Extreme Rain Generation Modeling using Score-Based Generative Models

Tiziano Fassina - PhD Student - Mines Paris

Tuesday 2nd December, 2025

~~Extreme Rain Generation Modeling using Score-Based Generative Models~~

Tiziano Fassina - PhD Student - Mines Paris

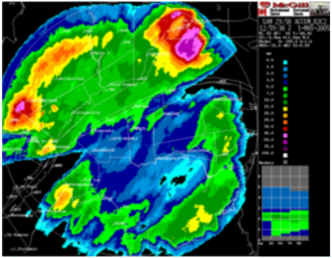
Tuesday 2nd December, 2025

Score-based Generative Models for Heavy-tail distributions: Challenges and Perspectives

Tiziano Fassina - PhD Student - Mines Paris

Tuesday 2nd December, 2025

Modeling distributions



(3.5, 7.6, 0.3, 5.6, 7.6)

Score-based Generative Models (SGMs) currently represent the **state of the art** in image, tabular and video generation thanks to their ability to accurately capture data modes and model complex dependencies.

LPSM Laboratoire de probabilités statistique & modélisation GEOLEARNING MINES PARIS | PSL

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on [0,T]

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on [0,T] ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on [0,T] ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\sigma_{T-t}\nabla \log p_{T-t}(y_t)$

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on [0,T] ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\sigma_{T-t}\nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on [0,T] ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\sigma_{T-t}\nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on [0,T] ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\sigma_{T-t}\nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$

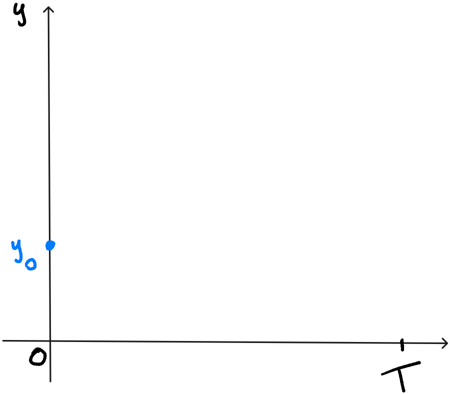
LPSM Laboratoire de probabilités statistique & modélisation GEOLEARNING MINES PARIS PSL

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t}\, \nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$

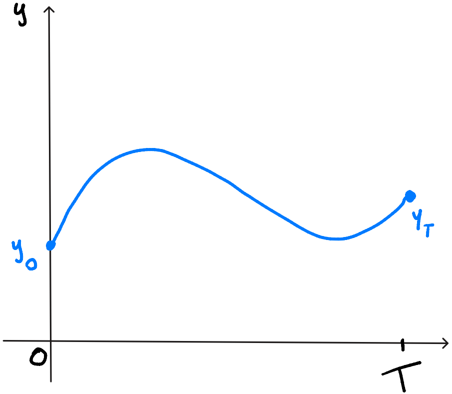
Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t}\, \nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



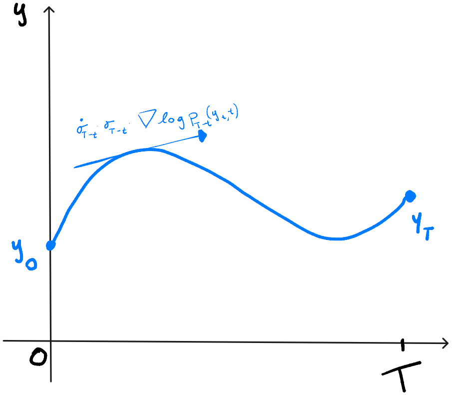
Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t} \,\nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



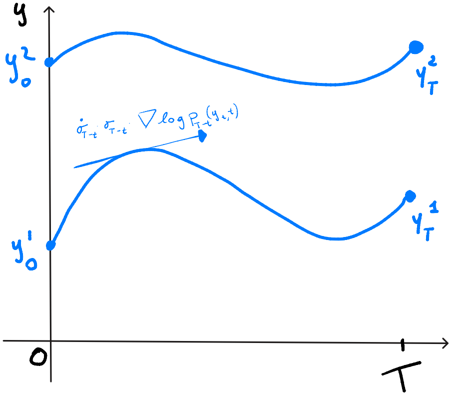
Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t}\, \nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



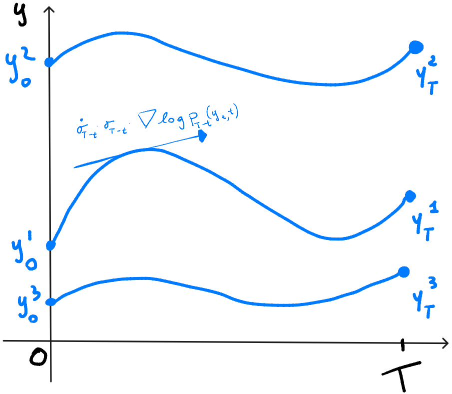
Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t}\, \nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



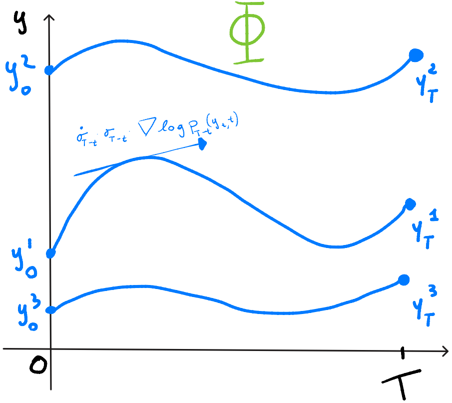
Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t}\, \nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



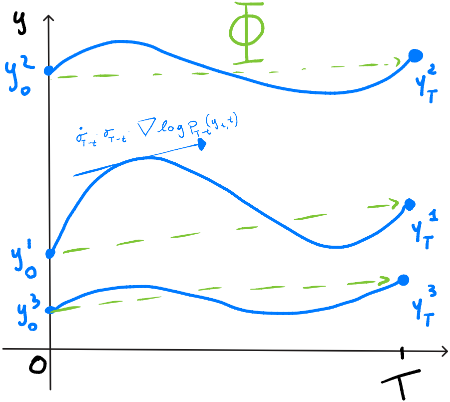
Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t}\, \nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



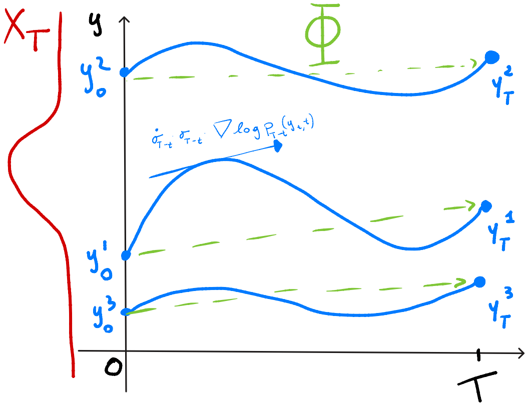
Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\,\sigma_{T-t}\,\nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



Score based Generative Models (SGM)

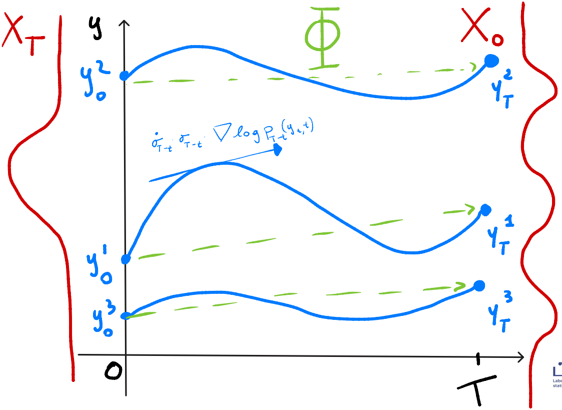
- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t}\, \nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



LPSM Laboratoire de probabilités statistique & modélisation
GEOLEARNING
MINES PARIS | PSL

Score based Generative Models (SGM)

- $X_0 \in \mathbb{R}^d$ a r.v. $p_0(x)$
- $t \mapsto \sigma_t$ for $t \in [0, T]$ a noise schedule
- Set of densities $p_t(x)$ on $[0, T]$ ($p_t(x)$ the law of $X_t = X_0 + \sigma_t Z$)
- $\dot{y}_t = \dot{\sigma}_{T-t}\, \sigma_{T-t}\, \nabla \log p_{T-t}(y_t)$
- We can introduce the map $\Phi : y_0 \mapsto y_T$
- $Y_0 \overset{d}{\sim} X_T \implies \Phi(Y_0) = Y_T \overset{d}{\sim} X_0$



Needed approximation

- We don't know $p_0(x)$

Needed approximation

- We don't know $p_0(x) \implies$ Don't know $p_t(x)$
- We don't know $\nabla_x \log p_t(x)$ and $X_T \overset{d}{\sim} p_T$

Needed approximation

- We don't know $p_0(x) \implies$ Don't know $p_t(x)$
- We don't know $\nabla_x \log p_t(x)$ and $X_T \overset{d}{\sim} p_T$
- $p_0(x)$ represented by $\{X_0^i\}_i$

Needed approximation

- We don't know $p_0(x) \implies$ Don't know $p_t(x)$
- We don't know $\nabla_x \log p_t(x)$ and $X_T \overset{d}{\sim} p_T$
- $p_0(x)$ represented by $\{X_0^i\}_i$
- $s_\theta(t, x) \simeq \nabla_x \log p_t(x)$ trained on $\{X_0^i\}_i$

  $$\mathbb{E}_{t\sim q(t), i\sim\mathcal{U}(n), Z\sim\mathcal{N}(0,I)}[\|s_\theta(t, X_0^i + \sigma_t Z) - \nabla_x \log p_t(X_0^i + \sigma_t Z)\|^2]$$

- We approximate $p_T(x) \sim X_0 + \sigma_T Z$ with $\pi_\infty \overset{d}{\sim} \sigma_T Z$

LPSM Laboratoire de probabilités statistique & modélisation GEOLEARNING MINES PARIS | PSL

Needed approximation

- We don't know $p_0(x) \implies$ Don't know $p_t(x)$
- We don't know $\nabla_x \log p_t(x)$ and $X_T \overset{d}{\sim} p_T$
- $p_0(x)$ represented by $\{X_0^i\}_i$
- $s_\theta(t, x) \simeq \nabla_x \log p_t(x)$ trained on $\{X_0^i\}_i$
- We approximate $p_T(x)$ with π_∞
- (We approximate Φ solving the equation with numerical methods)

$$\dot{y}_t = \dot{\sigma}_{T-t}\sigma_{T-t}s_\theta(t, y_t)$$

Needed approximation

- We don't know $p_0(x) \implies$ Don't know $p_t(x)$
- We don't know $\nabla_x \log p_t(x)$ and $X_T \overset{d}{\sim} p_T$
- $p_0(x)$ represented by $\{X_0^i\}_i$
- $s_\theta(t, x) \simeq \nabla_x \log p_t(x)$ trained on $\{X_0^i\}_i$
- We approximate $p_T(x)$ with π_∞
- (We approximate Φ solving the equation with numerical methods)

$$\dot{y}_t = \dot{\sigma}_{T-t}\sigma_{T-t}s_\theta(t, y_t)$$

- Initialize $\hat{Y}_0 \overset{d}{\sim} \sigma_T Z$

Needed approximation

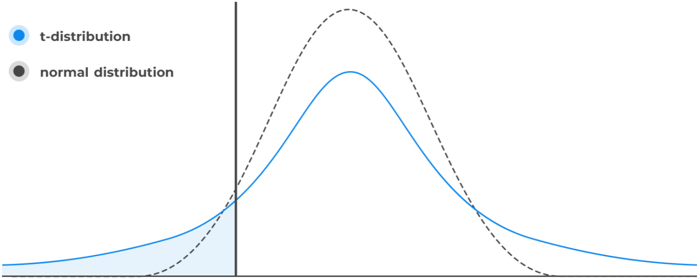
- We don't know $p_0(x) \implies$ Don't know $p_t(x)$
- We don't know $\nabla_x \log p_t(x)$ and $X_T \overset{d}{\sim} p_T$
- $p_0(x)$ represented by $\{X_0^i\}_i$
- $s_\theta(t, x) \simeq \nabla_x \log p_t(x)$ trained on $\{X_0^i\}_i$
- We approximate $p_T(x)$ with π_∞
- (We approximate Φ solving the equation with numerical methods)

$$\dot{y}_t = \dot{\sigma}_{T-t}\sigma_{T-t}s_\theta(t, y_t)$$

- Initialize $\hat{Y}_0 \overset{d}{\sim} \sigma_T Z$
- Generate $\hat{Y}_T = \hat{\Phi}(\hat{Y}_0)$

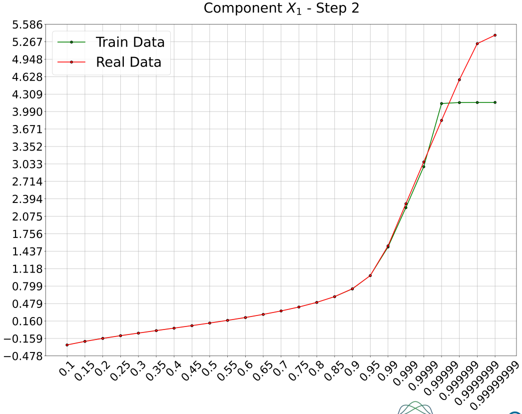
Modeling distributions

- (Data suggest that) Data distribution is heavytail
- $\mathbb{P}(X^i > t) \sim t^{-\alpha}$ for some pixel/component ($\alpha > 0$)



Example : The Frechet Distribution - Log-quantiles

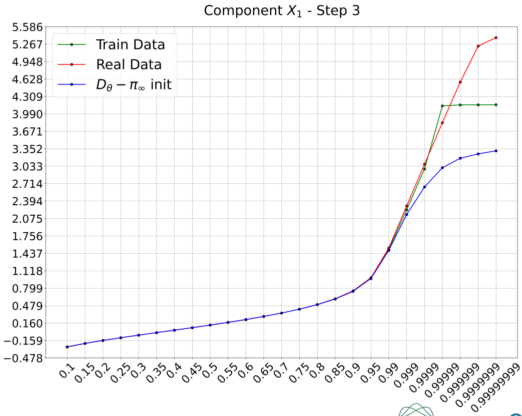
- $f_{Fr}(x) = \alpha x^{-\alpha-1}$ for $x \in \mathbb{R}^+, \ \alpha = 3$
- 10^5 training datapoints



LPSM Laboratoire de probabilités statistique & modélisation GEOLEARNING MINES PARIS | PSL

Example : The Frechet Distribution - Log-quantiles

- $f_{Fr}(x) = \alpha x^{-\alpha-1}$ for $x \in \mathbb{R}^+, \ \alpha = 3$
- 10^5 training datapoints



LPSM Laboratoire de probabilités statistique & modélisation
GEOLEARNING
MINES PARIS | PSL

What's wrong ?

Something went wrong: Tail events are under-generated

What's wrong ?

Something went wrong: Tail events are under-generated

- The trained score $s_\theta(t, x)$ is not a good estimator

What's wrong ?

Something went wrong: Tail events are under-generated

- The trained score $s_\theta(t, x)$ is not a good estimator
- The initialization $Y_0 \sim \sigma_T Z$ is very different from $X_T \sim p_T$

What's wrong ?

Something went wrong: Tail events are under-generated

- The trained score $s_\theta(t, x)$ is not a good estimator
- The initialization $Y_0 \sim \sigma_T Z$ is very different from $X_T \sim p_T$
- The numerical scheme to approximate $\Phi(y_0)$ is problematic

What's wrong ?

Something went wrong: Tail events are under-generated

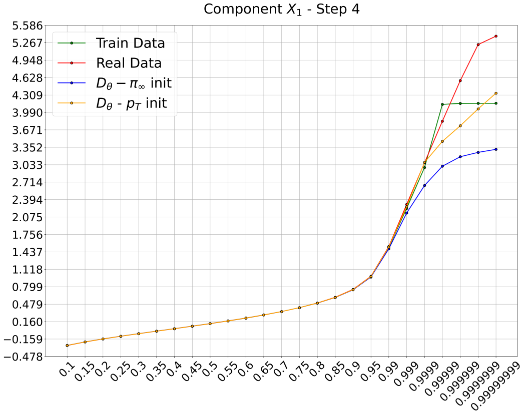
- The trained score $s_\theta(t, x)$ is not a good estimator
- The initialization $Y_0 \sim \sigma_T Z$ is very different from $X_T \sim p_T$
- ~~The numerical scheme to approximate $\Phi(y_0)$ is problematic~~

Initialization problems

$X_0 + \sigma_T Z \sim p_T$ is α-heavytail distribution - π_∞ is light-tailed

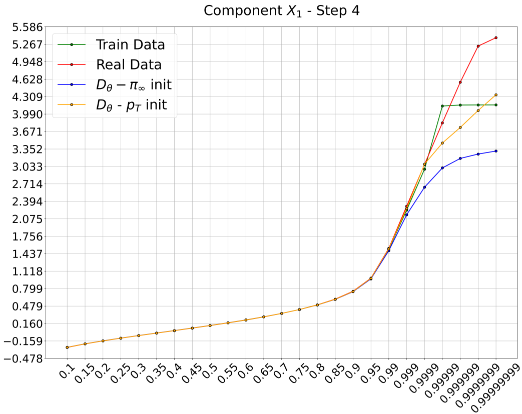
Initialization problems

$X_0 + \sigma_T Z \sim p_T$ is α-heavytail distribution - π_∞ is light-tailed



Initialization problems

$X_0 + \sigma_T Z \sim p_T$ is α-heavytail distribution - π_∞ is light-tailed



p_T is a simple unimodal distribution - fit a heavytail model π_θ on noised data

LPSM Laboratoire de probabilités statistique & modélisation GEOLEARNING MINES PARIS | PSL

Score estimation problem

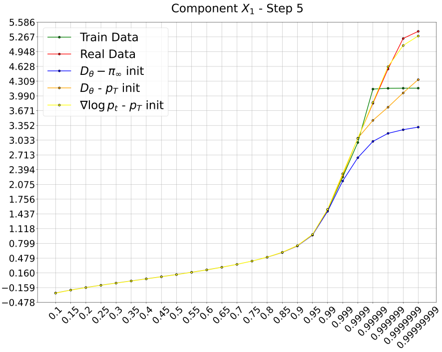
$$s_\theta(t, x) \neq \nabla_x \log p_t(x)$$

Score estimation problem

$s_\theta(t, x) \neq \nabla_x \log p_t(x)$ What if we use the true score ?

Score estimation problem

$s_\theta(t, x) \neq \nabla_x \log p_t(x)$ What if we use the true score ?



The real solved equation allows
to perfectly generate the tail.

Score estimation problem

The problem is that the score learns to generalize generation

Score estimation problem

The problem is that the score learns to generalize generation

But it learns to generalize on the data support

Score estimation problem

The problem is that the score learns to generalize generation

But it learns to generalize on the data support

This is not an error but a reasonable behaviour of the neural net.

Score estimation problem

The problem is that the score learns to generalize generation

But it learns to generalize on the data support

This is not an error but a reasonable behaviour of the neural net.

The denoiser doesn't know whether the data law is heavy-tail or compactly supported

Score estimation problem

The problem is that the score learns to generalize generation
But it learns to generalize on the data support
This is not an error but a reasonable behaviour of the neural net
The neural net doesn't know whether the data law is heavy-tail or compactly supported

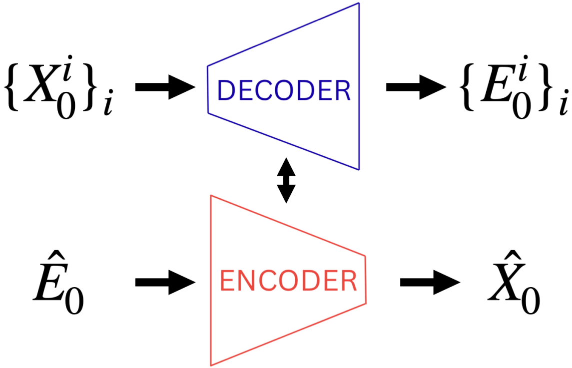
Therefore, to make the denoiser generate a heavy-tailed distribution, we need to **explicitly force it** to model a heavy-tailed law.

Encoding & Latent Space

- We train a score approximator on heavy-tail data $\{X_0^i\}_i$
- Maybe we could preprocess $\{X_0^i\}_i$ (heavytail) to $\{E_0^i\}_i$ (light-tailed)
- Learn $\hat{E}_0$ and reverse the preprocessing to get $\hat{X}_0$

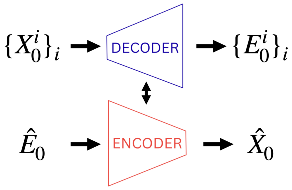
Encoding & Latent Space

- We train a score approximator on heavy-tail data $\{X_0^i\}_i$
- Maybe we could preprocess $\{X_0^i\}_i$ (heavytail) to $\{E_0^i\}_i$ (light-tailed)
- Learn $\hat{E}_0$ and reverse the preprocessing to get $\hat{X}_0$



Encoding & Latent Space

- We train a score approximator on heavy-tail data $\{X_0^i\}_i$
- Maybe we could preprocess $\{X_0^i\}_i$ (heavytail) to $\{E_0^i\}_i$ (light-tailed)
- Learn $\hat{E}_0$ and reverse the preprocessing to get $\hat{X}_0$



This is a viable approach only if the neural net is able to correctly reproduce tails of a light-tailed distribution E_0

Conclusions & things i didn't talk about

- SGM are the most powerful modeling tool
- We have serious problems in modeling tails
- The problem is posed by the score function
- Optimal Transport map, properties of Unets architecture