Flow Matching

Karl Stratos

See Appendix A for notation (e.g., $\dot{f}_t(x_t) = (\partial_t f_t)(x_t)$, $\nabla \cdot f(x) = \operatorname{tr}(J_f(x))$) and basic flow concepts.

1 Unconditional Generation

We can define a random bridge $X_t \sim p_t$ between $p_0 := \mathcal{N}(0_d, I_d)$ and $p_1 = q$ (dataset) by

$$X_t = (1-t)X_\varnothing + tX_\star \tag{1}$$

where $X_\varnothing \sim p_0$ and $X_\star \sim q$.[1] Lipman *et al.* (2022) propose to "match" the density of X_t by a unidirectional flow $Y_t = \phi_t(Y_0)$ where $Y_0 \sim p_0$. WLOG, it is more convenient to use the flow's velocity $u_t = \dot{\phi}_t \circ \phi_t^{-1}$ and model the per-sample speed of (1) which takes the simple form $U_t = \dot{X}_t = X_\star - X_\varnothing$. Indeed, if

$$u_t(x) = \mathbf{E}\left[U_t \middle| X_t = x\right] \tag{2}$$

then $p_t = (\phi_t)_\# p_0$. Thus $Y_t \sim p_t$, in particular $Y_1 \sim q$.

1.1 Justification

A sufficient condition for any velocity field u_t to imply $p_t = (\phi_t)_\# p_0$ is the mass-volume flux equilibrium (Lemma A.1)

$$\dot{p}_t + \nabla \cdot (p_t u_t) = 0 \tag{3}$$

because it makes $p_t(Y_t) \det J_{\phi_t}(Y_0)$ time invariant, which implies $p_t(Y_t) = p_0(Y_0) \det J_{\phi_t^{-1}}(Y_t)$ (using $\phi_0(x) = x$). The equilibrium holds for (2) mainly due to the chain rule $\frac{d}{dt} f(X_t) = \nabla f(X_t)^\top U_t$. Taking expectation on both sides and applying integration by parts gives us (Lemma B.2 and B.1):

$$\int_x f(x)\dot{p}_t(x)dx = -\int_x f(x)\nabla \cdot (p_t(x)\mathbf{E}[U_t|X_t = x])\, dx$$

which implies (3).

1.2 Training

The most direct approach to obtain (2) is to train a velocity predictor $u_t^\theta : \mathbb{R}^d \to \mathbb{R}^d$ by

$$\theta^\star = \arg\min_\theta \ \mathbf{E}\left[\left|\left|U_t - u_t^\theta(X_t)\right|\right|_2^2\right] \tag{4}$$

where $t \sim \mathcal{T}[0,1]$ for some density, since the optimal model satisfies $u_t^{\theta^\star}(x) = \mathbf{E}\left[U_t \middle| X_t = x\right]$ by the property of squared error (assuming universality). However, we can reparameterize (2) as (for $0 < t < 1$)

$$u_t(x) = \frac{\tau_t(x) - x}{1-t} = \frac{x - \epsilon_t(x)}{t} \qquad \tau_t(x) = \mathbf{E}[X_\star | X_t = x] \qquad \epsilon_t(x) = \mathbf{E}[X_\varnothing | X_t = x] \tag{5}$$

Thus we can alternatively train a target τ_t or noise predictor ϵ_t, then convert them into velocity u_t at inference time. While these are equivalent under perfect models, Li and He (2026) show that target prediction is often more performant with finite networks, likely because $X_\star \sim q$ lies on a low-dimensional manifold (e.g., natural images) unlike pure noise $X_\varnothing$ or noised residual U_t.

[1] Note that (1) implies $(X_t | X_\star = x) \sim \mathcal{N}(tx, (1-t)^2 I_d)$, so the conditional marginal density is a Gaussian perturbation as in DDPM's forward noising process. Thus they are roughly same model (up to time reversal and noise schedules). But instead of relying on MLE, or generating in one shot by adversarial training like GANs, FM models conditional velocity (2).

1.2.1 Target/noise loss

Under infinite sample, it is sufficient to use vanilla squared error to recover the correct target/noise predictors (5) (e.g., $\min_\theta \mathbf{E}[\|X_\star - \tau_t^\theta(X_t)\|_2^2]$). But under finite sample, since they are converted to velocity at inference time, it may be more efficient to match the velocity loss (4) and optimize

$$\min_\theta \mathbf{E}\left[\left|\left|U_t - u_t^\theta(X_t)\right|\right|_2^2\right] = \min_\theta \mathbf{E}\left[\frac{1}{(1-t)^2}\left|\left|X_\star - \tau_t^\theta(X_t)\right|\right|_2^2\right] = \min_\theta \mathbf{E}\left[\frac{1}{t^2}\left|\left|X_\varnothing - \epsilon_t^\theta(X_t)\right|\right|_2^2\right]$$

Note the intuitive time corrections (heavier penalty as $t \to 1$ for target prediction, $t \to 0$ for noise).

1.3 Inference

Assuming the possession of $u_t(x) = \mathbf{E}[U_t|X_t = x]$ (under standard regularity conditions) corresponding to some implicit flow ϕ_t, we can sample $Y_0 \sim p_0$ and transform it into $Y_1 \sim q$ by solving the ODE or SDE.

ODE. We can find the unique $Y_t = \phi_t(Y_0)$ deterministically by solving

$$\dot{Y}_t = u_t(Y_t) \quad \Leftrightarrow \quad dY_t = u_t(Y_t)dt \quad \Rightarrow \quad \boxed{Y_{t+\Delta t} = Y_t + \Delta t \times u_t(Y_t)} \tag{6}$$

The last expression is a decoding algorithm (Euler's method) where Δt partitions $[0, 1]$ into T intervals.

SDE. If we want to make the flow itself stochastic, we can find $Y_t \in \mathbb{R}^d$ satisfying

$$dY_t = \left(u_t(Y_t) + \frac{w_t}{2}\nabla_x \log p_t(x)|_{x=Y_t}\right)dt + \sqrt{w_t}dW_t \tag{7}$$

for some noise schedule $w_t \geq 0$ (back to ODE if $w_t = 0$). Here $W_t \in \mathbb{R}^d$ is a cumulative random noise up to time t (aka. Brownian motion/Wiener process), with the property $dW_t \approx W_{t+\Delta t} - W_t \sim \mathcal{N}(0_d, \Delta t I_d)$. The solution is known to satisfy $Y_t \sim p_t$ (Lemma B.3). The score function has the closed-form expression $\nabla_x \log p_t(x) = \frac{tu_t(x)-x}{1-t}$ under the Gaussian conditional $p_t(\cdot|X_\star = z) = \mathcal{N}(tz, (1-t)^2 I_d)$ (Lemma B.4). Thus the stochastic decoding algorithm (Euler-Maruyama sampler) is

$$Y_{t+\Delta t} = Y_t + \Delta t \times \left(u_t(Y_t) + \frac{w_t(tu_t(Y_t) - Y_t)}{2(1-t)}\right) + \sqrt{w_t \Delta t} \times \xi_t \qquad \xi_t \sim \mathcal{N}(0_d, I_d) \tag{8}$$

2 Conditional Generation

We now assume jointly distributed $(X_\star, C) \sim q$ where C is the "class" of target $X_\star$ (e.g., image-text pairs). The framework in Section 1 remains the same except that we continuously condition on $C = c$:

$$u_t(x|c) = \mathbf{E}\left[U_t \middle| X_t = x, C = c\right] = \frac{\tau_t(x|c) - x}{1-t} = \frac{x - \epsilon_t(x|c)}{t} \tag{9}$$

where $\tau_t(x|c) = \mathbf{E}[X_\star|X_t = x, C = c]$ and $\epsilon_t(x|c) = \mathbf{E}[X_\varnothing|X_t = x, C = c]$ are target and noise reparameterizations. Again, (9) can be obtained by training a class-conditional velocity/target/noise model on samples of $(X_\star, C)$, e.g., by minimizing the velocity loss

$$\min_\theta \mathbf{E}\left[\left|\left|U_t - u_t^\theta(X_t|C)\right|\right|_2^2\right] = \min_\theta \mathbf{E}\left[\frac{1}{(1-t)^2}\left|\left|X_\star - \tau_t^\theta(X_t|C)\right|\right|_2^2\right] = \min_\theta \mathbf{E}\left[\frac{1}{t^2}\left|\left|X_\varnothing - \epsilon_t^\theta(X_t|C)\right|\right|_2^2\right] \tag{10}$$

so that $u_t(x|c) = u_t^{\theta^\star}(x|c)$, $\tau_t(x|c) = \tau_t^{\theta^\star}(x|c)$, and $\epsilon_t(x|c) = \epsilon_t^{\theta^\star}(x|c)$. Once we have (9), conditioned on a particular class $C = c$ we can find a flow from $Y_0 \sim p_0$ satisfying $\dot{Y}_t = u_t(Y_t|c)$ (or SDE) to generate $Y_1 \sim q(\cdot|C = c)$.

2.1 Classifier-Free Guidance

At inference time, it is popular to use classifier-free guidance (CFG):[2] given $w \geq 0$ define the modified velocity

$$u_t^{\mathrm{CFG}}(x|c,w) = (1-w)u_t(x|\varnothing) + wu_t(x|c) = u_t(x|c) + (w-1)\underbrace{(u_t(x|c) - u_t(x|\varnothing))}_{\text{residual velocity}} \tag{11}$$

where $u_t(x|\varnothing) = \mathbf{E}[U_t|X_t = x]$ is obtained by conditioning on a null class $C = \varnothing$.[3] Intuitively, CFG sharpens conditioning when $w > 1$. Under target/noise parameterization, we can guide generation at the target/noise level (e.g., $\tau_t^{\mathrm{CFG}}(x|c,w) = (1-w)\tau_t(x|\varnothing) + w\tau_t(x|c)$) and recover

$$u_t^{\mathrm{CFG}}(x|c,w) = \frac{\tau_t^{\mathrm{CFG}}(x|c,w) - x}{1-t} = \frac{x - \epsilon_t^{\mathrm{CFG}}(x|c,w)}{t} \tag{12}$$

thanks to the affine relationship $u_t(x|c) = \frac{\tau_t(x|c)-x}{1-t} = \frac{x-\epsilon_t(x|c)}{t}$.

2.1.1 Training-time CFG

CFG requires doing 2 forward passes at inference time. We can delegate the compute overhead to training time by self-distillation: train $u_t^\theta(x|c,w)$ to directly predict (11) by

$$\min_\theta \ \mathbf{E}\left[\left|\left|U_t - u_t^\theta(X_t|\varnothing,w)\right|\right|_2^2\right] + \mathbf{E}\left[\left|\left|\left((1-w)\mathbf{sg}\{u_t^\theta(X_t|\varnothing,w)\} + wU_t\right) - u_t^\theta(X_t|C,w)\right|\right|_2^2\right] \tag{13}$$

where $w \sim \mathcal{W}[w_{\min}, w_{\max}]$ for some appropriate density. The first term enforces $u_t^{\theta^\star}(x|\varnothing,w) = u_t(x|\varnothing)$ at optimum. The second term bootstraps the first term by the stop-gradient trick and enforces the correct behavior $u_t^{\theta^\star}(x|c,w) = (1-w)u_t(x|\varnothing) + wu_t(x|c)$. One downside of (13) is variance amplification, specifically

$$\mathrm{Var}\left((1-w)\mathbf{sg}\{u_t^\theta(X_t|\varnothing,w)\} + wU_t \mid X_t = x,\ C = c\right) = w^2\mathrm{Var}\left(U_t \mid X_t = x,\ C = c\right)$$

grows in $w > 1$ and affects the variance of the gradient. Geng *et al.* (2026) propose a variance reduction method that uses a fixed-point characterization to avoid scaling U_t:

$$\min_\theta \ \mathbf{E}\left[\left|\left|U_t - u_t^\theta(X_t|\varnothing,w)\right|\right|_2^2\right] + \mathbf{E}\left[\left|\left|\left(U_t + \left(1-\frac{1}{w}\right)\mathbf{sg}\left\{u_t^\theta(X_t|C,w) - u_t^\theta(X_t|\varnothing,w)\right\}\right) - u_t^\theta(X_t|C,w)\right|\right|_2^2\right] \tag{14}$$

At optimum, the model satisfies $u_t^{\theta^\star}(x|c,w) = u_t(x|c) + \left(1-\frac{1}{w}\right)\left(u_t^{\theta^\star}(x|c,w) - u_t(x|\varnothing)\right)$, which is equivalent to $u_t^{\theta^\star}(x|c,w) = (1-w)u_t(x|\varnothing) + wu_t(x|c)$. We can self-distill CFG target/noise prediction in the same vein, after which we recover the CFG velocity by $u_t^{\mathrm{CFG}}(x|c,w) = \frac{\tau_t^{\theta^\star}(x|c,w)-x}{1-t} = \frac{x-\epsilon_t^{\theta^\star}(x|c,w)}{t}$ in 1 forward pass (12).

References

Geng, Z., Deng, M., Bai, X., Kolter, Z., and He, K. (2026). Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, **38**, 75460–75482.

Li, T. and He, K. (2026). Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 36115–36125.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2022). Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

[2] The name is historical. Earlier work used an explicit classifier $p_t(c|x)$ in the context of image classification, and guided generation by adding the class direction $\nabla_x \log p_t(c|x)$. Later work avoided the classifier by Bayes' rule $\nabla_x \log p_t(c|x) = \nabla_x \log p_t(x|c) - \nabla_x \log p_t(x)$ which corresponds to accentuating the residual in the generative model itself.

[3] In practice, this is done by replacing $C \leftarrow \varnothing$ with probability p (e.g., 0.1) for compute efficiency (i.e., rather than doing 2 passes).

A Flow Concepts

A.1 Notation

We often differentiate time-varying functions $f_t : \mathbb{R}^d \to \mathbb{R}^d$ taking time-varying inputs $x_t \in \mathbb{R}^d$ with respect to t. To avoid ambiguity, we write $\dot{f}_t(x_t) = \partial_t f_t(x_t)$ to denote fixed-input derivatives where $\partial_t f_t : \mathbb{R}^d \to \mathbb{R}^d$ is the time derivative of f_t. This is in contrast with total derivatives $\frac{d}{dt}(f_t(x_t)) = \dot{f}_t(x_t) + J_{f_t}(x_t)\dot{x}_t$.

We often write X_0 and X_1 to denote the beginning and end of a random path, with the condition they are distributed as p (source) and q (target). We distinguish them from the actual samples drawn from these distributions, denoted by $X_\varnothing \sim p$ and $X_\star \sim q$. For instance, a straight-line path is written as $X_t = (1-t)X_\varnothing + tX_\star$ with $X_0 = X_\varnothing$ and $X_1 = X_\star$.

A.2 Random Path

Let p, q be arbitrary distributions over $\mathbb{R}^d$ (e.g., $X_\varnothing \sim p$ and $X_\star \sim q$ may not be independent). A **random path from p to q** is a smooth-changing random variable $\{X_t\}_{t\in[0,1]}$ such that $X_0 \sim p$ and $X_1 \sim q$. A path can be any bridge between p, q. For example, the straight line depends on the target distribution:

$$X_t = (1-t)X_\varnothing + tX_\star \tag{15}$$

But it is a valid random path since $\dot{X}_t = X_\star - X_\varnothing$ is smooth in t, $X_0 = X_\varnothing \sim p$, and $X_1 = X_\star \sim q$.

A.3 Diffeomorphism

A **diffeomorphism** $\phi : \mathbb{R}^d \to \mathbb{R}^d$ is a smooth invertible mapping, such that its inverse is also smooth. The last property allows us to apply the chain rule on $\phi(\phi^{-1}(x)) = x$ to obtain the Jacobian relationship (consistent with the inverse function theorem)

$$J_\phi(\phi^{-1}(x))J_{\phi^{-1}}(x) = I_d \qquad \forall x \in \mathbb{R}^d \tag{16}$$

which guarantees that the Jacobian (of ϕ and ϕ^{-1}) is never singular. In contrast, a function which is just smooth and invertible may have a singular Jacobian (e.g., $f(x) = x^3$ at 0, it fails to be a diffeomorphism because $f^{-1}(x) = x^{1/3}$ is not differentiable at 0).

A.4 Flow

A **flow** is a smooth-changing diffeomorphism $\{\phi_t : \mathbb{R}^d \to \mathbb{R}^d\}_{t\in[0,1]}$ with $\phi_0(x) = x$. A flow then naturally generates a smooth-changing random path from p to q by

$$X_t = \phi_t(X_\varnothing) \tag{17}$$

where $X_0 = \phi_0(X_\varnothing) = X_\varnothing \sim p$ and the corresponding target density is whatever $q = \text{Law}(X_1 = \phi_1(X_\varnothing))$. But clearly not every path from p to q can be generated by a flow. For instance, the straight line $X_t = (1-t)X_\varnothing + tX_\star$ (15) is not directly generatable by a flow as that would imply $\text{Var}(X_t|X_\varnothing) = 0$ for $t > 0$ (false if $X_\star$ is non-degenerate and independent of $X_\varnothing$).

A.4.1 Velocity

A "dual" view of a flow $\{\phi_t : \mathbb{R}^d \to \mathbb{R}^d\}_{t\in[0,1]}$ is the **velocity** $\{u_t : \mathbb{R}^d \to \mathbb{R}^d\}_{t\in[0,1]}$ defined by

$$u_t(x) = \dot{\phi}_t(\phi_t^{-1}(x)) \tag{18}$$

This is the rate of change at any $x \in \mathbb{R}^d$ generated by the flow at time t. Conversely, under standard regularity conditions, a smooth-changing vector field $\{u_t : \mathbb{R}^d \to \mathbb{R}^d\}_{t\in[0,1]}$ corresponds to the velocity of some flow. We can sample from the flow by finding $\{X_t\}_{t\in[0,1]}$ such that $\dot{X}_t = u_t(X_t)$ ("solve the ODE"); this solution is unique if $u_t(x)$ is Lipschitz in x. A simple numerical algorithm (Euler's method) is set $x_0 \leftarrow X_\varnothing \sim p$ and compute

$$x_{t+\Delta t} \leftarrow x_t + \Delta t \times u_t(x_t)$$

for some small interval (e.g., $\Delta t = 1/T$). The correctness of the algorithm can be seen from the fact that $u_t(x_t) = \frac{x_{t+\Delta t} - x_t}{\Delta t} \to \dot{x}_t$ as $\Delta t \to 0$.

A.5 Flow Matching

Let $\{p_t : \mathbb{R}^d \to \mathbb{R}_{\geq 0}\}_{t\in[0,1]}$ denote the marginal density of an arbitrary random path $\{X_t\}_{t\in[0,1]}$. We saw that X_t need not have even been generated by a flow (Section A.4). But

Can we nonetheless find a matching flow ϕ_t in density?

Specifically, let $Y_t = \phi_t(X_0)$ denote the flow path from $X_0 \sim p_0$. Its marginal density $\hat{p}_t$ is then given in the usual closed form (e.g., see Appendix A of this note):

$$\hat{p}_t(x) = p_0\left(\phi_t^{-1}(x)\right)\left|\det\left(J_{\phi_t^{-1}}(x)\right)\right| \tag{19}$$

The determinant is always positive so we may drop the absolute value.[4] (19) is denoted by the "pushforward" operation $\hat{p}_t = (\phi_t)_{\#}p_0$. So the question becomes: can we find a flow ϕ_t that matches $\hat{p}_t = p_t$?

Divergence Before answering the question, it is useful to introduce the divergence notation:

$$\nabla \cdot u_t(x) := \sum_{i=1}^{d} \frac{\partial (u_t(x))_i}{\partial x_i} = \operatorname{tr}\left(J_{u_t}(x)\right) = \left[\frac{d}{dt}\log\left|\det J_{\phi_t}(x_0)\right|\right]_{x_0=\phi_t^{-1}(x)} \tag{20}$$

where the last equality can be verified.[5] Thus it gives the log-volume expansion rate of the flow at x at time t.

Lemma A.1. Let p_t be the marginal density of a random path X_t. Let ϕ_t be a flow with velocity u_t, generating a path $Y_t = \phi_t(X_0)$ from $X_0 \sim p_0$. If

$$\dot{p}_t(x) + \nabla \cdot (p_t(x)u_t(x)) = 0 \qquad \forall x \in \mathbb{R}^d \tag{21}$$

then $Y_t \sim p_t$.

Proof. (21) is equivalent to $\dot{p}_t(x) + \nabla p_t(x)^\top u_t(x) = -p_t(x)\nabla \cdot u_t(x)$ (easily verifiable by the product rule). Observe

$$\begin{aligned}
\frac{d}{dt}\log p_t(Y_t) &= \frac{1}{p_t(Y_t)}\left(\dot{p}_t(Y_t) + \nabla p_t(Y_t)^\top u_t(Y_t)\right) && \text{(multivariable chain rule using } \dot{Y}_t = u_t(Y_t)) \\
&= -\nabla \cdot u_t(Y_t) && (21) \\
&= -\frac{d}{dt}\log\det J_{\phi_t}(X_0) && (20)
\end{aligned}$$

Thus $\frac{d}{dt}\log(p_t(Y_t)\det J_{\phi_t}(X_0)) = 0$, implying that $p_t(Y_t)\det J_{\phi_t}(X_0)$ is time invariant. In particular,

$$p_t(Y_t)\det J_{\phi_t}(X_0) = p_t(Y_t)\frac{1}{\det J_{\phi_t^{-1}}(Y_t)} = p_0(X_0) \qquad \Leftrightarrow \qquad p_t(Y_t) = p_0(X_0)\det J_{\phi_t^{-1}}(Y_t)$$

thus $p_t = (\phi_t)_{\#}p_0$. □

B Lemmas

Lemma B.1. Let $f : \mathbb{R}^d \to \mathbb{R}$ and $u : \mathbb{R}^d \to \mathbb{R}^d$ be any smooth functions where f is zero outside a bounded region (compactly supported), and $X \sim p$ any smooth density over $\mathbb{R}^d$. Then

$$\mathbf{E}[\nabla f(X)^\top u(X)] = -\int_x f(x)\nabla \cdot (p(x)u(x))dx \tag{22}$$

[4] To see this, note that $\det(J_{\phi_t}(x))$ starts at 1 and changes continuously while remaining nonzero.

[5] $\frac{d}{dt}\log\left|\det J_{\phi_t}(x_0)\right| = \operatorname{tr}\left(J_{\phi_t}(x_0)^{-1}\left(\frac{d}{dt}J_{\phi_t}(x_0)\right)\right) = \operatorname{tr}\left(J_{\phi_t}(x_0)^{-1}J_{u_t}(\phi_t(x_0))J_{\phi_t}(x_0)\right) = \operatorname{tr}\left(J_{u_t}(\phi_t(x_0))\right)$ where $\frac{d}{dt}J_{\phi_t}(x_0) = J_{u_t}(\phi_t(x_0))J_{\phi_t}(x_0)$ follows by differentiating both sides of $\dot{\phi}_t(x) = u_t(\phi_t(x))$ wrt x at x_0.

Proof.

$$\begin{aligned}\mathbf{E}[\nabla f(X)^\top u(X)] &= \int_x p(x) \sum_i \frac{\partial f(x)}{\partial x_i} u_i(x) dx \\ &= \sum_i \int_x \frac{\partial f(x)}{\partial x_i} h_i(x) dx \qquad\qquad h(x) := p(x)u(x) \in \mathbb{R}^d\end{aligned}$$

The partial derivatives of h have the form that we want. Reduce the inner integral from over $x \in \mathbb{R}^d$ to over $x_i \in \mathbb{R}$:

$$\int_{x \in \mathbb{R}^d} \frac{\partial f(x)}{\partial x_i} h_i(x) dx = \int_{x_{-i} \in \mathbb{R}^{d-1}} \left(\int_{x_i \in \mathbb{R}} \frac{\partial f(x)}{\partial x_i} h_i(x) dx_i \right) dx_{-i}$$

Apply integration by parts $\int_a^b f(x)G(x)dx = F(x)G(x)|_a^b - \int_a^b F(x)g(x)dx$:

$$\int_{x_i \in \mathbb{R}} \frac{\partial f(x)}{\partial x_i} h_i(x) dx_i = f(x)h_i(x)|_{x_i=-\infty}^{x_i=\infty} - \int_{x_i \in \mathbb{R}} f(x) \frac{\partial p(x)u_i(x)}{\partial x_i} dx_i$$

The first term vanishes at the end points by the compactness of f. Plugging back in,

$$\mathbf{E}[\nabla f(X)^\top u(X)] = -\sum_i \int_x f(x) \frac{\partial p(x)u_i(x)}{\partial x_i} dx = -\int_x f(x) \nabla \cdot (p(x)u(x)) dx$$

□

Lemma B.2. Let $X_t \sim p_t$ be a random path. Define a velocity vector field $u_t : \mathbb{R}^d \to \mathbb{R}^d$ by

$$u_t(x) = \mathbf{E}\left[\dot{X}_t \middle| X_t = x\right]$$

Assume that u_t is regular enough to generate a flow $\dot{Y}_t = u_t(Y_t)$ from $Y_0 \sim p_0$. Then $Y_t \sim p_t$.

Proof. Let $f : \mathbb{R}^d \to \mathbb{R}$ be any smooth compactly supported test function. Then

$$\frac{d}{dt}\mathbf{E}[f(X_t)] = \mathbf{E}\left[\nabla f(X_t)^\top \dot{X}_t\right] = \mathbf{E}\left[\nabla f(X_t)^\top u_t(X_t)\right]$$

The first term is $\frac{d}{dt}\int_x p_t(x)f(x)dx = \int_x f(x)\dot{p}_t(x)dx$ whereas the last term is $-\int_x f(x)\nabla\cdot(p_t(x)u_t(x))dx$ (Lemma B.1). Thus

$$\int_x f(x) \left(\dot{p}_t(x) + \nabla \cdot (p_t(x)u_t(x))\right) dx = 0$$

Assuming p_t and u_t are smooth, this implies pointwise equality $\dot{p}_t(x) + \nabla \cdot (p_t(x)u_t(x)) = 0$ for all $x \in \mathbb{R}^d$. Thus $Y_t \sim p_t$ by Lemma A.1 □

Lemma B.3. Let $X_t \sim p_t$ be a random path. Define a velocity vector field $u_t : \mathbb{R}^d \to \mathbb{R}^d$ by $u_t(x) = \mathbf{E}[\dot{X}_t | X_t = x]$. Let $\{Y_t \in \mathbb{R}^d\}_{t\in[0,1]}$ be a path satisfying $Y_0 \sim p_0$ and (7), repeated here:

$$dY_t = \left(u_t(Y_t) + \frac{w_t}{2}\nabla_x \log p_t(x)|_{x=Y_t}\right)dt + \sqrt{w_t}dW_t \tag{23}$$

for some noise schedule $w_t \geq 0$; $W_t \in \mathbb{R}^d$ is the Wiener process. Then $Y_t \sim p_t$ under standard assumptions.

Proof sketch. By the Fokker-Planck equation, the density of Y_t satisfying (23) evolves as $\dot{\rho}_t = -\nabla \cdot (\rho_t u_t) - \nabla \cdot (\frac{w_t}{2}\rho_t \nabla \log p_t) + \frac{w_t}{2}\Delta\rho_t$, and the solution is unique under standard assumptions. Now we check $\rho_t = p_t$ is a solution. First, both ρ_t and p_t start at p_0; p_t by definition, and $\rho_0 = p_0$ since $Y_0 \sim p_0$ by premise. Next, $p_t \nabla \log p_t = \nabla p_t$ so the last two terms cancel, yielding $\dot{p}_t = -\nabla \cdot (p_t u_t)$. But this is the defining property of $u_t(x) = \mathbf{E}[\dot{X}_t | X_t = x]$ (Lemma B.2). □

Lemma B.4. Let $p_t(x) = \mathbf{E}_{X_\star \sim q}[p_t(x|X_\star)]$ where $p_t(x|z) = \mathcal{N}(tz, (1-t)^2 I_d)$ be the density of $X_t = (1-t)X_\varnothing + tX_\star$ in (1). Then

$$\nabla_x \log p_t(x) = \frac{tu_t(x) - x}{1-t} = \frac{t\tau_t(x) - x}{(1-t)^2} = \frac{-\epsilon_t(x)}{1-t} \tag{24}$$

where $(u_t(x), \tau_t(x), \epsilon_t(x))$ are perfect predictors of (velocity, target, noise) given $X_t = x$.

Proof. Since

$$\nabla_x p_t(x) = \int q(z) \nabla_x p_t(x|z) dz = \int q(z) p_t(x|z) \nabla_x \log p_t(x|z) dz$$

we have

$$\nabla_x \log p_t(x) = \frac{\nabla_x p_t(x)}{p_t(x)} = \int \frac{q(z)p_t(x|z)}{p_t(x)} \nabla_x \log p_t(x|z) dz = \mathbf{E}[\nabla_x \log p_t(x|X_\star)|X_t = x]$$

Using the log Gaussian gradient and the optimality of $(u_t(x), \tau_t(x), \epsilon_t(x))$

$$\mathbf{E}[\nabla_x \log p_t(x|X_\star)|X_t = x] = \mathbf{E}\left[-\frac{x - tX_\star}{(1-t)^2}\middle| X_t = x\right] = \frac{t\tau_t(x) - x}{(1-t)^2}$$

Other expressions follow by the relation $u_t(x) = \frac{\tau_t(x) - x}{1-t} = \frac{x - \epsilon_t(x)}{t}$. □