LINKÖPING UNIVERSITY

Diffusion models, LLMs, and SMC

Filip Ekström Kelvinius

2025-02-28

Outline of Lecture 17

Aim: See examples how the general SMC framework can be used within the world of **Deep Generative Models**

Outline:

1. (Generative) Diffusion models
2. Image inpainting as a Bayeisan inverse problem
3. SMC for image inpainting
4. (Generative) Autoregressive models
5. SMC for LLMs

Generative modeling

Given unlabelled dataset

$$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$$

estimate data distribution, i.e.,

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

In particular, enable sampling

$$\mathbf{x}_{\text{new}} \sim p_\theta(\mathbf{x})$$

Deep generative models: $p_\theta(\mathbf{x})$ implictly defined by a neural network

Diffusion Models

(Generative) Diffusion models

$$\mathbf{x}_0 \sim \mathcal{N}(0, \sigma_0^2 I) \longleftarrow \mathbf{x}_1 \longleftarrow \mathbf{x}_2 \longleftarrow \cdots \longleftarrow \mathbf{x}_{K-2} \longleftarrow \mathbf{x}_{K-1} \longleftarrow \mathbf{x}_K \sim \mathcal{D}$$

$$\mathbf{x}_0 \sim \mathcal{N}(0, \sigma_0^2 I) \longrightarrow \mathbf{x}_1 \longrightarrow \mathbf{x}_2 \longrightarrow \cdots \longrightarrow \mathbf{x}_{K-2} \longrightarrow \mathbf{x}_{K-1} \longrightarrow \mathbf{x}_K \sim p_\theta(\mathbf{x}_K)$$

Illustration adapted from slides by Fredrik Lindsten

(Generative) Diffusion models

Start by assuming a fixed data-to-noise process (data $\mathbf{x} \equiv \mathbf{x}_K$)

$$q(\mathbf{x}_{0:K}) = q(\mathbf{x}_K) \prod_{k=0}^{K-1} q(\mathbf{x}_k|\mathbf{x}_{k+1})$$

where

$$q(\mathbf{x}_k|\mathbf{x}_{k+1}) = \mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k+1}, \beta_k I).$$

Note: $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_K)$ is available in closed form.

Generation via a procedure

$$p_\theta(\mathbf{x}_{0:K}) = p_\theta(\mathbf{x}_0) \prod_{k=1}^{K} p_\theta(\mathbf{x}_k|\mathbf{x}_{k-1}),$$

where $p_\theta(\mathbf{x}_0) \approx q(\mathbf{x}_0)$ and $p_\theta(\mathbf{x}_k|\mathbf{x}_{k-1}) \approx q(\mathbf{x}_k|\mathbf{x}_{k-1})$

Parametrization of the generative procedure

Initial distribution approximated by $p_\theta(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0|0, \sigma_0^2 I)$.

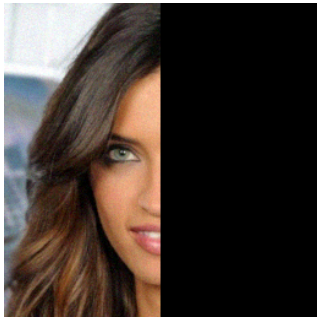
Transitions $q(\mathbf{x}_k|\mathbf{x}_{k-1}) = \int q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_K) q(\mathbf{x}_K|\mathbf{x}_{k-1}) d\mathbf{x}_K$ intractable

Neural network *reconstructs* $\mathbf{x}_K$ as $f_\theta(\mathbf{x}_{k-1})$, and with approximation $p_\theta(\mathbf{x}_K|\mathbf{x}_{k-1}) = \delta_{f_\theta(\mathbf{x}_{k-1})}(\mathbf{x}_K)$,

$$\begin{aligned} p_\theta(\mathbf{x}_k|\mathbf{x}_{k-1}) &= \int q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_K) p_\theta(\mathbf{x}_K|\mathbf{x}_{k-1}) d\mathbf{x}_K, \\ &= q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_K = f_\theta(\mathbf{x}_{k-1})) \end{aligned}$$

Image inpainting as a Bayesian inverse problem

Task: fill in the missing pixels



But many potential possibilities...

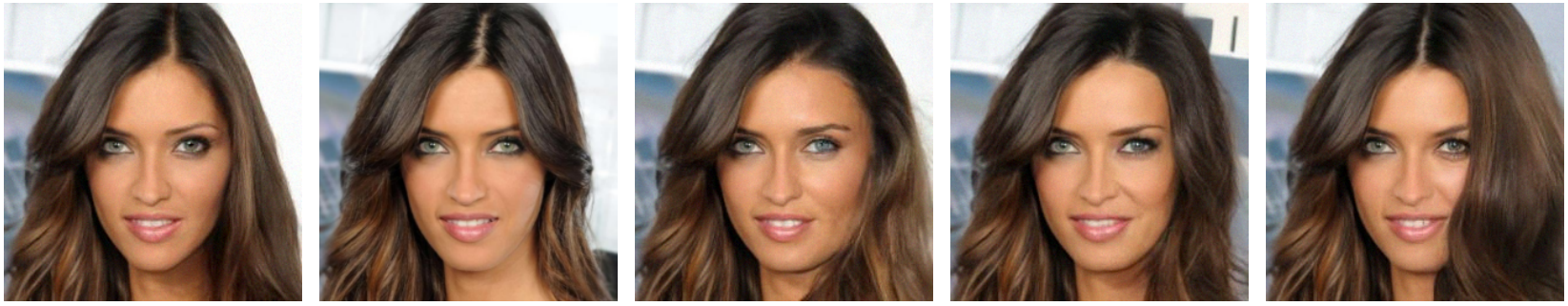


Image inpainting as a Bayesian inverse problem

Probabilistic formulation

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

where $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|A\mathbf{x}, \sigma_y^2 I)$

Use diffusion model $p_\theta(\mathbf{x}_K)$ as prior $p(\mathbf{x})$!

Bayesian inverse problem with diffusion prior: Sample from the posterior

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

The sequential structure of the diffusion model makes SMC a suitable framework

Target distributions

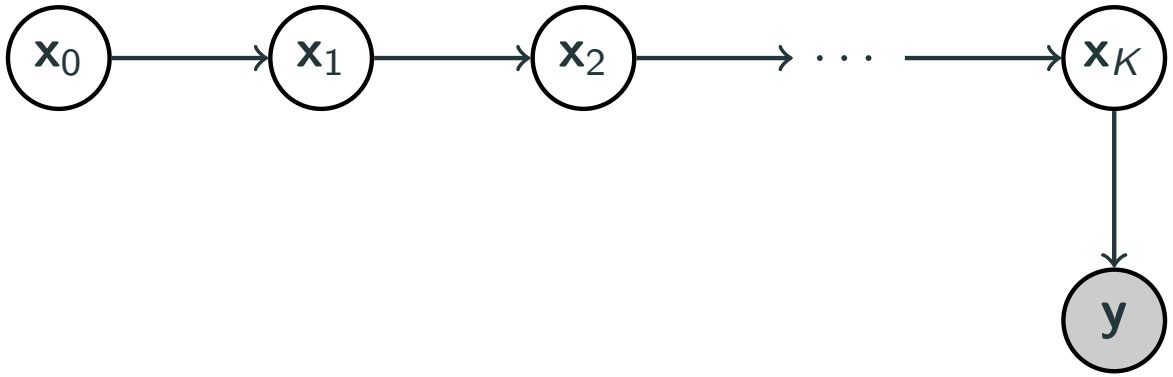
Recall:

As long as $\int \pi_K(\mathbf{x}_{0:K}) d\mathbf{x}_{0:K-1} = p(\mathbf{x}_K|\mathbf{y})$, the SMC algorithm is consistent. Targets for $k = 0, \ldots, K-1$ are auxiliary quantities.

Can design $\tilde{\pi}_k(\mathbf{x}_{0:k})$ however we want, as long as $\tilde{\pi}_K(\mathbf{x}_{0:K}) = p(\mathbf{y}|\mathbf{x}_K)p_\theta(\mathbf{x}_{0:K})$

Target distributions

First attempt:



Unnormalized targets:

$$\tilde{\pi}_k(\mathbf{x}_{0:k}) = \begin{cases} p_\theta(\mathbf{x}_0) \prod_{l=1}^{k} p_\theta(\mathbf{x}_l|\mathbf{x}_{l-1}) & k < K \\ p(\mathbf{y}|\mathbf{x}_K) p_\theta(\mathbf{x}_0) \prod_{l=1}^{K} p_\theta(\mathbf{x}_l|\mathbf{x}_{l-1}) & k = K \end{cases}$$

Not very efficient

Target distributions

Second attempt:

$$\tilde{\pi}_k(\mathbf{x}_{0:k}) = p(\mathbf{y}|\mathbf{x}_k)p_\theta(\mathbf{x}_0)\prod_{l=1}^{k} p_\theta(\mathbf{x}_l|\mathbf{x}_{l-1})$$

Likelihoods $p(\mathbf{y}|\mathbf{x}_k)$ intractable for $k < K$

Need to find approximations $\hat{p}(\mathbf{y}|\mathbf{x}_k)$

Can approximate likelihoods as we want, as long as $\hat{p}(\mathbf{y}|\mathbf{x}_K) \coloneqq p(\mathbf{y}|\mathbf{x}_K)$

Target distributions – likelihood approximations

TDS: Use reconstrution $f_\theta(\mathbf{x}_k)$:

$$\hat{p}(\mathbf{y}|\mathbf{x}_k) = p(\mathbf{y}|\mathbf{x}_K = f_\theta(\mathbf{x}_k))$$

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P. Cunningham. **Practical and Asymptotically Exact Conditional Sampling in Diffusion Models.** *NeurIPS*, 2023.

DDSMC: Use approximation $\hat{p}_\theta(\mathbf{x}_K|\mathbf{x}_k) = \mathcal{N}(f_\theta(\mathbf{x}_k), \rho_k^2 I)$

$$\hat{p}(\mathbf{y}|\mathbf{x}_k) = \int p(\mathbf{y}|\mathbf{x}_K)\hat{p}_\theta(\mathbf{x}_K|\mathbf{x}_k)d\mathbf{x}_K = \mathcal{N}(Af_\theta(\mathbf{x}_k), \sigma_y^2 I + \rho_k^2 AA^T)$$

Filip Ekström Kelvinius, Zheng Zhao, and Fredrik Lindsten. **Solving Linear-Gaussian Bayesian Inverse Problems with Decoupled Diffusion Sequential Monte Carlo.** *arXiv*, 2025.

Proposals

Design proposal informed by $\mathbf{y}$ for better efficiency

TDS:

$$r_k(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}) = \mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k-1} + \beta_k \hat{s}, \hat{\beta}_k I)$$

where $\hat{s} = s_\theta(\mathbf{x}_{k-1}) + \nabla_{\mathbf{x}_{k-1}} \log \hat{p}(\mathbf{y}|\mathbf{x}_{k-1})$ and
$s_\theta(\mathbf{x}_{k-1}) \approx \nabla_{\mathbf{x}_{k-1}} \log q(\mathbf{x}_{k-1})$ is approximation of unconditional score (a neural network)

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P. Cunningham. **Practical and Asymptotically Exact Conditional Sampling in Diffusion Models.** *NeurIPS*, 2023.

Proposals, DDSMC

Start with approximation of posterior $\hat{p}_\theta(\mathbf{x}_K|\mathbf{x}_{k-1}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}_K)\hat{p}_\theta(\mathbf{x}_K|\mathbf{x}_k)$ (Gaussian). Then "push back" to step k:

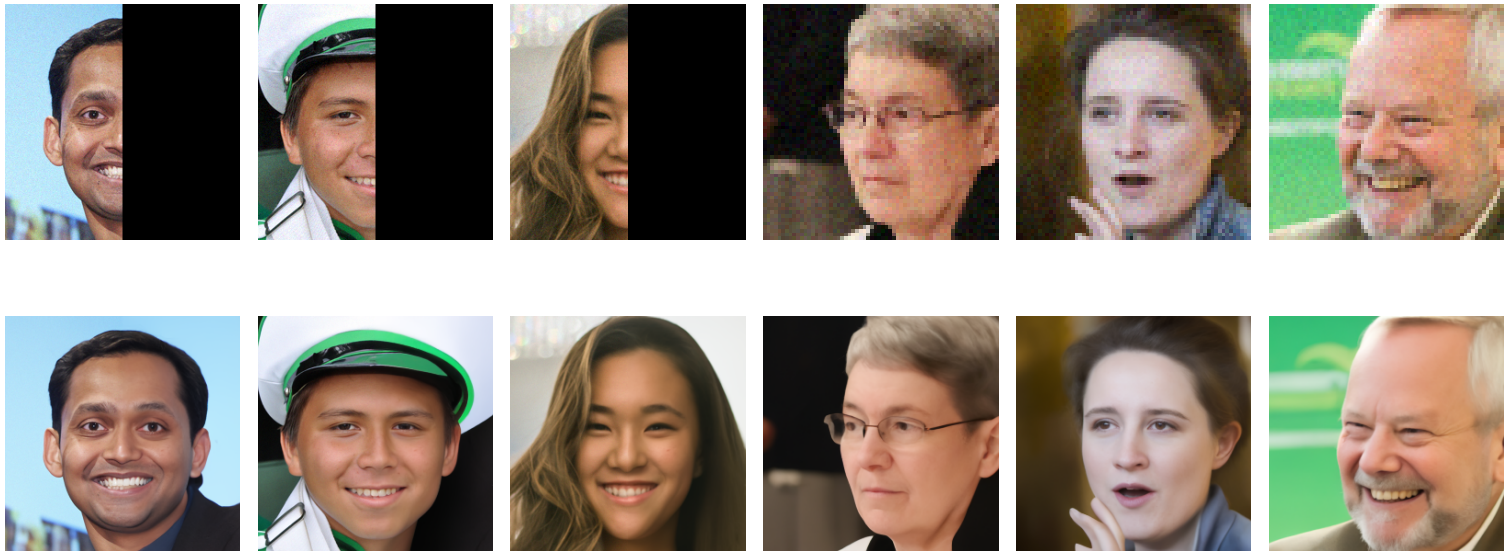
$$r_k(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}) = \int q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_K)\hat{p}(\mathbf{x}_K|\mathbf{x}_{k-1}, \mathbf{y})d\mathbf{x}_K$$

Somewhat simplified, see paper for details

Filip Ekström Kelvinius, Zheng Zhao, and Fredrik Lindsten. **Solving Linear-Gaussian Bayesian Inverse Problems with Decoupled Diffusion Sequential Monte Carlo.** *arXiv*, 2025.

Some DDSMC results

Same prior, different likelihoods



Autoregressive models/LLMs

(Generative) Autoregressive Models

Intuition: Can always decompose
$p(x_1, x_2, x_3) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)$.

(Generative) Autoregressive model:

$$p_\theta(\mathbf{x}_{1:K}) = \prod_{k=1}^{K} p_\theta(\mathbf{x}_k|\mathbf{x}_{1:k-1}),$$

conditionals $p_\theta(\cdot|\cdot)$ defined by neural network.

LLMs: full sequence $\mathbf{x}_{1:K}$ is of interest, and $\mathbf{x}_{1:k} \in \{0, \ldots, S\}^k$

Incorporating constraints

We want to incoporate some constraints via a potential $\Phi(\mathbf{x}_{1:K})$,

$$p_\theta^\Phi(\mathbf{x}_{1:K}) = \frac{1}{Z^\Phi} p_\theta(\mathbf{x}_{1:K})\Phi(\mathbf{x}_{1:K})$$

Analogy to diffusion: $\Phi(\mathbf{x}_{1:K}) = p(\mathbf{y}|\mathbf{x}_K)$

Can use SMC with target $\pi_K(\mathbf{x}_{1:K}) = p_\theta^\Phi(\mathbf{x}_{1:K})$.

A note on "twisting"

For efficiency, want to design intermediate targets

$$\tilde{\pi}_k(\mathbf{x}_{1:k}) = \hat{\Phi}(\mathbf{x}_{1:k}) p_\theta(\mathbf{x}_{1:k})$$

Compare with diffusion: $\hat{\Phi}(\mathbf{x}_{1:k}) = \hat{p}(\mathbf{y}|\mathbf{x}_k)$

Twisting: Multiplying the base distribution with a potential $\hat{\Phi}(\mathbf{x}_{1:k})$ also for intermediate ($k < K$) targets is sometimes called "twisting", or "twisted SMC" in litterature. New term, but not a new SMC-method: the intermediate targets are design choices anyway.

Product-of-experts, targets

Product-of-experts: $\Phi(\mathbf{x}_{1:K}) = \prod_{i=1}^{P} \psi_i(\mathbf{x}_{1:K})$.

Assuming $\hat{\psi}_i$ can be evaluated on partial sequences $\mathbf{x}_{1:k}$, define targets

$$\tilde{\pi}_k(\mathbf{x}_{1:k}) = p_\theta(\mathbf{x}_{1:k}) \prod_{i=1}^{P} \hat{\psi}_i(\mathbf{x}_{1:k})$$

Again, no matter how the partial potentials $\hat{\psi}_i$ are defined, the algorithm will be consistent

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Alexander K. Lew, Tim Vieira, Timothy J. O'Donnell. **Syntactic and Semantic Control of Large Language Models via Sequential Monte Carlo.** *ICLR*, 2025.

Product-of-experts, proposal

Tempting to use proposal

$$r_t^*(\mathbf{x}_k|\mathbf{x}_{1:k-1}) = \frac{1}{Z_k^*(\mathbf{x}_{1:k-1})} p_\theta(\mathbf{x}_k|\mathbf{x}_{1:k-1}) \prod_{i=1}^{P} \hat{\psi}_i(\mathbf{x}_{1:k})$$

But the potentials $\hat{\psi}_i$ can be more or less expensive to evaluate for each value of $\mathbf{x}_k$. Instead use

$$r_t(\mathbf{x}_k|\mathbf{x}_{1:k-1}) = \frac{1}{Z_k(\mathbf{x}_{1:k-1})} p_\theta(\mathbf{x}_k|\mathbf{x}_{1:k-1}) \prod_{j \in \Phi_{\text{eff}}} \hat{\psi}_j(\mathbf{x}_{1:k})$$

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Alexander K. Lew, Tim Vieira, Timothy J. O'Donnell. **Syntactic and Semantic Control of Large Language Models via Sequential Monte Carlo.** *ICLR*, 2025.

Learning potentials

If the potential $\Phi(\cdot)$ requires full trajectories, the true distributions

$$\pi_k^*(\mathbf{x}_{1:k}) \propto \sum_{\mathbf{x}_{k+1:K}} \Phi(\mathbf{x}_{1:K}) p_\theta(\mathbf{x}_{1:K})$$

are intractable.

Option: approximate $\pi_k^*(\mathbf{x}_{1:k})$ with $\pi_k(\mathbf{x}_{1:k}) \propto \hat{\Phi}^\eta(\mathbf{x}_{1:k}) p_\theta(\mathbf{x}_{1:k})$, and learn $\hat{\Phi}^\eta$ by minimizing

$$\sum_{k=1}^{K} D_{KL}(\pi_k^*(\mathbf{x}_{1:k}) || \pi_k(\mathbf{x}_{1:k})),$$

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, Roger Baker Grosse. **Probabilistic Inference in Language Models via Twisted Sequential Monte Carlo.** *ICML*, 2024.

Proposal with learned potentials

Parametrizing $\hat{\Phi}^{\eta}$ such that it is possible to (efficiently) compute proposal

$$r_t(\mathbf{x}_k|\mathbf{x}_{1:k-1}) = \frac{1}{Z_k(\mathbf{x}_{1:k-1})} p_\theta(\mathbf{x}_k|\mathbf{x}_{1:k-1})\hat{\Phi}^{\eta}(\mathbf{x}_{1:k})$$

gives weights independent of $\mathbf{x}_k$, opens up for a fully-adapted SMC algorithm

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, Roger Baker Grosse. **Probabilistic Inference in Language Models via Twisted Sequential Monte Carlo.** *ICML*, 2024.

Some LLM results

Method	Score			
	Goal inference	Molecular synthesis	Data science	Text-to-SQL
LM	0.063 (0.05, 0.08)	0.132 (0.12, 0.15)	0.213 (0.19, 0.24)	0.531 (0.51, 0.55)
w/ grammar constraint (Locally-constrained Decoding)	0.086 (0.07, 0.11)	0.189 (0.17, 0.21)	-	0.559 (0.54, 0.58)
w/ grammar constraint, weight correction (Grammar-only IS)	0.083 (0.06, 0.11)	0.228 (0.21, 0.25)	-	0.597 (0.57, 0.62)
w/ grammar constraint, potential (Sample-Rerank)	0.289 (0.24, 0.34)	0.392 (0.36, 0.42)	-	0.581 (0.56, 0.60)
w/ grammar constraint, correction, and resampling (Grammar-only SMC)	0.401 (0.34, 0.46)	0.205 (0.18, 0.23)	-	0.596 (0.57, 0.62)
w/ grammar constraint, potential, and correction (Full IS)	0.257 (0.21, 0.31)	0.404 (0.37, 0.44)	0.346 (0.31, 0.39)	**0.618** (0.59, 0.64)
w/ grammar constraint, potential, correction, and resampling (Full SMC)	**0.419** (0.37, 0.48)	**0.577** (0.56, 0.59)	**0.407** (0.36, 0.45)	**0.620** (0.60, 0.64)

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Alexander K. Lew, Tim Vieira, Timothy J. O'Donnell. **Syntactic and Semantic Control of Large Language Models via Sequential Monte Carlo.** *ICLR*, 2025.

Conclusion

The sequential structure of modern deep generative models opens up the possibility to combine them with SMC

Approximations when designing intermediate targets, but:

As long as $\pi_K(\mathbf{x}_{1:K})$ (or its marginal) coincides with the desired distribution, SMC provides asymptotic guarantees

filip.ekstrom@liu.se