Generative Models: Diving into CFM

Diving into Conditional Flow Matching (CFM)

2025-04-11

Rémi Emonet– Quention Bertrand – Mathurin Massias
SenHubIA

UMR ▪ CNRS ▪ 5516 ▪ Saint-Étienne

whoami

- Ensimag (programmation, génie logiciel, logique, optimisation)
- et M2 « Image/Vision/Robotique » (probabilité, raisonnement, système experts)
- Thèse à l'Inria Grenoble en « intelligence ambiente » (génie logiciel)
- Postdoc à Martigny, Suisse en vison par ordinateur et machine learning
- Maître de conférence à l'UJM
  - Enseignant à la Faculté des Sciences (web, prog., prog. pour physique et chimie, machine learning)
  - Chercheur au laboratoire Hubert Curien (machine learning)

Overview

- Introduction
- A quick tour of generative models
- A focus on flow approaches
- Conditional Flow Matching (CFM)
- Optimal Transport and CFM
- Generalization for Generative Models

Introduction

Disclaimer

- I won't show any generated images
- I won't cover advanced approaches
- Focus: understanding the concepts

CFM Blogpost (ICLR Blogpost track) and CFM playground

Generative Modeling[W] = Density Estimation[W]

Given some dataset $\{x_i\}_{i=1}^N$

supposed drawn i.i.d. from an unknown distribution $P(X)$ … or $p(X)$ or $p(X=x)$ or $p(x)$

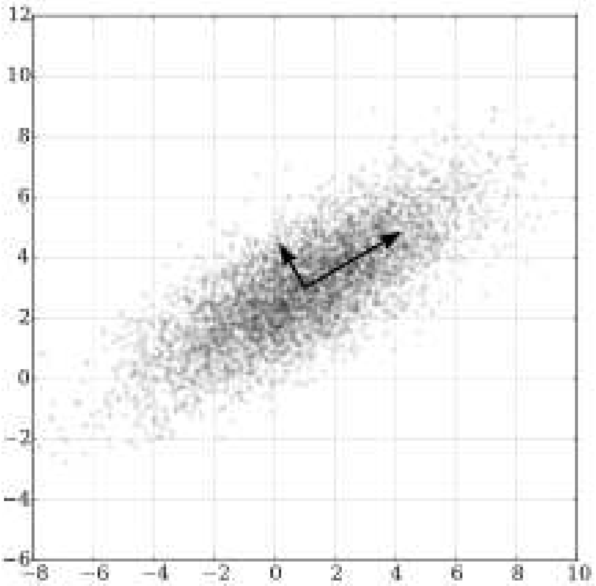
try to recover $p(X)$

Generative Model vs Discriminative Model

- Discriminative: $P(Y|X)$
- Generative: $P(X, Y)$
- Generative: $P(X)$

A quick tour of generative models

Principal Components Analysis[W] (PCA)



Find an orthogonal subspace (lower dimension)
maximizing the captured variance
i.e. minimizing the residual variance
i.e. minimizing the reconstruction error

$$\arg\min_{\{z_i\}_i, W} \sum_i ||x_i - W z_i||_2^2$$

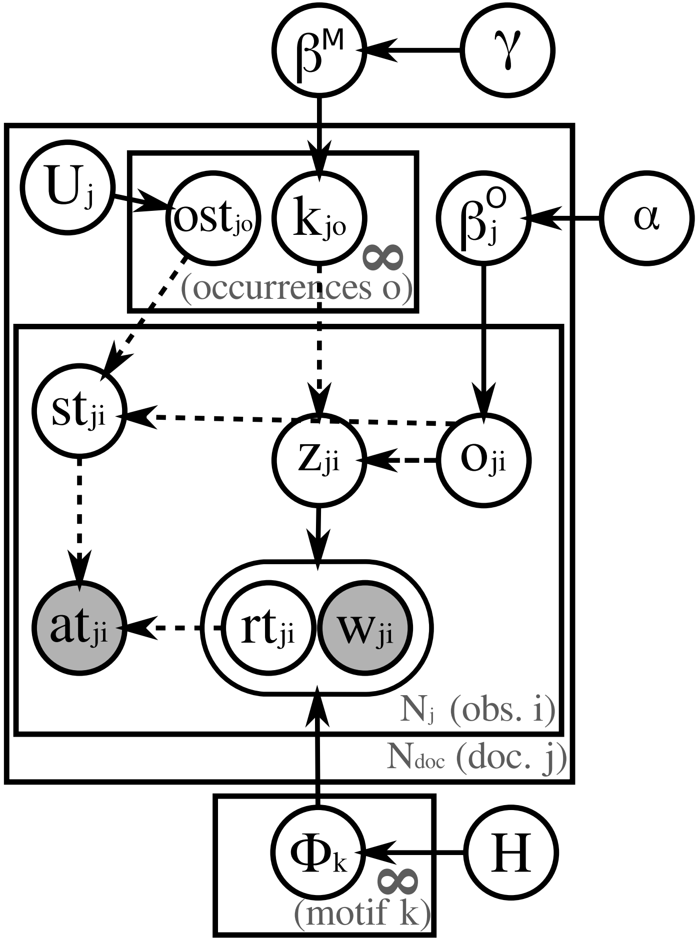
Probabilistic Graphical Models, Bayesian Networks, …

Suppose a structured probability distribution
i.e. a (probablistic) generative story

$$P_\theta(X)$$

based on conditional probabilities and latent variables.

$$P_\theta(X, \ldots) = \prod_{var} P_\theta(var | parents(var))$$



Energy-Based Models[W]

Replace the probability (that would be constrained to be normalized)
by an energy (un-normalized negative log-probability)

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp(-E_\theta(x))$$

$Z(\theta)$: normalization constant, partition function, …

$$Z(\theta) := \int_{x \in X} \exp(-E_\theta(x)) dx$$

Challenge: find a way to work around computing Z

Autoencoders (AE), Variational AE[W] (VAE)

PCA: $\arg\min_{\{z_i\}_i, W} \sum_i ||x_i - W z_i||_2^2$ … ⇒ Reconstruction-error minimizer

Autoencoder (AE): A non-linear version of PCA

- replace $W z_i$ by $Dec_\theta(z_i)$ … no simple projection (W^T) to get $\{z\}_i$
- need to estimate all $\{z_i\}_i$ (as in any Bayesian Network) ⇒ trick: "amortize" (share the cost) by
  - replacing the estimation of all $\{z_i\}_i$
  - by a z_i -guesser… $z_i = Enc_{\theta'}(x_i)$

we have $||x_i - W z_i||_2^2 = -\log(\exp(-||x_i - W z_i||_2^2)) = K - \lambda \cdot \log \mathcal{N}(\mu = W z_i, \sigma = 1)(x_i))$

PCA = Maximum Likelihood Estimator (minimizer of "constant minus log-likelihood")

VAE: A probabilistic version of non-linear PCA… $z_i \sim \mathcal{N}(Enc_{\theta'})$ + prior[1] (maximum a posteriori)

1. usually $\mathcal{N}(0, Id)$ … for **each** z_i , but/so, NO, the distribution of all the $\{z_i\}_i$ taken together is not $\mathcal{N}(0, Id)$ ↩

Generative Adversarial Networks (GANs)

Like a VAE

- still a (small) latent "noise" space
- still a decoder, called generator

But

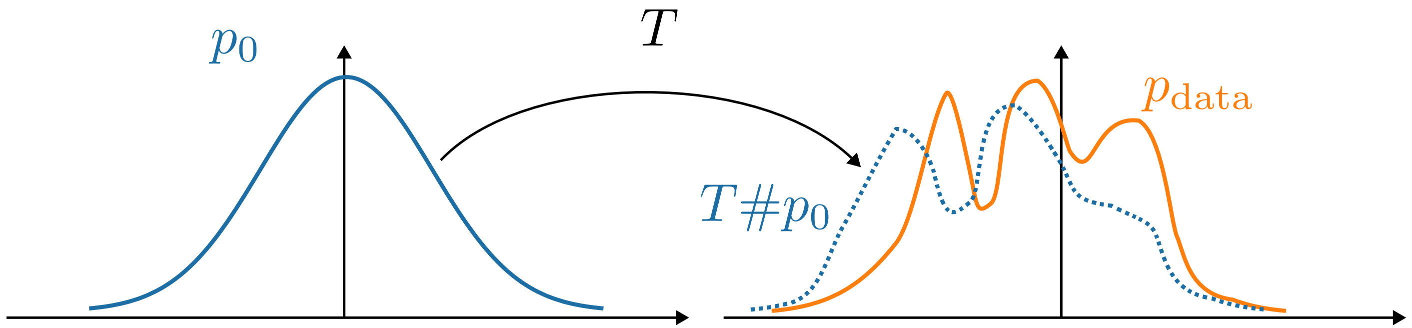
- likelihood free
  - not (explicitly) maximizing the likelihood
  - using a discriminator: estimate the likelihood ratio between real and fake
  - or, in OT, minimizing the Wasserstein distance
    - using a critic: Kantorovich-Rubinstein duality
- a latent representation that is really $\mathcal{N}(0, Id)$

In practice

- VAE + GANs
- VAE + …

A focus on flow approaches

Normalizing Flows



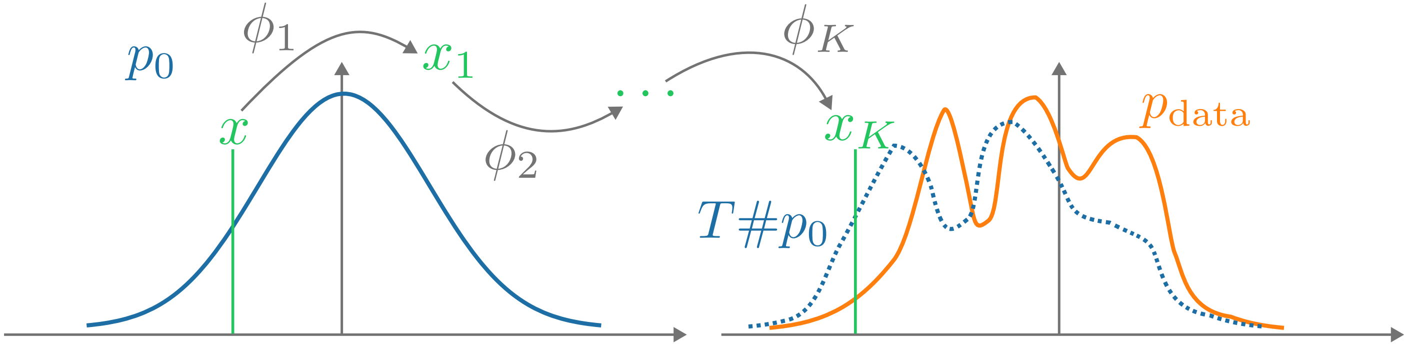
Definition (push-forward): if $x \sim p_0$ then $T(x) \sim T_{\#}p_0$

Normalizing flow (intuition):

- denoting $p_{gen} = T_{\#}p_0$
- locally, if T compresses the space by a factor 42, then $p_{gen}(T(x)) = 42 \cdot p_0(x)$
- formally, change of variable, $p_{gen}(T(x)) = |det(J_{T^{-1}}(x))| \cdot p_0(x)$ *(determinant of the jacobian of* T^{-1}*)*

Principle: parametrize and learn T … so that its inverse exists (and has an easy jacobian det).

Normalizing Flows, with composed functions



Learn a deep T, i.e.,

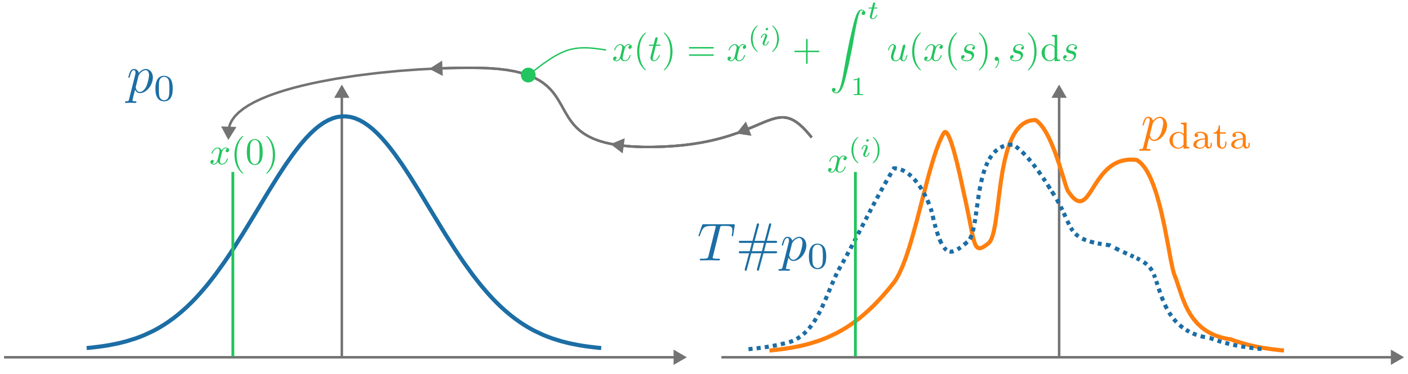
$$T = \phi_1 \circ \phi_2 \circ \ldots \circ \phi_K$$

Chain rule of change of variable,

$$|det(J_{T^{-1}}(x))| = \prod_k |det(J_{\phi_k^{-1}}(x))|$$

Principle: compose invertible blocks (with easy jacobian det)

Continuous Normalizing Flows



Pushing to the limit

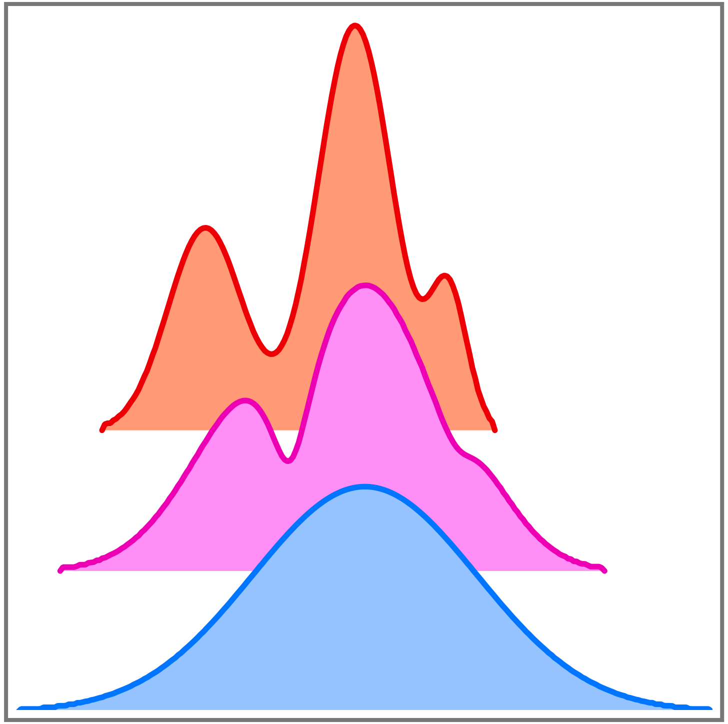
- infinitely many infinitely-small steps
- making depth continuous $k \mapsto t$
- replacing $\phi_k(x)$ by $u(x,t)$, or $u_t(x)$

Continuous Normalizing Flow

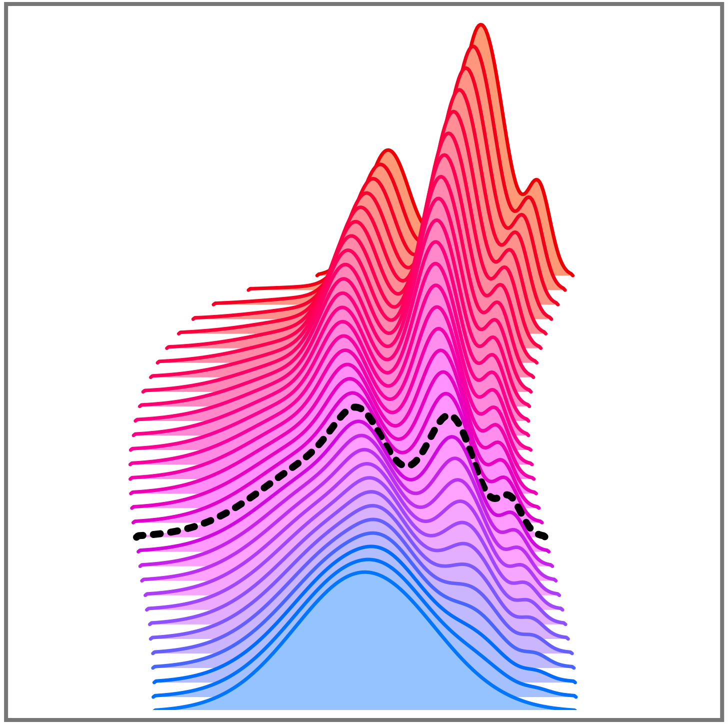
- easier: less constraints on u than ϕ

Forward and reverse ODE

Continuous Normalizing Flows: visual summary

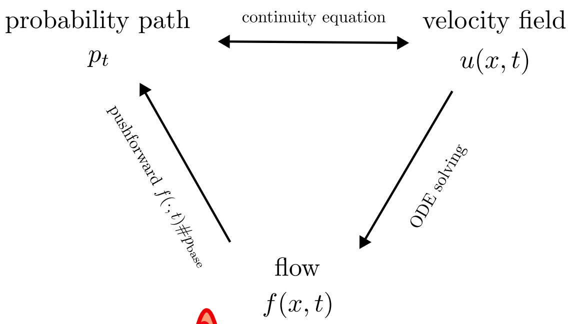


Continuous Normalizing Flows: "limitation"



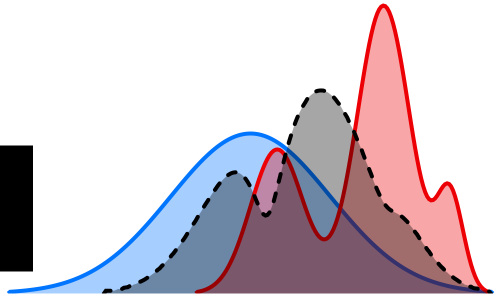
The flow is unspecified!
(there is an infinity of equally good solutions)

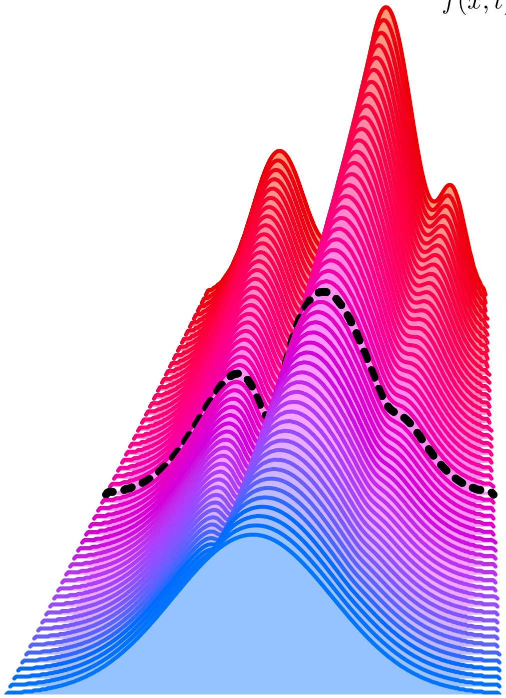
Probability paths, velocity fields (and flows)



Continuity Equation / Transport Equation

$$\partial_t p_t + \nabla \cdot u_t p_t = 0$$





Parenthesis: understanding the continuity equation

Continuity Equation between a **time-varying density** p_t and a **time-varying velocity field** u_t 🖊

$$\partial_t p_t + \nabla \cdot u_t p_t = 0$$

i.e.

$$\begin{aligned}
\partial_t p_t &= -\nabla \cdot u_t p_t && (1)\\
\frac{\partial}{\partial t} p_t &= -\nabla \cdot (u_t p_t) && (2)\\
\text{with, in 2D} & && (3)\\
\frac{\partial}{\partial t} p_t &= -\frac{\partial}{\partial x}(u_t p_t) - \frac{\partial}{\partial y}(u_t p_t) && (4)\\
\frac{\partial p_t}{\partial t}(x,y) &= -\frac{\partial u_t p_t}{\partial x}(x,y) - \frac{\partial u_t p_t}{\partial y}(x,y) && (5)
\end{aligned}$$

Diffusion: denoising diffusion probabilistic models (DDPM)

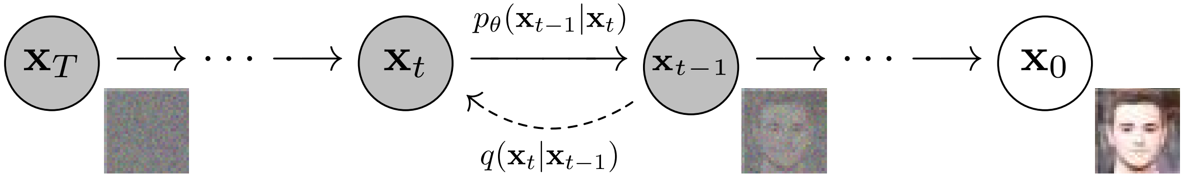


Figure 2: The directed graphical model considered in this work.

Principle

- progressively noise you data
- use that data to learn an infinitesimal denoiser

Actually

- a VAE with successive latent representations[1]
- learning a velocity field (notation trap: $t \in [\![T, 0]\!]$ instead of $t \in [0, 1]$)
- specifying a unique probability path (but stochastic flow)
- effectively supervising at every step (vs CNF)

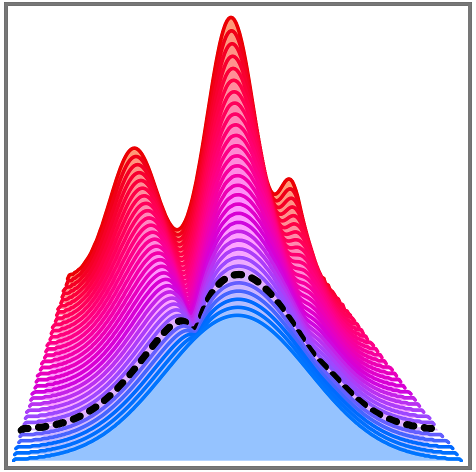
1. … the latent space dimension is the same as the data space dimension (like CNF, contrary to PCA, GAN, VAE) ↩

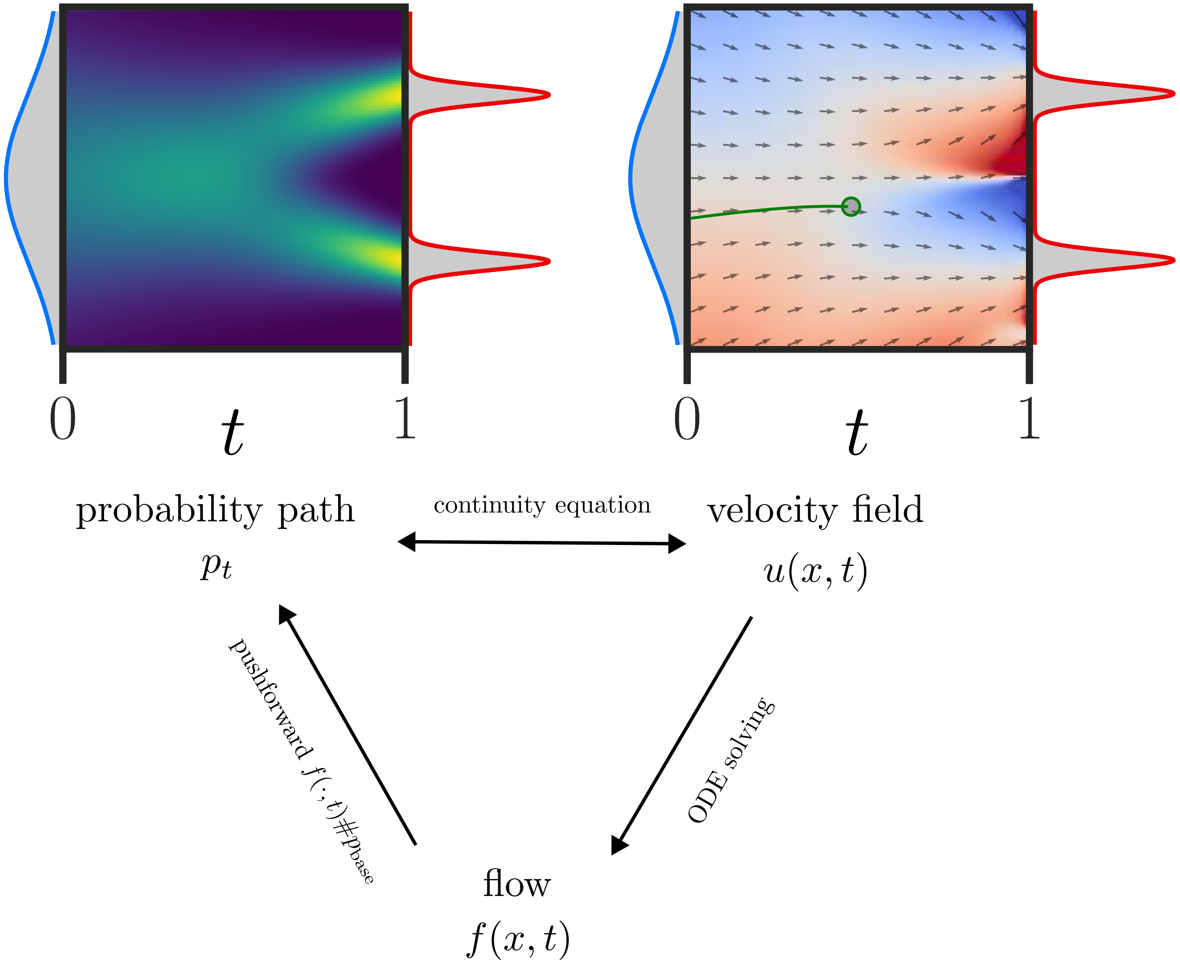
Overview

- Introduction
- A quick tour of generative models
- A focus on flow approaches
- Conditional Flow Matching (CFM)
- Optimal Transport and CFM
- Generalization for Generative Models

Conditional Flow Matching (CFM)

Visuals




0
t
1
0
t
1
probability path
p_t
continuity equation
velocity field
$u(x,t)$
pushforward $f(\cdot,t)\#p_{\mathrm{base}}$
ODE solving
flow
$f(x,t)$

Conditional Flow Matching (CFM) Principles

- Fully specify a probability path / velocity field / flow (like diffusion, unlike CNF)
- Use a ordinary (non-stochastic) differential equation (like CNF, unlike diffusion)

Solution ?

- introduce an arbitrary conditioning variables z
- specify the flow as an aggregation of conditional flows

Before diving into the details, let's look at one algorithm.

Typical CFM algorithm

Design choices

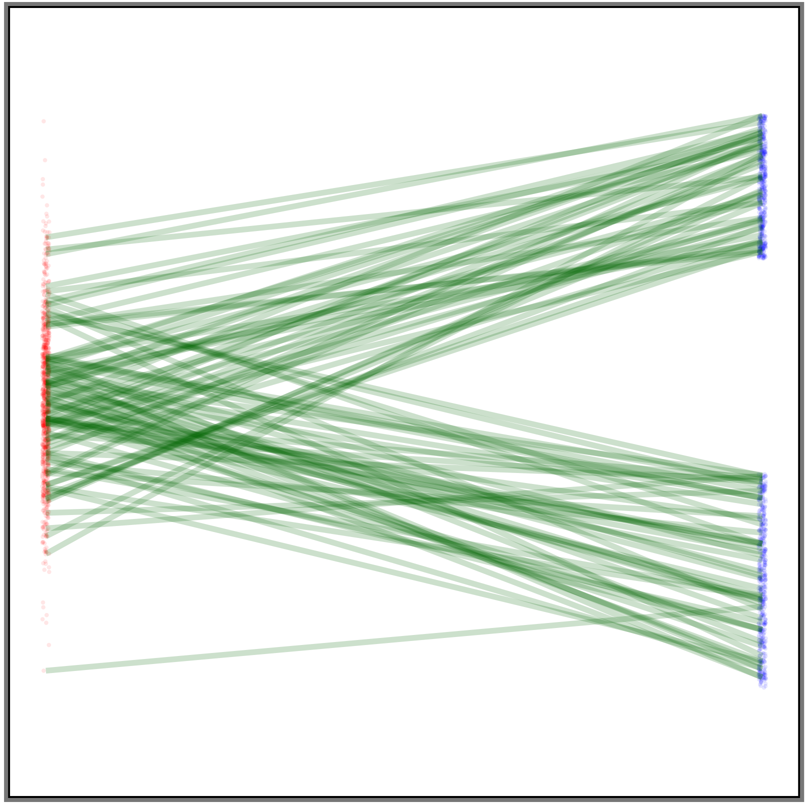
- conditioning variable z is a pair
  - a *source* point, typically from $\mathcal{N}(0,1)$ *(but not necessarily, vs diffusion)*
  - a *target* point, typically form the (training) dataset
- conditional probability path/flow is a straight constant-velocity *(OT between two points)*

Algorithm

$$\begin{aligned} z_0 &\sim \mathcal{N}(0, I) \\ z_1 &\sim Dataset \\ t &\sim Uniform([0,1]) \\ x &= t \cdot z_1 + (1-t) \cdot z_0 \\ \text{SGD step on } \theta \text{ with loss: } & \|u_\theta(x,t) - (z_1 - z_0)\|_2^2 \end{aligned}$$

That's it! *(up to practical hacks and a few days of training)*

CFM: Does it works? the "inversion", path un-mixing



CFM: Design choices

Decide on p_0 , typically $\mathcal{N}(0, I)$ *(but not necessarily, vs diffusion)*

Decide on the conditioning variable (and its distribution), e.g.

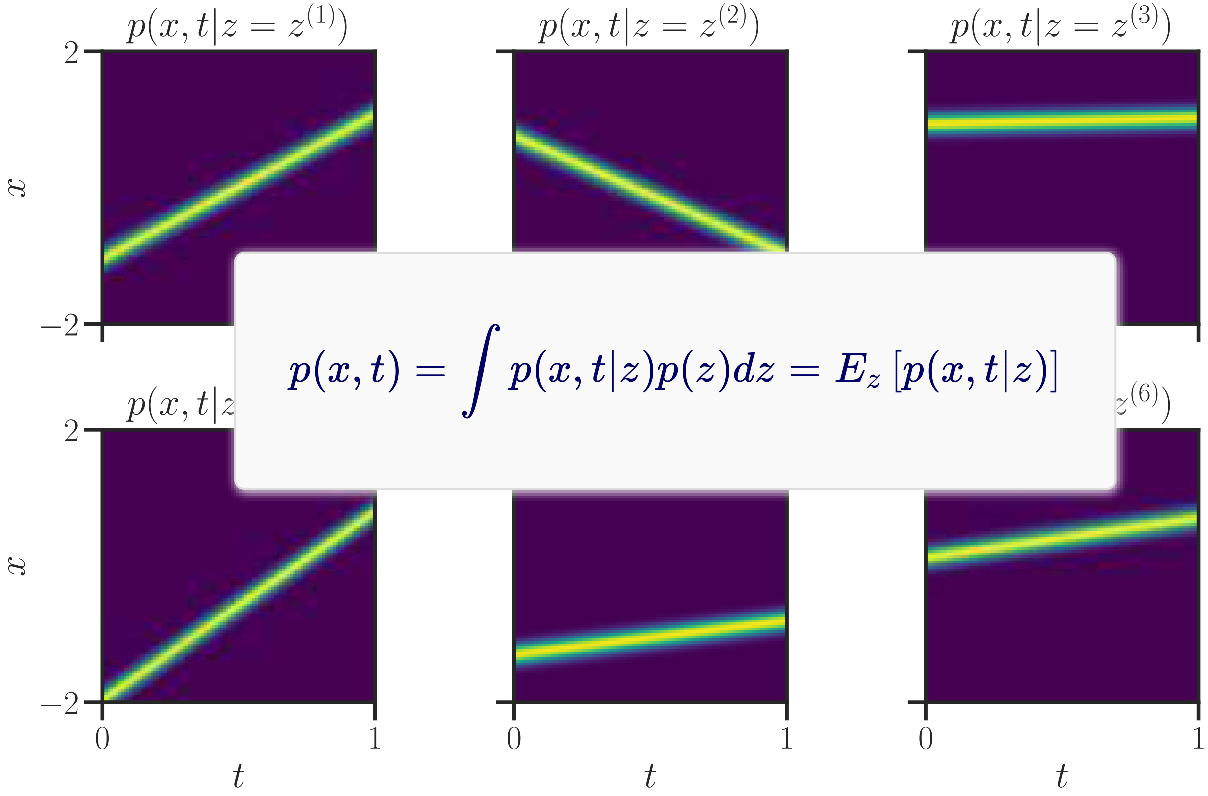
- z is a pair (x_0, x_1)
- z is a target point x_1
- z is a minibatch of source and target
- z is a pair, constrained by some clusters

Decide on the conditional "flow"

- conditional probability path $p_t(x|z)$ (or $p(x, t|z)$)
- and associated velocity field $u^{cond}(x, t)$

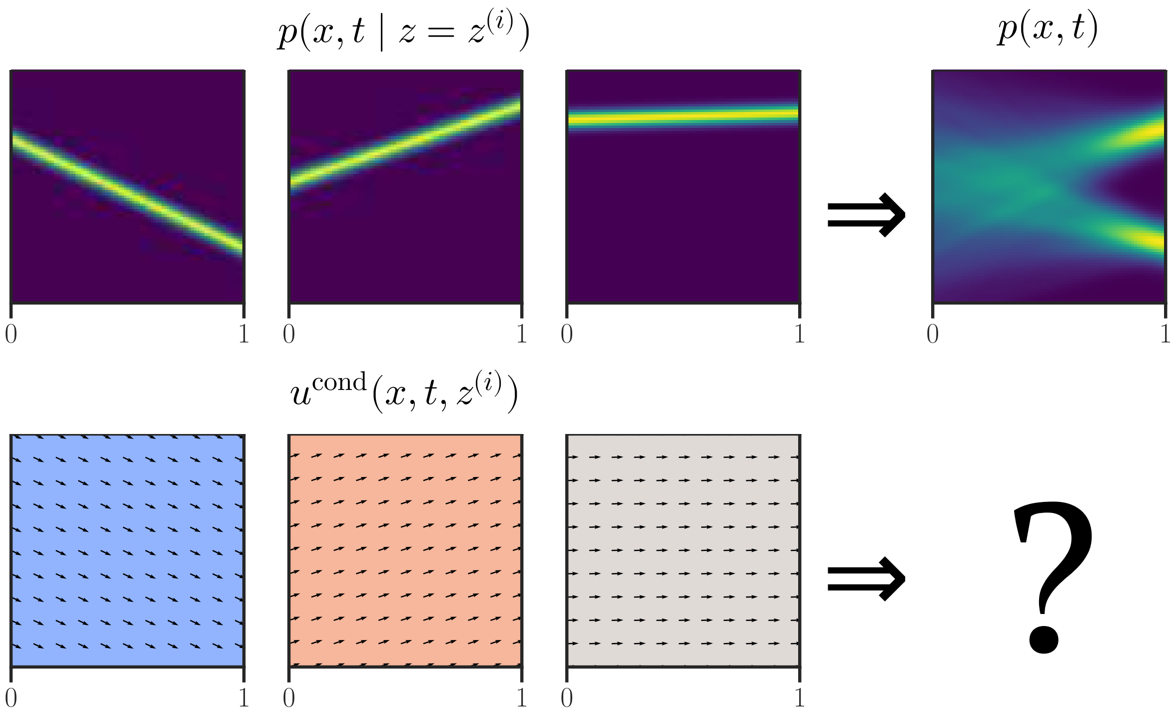
(under marginal constraints, on $p(x, t)$)

CFM: $p(x,t|z)$ (conditional) to $p(x,t)$ is easy



$$p(x,t) = \int p(x,t|z)p(z)dz = E_z\left[p(x,t|z)\right]$$

CFM: $u^{cond}(x,t,z)$ to $u(x,t)$ is less easy



Practical session

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\forall\, \overbrace{z \in Z}^{\textbf{choice 1}} \quad \overbrace{p(x|t,z)}^{\textbf{choice 2}} \arrow[r, leftrightarrow, "\begin{array}{c}\text{determines}\\\text{explicitly}\end{array}", "\text{(by continuity eq.)}"'] \arrow[d, "\begin{array}{c}\text{determines}\\\text{(by expectation over } z\text{)}\end{array}"'] & u^{\text{cond}}(x,t,z) \arrow[d, "\begin{array}{c}\text{allows}\\\text{computing}\\\text{(Thm. 1)}\end{array}"] \\
p(x|t) \arrow[r, leftrightarrow, "\begin{array}{c}\text{determines}\\\text{implicitly}\end{array}", "\text{(by continuity eq.)}"'] & u(x,t)
\end{tikzcd}$$

CFM: Closed form expression (Theorem 1)

$$\forall t, \forall x,$$

$$u(x,t) = E_{z|x,t}[u^{cond}(x,t,z)]$$

(also written as)

$$\forall\, t, \forall\, x,$$

$$u(x,t) = \int_z u^{cond}(x,t,z) p(z|x,t)$$

(or bayes)

$$\forall\, t, \forall\, x,$$

$$u(x,t) = \int_z u^{cond}(x,t,z) \frac{p(x,t|z)p(z)}{p(x,t)} = E_z\left[\frac{u^{cond}(x,t,z)p(x,t|z)}{p(x,t)}\right] = E_z\left[\frac{u^{cond}(x,t,z)p(x,t|z)}{\sum_{z'} p(x,t|z')p(z')}\right]$$

CFM: Some intuition on the loss

$$L_{CFM} = \ldots$$

🖊

Least squares!

CFM playground

R

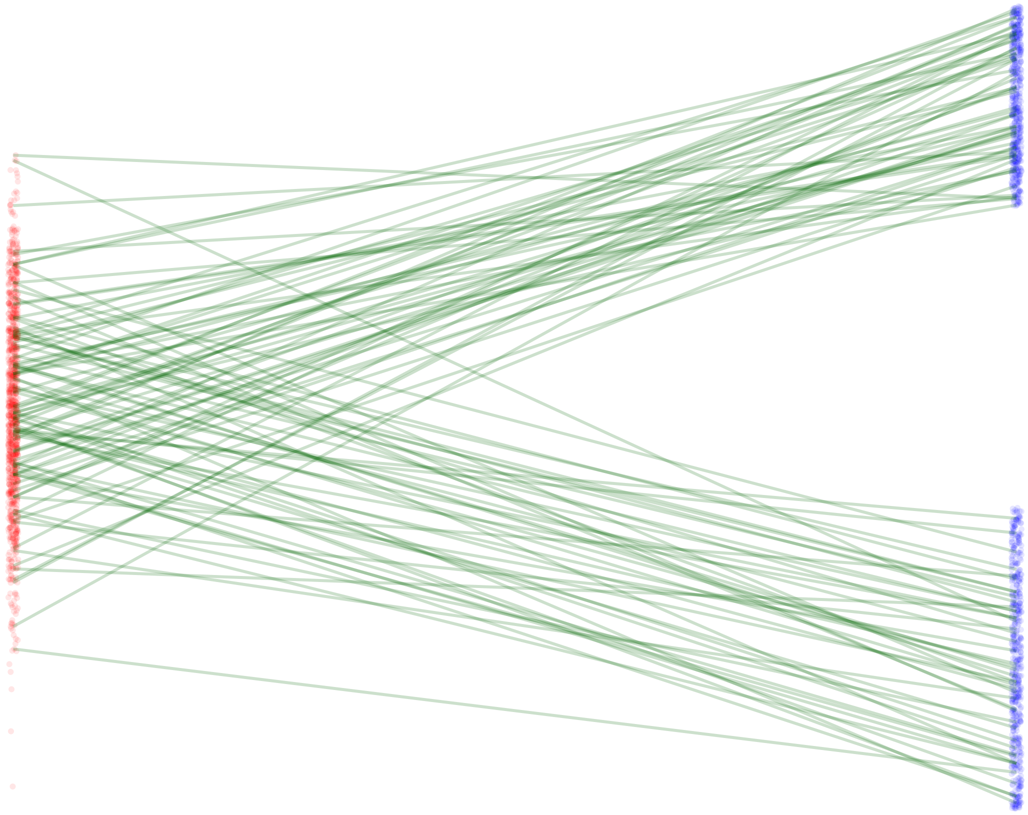
z

p_0

G⇝.
U⇝.
G2⇝.
U2⇝.

p_1

.⇝G
.⇝U
.⇝G2
.⇝U2



z

1
3
*

×
1s
.5s
.2s

Overview

- Introduction
- A quick tour of generative models
- A focus on flow approaches
- Conditional Flow Matching (CFM)
- Optimal Transport and CFM
- Generalization for Generative Models

Optimal Transport and CFM

Links with OT: OT-CFM

"OT-CFM" (e.g. TMLR2024, https://arxiv.org/abs/2302.00482)

- use minibatch OT to create pairs
- less un-mixing to do
- may improve training stability

Intuition: OT pre-unmixes, given also straighter paths

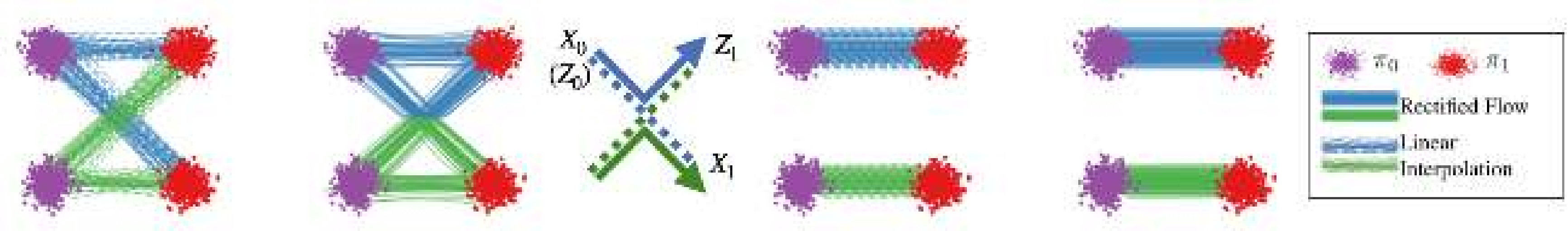
Q: why not do directly OT?

Links with OT: Recified Flow

Rectified Flows (https://arxiv.org/pdf/2209.03003)

- do CFM to unmix
- relearn with the unmixed coupling
- iterate

Intuition: make paths straight, closer to OT, faster to sample



(a) Linear interpolation $X_t = tX_1 + (1-t)X_0$

(b) Rectified flow Z_t induced by (X_0, X_1)

(c) Linear interpolation $Z_t = tZ_1 + (1-t)Z_0$

(d) Rectified flow Z'_t induced by (Z_0, Z_1)

Q: why not do directly OT?

Underlying links between Optimal Transport and CFM

- Kantorovich-Rubinstein duality is $u(x,t)$ without depending on t
- "Everything" points towards making CFM closer to OT
  - less mixing is easier to learn
  - straighter path lead to faster sampling (incl. single-step generation)

Q: why not do directly OT, learning the Monge Map?

Generalization for Generative Models

Modern Generative Models: performance measures

Desired goals

- measure sample quality
- avoid mode collapse and memorization

Typically

- FID (Fréchet inception distance), between generated and training data
  - Wasserstein in some in a feature space
  - with a gaussian approximation of each dataset
- "recall" : coverage of the training set
- "precision" : only generate good (i.e. coverage of gen set by training set)

The goal is missed.

Generalization bounds?

On $L(train) - L(test)$

with $L \in \{KL, W, ...\}$

…

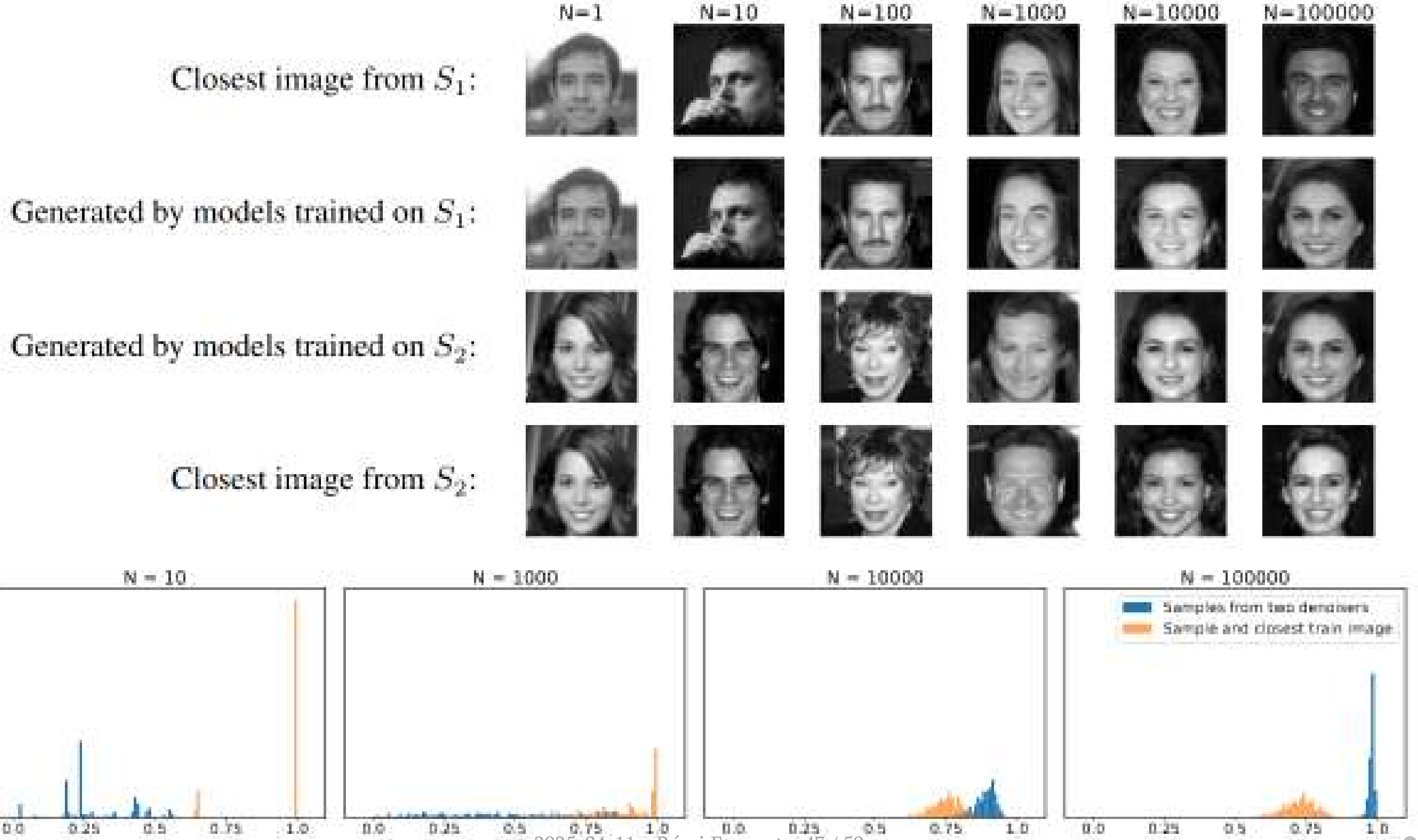
NB

- closed-form solution of CFM says we generate only training points (memorization)
- small gaussian noise present in CFM formulations don't change that

Generalization vs Creativity

(e.g. https://arxiv.org/pdf/2310.02557 next slide)

- ML-type Generalization
  - memorization can be observed with big models and "small" data
  - no memorization with big data
  - it seems, no double descent (better generalization with bigger models, lottery ticket etc)
- Creativity
  - open/ill-posed problem
  - inductive bias
  - for images


N=1
N=10
N=100
N=1000
N=10000
N=100000
Closest image from S_1:
Generated by models trained on S_1:
Generated by models trained on S_2:
Closest image from S_2:
N = 10
N = 1000
N = 10000
N = 100000
Samples from two denoisers
Sample and closest train image
0.0
0.25
0.5
0.75
1.0

Overview

- Introduction
- A quick tour of generative models
- A focus on flow approaches
- Conditional Flow Matching (CFM)
- Optimal Transport and CFM
- Generalization for Generative Models

CFM Blogpost (ICLR Blogpost track) and CFM playground

R

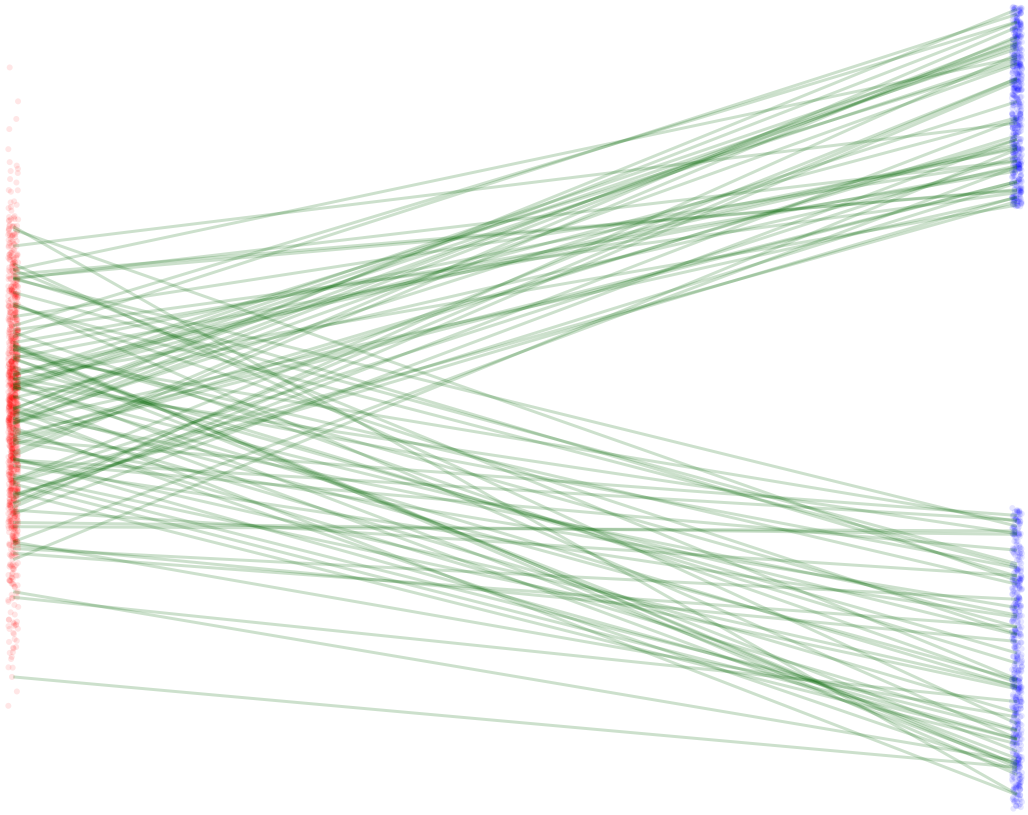
p_0

G⇝.
U⇝.
G2⇝.
U2⇝.

p_1

.⇝G
.⇝U
.⇝G2
.⇝U2

z



z

1
3
*

×
1s
.5s
.2s

Generative Models: Diving into CFM

Diving into Conditional Flow Matching (CFM)

2025-04-11

Rémi Emonet– Quention Bertrand – Mathurin Massias
SenHubIA