

Expectation vs reality: on the role of stochasticity in flow matching generalization

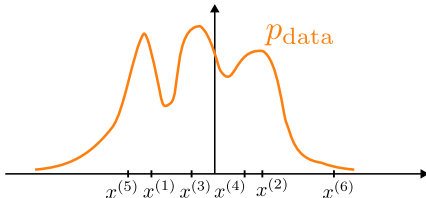
Mathurin Massias

Work with Q. Bertrand, A. Gagneux, S. Martin, and R. Emonet

SHARP+Foundry workshop @COLT, 2025/06/30

Generative modelling

Given $x^{(1)}, \dots, x^{(n)}$ sampled from p_{data} , learn to sample from p_{data}

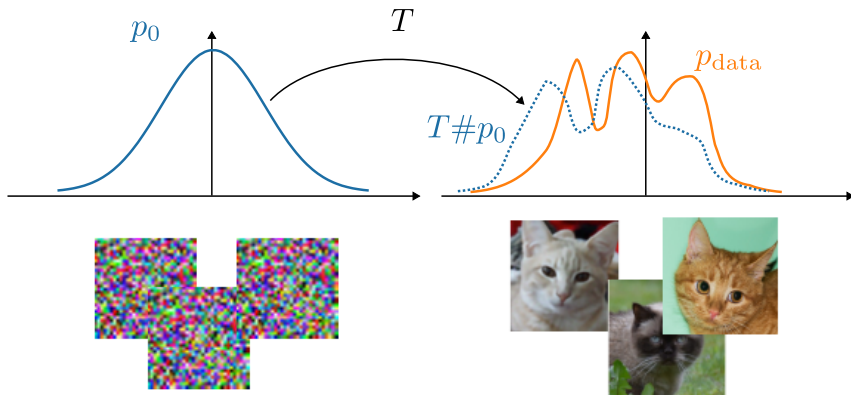



3 challenges:

- enforce fast sampling
- generate high quality samples
- properly cover the diversity of p_{data}

Modern way to do generative modelling

Map simple *base distribution*, p_0 , to p_{data} through a **map** T



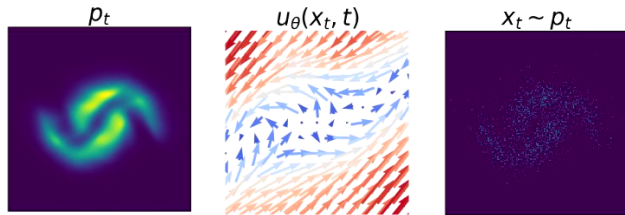
 A Visual Dive into Conditional Flow Matching, Martin, Gagneux, Emonet, Bertrand & Massias, ICLR Blogpost 2025, <https://dl.heeere.com/cfm/>

Continuous normalizing flows (CNF)

The map T is defined implicitly through an ODE:

$$\begin{cases} x(0) = x_0 \\ \dot{x}(t) = u(x(t), t) \quad \forall t \in [0, 1] \end{cases}$$

- $T(x_0) := x(1)$ (ODE solution at time 1)
- learn the *velocity field* u as $u_\theta : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$
- sample by solving the initial value problem with $x(0) = x_0 \sim p_0$



(dynamic animation in [blog post](#))

Framework recap

We have:

- source distribution $p_0 = \mathcal{N}(0, \text{Id})$
- target distribution p_{data} (e.g. realistic images)

We want:

- to generate new samples from p_{data}

How?

- by solving on $[0, 1]$

$$\begin{cases} x(0) = x_0 \\ \dot{x}(t) = u(x(t), t) \quad \forall t \in [0, 1] \end{cases}$$

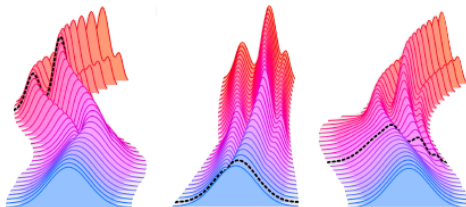
- such that solution $x(1) \sim p_{\text{data}}$ when $x(0) \sim p_0$



how to find a u that works well?

Searching for a good u

$$\begin{cases} x(0) = x_0 \\ \dot{x}(t) = u(x(t), t) \quad \forall t \in [0, 1] \end{cases}$$



- ODE defines *probability path* $(p_t)_{t \in [0,1]}$ = laws of the solution $x(t)$ when $x(0) \sim p_0$
- we want $p_0 = p_0$ and $p_1 = p_{\text{data}}$

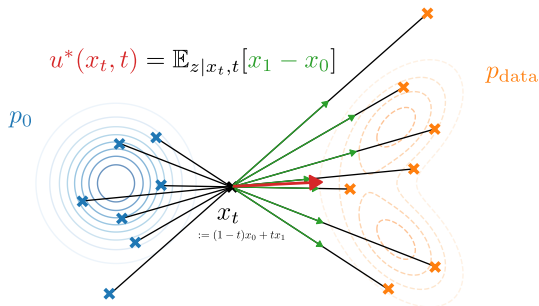
u must drive a progressive transformation of p_0 into p_{data}

Flow matching: building a u

Break down complex problem into small ones:

- introduce conditioning variable $z = (x_0, x_1) \sim p_0 \times p_{\text{data}}$
- define **conditional** probability path $p(\cdot | z = (x_0, x_1), t) = \delta_{(1-t)x_0 + tx_1}$
- we know the associated conditional velocity: $u^{\text{cond}}(x, z = (x_0, x_1), t) = x_1 - x_0$

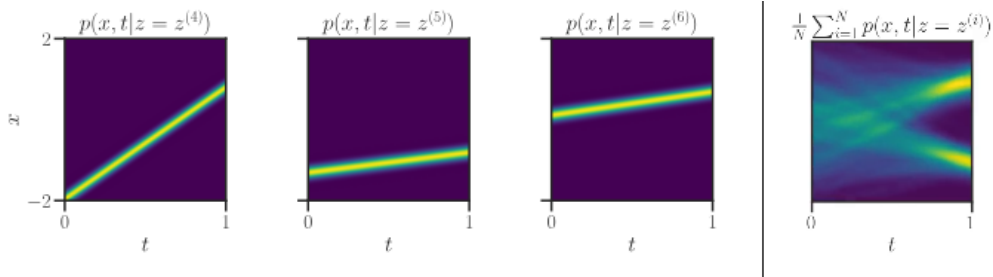
“Uncondition”: define u^* by marginalizing against $z = (x_0, x_1)$:



The magic happens

Theorem 1:

- Averaging the conditional paths give a probability path going from p_0 to p_{data}
- u^* transports p_0 to p_{data}



We are done

We have our target, valid velocity:

$$u^{\star}(x, t) = \mathbb{E}_{z|x, t}[x_1 - x_0]$$

We are done

We have our target, valid velocity:

$$u^*(x, t) = \mathbb{E}_{z|x, t}[x_1 - x_0]$$

We just need to approximate it with a neural net $u_\theta : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$:

$$\min_{\theta} \left\{ \mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{\substack{t \sim \mathcal{U}([0, 1]) \\ x_t \sim p(\cdot|t)}} \|u_\theta(x_t, t) - u^*(x_t, t)\|^2 \right\}$$

We are done

We have our target, valid velocity:

$$u^*(x, t) = \mathbb{E}_{z|x, t}[x_1 - x_0]$$

We just need to approximate it with a neural net $u_\theta : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$:

$$\min_{\theta} \left\{ \mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{\substack{t \sim \mathcal{U}([0, 1]) \\ x_t \sim p(\cdot|t)}} \|u_\theta(x_t, t) - u^*(x_t, t)\|^2 \right\}$$

We are not done at all :(

Theorem 2 to the rescue

Ideal loss:

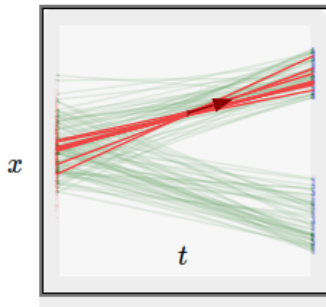
$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{\substack{t \sim \mathcal{U}([0,1]) \\ x_t \sim p(\cdot|t)}} \|u_\theta(x_t, t) - u^\star(x_t, t)\|^2$$

$$u^\star(x, t) = \mathbb{E}_{z|x,t}[x_1 - x_0]$$

Theorem 2: Up to a constant, \mathcal{L}_{FM} is equal to

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \|u_\theta(x_t, t) - (x_1 - x_0)\|^2$$

where $x_t := (1 - t)x_0 + tx_1$



Minimizing \mathcal{L}_{CFM}

To minimize

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \|u_\theta(x_t, t) - u^{\text{cond}}(x_t, z = x_1, t)\|^2$$

$$(x_t := (1 - t)x_0 + tx_1)$$

- sample $x_0 \sim p_0$: easy!
- sample $t \sim \mathcal{U}([0, 1])$: easy!
- sample $x_1 \sim p_{\text{data}}$? easy if we replace by $x_1 \sim \hat{p}_{\text{data}} := \frac{1}{n} \sum_{i=1}^n \delta_{x^{(i)}}$

Training flow matching

$$\min_{\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} [\|u_{\theta}(x_t, t) - u^{\text{cond}}(x_t, z = x_1, t)\|^2] \quad (x_t := (1 - t)x_0 + tx_1)$$

p_0



p_{data}



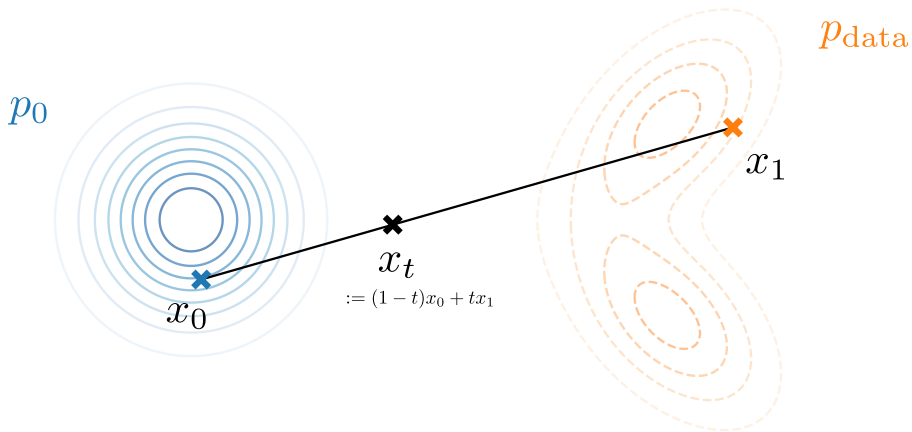
Training flow matching

$$\min_{\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} [\|u_{\theta}(x_t, t) - u^{\text{cond}}(x_t, z = x_1, t)\|^2] \quad (x_t := (1-t)x_0 + tx_1)$$



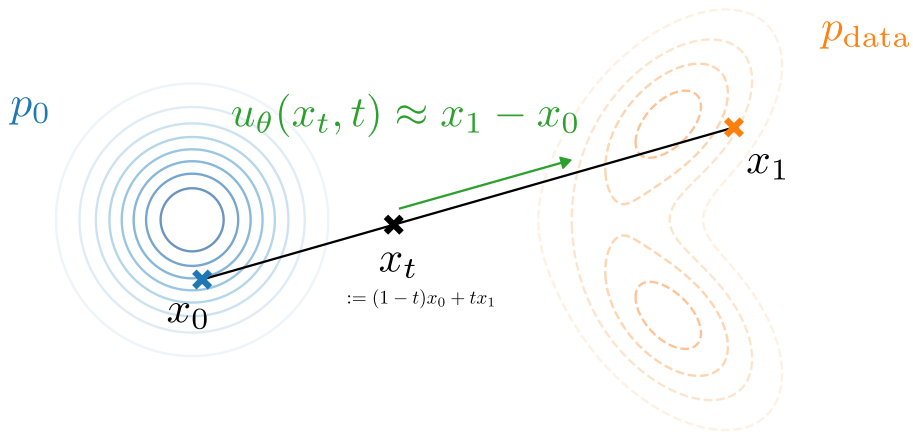
Training flow matching

$$\min_{\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} [\|u_{\theta}(x_t, t) - u^{\text{cond}}(x_t, z = x_1, t)\|^2] \quad (x_t := (1 - t)x_0 + tx_1)$$



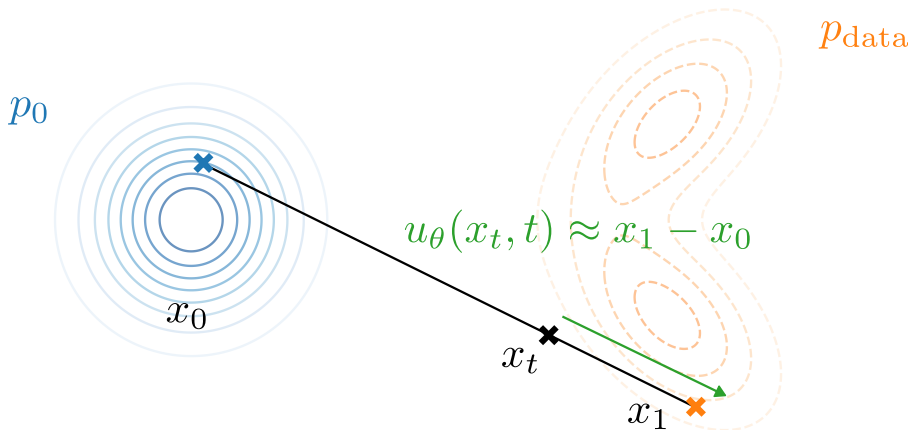
Training flow matching

$$\min_{\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} [\|u_{\theta}(x_t, t) - u^{\text{cond}}(x_t, z = x_1, t)\|^2] \quad (x_t := (1 - t)x_0 + tx_1)$$



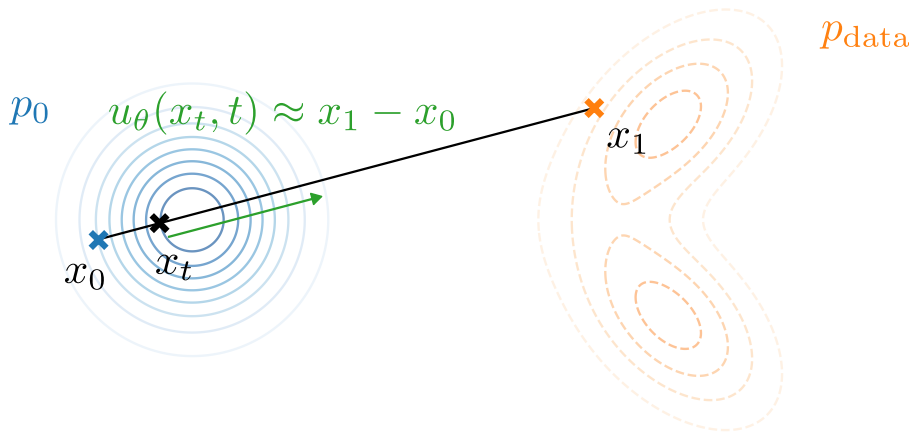
Training flow matching

$$\min_{\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} [\|u_{\theta}(x_t, t) - u^{\text{cond}}(x_t, z = x_1, t)\|^2] \quad (x_t := (1 - t)x_0 + tx_1)$$



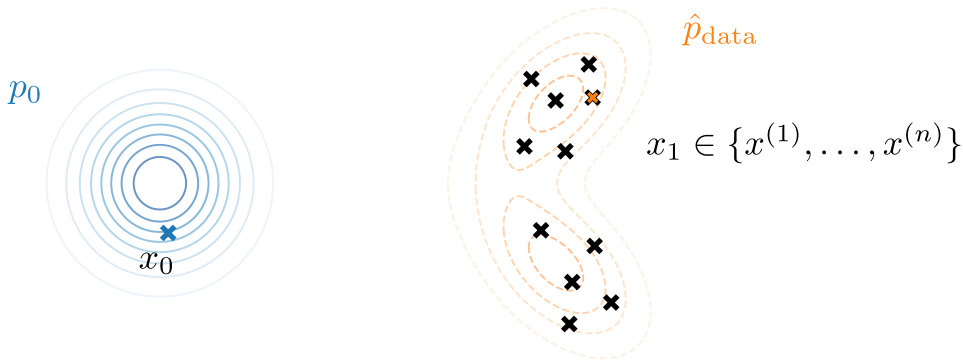
Training flow matching

$$\min_{\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} [\|u_{\theta}(x_t, t) - u^{\text{cond}}(x_t, z = x_1, t)\|^2] \quad (x_t := (1-t)x_0 + tx_1)$$



A small caveat

But in practice we replace p_{data} by \hat{p}_{data}



Remember the ideal “unavailable” velocity?

$$u^*(x, t) = \mathbb{E}_{z|x, t} [x_1 - x_0]$$

Prop: If p_{data} is replaced by $\hat{p}_{\text{data}} := \frac{1}{n} \sum_{i=1}^n \delta_{x^{(i)}}$, the optimal velocity has a closed-form:

$$\hat{u}^*(x, t) = \sum_{i=1}^n \lambda_i(x, t) \frac{x^{(i)} - x}{1 - t}$$

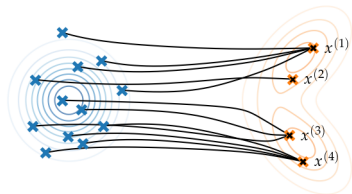
with $\lambda(x, t) = \text{softmax}((- \frac{1}{2(1-t)^2} \|x - tx^{(i')}\|^2)_{i'=1, \dots, n}) \in \mathbb{R}^n$

\hat{u}^* is now a finite sum!


What can we observe for \hat{u}^* as $t \rightarrow 1$?

Flow matching should not work

- because in practice we use \hat{p}_{data} instead of p_{data} , the minimizer of \mathcal{L}_{CFM} is available in closed-form
- this closed-form $\hat{u}^*(x, t)$ blows up for $t \rightarrow 1$ if $x \notin \{x^{(1)}, \dots, x^{(n)}\}$
- it can only generate training points!



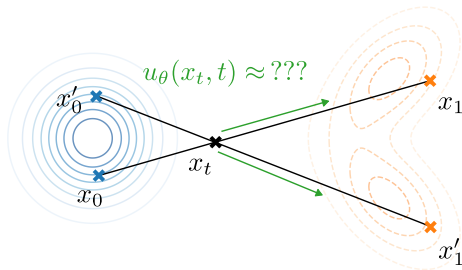
So why does flow matching generalize?

 *On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity*, Bertrand, Gagneux, Massias & Emonet, <https://www.arxiv.org/abs/2506.03719>

Generalization through variance?

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \|u_\theta(x_t, t) - (x_1 - x_0)\|^2$$

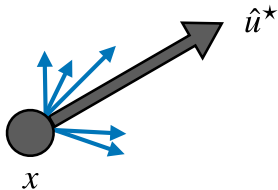
$$\hat{u}^*(x, t) = \sum_{i=1}^n \lambda_i(x, t) \frac{x^{(i)} - x}{1 - t}$$



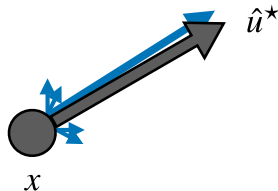
- an x_t is on n different segments $[x_0, x_1 = x^{(i)}]$
- instead of regressing against \hat{u}^* , we pick one of the $\frac{x^{(i)} - x}{1 - t}$ (w. proba $\lambda_i(x, t)$) in the sum and regress against it
- \hookrightarrow in training, u_θ is forced to learn various directions at the same (x, t)
- the noise in training may explain imperfect training hence generalization

Non stochasticity of \hat{u}^*

$$\hat{u}^*(x, t) = \sum_{i=1}^n p(z = x^{(i)} | x, t) u^{\text{cond}}(x, t, z = x^{(i)})$$



Common belief
STOCHASTICITY

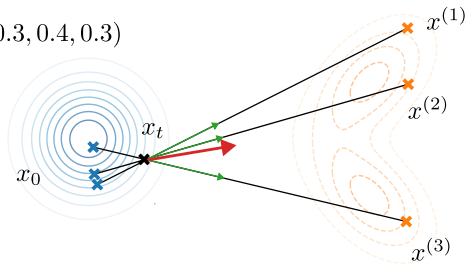


What really happens
NON-STOCHASTICITY

Non stochasticity of \hat{u}^*

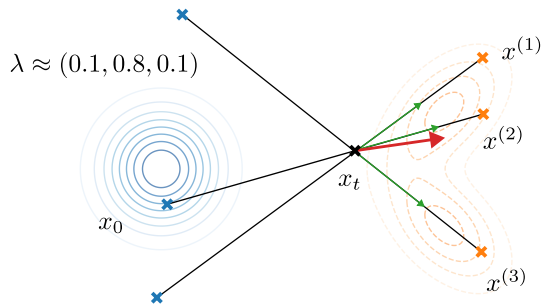
$$\hat{u}^*(x_t, t) = \sum_{i=1}^3 \lambda_i(x_t, t) \frac{x^{(i)} - x_t}{1-t}$$

$$\lambda \approx (0.3, 0.4, 0.3)$$



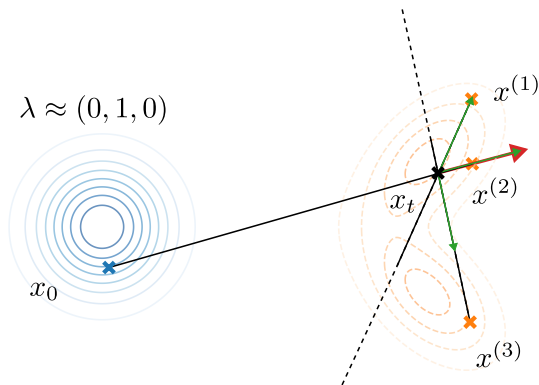
Non stochasticity of \hat{u}^*

$$\hat{u}^*(x_t, t) = \sum_{i=1}^3 \lambda_i(x_t, t) \frac{x^{(i)} - x_t}{1-t}$$

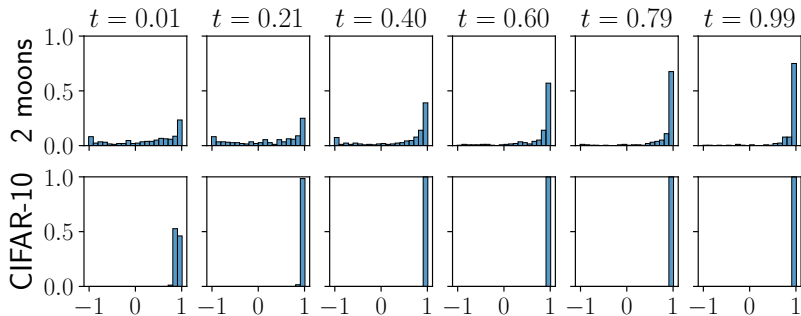


Non stochasticity of \hat{u}^*

$$\hat{u}^*(x_t, t) = \sum_{i=1}^3 \lambda_i(x_t, t) \frac{x^{(i)} - x_t}{1-t}$$

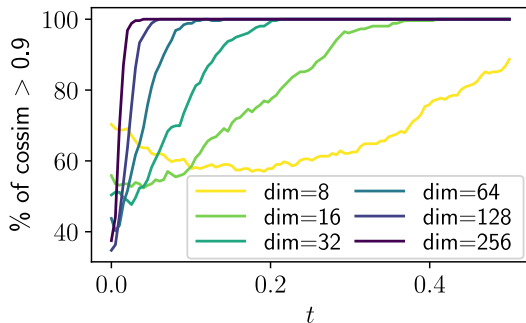


Non stochasticity for real data



histograms of cosine similarities between $\hat{u}^*((1-t)x_0 + tx_1, t)$ and $u^{\text{cond}}((1-t)x_0 + tx_1, z = x_1, t) = x_1 - x_0$

Issues of intuitions from small dimension



Alignment of \hat{u}^* and u^{cond} over time for varying image dimensions d on Imagenette

Stochasticity only occurs for very small t as dimension increases

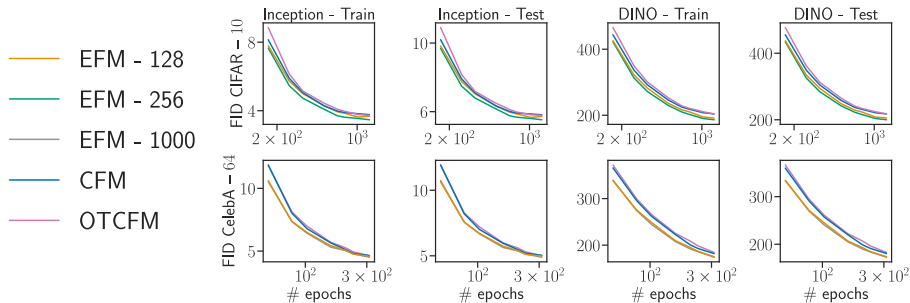
Refuting the stochasticity argument: regressing against \hat{u}^*

From

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \|u_\theta(x_t, t) - (x_1 - x_0)\|^2$$

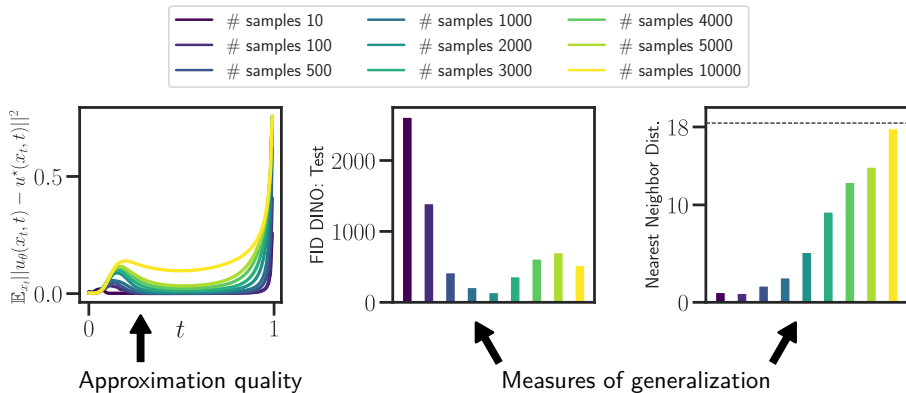
to

$$\mathcal{L}_{\text{EFM}}(\theta) = \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \|u_\theta(x_t, t) - \hat{u}^*(x_t, t)\|^2$$



Learning with a non-stochastic target *does not* degrade performance

Importance of model capacity

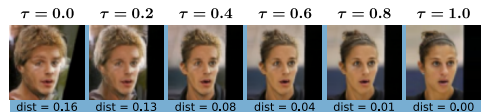
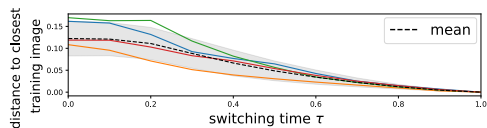
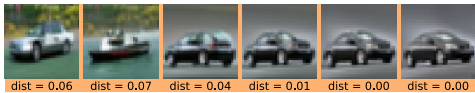
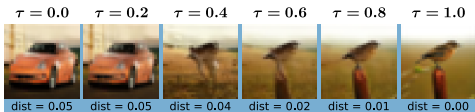
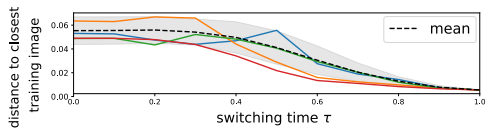


- generalization occurs when approximation degrades
- model u_θ has trouble learning \hat{u}^* for both $t \approx 0.2$ and $t \approx 0.9$

Which t matters most?

From a good trained u_θ , we build a *hybrid* model (fixed $\tau \in [0, 1]$):


- on $[0, \tau]$: follow \hat{u}^*
 - on $[\tau, 1]$: follow u_θ
-
- $\tau = 1$ means completely following \hat{u}^* (no generalization)
 - $\tau = 0$ means completely following u_θ (good generalization)




generalization arises early!

Summary

- by design, the true velocity in flow matching is available in closed-form
- flow matching should not create new images, yet it does
- stochasticity is definitely not the reason for it
- small and large times appear to matter most
- failure of u_θ to learn \hat{u}^* for small t is critical

 *On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity*, Bertrand, Gagneux, Massias & Emonet, <https://www.arxiv.org/abs/2506.03719>

 *A Visual Dive into Conditional Flow Matching*, Martin, Gagneux, Emonet, Bertrand & Massias, ICLR Blogpost 2025, <https://dl.heeere.com/cfm/>