DDIM 2021

Monday, July 7, 2025 4:37 PM

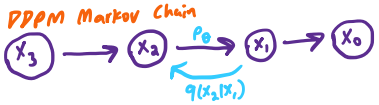
Denoising Diffusion Implicit Models

- More efficient/fast compared to DDPM (10 to 50x faster)
- Generates superior samples

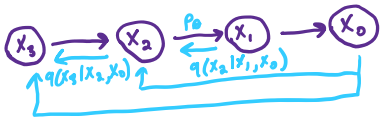
Introduction

- Introduces non-markovian inference models

DDPM Markov Chain



DDIM non-Markovian inference model



DDPM
Goal: learn model distribution $p_\theta(x_0)$ that approximates $q(x_0)$ and easy to sample from

Variational Inference For Non-Markovian Forward Processes

- Need to find faster inference process to reduce # of iterations in forward process
  - observation: The DDPM objective, L_σ, only depends on $q(x_t|x_0)$, but not $q(x_{1:T}|x_0)$

Forward process

- Consider family Q of inference distributions, indexed by the vector $\sigma \in \mathbb{R}^T_{\geq 0}$

Formula 1: $q_\sigma(x_{1:T}|x_0) := q_\sigma(x_T|x_0)\prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0)$

- $q_\sigma(x_T|x_0)$: distribution of final noised sample, x_T
- $q_\sigma(x_{t-1}|x_t, x_0)$: distribution of the previous-step latent, x_{t-1}, given current noisy sample, x_t, and the original clean image x_0, under a noise schedule, σ

Meaning: The full forward noising process, expressed as $q_\sigma(x_{1:T}|x_0)$ (where σ denotes we parameterize by the noise) is defined as the joint distribution of all the noisy variables $x_1, x_2, \ldots, x_T$ given the original data x_0.

Purpose: Makes the reverse process deterministic (no neural network like with the DDPM), so the DDIM just learns to predict the noise.

Formula 2: $q_\sigma(x_t|x_{t-1}, x_0) := \mathcal{N}\left(\sqrt{\alpha_{t-1}}\,x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{x_t-\sqrt{\alpha_t}\,x_0}{\sqrt{1-\alpha_t}},\ \sigma_t^2 I\right)$

- $\alpha_t := 1-\beta_t$: how much of original image should remain at step $t-1$
- $\sqrt{1-\alpha_{t-1}-\sigma_t^2}$: scale factor
- $\frac{x_t-\sqrt{\alpha_t}\,x_0}{\sqrt{1-\alpha_t}}$: estimate of pure noise added to x_0 to get x_t
- $\sigma_t^2 I$: standard variance (in all directions)
- σ_t = the SD of the noise added at step t
- $q_\sigma(x_t|x_{t-1}, x_0)$: distribution of less noisy image given it's more noisy image, and the fully denoised image
- the mean is the sum of a clean signal (from x_0) and a residual noise signal (from x_t), carefully scaled so the resulting sample x_{t-1} stays consistent with the diffusion schedule.

Meaning: Given how noisy the image is now, and what the original image looked like, how should I compute the next denoised image?

Purpose: Defines how to take one step backwards in the diffusion process.

Bayes' Theorem: $P(B|A) = \frac{P(A|B,B)}{P(B)}$

Formula 3: $q_\sigma(x_t|x_{t-1},x_0) = \frac{q_\sigma(x_{t-1}|x_t,x_0)\, q_\sigma(x_t|x_0)}{q_\sigma(x_{t-1}|x_0)}$

- $q_\sigma(x_t|x_{t-1},x_0)$: forward transition probability from $x_{t-1} \to x_t$ given the original image (derived from Bayes')
- $q_\sigma(x_{t-1}|x_t,x_0)$: the "reverse" step (Formula 2)
- $q_\sigma(x_t|x_0)$: overall distribution of noisy image x_t given x_0
- $q_\sigma(x_{t-1}|x_0)$: overall distribution of x_{t-1} given x_0

Meaning: the forward transition probability from $x_{t-1} \to x_t$ given the clean image

Purpose: defines a forward step using reverse-style steps, so the model can be reversed deterministically.

Reverse Process

- Generative process: $p_\theta(x_{0:T})$
  - each $p_\theta^{(t)}(x_{t-1}|x_t)$ leverages knowledge of $q_\sigma(x_{t-1}|x_t,x_0)$ (defined already)

Formula 4: $f_\theta^{(t)}(x_t) := \frac{x_t - \sqrt{1-\alpha_t}\cdot \epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}}$

- $f_\theta^{(t)}(x_t)$: estimate of a clean image at time step t
- x_t: noisy image
- $\epsilon_\theta^{(t)}(x_t)$: predicted noise

Meaning: The model's estimate of a clean image is the noisy image minus the predicted noise (scaled).

Purpose: Tells us how to predict the clean image

Formula 5: $p_\theta^{(t)}(x_{t-1}|x_t) := \begin{cases} \mathcal{N}(f_\theta^{(t)}(x_1), \sigma_1^2 I) & \text{if } t=1 \\ q_\sigma(x_{t-1}|x_t, f_\theta^{(t)}(x_t)) & \text{otherwise} \end{cases}$

- $f_\theta^{(t)}(x_1)$: estimate of clean image at $t=1$
- $\sigma_1^2 I$: standard unit variance
- $q_\sigma(x_{t-1}|x_t, f_\theta^{(t)}(x_t))$: forward transition probability given the predicted clean image (similar to Formula 3)

Meaning: The predicted next image given the current more noisy image is either:

1) Add a small Gaussian centered at the prediction of the clean image if we're at the end ($t=1$)
2) Use the clean image estimate to deterministically guide the reverse process in all other cases

Purpose: tells us how to sample next less noisy image

Training Objective

- Training Loss function: $J_\sigma(\epsilon_\theta)$

Formula 6: $J_\sigma(\epsilon_\theta) := \mathbb{E}_{x_{0:T}\sim q_\sigma(x_{0:T})}\left[\log q_\sigma(x_{1:T}|x_0) - \log p_\theta(x_{0:T})\right]$

- $J_\sigma(\epsilon_\theta)$: function
- $x_{0:T}\sim q_\sigma(x_{0:T})$: distribution over noisy images $x_1, x_2, \ldots, x_T$ generated from clean image x_0
- $p_\theta(x_{0:T})$: model's learned joint distribution in reverse, from pure noise (x_T) to a clean image (x_0)

$\log(a) - \log(b) = \log\left(\frac{a}{b}\right) \Rightarrow$

comparing two stories
→ a: starting from a real image x_0, if I add noise in a known way, I know how images should look at each step
from random noise x_T, I use my neural network to try to
I might be wrong or uncertain.

loss function:

the expectation/weighted average over the full trajectory $(x_0, x_1, \dots, x_T)$ that were sampled from the known forward process q_σ

→b: starting from ... reverse-engineer the path back to x_0 – but ...

$$\rightarrow \log\left(\frac{\text{true noising process}}{\text{model's reverse process}}\right)$$

true/known probability of getting final noised image x_T starting from the clean image, x_0

sum of known conditional probabilities describing how each prior noisy image x_{t-1} looks given the next one, x_t, and the original, x_0

the prior/initial belief of the final noised sample, x_T. (usually just a standard normal: $p_\theta(x_T) = N(x_T; 0, I)$)

$$= \mathbb{E}_{x_{0:T} \sim q_\sigma(x_{0:T})}\left[\log\left(q_\sigma(x_T|x_0)\right) + \sum_{t=2}^{T} \log q_\sigma(x_{t-1}|x_t, x_0) - \sum_{t=1}^{T} \log p_\theta^{(t)}(x_{t-1}|x_t) - \log p_\theta(x_T)\right]$$

model's predicted probabilities for reversing noise – from x_t to x_{t-1} – at every step.

Meaning: The objective function quantifies the expected difference between the true forward noising process and the model's learned reverse denoising process, using KL-divergence

Purpose: we take the derivative of the loss to update parameters (using autograd)

Sampling

L_1 loss = mean absolute error

L_2 loss = mean squared error

Denoising Diffusion Implicit Models

we generate a sample $x_{t-1} \sim p_\theta(x_{1:T})$ as follows:

image sample

predicted clean image

direction pointing to x_t, the more noisy image

random noise

Formula 7: $$x_{t-1} := \sqrt{\alpha_{t-1}}\left(\frac{x_t - \sqrt{1-\alpha_t}\,\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^{(t)}(x_t) + \sigma_t \epsilon_t$$

$\alpha_0 = 1$

$\epsilon_t = N(0, I)$

Meaning: given the current (more noisy) image, we predict/generate the next (slightly denoised) image by adding the predicted clean image, a vector pointing in the direction of the more noisy image, and random noise (to make it more stochastic) – all with scaling factors.

Purpose: tells us how to generate the next denoised sample, x_{t-1}, given x_t during sampling.

Accelerated Generation

- key to the performance speed-up:
  - the DDPM takes 100's to 1000's of steps to turn noise into an image
  - the DDIM speeds up generation by changing how we sample

1) from DDPM, we know $q(x_t|x_0) = N\left(\sqrt{\bar{\alpha}_t}\,x_0, (1-\bar{\alpha}_t)I\right)$
   - we then know you can go directly from a clean image to a noisy version in one step

2) Predict x_0 from any x_t using the model
   - The trained model $\epsilon_\theta(x_t, t)$ predicts the noise in x_t
   - we rearrange the forward equation: $\hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\cdot\epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}$

gives a guess of the clean image from any noisy image

- gives ... g...

3) Define deterministic reverse rule

- using predicted $\hat{x}_0$, we define a new reverse rule to go from $x_t \rightarrow x_{t-1}$, without sampling

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \cdot \hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}} \cdot \epsilon_\theta(x_t, t)$$

- resembles the forward process - using model's predicted noise
  - → fully deterministic
  - → works with any schedule of steps
    - e.g., t=1000 → 950 → 900
    - trade off: $\hat{x}_0$ isn't perfect so balance results with desired sampling time

Summary

DDPM vs. DDIM

Feature	DDPM (original)	DDIM (improved)
Sampling type	Stochastic (adds noise at each step)	Deterministic (no noise added)
Reverse step Formula	$x_{t-1} \sim N(\mu_\theta, \sigma_t^2)$	$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\cdot\epsilon_\theta$
Steps Required	~1000 for quality images	10-50 (speed vs performance)
Randomness During Inference	Yes - samples at each step	No - completely deterministic
Can Skip Steps	No - must go step-by-step	Yes - jumps between custom steps
key trick	Learns to predict noise	Uses predicted noise to recover $\hat{x}_0$, then rewinds deterministically using known marginals
why it works	Reverse process modeled as a markov chain of Gaussians	Reverse process derived from forward marginals $q(x_t \mid x_0)$ and the model's noise predictions
Noise Prediction Used	Yes - to define mean of each reverse Gaussian	Yes - to reconstruct $\hat{x}_0$, which the element then determines the next image in the reverse process.