

DDPM은 뛰어난 성능을 보여줬지만, 많은 iteration이 필요하다.

Improved DDPM, DDIM과 같이 noise scheduling이나 denoising equation을 통해 Sampling을 빠르게 해하기 위한 노력이 있었지만, quality를 유지하기가 힘들다.

따라서, quality를 유지하면서, 가속화가 위해, DDPM이 manifold에 대한 미분방식을 푸는 것으로 취급되어야 한다는 관점을 제시한다.

∴ DDPM이 대체 pseudo numerical method를 제시. (PNDM)

그리고, manifold의 differential equation을 푸는 방법과

DDIM이 결국 PNDM의 근본적인 것이라는 것을 보인다.

1. Introduction.

PNDM은 data density가 높은 manifold

특정 manifold를 따라 sample 생성하기 위한 numerical method 설계

DDPM과 numerical method 사이에 이론적인 연결을 구축하는

DM의 적합적이고 일관되게 differential equation을 먼저 계산한다.

다른 이론과 다르게 특정 manifold에 대한 sample을 보장한다.

그리고, DDIM을 더 잘 이해.

제안된 pseudo linear multi-step method가 빠른 방법

Contribution:

1. DDIM과 numerical method의 결합, 그리고 더 빠른 generation.

2. 짧은 sampling으로도 FID 성능↑

3. 다양한 variance schedule 기반 작동 (일반화 잘됨)

2. Background.

-DDPM

→ Markov

$$\begin{aligned} x_{t+1} &\sim \mathcal{N}(\sqrt{1-\beta_t}x_t, \beta_t\mathbf{I}), \quad t = 0, 1, \dots, N-1. \\ x_{t-1} &\sim \mathcal{N}(\mu_\theta(x_t, t), \beta_\theta(x_t, t)\mathbf{I}), \quad t = N, N-1, \dots, 1. \end{aligned} \quad (1)$$

↳ reverse

$$\begin{aligned} q(x_t|x_0) &= \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \\ q(x_{t-1}|x_t, x_0) &= \mathcal{N}(\bar{\mu}_t(x_t, x_0), \bar{\beta}_t\mathbf{I}). \end{aligned}$$

Objective function.

$$\begin{aligned} L_{t-1} &= \mathbb{E}_q [||\bar{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)||^2] \\ &= \mathbb{E}_{x_0, \epsilon} \left[\left| \left| \frac{1}{\sqrt{\alpha_t}} \left(x_t(x_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right) - \mu_\theta(x_t(x_0, \epsilon), t) \right| \right|^2 \right] \\ &= \mathbb{E}_{x_0, \epsilon} \left[\frac{\beta_t^2}{\alpha_t(1-\bar{\alpha}_t)} ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2 \right]. \end{aligned} \quad (3)$$

-SDE

→ denoising process

$$dx = \left((\sqrt{1-\beta(t)} - 1)x(t) - \beta(t)\epsilon_\theta(x(t), t) \right) dt + \sqrt{\beta(t)}d\bar{w}. \quad (4)$$

$$dx = \left((\sqrt{1-\beta(t)} - 1)x(t) - \frac{1}{2}\beta(t)\epsilon_\theta(x(t), t) \right) dt. \quad (5)$$

→ ODE 버전.

→ reverse DPM 가능성의 가능성

- Numerical Method.

forward Euler, runge-kutta, linear multi-step 등을 이용해, ODE를 풀 수 있다.

DE가 $\frac{dx}{dt} = f(x, t)$ 일 때,

- Forward Euler Method

$$x_{t+\delta} = x_t + \delta f(x_t, t)$$

- Runge-kutta Method

모든 step에서 정보 있음.

$$\begin{cases} k_1 = f(x_t, t), & k_2 = f(x_t + \frac{\delta}{2}k_1, t + \frac{\delta}{2}) \\ k_3 = f(x_t + \frac{\delta}{2}k_2, t + \frac{\delta}{2}), & k_4 = f(x_t + \delta k_3, t + \delta) \\ x_{t+\delta} = x_t + \frac{\delta}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{cases} \quad (6)$$

- Linear Multi-step Method.

$$x_{t+\delta} = x_t + \frac{\delta}{24}(55f_t - 59f_{t-\delta} + 37f_{t-2\delta} - 9f_{t-3\delta}), \quad f_t = f(x_t, t). \quad (7)$$

3. Pseudo Numerical Method for DDPM.

DDPM과 numerical의 연결을 위해 DDPM의 DE 계산.

이후 고전적 수치방법의 한계를 제거.

→ 해설마다 다른 numerical을 gradient transfer 부분으로 나눔

→ 비선형의 pseudo numerical 정의.

→ DDIM이 포함됨!

→ 더 빠르기 향상된 linear multi-step

PNDPM의 대안 새로운 pseudo numerical 제작.

Pseudo의 convergence 분석. → 효율성 입증

- Formula Transformation.

DDPM과 DDIM의 reverse는 다음을 만족

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \epsilon_\theta(x_t, t) + \sigma_t \epsilon_t. \quad (8)$$

$\sigma_t > 0$ 이면 DDPM, 0 이면 DDIM.

0일 때, random성을 0으로 만들 수 있고, 특정 ODE의 discrete form이 될 수 있다.

random이 있는 DE는 numerical method의 계한이 있다. (이론적으로)

DDIM은 전개 전개 단계가 적을 때, 더 나은 acceleration. (경험적으로)

$\therefore \sqrt{\alpha_t} = 0$ 일 때에 합동, 아닐 때는 성능도 L, 적용도 힘들.

(8)의 해석하는 ODE를 찾기 위해 $t-1$ 을 $t-\delta$ 로 변경, 식을 differential form으로 변경.

$$x_{t-\delta} - x_t = (\bar{\alpha}_{t-\delta} - \bar{\alpha}_t) \left(\frac{x_t}{\sqrt{\bar{\alpha}_t}(\sqrt{\bar{\alpha}_{t-\delta}} + \sqrt{\bar{\alpha}_t})} - \frac{\epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}(\sqrt{(1 - \bar{\alpha}_{t-\delta})\bar{\alpha}_t} + \sqrt{(1 - \bar{\alpha}_t)\bar{\alpha}_{t-\delta}})} \right). \quad (9)$$

↳ 그냥 살피기엔 유도법

이제 x_t 의 미분을 계산하고, 아래를 연습.

$$\lim_{\delta \rightarrow 0} \frac{x_t - x_{t-\delta}}{\delta} = -\bar{\alpha}'(t) \left(\frac{x(t)}{2\bar{\alpha}(t)} - \frac{\epsilon_\theta(x(t), t)}{2\bar{\alpha}(t)\sqrt{1 - \bar{\alpha}(t)}} \right).$$

↳ 미찬까지로 $\delta \rightarrow 0$ 이면 정리됨

(9)의 $\delta \rightarrow 0$ 이 가까워질 때,

$$\frac{dx}{dt} = -\bar{\alpha}'(t) \left(\frac{x(t)}{2\bar{\alpha}(t)} - \frac{\epsilon_\theta(x(t), t)}{2\bar{\alpha}(t)\sqrt{1 - \bar{\alpha}(t)}} \right). \quad (10)$$

↳ ODE를 봄

- Classical Numerical Method.

고전적인 method로 위의 ODE를 풀면 간단하지만, 속도가 놓아지면, noise↑.

→ 따라서 DDIM보다 비효율적.

∴ 일반적인 ODE와 (10)의 차이를 분석 후, diffusion과 numerical을 사용할 때 그에의 문제 분석

- 문제 1.

예외. 식 (10)이 제한된 영역에서만 well-defined.

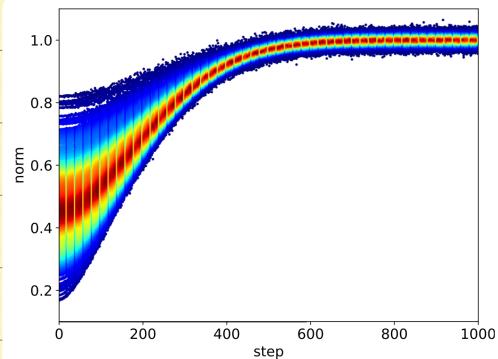
x_t 는 data density에서 초기 가우스 곡선을 따라서 생성된다.

그러나, 0.1 넓이의 band에 침투되어 있으면, 이는 특별 영역에서만 well-defined를 의미한다.

classical numeric은 허가 아닌 직선을 따라서 생성됨

이는 well-defined 밖에서 error를 야기하고

추가적으로 다른 문제를 야기함 → section 4.3.



- 문제 2.

식 (10)이 대부분의 문제에서 unbounded CL.

Figure 2: the density distribution of the norm of the data.

linear variance schedule은 식 (10)에서 $t \rightarrow 0$ 일수록 ∞ 가 되는 경향이 있음.

↳ section 2.3의 numerical의 조건을 만족x.

반대로, DDPM, DDIM에서는 $t \rightarrow 0$ 일수록 x_t 가 ϵ_0 의 predict가 정확해짐. → Appendix A.5

즉, numerical이든 오류가 발생할 수 있음

- Pseudo Numerical Method on Manifold.

문제. 1은 특정 manifold에서 문제를 해결해야 할 수 있음

Target manifold는 DDPM의 data x_t 의 high-density 영역.

↳ 기존 numerical은 x_0 와 reverse way

↳ ϵ 를 모르기 때문에 적용이 힘들

$$x_t(x_0, \epsilon) = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon$$

TILTM DDIM이 더 적합한 denoising process 설계 → 보고서 비슷한 글로 공유

1. gradient PL transfer를 분리

↳ gradient 설정 ↳ 다음 단계에서 보고 생략

ex) linear multi-step

$$\text{Gradient: } f' = \frac{\delta}{24} (55f_t - 59f_{t-\delta} + 37f_{t-2\delta} - 9f_{t-3\delta})$$

$$\text{Transfer: } x_{t+\delta} = x_t + \delta f'$$

모든 classical method는 linear transfer part는 다 같다. (gradient는 다른)

∴ pseudo numerical method로 nonlinear transfer를 사용하는 numerical method 정의.

expected transfer part는 gradient part가 보고가 정학하면,

transfer part의 결과는 최대한 manifold에 가깝다.

↳ 식(a)가 만족

Property 3.1.

x_t 와 ϵ 가 정학할 때, 식(a)에서의 $x_{t-\delta}$ 도 정학하다. — Appendix A.5.

$$\phi(x_t, \epsilon_t, t, t - \delta) = \frac{\sqrt{\bar{\alpha}_{t-\delta}}}{\sqrt{\bar{\alpha}_t}} x_t - \frac{(\bar{\alpha}_{t-\delta} - \bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t} (\sqrt{(1 - \bar{\alpha}_{t-\delta}) \bar{\alpha}_t} + \sqrt{(1 - \bar{\alpha}_t) \bar{\alpha}_{t-\delta}})} \epsilon_t \quad (11)$$

↳ 는 transfer part, ϵ_0 는 gradient part

즉 ϵ_0 가 최종 보고 생생의 방향을 결정한다. (= gradient part)

위의 결과는 2가지 문제 해결

1. transfer part \rightarrow 새로운 error 생성 x

2. reverse에서 ϕ \rightarrow 점점 정확해짐.

\rightarrow 더 정확한 transfer

$\phi(x_t, \epsilon_\theta(x_t, t), t, t-1)$ 는 DDIM의 일반화

- Gradient part

numerical을 두 부분으로 나누기 때문에 transfer를 번갈아도

다른 gradient 방법 사용 가능

- linear multi-step : $x_{t+\delta}, e_t = \text{PLMS}(x_t, \epsilon_{\text{PLMS}}, t, t+\delta)$

$$\left\{ \begin{array}{l} e_t = \epsilon_\theta(x_t, t) \\ e'_t = \frac{1}{24}(55e_t - 59e_{t-\delta} + 37e_{t-2\delta} - 9e_{t-3\delta}) \\ x_{t+\delta} = \phi(x_t, e'_t, t, t+\delta). \end{array} \right. \quad (12)$$

- Runge-kutta : $x_{t+\delta}, e'_t = \text{PRK}(x_t, t, t+\delta)$

$$\left\{ \begin{array}{l} e_t^1 = \epsilon_\theta(x_t, t) \\ x_t^1 = \phi(x_t, e_t^1, t, t + \frac{\delta}{2}) \\ e_t^2 = \epsilon_\theta(x_t^1, t + \frac{\delta}{2}) \\ x_t^2 = \phi(x_t, e_t^2, t, t + \frac{\delta}{2}) \\ e_t^3 = \epsilon_\theta(x_t^2, t + \frac{\delta}{2}) \\ x_t^3 = \phi(x_t, e_t^3, t, t + \delta) \\ e_t^4 = \epsilon_\theta(x_t^3, t + \delta) \\ e'_t = \frac{1}{6}(e_t^1 + 2e_t^2 + 2e_t^3 + e_t^4) \\ x_{t-\delta} = \phi(x_t, e'_t, t, t + \delta). \end{array} \right. \quad (13)$$

3) 미니 배치법을 개선한데 RK는 ϵ_0 에 비슷한거.

LMS는 Rkel 결과 같은 결과로 내면화. 비싼 ↓

∴ PNDM을 위해 LMS 사용

ϕ	order	first	non-first
linear		forward Euler	linear multi-step, Runge-Kutta...
nonlinear		DDIM	PNDM

Table 1: The relationship between different numerical methods.

- Algorithm.

LMS는 차울부터 사용 불가능이므로 처음 3개는 RK 혹은 LMS - F-PNDM.

그 이후 numerical gradient를 활용하는 것

- S-PNDM (Appendix A.3)

Algorithm 1 DDIMs

```
1:  $x_T \sim \mathcal{N}(0, I)$ 
2: for  $t = T - 1, \dots, 1, 0$  do
3:    $x_t = \phi(x_{t+1}, \epsilon_\theta(x_{t+1}, t+1), t+1, t)$ 
4: end for
5: return  $x_0$ 
```

Algorithm 2 PNDMs

```
1:  $x_T \sim \mathcal{N}(0, I)$ 
2: for  $t = T - 1, T - 2, T - 3$  do
3:    $x_t, e_t = \text{PRK}(x_{t+1}, t+1, t)$ 
4: end for
5: for  $t = T - 4, \dots, 1, 0$  do
6:    $x_t, e_t = \text{PLMS}(x_{t+1}, \{e_p\}_{p>t}, t+1, t)$ 
7: end for
8: return  $x_0$ 
```

- Convergence Order.

numerical el transfer 부분을 변경하면, 알 수 있는 error가 발생할 수 있다.

영향 분석을 위해 식(10)의 $x(t+\Delta)$ 의 결과와 제시 방법 사이의 local, global error 계산.

Property 3.2 S/F-PNDMs have third-order local error and are second-order convergent.

4. Experiments.

Pre-trained 를 사용

4.1 Sampling efficiency and quality

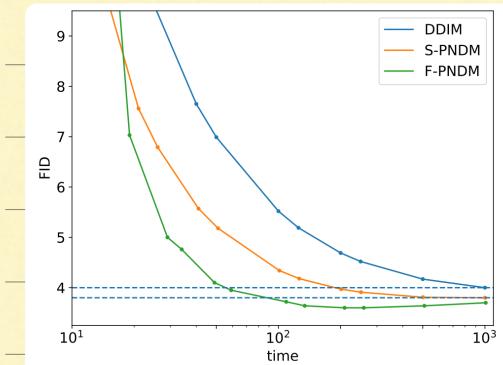
dataset	FID model	step 10	20	50	100	250	1000	time
Cifar10	DDIM	13.4	6.84	4.67	4.16	4.04		
	PF		13.8	3.89	3.69	3.71	3.72	
Cifar10 (linear)	DDIM*	18.5	10.9	6.99	5.52	4.52	4.00	0.337
	FON	13.1	7.41	5.26	4.65	4.12	3.71	0.390
	S-PNDM	11.6	7.56	5.18	4.34	3.91	3.80	0.344
	F-PNDM	7.03	5.00	3.95	3.72	3.60	3.70	0.391
Cifar10 (cosine)	DDIM	14.5	8.79	5.86	4.92	4.30	3.69	0.505
	S-PNDM	8.64	5.77	4.46	3.94	3.71	3.38	0.517
	F-PNDM	7.05	4.61	3.68	3.53	3.49	3.26	0.595
CelebA	DDIM	17.3	13.7	9.17	6.53	3.51		
CelebA (linear)	DDIM*	16.9	13.4	8.95	6.36	4.44	3.41	1.237
	FON	16.0	11.6	8.13	6.70	5.14	4.17	1.431
	S-PNDM	12.2	9.45	5.69	4.03	3.19	2.99	1.258
	F-PNDM	7.71	5.51	3.34	2.81	2.71	2.86	1.433

Table 2: Image generation measured in FID on Cifar10 and CelebA. PFs use black box ODE solvers and we use the number of score function evaluations as the step of PFs. DDIM* is a retest of DDIM. The bold results mean the best ones using the same pretrained model. We use the 50-step, 512 batch size experiment on an RTX-3090 to test the computational cost and the column time is the average computational cost per step in seconds. And we put the results of standard deviation in Appendix A.12

Efficiency

DDIM과 PF를 비교. (DDIM: pseudo numerical simple case, PF: classical numerical)

제시된 F/s-PNDM은 step 수에 관계없이 성능 향상



→ 계산 비용과 FID 그래프

Figure 3: The FID results under different computation costs and different numerical methods on Cifar10. The unit of time is the computational cost of 1-step DDIM, which is 0.337s.

Quality

불리는 step이 많으면 비슷하지만 FON은 더 빨리 성능이 낸다. (250step)

4.3 Sample on Manifolds.

Manifold 위에서 비교.

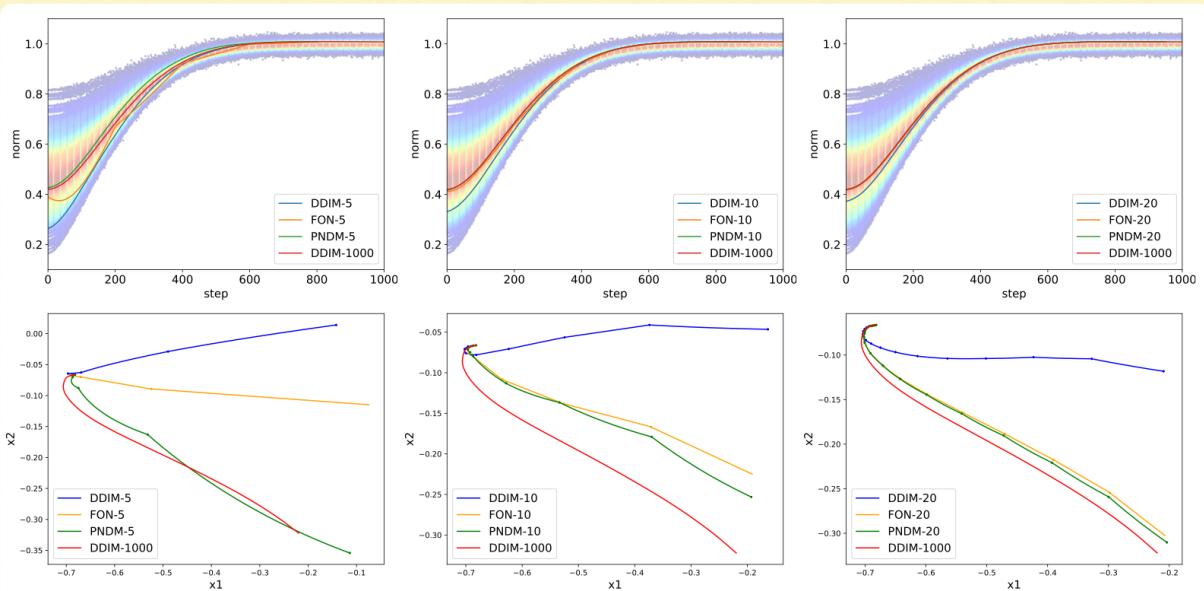


Figure 4: The upper part shows the change of norm with the number of steps using different methods and different steps. The lower part shows the generation curves of two points using different methods and different steps. DDIM-n means n-step DDIM method. Experiments in this subsection all use the Cifar10 dataset and we use the 1000-step DDIM's result as our target result.

→ PNDM01 DDIM-1000 블로우업 가까움 (Error↓)

5. Discussion.

DDPM이 ODE에 맞는 새로운 numerical method

→ 더 작은 stepsize high-quality

ODE를 transfer, gradient로 나눠 오류 줄임