Score-Based Generative Models: Flow Matching

Nathan Cantafio

Jun 04, 2026

Probability flow ODE can be viewed as a continuous normalizing flow whose vector field is determined by the score associated with a pre-specified noise schedule. This note asks whether one can instead *directly* parametrize and train a generative ODE without fixing the noise schedule first. We develop continuous normalizing flows and Neural ODEs as a natural answer, then expose their central bottleneck: training requires numerical ODE integration at every gradient step. Flow matching [1, 2, 3] resolves this with a simulation-free regression objective. The key insight, structurally identical to the passage from score matching to denoising score matching, is to condition on a latent variable z so that the intractable marginal velocity field is replaced by a closed-form conditional one, without changing the set of optimal solutions. We work through two explicit examples: a one-sided Gaussian path and a two-sided linear interpolation path. We close by situating diffusion models as a special case of flow matching and identifying the simulation-free conditioning trick as a unifying motif across the series.

1. Introduction

The previous three notes developed denoising score matching as a method to learn the score $\nabla_x \log p_t(x)$, and showed that this score determines two equivalent sampling procedures: the reverse-time SDE and the probability flow ODE. Both are tied to a specific choice of forward process (VE, VP, or sub-VP), which fixes the noise schedule and, with it, the entire family of marginals $\{p_t\}$.

A natural question is whether this coupling is necessary. The probability flow ODE is an instance of a *continuous normalizing flow* (CNF): a generative model defined by an ODE $dx/dt = v_\phi(x, t)$ whose vector field v_ϕ is a neural network [4]. In the diffusion setting, v_ϕ is constrained to take the specific form $f - \frac{g^2}{2}\nabla_x \log p_t$; but nothing prevents one from parametrizing v_ϕ freely and training it directly. The obstacle, as we shall see, is computational: training CNFs via maximum likelihood requires integrating the ODE at every gradient step, which is prohibitively expensive at scale.

Flow matching [1, 2, 3] avoids this by framing the problem as regression. Rather than training v_ϕ to maximize likelihood, one specifies a target probability path $\{p_t\}_{t\in[0,1]}$ from a source distribution p_0 to a target distribution p_1, and regresses v_ϕ directly onto the vector field v_t that generates this path. The regression is simulation-free: no ODE integration is required during training.

The note is organized as follows. Section 2 reviews normalizing flows, motivates the continuous limit as a Neural ODE, and exposes the training cost. Section 3 develops the flow matching framework, introduces the conditional flow matching objective, proves that the two objectives share the same gradients, and works through the Gaussian and linear interpolation examples. Section 4 situates the results: diffusion models as a special case, and the recurring simulation-free conditioning motif.

2. Normalizing Flows and Continuous Normalizing Flows

2.1. Normalizing flows

A *normalizing flow* is an invertible, differentiable map $S_\phi : \mathbb{R}^d \to \mathbb{R}^d$ that pushes a simple prior p_{prior} forward to an approximation of the data distribution. Given $z \sim p_{\text{prior}}$, a sample is produced as $x = S_\phi(z)$. By the change-of-variables formula, the induced density at x is

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

where $z = S_\phi^{-1}(x)$. The model can therefore be trained by maximum likelihood estimation (MLE): maximize $\mathbb{E}_{x \sim p_{\text{data}}}[\log p_\phi(x)]$ over ϕ.

The names are suggestive. "Normalizing" refers to the Jacobian factor, which re-normalizes the density after the change of variables. "Flow" refers to the fact that the latent z is *flowed* through the map S_ϕ to reach the data space x.

The principal architectural constraint is that the Jacobian determinant must be tractable. One standard approach is to compose L layers, each with a cheap log-determinant:

$$S_\phi = S_{L-1} \circ \cdots \circ S_0, \qquad \log p_\phi(x) = \log p_{\text{prior}}(z) + \sum_{k=0}^{L-1} \log \left| \det \frac{\partial S_{\phi_k}(x_k)}{\partial x_k} \right|,$$

where x_k is the intermediate state after k layers.

2.2. From discrete to continuous: Neural ODEs

Consider a residual layer of the form

$$x_{k+1} = S_k(x_k) = x_k + v_{\phi_k}(x_k, k),$$

where v_{ϕ_k} is a Lipschitz network with Lipschitz constant less than one (so that the map is invertible). This is the Euler discretization of the ODE

$$\frac{dx(t)}{dt} = v_\phi(x(t), t), \qquad t \in [0, T], \tag{2}$$

in the limit where the step size goes to zero and the depth goes to infinity. The resulting model — a *Neural ODE* [4] — parametrizes the generative process directly as a continuous-time flow. The latent $z = x(0)$ is drawn from the prior p_{prior}, and the sample is $x = x(T)$, obtained by integrating (2) forward.

The continuous analogue of the log-determinant sum is the *instantaneous change-of-variables formula* (derived in Note 03, §4):

$$\frac{d}{dt} \log p_\phi(x(t), t) = -\nabla_x \cdot v_\phi(x(t), t), \tag{3}$$

where $\nabla_x \cdot$ denotes the divergence. Integrating from 0 to T:

$$\log p_\phi(x(T)) = \log p_{\text{prior}}(x(0)) - \int_0^T \nabla_x \cdot v_\phi(x(t), t)\, dt. \tag{4}$$

This is the CNF log-likelihood. It recovers the discrete sum (1) in the Euler limit. Probability flow ODE is a specific CNF whose vector field is constrained to the form $f(x,t) - \frac{g(t)^2}{2} \nabla_x \log p_t(x)$; a Neural ODE places no such constraint on v_ϕ.

2.3. Training CNFs and its cost

Given the likelihood (4), one can in principle train v_ϕ by MLE. The trajectory $\{x(t)\}_{t\in[0,T]}$ and the integrated divergence are computed simultaneously by solving the ODE (4).

However, at every gradient step, one must numerically integrate the ODE over $[0, T]$, which requires many evaluations of v_ϕ and $\nabla_x \cdot v_\phi$. The divergence term alone costs $\mathcal{O}(d)$ backward passes per time-step (or one pass with Hutchinson's estimator, but still with non-trivial variance). At the scale of modern generative models, running a full ODE solve inside every training step is prohibitive.

This is the central motivation for flow matching: can we train v_ϕ to match a target flow *without ever integrating the ODE*?

3. Flow Matching

3.1. Setup and goal

Flow matching [1, 2, 3] begins by specifying a target *probability path*: a family of densities $\{p_t\}_{t\in[0,1]}$ interpolating between a source $p_0 = p_{\text{src}}$ and a target $p_1 = p_{\text{target}}$. We adopt the convention $t = 0$ for source and $t = 1$ for target throughout this section. The goal is to find a time-dependent vector field $v_t : \mathbb{R}^d \to \mathbb{R}^d$ such that the ODE

$$\frac{dx}{dt} = v_t(x_t), \qquad x_0 \sim p_0,$$

generates samples $x_1 \sim p_1$. More precisely, we require that the marginal of x_t under this ODE matches p_t for all t. By the continuity equation (equivalently, the Fokker–Planck equation with no diffusion term), this is the condition

$$\frac{\partial p_t(x)}{\partial t} + \nabla_x \cdot \big(v_t(x)\, p_t(x)\big) = 0. \tag{5}$$

Any vector field satisfying (5) will generate the desired path. Note that the continuity equation does not uniquely determine v_t: one may always add a divergence-free (curl-like) perturbation u_t satisfying $\nabla_x \cdot (u_t\, p_t) = 0$ and obtain another valid field. This gauge freedom is the continuous-time analogue of the fact that many normalizing flow architectures can represent the same distribution.

3.2. The flow matching objective and its intractability

Given a valid target field v_t, the most natural training objective is a squared regression loss:

$$\mathcal{L}_{\text{FM}}(\phi) = \mathbb{E}_t\, \mathbb{E}_{x_t\sim p_t}\big[\|v_\phi(x_t, t) - v_t(x_t)\|^2\big], \tag{6}$$

where the outer expectation is over t drawn uniformly on $[0, 1]$.

Unfortunately, $v_t(x)$ is the *marginal* velocity field, which requires integrating over the entire data distribution:

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

This integral is generally intractable, mirroring exactly the situation in score matching where $\nabla_x \log p_t(x)$ is intractable because it requires knowing the marginal density. The solution in both cases is the same: introduce a latent variable and condition.

3.3. Conditional flow matching

Introduce a latent variable $z \in \mathbb{R}^m$ with known distribution $\pi(z)$, and define a *conditional probability path* $p_t(x \mid z)$ such that the marginal $p_t(x) = \int p_t(x \mid z)\, \pi(z)\, dz$ recovers the

desired path. Suppose further that, for each z, there exists a *conditional velocity field* $v_t(x \mid z)$ generating the conditional path $p_t(\cdot \mid z)$:

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

Then the *conditional flow matching* (CFM) objective is

$$\mathcal{L}_{\mathrm{CFM}}(\phi) = \mathbb{E}_t\, \mathbb{E}_{z\sim\pi}\, \mathbb{E}_{x_t\sim p_t(\cdot|z)} \big[\|v_\phi(x_t,t) - v_t(x_t \mid z)\|^2\big]. \tag{7}$$

The conditional field $v_t(x \mid z)$ can be designed to be closed-form, and sampling from $p_t(x \mid z)$ is cheap by construction. The key result is that the two objectives (6) and (7) have the same gradients with respect to ϕ, so minimizing either yields the same optimal network.

Theorem 3.1 ($\mathcal{L}_{\mathrm{FM}} = \mathcal{L}_{\mathrm{CFM}} + C$)**.** *For any ϕ,*

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

where C does not depend on ϕ. In particular, $\nabla_\phi \mathcal{L}_{\mathrm{FM}}(\phi) = \nabla_\phi \mathcal{L}_{\mathrm{CFM}}(\phi)$.

Proof. Fix t and x_t, and expand $\mathcal{L}_{\mathrm{FM}}$ by adding and subtracting the conditional field:

$$\mathbb{E}_{x_t\sim p_t}\big[\|v_\phi(x_t,t) - v_t(x_t)\|^2\big] = \mathbb{E}_{x_t\sim p_t}\big[\|v_\phi(x_t,t)\|^2\big] - 2\,\mathbb{E}_{x_t\sim p_t}\big[\langle v_\phi(x_t,t),\, v_t(x_t)\rangle\big] + \mathbb{E}_{x_t\sim p_t}\big[\|v_t(x_t)\|^2\big].$$

The last term is independent of ϕ. We show the cross term in $\mathcal{L}_{\mathrm{FM}}$ equals the corresponding cross term in $\mathcal{L}_{\mathrm{CFM}}$. By the tower property and the definition of the marginal field:

$$\begin{aligned}
\mathbb{E}_{x_t\sim p_t}\big[\langle v_\phi(x_t,t),\, v_t(x_t)\rangle\big] &= \int \langle v_\phi(x_t,t),\, v_t(x_t)\rangle\, p_t(x_t)\, dx_t \\
&= \int \left\langle v_\phi(x_t,t), \frac{\int v_t(x_t \mid z)\, p_t(x_t \mid z)\, \pi(z)\, dz}{p_t(x_t)} \right\rangle p_t(x_t)\, dx_t \\
&= \iint \langle v_\phi(x_t,t),\, v_t(x_t \mid z)\rangle\, p_t(x_t \mid z)\, \pi(z)\, dz\, dx_t \\
&= \mathbb{E}_{z\sim\pi}\, \mathbb{E}_{x_t\sim p_t(\cdot|z)}\big[\langle v_\phi(x_t,t),\, v_t(x_t \mid z)\rangle\big].
\end{aligned}$$

The first term $\mathbb{E}_{x_t\sim p_t}[\|v_\phi(x_t,t)\|^2]$ is equal in both objectives because $p_t(x_t) = \int p_t(x_t \mid z)\, \pi(z)\, dz$:

$$\begin{aligned}
\mathbb{E}_{z\sim\pi}\, \mathbb{E}_{x_t\sim p_t(\cdot|z)}\big[\|v_\phi(x_t,t)\|^2\big] &= \iint \|v_\phi(x_t,t)\|^2\, p_t(x_t \mid z)\, \pi(z)\, dz\, dx_t \\
&= \int \|v_\phi(x_t,t)\|^2\, p_t(x_t)\, dx_t = \mathbb{E}_{x_t\sim p_t}\big[\|v_\phi(x_t,t)\|^2\big].
\end{aligned}$$

Combining, for each fixed t:

$$\mathbb{E}_{x_t\sim p_t}\big[\|v_\phi - v_t\|^2\big] = \mathbb{E}_{z,x_t}\big[\|v_\phi - v_t(\cdot \mid z)\|^2\big] - \mathbb{E}_{z,x_t}\big[\|v_t(\cdot \mid z)\|^2\big] + \mathbb{E}_{x_t}\big[\|v_t\|^2\big].$$

The last two terms are both independent of ϕ. Integrating over t completes the proof, with $C = \mathbb{E}_t[\mathbb{E}_{x_t}[\|v_t\|^2] - \mathbb{E}_{z,x_t}[\|v_t(\cdot \mid z)\|^2]]$. □

This theorem is the flow matching analogue of the identity $\mathcal{L}_{\mathrm{SM}} = \mathcal{L}_{\mathrm{DSM}} + C$ from Note 02. In both cases, the proof proceeds by the tower property: marginalizing the conditional objective over z recovers the marginal objective, with a constant left over. In both cases, the practical payoff is the same: the intractable marginal objective ($\mathcal{L}_{\mathrm{FM}}$, $\mathcal{L}_{\mathrm{SM}}$) is replaced by a tractable conditional one ($\mathcal{L}_{\mathrm{CFM}}$, $\mathcal{L}_{\mathrm{DSM}}$), without affecting the optimal solution.

Concretely, to evaluate $\mathcal{L}_{\mathrm{CFM}}$ at a single training step, one: (i) draws $t \sim \mathrm{Uniform}[0,1]$; (ii) draws $z \sim \pi(z)$; (iii) draws $x_t \sim p_t(\cdot \mid z)$; and (iv) evaluates $\|v_\phi(x_t,t) - v_t(x_t \mid z)\|^2$. No ODE integration is required. Flow matching is therefore a *simulation-free* alternative to training CNFs via MLE.

3.4. Gaussian conditional paths

A simple choice is the Gaussian conditional path. Fix $z = x_0 \sim p_{\text{data}}$ (so $\pi = p_{\text{data}}$) and set

$$p_t(x_t \mid x_0) = \mathcal{N}\big(x_t;\, \mu(t),\, \sigma(t)^2 I\big), \tag{8}$$

where $\mu : [0,1] \to \mathbb{R}^d$ and $\sigma : [0,1] \to \mathbb{R}_{>0}$ are smooth schedules with $\mu(0) = 0$, $\sigma(0) = 1$ (so $p_0 = \mathcal{N}(0, I)$) and $\mu(1) = x_0$, $\sigma(1) = \sigma_{\min} \approx 0$ (so $p_1 \approx \delta_{x_0}$). The marginal path is $p_t(x_t) = \int p_t(x_t \mid x_0)\, p_{\text{data}}(x_0)\, dx_0$, a mixture of Gaussians.

To find the conditional velocity field, we construct the *flow map* $\psi_t : \mathbb{R}^d \to \mathbb{R}^d$ such that if $y \sim p_0 = \mathcal{N}(0, I)$ then $\psi_t(y) \sim p_t(\cdot \mid x_0)$ for all t. The affine map

$$\psi_t(y) = \mu(t) + \sigma(t)\, y \tag{9}$$

clearly satisfies this: $\psi_t(y) \sim \mathcal{N}(\mu(t), \sigma(t)^2 I)$ when $y \sim \mathcal{N}(0, I)$. Since ψ_t satisfies the ODE $d\psi_t(y)/dt = v_t(\psi_t(y) \mid x_0)$, we can read off the conditional velocity field. Differentiating (9) with respect to t:

$$\frac{d}{dt}\psi_t(y) = \mu'(t) + \sigma'(t)\, y.$$

Inverting the flow map, $y = \psi_t^{-1}(x_t) = (x_t - \mu(t))/\sigma(t)$, and substituting:

$$v_t(x_t \mid x_0) = \mu'(t) + \sigma'(t)\, \psi_t^{-1}(x_t) = \mu'(t) + \frac{\sigma'(t)}{\sigma(t)}\big(x_t - \mu(t)\big). \tag{10}$$

This is the unique vector field consistent with the Gaussian path (8).

Sampling from $p_t(\cdot \mid x_0)$ reduces to: draw $\varepsilon \sim \mathcal{N}(0, I)$ and return $x_t = \mu(t) + \sigma(t)\, \varepsilon$. The CFM objective (7) then becomes

$$\mathcal{L}_{\text{CFM}}(\phi) = \mathbb{E}_t\, \mathbb{E}_{x_0 \sim p_{\text{data}}}\, \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,I)} \left[\left\| v_\phi(\mu(t) + \sigma(t)\varepsilon,\, t) - \mu'(t) - \frac{\sigma'(t)}{\sigma(t)}\big(\mu(t) + \sigma(t)\varepsilon - \mu(t)\big) \right\|^2 \right],$$

which simplifies to

$$\mathcal{L}_{\text{CFM}}(\phi) = \mathbb{E}_t\, \mathbb{E}_{x_0, \varepsilon}\big[\|v_\phi(x_t, t) - (\mu'(t) + \sigma'(t)\varepsilon)\|^2\big], \qquad x_t = \mu(t) + \sigma(t)\varepsilon. \tag{11}$$

Each term involves only a single forward pass through v_ϕ and no ODE solver.

3.5. Two-sided linear interpolation

The Gaussian example above uses a one-sided latent $z = x_0$: the conditional path starts at the prior and collapses to a data point. A more symmetric choice, introduced under the name *stochastic interpolants* by Albergo and Vanden-Eijnden [3] and independently in Liu, Gong, and Liu [2], uses a *two-sided* latent $z = (x_0, x_1)$.

Set $\pi(z) = p_{\text{src}}(x_0)\, p_{\text{target}}(x_1)$ (independent coupling) and define

$$p_t(x_t \mid z) = \mathcal{N}\big(x_t;\, (1-t)x_0 + t\, x_1,\, \sigma^2 I\big), \tag{12}$$

with $\sigma \geq 0$. The boundary conditions are satisfied: at $t = 0$ the mean is x_0 and at $t = 1$ it is x_1. The corresponding flow map is

$$\psi_t(y) = (1-t)x_0 + t\, x_1 + \sigma\, y, \qquad y \sim \mathcal{N}(0, I).$$

Differentiating with respect to t and inverting as before:

$$v_t(x_t \mid z) = x_1 - x_0. \tag{13}$$

The conditional velocity is *constant*: it does not depend on t or on x_t, only on the endpoints x_0 and x_1. In the deterministic limit $\sigma \to 0$, the conditional path collapses to the straight line segment $x_t = (1-t)x_0 + t\, x_1$, and the CFM objective becomes

$$\mathcal{L}_{\text{CFM}}(\phi) = \mathbb{E}_t\, \mathbb{E}_{x_0 \sim p_{\text{src}}}\, \mathbb{E}_{x_1 \sim p_{\text{target}}}\big[\|v_\phi(x_t, t) - (x_1 - x_0)\|^2\big], \qquad x_t = (1-t)x_0 + t\, x_1. \tag{14}$$

4. Connections and Discussion

4.1. Diffusion models as a special case

The connection to diffusion models is essentially definitional. A VP forward process specifies a Gaussian conditional $p_t(x_t \mid x_0)$ with a particular mean and variance schedule; plugging those schedules into the Gaussian path construction of Section 3.4 yields exactly the same conditional velocity field. The probability flow ODE of Note 03 is therefore a CNF whose vector field happens to be the one flow matching would have derived from the VP schedule. The two frameworks are producing the same object by different routes. What flow matching adds is the recognition that the Gaussian structure is not essential: any probability path with a tractable conditional velocity will do.

4.2. Optimal transport and OT-CFM

A practical limitation of the independent coupling $\pi = p_{\text{src}} \otimes p_{\text{target}}$ is that the marginal flow can be curved even though each conditional trajectory is a straight line. Because unrelated source and target points are paired at random, the superposition of straight-line paths produces a tangled marginal vector field, and integrating it accurately at inference requires many function evaluations.

Optimal transport [5] offers a remedy: rather than pairing (x_0, x_1) at random, one can choose the coupling π^* that minimizes expected squared displacement between source and target. Under this pairing, mass travels as directly as possible, and the resulting marginal flow is approximately straight. Tong et al. [6] show that substituting π^* for the independent coupling in (14) yields a learned ODE that can be integrated accurately with far fewer steps at inference, at no change to the training procedure beyond how pairs are drawn.

5. Conclusion

We have traced a progression from normalizing flows through Neural ODEs to flow matching. Normalizing flows provide exact likelihoods at the cost of constrained architectures. Neural ODEs remove the architectural constraint by taking the continuous limit, but training via MLE requires integrating the ODE at every gradient step. Flow matching eliminates this cost entirely: the CFM objective is evaluated by drawing a latent z, sampling x_t from the conditional path, and computing a single forward pass.

Two examples are illustrated in Section 3. The Gaussian path recovers diffusion models as a special case; the two-sided linear interpolation admits non-Gaussian sources and, when pairs are drawn from an optimal transport plan, produces nearly straight trajectories that are cheap to integrate at inference. The underlying trick of conditioning on a latent to make an intractable marginal objective tractable is the same trick that carries denoising score matching from score matching. It is perhaps the central motif of this series: a well-chosen conditional structure makes an otherwise intractable objective useable.

References

[1] Y. Lipman et al. “Flow Matching for Generative Modeling”. *International Conference on Learning Representations* (2023).
[2] X. Liu, C. Gong, and Q. Liu. “Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow”. *International Conference on Learning Representations* (2023).
[3] M. S. Albergo and E. Vanden-Eijnden. “Building Normalizing Flows with Stochastic Interpolants”. *International Conference on Learning Representations* (2023).
[4] R. T. Q. Chen et al. “Neural Ordinary Differential Equations”. *Advances in Neural Information Processing Systems* 31 (2018).
[5] C. Villani. *Optimal Transport: Old and New*. Springer, 2009.
[6] A. Tong et al. “Improving and Generalizing Flow-Matching for Conditional Generation”. *International Conference on Machine Learning* (2023).