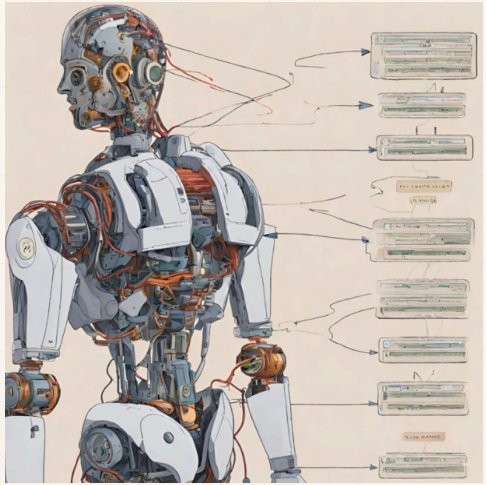
A TALE OF



VAEs, (VARIATIONAL AUTOENCODERS)
WGANs, (WASSERSTEIN GANs)
DDPMs (DENOISING DIFFUSION PROBABILISTIC MODELS)

ASHWIN DE SILVA

Generative Modeling

Suppose that observed datum $x \sim p^*(x)$ "true distribution"

We wish to approximate $p^*(x)$ with a chosen model

$P_\theta(x)$ with parameters $\theta \in (H)$

Generative Modeling $\equiv$ Process of searching for $\theta^* \in (H)$ such that

$$P_{\theta^*}(x) \approx p^*(x)$$

Generative Modeling

Dataset of observations $\mathcal{D} = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\} \sim p^*(x)$ (iid)

Under the model $p_\theta(x)$,

$$\log p_\theta(\mathcal{D}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)}) \equiv \text{log-likelihood of the data}$$

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

- Maximum Likelihood Estimate (MLE)
- same as minimizing KL divergence between $p^*(x)$ and $p_\theta(x)$
- can be solved by minibatch SGD

* In Bayesian inference, one may use MAP.

Latent Variable Models

Latent variables are the model variables that we don't observe.

$p_\theta(x, z)$ ⟶ $p_\theta(x|z)\, p_\theta(z)$ is a common factorization

- $p_\theta(x, z)$: observed variables x, latent variables z
- prior $p_\theta(z)$
- conditional $p_\theta(x|z)$
- posterior $p_\theta(z|x)$

$$p_\theta(x) = \int p_\theta(x, z)\, dz$$

$p_\theta(x)$: marginal likelihood

↳ can lead to highly expressive models

eg:- If z is discrete & $p_\theta(x|z)$ is Gaussian, then $p_\theta(x)$ is a GMM.

Latent Variable Models

Consider a LVM $p_{\theta^*}(x, z)$ with unknown θ^*

We have $x^{(1)}, \ldots, x^{(N)} \sim p_{\theta^*}(x)$

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_\theta(x^{(i)})$$

* Cannot use SGD directly since taking $\nabla_\theta \log p_\theta(x)$ requires integrating over z $\left(\because p_\theta(x) = \int p_\theta(x, z)\, dz\right)$

We could use Bayes Theorem to obtain $p_\theta(x) = \dfrac{p_\theta(x, z)}{p_\theta(z|x)}$

But $p_\theta(z|x)$ is intractable to compute

Solution: approximate $p_\theta(z|x)$ by "$q_\phi(z|x)$"

Evidence Lower Bound (ELBO)

$$\log p_\theta(x) \geq \mathbb{E}_{z \sim q(\cdot|x)} \left[\log \frac{p_\theta(x, z)}{q_\phi(z|x)} \right]$$

Instead of maximizing $\log p_\theta(x)$, maximize the ELBO:

$$\max_{\theta, \phi} \sum_{i=1}^{N} \mathbb{E}_{z \sim q(\cdot|x^{(i)})} \left[\log \frac{p_\theta(x^{(i)}, z)}{q_\phi(z|x^{(i)})} \right]$$

$$\equiv \min_{\theta, \phi} \sum_{i=1}^{N} \left\{ KL(q_\phi(z|x^{(i)}) \| p_\theta(z)) + \mathbb{E}_{z \sim q_\phi(\cdot|x^i)} \left[\log p_\theta(x^i|z) \right] \right\}$$

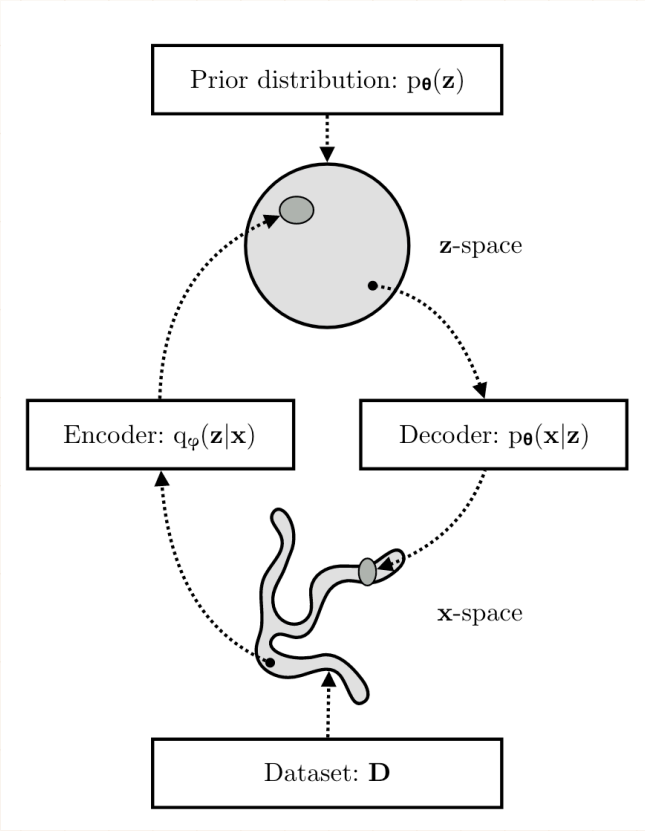
Variational Autoencoders (For Bernoulli data)

$$P(z) = \mathcal{N}(z; 0, I)$$

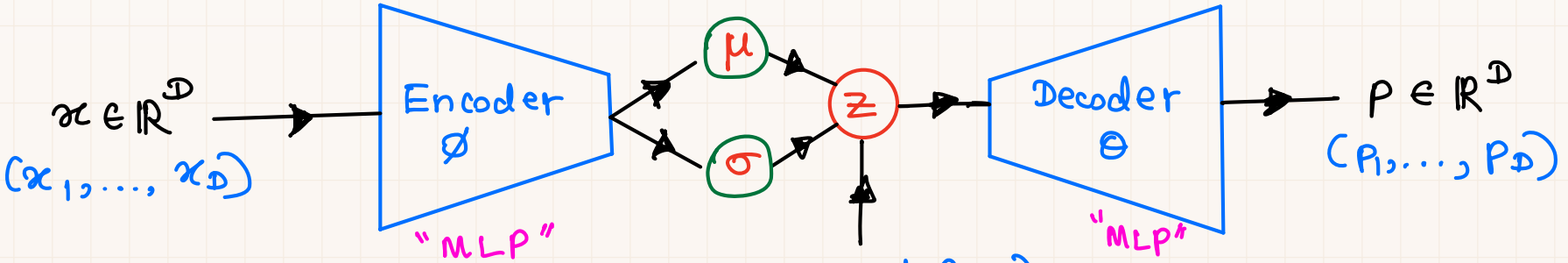
$$(p_1, \ldots, p_D) = \text{Decoder } NN_\theta(z)$$

$$(\mu, \log \sigma) = \text{Encoder}_\phi(x)$$

$$q_\phi(z|x) = \mathcal{N}(z; \mu, \text{diag}(\sigma))$$



Variational Autoencoders (For Bernoulli data)



reparameterization trick

$$z = \mu + \sigma \odot \varepsilon = g(\varepsilon, \phi, x)$$

Loss function

$$\min_{\theta, \phi} \sum_{i=1}^{N} \left\{ KL\left(q_\phi(z|x^{(i)}) \,\|\, p_\theta(z)\right) - \mathbb{E}_{z \sim q_\phi(\cdot|x^i)}\left[\log p_\theta(x^i|z)\right] \right\}$$

KL between 2 gaussians

Binary Cross Entropy loss

Wasserstein GAN

Recall that $\max_\theta \sum_{i=1}^{N} \log P_\theta(x^{(i)}) \equiv \min_\theta KL(P_\theta \| P^*)$

There're many measures of "distance" between distributions.

- KL-Divergence, J-S Divergence, ... (f-Divergences)
- Total Variation Distance
- Hellinger Distance.
- Wasserstein Distance

$$W(P^*, P_\theta) = \inf_{\gamma \in \Pi(P^*, P_\theta)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x-y\|^2\right]$$

$\Pi(P^*, P_\theta)$: set of all joint distributions whose marginals are P^* & P_θ

$\rightsquigarrow$ cost of the optimal transport plan

Wasserstein GAN

Unlike other distances, $W(P^*, P_\theta)$

- is Continuous everywhere
- is Differentiable almost everywhere
- induces a weak topology in the space of distributions

} Desirable properties for a gradient based minimization.

$$W(P^*, P_\theta) = \inf_{\gamma \in \Pi(P^*, P_\theta)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x-y\|^2\right]$$

↕

↳ hard to evaluate

$$W(P^*, P_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim P^*}[f(x)] - \mathbb{E}_{x\sim P^*}[f(x)]$$

↳ supremum over 1-Lipschitz functions

"Kantorovich-Rubinstein Duality"

Wasserstein GAN

- Suppose $z \sim \mathcal{N}(0, I)$. Then let g_θ be a generator

$$x = g_\theta(z) \sim P_\theta(x)$$

- Suppose $\{f_\phi\}$ be a family of functions parameterized by ϕ that are 1-Lipschitz

$$W(p^*, p_\theta) = \max_\phi \mathbb{E}_{x \sim p^*}[f_\phi(x)] - \mathbb{E}_z[f_\phi(g_\theta(z))]$$

Now we can minimize $W(p^*, p_\theta)$ w.r.t θ

$$\min_\theta W(p^*, p_\theta) = \min_\theta \max_\phi \mathbb{E}_{x \sim p^*}[f_\phi(x)] - \mathbb{E}_z[f_\phi(g_\theta(z))]$$

Wasserstein GAN

$$\min_{\theta} \max_{\phi} \; \mathbb{E}_{x \sim p^*}[f_\phi(x)] - \mathbb{E}_z[f_\phi(g_\theta(z))]$$

"WGAN objective"

Algorithm 1 Training procedure of the Wasserstein GAN

Require: the learning rate α, the clipping parameter c, batch size m, number of iterations n_{critic} of the critic per generator iteration

Require: The initial critic parameters w_0, the initial generator parameters θ_0

while θ has not converged **do**

  for $t = 0, \ldots, n_{\text{critic}}$ **do**

    Sample m real data $x^{(1)}, \ldots, x^{(m)} \sim \mathbb{P}_r$

    Sample m latent realizations $z^{(1)}, \ldots, z^{(m)} \sim p(z)$

    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$

    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$

    $w \leftarrow \text{clip}(w, -c, c)$

  Sample m latent realizations $z^{(1)}, \ldots, z^{(m)} \sim p(z)$

  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$

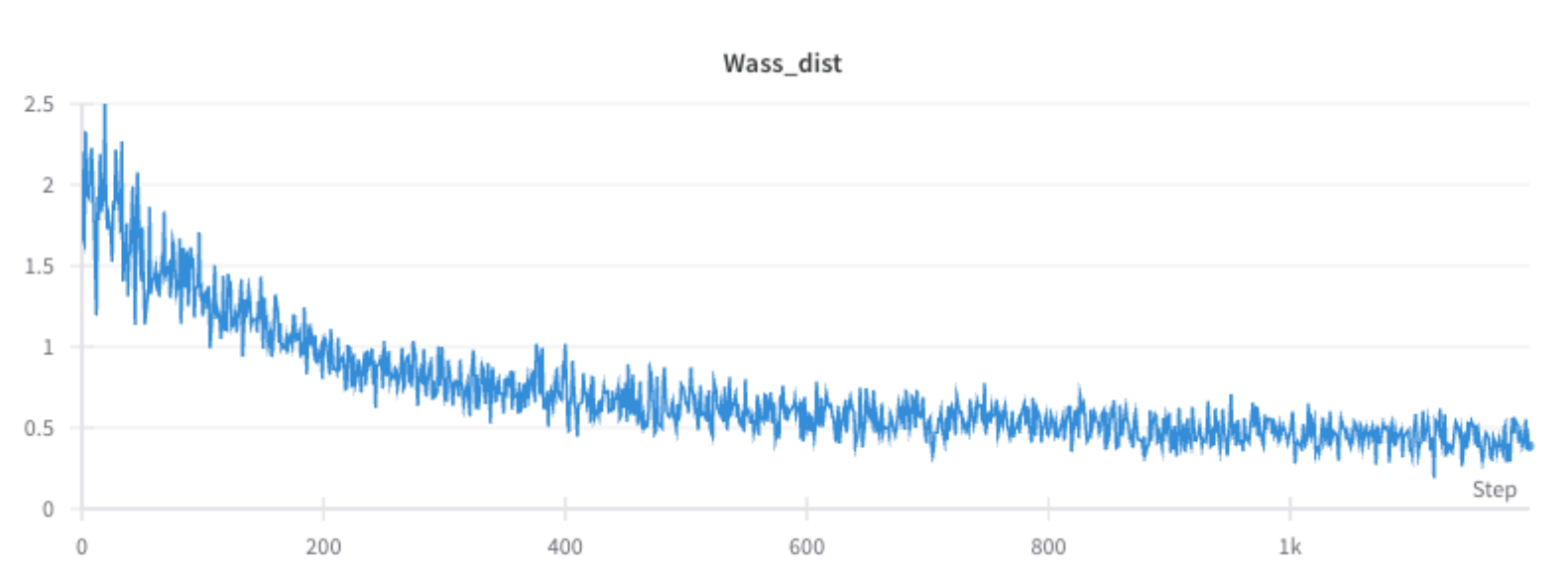
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$

"critic update" & "discriminator"

"generator update"

Wasserstein GAN on MNIST

$W(p^*, p_\theta)$ estimate



iterations

Denoising Diffusion Probabilistic Models

- Also a latent variable model $p_\theta(x_0, x_1, \dots, x_T)$
  - x_0: observed
  - $x_1, \dots, x_T$: latent variables
- $x_0 \sim q(x_0)$ true distribution
- let the joint $p_\theta(x_0, \dots, x_T)$ be a markov-chain (Reverse Process)

$$p_\theta(x_0, \dots, x_T) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} | x_t) \rightarrow \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

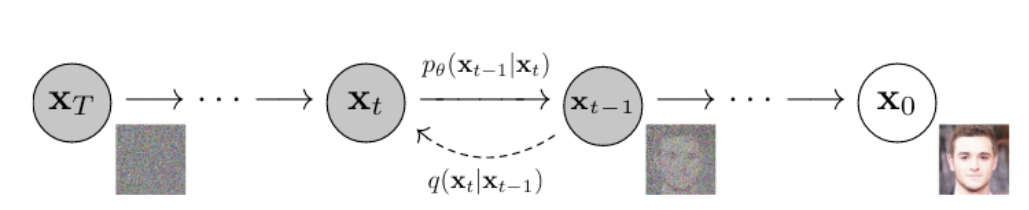
- Let the approximate posterior $q(x_1, \dots, x_T | x_0)$ also be a markov chain (forward process)

$$q(x_1, \dots, x_T | x_0) = \prod_{t=1}^{T} q(x_t | x_{t-1})$$

$$q(x_t | x_{t-1}) \rightarrow \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_t, \beta_t I)$$

$\beta_t \sim$ Noise Schedule

Denoising Diffusion Probabilistic Models



ELBO $\Rightarrow$

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(x_0, x_1, \ldots, x_T)}{q(x_1, \ldots x_T | x_0)}\right] := L$$

"simplify"

$$h(\theta) = \mathbb{E}_{t, x_0, \varepsilon}\left[\| \varepsilon - \varepsilon_\theta(x_t, t) \|^2\right]$$

Denoising Diffusion Probabilistic Models

$$L(\theta) = \mathbb{E}_{t, x_0, \varepsilon}\left[\| \varepsilon - \varepsilon_\theta(x_t, t) \|^2 \right]$$

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon$$

1. Draw an image x_0
2. Draw a noise $\varepsilon \sim \mathcal{N}(0, I)$
3. Create a noisy image $x_t = x_0 + \varepsilon$
4. Force network ε_θ to denoise x_t i.e. force ε_θ to predict ε.

Algorithm 1 Training

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

But why?

Langevin Monte-Carlo, Score Matching & Denoising

(A high-level analogy)

Score function of probability density $p(x)$ is $s(x) = \nabla \log p(x)$

Langevin MC $\Rightarrow$ $x_{t+1} = x_t - \eta \nabla \log p(x_t) + \sqrt{2\eta}\,\varepsilon_t$; $\varepsilon_t \sim \mathcal{N}(0, I)$

Stationary Distribution of this SDE is $p(x)$

How to estimate $\nabla \log p(x)$?

$$L(\theta) = \mathbb{E}_{x \sim p}\left[\| \nabla \log p(x) - \varepsilon_\theta(x) \|^2\right]$$

$\Big\}$ approximately equivalent to

"Denoising Loss" $\rightsquigarrow$ $\mathbb{E}_{x \sim p,\, \varepsilon \sim \mathcal{N}(0,I)}\left[\| \varepsilon - \varepsilon_\theta(x + \varepsilon) \|^2\right]$

$\therefore$ By forcing a denoising objective, we let $\varepsilon_\theta(\cdot)$ approximate the score function $\nabla \log p(x)$

Sampling

Algorithm 2 Sampling

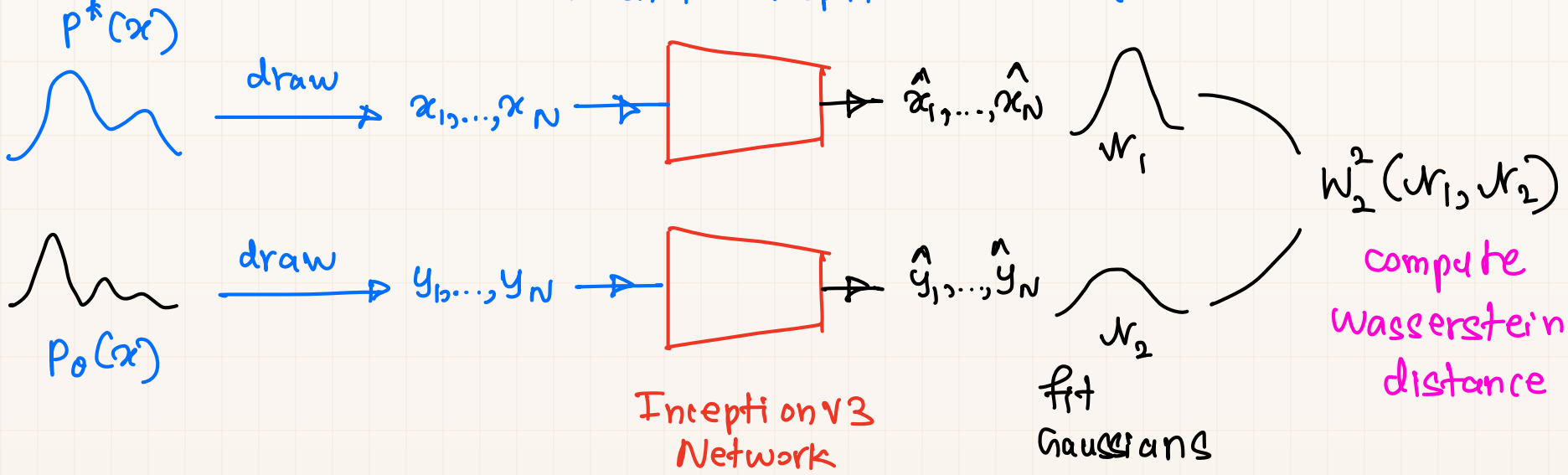
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}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

Notice the similarity to the Langevin Monte-Carlo

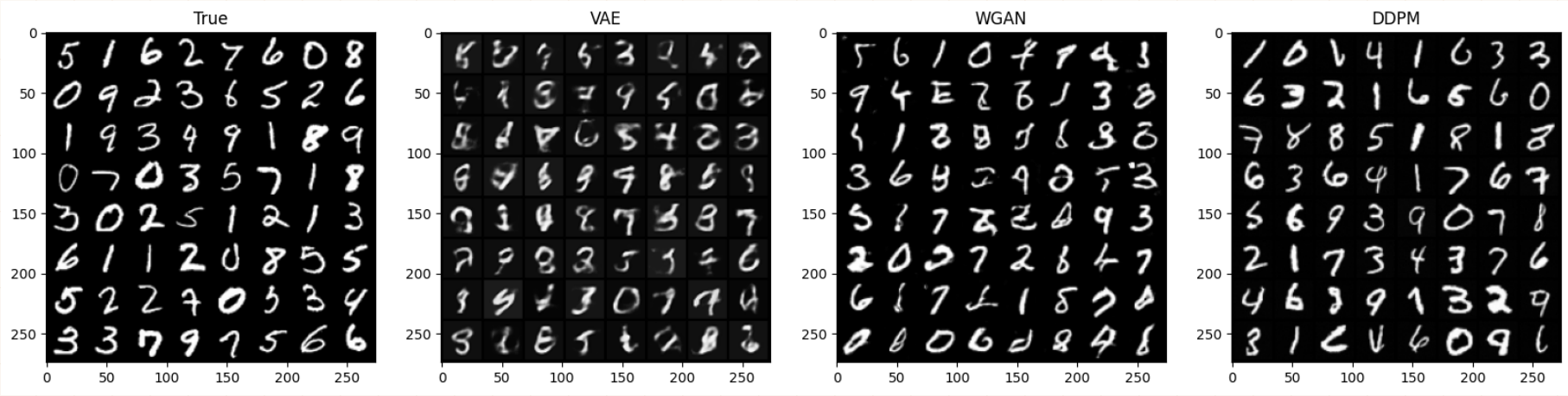
- Currently the state-of-the-art in generating images.
- Sampling is very slow
- Active Research Area.

Evaluating Generative Models (FID Score)

Frechet Inception Distance



Some Results

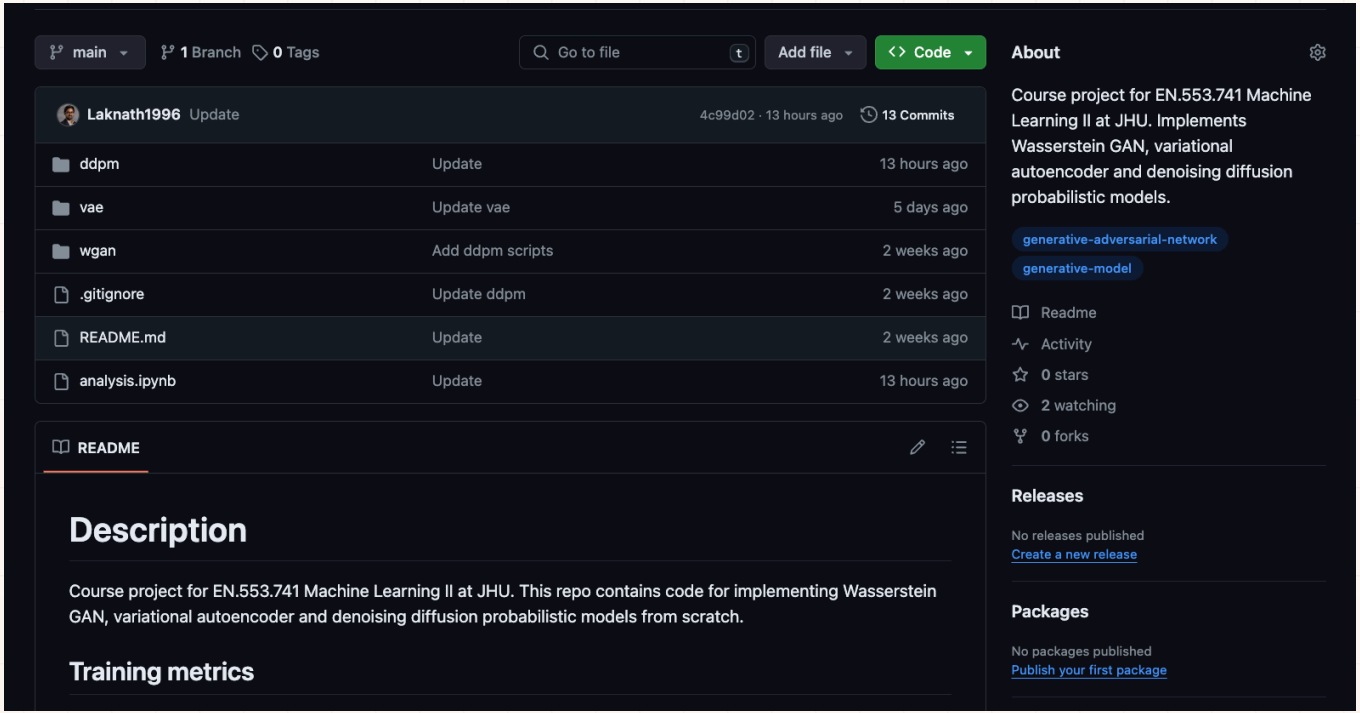


FID
16.3

FID
12.4

FID
13.4

Code



link to repo

THANK YOU!