Reading Seminar, Flow Matching

Flow Matching Guide and Code, Yaron Lipman et al.

Jonathan Ma

March 6, 2025

University of Delaware

Table of contents

Introduction

Flow Matching

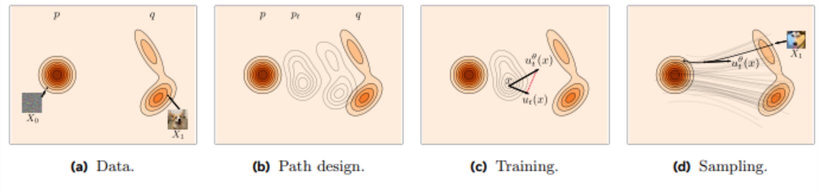


Figure 1: Flow Matching blueprint, described by Lipman

Goal

Given samples $X_0 \sim p$ from a source distribution, build a model to sample $X_1 \sim q$, for some target distribution q.

Preliminary Knowledge

Assumptions

1. Consider data in the d-dimensional Euclidean space $\mathbb{R}^d$, with the standard Euclidean inner product and norm.
2. We will consider random variables $X \in \mathbb{R}^d$.
3. We will omit integration interval when integrating over the whole space, that is, $\int \equiv \int_{\mathbb{R}^d}$.

Diffeomorphisms

Denote by $C^r(\mathbb{R}^m, \mathbb{R}^n)$ the space of functions $f : \mathbb{R}^m \to \mathbb{R}^n$ with continuous partial derivatives of order r.

Definition (Diffeomorphism)

A function $\psi \in C^r(\mathbb{R}^m, \mathbb{R}^n)$ is called a diffeomorphism if its inverse $\psi^{-1} \in C^r(\mathbb{R}^n, \mathbb{R}^m)$.

Given a RV $X \sim p_X$ (with density p_X), consider a RV $Y = \psi(X)$, where $\psi : \mathbb{R}^d \to \mathbb{R}^d$ is a C^1 diffeomorphism. The PDF of Y is called the push-forward of p_X, which we can compute as

$$p_Y(y) = p_X(\psi^{-1}(y))|\det \partial_y \psi^{-1}(y)|.$$

We will denote the push forward operator with the symbol $\sharp$, that is

$$[\psi_\sharp p_X](y) := p_X(\psi^{-1}(y))|\det \partial_y \psi^{-1}(y)|.$$

Flow Models

Flow Models

Definition (Flow)

A C^r flow is a time-dependent mapping $\psi : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ such that $\psi : (t, x) \mapsto \psi_t(x)$, and $\psi \in C^r([0, 1] \times \mathbb{R}^d, \mathbb{R}^d)$, with each ψ_t being a C^r diffeomorphism in x, for all $t \in [0, 1]$.

Definition (Flow Model)

A flow model is a continuous-time Markov process $(X_t)_{0 \le t \le 1}$ defined by applying a flow ψ_t to the RV X_0, such that

$$X_t = \psi_t(X_0), \quad t \in [0, 1], \text{ where } X_0 \sim p.$$

Flow and Velocity Field Correspondence

We claim a C^r flow ψ can be defined in terms of a $C^r([0,1] \times \mathbb{R}^d, \mathbb{R}^d)$ velocity field $u : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ satisfying $u : (t,x) \mapsto u_t(x)$ via the following ODE:

$$\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)), \quad \psi_0(x) = x.$$

Theorem (Flow local existence and uniqueness)

If u is $C^r([0,1] \times \mathbb{R}^d, \mathbb{R}^d)$, $r \geq 1$ (in particular, locally Lipschitz), then the ODE described above has a unique solution which is a $C^r(\Omega, \mathbb{R}^d)$ diffeomorphism $\psi_t(x)$ defined over an open set Ω, which is a super-set of $\{0\} \times \mathbb{R}^d$.

Flow and Velocity Field Correspondence

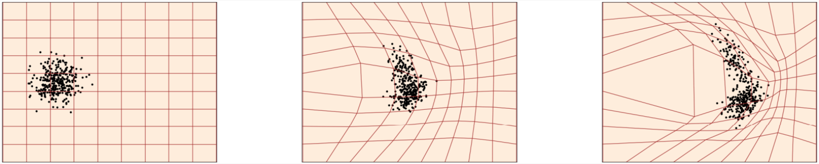


Figure 2: A flow model $X_t = \psi_t(X_0)$ is defined by a diffeomorphism $\psi_t : \mathbb{R}^d \to \mathbb{R}^d$ (visualized with a brown square grid) pushing samples from a source RV X_0 (left, black points) toward some target distribution q (right).

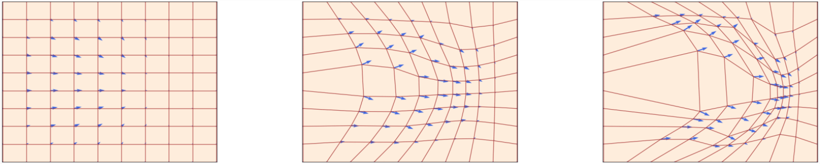


Figure 3: A flow $\psi_t : \mathbb{R}^d \to \mathbb{R}^d$ (square grid) is defined by a velocity field $u_t : \mathbb{R}^d \to \mathbb{R}^d$ (visualized with blue arrows) that prescribes its instantaneous movements at all locations.

Computing Target Samples From Source Samples

To compute a target sample X_1, or in general, any sample X_t, we approximate the solution of the ODE described on slide 7.

- Euler Method
- Midpoint Method

Euler Method

In the Euler method, we use the update rule

$$X_{t+h} = X_t + n^{-1} u_t(X_t),$$

where $n \in \mathbb{Z}^+$ is a step-size hyperparameter.

- To draw a sample X_1 from the target distribution, apply the Euler method starting at some $X_0 \sim p$, to produce the sequence $X_h, X_{2h}, \ldots, X_1$.
- Similarly for method such as the midpoint method.

Tools for Flow Matching

Why Flow Matching?

- Flow modeling requires simulation of ODE.
- Can we avoid this simulation?
- Yes! We will need some strong mathematical tools, and a carefully chosen probability path (p_t).

Definition (Probability Path)

We call a time-dependent probability distribution $(p_t)_{0\leq t\leq 1}$ a probability path.

Probability Paths

An important probability path is the marginal PDF of a flow model $X_t = \psi_t(X_0)$ at time t:

$$X_t \sim p_t.$$

At each time $t \in [0, 1]$, we claim these marginal PDFs are obtained via the push-forward formula

$$p_t(x) = [\psi_{t\sharp p}](x).$$

Definition

Given some arbitrary probability path p_t, we say a velocity field u_t generates [1] p_t if $X_t = \psi_t(X_0) \sim p_t$, for all $t \in [0, 1)$.

[1] Recall that velocity fields correspond with flow models.

Continuity Equation

We can verify that a velocity field u_t generates a probability path p_t by verifying that the pair (u_t, p_t) satisfies a PDE known as the continuity equation:

$$\frac{d}{dt}p_t(x) + \mathrm{div}(p_t u_t)(x) = 0.$$

Theorem (Mass Conservation)

Let (p_t) be a probability path, and u_t a locally Lipschitz integrable vector field. Then TFAE:

1. *The continuity equation above holds, for $t \in [0, 1)$.*
2. *The vector field u_t generates p_t.*

Log-Likelihood Estimation

Flows allow tractable computation of exact likelihoods $\log p_1(x)$, for all $x \in \mathbb{R}^d$. By using the continuity equation, and some computation, we can find that by solving the ODE

$$\frac{d}{dt}\begin{bmatrix} f(t) \\ g(t) \end{bmatrix} = \begin{bmatrix} u_t(f(t)) \\ -\operatorname{tr}(Z^T \partial_x u_t(f(t))Z \end{bmatrix}, \quad \begin{bmatrix} f(1) \\ g(1) \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix},$$

backwards in time from $t = 1$ to $t = 0$, we can compute an unbiased log-likelihood estimate [2]

$$\widehat{\log p_1(x)} = \log p_0(f(0)) - g(0).$$

[2] This work also allows us to train a flow model by maximizing the log-likelihood of training data, and using a KL-divergence loss.

Flow Matching

Flow Matching

We will spend our remaining time building the Flow Matching algorithm. At the end, we will be presented with a simulation-free and scalable approach towards training Flow Models.

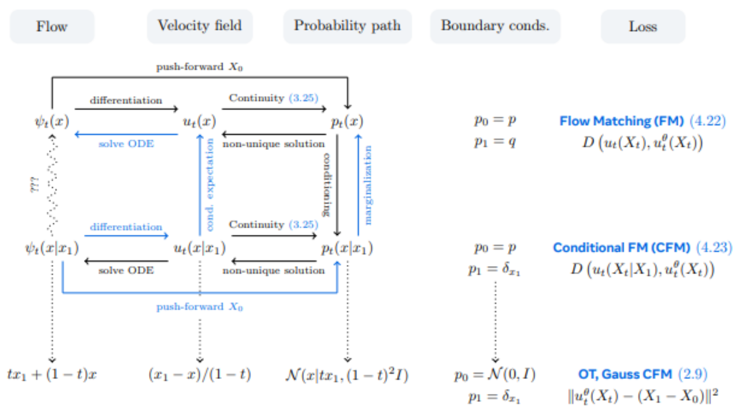


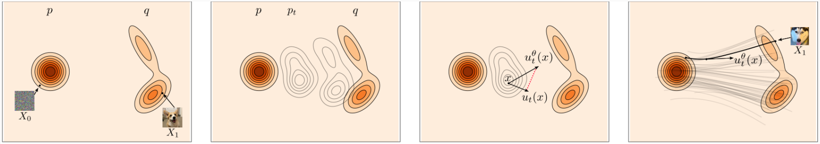
Figure 4: Diagram of FM Algorithm

The Flow Matching Problem

The Flow Matching Problem

Given source distribution p, and target distribution q, find u_t^θ generating p_t, with $p_0 = p$, and $p_1 = q$.

Flow Matching



(a) Data. **(b)** Path design. **(c)** Training. **(d)** Sampling.

Figure 5: The Flow Matching blueprint.

(a) Find a flow mapping samples X_0 from a known source p into samples X_1 from a target data distribution q.

(b) Design a time-continuous probability path $(p_t)_{0 \leq t \leq 1}$ interpolating between $p := p_0$ and $q := p_1$.

(c) During training, use regression to estimate the velocity field u_t known to generate p_t.

(d) Draw a new target sample $X_1 \sim q$ by integrating the estimated velocity field $u_t^\theta(X_t)$ from $t = 0$ to $t = 1$, where $X_0 \sim p$ is a new source sample.

Coupling

Source samples $X_0 \sim p$, and target samples $X_1 \sim q$ may be independent or they can originate from a general joint distribution known as coupling:

$$(X_0, X_1) \sim \pi_{0,1}(X_0, X_1).$$

If no coupling is known, then we set $\pi_{0,1}(X_0, X_1) = p(X_0)q(X_1)$.

Example

The case of producing images from random Gaussian noise vectors is an example of independent source-target distribution.

Example

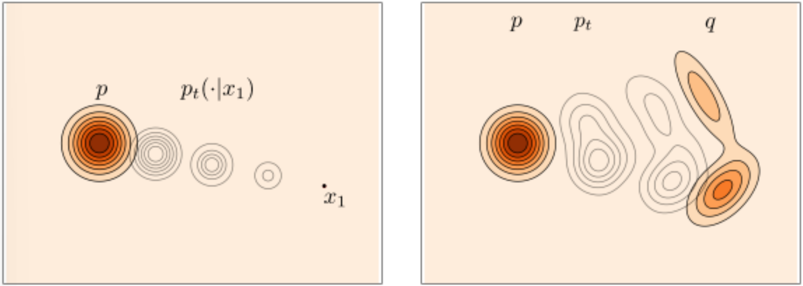
The production of high-resolution images from their low resolution versions, or producing colorized versions from their gray-scale counterparts are examples of dependent couplings.

Designing a Probability Path - An Example

- Flow Matching adopts a conditional strategy for designing probability paths p_t, and corresponding velocity fields u_t. Consider as an example, designing p_t such p_t is conditioned on $X_1 = x_1$, which yields the conditional proabability path $p_{t|1}(x \mid x_1)$.
- We may then construct the overall marginal probability path p_t by aggregating such conditional probability paths[3] $p_{t|1}$:

$$p_t(x) = \int p_{t|1}(x \mid x_1) q(x_1) \, dx_1.$$

[3] Recall $p_1 = q$.



(a) Conditional probability path $p_t(x|x_1)$. **(b)** (Marginal) Probability path $p_t(x)$.

Figure 6: Probability paths described last slide.

Designing a Probablity Path - Boundary Conditions

- To solve the Flow Matching Problem, we would like to design p_t such that the boundary conditions

$$p_0 = p, \quad p_q = q,$$

  are satisfied.
- That is, the marginal probability path p_t interpolates from the source distribution p at $t = 0$, to q at $t = 1$.

Designing a Probablity Path - Boundary Conditions

- We can enforce these boundary conditions by requiring the conditional probability paths to satisfy

$$p_{0|1}(x \mid x_1) = \pi_{0|1}(x \mid x_1), \text{ and } p_{1|1}(x \mid x_1) = \delta_{x_1}(x),$$

  where
  - the conditional coupling $\pi_{0|1}(x_0 \mid x_1) = \pi_{0,1}(x_0, x_1)/q(x_1)$,
  - and δ_{x_1} is the Dirac delta measure centered at x_1.
- For the independent coupling $\pi_{0,1} = p(x_0)q(x_1)$, the first constraint reduces to $p_{0|1}(x \mid x_1) = p(x)$.
- Because the delta measure does not have a density, the second constraint should be read as

$$\int p_{t|1}(x \mid y) f(y)\, dy \to f(x), \text{ as } t \to 1, \text{ for continuous } f.$$

Deriving Generating Velocity Fields

Given a marginal probability path p_t, we now build a velocity field u_t generating p_t.

Claim

The generating velocity field u_t is an average of conditional velocity fields $u_t(x \mid x_1)$, satisfying

$$u_t(\cdot \mid x_1) \text{ generates } p_{t|1}(\cdot \mid x_1).$$

Then the marginal velocity field $u_t(x)$ generating the marginal path $p_t(x)$ is given by averaging the conditional velocity fields $u_t(x \mid x_1)$ across target examples:

$$u_t(x) = \int u_t(x \mid x_1) p_{1|t}(x_1 \mid x) \, dx_1.$$

Deriving Generating Velocity Fields

- We can express the aforementioned equation in known terms using Bayes' rule:

$$p_{1|t}(x_1 \mid x) = \frac{p_{t|1}(x \mid x_1)q(x_1)}{p_t(x)},$$

defined $\forall x, p_t(x) > 0$.

- We can interpret the prev. equation using conditional expectation. If X_t is any random variable such that $X_t \sim p_{t|1}(\cdot \mid X_1)$, then using LOTUS,

$$u_t(x) = \mathbb{E}[u_t(X_t \mid X_1) \mid X_t = x],$$

which is a useful interpretation of $u_t(x)$ as the least-squares approx. to $u_t(X_t \mid X_1)$ given $X_t = x$.

Deriving Generating Velocity Fields - Proof Sketches

- To prove our claimed construction works, we need to show that the described marginal velocity field u_t generates the marginal probability path p_t under some mild assumptions.
- Lipman uses the Mass Conservation Theorem discussed earlier to prove this.

Theorem (Mass Conservation Restated)

Let (p_t) be a probability path, and u_t a locally Lipschitz integrable vector field. Then TFAE:

1. *The continuity equation holds, for $t \in [0, 1)$.*
2. *The vector field u_t generates p_t.*

Deriving Generating Velocity Fields - Proof Sketches

Our mild assuimptions are as follows.

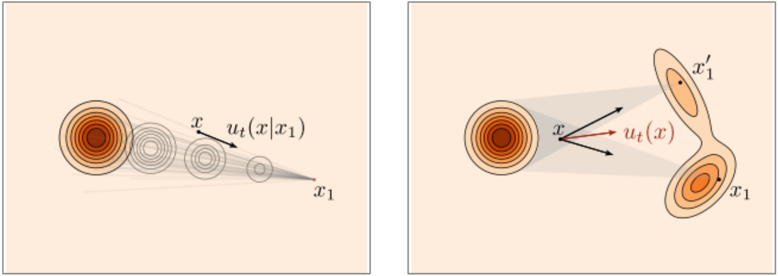
Assumption

Assume $p_{t|Z}(x \mid z)$ is $C^1([0,1) \times \mathbb{R}^d)$ and $u_t(x \mid z)$ is $C^1([0,1) \times \mathbb{R}^d, \mathbb{R}^d)$ as a function of (t,x). Furthermore, suppose p_Z has bounded support, that is, $p_Z(x) = 0$ outside some bounded set in $\mathbb{R}^m$. Finally, assume $p_t(x) > 0$, for all $x \in \mathbb{R}^d$, and $t \in [0,1)$.

Theorem (Marginalization Trick)

Under the previous assumption, if $u_t(x \mid z)$ is conditionally integrable, and generates the conditional probability path $p_t(\cdot \mid z)$, then the marginal velocity field u_t generates the marginal probability path p_t, for all $t \in [0,1)$.

The proof of the above is essentially some calculations and functional analysis to verify the two conditions of Mass Conservation.



(c) Conditional velocity field $u_t(x|x_1)$.

(d) (Marginal) Velocity field $u_t(x)$.

Figure 7: Velocity fields constructed previously.

Flow Matching Loss

We are missing one last ingredient – a tractable loss function to learn a velocity field model u_t^θ as close as possible to the target u_t.

- Directing computing the target u_t is infeasible, since it requires marginalizing over the entire training set.
- We are saved by Bregman divergence loss functions, which provide unbiased gradients to learn u_t^θ in terms of conditional velocities $u_t(x \mid z)$ alone.

Definition

A Bregman divergence measures dissimilarity between two vectors $u, v \in \mathbb{R}^d$ as

$$D(u, v) := \Phi(u) - [\Phi(v) + \langle u - v, \nabla\Phi(v)\rangle],$$

where $\Phi : \mathbb{R}^d \to \mathbb{R}$ is a strictly convex function defined over some convex set $\Omega \subset \mathbb{R}^d$.

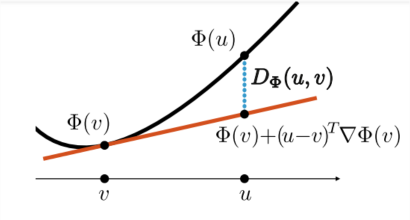


Figure 8: Bregman divergence.

Flow Matching Loss - Bregman Divergences

- Bregman divergence measures the difference between $\Phi(u)$, and the linear approximation to Φ developed around v and evaluated at u.
- Linear approximations are global lower bounds for convex functions $\Longrightarrow D(u, v) \geq 0$.
- Since Φ is strictly convex $\Rightarrow D(u, v) = 0$ iff $u = v$.
- The most basic Bregman divergence is squared Euclidean distance $D(u, v) = \|u - v\|^2$, resulting from choosing $\Phi(u) = \|u\|^2$.
- Importantly for FM, Bregmen divergences have affine invariant gradients w.r.t their second arguments:

$$\nabla_v D(au_1 + bu_2, v) = a\nabla_v D(u_1, v) + b\nabla_v D(u_2, v), \quad \forall a, b \in \mathbb{R}, a + b = 1.$$

Flow Matching Loss - Bregman Divergences

Affine invariance is important in FM, as it allows us to swap expected values with gradients:

$$\nabla_v D(\mathbb{E}[Y], v) = \mathbb{E}[\nabla_v D(Y, v)], \text{ for any RV } Y \in \mathbb{R}^d.$$

Flow Matching Loss - Loss Function

The Flow Matching loss regressing our learnable velocity $u_t^\theta(x)$ onto the target velocity $u_t(x)$ along prob. path p_t is

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t, X_t \sim p_t} D(u_t(X_t), u_t^\theta(X_t)),$$

where $t \sim U[0,1]$. Since u_t is not tractable, we instead use the simpler and tractable Conditional Flow Matching (CFM) loss:

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t, Z, X_t \sim p_{t|Z}(\cdot, Z)} D(u_t(X_t \mid Z), u_t^\theta(X_t)).$$

These two losses are equivalent for learning purposes, because their gradients coincide. (Next slide.)

Flow Matching Loss - Loss Function

Theorem

The gradients of the Flow Matching loss and the Conditional Flow Matching loss coincide:

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

In particular, the minimizer of the CFM loss is the marginal velocity $u_t(x)$.

This theorem falls as a result of direct computation, and the expectation-gradient swapping property of Bregman divergences.

Solving Conditional Generation With Conditional Flows

We have reduced the problem of training a flow model u_t^θ to

1. Find conditional probability paths $p_{t|Z}(x \mid z)$ yielding a marginal probablity path $p_t(x)$ satisfying the boundary conditions $p_0 = p, p_1 = q$.
2. Find conditional velocity fields $u_t(x \mid z)$ generating the conditional probability path.
3. Train using the CFM loss.

Our final discussion in this presentation will show us how to perform steps $1, 2$, yielding our final FM algorithm.

Conditional Flows

Our flexible method to design our conditional probability paths and velocity fields uses conditional flows, described as follows.

1. Define a flow model $X_{t|1}$, satisfying $p_0 = p, p_1 = q$.
2. Extract the velocity field from $X_{t|1}$ by differentiation.

Conditional Flows - Step 1

Define the conditional flow model

$$X_{t|1} = \psi_t(X_0 \mid x_1), \quad \text{where } X_0 \sim \pi_{0|1}(\cdot \mid x_1),$$

and where $\psi : [0,1) \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is a conditional flow defined by

$$\psi_t(x \mid x_t) = \begin{cases} x & t = 0 \\ x_1 & t = 1, \end{cases}$$

is smooth in (t, x) and a diffeomorphism in x.

Conditional Flows - Step 1

Push-Forward Formula Revisited

$$[\psi_\sharp p_X](y) := p_X(\psi^{-1}(y))|\det \partial_y \psi^{-1}(y)|.$$

The push-forward formula discussed earlier defines the probability density of $X_{t|1}$ as

$$p_{t|1}(x \mid x_1) := [\psi_t(\cdot, x_1)_\sharp \pi_{0|1}(\cdot, x_1)](x),$$

although this expression is not used in practical optimization of the CFM loss, and is used only to show that $p_{t|1}$ satisfies $p_0 = p, p_1 = q$.

Conditional Flows - Step 2

We will use the equivalence of flows and velocity fields theorem we showed in the beginning of this presentation.

1. According to the definition of ψ, $\psi_0(\cdot, x_1)$ is the identity map, keeping $\pi_{0|1}(\cdot, x_1)$ intact[4] at time $t = 0$.
2. Note that $\psi_1(\cdot, x_1) = x_1$ is the constant map, concentrating all probability mass at x_1 as $t \to 1$.
3. Note that $\psi_t(\cdot, x_1)$ is a smooth diffeomorphism for $t \in [0, 1]$.

[4] $\pi_{0|1}(x_0 \mid x_1) = \pi_{0,1}(x_0, x_1)/q(x_1)$.

Conditional Flows - Step 2

Therefore, there exists a unique smooth conditional velocity field taking form

$$u_t(x \mid x_1) = \dot{\psi}_t(\psi_t^{-1}(x \mid x_1) \mid x_1).$$

This formula is derived within the proof of the equivalence theorem.

Conditional Flows - Optimal Transport

There are many useful choices for conditional flow $\psi_t(x \mid x_t)$. Motivated by wanting to choose a minimizer of a natural cost functional with desirable properties, we present the dynamic Optimal Transport problem with quadratic cost:

$$
\begin{aligned}
(p_t^*, u_t^*) = \operatorname*{argmin}_{p_t, u_t} \int_0^1 \int \|u_t(x)\|^2 p_t(x)\, dx\, dt, \qquad & \text{(Kinetic Energy)} \\
\text{s.t.} \quad p_0 = p, p_1 = q, \qquad & \text{(Interpolation)} \\
\frac{d}{dt} p_t + \mathrm{div}(p_t u_t) = 0. \qquad & \text{(Continuity Equation)}
\end{aligned}
$$

The (p_t^*, u_t^*) above defines a flow

$$\psi_t^*(x) = t\phi(x) + (1-t)x,$$

called the OT displacement interpolant, where $\phi : \mathbb{R}^d \to \mathbb{R}^d$ is the optimal transport map.

Conditional Flows - Optimal Transport

The OT displacement interpolant solves the FM problem by defining

$$X_t = \psi_t^*(X_0) \sim p_t^* \quad \text{when } X_0 \sim p.$$

We're about out of time, so through some optimization, variational calculus, and computation, we find the minimizer to our OT problem, and our candidate for conditional flow to be

$$\psi_t(x \mid x_1) = tx_1 + (1-t)x.$$

Conclusion

Summary

- Lipman's manuscript provides a very detailed glimpse into the Flow Matching framework [1].
- We presented the intuition and machinery behind Flow Matching.
- Hopefully we will have time to look at the standalone code for Flow Matching, and I also want to point us towards section 2 of Lipman's paper to continue our discussion.

Questions?

Y. Lipman, M. Havasi, P. Holderrieth, N. Shaul, M. Le, B. Karrer, R. T. Q. Chen, D. Lopez-Paz, H. Ben-Hamu, and I. Gat.
Flow matching guide and code, 2024.