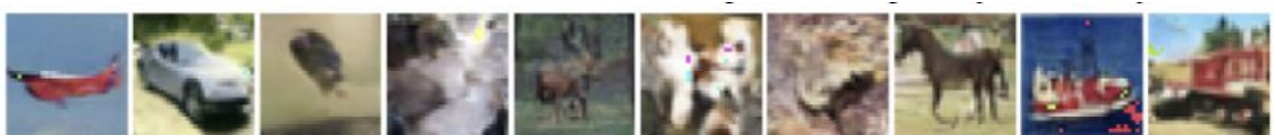
Training Image Classification Models with Synthetic Data:

A Study using Denoising Diffusion Probabilistic Models and ResNet



IT UNIVERSITY OF CPH

Johan Ødum Lundberg, jolu@itu.dk

Outline

Introduction

Motivation

Denoising Diffusion Probabilistic Model

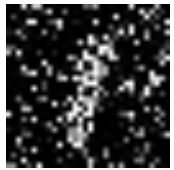
ResNet Model for Image Classification

Results

Key Findings & Future Work

Introduction: Overview of Topic

- Train a model to sample from a distribution P(x) ~
- Model ⟶ Neural Network
- Generate new data



Samples from a Data Distribution

- Improve a classifier with more data
- Data Sets
  - MNIST & CIFAR-10

Real image

Sampled Image

Introduction: Background of Denoising Diffusion Models

2015:

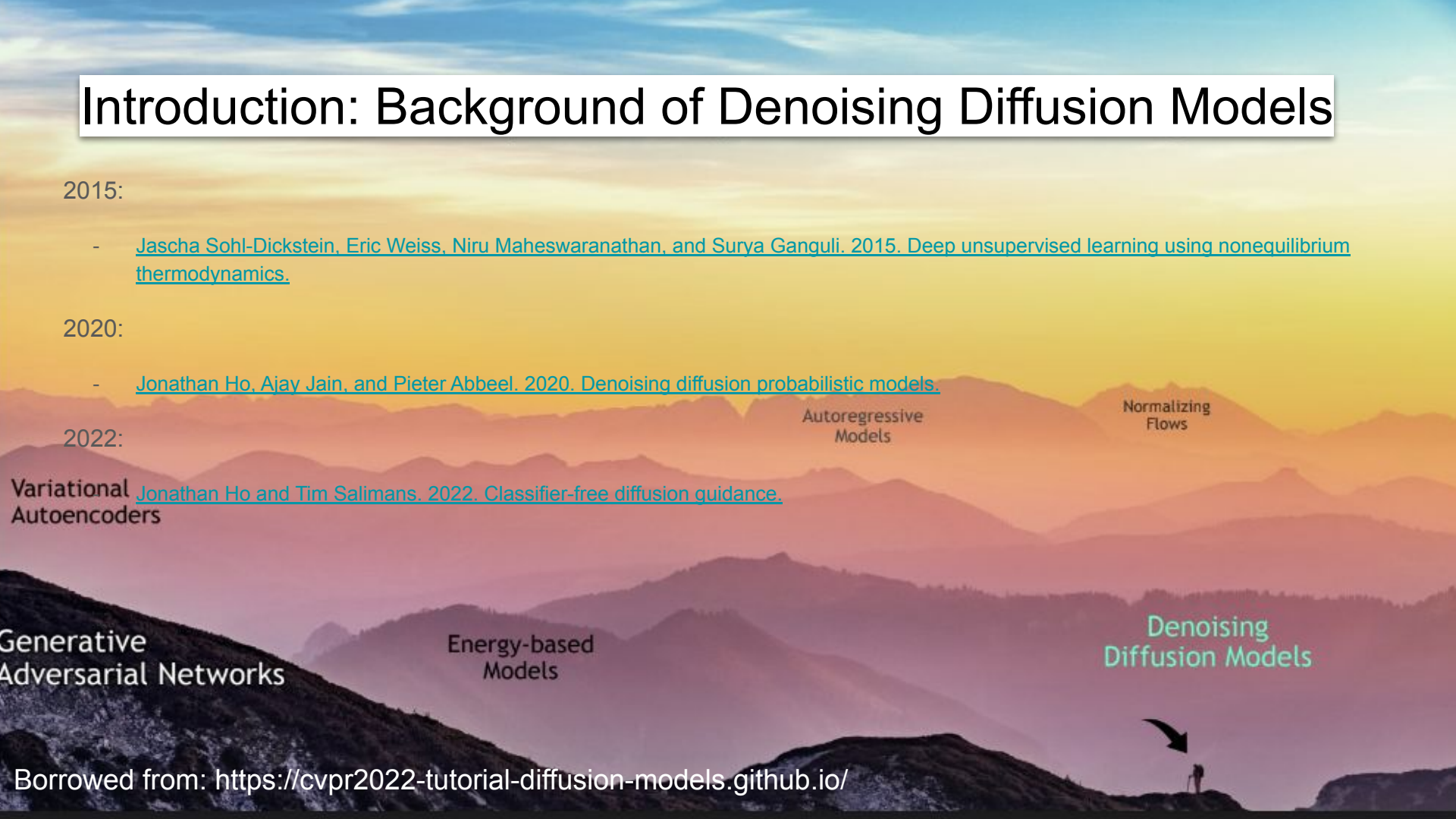
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics.

2020:

- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models.

2022:

- Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance.



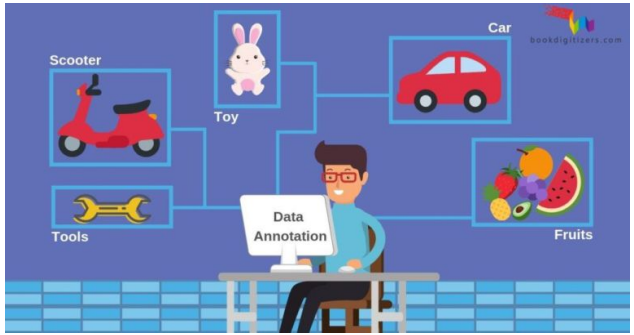
Borrowed from: https://cvpr2022-tutorial-diffusion-models.github.io/

Introduction: Research Question

Can we use the learned representation from a diffusion model to improve a downstream task by generating more data?

Motivation for using Synthetic Data (1): Real Data is Hard to Obtain

- Scraping
- Annotation: Mechanical Turk
  - Expensive and error prone
- Private data / restricted data



Motivation for Using Synthetic Data (2): Pros of Using Synthetic Data

- Generate large amounts of data quickly.

E.g. Stable Diffusion with 6.80 samples/sec with 100 inference steps on the LSUN Churches data set.

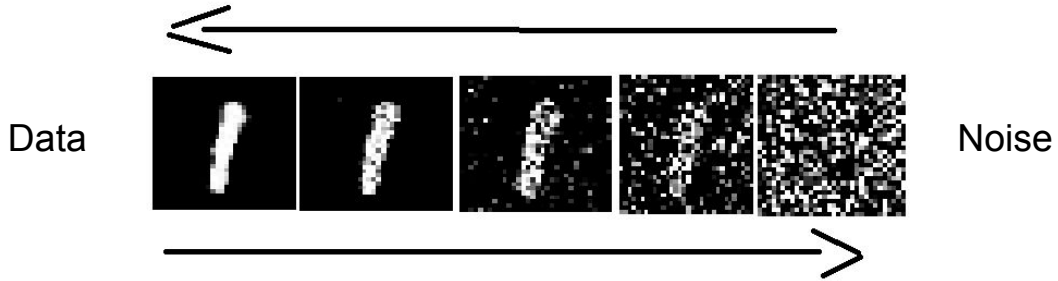
- Control properties of the data (text-to-image diffusion).

Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

Denoising Diffusion Probabilistic Model(DDPM)

Consists of two processes:

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



DDPM: Forward Process

- Down-scales the mean of the data input and adds noise according to a noise schedule beta.
- Produces a sequence of latent variables represented by the joint of conditionals.
- The beta schedule controls how fast noise is added to the image and ensures that x at time step T resembles a gaussian distribution.

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \quad (2)$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t\geq 1} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad (1)$$

$$\beta_t \in [0,1] \qquad \beta_1 < \beta_2 < \cdots < \beta_{T-1} < \beta_T$$

DDPM: Reverse Process

- A parameterized network predicts the mean of x at time step and adds noise back into the image.
- Produces the joint probability of a sequence of conditionals.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)). \quad (5)$$

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t \geq 1} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad (4)$$

DDPM

How the noise-prediction network learns:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\hat{\alpha}_t}\sqrt{\alpha_t}}\epsilon_\theta(\mathbf{x}_t, t)). \quad (6)$$

$$L = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right] \quad (7)$$

DDPM
Classifier-Free Guidance

- Training with guidance
  - 90% of the time, 10% without
  - An embedding is created of the label and added to the time embedding
- Sampling is done with and without labels and the predictions are interpolated:

$$\hat{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}) = (1 + w)\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) - w\epsilon_\theta(\mathbf{x}_t)$$

DDPM
Training & Sampling

Table 1: DDPM training algorithm.

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(1, \dots, T)$
4: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient step on
$$\nabla \|\epsilon - \epsilon_\theta((\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2$$
6: **until converged**

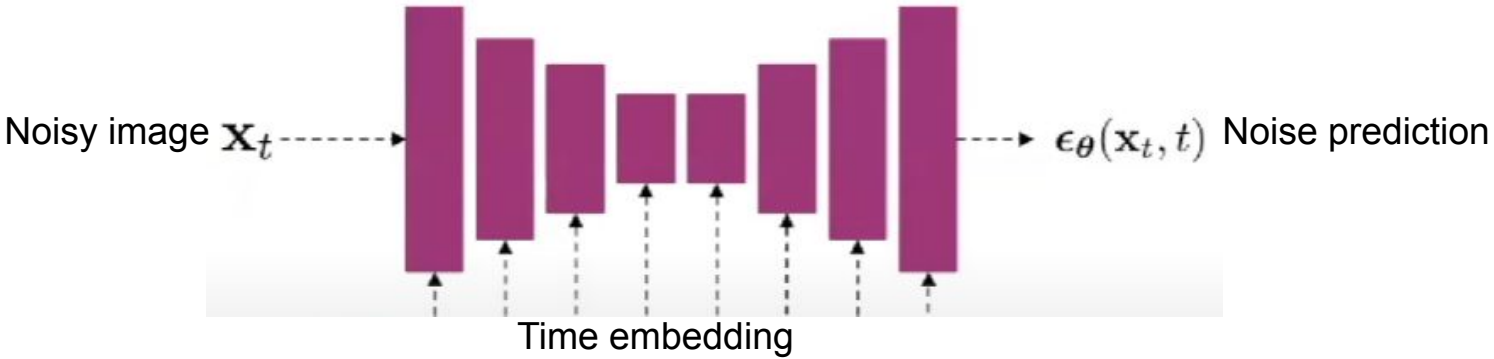
Table 2: DDPM sampling algorithm.

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if t > 1, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x_{t-1}} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x_t}, t)) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

DDPM: Noise Prediction Network

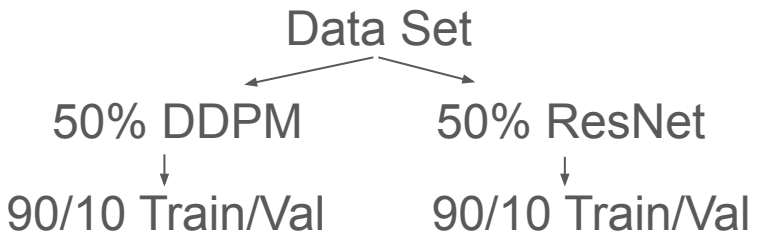
- U-Net (64, 128, 256, 512)
- ResNet blocks + Attention
- Time embedding = time mlp + label embedding
- Time mlp = Sequential(SinusoidalPosEmb, Linear, GELU, Linear)



ResNet Model for Image Classification

- U-Net type architecture with ResNet blocks + attention layers
- Final layers
  - Global average pooling across height & width
  - Linear layer
  - SoftMax activation
- Cross-entropy loss
- Parameters
  - Learning rate of 0.0005
  - Batch size of 64
- Adam optimizer
- Evaluation: Global average F1 score

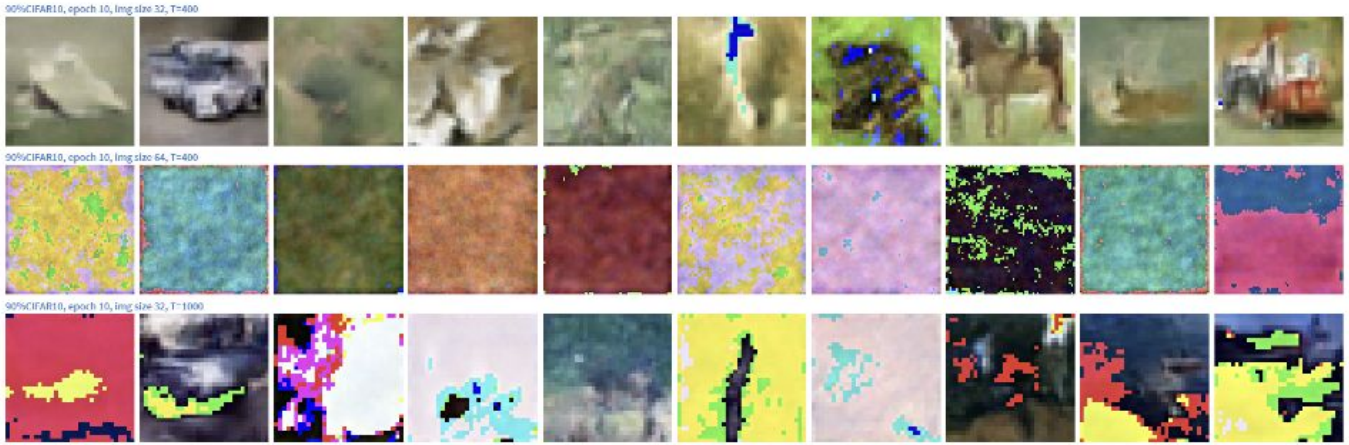
Training the ResNet Model on Synthetic Data



The ResNet is trained in five different ways. Using 100% of the 90%, some percentage of the 90% mixed with synthetic data or only synthetic data.

Results of Exploring DDPM Parameters

- Parameters of the model
  - T = 400
  - Cfg = 3
  - $\Sigma_\theta = \beta$
  - Img size: 32



Results of Using Synthetic Data for Training

Table 6: CIFAR-10 and MNIST Experiments.

Experiment	% real images	% synthetic images	CIFAR10 test set f1 score	MNIST test set f1 score
Exp1	100%	0 %	0.69	0.97
Exp2	0 % %	100 %	0.36	0.71
Exp3	50 %	50 %	0.50	0.98
Exp4	10 %	90 %	0.44	0.95
Exp5	90 %	10 %	0.36	0.99

Influence of Image Complexity and Data Availability

Figure 3: MNIST digits generated from DPM during training at epoch 10 using img size 32 and T=400.

(a) DPPM trained on 90% of the 50% of CIFAR-10 training data set. Images sampled after 38 epochs.

(b) DDPM trained on 90% of CIFAR-10 training data set. Images sampled after 62 epochs.

Figure 5: DPPM image generations with image size 32 and T=400. Images were sampled after model convergence. Classe are (left to right): Plane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

Key Findings and Future Work

Key findings

The effect of synthetic data **diminishes** as the complexity of the images increases, **requiring more data** to achieve good performance.

Achieving results comparable to using real data requires a larger amount of synthetic data or the mixture of real and synthetic data.

A classifier can be trained using synthetic data and **achieve good performance** depending on the amount of available data and its complexity.

Future work

Studying the impact of **varying the classifier-free guidance** scale parameter could be interesting as it has been shown to affect the quality of the final images.

In a real-world scenario, it would be more practical to train the DDPM on the full data set and the classifier on the full data set as well as the synthetic data set.