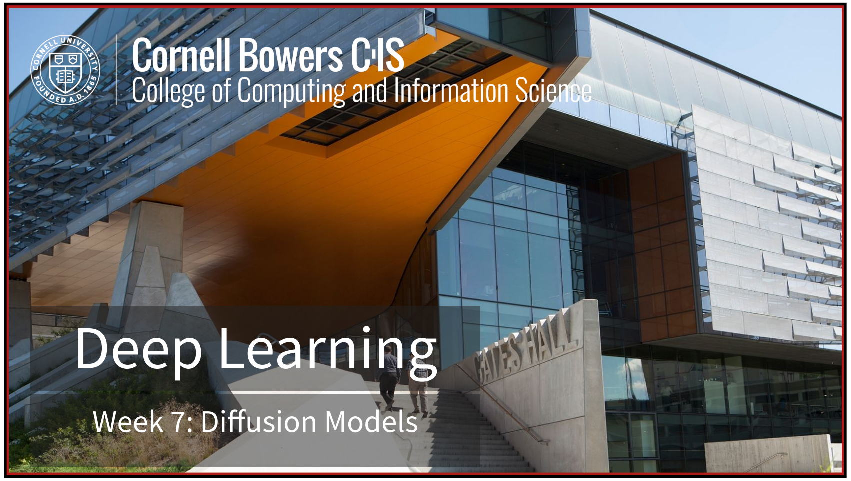


Cornell Bowers C·IS

Logistics

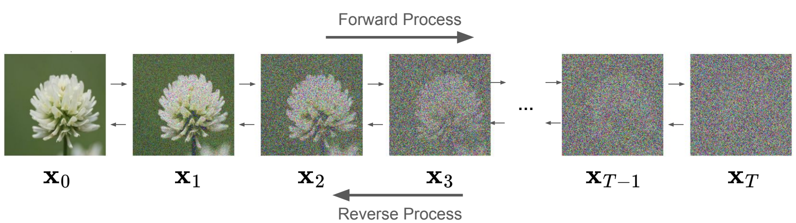
- Homework 4
  - Due Sunday March 24th (still have two slip days
  - Shorter assignment on VAEs and Diffusion
  - Pinned posts
- Feedback form for HW3 and associated content is due on tomorrow
- Reselased project feedback and HW2 feedback
  - Check graded documents for feedback!
- Midterm next Thursday (03/28)
  - During regular class time
  - Alternate time offered Wednesday at (03/27)
  - Location and time will be posted on Ed

Cornell Bowers C·IS

Denoising Diffusion Models

Denoising diffusion models consist of two processes:

- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising

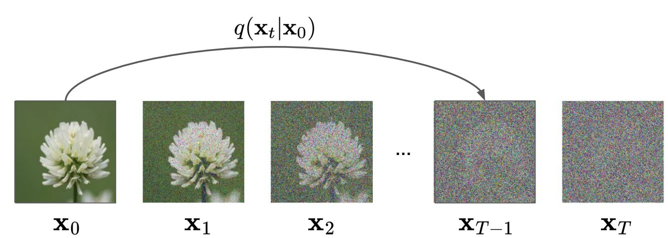


Cornell Bowers C·IS

Details: Forward Process

Can sample $\mathbf{x}_t$ in closed-form as $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \bar{\alpha}_t \in (0,1)$$



Cornell Bowers CIS

Aside: Noise Schedules

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \bar{\alpha}_t \in (0,1)$$

- Define the noise schedule in terms of $\bar{\alpha}_t \in (0,1)$
  - Some monotonically decreasing function from 1 to 0
- Cosine Noise schedule:

$$\bar{\alpha}_t = \cos(.5\pi t/T)^2$$

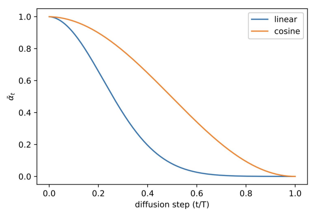


Figure 5. $\bar{\alpha}_t$ throughout diffusion in the linear schedule and our proposed cosine schedule.

Nichol, Alexander Quinn, and Prafulla Dhariwal. "Improved denoising diffusion probabilistic models." International conference on machine learning. PMLR, 2021.

Cornell Bowers CIS

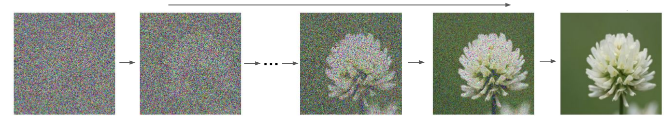
Key Idea

We introduce a generative model to approximate the reverse process:

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$



$\mathbf{x}_T$ $\mathbf{x}_{T-1}$ $\mathbf{x}_2$ $\mathbf{x}_1$ $\mathbf{x}_0$

Cornell Bowers CIS

Key Idea

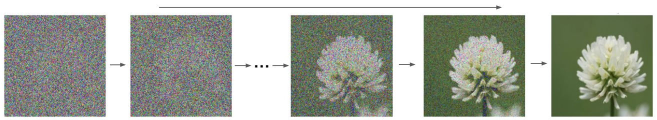
We introduce a generative model to approximate the reverse process:

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Learning Objective! $\boxed{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]}$

$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$



$\mathbf{x}_T$ $\mathbf{x}_{T-1}$ $\mathbf{x}_2$ $\mathbf{x}_1$ $\mathbf{x}_0$

Cornell Bowers CIS

Training: Principled Derivation

Find the model that maximizes the likelihood of the training data

i.e. same as VAEs, variational inference; approximate the true posterior

Cornell Bowers CIS

Training Objective

- Bound the likelihood with the ELBO
  - Exactly like hierarchical VAEs

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0)\parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T}\underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]}_{\text{denoising matching term}}$$

Cornell Bowers CIS

Training Objective

- Bound the likelihood with the ELBO
  - Exactly like hierarchical VAEs

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

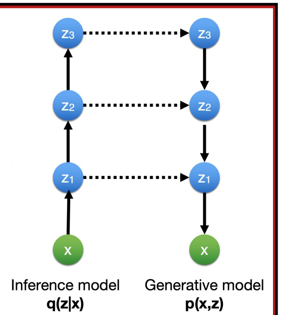
$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0)\parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T}\underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]}_{\text{denoising matching term}}$$

Discuss: How does the diffusion model construction help minimize the first two terms?

Cornell Bowers CIS

Discuss:

Differences between diffusion models and hierarchical VAEs?



$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

Learning Objective!

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0)\parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \boxed{\sum_{t=2}^{T}\underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]}_{\text{denoising matching term}}}$$

Cornell Bowers CIS

Training Objective

- Bound the likelihood with the ELBO
  - Exactly like VAEs

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

Learning Objective!

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0)\parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \boxed{\sum_{t=2}^{T}\underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]}_{\text{denoising matching term}}}$$

where $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ is the tractable posterior distribution:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1};\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0),\tilde{\beta}_t\mathbf{I}),$$

$$\text{where } \tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) \text{ and } \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

Cornell Bowers C·IS

Parameterizing the Denoising Model

Since both $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ are Normal distributions, the KL divergence has a simple form:

$$L_{t-1} = D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}||\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0)-\mu_\theta(\mathbf{x}_t,t)||^2\right]+C$$

Recall that $\mathbf{x}_t=\sqrt{\bar\alpha_t}\;\mathbf{x}_0+\sqrt{(1-\bar\alpha_t)}\;\epsilon$. Ho et al. NeurIPS 2020 observe that:

$$\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0)=\frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar\alpha_t}}\epsilon\right)$$

They propose to represent the mean of the denoising model using a *noise-prediction* network:

$$\mu_\theta(\mathbf{x}_t,t)=\frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar\alpha_t}}\boxed{\epsilon_\theta(\mathbf{x}_t,t)}\right)$$

With this parameterization

$$L_{t-1}=\mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar\alpha_t)}||\epsilon-\boxed{\epsilon_\theta(\underbrace{\sqrt{\bar\alpha_t}\;\mathbf{x}_0+\sqrt{1-\bar\alpha_t}\;\epsilon}_{\mathbf{x}_t},t)}||^2\right]+C$$

Cornell Bowers C·IS

$$\mathbf{x}_t=\sqrt{\bar\alpha_t}\mathbf{x}_0+\sqrt{1-\bar\alpha_t}\epsilon,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})$$

Training Objective Weighting

ELBO objective leads to a specific regression weight at each time step:

$$L_{t-1}=\mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\underbrace{\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)\boxed{(1-\bar\alpha_t)}}}_{\lambda_t}||\epsilon-\epsilon_\theta(\sqrt{\bar\alpha_t}\;\mathbf{x}_0+\sqrt{1-\bar\alpha_t}\;\epsilon,t)||^2\right]$$

Approaches zero!

However, this weight is often very large for small t's

Ho et al., 2020 proposed the following objective to improve perceptual quality:

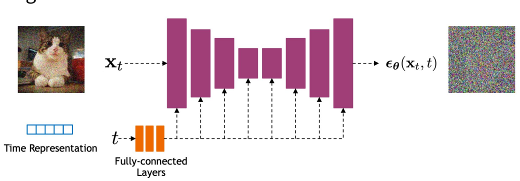
$$L_{\mathrm{simple}}=\mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(1,T)}\left[||\epsilon-\epsilon_\theta(\underbrace{\sqrt{\bar\alpha_t}\;\mathbf{x}_0+\sqrt{1-\bar\alpha_t}\;\epsilon}_{\mathbf{x}_t},t)||^2\right]$$

Cornell Bowers C·IS

What Network Architecture to Use For ϵ_θ ?

People often use U-Nets with residual blocks and self-attention layers at low resolutions

Has same input and output image dimensions



Time representation: sinusoidal positional embeddings

Inject time embedding throughout the network (e.g. additive positional embedding)

Cornell Bowers C·IS

Diffusion Results

Outperforms prior generative models when using the **simplified** training objective

ELBO objective performs worse!

Model	IS	FID
Gated PixelCNN [59]	4.60	65.93
Sparse Transformer [7]		
PixelIQN [43]	5.29	49.46
EBM [11]	6.78	38.2
NCSNv2 [56]		31.75
NCSN [55]	8.87±0.12	25.32
SNGAN [39]	8.22±0.05	21.7
SNGAN-DDLS [4]	9.09±0.10	15.42
StyleGAN2 + ADA (v1) [29]	**9.74** ± 0.05	3.26
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67±0.13	13.51
Ours (L_{simple})	9.46±0.11	**3.17**

ELBO

$$L_{\mathrm{simple}}=\mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(1,T)}\left[||\epsilon-\epsilon_\theta(\underbrace{\sqrt{\bar\alpha_t}\;\mathbf{x}_0+\sqrt{1-\bar\alpha_t}\;\epsilon}_{\mathbf{x}_t},t)||^2\right]$$

Ho et al. 2020

Cornell Bowers C·IS

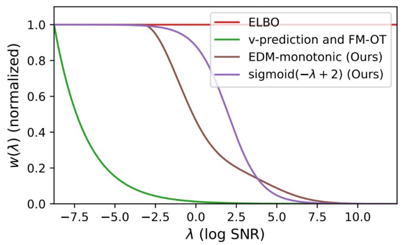
Training Objective Weighting

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathrm{SNR}(t) = \bar{\alpha}_t/(1-\bar{\alpha}_t)$$

$$\log(\mathrm{SNR}(t)) = \log(\bar{\alpha}_t/(1-\bar{\alpha}_t))$$

- ELBO forces the network to model imperceptible details
  - Less modeling capacity dedicated to perceptible details (global image structure, etc.)
- If you care about perceptual quality:
  - Decrease the loss weighting for low noise levels



Kingma, Diederik, and Ruiqi Gao. "Understanding diffusion objectives as the ELBO with simple data augmentation." Advances in Neural Information Processing Systems 36 (2023).

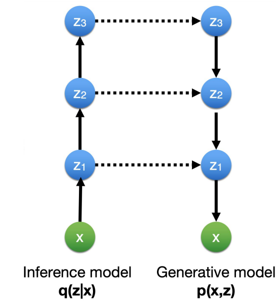
Cornell Bowers C·IS

Connection to VAEs

Diffusion models can be considered as a special form of hierarchical VAEs.

However, in diffusion models:

- The inference model is fixed: easier to optimize
- The latent variables have the same dimension as the data.
- The ELBO is decomposed to each time step: fast to train
- Can be made extremely deep (even infinitely deep)
- The model is trained with some reweighting of the ELBO
  - Can trade off likelihood for improved perceptual quality



Cornell Bowers C·IS

Alternative Diffusion Parameterization: Data Prediction

Can also view the diffusion network as learning to predict the **original data**

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

$$\Longrightarrow \mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon}{\sqrt{\bar{\alpha}_t}}$$

$$\Longrightarrow \mathbf{x}_\theta(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$$

Cornell Bowers C·IS

Alternative Diffusion Parameterization: Data Prediction

Can also view the diffusion network as learning to predict the **original data**

$$\mathbf{x}_\theta(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$$

Diffusion training objective: $\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]$

For sampling, want $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$, but don't have access to the original data

Use our estimate of the original data, $\mathbf{x}_\theta(\mathbf{x}_t, t)$, to sample:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \mathbf{x}_\theta(\mathbf{x}_t, t)) \approx q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \mathbf{x}_0)$$

Cornell Bowers CIS

$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Training Algorithm

Repeat until convergence

1. $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ ← Sample original image from image distribution
2. $t \sim U\{1, 2, \ldots, T\}$ ← Sample random time step uniformly
3. $\epsilon \sim \mathcal{N}(0, 1)$ ← Sample Gaussian noise
4. Optimizer step on $L(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2]$

← Model predicts noise applied at time step t and calculate loss

Cornell Bowers CIS

Sampling Algorithm

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ← Sample pure Gaussian noise

For $t = T, T-1 \ldots, 1$

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$ ← Sample Gaussian noise to apply to image

$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t\mathbf{z}$ ← Predict noise applied to image and remove that noise

$$\boxed{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \mathbf{x}_\theta(\mathbf{x}_t, t))}$$

Return $\mathbf{x}_0$

Cornell Bowers CIS

Generative Modeling

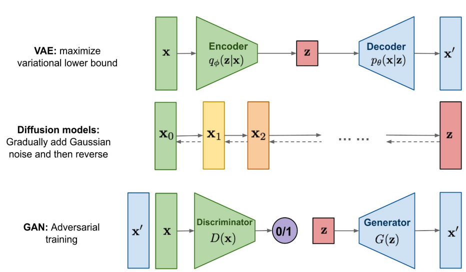


Image Source

Cornell Bowers CIS

Generative Modeling

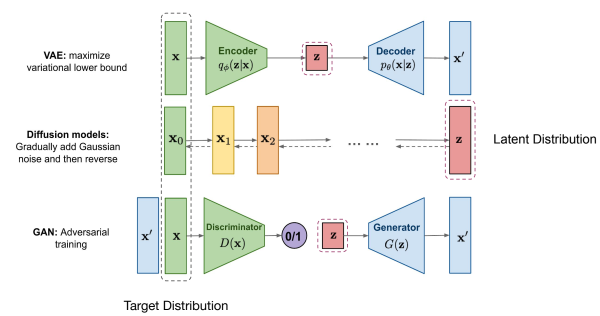


Image Source

Generative Modeling

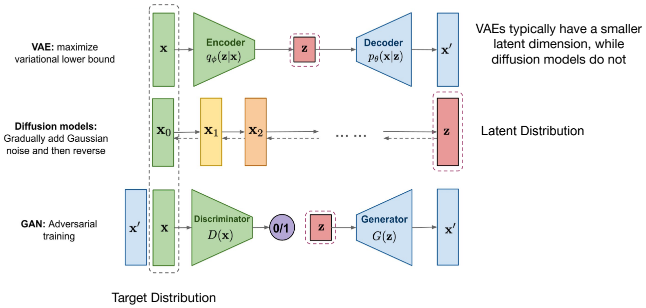
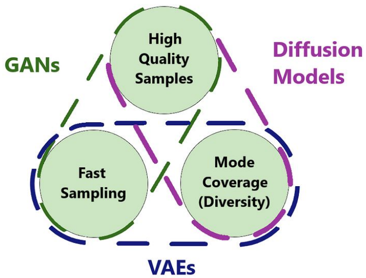


Image Source

Cornell Bowers C·IS

Diffusion Models vs. VAEs vs. GAN

Source



Cornell Bowers C·IS

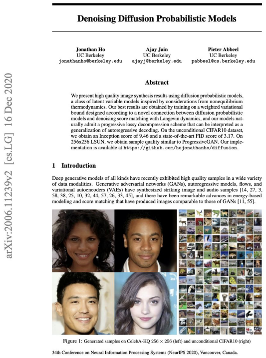
Stable Diffusion Demo!

https://huggingface.co/spaces/stabilityai/stable-diffusion

Sample input: "messi as a real madrid player"

Cornell Bowers C·IS

Two Perspectives



arXiv:2006.11239v2 [cs.LG] 16 Dec 2020

Denoising Diffusion Probabilistic Models

Jonathan Ho, UC Berkeley — Ajay Jain, UC Berkeley — Pieter Abbeel, UC Berkeley

Abstract

1 Introduction

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

arXiv:1907.05600v3 [cs.LG] 10 Oct 2020

Generative Modeling by Estimating Gradients of the Data Distribution

Yang Song, Stanford University — Stefano Ermon, Stanford University

Abstract

1 Introduction

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Score-based Models

Would like to model the probability density function as follows:

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta}$$

Discuss: Any problems with directly modelling this?

Score-based Models

Would like to model the probability density function as follows:

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta}$$

Want to maximize the log-likelihood of the data:

$$\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{x}_i).$$

Instead approximate the score function:

$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} f_\theta(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z_\theta}_{=0} = -\nabla_{\mathbf{x}} f_\theta(\mathbf{x})$$

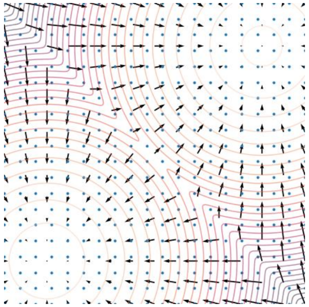
Loss function

Predicted score!

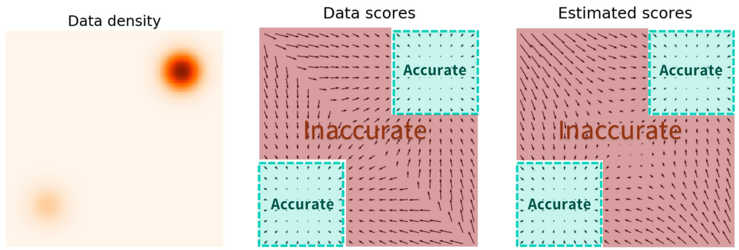
unknown!

$$\mathbb{E}_{\mathbf{x}}\left[\|\boxed{\mathbf{s}_\theta(\mathbf{x})} - \boxed{\nabla_{\mathbf{x}} \log p(x)}\|_2^2\right]$$

Score-based Models

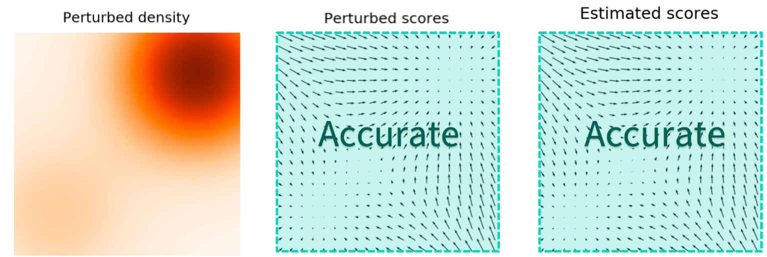


Cornell Bowers C·IS



Cornell Bowers C·IS

Add noise!



Cornell Bowers C·IS

$\mathbf{x_t} \sim \mathcal{N}(\mathbf{x}, \sigma_t^2 I)$

Training

Training Objective for noise level t:

$$\sum_{t=1}^{T} \lambda(t)\mathbb{E}_{\mathbf{x},t}[||\mathbf{s}_\theta(\mathbf{x_t}, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x_t})||_2^2]$$

Using results from denoising score matching [1]:

$$\sum_{t=1}^{T} \lambda(t)\mathbb{E}_{\mathbf{x},t}[||\mathbf{s}_\theta(\tilde{\mathbf{x}}, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x})||_2^2]$$

Using the definition of the pdf of a gaussian,

$$\sum_{t=1}^{T} \lambda(t)\mathbb{E}_{\mathbf{x},t}[||\mathbf{s}_\theta(\mathbf{x}_t, t) - \frac{\mathbf{x}_t - \mathbf{x}}{\sigma_t^2}||_2^2]$$

[1] P. Vincent. A connection between score matching and denoising autoencoders. Neural computation, 23(7):1661–1674, 2011.

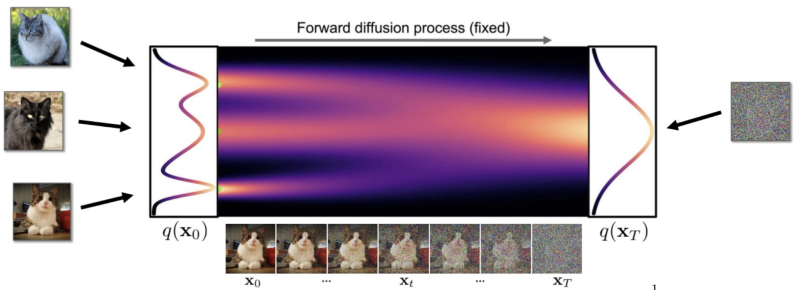
Cornell Bowers C·IS

Discuss: How does this training objective relate to that of DDPMs?

$$\sum_{t=1}^{T} \lambda(t)\mathbb{E}_{\mathbf{x},t}[||\mathbf{s}_\theta(\mathbf{x}_t, t) - \frac{\mathbf{x}_t - \mathbf{x}}{\sigma_t^2}||_2^2]$$

Cornell Bowers CIS

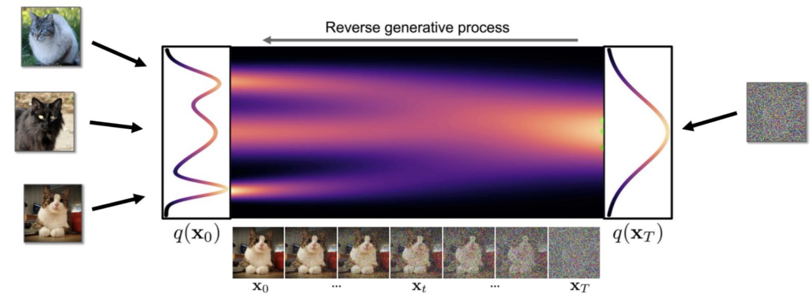
Continuous time (Stochastic Differential Equation Perspective)



$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Cornell Bowers CIS

Continuous time (Stochastic Differential Equation Perspective)



Score from network

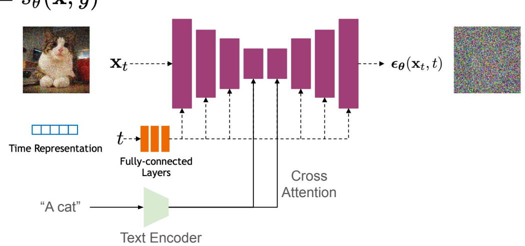
$$\mathrm{d}\mathbf{x}_t = \left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)\right]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t$$

Cornell Bowers CIS

Conditional Diffusion

- We want to condition on images or text
- Learn a conditional diffusion model

$$\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{y}) = s_\theta(\mathbf{x}, y)$$



Cornell Bowers CIS

Conditional Diffusion with Classifier Guidance

- May not have access to paired data for training
- Use Bayes’ rule to decompose the conditional score into the unconditional score and a likelihood term

$$\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t}\log p_t(\mathbf{y}|\mathbf{x}_t)$$

- Only need to train a classifier on noised data

Cornell Bowers C·IS

Unconditional Model

Conditional Model

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, *34*, 8780-8794.

Cornell Bowers C·IS

Classifier-Free Guidance

- Train a joint conditional and unconditional diffusion model
- Conditioning information is added by concatenating to input or cross attending
- Modified conditional distribution

$$\log \tilde{p}_t(\mathbf{x}_t|\mathbf{y}) \propto \ p_t(\mathbf{x}_t|\mathbf{y})\, p_t(\mathbf{y}|\mathbf{x}_t)^w$$

- Conditional sampling

$$\nabla_{\mathbf{x}_t} \log \tilde{p}_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log\ p_t(\mathbf{x}_t) + w(\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t))$$

Cornell Bowers C·IS

Classifier-Free Guidance

Significantly improves quality of conditional models

Used by practically **every** conditional diffusion model

Increasing Guidance

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., ... & Norouzi, M. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, *35*, 36479-36494.

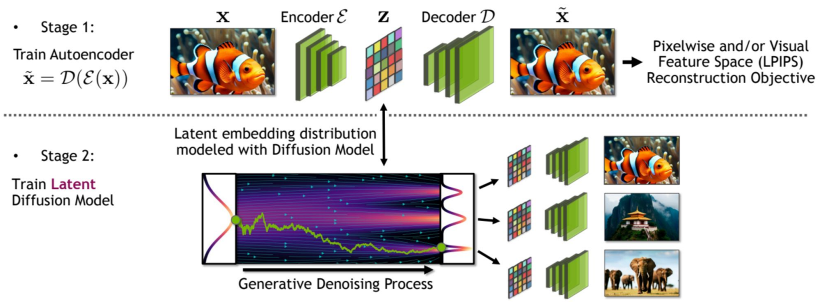
Cornell Bowers C

Sprouts in the shape of text 'Imagen' coming out of a fairytale book. | A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat. | A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.

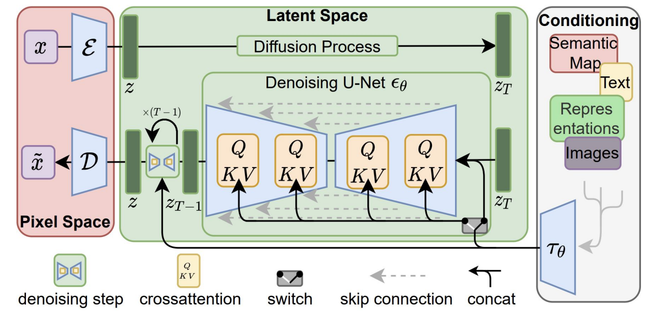
Teddy bears swimming at the Olympics 400m Butter-fly event. | A cute corgi lives in a house made out of sushi. | A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., ... & Norouzi, M. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, *35*, 36479-36494.

Latent Diffusion



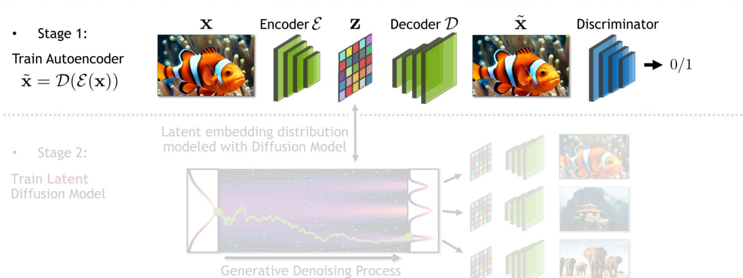
High-Resolution Image Synthesis with Latent Diffusion Models



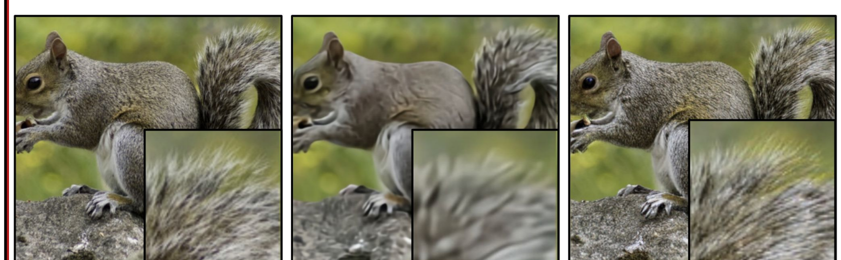
Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models", CVPR, 2022

Latent Diffusion

Add Adversarial Patch-based Discriminator on top of Reconstruction Loss for Perceptual Compression



Impact of Patch Discriminator

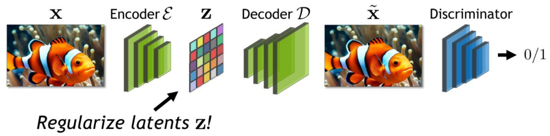


Input | Reconstruction without Discriminator | Reconstruction with Discriminator

Cornell Bowers C·IS

Latent Diffusion

Regularize Latent Space for better Compression and easier Training of Latent Space Diffusion Models



Parametrize encoder by diagonal Gaussian, regularize towards standard normal distribution, as in regular VAEs.

Use very small weight for KL regularization term (weak regularization).

Encoder distribution:

$$q_{\mathcal{E}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathcal{E}_\mu, \mathcal{E}_\sigma^2)$$

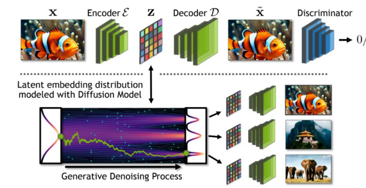
KL regularization in latent space:

$$\mathrm{KL}\left(q_{\mathcal{E}}(\mathbf{z}|\mathbf{x})||\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})\right)$$

Cornell Bowers C·IS

Latent Diffusion

Latent Diffusion Models offer Excellent Trade-off between Performance and Compute Demands



➡ LDM with appropriate regularization, compression, downsampling ratio and strong autoencoder reconstruction:

- **Computationally efficient** diffusion model in latent space (compression & lower resolution).
- Yet **very high-performance** (latent diffusion + autoencoder + discriminator = ❤).
- **Highly flexible** (can adjust autoencoder for different tasks and data).

Cornell Bowers C·IS

Latent Diffusion

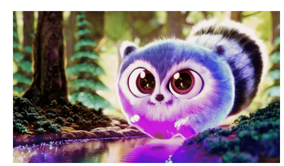
Many state-of-the-art large-scale text-to-image models are latent diffusion models

- Stability AI's Stable Diffusion
- Meta's Emu
- OpenAI's Dall-E 3

Cornell Bowers C·IS

Latent Diffusion for Video Generation

Sora



Prompt: 3D animation of a small, round, fluffy creature with big, expressive eyes explores a vibrant, enchanted forest.

Cornell Bowers C·IS

Latent Diffusion for Video Generation

Sora

Prompt: 3D animation of a small, round, fluffy creature with big, expressive eyes explores a vibrant, enchanted forest.

Prompt: A cat waking up its sleeping owner demanding breakfast.

Cornell Bowers C·IS

Review

- Diffusion models can be used to generate high quality samples
- They were introduced simultaneously from two different perspecteives
- Conditional diffusion models can be used to generate samples conditioned on other text or image
- Diffusion can also be performed in latent spaces