Generative LA:

↳ Generate new Objects

↳ Images, Videos, proteins

→ Flow / Diffusion models → Theory: Ordinary/stochastic Diff. Equats.

Sampling:

From Generation → Sampling.

How we represent Images/Videos As Vectors?

• Images:

- Heing/width
- 3 Color Channels RGB

$z \in \mathbb{R}^{H \times W \times 3}$ — dim

vector z — vector space

• Videos (string Image)

→ Each frame is Image

$z \in \mathbb{R}^{T \times H \times W \times 3}$

$z \in \mathbb{R}^d$ → Generate Objects

Generation as Sampling from Data Dist.

Data Distribution: Dist. of Objects we want to Generate p_{data}

↳ How prob is each possible Data in a Group of Data.

Probability Density:

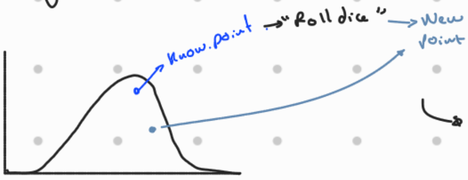
$$\boxed{\begin{aligned} p_{data} &: \mathbb{R}^d \to \mathbb{R}_{\geq 0}, \\ z &\longmapsto p_{data}(z) \end{aligned}}$$

"Goes from Vector Space R to D (space Objects) and gives you a Non-Neg. Number".

"Given an Object z, it gives u a prob of How likely that Object is"

what Does it mean to generate?

↳ ! WE DON'T KNOW PROB DENSITY → we approx as a Dist. from Samples.



Generation means Sampling Dist. Data.

↳ $z \sim p_{data} \implies z = \text{Image}$

A Dataset Consists of Samples from Data Dist.

- we Need Dataset

A Dataset consists of finite number of samples from the Data Distribution.

$$\boxed{z_1, \dots, z_n \sim p_{data}}$$

Conditional Generative Allows us to condition on Prompt.

- Data Dist (p_{data})

  Fixed prompt

  "Dog"

  | Dog 1 | Dog 2 | Dog 3 |

- Conditional Data Dist $p_{data}(\cdot|y)$

  ↳ condition Variable: y

  y = "Dog" y = "cat" y = "Landscape"

  [Dog] [cat] [Landscape]

→ Important

→ Conditional Data Distribution

$$z \sim p_{data}(\cdot|y)$$

Generative models generate Samples from Data Dist.

Initial Dist: p_{init} → Normally $p_{init} = N(0, I_d)$

↳ Gaussian Dist.

↳ 0 mean, Identity matrix covariance matrix

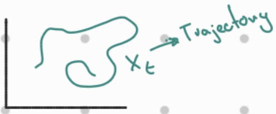
General Idea: $x \sim p_{init} \Rightarrow$ Generative model $\Rightarrow z \sim p_{data}$

Flow / Diffussion models.

- For flows:

Trajectory: Function Time

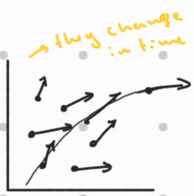
↳ $\underbrace{X : [0,1] \to \mathbb{R}^d}_{\text{Function time}}$, $\underbrace{t}_{\text{Given time } t} \mapsto \overset{\text{gives } X_t \text{ point}}{X_t}$



Vector field: Spatial component and time component

$\underbrace{u}_{\text{vector field}} \; \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$

$\underbrace{(x, t)}_{\text{vector } x \text{ at time } t} \to u_t(x)$ ↳ vector $u_t(x)$



Ordinary Diff. Eq: (ODE) → Describes cond. on trajectory

$X_0 = x_0$ (initial condition)

$$\underbrace{\frac{d}{dt} X_t}_{\text{vel trag}} = \underbrace{u_t(X_t)}_{\text{location } x \text{ in vector space}} \quad \text{(ODE)}$$

Flow: Collection Trajectories that follow the ODE

↳ Diff. Solutions with diff. Initial conditions, and gather them in a function!

↳ $\varphi : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$

↳ $(x_0, t) \to \varphi_t(x_0)$ → For every initial cond. is a sol for ODE

↳ $\varphi_0(x_0) = x_0$ (we start at x_0)

↳ $\frac{d}{dt}\varphi_t(x_0) = u_t(\varphi_t(x_0))$

Flow

$\boxed{\text{Trajectory}} \xrightarrow{\text{sol}} \boxed{\text{ODE}} \xleftarrow{\text{def}} \boxed{\text{VF}}$

***Rule:** If vector field $u_t(x)$ is continuously Diff. ~~with derivatives~~, then a Unique sol to ODE exist.

↳ FLOW MAP EXIST

Ex.

Simple vector field:

$u_t(x) = -\theta x \quad (\theta > 0)$

Claim: Flow given by:

$\varphi_t(x_0) = \exp(-\theta t)\, x_0$

Proof:

1: Initial Cond

$\varphi_t(x_0) = \exp(0) x_0 = x_0$

2: ODE:

chain Rule.

$\frac{d}{dt}\varphi_t(x_0) = \frac{d}{dt}(\exp(-\theta t)\, x_0) = -\theta \exp(-\theta t)\, x_0 = -\theta \varphi_t(x_0)$

↳ $u_t(\varphi_t(x_0))$

* Hard to compute it:

↳ Euler method!

↳ Going Direction vector field in small timestep each time.

Algorithm 1 Sampling from a Flow Model with Euler method

Require: Neural network vector field u_t^θ, number of steps n

Set $t = 0$

Set step size $h = \frac{1}{n}$

Draw a sample $X_0 \sim p_{\text{init}}$

for $i = 1, \ldots, n$ **do** → Until $t = 1$

  $X_{t+h} = X_t + h u_t^\theta(X_t)$ → Small step into Direction vector field

  Update $t \leftarrow t + h$

end for

return X_1 → Trajectory

Flow model: $p_{init} \xrightarrow{\text{ODE}} p_{data}$

VECTOR FIELD

Gaussian Dist

FLOW IS AN ALGORITHM NOT A NEURAL N

Look for VECTOR FIELD

Neural Network: $u_t^\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$; θ : Parameters

Numbers that changes of model

Random Init: $X_0 \sim p_{init}$

Initial Cond

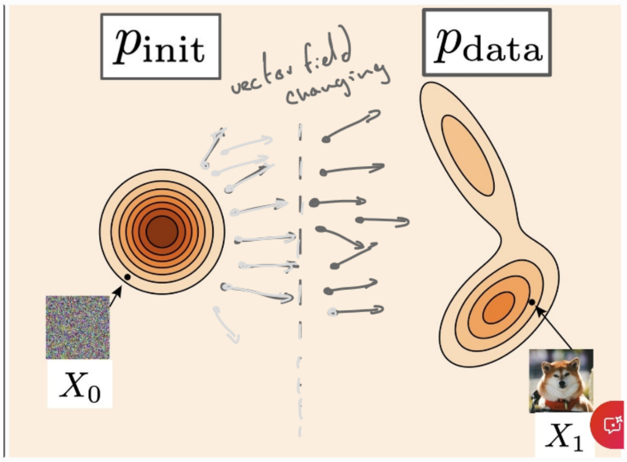
sample

" X_0 is A sample of p_{init} "

ODE: $\frac{d}{dt} X_t = u_t^\theta (X_t)$

Goal:

$X_1 \sim p_{data}$



Algorithm 1 Sampling from a Flow Model with Euler method

Require: Neural network vector field u_t^θ, number of steps n

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Draw a sample $X_0 \sim p_{\text{init}}$ *Random initialization!*
4: **for** $i = 1, \dots, n-1$ **do**
5: $X_{t+h} = X_t + h u_t^\theta (X_t)$
6: Update $t \leftarrow t + h$
7: **end for**
8: **return** X_1 *Return final point*

DIFFUSION MODELS.

* STOCHASTIC DIFF EQ:

↳ Solutions are Random Trajectories

↳ Stochastic Process

Stochastic Process:

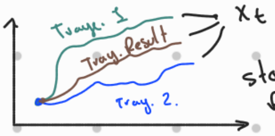
Randomness

X_t random variable $(0 \le t \le 1)$

$X : [0,1] \xrightarrow{\text{goes}} \mathbb{R}^d, \ t \overset{\text{to}}{\longmapsto} X_t$

Trajectory Random!

time: Function time; $\mathbb{R}^d$; t: time Step; X_t: vector X_t



stochastic process

Vector field: Same for ODE

$u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ + Diffusion Coefficient

Include Randomness (noise)

$\sigma [0,1] \to \mathbb{R}$

$t \longrightarrow \sigma_t$

Stochastic Differential Equation (SDE)

Process

$X_0 = x_0$ (Init conditions)