

# Interpreting and improving diffusion models using the Euclidean distance function

Chenyang Yuan (Joint work with Frank Permenter)

Toyota Research Institute

Wednesday 4<sup>th</sup> October, 2023

# Introduction

Diffusion models achieve state-of-the-art results in multiple domains such as:



Image generation

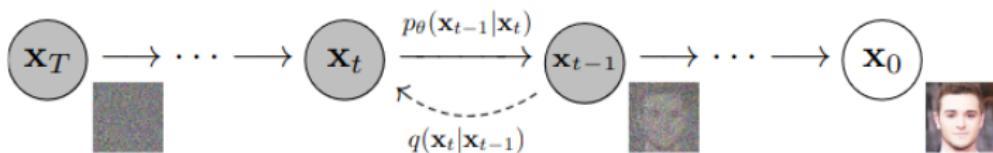


Trajectory planning

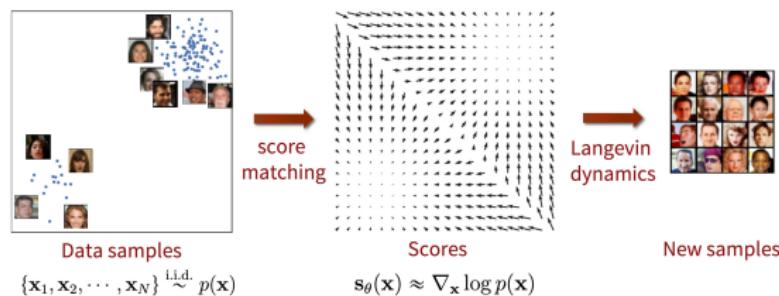
A powerful generative framework for sampling from multimodal distributions

# Motivation

Diffusion models are motivated by probabilistic models



Reversal of stochastic process that adds noise to data (Ho et.al. 2020)



Sampling of data distribution using score function (Song et.al. 2021)

However, commonly used sampling procedures (e.g. DDIM) are deterministic

# Motivation

Given commonly used diffusion training and sampling algorithms,

Is there a deterministic model that motivates the same algorithms?

Can we make reasonable assumptions on learned NN model to analyze performance of sampling algorithm?

Our optimization-based interpretation:

Denoising approximates projection under manifold hypothesis

Diffusion sampling finds projection to data manifold by minimizing distance via gradient descent

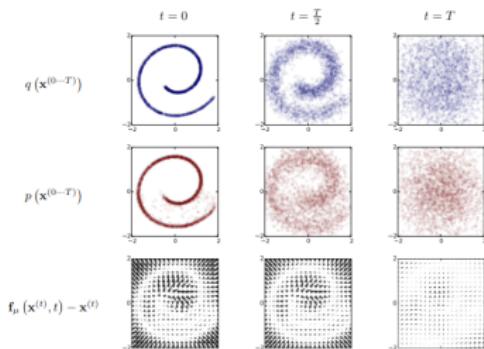
# Training diffusion models

Denoising diffusion models estimate a **noise vector**  $\epsilon \in \mathbb{R}^n$  from a given **noise level**  $\sigma > 0$  and noisy input  $x_\sigma \in \mathbb{R}^n$  such that for some  $x_0$  in the **data manifold**  $\mathcal{K}$ ,

$$x_\sigma \approx x_0 + \sigma\epsilon$$

A **denoiser**  $\epsilon_\theta : \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \mathbb{R}^n$  is learned by minimizing

$$L(\theta) := \mathbf{E}_{x_0, \sigma, \epsilon} \|\epsilon_\theta(x_0 + \sigma\epsilon, \sigma) - \epsilon\|^2$$



\*Note: to get expressions commonly used in literature, change of coordinates  $z_t = \sqrt{\alpha_t}x_t$ , where  $\sigma_t^2 = (1 - \alpha_t)/\alpha_t$ .

Visualization of training process

# Distance and projection

The *distance function*  $\text{dist}_{\mathcal{K}} : \mathbb{R}^n \rightarrow \mathbb{R}$  to a set  $\mathcal{K} \subseteq \mathbb{R}^n$ , is defined via

$$\text{dist}_{\mathcal{K}}(x) := \inf\{\|x - x_0\| : x_0 \in \mathcal{K}\}$$

The *projection* of  $x \in \mathbb{R}^n$ , is the set of points that attain this distance

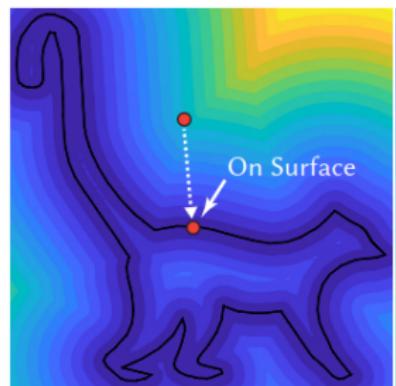
$$\text{proj}_{\mathcal{K}}(x) := \{x_0 \in \mathcal{K} : \text{dist}_{\mathcal{K}}(x) = \|x - x_0\|\}$$

Intuitively,  $x - \text{proj}_{\mathcal{K}}(x)$  is the direction of steepest descent (i.e. neg. gradient) of  $\text{dist}_{\mathcal{K}}(x)$ .

## Proposition

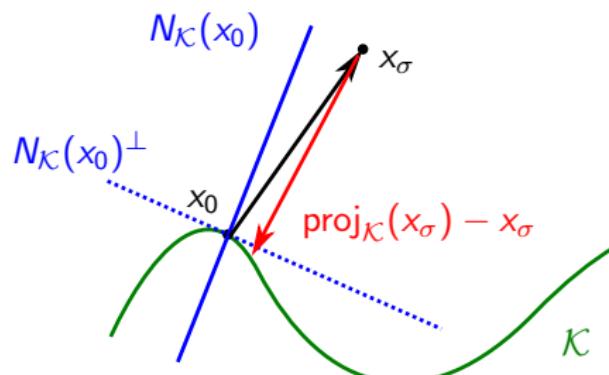
Suppose  $\mathcal{K} \subseteq \mathbb{R}^n$  is closed and  $x \notin \mathcal{K}$ . If  $\text{proj}_{\mathcal{K}}(x)$  is a singleton, then

$$\nabla_{\frac{1}{2}} \text{dist}_{\mathcal{K}}(x)^2 = x - \text{proj}_{\mathcal{K}}(x)$$



## Denoising approximates projection: Low noise

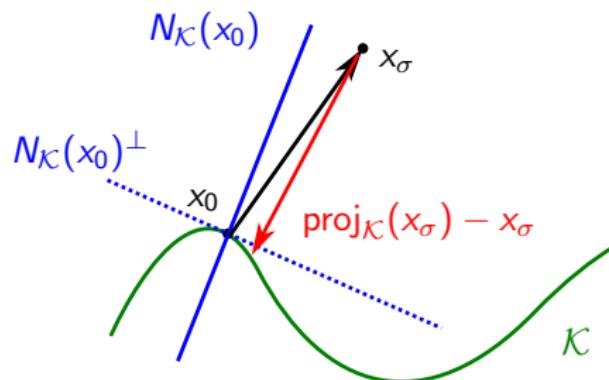
Manifold hypothesis: “real-world” datasets are (approximately) contained in low-dimensional manifolds  $\mathcal{K}$  of  $\mathbb{R}^n$ .



Given  $x_\sigma = x_0 + \sigma\epsilon$ , most of the added noise lies in  $N_{\mathcal{K}}(x_0)$  with high probability, thus denoising approximates projection

# Denoising approximates projection: Low noise

The *reach* of  $\mathcal{K}$  is the largest  $\tau$  so that  $\text{proj}_{\mathcal{K}}(x)$  is unique when  $\text{dist}_{\mathcal{K}}(x) \leq \tau$



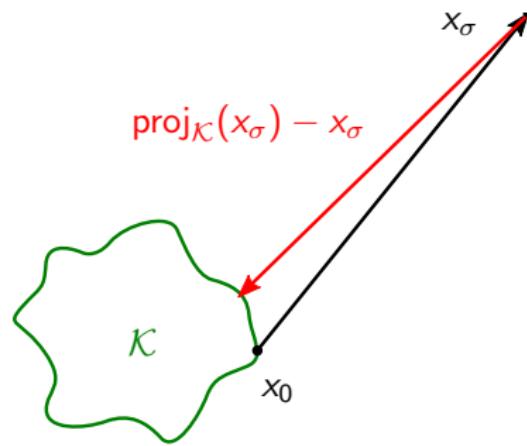
## Theorem

Fix  $\sigma > 0$  and suppose that  $\text{reach}(\mathcal{K}) \gtrsim \sigma\sqrt{n}$ . Given  $x_0 \in \mathcal{K}$  and  $\epsilon \sim \mathcal{N}(0, I)$ , let  $x_\sigma = x_0 + \sigma\epsilon$ . With high probability, we have:

$$\|\text{proj}_{\mathcal{K}}(x_\sigma) - x_0\| \lesssim \sigma\sqrt{d}.$$

## Denoising approximates projection: High noise

Diffusion models often add large levels of noise to  $x_0$  in training, in order to start sampling from a gaussian distribution

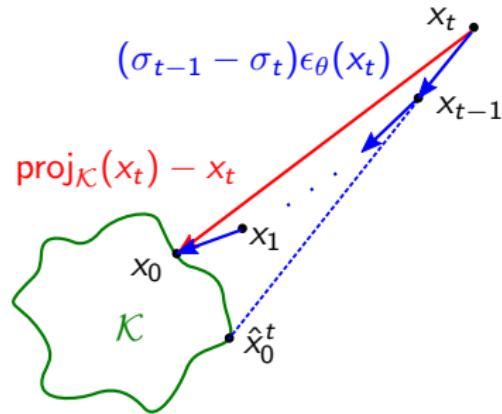


When  $\sigma$  is large, both denoising and projection point in the same direction towards  $\mathcal{K}$

# Sampling from diffusion models (Deterministic)

Given noisy  $x_\sigma$  and noise level  $\sigma$ , the learned denoiser  $\epsilon_\theta(x_\sigma, \sigma)$  estimates

$$x_0 \approx \hat{x}_0(x_\sigma, \sigma) := x_\sigma - \sigma \epsilon_\theta(x_\sigma, \sigma).$$



Sampling algorithms (e.g. DDIM) construct a sequence  $\hat{x}_0^t := \hat{x}_0(x_t, \sigma_t)$  of estimates from a sequence of points  $x_t$  using the update:

$$x_{t-1} = x_t + (\sigma_{t-1} - \sigma_t)\epsilon_\theta(x_t, \sigma_t)$$

# Sampling from diffusion models (Probabilistic)

Deterministic (DDIM) update:

$$x_{t-1} = x_t + (\sigma_{t-1} - \sigma_t)\epsilon_\theta(x_t, \sigma_t)$$

Probabilistic (DDPM) update:

$$x_{t-1} = x_t + (\sigma_{t'} - \sigma_t)\epsilon_\theta(x_t, \sigma_t) + \eta w_t$$

Where  $w_t \sim \mathcal{N}(0, I)$ ,  $\sigma_{t'} = \sigma_{t-1}^2 / \sigma_t$  and  $\eta = \sqrt{\sigma_{t-1}^2 - \sigma_{t'}^2}$  (to match norm of update in expectation)

Note:  $\sigma_{t-1} = \sqrt{\sigma_t \sigma_{t'}}$ , thus  $\sigma_{t'} < \sigma_{t-1} < \sigma_t$ .

These iterations look like gradient descent! But on which function?

# Our error model

Let  $f(x) := \frac{1}{2}\text{dist}_{\mathcal{K}}(x)^2$ . Intuitively,  $\nabla f(x) = x - \text{proj}_{\mathcal{K}}(x) \approx \text{dist}_{\mathcal{K}}(x)\epsilon_{\theta}(x)/\sqrt{n}$

**Assumption (Projection with relative error)**

*There exists  $\nu \geq 1$  and  $\eta \geq 0$  such that if  $\frac{1}{\nu}\text{dist}_{\mathcal{K}}(x) \leq \sqrt{n}\sigma_t \leq \nu\text{dist}_{\mathcal{K}}(x)$  and  $\nabla f(x)$  exists, then  $\|\sigma_t\epsilon_{\theta}(x, t) - \nabla f(x)\| \leq \eta\text{dist}_{\mathcal{K}}(x)$ .*

If  $\sqrt{n}\sigma_t$  closely tracks  $\text{dist}_{\mathcal{K}}(x)$ , then  $\sigma_t\epsilon_{\theta}(x, t)$  is approximately  $\nabla f(x_t)$

Relative error model where error depends on distance to  $\mathcal{K}$

Implications can be empirically tested on real datasets

**DDIM is approximate gradient descent on  $f$  with stepsize  $1 - \frac{\sigma_{t-1}}{\sigma_t}$ , with  $\nabla f(x_t)$  estimated by  $\epsilon_{\theta}(x_t, \sigma_t)$**

# Analysis under the error model

A schedule is  $\{\sigma_t\}_{t=0}^N$  is  $(\eta, \nu)$ -admissible when  $\sigma_t$  is decreased slow enough to maintain **relative error assumption**

Log-linear (geometrically decreasing) schedules are  $(\eta, \nu)$ -admissible

$$\sigma_{t-1} = (1 - \beta)\sigma_t, \text{ with } \beta < C(\eta, \nu)$$

We show convergence under the **relative error assumption**

## Theorem (DDIM with relative error)

Let  $x_t$  denote the sequence generated by DDIM and suppose that the gradient of  $f(x) := \frac{1}{2}\text{dist}_{\mathcal{K}}(x)^2$  exists for all  $x_t$ . Then for all  $t$ :

$$\frac{1}{\nu} \text{dist}_{\mathcal{K}}(x_t) \leq \sqrt{n}\sigma_t \leq \nu \text{dist}_{\mathcal{K}}(x_t),$$

$$\text{dist}_{\mathcal{K}}(x_N) \prod_{i=t}^N (1 - \beta_i(\eta + 1)) \leq \text{dist}_{\mathcal{K}}(x_{t-1}) \leq \text{dist}_{\mathcal{K}}(x_N) \prod_{i=t}^N (1 + \beta_i(\eta - 1)).$$

**Admissible schedule**  $\implies$  **Control of relative error**  $\implies$   $\text{dist}_{\mathcal{K}}$  decreases

# Improving sampling by gradient estimation

Our error model asserts that  $\epsilon_\theta(x, \sigma) \approx \sqrt{n} \nabla \text{dist}_\mathcal{K}(x)$  when  $\text{dist}_\mathcal{K}(x) \approx \sqrt{n}\sigma$ .

Since  $\nabla \text{dist}_\mathcal{K}(x)$  is *invariant* between  $x$  and  $\text{proj}_\mathcal{K}(x)$ , we aim to minimize estimation error  $\bar{\epsilon} - \epsilon_\theta(x_t, \sigma_t)$ , resulting in the update

$$\bar{\epsilon}_t = \gamma \epsilon_\theta(x_t, \sigma_t) + (1 - \gamma) \epsilon_\theta(x_{t+1}, \sigma_{t+1})$$

Replace  $\epsilon_\theta(x_t, \sigma_t)$  in sampling algorithm with  $\bar{\epsilon}_t$

Empirically,  $\gamma = 2$  achieves best results across many datasets and number of sampling steps

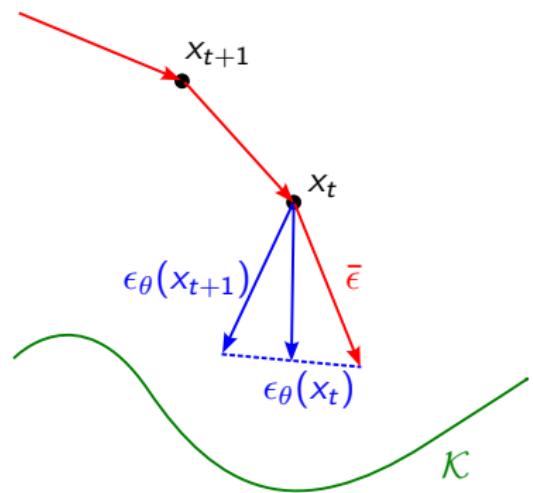


Illustration of our choice of  $\bar{\epsilon}_t$

# Improved sampling algorithm

Given  $(\sigma_N, \dots, \sigma_0)$ ,  $x_N \sim \mathcal{N}(0, I)$  and  $\epsilon_\theta$ , to compute  $x_0$  with  $N$  evaluations of  $\epsilon_\theta$ :

---

**Algorithm 1** DDIM sampler

---

```
for t = N, ..., 1 do
     $x_{t-1} \leftarrow x_t + (\sigma_{t-1} - \sigma_t) \epsilon_\theta(x_t, \sigma_t)$ 
return  $x_0$ 
```

---

---

**Algorithm 2** Our sampler

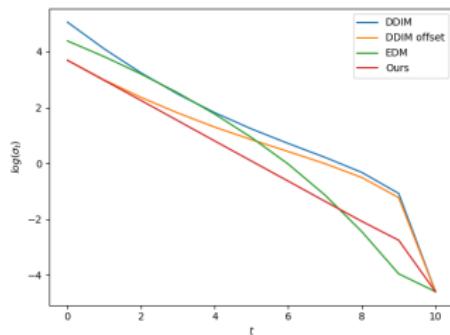
---

```
 $x_{N-1} \leftarrow x_N + (\sigma_{N-1} - \sigma_N) \epsilon_\theta(x_N, \sigma_N)$ 
for t = N - 1, ..., 1 do
     $\bar{\epsilon}_t \leftarrow 2\epsilon_\theta(x_t, \sigma_t) - \epsilon_\theta(x_{t+1}, \sigma_{t+1})$ 
     $x_{t-1} \leftarrow x_t + (\sigma_{t-1} - \sigma_t) \bar{\epsilon}_t$ 
return  $x_0$ 
```

---

# Experiments on noise schedule

How should we choose  $\sigma_t$ ? Relative noise model suggests log-linear schedule



Plot of different choices of  $\log(\sigma_t)$  for  $N = 10$ .

Schedule	CIFAR-10	CelebA
DDIM	16.86	18.08
DDIM Offset	14.18	15.38
EDM	20.85	16.72
Ours	<b>13.25</b>	<b>13.55</b>

FID scores of the DDIM sampler with different  $\sigma_t$  schedules on the CIFAR-10 model for  $N = 10$  steps.

# Sampler comparison experiments (Visual)

Visualizing  $\hat{x}_0^t$  throughout the denoising process:



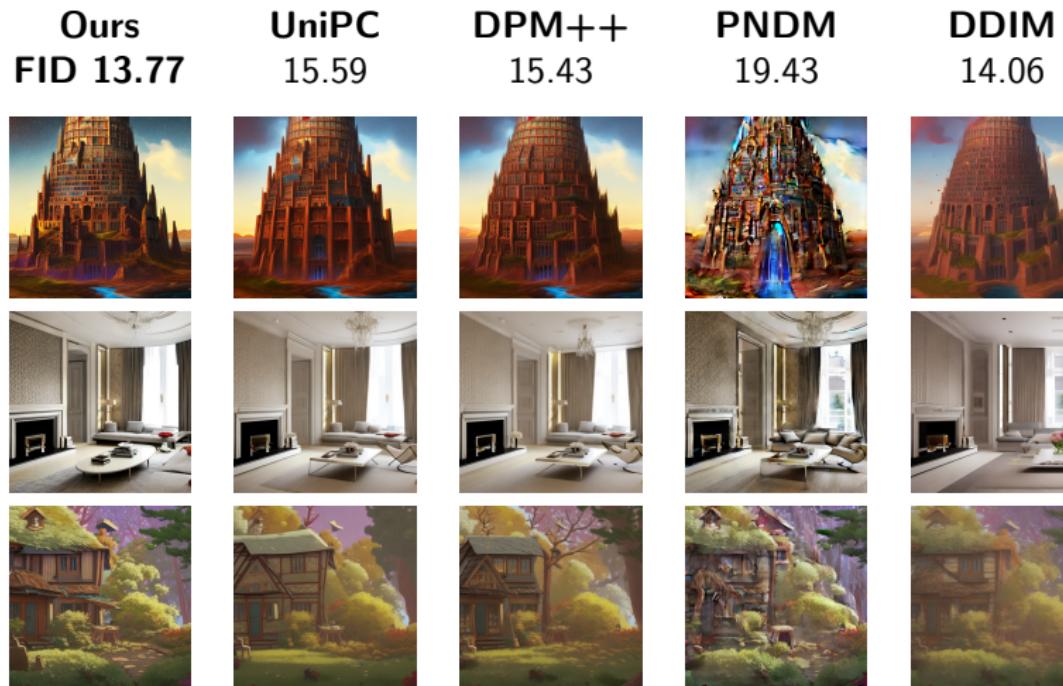
A comparison of our sampler with DDIM on the CelebA dataset with  $N = 5$  steps.

## Sampler comparison experiments (FID)

Sampler	CIFAR-10 FID				CelebA FID			
	$N = 5$	$N = 10$	$N = 20$	$N = 50$	$N = 5$	$N = 10$	$N = 20$	$N = 50$
Ours	<b>12.53</b>	<b>3.85</b>	<b>3.39</b>	<b>3.43</b>	<b>10.73</b>	<b>4.30</b>	3.56	3.78
DDIM	47.20	16.86	8.28	4.81	32.21	18.08	11.81	7.39
PNDM	13.9	7.03	5.00	3.95	11.3	7.71	5.51	3.34
DPM		6.37	3.72	<b>3.48</b>		5.83	<b>2.82</b>	<b>2.71</b>
DEIS	18.43	7.12	4.53	3.78	25.07	6.95	3.41	2.95
UniPC	23.22	<b>3.87</b>						
A-DDIM		14.00	5.81*	4.04		15.62	9.22*	6.13

FID scores of our sampler compared to that of other samplers for pretrained CIFAR-10 and CelebA models with a discrete linear schedule. \*Results for  $N = 25$

# Comparison on latent diffusion models



Example outputs on text-to-image Stable Diffusion when limited to  $N = 10$  function evaluations. FID scores for text-to-image generation on MS-COCO 30K.

# Conclusion

Elementary deterministic framework for analyzing and generalizing diffusion models

Simplified exposition of existing algorithms and methods

New fast and simple-to-implement sampler designed with our interpretation

Framework for incorporating ideas from optimization into diffusion models

Constraining diffusion models  $\leftrightarrow$  constrained optimization

Use diffusion models in optimization problems (e.g. as a regularizer for compressed sensing)