 보임.

또한 DDPM은 **Score Matching, Langevin Dynamics, Variational Inference**와 이론적으로 연결되어 있음.

해당 모델은 생성뿐 아니라 **압축, 표현 학습, 인터폴레이션**과 같은 다양한 응용 가능성도 내포함.

Method

DDPM은 정방향 확산 과정(forward diffusion)과 역방향 복원 과정(reverse denoising)으로 구성된 확률 생성 모델임.

정방향에서는 입력 데이터에 점차적으로 가우시안 노이즈를 더하여 완전히 무작위화된 상태 x_T까지 진행함.

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

원래 수식

여기서 βt는 시간에 따라 증가하는 노이즈 분산 파라미터임. 이 과정을 반복하면 다음처럼 나타남:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

따라서 xt는 x0과 epsilon 에 대해 다음과 같이 sampling이 가능함

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

역방향에서는 학습된 신경망이 이를 순차적으로 제거하여 원본 데이터를 복원함.

모델은 시간 단계마다 조건부 가우시안 분포를 사용함:

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

원래 수식

여기서 평균 μθ와 분산 Σθ는 학습된 신경망이 예측함. 초기 diffusion 모델들은 역방향 평균 μθ를 직접 예측하는 방식을 사용함. 이를 위해서는 원래 수식에서 유도된 **이론적인 역방향 평균 μ̃**을 사용해야 했음:

$$\tilde{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

하지만 이 수식은 복잡하고, μ 자체를 예측하는 방식은 학습이 불안정하며 구현도 복잡하다는 단점이 있었음.

[ε 예측 방식 도입]

기존에는 평균(μ)을 직접 예측했지만, 본 논문에서는 **노이즈 ε를 직접 예측**하는 방식으로 구조를 단순화함.

이 방식은 **Langevin dynamics**와 유사하며, **denoising score matching**으로 해석 가능함.

예측은 다음과 같은 수식을 따름:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$$

[손실 함수 설계]

학습 손실은 단순화된 변분 경계(L_simple)를 사용함. 이는 평균 제곱 오차 기반의 손실로 다음과 같음

$$L_{\text{simple}} = \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

$$L_{t-1} - C = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}), \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}) - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon})\right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}), t)\right\|^2\right] \tag{9}$$

$$= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}), t)\right\|^2\right] \tag{10}$$

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

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Equation (10) reveals that $\boldsymbol{\mu}_\theta$ must predict $\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)$ given $\mathbf{x}_t$. Since $\mathbf{x}_t$ is available as input to the model, we may choose the parameterization

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t))\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) \tag{11}$$

where $\boldsymbol{\epsilon}_\theta$ is a function approximator intended to predict $\boldsymbol{\epsilon}$ from $\mathbf{x}_t$. To sample $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is to compute $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The complete sampling procedure, Algorithm 2, resembles Langevin dynamics with $\boldsymbol{\epsilon}_\theta$ as a learned gradient of the data density. Furthermore, with the parameterization (11), Eq. (10) simplifies to:

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{12}$$

which resembles denoising score matching over multiple noise scales indexed by t [55]. As Eq. (12) is equal to (one term of) the variational bound for the Langevin-like reverse process (11), we see that optimizing an objective resembling denoising score matching is equivalent to using variational inference to fit the finite-time marginal of a sampling chain resembling Langevin dynamics.

[네트워크 구조]

- **U-Net 구조**를 기반으로 하고 있으며, 중간에 **Self-Attention 블록**도 포함함.
- 시간 정보는 **sinusoidal positional embedding**을 통해 네트워크에 전달함.
- 학습 시 T = 1000 단계까지 확산을 시뮬레이션하며, 샘플링도 동일한 역순으로 진행됨.

[프레임워크 해석]

Diffusion 과정을 **프로그레시브 로시 압축(progressive lossy decompression)** 과정으로 해석함.

노이즈화는 정보를 점차 제거하는 인코딩, 복원은 점진적으로 정보를 회복하는 디코딩으로 볼 수 있음.

이 해석은 오토리그레시브 모델에서의 픽셀 순서 대신 **정보 순서화된 디코딩 구조**로 일반화된 형태임.

Result & Analysis

method	COCO [66] AP	AP^S	AP^M	AP^L	LVIS v1 [44] AP	AP^S	AP^M	AP^L
ViTDet-H [62]	51.0	32.0	54.3	68.9	46.6	35.0	58.0	66.3
zero-shot transfer methods (segmentation module only):								
SAM	46.5	30.8	51.0	61.7	44.7	32.5	57.6	65.5

Table 5: Instance segmentation results. SAM is prompted with ViTDet boxes to do zero-shot segmentation. The fully-supervised ViTDet outperforms SAM, but the gap shrinks on the higher-quality LVIS masks. Interestingly, SAM outperforms ViTDet according to human ratings (see Fig. 11).

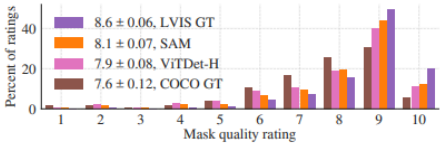

Figure 11: Mask quality rating distribution from our human study for ViTDet and SAM, both applied to LVIS ground truth boxes. We also report LVIS and COCO ground truth quality. The legend shows rating means and 95% confidence intervals. Despite its lower AP (Table 5), SAM has higher ratings than ViTDet, suggesting that ViTDet exploits biases in the COCO and LVIS training data.

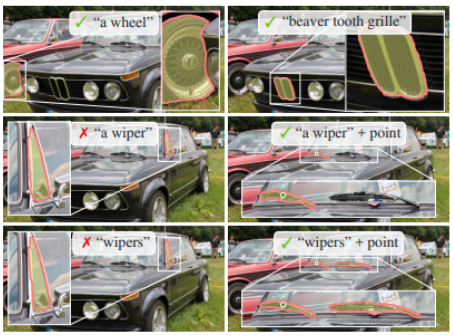

Figure 12: Zero-shot text-to-mask. SAM can work with simple and nuanced text prompts. When SAM fails to make a correct prediction, an additional point prompt can help.

Results. We show qualitative results in Fig. 12. SAM can segment objects based on simple text prompts like "a wheel" as well as phrases like "beaver tooth grille". When SAM fails to pick the right object from a text prompt only, an additional point often fixes the prediction, similar to [31].

7.6. Ablations

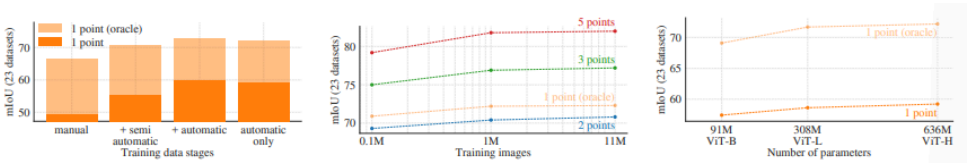

Figure 13: Ablation studies of our data engine stages, image encoder scaling, and training data scaling. (Left) Each data engine stage leads to improvements on our 23 dataset suite, and training with only the automatic data (our default) yields similar results to using data from all three stages. (Middle) SAM trained with ∼10% of SA-1B and full SA-1B is comparable. We train with all 11M images by default, but using 1M images is a reasonable practical setting. (Right) Scaling SAM's image encoder shows meaningful, yet saturating gains. Nevertheless, smaller image encoders may be preferred in certain settings.

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