Part I

Flow Matching

Table of contents

Motivation and Background

Origin and intuition

- Flow matching combines ideas from *continuous normalizing flows* (CNFs), *diffusion/score-based models*, and *optimal transport* (OT).
- It reuses *ODE-based generation* from CNFs, *noise-to-data paths* idea from diffusion models, and *mass-transport geometry* from OT.
- Its main motivation is to train continuous-time generative models *without simulating the ODE during training*.
- It learns a time-dependent *velocity field* transporting Gaussian noise to the data distribution.
- Training becomes *supervised regression* on known velocities along simple interpolating paths.

Main papers

- The area builds on *Neural ODEs* Chen et al. (2018), *score-based SDEs and probability-flow ODEs* Song et al. (2021), and optimal-transport ideas.
- It was formalized through *Flow Matching* Lipman et al. (2023), *Rectified Flow* Liu et al. (2022), and *Stochastic Interpolants* Albergo and Vanden-Eijnden (2023).

Continuous Flows and Transport

Continuous flows

- A continuous flow transforms samples by solving an *ODE*:

$$\frac{dX_t}{dt} = v_t(X_t), \qquad X_0 \sim p_0.$$

- The vector field v_t defines how each particle moves from a simple base distribution p_0 toward the data distribution p_1.
- Generation becomes deterministic: sample $X_0 \sim p_0$, integrate the ODE, and output X_1.

Transport viewpoint

- At the distribution level, the density evolves according to the *continuity equation*:

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

- This equation means that probability mass is *transported*, not created or destroyed.
- Flow matching learns the velocity field v_t that realizes this transport from noise to data.

Flow Matching Principle

We need a vector field v_t to define the ODE.

Choose a path, not directly a vector field

- Flow matching starts by choosing an *interpolation* between noise and data:

$$X_t = \psi_t(X_0, X_1), \qquad X_0 \sim p_0, \quad X_1 \sim p_{\mathrm{data}}.$$

- This choice defines both the intermediate distributions p_t and the target velocity

$$U_t = \partial_t \psi_t(X_0, X_1).$$

- Example: the linear interpolation $\quad X_t = (1-t)X_0 + tX_1, \qquad U_t = X_1 - X_0.$

The conditional trick

- The global velocity field $v_t(x)$ is unknown.
- But for each hidden pair $Z = (X_0, X_1)$, the conditional velocity U_t is known.
- Since the network only sees (t, X_t), not Z, it learns the *average velocity* of all paths crossing x:

$$v_t(x) = \mathbb{E}[U_t \mid X_t = x].$$

Conditional Flow Matching Objective

Marginal loss: ideal but inaccessible

- Ideally, one would regress on the true marginal velocity field:

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

- But $v_t(x)$ is unknown, so this loss cannot be used directly.

Conditional loss: tractable with the same gradient

- Use the known conditional velocity instead:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,X_0,X_1}\left[\|v_\theta(t, X_t) - U_t\|^2\right].$$

- The key identity is

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathcal{L}_{\mathrm{FM}}(\theta) + C, \qquad C \text{ independent of } \theta.$$

- Therefore,

$$\nabla_\theta \mathcal{L}_{\mathrm{CFM}}(\theta) = \nabla_\theta \mathcal{L}_{\mathrm{FM}}(\theta).$$

Training and Sampling Algorithms

Algorithmic view

- During training, the ODE is *not* solved: we only regress on known velocities along sampled conditional paths.

Algorithm 1 Conditional flow matching training

1: sample $X_1 \sim p_{\mathrm{data}}$, $X_0 \sim p_0$, $t \sim \mathcal{U}(0, 1)$
2: set $X_t = (1 - t)X_0 + tX_1$
3: set $U_t = X_1 - X_0$
4: minimize $\|v_\theta(t, X_t) - U_t\|^2$

- During sampling, the learned field is integrated as an ODE to transport noise into data.

Algorithm 2 Sampling with the learned flow

1: sample $X_0 \sim p_0$
2: solve $\frac{dX_t}{dt} = v_\theta(t, X_t)$ from $t = 0$ to $t = 1$
3: **return** generated sample X_1

References for this part I

- Albergo, M. S., & Vanden-Eijnden, E. (2023).**Building normalizing flows with stochastic interpolants.** *International Conference on Learning Representations*.
- Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018).**Neural ordinary differential equations.** *Advances in Neural Information Processing Systems*.
- Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., & Le, M. (2023).**Flow matching for generative modeling.** *International Conference on Learning Representations*.
- Liu, X., Gong, C., & Liu, Q. (2022).**Flow straight and fast: Learning to generate and transfer data with rectified flow.** *arXiv preprint arXiv:2209.03003*.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021).**Score-based generative modeling through stochastic differential equations.** *International Conference on Learning Representations*.