1.From DDPM To Flow

1.1 DDPM is a SDE

in DDPM forward pass, we add noise in x_{t-1} and get x_t . This is a discrete Markov chain.

$$q(x_i|x_{i-1}) = \mathcal{N}(x_i; \sqrt{1-\beta_i}x_{i-1}, \beta_i\mathbf{I})$$

where β_i is a small number.
what if we use a continuous variable $t \in [0, T]$?
Similar to $\beta_1, \beta_2, \cdots, \beta_t$, we can define a continuous $\beta(t)$ in continuous version DDPM, $\beta_i = \beta(t)\Delta t$, when $\Delta t \to 0$, we get a continuous time **SDE**(Stochastic Differential Equation) below is the proof.
in DDPM, we have $x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}noise_t \quad noise_t \sim \mathcal{N}(0, I)$, so in continuous version, we also have

$$x_t = \sqrt{1-\beta(t)\Delta t}\cdot x_{t-\Delta t} + \sqrt{\beta(t)\Delta t}\cdot z_{t-\Delta t} \qquad z \sim \mathcal{N}(0, \mathbf{I})$$

when Δt is samll, according to Taylor expansion , $\sqrt{1-x} \approx 1 - \frac{x}{2}$ we have:

$$\sqrt{1-\beta(t)\Delta t} \approx 1 - \frac{1}{2}\beta(t)\Delta t$$

Substitute into the above equation ,

$$x_t \approx (1 - \frac{1}{2}\beta(t)\Delta t)x_{t-\Delta t} + \sqrt{\beta(t)}\sqrt{\Delta t}z_{t-\Delta t}$$

we use $dx_t = x_t - x_{t-\Delta t}$ and we finally get :

$$dx_t \approx -\frac{1}{2}\beta(t)x_{t-\Delta t}dt + \sqrt{\beta(t)}dW_t \qquad dt = \Delta t \quad dW_t = \sqrt{\Delta t}z_{t-\Delta t}$$

this is the DDPM forward pass' **SDE**:

$$dx_t = f(x_t, t)dt + g(t)dW_t = -\frac{1}{2}\beta(t)x_t dt + \sqrt{\beta(t)}dW_t$$

this equation precisely describe how the Gaussian distribution $p_T(x)$ truns to the data distribution $p_0(x)$ with time t smoothly increasing.

But what we want is how to turn the Gaussian distribution to the data distribution, which need a kind of "reverse SDE" . Fortunately, according to the SDE theory, we have the "reverse SDE" :

$$dx_t = [f(x_t, t) - g(t)^2\nabla_{x_t}\log p_t(x_t)]dt + g(t)d\bar{W}_t$$

where $p_t(x_t)$ is the distribution of x_t at times , and $\nabla_{x_t} \log p_t(x_t)$ is the Score Function
substitute f and g above, we get :

$$dx_t = [-\frac{1}{2}\beta(t)x_t - \beta(t)\nabla_{x_t} \log p_t(x_t)]dt + \sqrt{\beta(t)}d\bar{W}_t$$

this equation shows that DDPM is implict learning the $\nabla_{x_t} \log p_t(x_t)$, which we already discuss in Diffusion Policy

1.2 DDPM can be ODE as well

Unfortunately, the reverse SDE also is a Stochastic process . However, for the same stochastic differential equation (SDE), there exists a corresponding **deterministic** ordinary differential equation (ODE) whose trajectories generate the exact same marginal probability densities $p_t(x)$ at each time step. This ODE is known as the **Probability Flow ODE** .
While the **SDE** describes a stochastic process with both drift and diffusion (random) components:

$$dx = f(x,t)dt + g(t)dw$$

the **Probability Flow ODE** removes the randomness and replaces it with a deterministic dynamics that preserves the marginal distributions:

$$dx_t = [f(x_t,t) - \frac{1}{2}g(t)^2\nabla_{x_t} \log p_t(x_t)]dt$$

subsitute f and g :

$$dx_t = \left[-\frac{1}{2}\beta(t)x_t - \frac{1}{2}\beta(t)\nabla_{x_t} \log p_t(x_t)\right]dt$$

It defines a **vector field**, such that any point sampled from the prior distribution p_T can be deterministically and smoothly transformed into a sample from the true data distribution p_0 by integrating along this vector field from $t = T$ to $t = 0$.

there may be a question, what is the different between ODE and reverse SDE? In Different cases, what should we use?
make sure our goal is clear :
we randomly sampling a x_T from a Gaussian distribution, sth like

$$x_T \sim \mathcal{N}(0, \mathbf{I})$$

and we want after using ODE or reverse SDE , we finally get a x_0' look pretty like the true data x_0 .

Feature	Stochastic SDE Sampling	Deterministic ODE Sampling
Mechanism	Randomness helps correct errors	Deterministic path; errors accumulate
Advantage	Potentially higher sample quality	Faster, unique path, better controllability
Disadvantage	Slower	Lower quality due to error accumulation

But, as we see above, whatever ODE or SDE are related to $\beta(t)$,which is kind of arbitrary. What if we have a more simple way?
that's what **flow matching** .

2.Flow Matching

2.1 General Understanding

In ODE(the same applies to the SDE) ,we want to learn a vector field function which can transform Gaussian distribution to the true data distribution.
the network need to learn a capability: given any coordinate (x, t) , it can output the direction of the vector (i.e., $v_\theta(x, t)$) at that point. Therefore, the trained neural network **itself becomes the vector field function**.
But , let's take ODE as an example. the vector field is below:

$$v_t(x_t) = \left[-\frac{1}{2}\beta(t)x_t - \frac{1}{2}\beta(t)\nabla_{x_t} \log p_t(x_t) \right]$$

the network wants to learn this vector field, but one term in the equation is too complex to learn : $\nabla_{x_t} \log p_t(x_t)$, because p_{data} is already complex, and $p_t(x_t)$ is adding noise to all probable x_0 in p_{data} and mixing them all.
Flow Matching tries to avoid this complex term , it directly uses a much simplier vector feild function which will be introduced below.

the simplest way to connect a point x_0 from p_{data} with a noise sampled from a distribution $z \sim \mathcal{N}(0, \mathbf{I})$ is a line!

Notice: in flow matching we usually use $t \in [0, 1]$ which $t = 1$ means noise, $t = 0$ means noise for any data point x_0 and a noise z , we can use interpolation with time t to represent x_t as:

$$x_t = (1 - t)x_0 + tz \qquad t \in [0, 1]$$

the vector field is defined by an ODE : $\frac{dx_t}{dt} = v_t(x_t)$, in flow matching ,

$$\frac{dx_t}{dt} = \frac{d}{dt}((1 - t)x_0 + tz)$$

$$v_t(x_t) = -x_0 + z = z - x_0$$

we can directly learn the target vector field $v_t(x_t)$ by using loss below:

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

But here lies a critical problem: during training, the model only observes the noisy input x_t and the time step t. It has no way of knowing which specific x_0 (original data) and which noise vector z were linearly interpolated to produce this particular x_t.
Imagine that a single point x_t in space can lie on infinitely many different linear interpolation paths — each corresponding to a different pair (x_0, z). Therefore, given only x_t, the model cannot uniquely determine what the target residual $z - x_0$ (or equivalently, the score or denoising direction) should be.

another problem is if we use only the linear field function, how can we get the generalization? we only solve the true data distribution but we can't create new "fake" data.

we will use conditional flow matching to solve these problem.

2.2 Conditional Flow Matching

Our ultimate goal: to train a neural network $u_\theta(x, t)$ such that it equals the **marginal vector field** $v_t(x)$. This $v_t(x)$ is the "macroscopic" vector field that transforms the entire prior distribution p_1 (for convenience, we use p_1 to denote the noise distribution) into the full data distribution p_0.
The challenge we face: directly minimizing the loss function

$$L_{\text{FM}} = \mathbb{E}_{t, x\sim p_t(x)}\left[\|u_\theta(x, t) - v_t(x)\|^2\right]$$

is impossible, because we have no knowledge of the true macroscopic vector field $v_t(x)$ or the marginal distribution $p_t(x)$.

There is how conditional flow matching work ! we construct a computable and simple loss function

$$L_{\text{CFM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_0\sim p_0(x),\, x_1\sim p_1(x)}\left[\|u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)\|^2\right]$$

Let’s break down this formula:

- We randomly sample a point x_0 from the data distribution p_0.
- We randomly sample a point x_1 from the noise distribution p_1.
- We randomly sample a time step $t \in [0, 1]$.
- We construct a point $x_t = (1-t)x_0 + tx_1$ along the straight-line path connecting x_0 and x_1.
- We ask the network u_θ to predict the velocity vector at x_t, and we expect it to output exactly the constant velocity vector of the line: $x_1 - x_0$.
  this loss function solve the problem in L_{FM} that we didn't know anything about $v_t(x)$, so we

can't use L_{FM} to train . but in L_{CFM} , we directly use exactly $x_1 - x_0$.
next question is : what is the connection between CFM and FM?

we are going to prove : the gradient of L_{CFM} is equal to the gradient to L_{FM} .
first, let's compute the gradient of L_{CFM} :

$$\nabla_\theta L_{CFM}(\theta) = \nabla_\theta \mathbb{E}_{t,x_0,x_1}\left[||u_\theta(x_t,t) - (x_1 - x_0)||^2\right]$$
$$= \mathbb{E}_{t,x_0,x_1}\left[2\left(u_\theta(x_t,t) - (x_1 - x_0)\right) \cdot \nabla_\theta u_\theta(x_t,t)\right]$$

but here we use a joint distribution : $p(t, x_0, x_1)$ and,

$$t \sim \mathcal{U}[0,1] \quad x_0 \sim p_0(x) \quad x_1 \sim p_1(x)$$

which is difficult to deal with. we will use the **law of total expectation** (i.e., $\mathbb{E}[Y] = \mathbb{E}_X[\mathbb{E}[Y \mid X]]$). We can decompose the expectation over (t, x_0, x_1) into two steps:

- First, take the expectation over (t, x_t),
- Then, condition on (t, x_t) and take the expectation over (x_0, x_1):

$$= \mathbb{E}_{t,x_t\sim p_t(x_t)}\left[\mathbb{E}_{x_0,x_1\sim p(x_0,x_1|x_t)}\left[2\left(u_\theta(x_t,t) - (x_1 - x_0)\right) \cdot \nabla_\theta u_\theta(x_t,t)\right]\right]$$

the $p_t(x_0, x_1|x_t)$ here is a **Posterior distribution** which means when we get a point x_t , its start and end respectively are x_0 , x_1 .
$\mathbb{E}_{x_0,x_1\sim p_t(x_0,x_1|x_t)}[\cdot]$ this term is independent with x_t, t, θ and only related to x_0, x_1 , we get :

$$\mathbb{E}_{x_0,x_1\sim p_t(x_0,x_1|x_t)}[\cdot] = 2\left(u_\theta(x_t,t) - \mathbb{E}_{x_0,x_1\sim p_t(x_0,x_1|x_t)}[x_1 - x_0]\right) \cdot \nabla_\theta u_\theta(x_t,t)$$

so as long as we find the relationship between $\mathbb{E}_{x_0,x_1\sim p_t(x_0,x_1|x_t)}[x_1 - x_0]$ and $v_t(x_t)$, we will find what's different between CFM and FM .
now, we are going to prove :

$$\mathbb{E}_{x_0,x_1\sim p_t(x_0,x_1|x_t)}[x_1 - x_0] = v_t(x_t)$$

Actually, we can prove a stronger statement:
the **marginal vector field** $v_t(x)$, which drives the evolution of the marginal distribution $p_t(x)$, equals the expectation of the **conditional vector field** $v_t(x \mid x_0, x_1)$ under the posterior distribution over endpoints given an intermediate point:

$$v_t(x) = \mathbb{E}_{p_t(x_0,x_1|x)}\left[v_t(x \mid x_0, x_1)\right]$$

the x here is equal to the x_t above, which means x is a sample in $[x_0, x_1]$. One Important thing is : **the** $v_t(x \mid x_0, x_1)$ **needn't any trajectories**, whatever is a starght line or a complex mapping.

at the beginning, it's important to clarify what we have and what we want.
1.Conditional Probability Flow: $\pi_t(x \mid x_0, x_1)$

A conditional probability density function that defines the distribution of an intermediate point x at time t, given fixed endpoints x_0 (data) and x_1 (noise). In the proof, the specific form is not required; however, for completeness, if we use the same linear interpolation path as in Flow Matching, it can be expressed as:

$$\pi_t(x \mid x_0, x_1) = \delta\left(x - [(1-t)x_0 + t x_1]\right)$$

2.Marginal Probability Flow: $p_t(x)$
The marginal probability density at time t, obtained by integrating over all possible endpoint pairs:

$$p_t(x) = \iint \pi_t(x \mid x_0, x_1)\, p_0(x_0)\, p_1(x_1)\, dx_0\, dx_1$$

It describes the aggregate distribution of particles across all trajectories.

3.Marginal Vector Field: $v_t(x)$
The velocity field that governs the evolution of $p_t(x)$ through the continuity equation(which will be introduced below):

$$\frac{\partial p_t}{\partial t} = -\nabla_x \cdot \left(p_t(x)\, v_t(x)\right)$$

It specifies how probability mass flows in space-time to transform $p_1 \to p_0$.

4.Conditional Vector Field: $v_t(x \mid x_0, x_1)$
The instantaneous velocity of a particle on a specific path from x_0 to x_1. For linear interpolation:

$$v_t(x \mid x_0, x_1) = \frac{d}{dt}[(1-t)x_0 + t x_1] = x_1 - x_0$$

Symbol	Category	Concept	Role in Proof
$\pi_t(x \mid x_0, x_1)$	**Defined / Known**	Density along a single deterministic path	Building block for $p_t(x)$
$v_t(x \mid x_0, x_1)$	**Defined / Known**	Velocity field for a single path	Directly computable target (e.g., $x_1 - x_0$)
$p_t(x)$	**Implicit / Unknown**	Marginal density over all paths	Macroscopic state; governed by continuity equation
$v_t(x)$	**Implicit / Unknown → Derived**	Velocity field driving $p_t(x)$	Ultimate learning objective; solved via posterior expectation

we have two tools , related to probability flow and defination of margin probablity.
1.**Continuity Equation (Probability Conservation Law):**

Any probability density $p_t(x)$ evolving under a vector field $v_t(x)$ satisfies:

$$\frac{\partial p_t(x)}{\partial t} = -\nabla_x \cdot (p_t(x) v_t(x))$$

where $\nabla_x \cdot$ denotes the divergence operator. This equation expresses local conservation of probability mass and applies both to:

- The **marginal flow**: $p_t(x)$
- The **conditional flow**: $\pi_t(x \mid x_0, x_1)$

2.**Definition of Marginal Probability:**
The marginal density $p_t(x)$ is obtained by integrating over all possible initial and final points x_0, x_1:

$$p_t(x) = \iint \pi_t(x \mid x_0, x_1)\, p_0(x_0)\, p_1(x_1)\, dx_0\, dx_1$$

For brevity, define $p(x_0, x_1) = p_0(x_0) p_1(x_1)$.

Step-by-Step Proof

Step 1: Time derivative of the marginal density

Differentiate both sides of the marginal definition with respect to time t:

$$\frac{\partial p_t(x)}{\partial t} = \frac{\partial}{\partial t} \left(\iint \pi_t(x \mid x_0, x_1)\, p(x_0, x_1)\, dx_0\, dx_1 \right)$$

Since the integral is over x_0, x_1, we can interchange differentiation and integration:

$$\frac{\partial p_t(x)}{\partial t} = \iint \frac{\partial \pi_t(x \mid x_0, x_1)}{\partial t}\, p(x_0, x_1)\, dx_0\, dx_1$$

Step 2: Apply the continuity equation to the conditional flow

For fixed x_0, x_1, the conditional density $\pi_t(x \mid x_0, x_1)$ evolves according to its own continuity equation driven by the conditional vector field $v_t(x \mid x_0, x_1)$:

$$\frac{\partial \pi_t(x \mid x_0, x_1)}{\partial t} = -\nabla_x \cdot (\pi_t(x \mid x_0, x_1)\, v_t(x \mid x_0, x_1))$$

Substitute this into the previous expression:

$$\frac{\partial p_t(x)}{\partial t} = \iint -\nabla_x \cdot (\pi_t(x \mid x_0, x_1)\, v_t(x \mid x_0, x_1)) p(x_0, x_1)\, dx_0\, dx_1$$

Step 3: Swap divergence and integration

The divergence operator $\nabla_x \cdot$ acts only on x, while the integral is over x_0, x_1. These operations commute:

$$\frac{\partial p_t(x)}{\partial t} = -\nabla_x \cdot \left(\iint \pi_t(x \mid x_0, x_1)\, v_t(x \mid x_0, x_1)\, p(x_0, x_1)\, dx_0\, dx_1 \right)$$

Step 4: Compare with the marginal continuity equation

From the continuity equation for the marginal flow:

$$\frac{\partial p_t(x)}{\partial t} = -\nabla_x \cdot \left(p_t(x)\, v_t(x) \right)$$

Comparing this with the expression derived above, we conclude:

$$p_t(x)\, v_t(x) = \iint \pi_t(x \mid x_0, x_1)\, v_t(x \mid x_0, x_1)\, p(x_0, x_1)\, dx_0\, dx_1$$

Step 5: Solve for the marginal vector field $v_t(x)$

Divide both sides by $p_t(x)$ (assuming $p_t(x) > 0$):

$$v_t(x) = \frac{1}{p_t(x)} \iint \pi_t(x \mid x_0, x_1)\, v_t(x \mid x_0, x_1)\, p(x_0, x_1)\, dx_0\, dx_1$$

Step 6: Rewrite using Bayes' Theorem

Recall the posterior distribution:

$$p_t(x_0, x_1 \mid x) = \frac{\pi_t(x \mid x_0, x_1)\, p(x_0, x_1)}{p_t(x)}$$

Thus, the right-hand side becomes:

$$v_t(x) = \iint v_t(x \mid x_0, x_1)\, p_t(x_0, x_1 \mid x)\, dx_0\, dx_1$$

Step 7: Final form — expectation under the posterior

By definition, this is the expected value of $v_t(x \mid x_0, x_1)$ conditioned on observing x at time t:

$$\boxed{v_t(x) = \mathbb{E}_{p_t(x_0, x_1 \mid x)} \left[v_t(x \mid x_0, x_1) \right]}$$

Q.E.D.

Applicating this conclusion to Linear Interpolation Paths in **Continuous Flow Matching**, we consider simple straight-line paths:

$$x_t = (1 - t)x_0 + t x_1$$

For such paths, the conditional vector field is constant:

$$v_t(x \mid x_0, x_1) = \frac{dx_t}{dt} = x_1 - x_0$$

Substituting into the main result gives:

$$v_t(x) = \mathbb{E}_{p_t(x_0, x_1 | x)}[x_1 - x_0]$$

This elegant formula shows that even though each trajectory follows a deterministic path, the effective marginal vector field at any point x is determined by averaging over all possible ways x could have been generated from pairs (x_0, x_1), weighted by their posterior likelihood.

Now, look back to our first goal: we want to prove that $\nabla_\theta L_{CFM}(\theta) = \nabla_\theta L_{FM}(\theta)$.

$$\begin{aligned}
\nabla_\theta L_{CFM}(\theta) &= \nabla_\theta \mathbb{E}_{t,x_0,x_1}\left[||u_\theta(x_t,t) - (x_1 - x_0)||^2\right] \\
&= \mathbb{E}_{t,x_0,x_1}\left[2\left(u_\theta(x_t,t) - (x_1 - x_0)\right) \cdot \nabla_\theta u_\theta(x_t,t)\right] \\
&= \mathbb{E}_{t,x_t\sim p_t(x_t)}\left[\mathbb{E}_{x_0,x_1\sim p(x_0,x_1|x_t)}\left[2\left(u_\theta(x_t,t) - (x_1 - x_0)\right) \cdot \nabla_\theta u_\theta(x_t,t)\right]\right] \\
\mathbb{E}_{x_0,x_1\sim p_t(x_0,x_1|x_t)}[\cdot] &= 2\left(u_\theta(x_t,t) - \mathbb{E}_{x_0,x_1\sim p_t(x_0,x_1|x_t)}[x_1 - x_0]\right) \cdot \nabla_\theta u_\theta(x_t,t)
\end{aligned}$$

according to the flow proof:

$$\mathbb{E}_{x_0,x_1\sim p_t(x_0,x_1|x_t)}[x_1 - x_0] = v_t(x_t)$$

A simple substitution and rearrangement suffices to obtain the result：

$$\nabla_\theta L_{FM}(\theta) = \nabla_\theta \mathbb{E}_{t,x\sim p_t(x)}[||u_\theta(x,t) - v_t(x)||^2] = \mathbb{E}_{t,x\sim p_t(x)}\left[2\left(u_\theta(x,t) - v_t(x)\right) \cdot \nabla_\theta u_\theta(x,t)\right] = \nabla_\theta L_{CFM}(\theta$$

3.Reference

blog:https://yang-song.net/blog/2021/score/ which interprete score function and gives a high level SDE, ODE Perspectives on Diffusion and Flow Matching
Gemini 2.5 pro