CS 395T: Foundations of Modern Generative AI Fall 2026

Lecture 18: Flow Matching and Consistency Models

October 26, 2026

Prof. Noah Golowich

In this lecture we build on the probability flow ODE discussed at the end of last lecture: first, we discuss flow matching, which learns the velocity field of an ODE sampler directly, and then we discuss consistency distillation, which learns the map from each point of an ODE trajectory to its endpoint.

1 Flow matching

The probability flow ODE from Lectures 16–17 generates data by following a velocity field computed from a score. Flow matching [1] instead learns the velocity field directly by regression.

1.1 Continuous normalizing flows

Let P_{data} be a distribution on $\mathbb{R}^d$ that we can access only through samples, and let p_{noise} be the density of $\mathcal{N}(0, I_d)$. In this section, time $t \in [0,1]$ runs *from noise to data*: $p_0 = p_{\text{noise}}$, and we want p_1 to equal or approximate P_{data}.

A *continuous normalizing flow* (CNF) parameterizes a time-dependent velocity by a neural network $v_\theta : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. Its flow map ϕ_t^θ is defined by

$$\frac{d}{dt}\phi_t^\theta(z) = v_\theta(t, \phi_t^\theta(z)), \qquad \phi_0^\theta(z) = z. \tag{1.1}$$

To generate a sample, draw $Z \sim \mathcal{N}(0, I_d)$ and output $\phi_1^\theta(Z)$, computed with an ODE solver. All of the randomness is in Z: once Z is drawn, the trajectory is deterministic.

For the moment, suppose v_θ is continuous in time, continuously differentiable and globally Lipschitz in space, uniformly over the time interval. Under these conditions the ODE has a unique flow, and each ϕ_t^θ is invertible; we compute the inverse by solving the ODE backward in time. Let p_t^θ be the density of $\phi_t^\theta(Z)$. The change-of-variables formula gives

$$p_t^\theta(x) = p_{\text{noise}}((\phi_t^\theta)^{-1}(x)) \left|\det D_x(\phi_t^\theta)^{-1}(x)\right|. \tag{1.2}$$

Here D_x denotes the Jacobian matrix. So the same flow gives both a way to sample and a formula for the model's density.

Which velocity generates a prescribed density path? Recall that the Fokker–Planck equation with zero noise coefficient (known as the *continuity equation*) tells us that a density p_t transported by a velocity $u_t : \mathbb{R}^d \to \mathbb{R}^d$ (namely, a particle X_t satisfies $dX_t = u_t(X_t)dt$, which we will write in this lecture as $\dot{X}_t = u_t(X_t)$) satisfies

$$\partial_t p_t(x) = -\nabla \cdot (p_t(x) u_t(x)). \tag{1.3}$$

How can we learn a vector field u_t for which the corresponding distribution p_t matches p_{data} (at say $t = 1$)? One natural approach is maximum likelihood. To compute the likelihood $p_t(y_t)$, suppose that

y_t satisfies $\dot{y}_t = u_t(y_t)$. The chain rule and (1.3) give

$$\frac{d}{dt}p_t(y_t) = \partial_t p_t(y_t) + \nabla p_t(y_t) \cdot u_t(y_t) = -p_t(y_t)\nabla \cdot u_t(y_t).$$

This scalar linear ODE determines $p_t(y_t)$ from $p_0(y_0)$, and every point at time t lies on a unique characteristic. In particular, where $p_t > 0$,

$$\log p_t(y_t) = \log p_0(y_0) - \int_0^t \nabla \cdot u_a(y_a)\, da. \tag{1.4}$$

With $u_t = v_\theta$, (1.4) computes the log-likelihood of a CNF. Training a CNF by maximum likelihood therefore involves integrating the flow and its divergence. We will next see how flow matching avoids this by regressing on explicit labels.

1.2 The flow matching objective and marginal velocity

Suppose we have chosen a target density path $(p_t)_{0\le t\le 1}$ and a velocity u_t satisfying (1.3). The *flow matching* objective is

$$\mathcal{L}_{\mathrm{FM}}(\theta) := \mathbb{E}_{t\sim \mathrm{Unif}(0,1),\, W_t\sim p_t} \|v_\theta(t, W_t) - u_t(W_t)\|^2. \tag{1.5}$$

If $v_\theta = u$, then by (1.3) the CNF generates the target densities. The problem is that we have not yet said how to choose p_t, how to compute u_t, or how to sample the regression pairs in (1.5).

Constructing the path one data point at a time. For each clean example x_1, choose a conditional density path $r_t(x \mid x_1)$ that begins at p_{noise} and ends concentrated near x_1. For example, we may require

$$r_0(x \mid x_1) = p_{\mathrm{noise}}(x), \qquad r_1(\cdot \mid x_1) = \mathcal{N}(x_1, \sigma_{\min}^2 I_d)$$

for a small $\sigma_{\min} > 0$. Average these paths over a clean sample $X_1 \sim P_{\mathrm{data}}$:

$$p_t(x) := \int r_t(x \mid x_1)\, P_{\mathrm{data}}(dx_1). \tag{1.6}$$

Then $p_0 = p_{\mathrm{noise}}$, and p_1 is the law of a data sample plus independent $\mathcal{N}(0, \sigma_{\min}^2 I_d)$ noise.

Suppose that for each x_1 we have a conditional velocity $u_t(x \mid x_1)$ that transports $r_t(\cdot \mid x_1)$. To combine these velocities, we should average them over the *posterior* distribution of X_1 given the current location, not over P_{data} itself.

Theorem 1.1 (Marginalizing conditional velocities; [1, Theorem 1])**.** *Suppose the conditional densities satisfy*

$$\partial_t r_t(x \mid x_1) = -\nabla \cdot (r_t(x \mid x_1) u_t(x \mid x_1)).$$

Define

$$u_t(x) := \frac{\int r_t(x \mid x_1) u_t(x \mid x_1)\, P_{\mathrm{data}}(dx_1)}{p_t(x)}. \tag{1.7}$$

Then p_t and u_t satisfy the continuity equation (1.3).

Proof. Differentiate (1.6) and substitute the conditional continuity equations:

$$\partial_t p_t(x) = \int \partial_t r_t(x \mid x_1)\, P_{\mathrm{data}}(dx_1)$$

$$
\begin{aligned}
&= -\int \nabla \cdot (r_t(x \mid x_1) u_t(x \mid x_1))\, P_{\text{data}}(dx_1) \\
&= -\nabla \cdot \left(\int r_t(x \mid x_1) u_t(x \mid x_1)\, P_{\text{data}}(dx_1) \right) \\
&= -\nabla \cdot (p_t(x) u_t(x)).
\end{aligned}
$$

□

For later use, sample $X_1 \sim P_{\text{data}}$ and then $W_t \sim r_t(\cdot \mid X_1)$. Bayes' rule rewrites (1.7) as

$$u_t(x) = \mathbb{E}[u_t(W_t \mid X_1) \mid W_t = x]. \tag{1.8}$$

Indeed, given $W_t = x$, the posterior measure of X_1 is $r_t(x \mid x_1) P_{\text{data}}(dx_1)/p_t(x)$. Each data point x_1 is weighted in proportion to $r_t(x \mid x_1)$, the density that its conditional path places at x.

A density path does not determine its velocity uniquely: if w_t satisfies $\nabla \cdot (p_t w_t) = 0$, then $u_t + w_t$ transports the same densities. Theorem 1.1 picks out one such velocity, which depends on our choice of conditional velocities.

1.3 Conditional flow matching and Gaussian paths

The integral in (1.7) involves the unknown posterior over data, so we cannot evaluate it. Instead, we train on the conditional velocities themselves. Define the *conditional flow matching* (CFM) loss by

$$\mathcal{L}_{\text{CFM}}(\theta) := \mathbb{E}_{\substack{t \sim \text{Unif}(0,1),\, X_1 \sim P_{\text{data}} \\ W_t \sim r_t(\cdot \mid X_1)}} \| v_\theta(t, W_t) - u_t(W_t \mid X_1) \|^2. \tag{1.9}$$

Here t and X_1 are drawn independently, and then W_t is drawn from $r_t(\cdot \mid X_1)$. The network sees only (t, W_t). The clean example X_1 enters only through the label.

Theorem 1.2 (Equivalence of FM and CFM; [1, Theorem 2])**.** *Let p_t and u_t be given by (1.6) and (1.7). Then*

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathcal{L}_{\text{FM}}(\theta) + C, \qquad C := \mathbb{E}\|u_t(W_t \mid X_1) - u_t(W_t)\|^2 \geq 0, \tag{1.10}$$

where C is independent of θ. Consequently, the two losses have the same minimizers and gradients.

Proof. Let $U = u_t(W_t \mid X_1)$ and $m = u_t(W_t)$. By (1.8), $\mathbb{E}[U \mid t, W_t] = m$. Write the regression error as

$$v_\theta(t, W_t) - U = (v_\theta(t, W_t) - m) - (U - m).$$

Its squared norm is the sum of the squared norms of the two terms minus twice their inner product. The expected inner product vanishes, because

$$\mathbb{E}[(v_\theta(t, W_t) - m) \cdot (U - m)] = \mathbb{E}[(v_\theta(t, W_t) - m) \cdot \mathbb{E}[U - m \mid t, W_t]] = 0.$$

Thus

$$\mathbb{E}\|v_\theta(t, W_t) - U\|^2 = \mathbb{E}\|v_\theta(t, W_t) - m\|^2 + \mathbb{E}\|U - m\|^2.$$

Given t, the marginal law of W_t is p_t, so the first term on the right is $\mathcal{L}_{\text{FM}}(\theta)$. The second term is C, which does not involve θ. This proves (1.10), and the statements about minimizers and gradients follow. □

Recall that denoising score matching in Lectures 16–17 was justified by the same conditional expectation argument.

Gaussian conditional paths. We now choose conditional paths for which we can compute both the network input and the label in closed form. Let $\mu_t(x_1) \in \mathbb{R}^d$ and $\sigma_t(x_1) > 0$ be differentiable in t, and set

$$r_t(\cdot \mid x_1) = \mathcal{N}(\mu_t(x_1), \sigma_t(x_1)^2 I_d), \qquad \mu_0(x_1) = 0, \quad \sigma_0(x_1) = 1. \tag{1.11}$$

We choose $\mu_1(x_1) = x_1$ and $\sigma_1(x_1) = \sigma_{\min}$, so that each path ends at a small Gaussian around its example. Perhaps the simplest way to parametrize a corresponding flow map $\phi_t(\cdot \mid x_1)$ is as follows:

$$\phi_t(z \mid x_1) := \mu_t(x_1) + \sigma_t(x_1) z, \tag{1.12}$$

which maps $Z \sim \mathcal{N}(0, I_d)$ to the desired conditional Gaussian.

Theorem 1.3 (Velocity of the affine Gaussian flow; [1, Theorem 3])**.** *For each fixed x_1, on any time interval where $\sigma_t(x_1) > 0$, the unique velocity field whose flow is (1.12) is*

$$u_t(x \mid x_1) = \dot{\mu}_t(x_1) + \frac{\dot{\sigma}_t(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)). \tag{1.13}$$

In particular, this velocity generates the Gaussian density path (1.11).

Proof. Differentiate the affine trajectory while keeping its starting point z fixed:

$$\frac{d}{dt}\phi_t(z \mid x_1) = \dot{\mu}_t(x_1) + \dot{\sigma}_t(x_1) z.$$

At a current position $x = \phi_t(z \mid x_1)$, positivity of $\sigma_t(x_1)$ lets us solve for the starting point:

$$z = \frac{x - \mu_t(x_1)}{\sigma_t(x_1)}.$$

Substituting this into the derivative gives $\frac{d}{dt}\phi_t(z \mid x_1) = u_t(\phi_t(z \mid x_1) \mid x_1)$, with u_t as in (1.13). Since $\phi_0(z \mid x_1) = z$, the affine map satisfies both the flow equation and its initial condition. Moreover, $\phi_t(\cdot \mid x_1)$ maps onto all of $\mathbb{R}^d$, so the flow equation pins down $u_t(x \mid x_1)$ at every x, and any other velocity with the same flow must agree with (1.13). Finally, the affine image of a standard Gaussian has mean $\mu_t(x_1)$ and covariance $\sigma_t(x_1)^2 I_d$, which gives the density claim. □

The uniqueness in Theorem 1.3 holds once the *flow map* (1.12) is fixed.

Using (1.12) to sample the conditional input gives the implementable loss

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t, X_1, Z} \| v_\theta(t, \mu_t(X_1) + \sigma_t(X_1) Z) - (\dot{\mu}_t(X_1) + \dot{\sigma}_t(X_1) Z) \|^2, \tag{1.14}$$

where $t \sim \mathrm{Unif}(0, 1)$, $X_1 \sim P_{\mathrm{data}}$, and $Z \sim \mathcal{N}(0, I_d)$ are independent. Each stochastic gradient step needs only samples of these three variables and one network evaluation per sample. We never integrate the conditional flow or the learned flow.

1.4 VE diffusion paths and straight-line paths

Example I: the VE diffusion path. For this example, let generation time range over $t \in [0, T]$, with forward noising time $\tau = T - t$. Let η_τ be an increasing noise standard deviation, with $\eta_0 = 0$, and consider the conditional path

$$r_t(\cdot \mid x_1) = \mathcal{N}(x_1, \eta_{T-t}^2 I_d). \tag{1.15}$$

Thus $\mu_t(x_1) = x_1$ and $\sigma_t(x_1) = \eta_{T-t}$. Here the starting marginal is the law of $X_1 + \eta_T Z$, where $X_1 \sim P_{\text{data}}$ and $Z \sim \mathcal{N}(0, I_d)$ are independent. For large η_T, sampling approximates this law by $\mathcal{N}(0, \eta_T^2 I_d)$.

Differentiating the affine representation $\phi_t(z \mid x_1) = x_1 + \eta_{T-t} z$, as in Theorem 1.3, gives the conditional velocity

$$u_t(x \mid x_1) = -\frac{\eta'_{T-t}}{\eta_{T-t}}(x - x_1), \qquad t < T, \tag{1.16}$$

where η'_τ denotes the derivative in forward time. To identify the marginal velocity, compute the Gaussian conditional score:

$$\nabla_x \log r_t(x \mid x_1) = -\frac{x - x_1}{\eta_{T-t}^2}, \qquad u_t(x \mid x_1) = \eta_{T-t}\eta'_{T-t}\nabla_x \log r_t(x \mid x_1).$$

The posterior average of the conditional score equals the marginal score, since

$$\int \nabla_x \log r_t(x \mid x_1) \frac{r_t(x \mid x_1)}{p_t(x)} P_{\text{data}}(dx_1) = \frac{\nabla_x \int r_t(x \mid x_1) P_{\text{data}}(dx_1)}{p_t(x)} = \nabla \log p_t(x).$$

Theorem 1.1 therefore gives

$$u_t(x) = \eta_{T-t}\eta'_{T-t}\nabla \log p_t(x). \tag{1.17}$$

In the notation of the previous lecture, the forward VE SDE is

$$dX_\tau = g(\tau)\, dB_\tau, \qquad g(\tau)^2 = \frac{d}{d\tau}\eta_\tau^2 = 2\eta_\tau\eta'_\tau, \qquad X_0 \sim P_{\text{data}}.$$

Write ρ_τ for its marginal density, so $p_t = \rho_{T-t}$. Substituting these identities into (1.17) turns the flow-matching ODE into

$$\frac{dY_t}{dt} = u_t(Y_t) = \frac{g(T-t)^2}{2}\nabla \log \rho_{T-t}(Y_t). \tag{1.18}$$

This is the reverse probability flow ODE from the previous lecture, i.e., the continuous-time ODE behind deterministic DDIM sampling [3, 4]. If we parameterize the velocity as $v_\theta(t, x) = \frac{1}{2}g(T-t)^2 s_\theta(T-t, x)$, the learned flow-matching ODE is the probability flow ODE with the score replaced by $s_\theta(T-t, x)$. For example, $g(\tau) = \sqrt{2\tau}$ gives $\eta_\tau = \tau$, and the velocity is $(T-t)s_\theta(T-t, x)$.

Example II: a straight-line path. Flow matching lets us choose other probability paths directly, without deriving them from a forward diffusion. Returning to $t \in [0, 1]$, choose linear mean and standard deviation, with $\sigma_{\min} = 0$:

$$\mu_t(x_1) = tx_1, \qquad \sigma_t(x_1) = 1 - t, \qquad r_t(\cdot \mid x_1) = \mathcal{N}(tx_1, (1-t)^2 I_d). \tag{1.19}$$

Theorem 1.3 gives

$$u_t(x \mid x_1) = x_1 - \frac{x - tx_1}{1 - t} = \frac{x_1 - x}{1 - t}, \qquad t < 1. \tag{1.20}$$

The corresponding conditional flow is

$$\phi_t(z \mid x_1) = (1-t)z + tx_1, \qquad \frac{d}{dt}\phi_t(z \mid x_1) = x_1 - z. \tag{1.21}$$

Each conditional trajectory is the straight line from the noise sample z to the data point x_1. Substituting (1.21) into (1.14) gives the CFM loss

$$\boxed{\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, X_1, Z}\|v_\theta(t, (1-t)Z + tX_1) - (X_1 - Z)\|^2,} \tag{1.22}$$

where $t \sim \text{Unif}(0, 1)$, $X_1 \sim P_{\text{data}}$, and $Z \sim \mathcal{N}(0, I_d)$ are independent.

2 Consistency models

A consistency model [5] learns to map any point on an ODE trajectory to the trajectory's endpoint, so that sampling takes one network evaluation. Return to the notation of Lectures 16–17: t is forward noising time, p_t is the forward marginal density, and $s = T - t$ is reverse time. The reverse probability flow ODE is

$$\frac{dY_s}{ds} = -f(T-s, Y_s) + \frac{g(T-s)^2}{2}\nabla \log p_{T-s}(Y_s). \tag{2.1}$$

Assume its flow is well-defined on the time interval we consider, and let $F(s, r, x)$ denote the solution at reverse time s starting from x at reverse time r. Then $Y_s = F(s, r, Y_r)$, and the endpoint map

$$C(t, x) := F(T, T-t, x) \quad \text{satisfies} \quad C(t, Y_{T-t}) = Y_T. \tag{2.2}$$

So C takes the same value at every point of a trajectory. We learn $C_\theta(t, x) \approx C(t, x)$ subject to the boundary condition $C_\theta(0, x) = x$. To generate a sample, draw $Y_0 \sim p_{\text{noise}}$ and output $C_\theta(T, Y_0)$.

Consistency distillation. Let s_ϕ be a fixed pretrained score model and choose a small step $\delta > 0$. Sample $t \sim \text{Unif}(\delta, T)$ and $Y_+ \sim p_t$ using the forward corruption. For the VE process with $g(t) = \sqrt{2t}$, this means drawing independent $X_0 \sim P_{\text{data}}$ and $Z \sim \mathcal{N}(0, I_d)$ and setting $Y_+ = X_0 + tZ$. Take one reverse ODE step:

$$Y_- = Y_+ + \delta\left[-f(t, Y_+) + \frac{g(t)^2}{2} s_\phi(t, Y_+)\right]. \tag{2.3}$$

For VE, this is $Y_- = Y_+ + \delta t\, s_\phi(t, Y_+)$. The pair (Y_+, Y_-) approximates two points on the same ODE trajectory, at reverse times $T-t$ and $T-t+\delta$. If the step were exact, giving $Y_-^* = F(T-t+\delta, T-t, Y_+)$, then composing flow maps would give

$$C(t, Y_+) = F(T, T-t, Y_+) = F(T, T-t+\delta, Y_-^*) = C(t-\delta, Y_-^*).$$

This motivates the consistency-distillation objective

$$\mathcal{L}_{\text{CD}}(\theta; \phi) = \mathbb{E}_{t, Y_+}\left[\lambda(t)\|C_\theta(t, Y_+) - C_\theta(t-\delta, Y_-)\|^2\right], \tag{2.4}$$

where $\lambda(t) > 0$ is a weight function and Y_- is computed using (2.3). Restricting to $t \geq \delta$ keeps both time arguments of C_θ in $[0, T]$.

References

[1] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR*, 2023. https://arxiv.org/abs/2210.02747.

[2] Wenpin Tang and Hanyang Zhao. *Score-based diffusion models via stochastic differential equations: A technical tutorial*. arXiv:2402.07487, 2024. https://arxiv.org/abs/2402.07487.

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021. https://arxiv.org/abs/2011.13456.

[4] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021. https://arxiv.org/abs/2010.02502.

[5] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *ICML*, 2023. https://arxiv.org/abs/2303.01469.