

# Flow-Based Perspective:

## From NFs to Flow Matching

Gunwoo Cho

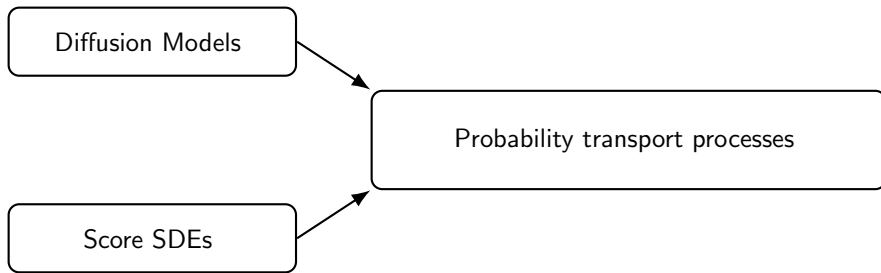
SNU Statistics

August 3, 2026

# Table of Contents

- Flow-Based Models: Normalizing Flows and Neural ODEs
- Flow Matching Framework
- Constructing Probability Paths and Velocities Between Distributions
- Properties of the Canonical Affine Flow
- Questions

# Introduction



Both frameworks describe how probability density is transported between a complex data distribution and a simple prior distribution.

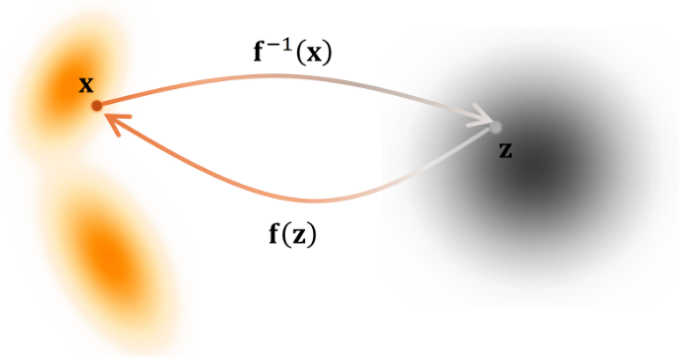
# Change-of-Variables Formula

## Change-of-Variables Formula of Densities.

Given an invertible transformation  $f$ , the density of  $x = f(z)$ , where  $z \sim p_{\text{prior}}$ , is given by

$$p(x) = p_{\text{prior}}(z) \left| \det \frac{\partial f^{-1}(x)}{\partial x} \right|, \quad \text{where } z = f^{-1}(x).$$

# Normalizing Flows



**Figure 5.1: Illustration of sample movement of NF under an invertible map.** It consists of a sequence of invertible functions  $f : z \mapsto x$  that transform latent variable  $z$  into a data  $x$ , together with the inverse mapping  $f^{-1} : x \mapsto z$  that reconstructs the data. An NF resembles an encoder-decoder structure, but with the encoder realized as a smooth invertible map and the decoder given exactly by its inverse. The corresponding change in density can be computed via the change-of-variables formula, as given in Equation (5.0.1).

*Source: Created by the authors.*

# Normalizing Flows

**Basic idea.** Normalizing Flows model a complex data distribution  $p_{\text{data}}(x)$  by transforming a simple prior distribution  $p_{\text{prior}}(z)$ , such as  $\mathcal{N}(0, I)$ , through an invertible mapping

$$f_{\phi} : \mathbb{R}^D \rightarrow \mathbb{R}^D.$$

The forward and inverse transformations are

$$x = f_{\phi}(z), \quad z = f_{\phi}^{-1}(x), \quad z \sim p_{\text{prior}}.$$

Using the change-of-variables formula, the model likelihood is

$$\log p_{\phi}(x) = \log p_{\text{prior}}(z) + \log \left| \det \frac{\partial f_{\phi}^{-1}(x)}{\partial x} \right|.$$

Equivalently,

$$\log p_{\phi}(x) = \log p_{\text{prior}}(z) - \log \left| \det \frac{\partial f_{\phi}(z)}{\partial z} \right|.$$

# Normalizing Flows

**Training Objective.** The parameters  $\phi$  are learned by maximizing the expected log-likelihood of the observed data:

$$\mathcal{L}_{\text{NF}}(\phi) = \mathbb{E}_{x \sim p_{\text{data}}} [\log p_{\phi}(x)] .$$

## Key computational challenge

Evaluating the likelihood requires computing the Jacobian determinant

$$\det \left( \frac{\partial f_{\phi}(z)}{\partial z} \right) ,$$

which generally costs

$$\mathcal{O}(D^3).$$

Therefore, Normalizing Flows require transformations that are both

- sufficiently expressive
- invertible with efficiently computable Jacobian determinants.

# Normalizing Flows

**Constructing Invertible Transformations.** Normalizing Flows compose  $L$  trainable invertible mappings:

$$f_\phi = f_{L-1} \circ f_{L-2} \circ \cdots \circ f_0,$$

where each  $f_k$  has an efficiently computable Jacobian determinant. The samples are transformed sequentially as

$$x_{k+1} = f_k(x_k), \quad k = 0, \dots, L-1,$$

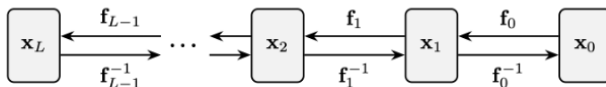
with

$$x_0 = z \sim p_{\text{prior}}, \quad x_L = x.$$

Applying the change-of-variables formula, the log-density is

$$\log p_\phi(x) = \log p_{\text{prior}}(x_0) - \sum_{k=0}^{L-1} \log \left| \det \frac{\partial f_k(x_k)}{\partial x_k} \right|$$

# Normalizing Flows



**Figure 5.2: Illustration of a NF.** An NF consists of a stack of invertible maps  $\mathbf{f}_\phi = \mathbf{f}_{L-1} \circ \mathbf{f}_{L-2} \circ \dots \circ \mathbf{f}_0$ . The transformation maps latent samples  $\mathbf{x}_0 \sim p_{\text{prior}}$  to data samples  $\mathbf{x}_L \sim p_{\text{data}}$ .

*Source: Created by the authors.*

# Examples of Normalizing Flows

## Planar Flows

A Planar Flow uses a simple residual transformation

$$f_k(z) = z + u_k h(w_k^\top z + b_k),$$

where

$$u_k, w_k \in \mathbb{R}^D, \quad b_k \in \mathbb{R},$$

and  $h(\cdot)$  is an activation function.

The Jacobian is

$$\frac{\partial f_k(z)}{\partial z} = I + u_k [h'(w_k^\top z + b_k) w_k]^\top.$$

Then its determinant reduces to

$$\left| \det \frac{\partial f_k(z)}{\partial z} \right| = |1 + h'(w_k^\top z + b_k) u_k^\top w_k|.$$

# Examples of Normalizing Flows

## Residual Flows

A Residual Flow defines each invertible transformation as

$$f_k(z) = z + v_k(z),$$

where  $v_k$  is contractive (Lipschitz constant  $< 1$ ).

This condition guarantees invertibility through the Banach fixed-point theorem. Since

$$\frac{\partial f_k(z)}{\partial z} = I + \frac{\partial v_k(z)}{\partial z},$$

the log-determinant can be written as

$$\log \left| \det \frac{\partial f_k(z)}{\partial z} \right| = \text{Tr} \left[ \log \left( I + \frac{\partial v_k(z)}{\partial z} \right) \right].$$

Using the matrix logarithm expansion,

$$\log \left| \det \frac{\partial f_k(z)}{\partial z} \right| = \sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m} \text{Tr} \left[ \left( \frac{\partial v_k(z)}{\partial z} \right)^m \right].$$

# Continuous-Time NFs

Normalizing Flows are typically constructed as a sequence of  $L$  discrete invertible transformations:

$$x_{k+1} = f_k(x_k).$$

From the residual flows perspective, each layer can be written as

$$x_{k+1} = f_k(x_k) := x_k + v_{\phi_k}(x_k, k), \quad k = 0, \dots, L-1,$$

where  $v_{\phi_k}(\cdot, k)$  is a layer-dependent velocity field parameterized by a neural network.

The velocity field specifies how each point should be moved at layer  $k$ . By repeatedly applying these small transformations, the model gradually transports a simple prior distribution into a complex data distribution:

$$x_0 \sim p_{\text{prior}} \longrightarrow x_1 \longrightarrow \dots \longrightarrow x_L \sim p_{\text{data}}.$$

# Continuous-Time NFs

The residual update can be interpreted as an Euler discretization:

$$x_{k+1} = x_k + \Delta t v_\phi(x_k, t_k).$$

As the number of layers increases and the step size vanishes,

$$L \rightarrow \infty, \quad \Delta t \rightarrow 0,$$

the discrete updates converge to the continuous-time ODE

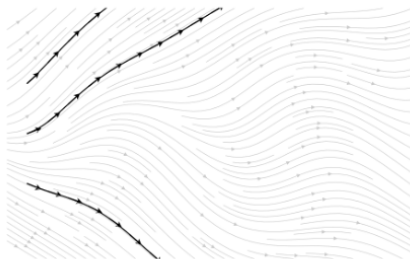
$$\frac{dx(t)}{dt} = v_\phi(x(t), t).$$

# Continuous-Time NFs

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) := \mathbf{x}_k + \mathbf{v}_\phi(\mathbf{x}_k, k)$$



$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{v}_\phi(\mathbf{x}(t), t)$$



**Figure 5.3: Discrete- vs. continuous-time normalizing flows.** (Left) A discrete NF transports samples by a finite sequence of invertible maps  $\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k)$ , yielding stepwise, non-crossing trajectories (dots with arrows). (Right) A continuous NF (Neural ODE) evolves states along integral curves of  $\frac{d\mathbf{x}(t)}{dt} = \mathbf{v}_\phi(\mathbf{x}(t), t)$ , where black paths with tangent arrows are shown over the gray vector field.

*Source: Created by the authors.*

# Neural ODEs

A Neural ODE defines a continuous transformation through

$$\frac{dx(t)}{dt} = v_{\phi}(x(t), t), \quad t \in [0, T],$$

where

$$x(t) \in \mathbb{R}^D$$

is the state at time  $t$ , and

$$v_{\phi}(x(t), t)$$

is a neural vector field parameterized by  $\phi$ .

Given an initial condition

$$x(0) \sim p_{\text{prior}},$$

the ODE continuously evolves each sample along a trajectory

$$x(0) \longrightarrow x(t) \longrightarrow x(T).$$

# Neural ODEs

As prior samples evolve under the same vector field, the ODE induces a family of time-dependent marginal distributions:

$$p_{\text{prior}}(\cdot) = p_{\phi}(\cdot, 0) \longrightarrow p_{\phi}(\cdot, t) \longrightarrow p_{\phi}(\cdot, T).$$

The goal is to learn  $v_{\phi}$  such that the terminal distribution matches the target data distribution:

$$p_{\phi}(\cdot, T) \approx p_{\text{data}}(\cdot).$$

This is similar to the probability flow ODE, which also describes deterministic continuous-time transport through a time-dependent vector field.

# Instantaneous Change-of-Variables

## Lemma 5.1.1: Instantaneous Change of Variables

Let  $z(t)$  be a continuous random process with time-dependent density  $p(z(t), t)$ , and suppose it evolves according to the ODE

$$\frac{dz(t)}{dt} = F(z(t), t).$$

Assume that  $F$  is uniformly Lipschitz in  $z$  and continuous in  $t$ . Then, along the trajectory  $z(t)$ , the log-density satisfies

$$\frac{d}{dt} \log p(z(t), t) = -\nabla_z \cdot F(z(t), t).$$

# Instantaneous Change-of-Variables

## Proof Approach 1: Change-of-Variables Formula

Using the Euler discretization

$$z_{t+\Delta t} = z_t + \Delta t F(z_t, t),$$

the change-of-variables formula gives

$$\begin{aligned} \log p_{t+\Delta t}(z_{t+\Delta t}) &= \log p_t(z_t) - \log |\det(I + \Delta t \nabla_z F(z_t, t))| \\ &= \log p_t(z_t) - \text{Tr}[\log(I + \Delta t \nabla_z F(z_t, t))]. \end{aligned}$$

Using

$$\log(I + \Delta t A) = \Delta t A + \mathcal{O}(\Delta t^2),$$

we obtain

$$\log p_{t+\Delta t}(z_{t+\Delta t}) = \log p_t(z_t) - \Delta t \text{Tr}(\nabla_z F(z_t, t)) + \mathcal{O}(\Delta t^2).$$

Dividing by  $\Delta t$  and taking  $\Delta t \rightarrow 0$ ,

$$\boxed{\frac{d}{dt} \log p(z(t), t) = -\text{Tr}(\nabla_z F) = -\nabla_z \cdot F(z(t), t)}.$$

# Instantaneous Change-of-Variables

## Proof Approach 2: Continuity Equation

The density satisfies

$$\frac{\partial p(z, t)}{\partial t} = -\nabla_z \cdot (F(z, t)p(z, t)).$$

Expanding the divergence,

$$\frac{\partial p}{\partial t} = -[(\nabla_z \cdot F)p + F \cdot \nabla_z p].$$

Along a trajectory satisfying

$$\frac{dz(t)}{dt} = F(z(t), t),$$

the total derivative is

$$\begin{aligned} \frac{d}{dt}p(z(t), t) &= \nabla_z p \cdot \frac{dz(t)}{dt} + \frac{\partial p}{\partial t} \\ &= \nabla_z p \cdot F - [(\nabla_z \cdot F)p + F \cdot \nabla_z p] \\ &= -(\nabla_z \cdot F)p. \end{aligned}$$

Dividing by  $p(z(t), t)$ ,

# Connection to Discrete-Time NFs

For a discrete Normalizing Flow,

$$x_{k+1} = f_k(x_k),$$

the log-density is

$$\log p_\phi(x_L) = \log p_{\text{prior}}(x_0) - \sum_{k=0}^{L-1} \log \left| \det \frac{\partial f_k(x_k)}{\partial x_k} \right|.$$

For a Continuous Normalizing Flow,

$$\log p_\phi(x(T), T) = \log p_{\text{prior}}(x(0), 0) - \int_0^T \nabla_x \cdot v_\phi(x(t), t) dt.$$

# Neural ODEs

For a data point  $x$ , the latent state  $z(t)$  is obtained by solving the ODE backward from  $t = T$  to  $t = 0$ :

$$\frac{dz(t)}{dt} = v_{\phi}(z(t), t), \quad z(T) = x.$$

The corresponding log-likelihood is

$$\log p_{\phi}(x, T) = \log p_{\text{prior}}(z(0)) - \int_0^T \nabla_z \cdot v_{\phi}(z(t), t) dt.$$

Since  $p_{\text{prior}}$  is typically a tractable distribution such as  $\mathcal{N}(0, I)$ , the prior term

$$\log p_{\text{prior}}(z(0))$$

can be evaluated directly.

Maximizing the NODE likelihood requires backpropagation through the ODE solver. Training remains computationally expensive due to repeated numerical integration.

# Neural ODEs

After training, sampling proceeds in the forward direction.  
First, draw an initial sample from the prior:

$$x(0) \sim p_{\text{prior}}.$$

Then integrate the learned ODE

$$\frac{dx(t)}{dt} = v_{\phi^*}(x(t), t)$$

from  $t = 0$  to  $t = T$ . Equivalently,

$$x(T) = x(0) + \int_0^T v_{\phi^*}(x(t), t) dt.$$

The terminal state approximately follows the data distribution:

$$x(T) \sim p_{\phi^*}(\cdot, T) \approx p_{\text{data}}.$$

# Neural ODEs

The divergence is the trace of the Jacobian:

$$\nabla_z \cdot v_\phi(z, t) = \text{Tr} \left( \frac{\partial v_\phi(z, t)}{\partial z} \right).$$

In high dimensions, it can be estimated using Hutchinson's estimator:

$$\text{Tr}(A) = \mathbb{E}_\epsilon [\epsilon^\top A \epsilon], \quad \mathbb{E}[\epsilon \epsilon^\top] = I.$$

Hence,

$$\nabla_z \cdot v_\phi(z, t) \approx \epsilon^\top \frac{\partial v_\phi(z, t)}{\partial z} \epsilon.$$

# Flow Matching Framework

Score-based SDEs and Neural ODEs view generative modeling as learning a continuous-time flow that transports a simple prior to the data distribution:

$$\epsilon \sim p_{\text{prior}} \longrightarrow x \sim p_{\text{data}}.$$

Flow Matching generalizes this idea to transport between arbitrary endpoint distributions:

$$p_{\text{src}} \longrightarrow p_{\text{tgt}},$$

where both distributions are assumed to be easy to sample from. It learns a time-dependent vector field  $v_t(x_t)$  defining the ODE

$$\frac{dx_t}{dt} = v_t(x_t),$$

whose induced flow follows a prescribed probability path

$$\{p_t\}_{t \in [0,1]}, \quad p_0 = p_{\text{src}}, \quad p_1 = p_{\text{tgt}}.$$

Standard generative modeling is the special case

$$p_{\text{src}} = p_{\text{prior}}, \quad p_{\text{tgt}} = p_{\text{data}}.$$

# Lesson from Score-Based Methods

## Step 1: Defining a Conditional Path and Its Marginal Densities.

Score-based diffusion defines a path

$$p_0 = p_{\text{data}}, \quad p_1 = p_{\text{prior}}.$$

For  $x_0 \sim p_{\text{data}}$ , the conditional path is defined as

$$p_t(x_t \mid x_0) = \mathcal{N}(x_t; \alpha_t x_0, \sigma_t^2 I),$$

or equivalently,

$$x_t = \alpha_t x_0 + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

Marginalizing over  $x_0$  induces the marginal density

$$p_t(x_t) := \int p_t(x_t \mid x_0) p_{\text{data}}(x_0) dx_0.$$

As the noise level increases,  $p_t$  evolves from the data distribution toward the Gaussian prior.

# Lesson from Score-Based Methods

## Step 2: Velocity Field.

From the Fokker–Planck equation, the dynamics can be written as an ODE:

$$\frac{dx}{dt} = v_t(x), \quad v_t(x) = f(t)x - \frac{1}{2}g^2(t)\nabla_x \log p_t(x).$$

This transports samples:

$$p_{\text{data}} \longleftrightarrow p_{\text{prior}}.$$

In practice, the score  $\nabla_x \log p_t(x)$  is learned instead of the full density.

# Lesson from Score-Based Methods

## Step 3: Learning via the Conditional Strategy.

The ideal objective learns the marginal score:

$$\mathcal{L}_{\text{SM}}(\phi) = \mathbb{E}_{\substack{t \sim \mathcal{U}[0,1] \\ x_t \sim p_t}} \left[ \|s_\phi(x_t, t) - \nabla_{x_t} \log p_t(x_t)\|^2 \right].$$

Since the marginal score is generally intractable, we use conditional score.

By

$$\nabla_{x_t} \log p_t(x_t) = \mathbb{E}_{x_0 \sim p(\cdot | x_t)} [\nabla_{x_t} \log p_t(x_t | x_0)],$$

the ideal objective can be written as

$$\mathcal{L}_{\text{SM}}(\phi) = \underbrace{\mathbb{E}_{\substack{t \sim \mathcal{U}[0,1], x_0 \sim p_{\text{data}} \\ x_t \sim p_t(\cdot | x_0)}} \left[ \|s_\phi(x_t, t) - \nabla_{x_t} \log p_t(x_t | x_0)\|^2 \right]}_{\mathcal{L}_{\text{DSM}}(\phi)} + C,$$

where  $C$  is independent of  $\phi$ .

Therefore,  $\mathcal{L}_{\text{SM}}$  and  $\mathcal{L}_{\text{DSM}}$  have the same minimizer:

$$s^*(x_t, t) = \nabla_{x_t} \log p_t(x_t).$$

# Flow Matching Framework

## Step 1: Defining a Conditional Path and Its Marginal Densities.

Consider arbitrary endpoint distributions on  $\mathbb{R}^D$ :

$$p_0(x) = p_{\text{src}}(x), \quad p_1(x) = p_{\text{tgt}}(x).$$

Flow Matching defines intermediate densities through a latent variable  $z \sim \pi(z)$  and a conditional path  $p_t(x_t | z)$ :

$$p_t(x_t) = \int p_t(x_t | z) \pi(z) dz.$$

The choices of  $\pi(z)$  and  $p_t(\cdot | z)$  are made to satisfy the endpoint conditions. Common choices are

$$z = (x_0, x_1)$$

for two-sided conditioning between arbitrary endpoint distributions, and

$$z = x_0 \quad \text{or} \quad z = x_1$$

for one-sided, diffusion-like conditioning.

# Flow Matching Framework

## Step 2: Velocity Field.

Flow Matching seeks a velocity field  $v_t(x)$  whose ODE flow transports samples along the prescribed path  $\{p_t\}_{t \in [0,1]}$ :

$$\frac{dx(t)}{dt} = v_t(x(t)), \quad t \in [0, 1].$$

The induced marginal density must match  $p_t$  at every time  $t$ . This condition is expressed by the continuity equation

$$\frac{\partial p_t(x)}{\partial t} + \nabla \cdot (v_t(x)p_t(x)) = 0.$$

Any velocity field satisfying this equation transports

$$p_{\text{src}} \longrightarrow p_{\text{tgt}}$$

while matching all intermediate densities.

Unlike Gaussian diffusion, the velocity field is generally not unique.

# Flow Matching Framework

## Step 3: Learning via the Conditional Strategy.

The ideal Flow Matching objective is

$$\mathcal{L}_{\text{FM}}(\phi) = \mathbb{E}_{t, x_t \sim p_t} \left[ \|v_\phi(x_t, t) - v_t(x_t)\|^2 \right].$$

Since the marginal velocity  $v_t(x_t)$  is generally intractable, FM uses a conditional velocity field  $v_t(x_t | z)$ :

$$\mathcal{L}_{\text{FM}}(\phi) = \underbrace{\mathbb{E}_{\substack{t, z \sim \pi(z) \\ x_t \sim p_t(\cdot|z)}} \left[ \|v_\phi(x_t, t) - v_t(x_t | z)\|^2 \right]}_{\mathcal{L}_{\text{CFM}}(\phi)} + C,$$

where  $C$  is independent of  $\phi$ .

Thus, minimizing conditional flow matching is equivalent to minimizing the intractable marginal objective. The learned predictor recovers a valid marginal velocity:

$$v^*(x_t, t) = \mathbb{E}_{z \sim p(\cdot|x_t)} [v_t(x_t | z)] = v_t(x_t).$$

# Theorem 5.2.1: Equivalence of Objectives

## Theorem 5.2.1

The marginal Flow Matching objective and the conditional Flow Matching objective differ only by a constant:

$$\mathcal{L}_{\text{FM}}(\phi) = \mathcal{L}_{\text{CFM}}(\phi) + C,$$

where  $C$  is independent of  $\phi$ .

Therefore, both objectives have the same minimizer. In particular,

$$v^*(x_t, t) = v_t(x_t), \quad x_t \sim p_t.$$

## Marginal Velocity

The marginal velocity is obtained by averaging the conditional velocity:

$$v_t(x_t) = \mathbb{E}_{z \sim p(\cdot | x_t)} [v_t(x_t | z)].$$

# Theorem 5.2.1: Equivalence of Objectives

## Proof Sketch.

Let

$$Y = v_t(x_t | z), \quad m(x_t, t) = \mathbb{E}[Y | x_t, t] = v_t(x_t).$$

The conditional Flow Matching loss is

$$\mathcal{L}_{\text{CFM}}(\phi) = \mathbb{E} \left[ \|v_\phi(x_t, t) - Y\|^2 \right].$$

By the conditional variance decomposition,

$$\mathbb{E} \left[ \|v_\phi(x_t, t) - Y\|^2 | x_t, t \right] = \|v_\phi(x_t, t) - m(x_t, t)\|^2 + \mathbb{E} \left[ \|Y - m(x_t, t)\|^2 | x_t, t \right].$$

Taking expectation gives

$$\mathcal{L}_{\text{CFM}}(\phi) = \mathcal{L}_{\text{FM}}(\phi) + C',$$

where  $C'$  is independent of  $\phi$ . Hence, the two losses have the same minimizer:

$$v^*(x_t, t) = m(x_t, t) = v_t(x_t). \quad \square$$

# Comparison: Diffusion, Flow Matching, and NODEs

| Aspect    | Diffusion / Gaussian FM                                  | General FM                                               |
|-----------|----------------------------------------------------------|----------------------------------------------------------|
| Endpoints | $\mathcal{N}(0, I) \rightarrow p_{\text{data}}$          | $p_{\text{src}} \rightarrow p_{\text{tgt}}$              |
| Path      | $\mathcal{N}(x_t; \alpha_t x_0, \sigma_t^2 I)$           | $p_t(x_t   z), z \sim \pi(z)$                            |
| Marginal  | $\int p_t(x_t   x_0) p_{\text{data}}(x_0) dx_0$          | $\int p_t(x_t   z) \pi(z) dz$                            |
| Velocity  | Score-induced PF-ODE velocity                            | Marginalized conditional velocity                        |
| Objective | $\mathcal{L}_{\text{SM}} = \mathcal{L}_{\text{DSM}} + C$ | $\mathcal{L}_{\text{FM}} = \mathcal{L}_{\text{CFM}} + C$ |

## Connection to NODEs

Flow Matching trains the velocity field through direct regression, eliminating the need for ODE integration during training. By contrast, CNFs and NODEs generally rely on repeated ODE solves for likelihood optimization.

# Theorem 5.2.2: Mass Conservation Criterion

**Background.** Suppose a probability path  $\{p_t\}_{t \in [0,1]}$  is prescribed, and a velocity field  $v_t$  defines the particle dynamics  $\frac{dx(t)}{dt} = v_t(x(t))$ .

The central question is:

When does the density induced by this ODE flow exactly match  $p_t$  at every time  $t$ ?

## Theorem 5.2.2

The flow-induced density matches the prescribed path,

$$p_t^{\text{fwd}} = p_t \quad \forall t \in [0, 1],$$

if and only if the pair  $(p_t, v_t)$  satisfies the continuity equation

$$\frac{\partial p_t(x)}{\partial t} + \nabla_x \cdot (p_t(x) v_t(x)) = 0.$$

# Proof Sketch: Necessity

Assume that the ODE-induced density matches the prescribed density:

$$p_t^{\text{fwd}} = p_t.$$

For any smooth compactly supported test function  $\varphi$ , the pushforward property gives

$$\int p_t^{\text{fwd}}(x) \varphi(x) dx = \int p_0(y) \varphi(\Psi_t(y)) dy.$$

Differentiating with respect to  $t$ ,

$$\frac{d}{dt} \int p_t^{\text{fwd}}(x) \varphi(x) dx = \int p_0(y) \nabla \varphi(\Psi_t(y)) \cdot v_t(\Psi_t(y)) dy.$$

Changing variables  $x = \Psi_t(y)$ , we obtain

$$\int \partial_t p_t^{\text{fwd}}(x) \varphi(x) dx = \int p_t^{\text{fwd}}(x) \nabla \varphi(x) \cdot v_t(x) dx.$$

# Proof Sketch: Necessity

Using integration by parts and the compact support of  $\varphi$ ,

$$\int p_t^{\text{fwd}}(x) \nabla \varphi(x) \cdot v_t(x) dx = - \int \varphi(x) \nabla_x \cdot (p_t^{\text{fwd}}(x) v_t(x)) dx.$$

Hence,

$$\int [\partial_t p_t^{\text{fwd}}(x) + \nabla_x \cdot (p_t^{\text{fwd}}(x) v_t(x))] \varphi(x) dx = 0.$$

Since  $\varphi$  is arbitrary,

$$\partial_t p_t^{\text{fwd}} + \nabla_x \cdot (p_t^{\text{fwd}} v_t) = 0.$$

Because  $p_t^{\text{fwd}} = p_t$ , we get

$$\partial_t p_t + \nabla_x \cdot (p_t v_t) = 0.$$

# Proof Sketch: Sufficiency

Conversely, suppose that the prescribed path satisfies

$$\partial_t p_t + \nabla_x \cdot (p_t v_t) = 0,$$

with initial condition

$$p_0 = p_{\text{src}}.$$

The flow-induced density  $p_t^{\text{fwd}}$  also satisfies the same continuity equation:

$$\partial_t p_t^{\text{fwd}} + \nabla_x \cdot (p_t^{\text{fwd}} v_t) = 0,$$

and has the same initial condition:

$$p_0^{\text{fwd}} = p_0.$$

Under standard regularity assumptions on  $v_t$  and  $p_t$ , the continuity equation has a unique solution. Therefore,

$$p_t^{\text{fwd}} = p_t \quad \forall t \in [0, 1].$$

# Prop 5.2.3: Marginal VF Generates Given Marginal Density

**Background.** Given conditional paths  $p_t(x | z)$  and conditional velocity fields  $v_t(x | z)$ , how can we construct a marginal velocity field  $v_t(x)$  that induces the prescribed marginal path

$$p_t(x) = \int p_t(x | z) \pi(z) dz?$$

## Proposition 5.2.3

If each conditional velocity field  $v_t(\cdot | z)$  induces the conditional density path  $p_t(\cdot | z)$ , then the marginal velocity field

$$v_t(x) = \int v_t(x | z) \frac{p_t(x | z) \pi(z)}{p_t(x)} dz$$

induces the marginal density path  $p_t(x)$ .

# Proof Sketch

Since each conditional pair satisfies the continuity equation,

$$\frac{d}{dt}p_t(x | z) = -\nabla_x \cdot (v_t(x | z)p_t(x | z)).$$

Starting from the marginal density,

$$p_t(x) = \int p_t(x | z)\pi(z) dz,$$

we differentiate with respect to  $t$ :

$$\frac{d}{dt}p_t(x) = \int \frac{d}{dt}p_t(x | z)\pi(z) dz.$$

Substituting the conditional continuity equation gives

$$\frac{d}{dt}p_t(x) = -\nabla_x \cdot \left( \int v_t(x | z)p_t(x | z)\pi(z) dz \right).$$

# Proof Sketch

To satisfy the marginal continuity equation,

$$\frac{d}{dt}p_t(x) = -\nabla_x \cdot (v_t(x)p_t(x)),$$

it is enough to define

$$v_t(x)p_t(x) = \int v_t(x | z)p_t(x | z)\pi(z) dz.$$

Thus,

$$v_t(x) = \int v_t(x | z) \frac{p_t(x | z)\pi(z)}{p_t(x)} dz.$$

Equivalently,

$$v_t(x) = \mathbb{E}_{z \sim p(\cdot | x)} [v_t(x | z)].$$

Therefore,

$\text{conditional continuity} \implies \text{marginal continuity}$

# Probability Paths and Velocities

Flow Matching aims to transform a source distribution  $p_0$  into a target distribution  $p_1$  through a continuous probability path

$$\{p_t\}_{t \in [0,1]}.$$

The evolution of this path is governed by a time-dependent velocity field  $v_t : \mathbb{R}^d \rightarrow \mathbb{R}^d$ , whose trajectories satisfy

$$\frac{dX_t}{dt} = v_t(X_t).$$

The probability path and velocity field must be consistent through the continuity equation,

$$\partial_t p_t(x) + \nabla_x \cdot (p_t(x) v_t(x)) = 0.$$

# Probability Paths and Velocities

Ideally, one would directly learn the marginal velocity field by minimizing

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, X_t \sim p_t} \left[ \|v_\theta(X_t, t) - v_t(X_t)\|^2 \right].$$

However, for complex target distributions, the true marginal velocity  $v_t(x)$  is generally unknown and cannot be evaluated directly.

**Conditional Flow Matching** resolves this difficulty by introducing a conditioning variable  $Z$  and constructing tractable conditional paths and velocities,

$$p_t(x | Z), \quad v_t(x | Z),$$

from which intermediate states can be sampled and target velocities can be computed in closed form.

# Gaussian-to-Gaussian Bridge

Let the marginal probability path be

$$p_t(x) = \mathcal{N}(x; \mu(t), \sigma^2(t)I), \quad \sigma(t) > 0,$$

with endpoints

$$p_0 = \mathcal{N}(x; \mu(0), \sigma^2(0)I), \quad p_1 = \mathcal{N}(x; \mu(1), \sigma^2(1)I).$$

A simple flow map transporting  $p_0$  to  $p_t$  is

$$\psi_{0 \rightarrow t}(x) = \mu(t) + \sigma(t) \left( \frac{x - \mu(0)}{\sigma(0)} \right).$$

Indeed, if  $X_0 \sim p_0$ , then

$$\frac{X_0 - \mu(0)}{\sigma(0)} \sim \mathcal{N}(0, I),$$

and therefore

$$\psi_{0 \rightarrow t}(X_0) \sim \mathcal{N}(\mu(t), \sigma^2(t)I) = p_t.$$

# Prop 5.3.1: Closed-Form Velocity Field for Gaussian Density Path

## Proposition 5.3.1

Let

$$p_t(x) = \mathcal{N}(x; \mu(t), \sigma^2(t)I), \quad \sigma(t) > 0,$$

and consider the flow map

$$\psi_{0 \rightarrow t}(x) = \mu(t) + \sigma(t) \left( \frac{x - \mu(0)}{\sigma(0)} \right).$$

Then the velocity field generating this flow is

$$v_t(x) = \frac{\sigma'(t)}{\sigma(t)} (x - \mu(t)) + \mu'(t).$$

# Proof

The flow-generated trajectory satisfies

$$\frac{d}{dt} \Psi_{0 \rightarrow t}(y) = v_t(\Psi_{0 \rightarrow t}(y)).$$

Since  $\sigma(t) > 0$ , the map  $\Psi_{0 \rightarrow t}$  is invertible. Setting

$$x = \Psi_{0 \rightarrow t}(y)$$

gives

$$v_t(x) = \partial_t \Psi_{0 \rightarrow t}(\Psi_{0 \rightarrow t}^{-1}(x)).$$

Differentiating the flow map with respect to  $t$ ,

$$\partial_t \Psi_{0 \rightarrow t}(y) = \mu'(t) + \sigma'(t) \left( \frac{y - \mu(0)}{\sigma(0)} \right).$$

# Proof

From

$$x = \mu(t) + \sigma(t) \left( \frac{y - \mu(0)}{\sigma(0)} \right),$$

we obtain

$$\frac{y - \mu(0)}{\sigma(0)} = \frac{x - \mu(t)}{\sigma(t)}.$$

Substituting this identity yields

$$v_t(x) = \mu'(t) + \sigma'(t) \frac{x - \mu(t)}{\sigma(t)} = \frac{\sigma'(t)}{\sigma(t)} (x - \mu(t)) + \mu'(t)$$



# Conditional Probability-Path-First Construction

We first construct a conditional probability path

$$p_t(\cdot \mid z),$$

and then derive its corresponding conditional velocity field

$$v_t(\cdot \mid z).$$

The overall construction is

$$p_t(x \mid z) \implies v_t(x \mid z) \implies v_t(x) = \mathbb{E}[v_t(x \mid Z) \mid X_t = x].$$

Two common choices of the conditioning variable are:

- ① two-sided conditioning:

$$z = (x_0, x_1),$$

- ② one-sided conditioning:

$$z = x_0 \quad \text{or} \quad z = x_1.$$

# Conditional Probability-Path-First Construction



**Figure 5.8: Illustrations of two common types of conditioning probability paths.** It includes: (i) *two-sided*, conditioned on  $\mathbf{x}_0 \sim p_{\text{tgt}}$  and  $\mathbf{x}_1 \sim p_{\text{src}}$  with general endpoint distributions; (ii) *one-sided*, conditioned at either  $\mathbf{x}_0 \sim p_{\text{tgt}}$  or  $\mathbf{x}_1 \sim p_{\text{src}}$ .

*Source: Created by the authors.*

# Boundary Constraints

Let  $p_{\text{src}}$  and  $p_{\text{tgt}}$  denote the source and target distributions.  
A valid conditional construction should satisfy

$$p_{\text{src}}(x) = \int p_0(x | z) \pi(z) dz,$$

and

$$p_{\text{tgt}}(x) = \int p_1(x | z) \pi(z) dz.$$

Thus, after marginalizing the conditioning variable  $Z$ , the conditional paths must recover the prescribed endpoint distributions.

# Two-Sided Conditioning: $Z = (X_0, X_1)$

Let

$$X_0 \sim p_{\text{src}}, \quad X_1 \sim p_{\text{tgt}},$$

independently. Then

$$\pi(x_0, x_1) = p_{\text{src}}(x_0)p_{\text{tgt}}(x_1).$$

Define the conditional Gaussian path

$$p_t(x \mid x_0, x_1) = \mathcal{N}(x; a_t x_0 + b_t x_1, \sigma^2 I),$$

where

$$a_0 = 1, \quad b_0 = 0, \quad a_1 = 0, \quad b_1 = 1.$$

Its conditional mean interpolates between the two endpoints:

$$m_t(x_0, x_1) = a_t x_0 + b_t x_1.$$

# Two-Sided Conditioning: $Z = (X_0, X_1)$

In the deterministic limit  $\sigma \rightarrow 0$ ,

$$p_t(x \mid x_0, x_1) = \delta(x - a_t x_0 - b_t x_1).$$

A commonly used linear schedule is

$$a_t = 1 - t, \quad b_t = t,$$

which gives

$$X_t = (1 - t)X_0 + tX_1.$$

# Two-Sided Conditioning: $Z = (X_0, X_1)$

By Proposition 5.3.1, the conditional velocity field is

$$v_t(x \mid x_0, x_1) = a'_t x_0 + b'_t x_1.$$

For the deterministic path

$$X_t = a_t X_0 + b_t X_1,$$

the conditional flow matching objective is

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, X_0, X_1} \left[ \|v_\phi(X_t, t) - (a'_t X_0 + b'_t X_1)\|_2^2 \right].$$

By MSE optimality, the minimizer is

$$v^*(x, t) = \mathbb{E} [a'_t X_0 + b'_t X_1 \mid X_t = x] = \mathbb{E} [\dot{X}_t \mid X_t = x].$$

# One-Sided Conditioning: $Z = X_1$

Consider the standard generative setting

$$p_{\text{src}} = \mathcal{N}(0, I), \quad p_{\text{tgt}} = p_{\text{data}}.$$

Choose

$$Z = X_1, \quad X_1 \sim p_{\text{data}}.$$

For a fixed data point  $x_1$ , define

$$p_t(x \mid x_1) = \mathcal{N}(x; b_t x_1, a_t^2 I),$$

where

$$a_0 = 1, \quad b_0 = 0, \quad a_1 = 0, \quad b_1 = 1.$$

# One-Sided Conditioning: $Z = X_1$

At  $t = 0$ ,

$$p_0(x \mid x_1) = \mathcal{N}(x; 0, I),$$

which is independent of  $x_1$ .

At  $t = 1$ , interpreted as the limit  $a_t \rightarrow 0$ ,

$$p_1(x \mid x_1) = \delta(x - x_1).$$

Therefore,

$$p_0(x) = \int p_0(x \mid x_1) p_{\text{data}}(x_1) dx_1 = \mathcal{N}(x; 0, I),$$

while

$$p_1(x) = \int \delta(x - x_1) p_{\text{data}}(x_1) dx_1 = p_{\text{data}}(x).$$

# One-Sided Conditioning: $Z = X_1$

By Proposition 5.3.1,

$$v_t(x | x_1) = \dot{b}_t x_1 + \frac{\dot{a}_t}{a_t} (x - b_t x_1).$$

This holds for  $t \in (0, 1)$  with  $a_t > 0$ .

Sample

$$t \sim \mathcal{U}(0, 1), \quad X_1 \sim p_{\text{data}},$$

and

$$X_t \sim p_t(\cdot | X_1).$$

The one-sided CFM objective is

$$\mathcal{L}_{\text{CFM}} = \mathbb{E} \left[ \left\| v_\phi(X_t, t) - \left( \dot{b}_t X_1 + \frac{\dot{a}_t}{a_t} (X_t - b_t X_1) \right) \right\|_2^2 \right].$$

Its MSE-optimal solution is

$$v^*(x, t) = \mathbb{E} [v_t(X_t | X_1) | X_t = x].$$

# Equivalence of One-Sided and Two-Sided Targets

Since

$$X_t = a_t X_0 + b_t X_1,$$

we have

$$X_t - b_t X_1 = a_t X_0.$$

Substituting this into the one-sided velocity gives

$$\begin{aligned} v_t(X_t | X_1) &= b'_t X_1 + \frac{a'_t}{a_t} (X_t - b_t X_1) \\ &= b'_t X_1 + \frac{a'_t}{a_t} (a_t X_0) \\ &= a'_t X_0 + b'_t X_1. \end{aligned}$$

Hence, the two-sided and one-sided CFM objectives induce the same marginal velocity field.

# Gaussian Flow Matching and Diffusion Models

Both methods use a Gaussian interpolation:

$$X_t = a_t X_0 + b_t X_1, \quad X_0 \sim \mathcal{N}(0, I), \quad X_1 \sim p_{\text{data}}.$$

Forward diffusion is written as

$$X_s = \alpha_s X_{\text{data}} + \sigma_s \epsilon.$$

With reversed time  $t = 1 - s$ ,

$$a_t = \sigma_{1-t}, \quad b_t = \alpha_{1-t}.$$

Thus, Gaussian flow matching and diffusion describe the same type of path under opposite time conventions.

# Common Gaussian Interpolants

| Method              | Prior coefficient $a_t$            | Data coefficient $b_t$             |
|---------------------|------------------------------------|------------------------------------|
| VE diffusion        | $\sigma_t$                         | 1                                  |
| VP diffusion        | $\sqrt{1 - b_t^2}$                 | $b_t$                              |
| FM / Rectified Flow | $1 - t$                            | $t$                                |
| Trigonometric       | $\cos\left(\frac{\pi t}{2}\right)$ | $\sin\left(\frac{\pi t}{2}\right)$ |

For FM/RF and the trigonometric interpolant,

$t = 0$  denotes the prior,  $t = 1$  denotes the data.

The VE and VP schedules are often presented using the opposite diffusion-time convention.

# Conditional Flow-First Construction

Instead of first specifying a conditional density path, we directly construct a conditional flow map

$$\psi_{0 \rightarrow t}(\cdot; z).$$

The construction is

$$\psi_{0 \rightarrow t}(\cdot; z) \implies p_t(\cdot | z) \quad \text{and} \quad v_t(\cdot | z).$$

For

$$X_0 \sim p_{\text{src}}, \quad Z \sim \pi,$$

we define the trajectory

$$X_t = \psi_{0 \rightarrow t}(X_0; Z).$$

The conditional path is the distribution of  $X_t$ , while its time derivative provides a closed-form CFM regression target.

# Conditional Flow-First Construction

Consider the conditional affine flow

$$\psi_{0 \rightarrow t}(x_0; z) = \mu_t(z) + A_t(z)x_0,$$

where

$$\mu_t(z) \in \mathbb{R}^D, \quad A_t(z) \in \mathbb{R}^{D \times D}.$$

We assume that  $A_t(z)$  is invertible for  $t \in (0, 1)$ , with

$$A_0(z) = I, \quad \mu_0(z) = 0.$$

Hence,

$$\psi_{0 \rightarrow 0}(x_0; z) = x_0,$$

so the flow begins at the source distribution.

# Conditional Flow-First Construction

The conditional flow induces the probability path

$$p_t(\cdot | z) = (\Psi_{0 \rightarrow t}(\cdot; z))_{\#} p_{\text{src}},$$

and the corresponding marginal path is

$$p_t(x) = \int p_t(x | z) \pi(z) dz.$$

Sampling does not require evaluating the density:

$$Z \sim \pi, \quad X_0 \sim p_{\text{src}},$$

followed by

$$X_t = \mu_t(Z) + A_t(Z)X_0$$

# Conditional Flow-First Construction

For fixed  $z$ ,

$$X_t = \mu_t(z) + A_t(z)X_0.$$

If

$$X_0 \sim \mathcal{N}(m_0, \Sigma_0),$$

then

$$X_t \mid z \sim \mathcal{N}(\mu_t(z) + A_t(z)m_0, A_t(z)\Sigma_0A_t(z)^\top).$$

Therefore, an affine flow produces a Gaussian conditional path when the source distribution is Gaussian.

For a general non-Gaussian source distribution,

$$p_t(\cdot \mid z)$$

is also generally non-Gaussian.

# Conditional Flow-First Construction

The flow map satisfies

$$\frac{d}{dt} \Psi_{0 \rightarrow t}(y; z) = v_t(\Psi_{0 \rightarrow t}(y; z) \mid z).$$

Let

$$x = \Psi_{0 \rightarrow t}(y; z).$$

Since the map is invertible for  $t \in (0, 1)$ ,

$$y = \Psi_{0 \rightarrow t}^{-1}(x; z) = \Psi_{t \rightarrow 0}(x; z).$$

Hence, the conditional velocity field is

$$v_t(x \mid z) = \frac{d}{dt} \Psi_{0 \rightarrow t}(\Psi_{t \rightarrow 0}(x; z); z)$$

# Conditional Flow-First Construction

For

$$x = \mu_t(z) + A_t(z)x_0,$$

the initial state is

$$x_0 = A_t(z)^{-1} (x - \mu_t(z)).$$

Differentiating the trajectory gives

$$\dot{X}_t = \mu'_t(z) + A'_t(z)X_0.$$

Substituting the inverse map yields

$$v_t(x | z) = \mu'_t(z) + A'_t(z)A_t(z)^{-1} (x - \mu_t(z))$$

On sampled trajectories, the equivalent regression target is

$$v_t(X_t | Z) = \mu'_t(Z) + A'_t(Z)X_0$$

# One-Sided Conditioning: $Z = X_1$

Choose

$$\mu_t(X_1) = b_t X_1, \quad A_t(X_1) = a_t I,$$

with

$$a_0 = 1, \quad b_0 = 0, \quad a_1 = 0, \quad b_1 = 1.$$

The resulting trajectory is

$$X_t = a_t X_0 + b_t X_1.$$

The conditional velocity is

$$v_t(x | X_1) = b'_t X_1 + \frac{a'_t}{a_t} (x - b_t X_1)$$

Since

$$X_t - b_t X_1 = a_t X_0,$$

substituting into the conditional velocity gives

$$\begin{aligned} v_t(X_t | X_1) &= b'_t X_1 + \frac{a'_t}{a_t} (X_t - b_t X_1) \\ &= b'_t X_1 + a'_t X_0. \end{aligned}$$

# Two-Sided Conditioning: $Z = (X_0, X_1)$

Conditioning on both endpoints gives the deterministic trajectory

$$X_t = a_t X_0 + b_t X_1.$$

The corresponding conditional path is

$$p_t(\cdot \mid X_0, X_1) = \delta_{a_t X_0 + b_t X_1}.$$

Differentiating the trajectory yields

$$v_t(X_t \mid X_0, X_1) = a'_t X_0 + b'_t X_1.$$

Thus, the one-sided and two-sided constructions lead to the same sample-level CFM regression target.

# Unconditional Gaussian Path as a Special Case

Let

$$X_0 \sim \mathcal{N}(\mu(0), \sigma^2(0)I),$$

and define

$$\psi_{0 \rightarrow t}(x_0) = \mu(t) + \frac{\sigma(t)}{\sigma(0)}(x_0 - \mu(0)).$$

Then

$$X_t \sim \mathcal{N}(\mu(t), \sigma^2(t)I).$$

The induced velocity field is

$$v_t(x) = \mu'(t) + \frac{\sigma'(t)}{\sigma(t)}(x - \mu(t)).$$

# Probability-Path-First vs. Flow-First

| Aspect      | Probability-Path-First                                     | Flow-First                                                                                      |
|-------------|------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Given       | Conditional density path                                   | Conditional flow map                                                                            |
| Velocity    | Solve the continuity equation                              | Differentiate trajectories                                                                      |
| Closed form | Easy for Gaussian or other structured densities            | Easy for structured maps such as affine or geometric flows                                      |
| Uniqueness  | Generally non-unique without an additional selection rule  | Determined by the prescribed trajectories; unique as a spatial field when the map is invertible |
| Validity    | Must verify that $v_t$ generates the intended density path | Holds automatically through $p_t(\cdot   z) = (\Psi_{0 \rightarrow t}(\cdot; z))_{\#} p_0$      |

# Conditional Straightness vs. Marginal Straightness

Under the canonical affine interpolation,

$$X_t = (1 - t)X_0 + tX_1,$$

each conditional trajectory is a straight line for a fixed pair  $(X_0, X_1)$ , since

$$\frac{dX_t}{dt} = X_1 - X_0.$$

However, sampling is governed by the marginal velocity field

$$v^*(x, t) = \mathbb{E}[X_1 - X_0 \mid X_t = x].$$

This averages the velocities of all endpoint pairs that may produce the same intermediate state  $x$ . Since this conditional distribution changes with  $t$ , the marginal velocity generally changes along the trajectory.

Therefore,

straight conditional paths  $\nrightarrow$  straight marginal ODE trajectories.

# One-Dimensional Gaussian Counterexample

Let

$$X_0 \sim \mathcal{N}(0, \sigma_0^2), \quad X_1 \sim \mathcal{N}(0, \sigma_1^2),$$

independently, and define

$$X_t = (1 - t)X_0 + tX_1.$$

The marginal velocity is

$$v^*(x, t) = \mathbb{E}[X_1 - X_0 \mid X_t = x].$$

Since  $(X_1 - X_0, X_t)$  is jointly Gaussian,

$$v^*(x, t) = \frac{\text{Cov}(X_1 - X_0, X_t)}{\text{Var}(X_t)} x.$$

Define

$$V(t) = \text{Var}(X_t) = (1 - t)^2 \sigma_0^2 + t^2 \sigma_1^2.$$

# One-Dimensional Gaussian Counterexample

A direct calculation gives

$$\text{Cov}(X_1 - X_0, X_t) = t\sigma_1^2 - (1 - t)\sigma_0^2.$$

Since

$$V'(t) = 2t\sigma_1^2 - 2(1 - t)\sigma_0^2,$$

we have

$$\text{Cov}(X_1 - X_0, X_t) = \frac{1}{2}V'(t).$$

Hence,

$$v^*(x, t) = \frac{V'(t)}{2V(t)}x.$$

Although every conditional target  $X_1 - X_0$  is constant in time, the marginal velocity depends on both the current position and time.

# One-Dimensional Gaussian Counterexample

The marginal sampling ODE is

$$\frac{dZ_t}{dt} = \frac{V'(t)}{2V(t)} Z_t.$$

Its exact solution is

$$Z_t = Z_0 \sqrt{\frac{V(t)}{V(0)}}.$$

Since  $V(0) = \sigma_0^2$ ,

$$Z_t = Z_0 \sqrt{(1-t)^2 + \frac{\sigma_1^2}{\sigma_0^2} t^2}.$$

This expression is generally not affine in  $t$ . Therefore,

$$Z_t \neq (1-t)Z_0 + tZ_1,$$

and the marginal ODE trajectory is not a straight line.

# One-Dimensional Gaussian Counterexample

Suppose

$$\sigma_0 = \sigma_1.$$

Then the source and target distributions are identical, and

$$Z_1 = Z_0.$$

A straight-line trajectory between identical endpoints would be

$$Z_t = Z_0.$$

However, the actual solution is

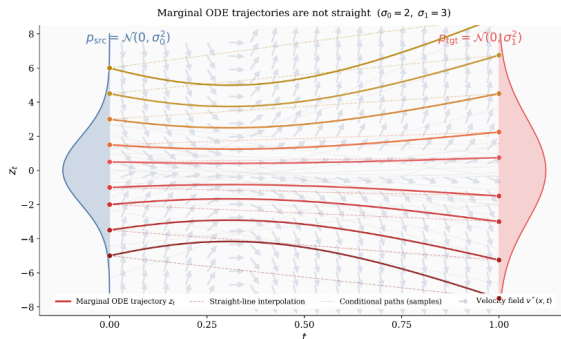
$$Z_t = Z_0 \sqrt{(1-t)^2 + t^2}.$$

At  $t = \frac{1}{2}$ ,

$$Z_{1/2} = \frac{Z_0}{\sqrt{2}} \neq Z_0.$$

Thus, the trajectory contracts toward the origin and expands again, despite all conditional interpolation paths being straight.

# One-Dimensional Gaussian Counterexample



**Figure 5.9: Marginal ODE trajectories are not straight under the canonical affine interpolation.** We take  $x_0 \sim \mathcal{N}(0, \sigma_0^2)$  and  $x_1 \sim \mathcal{N}(0, \sigma_1^2)$  with  $\sigma_0 = 2$ ,  $\sigma_1 = 3$ , and independent coupling. Gray lines show sampled conditional paths  $(1-t)x_0 + tx_1$ , which are straight by construction. Solid curves show the exact marginal ODE trajectories  $z_t = z_0 \sqrt{(1-t)^2 + (\sigma_1/\sigma_0)^2 t^2}$ , and dashed lines show the corresponding straight-line interpolations from  $z_0$  to  $z_1$ . The visible gap between solid and dashed curves confirms that the marginal ODE trajectories are not affine in  $t$ , even though every conditional path is a straight line. The marginal densities  $p_{\text{src}}$  and  $p_{\text{tgt}}$  are shown on the left and right margins, respectively.

Source: Created by the authors with AI-assisted coding.

# Canonical Affine Flow

Given

$$p_0 = p_{\text{src}}, \quad p_1 = p_{\text{tgt}},$$

the canonical affine interpolation is

$$X_t = (1 - t)X_0 + tX_1, \quad (X_0, X_1) \sim \pi,$$

where  $\pi$  is a coupling of  $p_{\text{src}}$  and  $p_{\text{tgt}}$ .

Since

$$\dot{X}_t = X_1 - X_0,$$

the flow matching objective becomes

$$\mathcal{L}(\phi \mid \pi) = \mathbb{E}_{t, (X_0, X_1) \sim \pi} \left[ \|v_\phi(X_t, t) - (X_1 - X_0)\|_2^2 \right].$$

A common initialization is the independent coupling

$$\pi^{(0)} = p_{\text{src}} \otimes p_{\text{tgt}}.$$

# Why Rectify the Coupling?

Under the independent coupling,

$$X_0 \sim p_{\text{src}}, \quad X_1 \sim p_{\text{tgt}}$$

are paired without considering geometric compatibility.

- Nearby intermediate states may have highly variable velocity targets.
- The learned velocity field can be difficult to approximate.
- Numerical integration may require many solver steps.

Rectification replaces these arbitrary pairings with endpoint pairs induced by a pretrained flow.

# Rectifying Flows

Let  $v_{\phi^*}$  be a pretrained velocity field and define

$$\frac{dZ_t}{dt} = v_{\phi^*}(Z_t, t), \quad Z_0 \sim p_{\text{src}}.$$

If  $\Phi_{0 \rightarrow t}$  denotes the corresponding flow map, then

$$Z_t = \Phi_{0 \rightarrow t}(Z_0), \quad \hat{Z}_1 = \Phi_{0 \rightarrow 1}(Z_0).$$

The rectified coupling is

$$\pi_{\text{rect}}(z_0, z_1) = p_{\text{src}}(z_0) \delta(z_1 - \Phi_{0 \rightarrow 1}(z_0)).$$

Thus, each source sample is paired with the endpoint produced by the current flow.

# Rectifying Flows

The source marginal is preserved:

$$\int \pi_{\text{rect}}(z_0, z_1) dz_1 = p_{\text{src}}(z_0).$$

The endpoint marginal is

$$(\Phi_{0 \rightarrow 1})_{\#} p_{\text{src}}.$$

If the pretrained flow is exact, then

$$(\Phi_{0 \rightarrow 1})_{\#} p_{\text{src}} = p_{\text{tgt}}.$$

For an approximate learned flow, the endpoint marginal is the model-generated distribution and only approximately matches  $p_{\text{tgt}}$ .

# Rectifying Flows

Given the dependent pair

$$(Z_0, \widehat{Z}_1),$$

construct a new canonical interpolation

$$\widetilde{X}_t = (1 - t)Z_0 + t\widehat{Z}_1.$$

Its velocity target is

$$\frac{d\widetilde{X}_t}{dt} = \widehat{Z}_1 - Z_0.$$

The rectified training objective is

$$\mathcal{L}_{\text{rect}}(\phi) = \mathbb{E}_{t, Z_0} \left[ \left\| v_{\phi}(\widetilde{X}_t, t) - (\widehat{Z}_1 - Z_0) \right\|_2^2 \right].$$

The endpoint assignment is retained, while the new model is trained using the direct chord between the endpoints.

# Reflow

Reflow repeatedly applies the rectification procedure:

$$\pi^{(k+1)} = \text{Rectify}\left(\pi^{(k)}\right), \quad \pi^{(0)} = p_{\text{src}} \otimes p_{\text{tgt}}.$$

At iteration  $k$ , train

$$\phi_{k+1} \in \arg \min_{\phi} \mathcal{L}(\phi \mid \pi^{(k)}),$$

where

$$Z_t^{(k)} = (1 - t)Z_0^{(k)} + tZ_1^{(k)}.$$

The corresponding objective is

$$\mathcal{L}(\phi \mid \pi^{(k)}) = \mathbb{E} \left[ \left\| v_{\phi}(Z_t^{(k)}, t) - (Z_1^{(k)} - Z_0^{(k)}) \right\|_2^2 \right].$$

# Reflow

After training  $v_{\phi_{k+1}}$ , sample

$$Z_0^{(k+1)} \sim p_{\text{src}}$$

and solve

$$\frac{dZ_t}{dt} = v_{\phi_{k+1}}(Z_t, t), \quad Z_{t=0} = Z_0^{(k+1)}.$$

The endpoint is

$$Z_1^{(k+1)} = Z_0^{(k+1)} + \int_0^1 v_{\phi_{k+1}}(Z_t, t) dt.$$

This defines the updated coupling

$$\pi^{(k+1)}(z_0, z_1) = p_{\text{src}}(z_0) \delta\left(z_1 - \Phi_{0 \rightarrow 1}^{(k+1)}(z_0)\right).$$

# Reflow

Each Reflow iteration updates both

the velocity field and the endpoint coupling.

The updated pairings are increasingly aligned with the transport induced by the learned model.

- Reduced trajectory curvature
- Improved numerical stability and alignment

# Three Properties of Reflow

Reflow has three important properties.

- 1 It does not increase the transport cost.
- 2 It progressively straightens the stochastic path.
- 3 Straight paths are not necessarily optimal transport paths.

# I. Reflow Never Increases Transport Cost

Let  $c(y)$  be a convex cost function, for example

$$c(y) = \|y\|_2^p, \quad p \geq 1.$$

Given an initial coupling

$$(X_0, X_1) \sim \pi,$$

Rectify constructs a new pair

$$(Z_0, \hat{Z}_1), \quad \hat{Z}_1 = \Phi_{0 \rightarrow 1}(Z_0).$$

## Proposition 5.4.1: Rectify May Reduce Transport Costs

Assume that the learned velocity field is ideal:

$$v_{\phi^*} = v^*.$$

Then

$$\mathbb{E} [c(\hat{Z}_1 - Z_0)] \leq \mathbb{E} [c(X_1 - X_0)].$$

Thus, each Rectify step does not increase the expected transport cost.

# I. Reflow Never Increases Transport Cost

The ideal velocity field is the conditional average

$$v^*(x, t) = \mathbb{E}[X_1 - X_0 \mid X_t = x].$$

Since  $c$  is convex, Jensen's inequality gives

$$c(v^*(X_t, t)) = c(\mathbb{E}[X_1 - X_0 \mid X_t]) \leq \mathbb{E}[c(X_1 - X_0) \mid X_t].$$

Therefore, replacing individual displacements by their conditional average cannot increase the convex cost.

Repeated application yields

$$\mathcal{C}(\pi^{(k+1)}) \leq \mathcal{C}(\pi^{(k)}),$$

where

$$\mathcal{C}(\pi) = \mathbb{E}_{(X_0, X_1) \sim \pi} [c(X_1 - X_0)].$$

## II. Reflow Straightens the Path

For a stochastic path

$$Y = \{Y_t\}_{t \in [0,1]},$$

define the straightness functional

$$S(Y) = \int_0^1 \mathbb{E} \left[ \left\| Y_1 - Y_0 - \frac{dY_t}{dt} \right\|_2^2 \right] dt.$$

Here,

$$Y_1 - Y_0$$

is the constant velocity of the straight chord, whereas

$$\frac{dY_t}{dt}$$

is the actual instantaneous velocity.

Thus,  $S(Y)$  measures the discrepancy between the actual path velocity and the velocity of the direct straight-line path.

## II. Reflow Straightens the Path

If

$$S(Y) = 0,$$

then

$$\frac{dY_t}{dt} = Y_1 - Y_0 \quad \text{for almost every } t.$$

Integrating over time gives

$$Y_t = Y_0 + t(Y_1 - Y_0) = (1 - t)Y_0 + tY_1.$$

### Proposition 5.4.2: Reflow Straightens the Stochastic Path

For the sequence of rectified paths

$$Z^{(0)}, Z^{(1)}, \dots, Z^{(K)},$$

we have

$$\min_{k \in \{0, \dots, K\}} S(Z^{(k)}) \leq \frac{\mathbb{E} [\|X_1 - X_0\|_2^2]}{K}.$$

Hence, among the first  $K$  Reflow iterations, at least one path has straightness

## II. Interpretation of Proposition 5.4.2

The proposition does not necessarily imply

$$S\left(Z^{(k+1)}\right) \leq S\left(Z^{(k)}\right) \quad \text{for every } k.$$

Instead, it guarantees

$$\min_{0 \leq k \leq K} S\left(Z^{(k)}\right) \rightarrow 0$$

at least at the rate

$$O(K^{-1}).$$

Therefore, as more Rectify steps are performed, the sequence must contain paths that become progressively closer to straight-line transport.

### III. Connection to Optimal Transport

In quadratic-cost optimal transport, particles move along straight lines:

$$X_t = (1 - t)X_0 + tT(X_0),$$

where  $T$  is the optimal transport map.

However, the converse is not true.

A map  $S$  may also generate straight-line paths

$$X_t = (1 - t)X_0 + tS(X_0),$$

but it is optimal only if it minimizes the Monge cost

$$\mathbb{E} [\|X_0 - S(X_0)\|_2^2].$$

#### Straightness Is Not Sufficient for Optimality

A straight trajectory only specifies how a particle moves between two assigned endpoints. Optimal transport additionally requires that the endpoint map itself minimize the transport cost.

### III. Straight Paths Need Not Be Optimal

Let

$$p_{\text{src}} = p_{\text{tgt}} = \mathcal{N}(0, I).$$

The identity map

$$T(x) = x$$

is optimal and has zero cost:

$$\mathbb{E} [\|X - T(X)\|_2^p] = 0.$$

Now consider a nontrivial rotation matrix  $A$  satisfying

$$A^\top A = I, \quad \det(A) = 1, \quad A \neq I.$$

Because

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

the map

$$S(x) = Ax$$

defines a valid coupling of the same source and target distributions.

### III. Rotation Coupling Example

The interpolation

$$X_t = (1 - t)X + tAX$$

is a straight-line path for every  $X$ .

However, its cost is

$$\mathbb{E} [\|AX - X\|_2^p] > 0,$$

whereas the identity coupling has cost zero.

Therefore,

straight path  $\nrightarrow$  optimal transport.

Optimal transport requires both

straight interpolation

and

a cost-minimizing endpoint map.

### III. Connection to Optimal Transport

Straightness concerns the shape of each path:

$$X_t = (1 - t)X_0 + tX_1.$$

Optimal transport concerns the endpoint assignment:

$$X_0 \mapsto X_1.$$

Thus,

OT optimality = straight path + optimal endpoint map.

Reflow encourages straightness, but it does not necessarily recover the OT map.

# Example: Rotation Coupling

Let

$$p_{\text{src}} = p_{\text{tgt}} = \mathcal{N}(0, I).$$

The optimal coupling for

$$c(x, y) = \|x - y\|^p$$

is the identity coupling:

$$Y = X.$$

Now let

$$Y = AX,$$

where  $A$  is a nontrivial rotation matrix:

$$A^\top A = I, \quad A \neq I.$$

Since  $AX \sim \mathcal{N}(0, I)$ , this is a valid coupling.

# Example: Rotation Coupling

The path

$$X_t = (1 - t)X + tAX$$

is still a straight line between  $X$  and  $AX$ .

But the transport cost is

$$\mathbb{E} [\|AX - X\|^p] > 0.$$

The identity coupling has cost

$$\mathbb{E} [\|X - X\|^p] = 0.$$

Therefore, the rotation coupling is straight but not optimal.

Straight coupling  $\nRightarrow$  optimal coupling.

# Q1. Coupling Strategies

| Method                 | Main Idea                                                             | Characteristics                                                 |
|------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------|
| Multisample FM         | Match multiple source and target samples jointly within a minibatch   | Can reduce gradient variance and transport cost                 |
| OT-CFM                 | Use an optimal transport plan to pair nearby samples                  | Encourages shorter and simpler transport paths                  |
| Model-Aligned Coupling | Select pairs aligned with the velocity predicted by the current model | Considers model compatibility in addition to geometric distance |

## Q2. Recent Applications of Flow Matching

| Domain                | Representative Work     | Main Idea                                                                                               |
|-----------------------|-------------------------|---------------------------------------------------------------------------------------------------------|
| Image generation      | Latent Flow Matching    | Train Flow Matching in a pre-trained autoencoder latent space to reduce high-resolution generation cost |
| Video generation      | Pyramidal Flow Matching | Generate video progressively across temporal and spatial resolutions                                    |
| Speech generation     | SpeechFlow              | Perform large-scale generative pre-training on unlabeled speech                                         |
| Biological structures | AlphaFlow, HarmonicFlow | Apply geometry-aware Flow Matching to protein ensembles and molecular structures                        |

## Q3. Non-Gaussian Source Distributions

- **Gaussian process priors**

Temporal correlations can be incorporated directly into the source:

$$X_0(\tau) \sim \mathcal{GP}(0, k(\tau, \tau')).$$

- **von Mises–Fisher distributions**

These are appropriate for directional or spherical data:

$$p(x) \propto \exp(\kappa \mu^\top x), \quad \|x\|_2 = 1.$$

## Q4. Why Does CFM Select a Unique Velocity Field?

CFM does not solve the continuity equation alone.  
It minimizes the regression objective

$$\mathcal{L}_{\text{CFM}}(v) = \mathbb{E} \left[ \|v(X_t, t) - u_t(X_t \mid Z)\|_2^2 \right].$$

The unique population minimizer is

$$v^*(x, t) = \mathbb{E} [u_t(x \mid Z) \mid X_t = x].$$

Thus, although many velocity fields may generate the same density path, the least-squares regression problem selects one velocity field almost everywhere under  $p_t$ .

## Q5. Is Reflow Still Simulation-Free?

The term *simulation-free* refers specifically to velocity-field regression. Given a fixed coupling  $(X_0, X_1)$ , Reflow trains with

$$X_t = (1 - t)X_0 + tX_1, \quad u_t = X_1 - X_0,$$

and minimizes

$$\mathcal{L}(v) = \mathbb{E} [\|v(X_t, t) - u_t\|_2^2].$$

Thus, the regression step itself does not require an ODE solver.

However, to construct a new coupling for the next Reflow round, the previous model must be integrated:

$$\frac{dZ_t}{dt} = v_{\theta_k}(Z_t, t), \quad \hat{Z}_1 = \Phi_{0 \rightarrow 1}^{\theta_k}(Z_0).$$

Therefore, Reflow regression is simulation-free, but coupling generation requires ODE integration.

## Q6. Why Does Standard NF Require Equal Dimensions?

A standard Normalizing Flow assumes

$$f : \mathbb{R}^d \rightarrow \mathbb{R}^d$$

is invertible, so that its Jacobian is square and

$$p_X(x) = p_Z(f^{-1}(x)) |\det Df^{-1}(x)|$$

is well defined.

If the input and output dimensions differ, the Jacobian is rectangular, so the ordinary determinant and the standard change-of-variables formula cannot be used.

## Q6. Why Does Standard NF Require Equal Dimensions?

For a conditional affine flow,

$$X_t = a_t X_0 + b_t Z,$$

the ambient dimension remains unchanged. For  $t < 1$ , the conditional map is invertible when  $a_t \neq 0$ . However, as  $t \rightarrow 1$ , the map may become singular and the conditional distribution may collapse to a point mass.

### Key Point

Flow Matching does not apply the standard change-of-variables formula to this singular endpoint. Therefore, endpoint invertibility is not required. The construction is best viewed as a singular endpoint limit of invertible maps, rather than as a relaxation of the equal-dimension condition.

# References I



Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M.

*Flow Matching for Generative Modeling.*

arXiv:2210.02747, 2022.



Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., and Bengio, Y.

*Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport.*

Transactions on Machine Learning Research, 2024.



Liu, X., Gong, C., and Liu, Q.

*Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow.*

arXiv:2209.03003, 2022.



Pooladian, A.-A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., and Chen, R. T. Q.

*Multisample Flow Matching: Straightening Flows with Minibatch Couplings.*

International Conference on Machine Learning, 2023.

# References II



Xing, S., Cao, J., Huang, H., Zhang, X.-Y., and He, R.  
*Exploring Straighter Trajectories of Flow Matching with Diffusion Guidance.*  
[arXiv:2311.16507](#), 2023.



Lin, Y., Yao, Y., and Liu, T.  
*Beyond Optimal Transport: Model-Aligned Coupling for Flow Matching.*  
[arXiv preprint](#), 2025.



Dao, Q., Phung, H., Nguyen, B., and Tran, A.  
*Flow Matching in Latent Space.*  
[arXiv:2307.08698](#), 2023.



Jin, Y. et al.  
*Pyramidal Flow Matching for Efficient Video Generative Modeling.*  
[arXiv:2410.05954](#), 2024.



Liu, A. H., Le, M., Vyas, A., Shi, B., Tjandra, A., and Hsu, W.-N.  
*Generative Pre-Training for Speech with Flow Matching.*  
[International Conference on Learning Representations](#), 2024.

# References III



Xu, X. et al.

*UniFlow-Audio: Unified Flow Matching for Audio Generation from Omni-Modalities.*  
arXiv preprint, 2025.



Kollovieh, M., Lienen, M., Luedke, D., Schwinn, L., and Guennemann, S.

*Flow Matching with Gaussian Process Priors for Probabilistic Time Series Forecasting.*  
International Conference on Learning Representations, 2025.



Stark, H., Jing, B., Barzilay, R., and Jaakkola, T.

*Harmonic Self-Conditioned Flow Matching for Multi-Ligand Docking and Binding Site Design.*  
arXiv:2310.05764, 2023.



Jing, B., Berger, B., and Jaakkola, T.

*AlphaFold Meets Flow Matching for Generating Protein Ensembles.*  
International Conference on Machine Learning, 2024.