Denoising Diffusion Implicit Models

Kayla Bennett

University of Arizona

March 20, 2024

Paper Contributions

- DDIM: Implicit model trained with the same objective function as DDPMs
- Generalize the forward process from DDPMs to non-Markovian process
- Consider non-Markovian forward process to skip iterations during reverse process
- Much faster diffusion model with small impact on quality
- Noise in DDIM acts as a latent encoding, enabling reconstruction & interpolation

Background: DDPMs

- Approximate samples from distribution $q(x_0)$ using learned model $p_\theta(x_0)$
- Forward process: Markov chain $q(x_{t:T}|x_0)$ adds gaussian noise each step of T
- Generative process: $p_\theta(x_{0:T})$ samples intractable reverse process $q(x_{t-1}|x_t)$

$$p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}, \quad \text{where} \quad p_\theta(x_{0:T}) := p_\theta(x_T) \prod_{t=1}^{T} p_\theta^{(t)}(x_{t-1}|x_t)$$

- models are learned with a fixed inference procedure
- Parameters θ learn to fit $q(x_0)$ by maximizing the VLB:

$$\max_\theta \mathbb{E}_{q(\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0)] \leq \max_\theta \mathbb{E}_{q(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)} [\log p_\theta(\boldsymbol{x}_{0:T}) - \log q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)] \tag{2}$$

Background: DDPMs (2)

- Special property of forward process $q(x_t|x_0)$

$$q(x_t|x_0) := \int q(x_{1:t}|x_0)dx_{1:(t-1)} = \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)\mathbf{I})$$

- x_t is a linear combination of x_0 and noise ϵ

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\epsilon, \quad \text{where} \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$$

- Ast α_T approaches 0, $q(x_T|x_0)$ becomes pure gaussian noise
- We can sample x_T as pure Gaussian noise: $p_\theta(x_T) = \mathcal{N}(0, \mathbf{I})$

Background: DDPMs (3)

- Variational lower bound in equation 2 simplifies to:

$$L_\gamma(\epsilon_\theta) := \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \epsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\left\| \epsilon_\theta^{(t)}(\sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\epsilon_t) - \epsilon_t \right\|_2^2 \right] \tag{5}$$

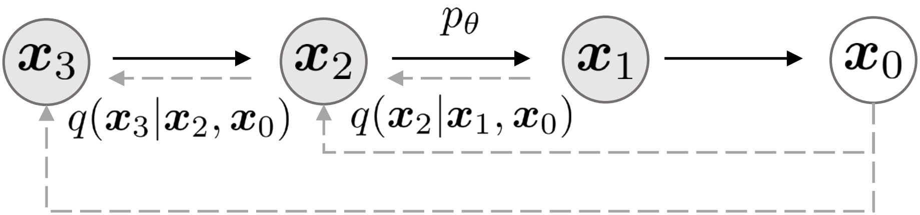
- ϵ_θ - set of learned gaussian noise functions for each time step
- γ - vector of positive variance coefficients that depend on α hyperparameter
- To sample x_0:
  1. sample x_T from $p_\theta(x_T)$ (just Gaussian noise)
  2. iteratively sample x_{t-1} from $p_\theta(x_{t-1}|x_t)$

Background: The Problem with DDPMs

- Number of iterations T is a hyperparameter
- A large T is needed to get a good approximation; T=1000 from Ho et al. (2020)
- Sampling from $p_\theta(x_{t-1}|x_t)$ means iterations can't be parallelized
- Main contribution of DDIMs paper: Sample $p_\theta(x_0)$ faster by making it non-Markovian!

Variational Inference for Non-Markovian Forward Processes

- Inference (forward) process iteratively adds noise, generative process reverses it
- To make the reverse process non-Markovian, define the forward process to be non-Markovian
- Key observation: objective L_γ depends directly on marginals $q(x_t|x_0)$ but not on joint $q(x_{1:T}|x_0)$
- Many joints have the same marginals, use this fact to define non-Markovian inference process below



Defining a Non-Markovian Forward Process

- consider family Q of inference distributions
- index family by vector $\sigma \in \mathbb{R}^T_{\geq 0}$

$$q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) := q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0) \prod_{t=2}^{T} q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \tag{6}$$

where $q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\alpha_T}\boldsymbol{x}_0, (1-\alpha_T)\boldsymbol{I})$ and for all $t > 1$,

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 \boldsymbol{I}\right). \tag{7}$$

- $q_\sigma(x_t|x_0) = \mathcal{N}(\sqrt{\alpha_t}x_0, (1-\alpha_t)I)$ for all t
- Each x_t depends on x_0 and our noise parameters
- Define whole forward process from Bayes rule

$$q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = \frac{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}$$

Generative process and Unified Variational Inference Objective

- Define trainable $p_\theta(x_{0:T})$ where $p_\theta(x_{t-1}|x_t)$ leverages $q_\sigma(x_{t-1}|x_t, x_0)$
- Given x_t:
  1. Predict x_0 using equation 4
  2. Use predicted x_0 and noise ϵ_t in $q_\sigma(x_{t-1}|x_t, x_0)$ to sample x_{t-1}
- Model $\epsilon_\sigma^{(t)}$ predicts ϵ_t from x_t

Generative Process (2)

- Predict x_0 using equation 4, and define generative process:

$$f_\theta^{(t)}(\boldsymbol{x}_t) := (\boldsymbol{x}_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta^{(t)}(\boldsymbol{x}_t))/\sqrt{\alpha_t}. \tag{9}$$

We can then define the generative process with a fixed prior $p_\theta(\boldsymbol{x}_T) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and

$$p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \begin{cases} \mathcal{N}(f_\theta^{(1)}(\boldsymbol{x}_1), \sigma_1^2 \boldsymbol{I}) & \text{if } t = 1 \\ q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, f_\theta^{(t)}(\boldsymbol{x}_t)) & \text{otherwise,} \end{cases} \tag{10}$$

- Optimize θ parameter as VLB on ϵ_θ:

$$J_\sigma(\epsilon_\theta) := \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q_\sigma(\boldsymbol{x}_{0:T})}[\log q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) - \log p_\theta(\boldsymbol{x}_{0:T})] \tag{11}$$

$$= \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q_\sigma(\boldsymbol{x}_{0:T})} \left[\log q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0) + \sum_{t=2}^{T} \log q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) - \sum_{t=1}^{T} \log p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) - \log p_\theta(\boldsymbol{x}_T) \right]$$

Denoising Diffusion Implicit Models

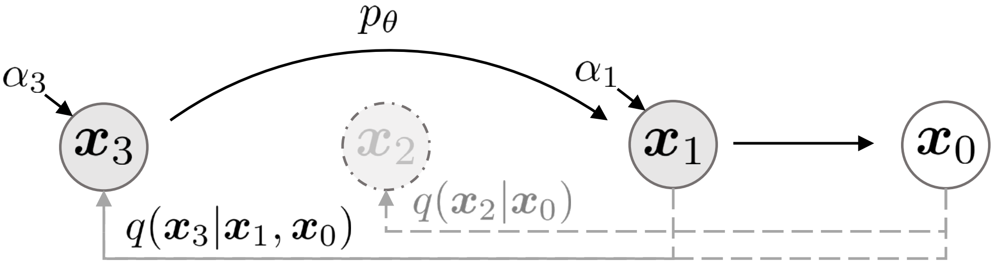
- From $p_\theta(x_{1:T})$ above, generate x_{t-1} from x_t as:

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{" predicted } \boldsymbol{x}_0\text{"}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{"direction pointing to } \boldsymbol{x}_t\text{"}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}} \tag{12}$$

- Changing σ results in a different generative process
- 2 special cases:
  1. $\sigma_t = \sqrt{(1-\alpha_{t-1}/(1-\alpha))}\sqrt{1-\alpha_t/\alpha_{t-1}}$, markovian DDPM
  2. $\sigma_t = 0$ for all t results in a deterministic forward process becomes deterministic except when $t = 1$
     - model becomes an implicit probablistic model, which the authors call DDIM
     - Forward process is no longer a diffusion
     - Samples generated from x_T using a fixed generative process
     - Since the generative process is fixed, we can think of x_T as an encoding of x_0

Accelerated Generation Process

- With $q_\sigma(x_t|x_0)$ fixed, L doesn't depend on the specific forward process
- This means we can skip some iterations when sampling
- Define τ as the sequence of iterations we actually run, call its length S
- Refer to reversed(τ) as the sampling trajectory
- Now we can train with many steps in the forward process, but only sample some of those steps in the generative process



Above: Generation model when $\tau = [1,3]$

Relation to Neural ODEs

- Rewriting eq. 12 shows similarity to Euler Integration:

$$\frac{\boldsymbol{x}_{t-\Delta t}}{\sqrt{\alpha_{t-\Delta t}}} = \frac{\boldsymbol{x}_t}{\sqrt{\alpha_t}} + \left(\sqrt{\frac{1-\alpha_{t-\Delta t}}{\alpha_{t-\Delta t}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}} \right) \epsilon_\theta^{(t)}(\boldsymbol{x}_t) \tag{13}$$

- DDIM is basically solving this ODE:

$$\mathrm{d}\bar{\boldsymbol{x}}(t) = \epsilon_\theta^{(t)} \left(\frac{\bar{\boldsymbol{x}}(t)}{\sqrt{\sigma^2+1}} \right) \mathrm{d}\sigma(t), \tag{14}$$

- with initial condition $x(T) \sim \mathcal{N}(0, \sigma(T))$
- Suggests that DDIM can obtain latent x_T and reconstruct x_0

Experiments

- Show that DDIMs produce similar quality images as DDPMs in less time
  - Asses sample quality using Frechet Inception Distance (FID)
  - Lower is better
- Demonstrate that DDIMs can interpolate directly from latent space since generative process is fixed
  - DDPMs can't do this due to stochasticity
- Evaluate DDIM ability to reconstruct CIFAR-10 images

Experiment Setup

- Authors use same trained model for each dataset, with $T = 1000$, $\gamma = 1$ for all experiments
- Authors only change τ and σ during experiments
- define hyperparameter "stochastity" η to manipulate σ_τ

$$\sigma_{\tau_i}(\eta) = \eta\sqrt{(1-\alpha_{\tau_{i-1}})/(1-\alpha_{\tau_i})}\sqrt{1-\alpha_{\tau_i}/\alpha_{\tau_{i-1}}}$$

- Note: $\eta = 1$ case and $\hat{\sigma}$ case are DDPMs, $\eta = 0$ case is the DDIM
- $\hat{\sigma}$ - DDPM with standard deviation $>$1
- Details in appendix D

Results: FID scores with changing τ and η

Table 1: CIFAR10 and CelebA image generation measured in FID. $\eta = 1.0$ and $\hat{\sigma}$ are cases of DDPM (although Ho et al. (2020) only considered $T = 1000$ steps, and $S < T$ can be seen as simulating DDPMs trained with S steps), and $\eta = 0.0$ indicates DDIM.

	S	CIFAR10 (32 × 32) 10	20	50	100	1000	CelebA (64 × 64) 10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
$\hat{\sigma}$		367.43	133.37	32.72	9.99	**3.17**	299.71	183.83	71.71	45.20	**3.26**

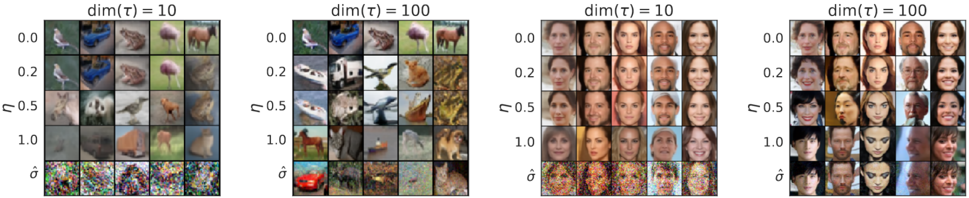
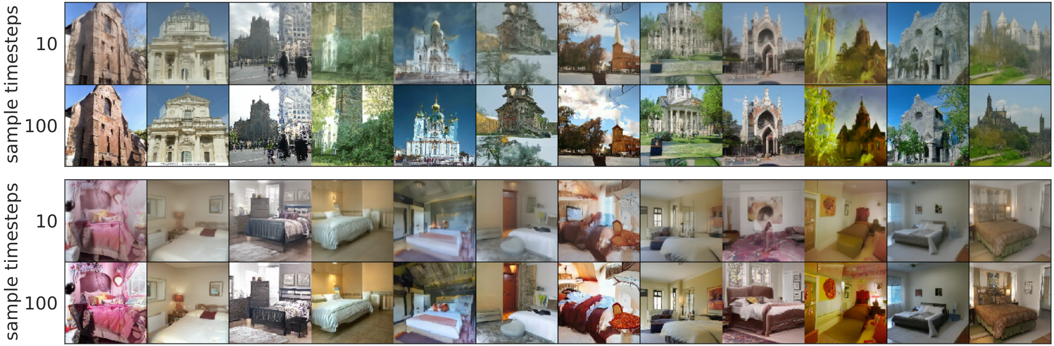


Figure 3: CIFAR10 and CelebA samples with $\dim(\tau) = 10$ and $\dim(\tau) = 100$.

Results: Image Quality and Consistency at Different Timesteps



- Starting from the same x_T produces similar high-level features, sample iterations seem to just add detail
- Strong evidence that x_T is actually a latent encoding of x_0

Results: Compute Time

- Compute time scales linearly with number of sampling steps

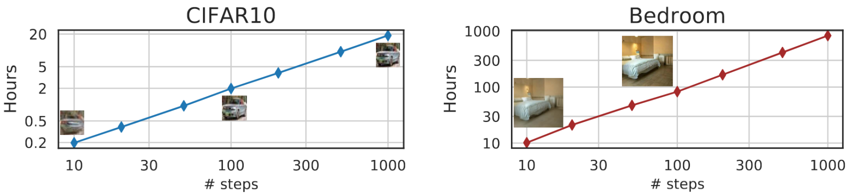


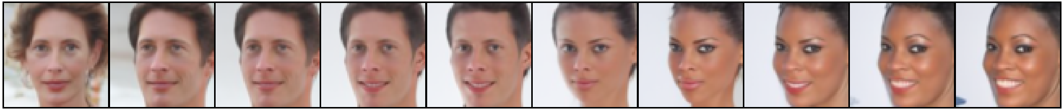
Figure 4: Hours to sample 50k images with one Nvidia 2080 Ti GPU and samples at different steps.

Results: Sample Quality

- Increasing $\dim(\tau)$ gives better results, as expected
- with low $\dim(\tau)$, $\eta = 0$ gives best results
- DDIM does much better than DDPM with fewer sampling steps
- Sampling time scales linearly

Results: Interpolation

- If x_T is a latent encoding, we can perturb it to interpolate between two samples



Results: CIFAR-10 Sample Reconstruction

Table 2: Reconstruction error with DDIM on CIFAR-10 test set, rounded to 10^{-4}.

S	10	20	50	100	200	500	1000
Error	0.014	0.0065	0.0023	0.0009	0.0004	0.0001	0.0001

- evaluation metric: per-dimension MSE

Questions?