RCP211 - Modèles génératifs

Modèles de diffusion

Arnaud Breloy arnaud.breloy@lecnam.net

8 décembre 2025

Conservatoire national des arts & métiers

Outline

Rappels (1/2)

Objectif : générer de nouvelles données similaires à celles existantes

Point de vue probabiliste : les données sont des réalisations de variables aléatoires

Ce qu'on veut :

- la densité de probabilité (d.d.p.) des données $p(\mathbf{x})$
- un moyen d'échantillonner selon $p(\mathbf{x})$

Ce qu'on a :

- Des échantillons $\{\mathbf{x}_i\}_{i=1}^n$
- Différentes approches pour en tirer une approximation de l'idéal

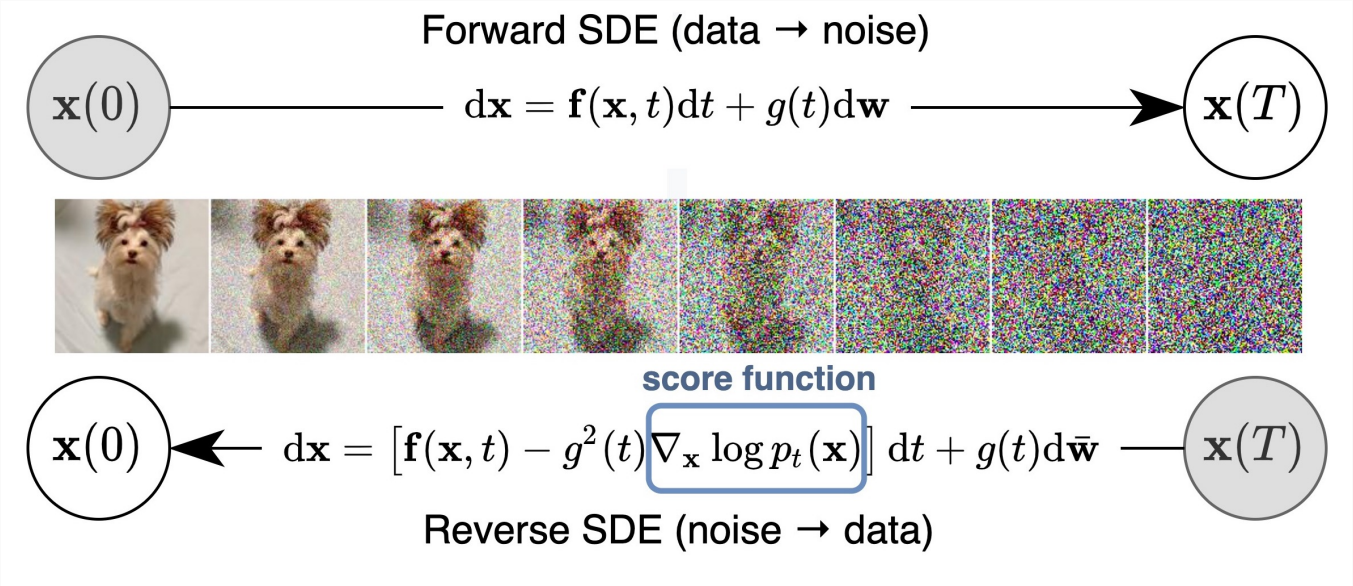
Rappels (2/2)

Méthodes vues précédemment

- **Modèles paramétriques classiques**
  - PCA probabiliste, GMM
  - Modèles AR
- **Auto-ecodeurs variationnels**
  - Auto-encodeur + échantillonnage de code lattent $\mathbf{z}$
  - Les paramètres modélisent les lois $q_\phi(\mathbf{z}|\mathbf{x})$ et $p_\theta(\mathbf{x}|\mathbf{z})$
  - Approx. le max. de vraisemblance par inférence variationnelle (ELBO)
- **Réseaux génératifs antagonistes**
  - Deux réseaux de neurones : générateur et discriminateur
  - plaque implicitement la d.d.p. des données générées sur $p(x)$ par un jeu min-max

Modèles de diffusion, vue d'ensemble

Modèles de diffusion : "longue séquence de petits dé-bruiteurs"



Principes généraux :

- Apprentissage et lien avec l'**estimation de score**
- Génération et lien avec **dynamique de Langevin**
- Le point de vue "**VAE**"

Références

Articles

Luo, C. (2022). Understanding diffusion models : A unified perspective. arXiv [pdf]

Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems [pdf]

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems [pdf]

Vincent, P. (2011). A connection between score matching and denoising autoencoders. Neural computation [pdf]

Hyvärinen, A., & Dayan, P. (2005). Estimation of non-normalized statistical models by score matching. Journal of Machine Learning Research [pdf]

Blog posts

Song, Yang. (2021). “Generative Modeling by Estimating Gradients of the Data Distribution” [html]

Weng, Lilian. (2021). “What are diffusion models ?” [html]

Vidéos

Stanford CS236 : Deep Generative Models (cours 13, 14, 16) [yt]

Cours de Stéphane Mallat au Collège de France [html] [yt]

pour information : ce cours s’inspire fortement des slides de CS236

Outline

Idée générale

On cherche a modéliser la d.d.p. “réelle” $p(x)$ par $p_\theta(x)$

La tache est difficile :

- Dans un VAE, on approche conceptuellement la modélisation par

$$p_\theta(x) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}$$

  mais on n’a pas réellement accès à la d.d.p en pratique.
- Si on utilise un réseau $f_\theta : \mathbb{R}^d \to \mathbb{R}^+$ et

$$p_\theta(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z(\theta)} \quad \text{avec} \quad Z(\theta) = \int e^{f_\theta(x)}\mathrm{d}\mathbf{x}$$

  impossible d’obtenir la normalisation $Z(\theta)$ raisonnablement

Arrive alors l’idée de **score matching**

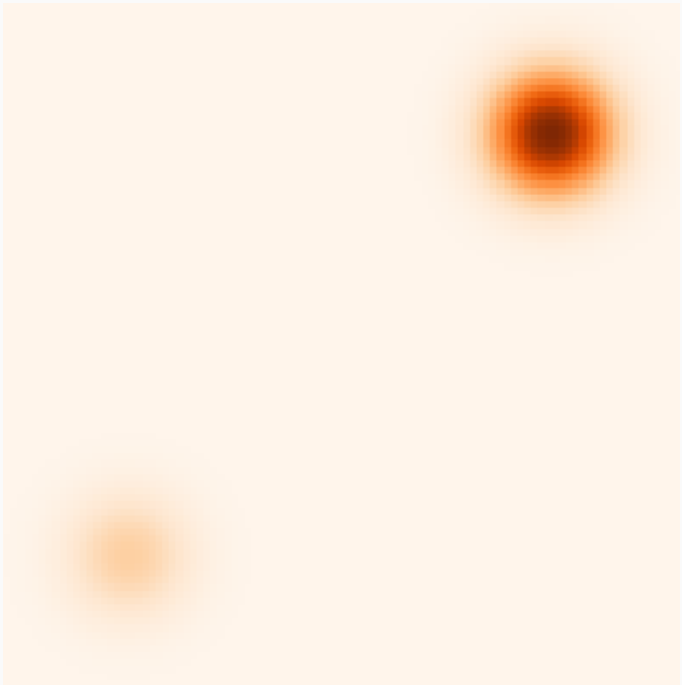
Score (ou fonction score)

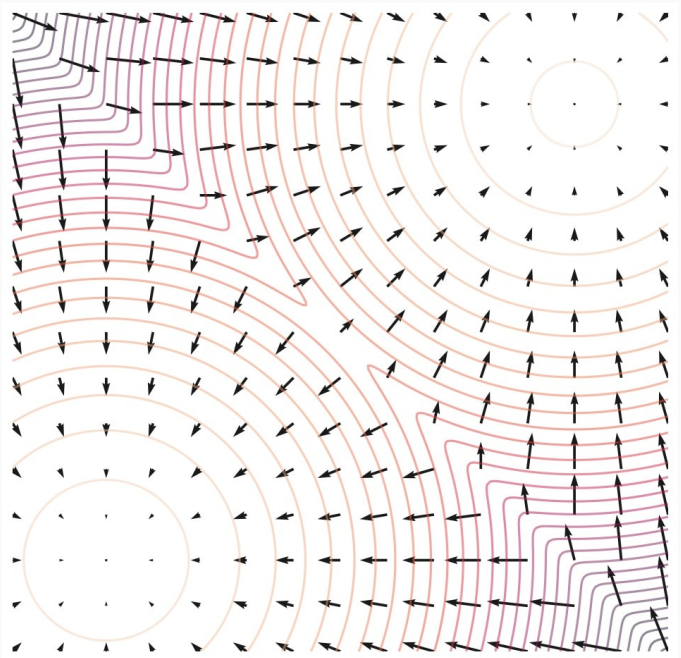
Definition

La fonction score $\boldsymbol{s} : \mathbb{R}^d \to \mathbb{R}^d$ d'une distribution p est

$$\boldsymbol{s}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

"donne à chaque x la direction de plus forte pente pour augmenter la probabilité"





Attention, légèrement différent du score de Fisher classique en théorie de l'information

Propriétés intéressantes du score

Le soucis de constante de normalisation disparaît

$$\begin{aligned}\nabla_x \log p_\theta(x) &= \nabla_x \log \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta} \\ &= \nabla_x \log e^{-f_\theta(\mathbf{x})} + \nabla_x \log(Z_\theta) \\ &= -\nabla_x f_\theta(\mathbf{x})\end{aligned}$$

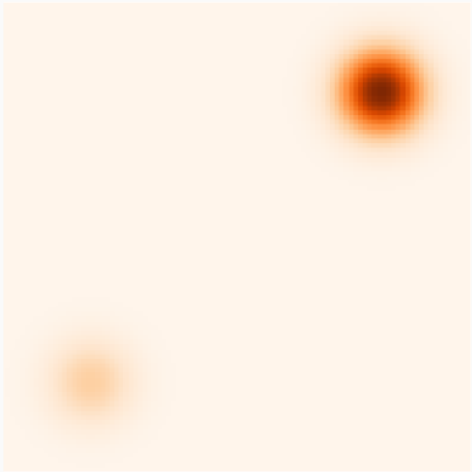
On peut retrouver la d.d.p. par intégration (+ la normalisation résout l'ambiguïté)

La quantité intervient dans les processus génératifs

Estimation du score

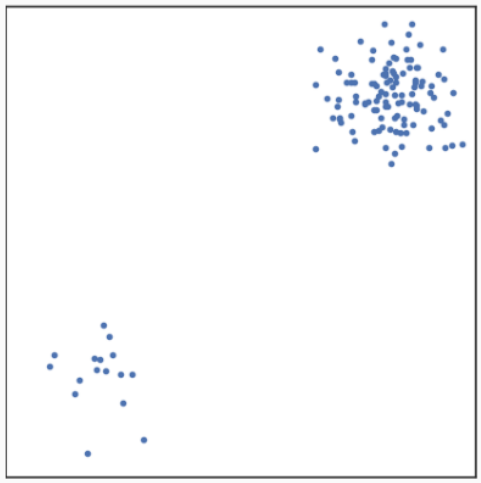
Probability density

$p_{\text{data}}(\mathbf{x})$



i.i.d. samples

$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$



Score function

$\boldsymbol{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$

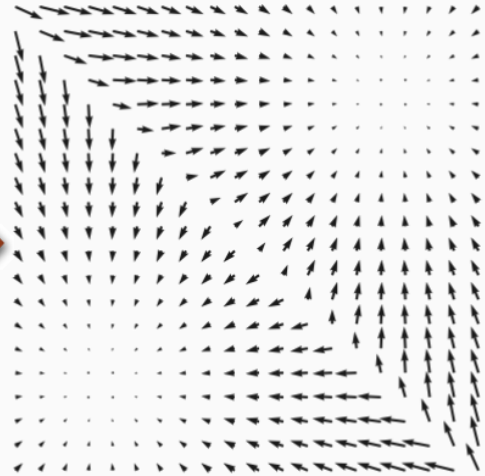


image taken from CS236

Estimation du score

- Jeu de données $\{\mathbf{x}_i\}_{i=1}^n \sim p_{\text{data}}(\mathbf{x})$
- Fonction paramétrée $\boldsymbol{s}_\theta : \mathbb{R}^d \to \mathbb{R}^d$ (réseau de neurones)
- Trouver θ tel que $\boldsymbol{s}_\theta(\mathbf{x}) \simeq \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$
- On doit quantifier une perte

Divergence de Fisher

$$\begin{aligned} \mathcal{L}(\theta) &= \tfrac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[||\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - s_\theta(\mathbf{x})||_2^2\right] \\ &= \tfrac{1}{2}\sum_{i=1}^n ||\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - s_\theta(\mathbf{x})||_2^2 \end{aligned}$$

Score matching

$$\underset{\theta\in\Theta}{\text{minimize}} \quad \mathcal{L}(\theta)$$

l'expression dépend de p_{data}, que l'on cherche !

Estimation du score

Théorème (Hyvärinen 2005)

Sous certaines conditions de régularité

$$\begin{aligned}\mathcal{L}(\theta) &= \tfrac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[||\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - s_\theta(\mathbf{x})||_2^2\right] \\ &= \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[\tfrac{1}{2}||s_\theta(\mathbf{x})||_2^2 + \text{tr}(\nabla_{\mathbf{x}} s_\theta(\mathbf{x}))\right] + \text{const.}\end{aligned}$$

où $\nabla_{\mathbf{x}} s_\theta(\mathbf{x})$ est la matrice Jacobienne de $s_\theta(\mathbf{x})$

- L'expression devient actionnable car elle ne dépend plus de p_{data}
- En pratique, évaluer la diagonale de $\nabla_{\mathbf{x}} s_\theta(\mathbf{x})$ est trop couteux

  $\mathcal{O}(d)$ rétropropagations

- En l'état, l'estimation de score ne passe pas a l'échelle
- Deux approximations possibles
  - Méthodes de slicing (Song 2020)
  - **Score matching par débruitage** (Vincent 2011)

Techniques d'approximation pour l'estimation de score

Estimation de score par débruitage

On considère une version bruitée des données

$$q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2\mathbf{I}) \qquad q_\sigma(\tilde{\mathbf{x}}) = \int q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) p_{\text{data}}(\mathbf{x}) \mathrm{d}\mathbf{x}$$

Bénéfices :

- L'estimation du score $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$ se simplifie grandement
- Si σ est petit, $q_\sigma(\tilde{\mathbf{x}}) \simeq p_{\text{data}}(\tilde{\mathbf{x}})$
- Lien théoriques avec un problème de débruitage et les VAE

Estimation de score par débruitage

$\mathbf{x}$

$p_{\text{data}}(\mathbf{x})$

$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$

$\tilde{\mathbf{x}}$

$q_\sigma(\tilde{\mathbf{x}})$

On ajoute un bruit Gaussien aux données

$$\tilde{\mathbf{x}} = \mathbf{x} + \sigma \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

$$q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2 \mathbf{I})$$

$$q_\sigma(\tilde{\mathbf{x}}) = \int q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) p_{\text{data}}(\mathbf{x}) \mathrm{d}\mathbf{x}$$

Théorème (Vincent 2011)

$$\begin{aligned} \tilde{\mathcal{L}}(\theta) &= \tfrac{1}{2}\mathbb{E}_{\tilde{\mathbf{x}} \sim q_\sigma}\left[||\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) - \boldsymbol{s}_\theta(\tilde{\mathbf{x}})||_2^2\right] \\ &= \tfrac{1}{2}\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}, \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left[||\boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})||_2^2\right] + \text{const.} \end{aligned}$$

Estimation de score par débruitage

On considère donc la loss

$$\tilde{\mathcal{L}}(\theta) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}},\tilde{\mathbf{x}}\sim q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left[||\boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}}\log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})||_2^2\right]$$

Or, si $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2\mathbf{I})$, tout se simplifie, car

$$\nabla_{\tilde{\mathbf{x}}}\log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2}$$

En pratique (reparameterization trick)

$$\begin{aligned}
\tilde{\mathcal{L}}(\theta) &= \tfrac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}},\tilde{\mathbf{x}}\sim q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left[||\boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \tfrac{\tilde{\mathbf{x}}-\mathbf{x}}{\sigma^2}||_2^2\right] \\
&= \tfrac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}},\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[||\boldsymbol{s}_\theta(\mathbf{x}+\sigma\mathbf{z})||_2^2 - 2\boldsymbol{s}_\theta(\mathbf{x}+\sigma\mathbf{z})^\top\tfrac{\mathbf{z}}{\sigma}\right]
\end{aligned}$$

Denoising Score Matching

Entrainement

- Echantilloner un mini-batch $\{\mathbf{x}_1, ..., \mathbf{x}_n\} \sim p(\mathbf{x})$
- Echantilloner les versions bruitées $\{\tilde{\mathbf{x}}_1, ..., \tilde{\mathbf{x}}_n\} \sim q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$
- Evaluer la loss

$$\frac{1}{n}\sum_{i=1}^{n} ||s_\theta(\tilde{\mathbf{x}}_i) - \frac{\mathbf{x}_i - \tilde{\mathbf{x}}_i}{\sigma^2}||_2^2$$

- Effectuer un pas de descente de gradient

- Facile a mettre en oeuvre, même en grande dimension
- Mais on ne pourra pas estimer le score des données non bruitées
- Idealement on veut $\sigma \to 0$, mais numériquement instable

Lien avec le débruitage

$$\mathbb{E}_{\tilde{\mathbf{x}}\sim q_\sigma(\tilde{\mathbf{x}})}\left[||\nabla_{\tilde{\mathbf{x}}}\log q_\sigma(\tilde{\mathbf{x}}) - \boldsymbol{s}_\theta(\tilde{\mathbf{x}})||_2^2\right] = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\mathbb{E}_{\tilde{\mathbf{x}}\sim q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left[\left\|\frac{\mathbf{x}-\tilde{\mathbf{x}}}{\sigma^2} - \boldsymbol{s}_\theta(\tilde{\mathbf{x}})\right\|_2^2\right]$$

- $\boldsymbol{s}_\theta$ cherche a prédire le bruit ajouté à $\mathbf{x}$ pour produire $\tilde{\mathbf{x}}$
- L'estimation du score est équivalent à un problème de débruitage
- Formule de Tweedie :

$$\hat{\mathbf{x}} = \tilde{\mathbf{x}} + \sigma^2\nabla_{\tilde{\mathbf{x}}}\log q_\sigma(\tilde{\mathbf{x}})$$

"la meilleure stratégie de débruitage est de suivre le score"

Outline

Echantillonage par dynamique de Langevin

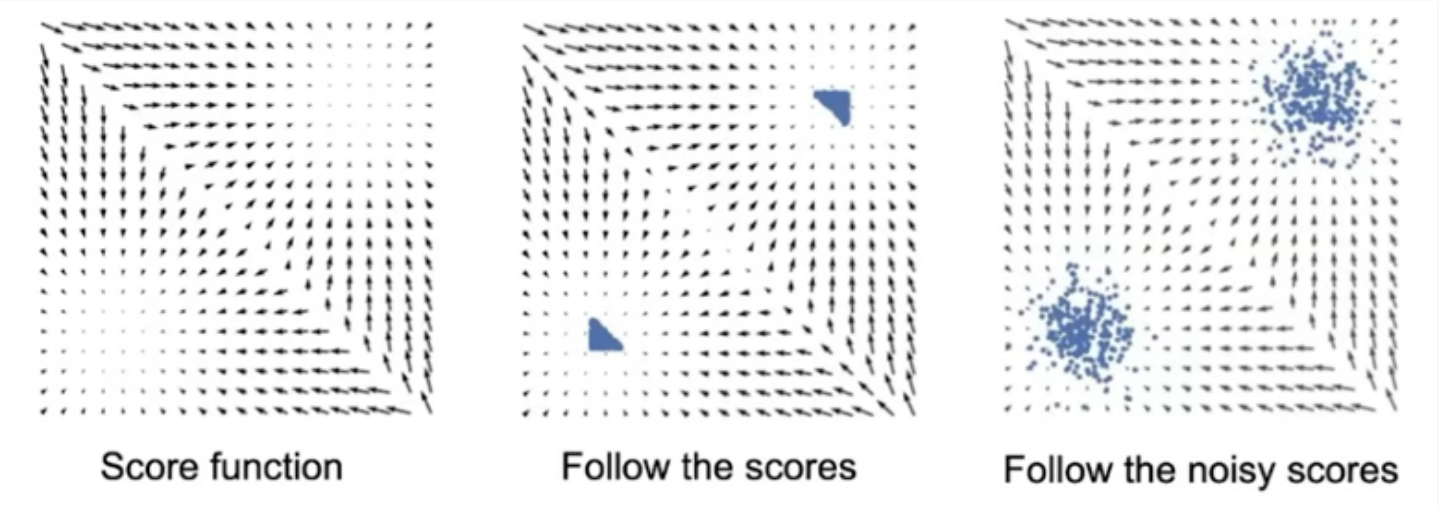
Dynamique de Langevin

- $\mathbf{x}_0 \sim \pi(\mathbf{x}) \leftarrow$ initialisation aléatoire
- Produire T itérations suivant

$$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \frac{\epsilon}{2}\nabla_{\mathbf{x}} \log p(\mathbf{x}_t) + \epsilon \boldsymbol{z}_t$$

avec $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

si $\epsilon << 0$ et $T \to \infty$, alors $\mathbf{x}_T \sim p$



Génération de données avec un modèle de score

Deux étapes :

- Apprendre un **estimateur du score** s_θ sur un ensemble $\{\mathbf{x}_i\}_{i=1}^n$
- Utiliser le **score appris** dans une **dynamique de Langevin**

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t+1} + \frac{\epsilon}{2} s_\theta(\mathbf{x}_t) + \epsilon \boldsymbol{z}_t, \quad t \in [\![1, T]\!]$$

- Si $s_\theta(\mathbf{x}) \simeq \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$, alors on approche $\mathbf{x}_T \sim p_{\text{data}}$!

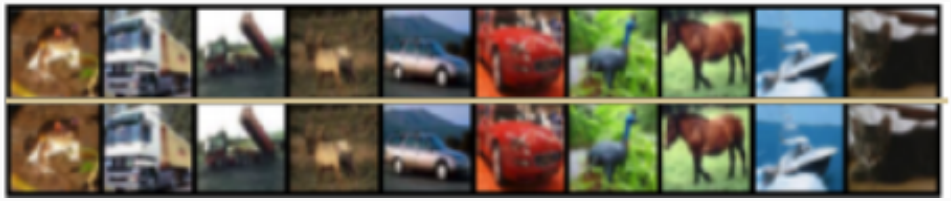
Mis en pratique : cela ne fonctionne pas :(

Limitations (1/2)

Problème #1 : dimension de l'espace ambiant

- Les données vivent dans une variété $\mathcal{M}$ et non dans $\mathbb{R}^d$
- Dimension intrinsèque $d' \ll d$
- Exemple avec un simple variété linéaire (PCA)

$d = 784 \rightarrow d' = 595$

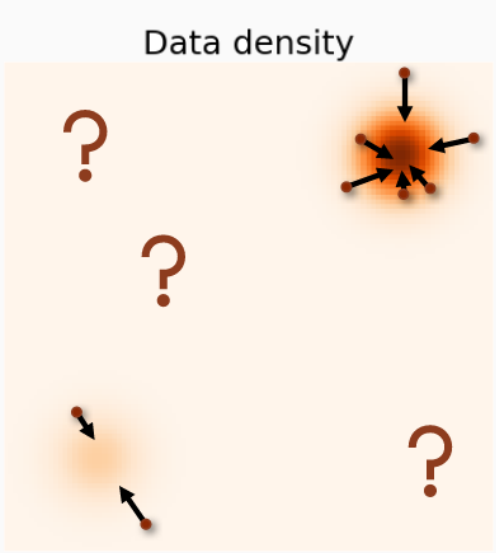


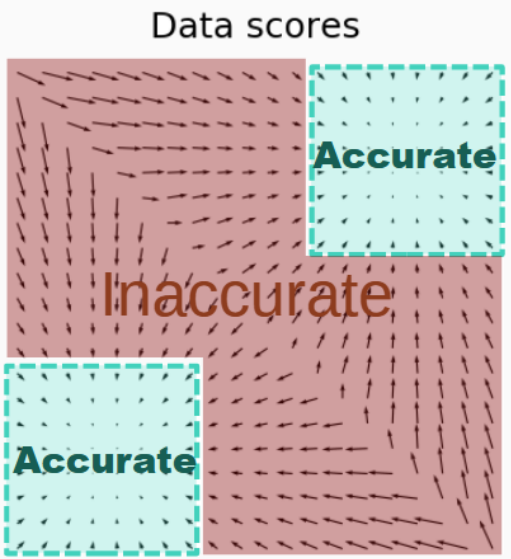
$d = 3072 \rightarrow d' = 2165$

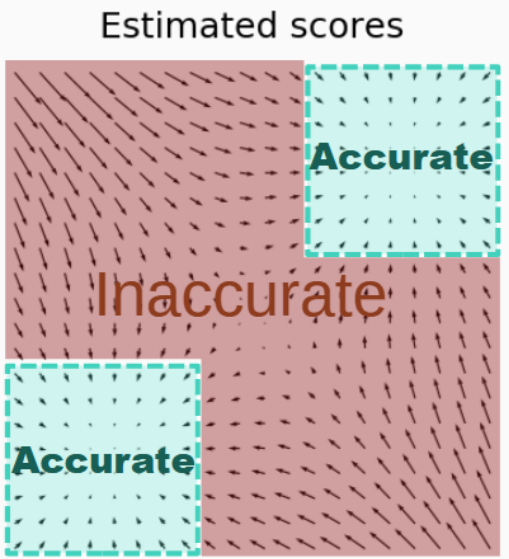
- Le score n'est pas défini partout (problème sur les hypothèses de régularité)

Limitations (2/3)

Problème #2 : dimension de l'espace ambiant







$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\|_2^2]$$

La dyn. de Langevin n'explore pas les zones de faible densité correctement

Limitations (3/3)

Problème #3 : proportion des modes

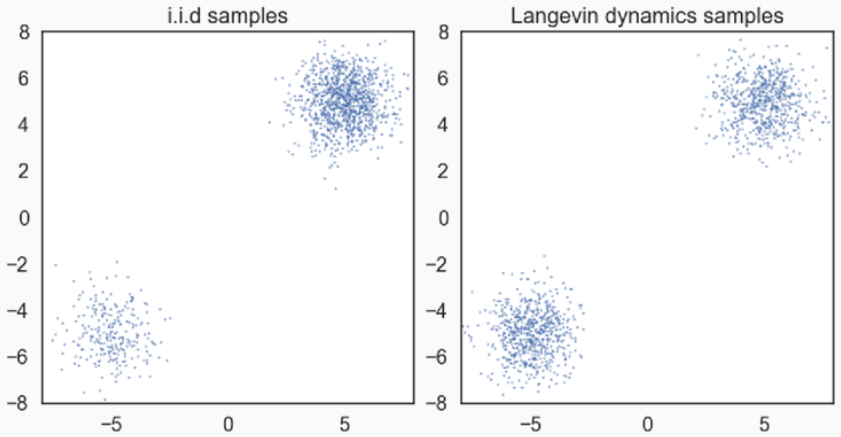
- Si la distribution a deux modes à supports disjoints $\mathcal{A} \cap \mathbb{B} = \varnothing$

$$p_{\text{data}}(\mathbf{x}) = \pi p_1(\mathbf{x}) + (1 - \pi) p_2(\mathbf{x})$$

- Le score est insensible aux propostions π

$$\begin{aligned} \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) &= \nabla_{\mathbf{x}} \log \pi p_1(\mathbf{x}) + \nabla_{\mathbf{x}} \log (1 - \pi) p_2(\mathbf{x}) \\ &= \nabla_{\mathbf{x}} \left[\log \pi + \log p_1(\mathbf{x}) \right] + \nabla_{\mathbf{x}} \left[\log (1 - \pi) + \log p_2(\mathbf{x}) \right] \\ &= \nabla_{\mathbf{x}} p_1(\mathbf{x}) + \nabla_{\mathbf{x}} p_2(\mathbf{x}) \end{aligned}$$

- La dynamique de Langevin ne refletera pas les proportions



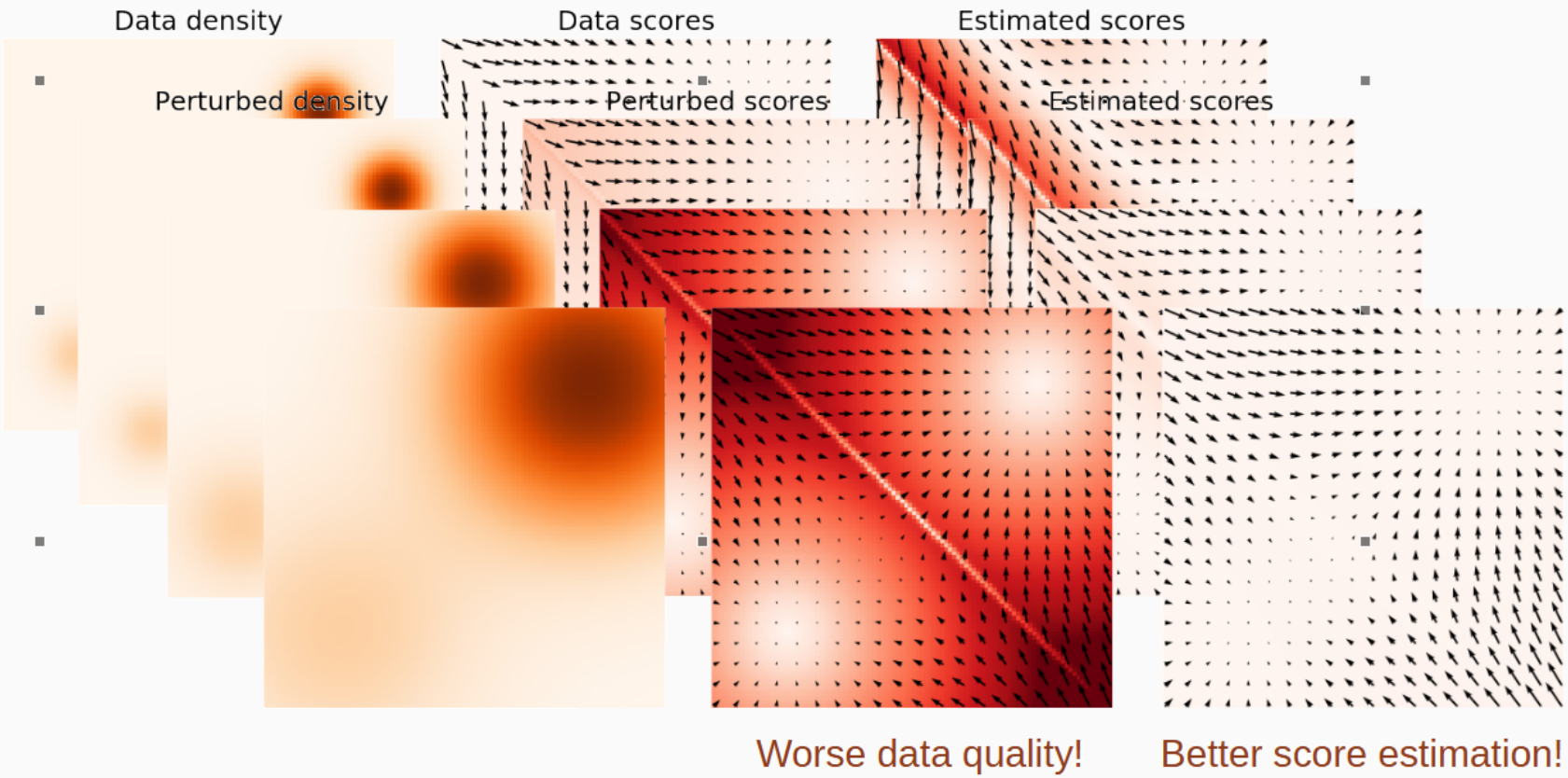
Solution : perturbation multi-échelle

$$\sigma_1 > \sigma_2 > \cdots > \sigma_{L-1} > \sigma_L$$

On peut tirer profit du compromis

- σ grand : densité moins piquée, score plus facile a estimer
- σ petit : plus proche de la distribution originale

Compromis piloté par le niveau de bruit σ



Apprentissage du score avec perturbation multi echelle

Modèle de score conditionné au bruit

On utilise le niveau de bruit comme paramètre d'entrée additionnel

$$\boldsymbol{s}_\theta(\mathbf{x}, \sigma) : \mathbb{R}^d \times \mathbb{R}^+ \longrightarrow \mathbb{R}^d$$

Loss multi-échelle

$$\begin{aligned}
\mathcal{L}(\theta) &= \frac{1}{L}\sum_{\ell=1}^{L} \lambda(\sigma_\ell)\mathbb{E}_{q_{\sigma_\ell}(\tilde{\mathbf{x}})}\left[||\nabla_{\tilde{\mathbf{x}}} \log q_{\sigma_\ell}(\tilde{\mathbf{x}}) - \boldsymbol{s}_\theta(\tilde{\mathbf{x}}, \sigma_\ell)||_2^2\right] \\
&= \frac{1}{L}\sum_{\ell=1}^{L} \lambda(\sigma_\ell)\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}, \mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[||\boldsymbol{s}_\theta(\mathbf{x} + \sigma_\ell\mathbf{z}, \sigma_\ell) - \mathbf{z}/\sigma_\ell||_2^2\right]
\end{aligned}$$

Réglages ?

- σ_1 distance maximale entre deux échantillons
- σ_L niveau de bruit invisible du point de vue de $\mathbf{x}$
- On doit fixer la séquence $\{\sigma_\ell\}_{\ell=1}^{L}$ et les poids $\lambda(\cdot)$
  - Ratio $\sigma_\ell/\sigma_{\ell+1}$ fixe recouvrement entre les différentes d.d.p.
  - $\lambda(\sigma_\ell) = \sigma_\ell^2$ balance les échelles et simplifie les expressions

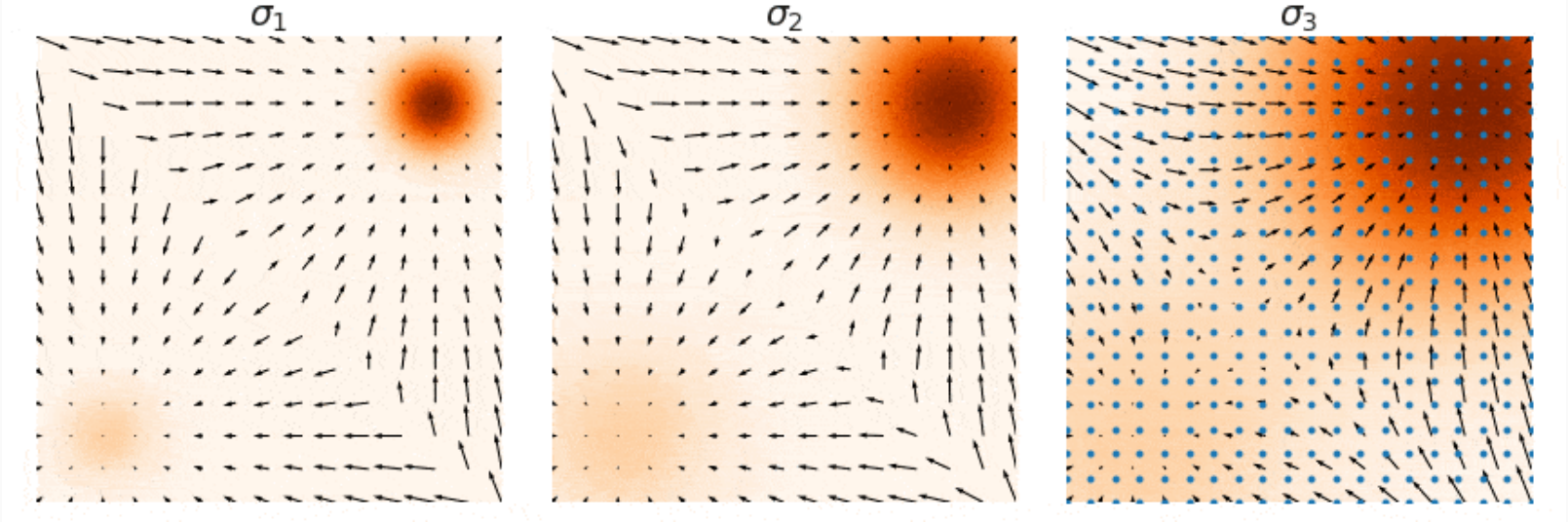
Dynamique de Langevin avec recuit simulé

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.

Initialize $\tilde{\mathbf{x}}_0$
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$ ▷ α_i is the step size.
  for $t \leftarrow 1$ to T **do**
    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$
  end for
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
end for
return $\tilde{\mathbf{x}}_T$

Dynamique de Langevin avec recuit simulé



En résumé

Estimation du score $\sim$ approche par maximum de vraisemblance

score matching

Noise perturbed score matching $\sim$ apprendre à d'ébruiter

Perturbation multi-échelle : apprendre sur plusieurs niveau de bruit

on surcharge un seul modèle $s_\theta(\mathbf{x}, \sigma)$ plutôt que d'en apprendre un par niveau de bruit $s_\theta^{\sigma_i}(\mathbf{x})$

Dynamique de Langevin : génération stochastique basée sur le score

Avec **recuit simulé** : raffiner séquentiellement l'échelle du bruit

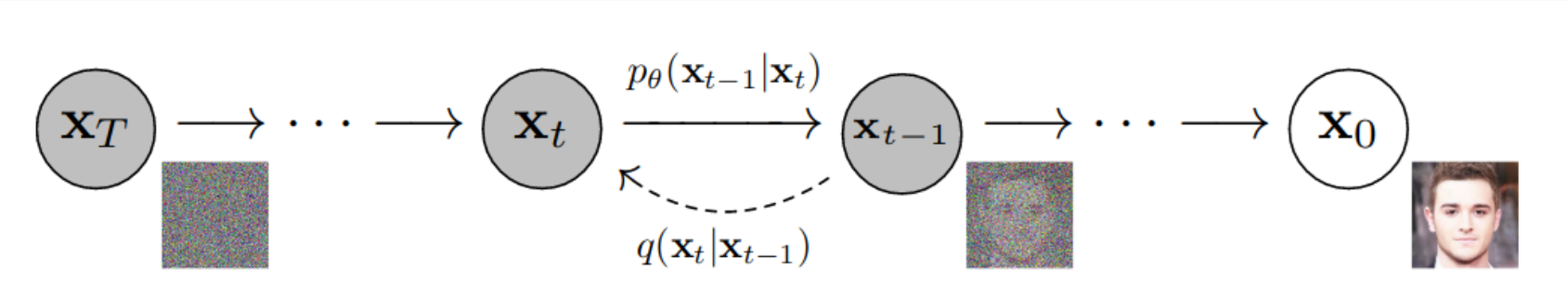
Désormais état de l'art pour la génération de données "continues" $\in \mathbb{R}^d$

Outline

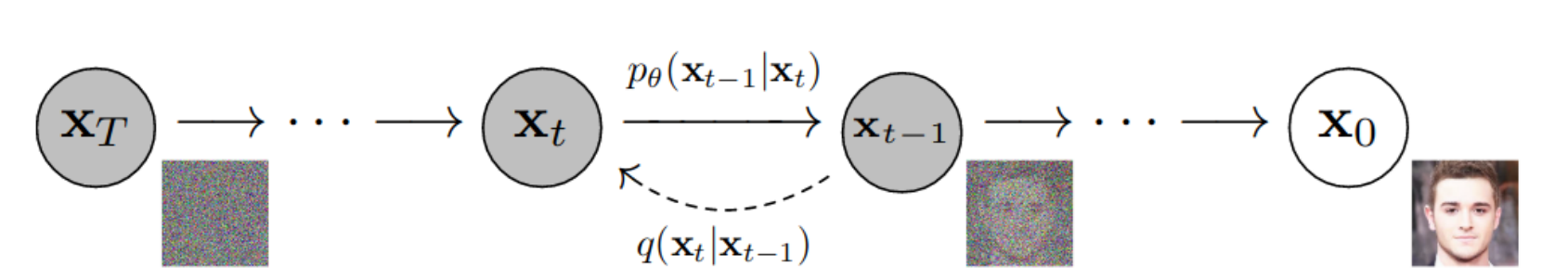
Modèles de diffusion (Ho et al. 2020)

Une formulation équivalente aux approches basées sur le score

- On ajoute progressivement du bruit à une image
- Le modèle apprend un débruiteur pour retrouver $\mathbf{x}_{t-1}$ depuis $\mathbf{x}_t$
- Echantilloner : appliquer T étapes de débruitage à $\mathbf{x}_T \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$



Processus forward et backward



Processus forward

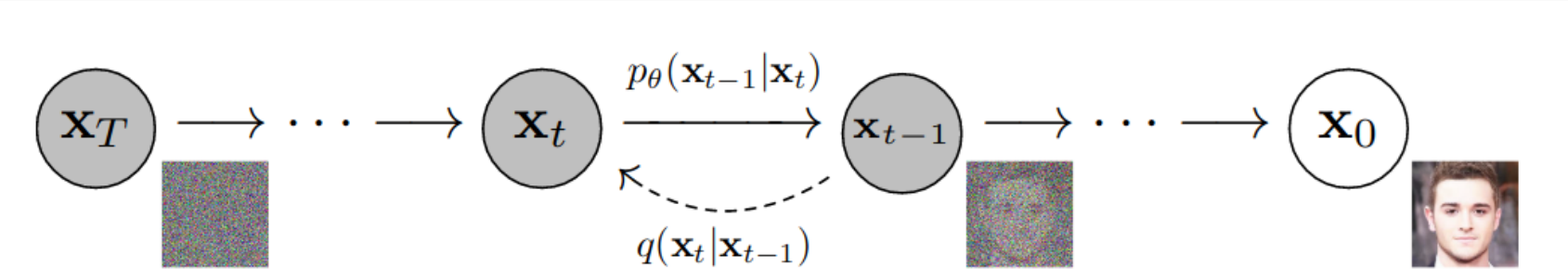
Bruite graduellement l'image. Définit une chaine de Markov :

- $q(\mathbf{x}_1, ..., \mathbf{x}_T|\mathbf{x}_0) = \prod_{t=1}^{T} q_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})$
- $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \boldsymbol{I})$

"Gaussien+Gaussien=Gaussien"

- $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$ $\qquad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$

Processus forward et backward



Processus backward

Débruite graduellement l'image. Aussi une chaine de Markov :

- $p(\mathbf{x}_T) = \mathcal{N}(0, \sigma^1 \boldsymbol{I})$
- $p_\theta(\mathbf{x}_0, ..., \mathbf{x}_T) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

p_θ est le modèle à apprendre, on va faire **un choix**

et le lien avec les VAEs !

Le point de vue VAE

Choix de l'encodeur-décodeur

- **"Encodeur"** (fixé, ne réduit pas la dimension)

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad \text{et} \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \boldsymbol{I})$$

- **Decodeur** (appris, choix de paramétrisation)

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad \text{et} \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$$

où on paramétrise $\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right)$

Entrainement du modèle

Evidential lower bound (ELBO)

On utilise la loss

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_{q(\mathbf{x}_0)\,q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$$

Pourles choix de paramétrisation, elle se réduit à

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}_0\sim(\mathbf{x}_0),t\sim\mathcal{U}(1,T),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\lambda_t\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right]$$

Conclusion : approche identique à l'estimation de score par débruitage

Modèle de diffusion (DDPM)

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

$\Leftrightarrow$ Estim. de score par débruitage

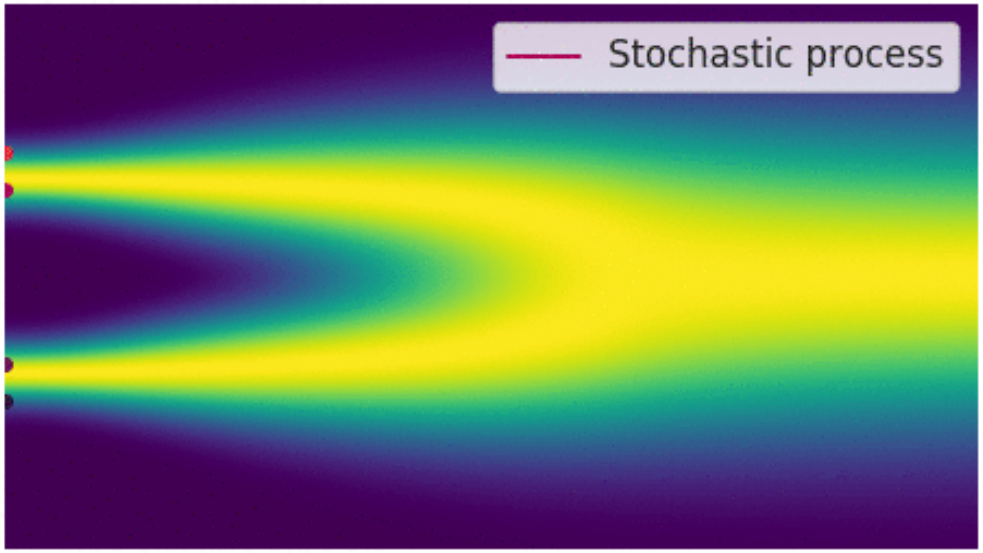
$$\sum_{\ell=1}^{L} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}, \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[||\epsilon_\theta(\mathbf{x} + \sigma_\ell \mathbf{z}, \sigma_\ell) - \mathbf{z}||_2^2 \right]$$

Echantilloner $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$

$\Leftrightarrow$ Langevin-recuit-simulé

$$\epsilon_\theta(\cdot, \sigma_i) := \sigma_i \boldsymbol{s}(\cdot, \sigma_i)$$

Le cas continu $T \to \infty$



Interprétable comme une **equation différentielle stochastique** (SDE)

$$\mathrm{d}\mathbf{x}_t = f(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t$$

Accélération de l'échantillonage

- Combinaisons de solveurs SDE + Langevin MCMC
- DDIM (utilisation d'ODE déterministes)

En résumé

Modèles de diffusion $\Leftrightarrow$ génération par modèles de score

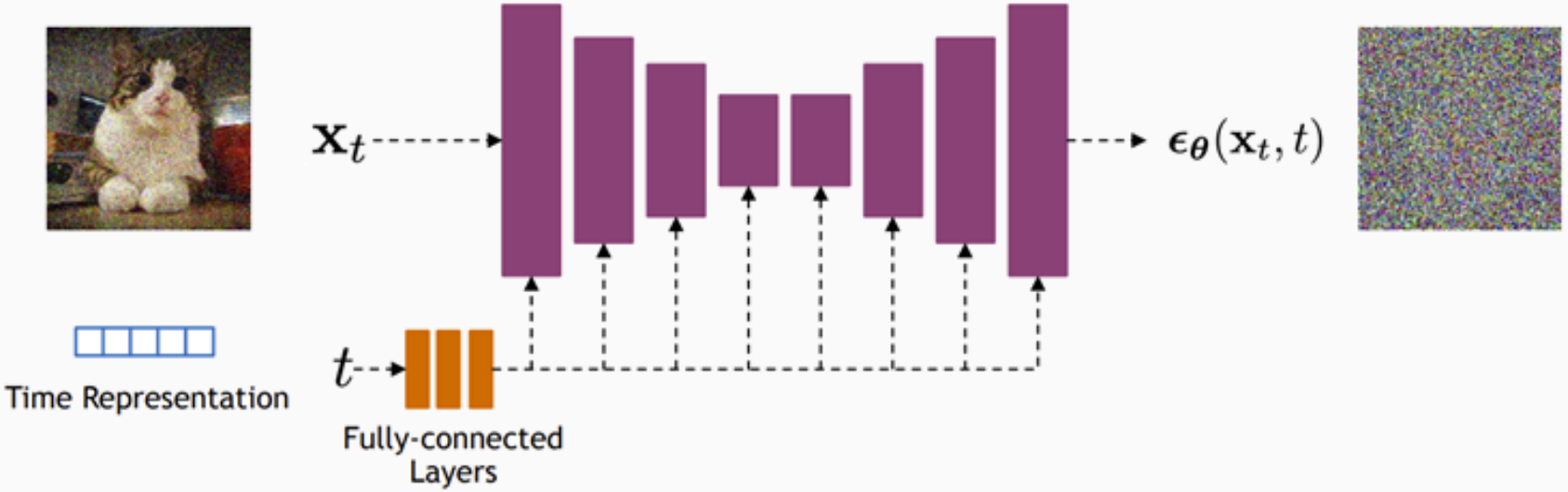
Estimation de score par débruitage $\Leftrightarrow$ lien avec une loss ELBO

Décoder successivement $\Leftrightarrow$ débruiter en suivant le score $\Leftrightarrow$ Langevin

Les deux points de vue apportent des outils complémentaires

Outline

Architectures



Deux options principales en computer vision :

- U-net
- Transformeurs

Conditionnement

Comment conditionner la diffusion ? (prompts)

- Architectures dédiées et entraînement sur données+lablels
- Adaptations d'architectures e.g., ControlNet (Zhang et al 2023)
- **Classifier guidance** : utiliser le Theorème de Bayes

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})}$$

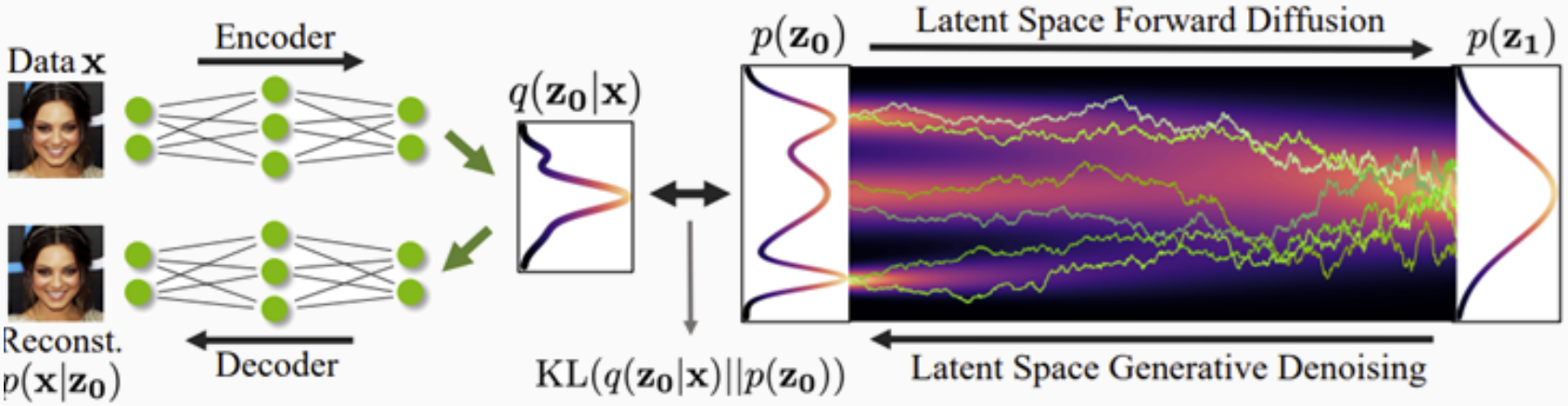
Sur le score

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log p(\mathbf{y})}_{=\mathbf{0}}$$

$\longrightarrow$ perturber le score par le gradient d'un classifieur pré-entraîné

Diffusion dans l'espace lattent

Combiner VAE et diffusion pour réduire la dimension du problème

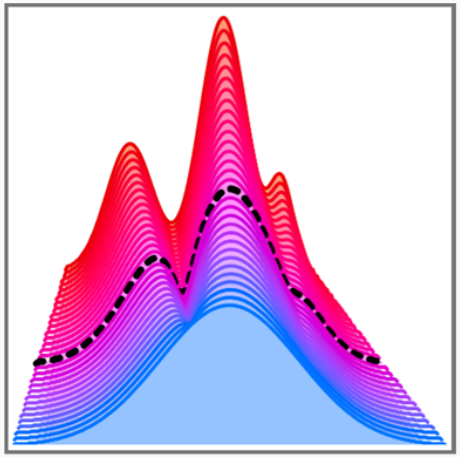


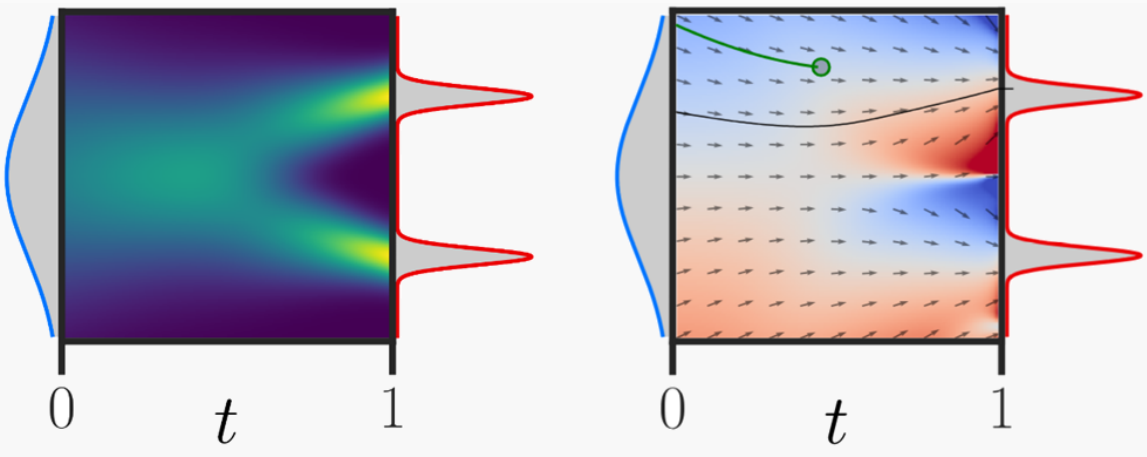
Différentes approches :

- Entraînement conjoint VAE + Diffusion
- D'abord VAE, puis modèle de diffusion dans l'espace lattent

(e.g. stable diffusion)

Modèles de flux (conditional flow matching)





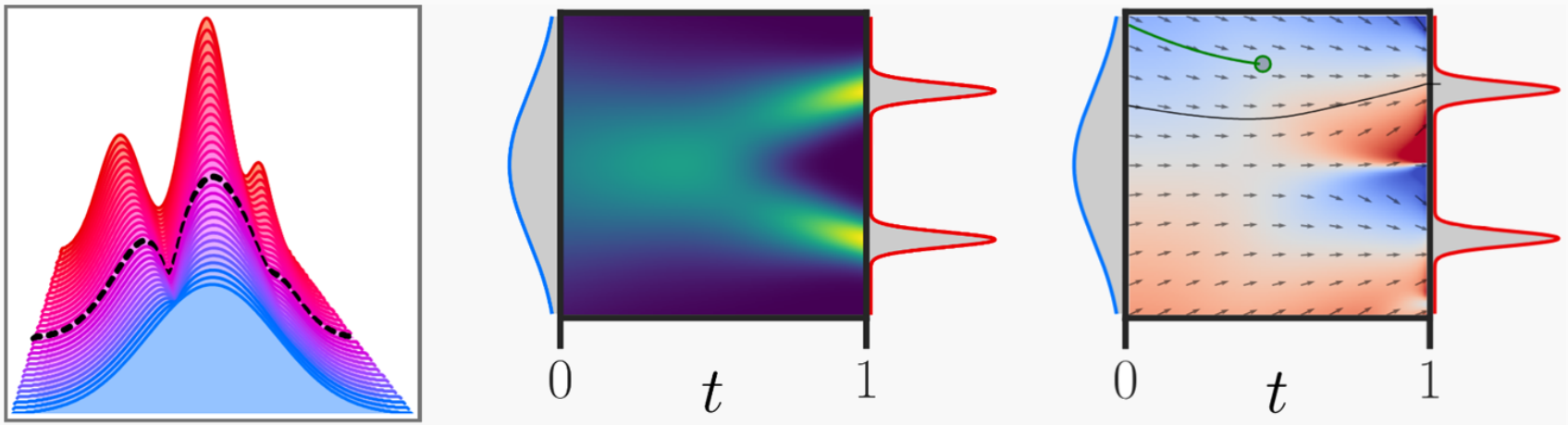
Apprendre un transport d'une distribution vers une autre

Génération = ODE (déterministe)

Proche de la diffusion, mais formalisme et entraînement différents

Introduction dans l'excellent [blog-post suivant]

Flow matching - goal



We seek a velocity field u_t such that

$$\frac{d}{dt}\varphi_t(x) = u_t(\varphi_t(x)), \quad \forall t \in [0,1] \tag{1}$$

for which $\varphi_0 = \pi$ and $\varphi_0 = p_{data}$

Flow matching - algorithm

Learn a function $v_\theta(t, x)$ with the algorithm

- Sample $z_0 \sim \mathcal{N}(0, \mathbf{I})$, $z_1 \sim p_{data}$, and $t \sim \mathcal{U}(0, 1)$
- Stochastic gradient descent on θ with loss

$$\left\| v_\theta(t, x_t) - \frac{z_1 - z_0}{1 - t} \right\|^2$$

Looks like score matching, but we predict a direction rather than noise

What does it do on average?

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{\substack{x_0, x_1 \sim p(x_0, x_1) \\ t \sim U([0,1])}} \left\| v_\theta(t, x_t) - u_t(x_t \mid x_0, x_1) \right\|^2$$

Visualize with the following [blog-post] or [playground]

Flow matching - sampling

Sampling from the model

$$\hat{x}_1 = \varphi_{t=1}(x_0) = \text{EDO}^{v_\theta}(x_0, 0 \to 1)$$

where $\text{EDO}^{v_\theta}(\cdot, t_0 \to t_1)$ solves (1) from t_0 to t_1 with $v_\theta(t, \cdot)$ instead of u_t