# PML session 4. Sequential Monte Carlo

## Probabilistic Machine Learning Reading Group

Mario Chacón-Falcón

December 17, 2025

Institute of Mathematical Sciences (ICMAT-CSIC)

## Logistics for the upcoming sessions

- Next session:
  - **Special session: VI + MCMC**
  - January 14th, 2026
  - *Max Hird (PostDoc @ University of Waterloo)*
- Following section:
  - **Bayesian Neural Networks (Ch. 17)**
  - January 21st, 2026
  - *Daniel Corrales (PhD @ Institute of Mathematical Sciences)*
- After that:
  - Every other week (so 4/2, 18/2, 4/3, ...)

# Contents

# Contents

# Sequential Monte Carlo (SMC): Estimating a Sequence of Distributions

- Goal of SMC: approximate a sequence of related distributions

$$\pi_t(z_{1:t}) = \frac{1}{Z_t}\tilde{\gamma}(z_{1:t})$$

  for $t = 1 : T$.
- Such sequences arise in:
  - **State Space Models (SSMs)**: $\pi_t(z_{1:t}) \propto p(z_{1:t}|y_{1:t})$
  - **SMC Samplers**: tempered or bridging distributions between prior $\pi_0$ and posterior $\pi_T$.
- We focus on:
  1. SMC for inference in SSMs (particle filtering)
  2. SMC samplers for general Bayesian inference

# Applications of Sequential Monte Carlo

1. **SMC for SSMs**
   - **Object Tracking**: Estimating position/velocity of aircraft or animals using noisy radar or GPS data.
   - **Stocks**: Stochastic volatility modelling for option pricing.

2. **SMC Samplers**
   - **Epidemiology**: Inferring transmission rates for disease outbreaks (e.g., COVID-19) using complex simulators.
   - **Reinforcement Learning**: Policy search and planning in sparse-reward environments.
   - **Climate**: Calibrating parameters of large-scale climate models.

# Contents

## State Space Models (SSMs)

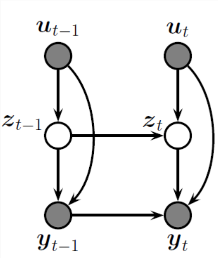
A SSM can be represented as a probabilistic model of the form:

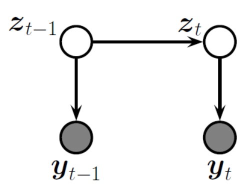
$$p(z_t|z_{t-1}, u_t) = p(z_t|f(z_{t-1}, u_t)) \qquad \text{(Transition model)}$$

$$p(y_t|z_t, u_t, y_{1:t-1}) = p(y_t|h(z_t, u_t, y_{1:t-1})) \qquad \text{(Observation model)}$$

**We focus on a simplified case:**

$$p(z_{1:T}, y_{1:T}) = p(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(y_t \mid z_t)$$



# Bayesian Filtering equations

We want to compute the belief state $p(z_t \mid y_{1:t})$ given the prior belief from the previous step, $p(z_{t-1}|y_{1-t-1})$:

**Prediction step (Chapman-Kolmogorov equation):**

$$p(z_t \mid y_{1:t-1}) = \int p(z_t \mid z_{t-1})\, p(z_{t-1} \mid y_{1:t-1})\, dz_{t-1}$$

# Bayesian Filtering equations

We want to compute the belief state $p(z_t \mid y_{1:t})$ given the prior belief from the previous step, $p(z_{t-1}|y_{1-t-1})$:

**Prediction step (Chapman-Kolmogorov equation):**

$$p(z_t \mid y_{1:t-1}) = \int p(z_t \mid z_{t-1})\, p(z_{t-1} \mid y_{1:t-1})\, dz_{t-1}$$

**Update step (Bayes' rule):**

$$p(z_t \mid y_{1:t}) = \frac{p(y_t \mid z_t)\, p(z_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}$$

# Linear–Gaussian State Space Models

A **linear dynamical system** is a special case of an SSM where both the transition and observation models are linear with Gaussian noise:

**State transition model**

$$z_t = F_t z_{t-1} + b_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, Q_t)$$

**Observation model**

$$y_t = H_t z_t + d_t + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, R_t)$$

# Linear–Gaussian State Space Models

A **linear dynamical system** is a special case of an SSM where both the transition and observation models are linear with Gaussian noise:

**State transition model**

$$z_t = F_t z_{t-1} + b_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, Q_t)$$

**Observation model**

$$y_t = H_t z_t + d_t + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, R_t)$$

Equivalently,

$$p(z_t \mid z_{t-1}) = \mathcal{N}(F_t z_{t-1} + b_t,\, Q_t), \qquad p(y_t \mid z_t) = \mathcal{N}(H_t z_t + d_t,\, R_t).$$

**Key property:** If $p(z_{t-1} \mid y_{1:t-1})$ is Gaussian, then $p(z_t \mid y_{1:t})$ is also Gaussian

This closure property enables **exact Bayesian filtering**.

# Kalman Filter: Predict and Update

Assume the filtering distribution at time $t-1$ is Gaussian:

$$p(z_{t-1} \mid y_{1:t-1}) = \mathcal{N}(\mu_{t-1|t-1}, \Sigma_{t-1|t-1})$$

**Predict step (time update)**

$$\mu_{t|t-1} = F_t \mu_{t-1|t-1} + b_t$$

$$\Sigma_{t|t-1} = F_t \Sigma_{t-1|t-1} F_t^\top + Q_t$$

# Kalman Filter: Predict and Update

Assume the filtering distribution at time $t-1$ is Gaussian:

$$p(z_{t-1} \mid y_{1:t-1}) = \mathcal{N}(\mu_{t-1|t-1}, \Sigma_{t-1|t-1})$$

**Predict step (time update)**

$$\mu_{t|t-1} = F_t \mu_{t-1|t-1} + b_t$$

$$\Sigma_{t|t-1} = F_t \Sigma_{t-1|t-1} F_t^\top + Q_t$$

**Update step (measurement update)**

$$\hat{y}_t = H_t \mu_{t|t-1} + d_t$$

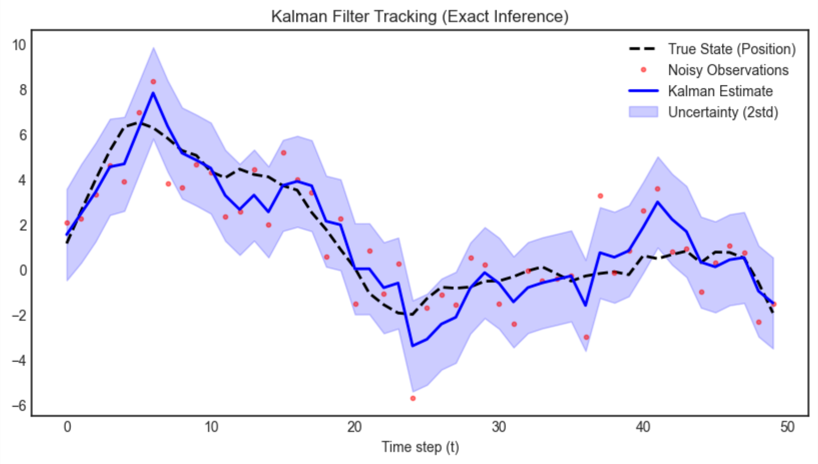
$$S_t = H_t \Sigma_{t|t-1} H_t^\top + R_t$$

$$K_t = \Sigma_{t|t-1} H_t^\top S_t^{-1}$$

$$\mu_{t|t} = \mu_{t|t-1} + K_t (y_t - \hat{y}_t)$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - K_t S_t K_t^\top$$

# Linear Gaussian Case: Example



# Contents

# Importance Sampling (Self-Normalized IS)

Goal: estimate expectations under a target distribution $\pi_t$:

$$\mathbb{E}_{\pi_t}[\varphi_t(z_{1:t})] = \int \varphi_t(z_{1:t})\, \pi_t(z_{1:t})\, dz_{1:t}, \qquad \pi_t(z_{1:t}) = \frac{\tilde{\gamma}_t(z_{1:t})}{Z_t}.$$

Using a proposal $q_t(z_{1:t})$, with $\mathrm{supp}(\pi) \subseteq \mathrm{supp}(q)$, we can rewrite:

$$\mathbb{E}_{\pi_t}[\varphi_t(z_{1:t})] = \frac{\int \left[\frac{\tilde{\gamma}(z_{1:t})}{q(z_{1:t})}\varphi_t(z_{1:t})\right] q(z_{1:t})\, \mathrm{d}z_{1:t}}{\int \left[\frac{\tilde{\gamma}(z_{1:t})}{q(z_{1:t})}\right] q(z_{1:t})\, \mathrm{d}z_{1:t}}$$

# Importance Sampling (Self-Normalized IS)

Drawing $z_{1:t}^{(i)} \sim q_t(z_{1:t}), \quad i = 1, \dots, N_s$, we estimate

$$\mathbb{E}_{\pi_t}[\varphi_t(z_{1:t})] \;\approx\; \sum_{i=1}^{N_s} W_t^{(i)}\, \varphi_t(z_{1:t}^{(i)}),$$

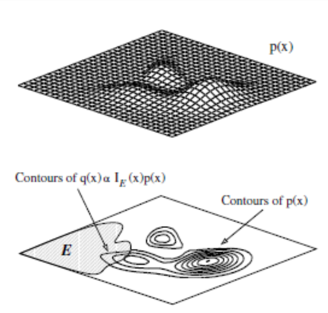
Where:

$$\tilde{w}_t^{(i)} = \frac{\tilde{\gamma}_t(z_{1:t}^{(i)})}{q_t(z_{1:t}^{(i)})}, \qquad W_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}}, \qquad \hat{Z}_t = \frac{1}{N_s}\sum_{i=1}^{N_s} \tilde{w}_t^{(i)}.$$

# Importance Sampling (Self-Normalized IS)

To approximate the target distribution:

$$\pi_t(z_{1:t}) \approx \sum_{i=1}^{N_s} W_t^{(i)}\, \delta(z_{1:t} - z_{1:t}^{(i)}).$$



# Sequential Importance Sampling (SIS)

**Key idea:** exploit the autoregressive (online) structure of the problem.

**Autoregressive proposal:**

$$q_t(z_{1:t}) = q_{t-1}(z_{1:t-1})\, q_t(z_t \mid z_{1:t-1})$$

Given particles $\{z_{1:t-1}^{(i)}\}$, extend each trajectory by sampling

$$z_t^{(i)} \sim q_t(z_t \mid z_{1:t-1}^{(i)}).$$

## Sequential Importance Sampling (SIS)

**Key idea:** exploit the autoregressive (online) structure of the problem.

**Autoregressive proposal:**

$$q_t(z_{1:t}) = q_{t-1}(z_{1:t-1})\, q_t(z_t \mid z_{1:t-1})$$

Given particles $\{z_{1:t-1}^{(i)}\}$, extend each trajectory by sampling

$$z_t^{(i)} \sim q_t(z_t \mid z_{1:t-1}^{(i)}).$$

**Recursive weight update:**

$$\tilde{w}_t(z_{1:t}) = \frac{\tilde{\gamma}_t(z_{1:t})}{q_t(z_{1:t})} = \tilde{w}_{t-1}(z_{1:t-1}) \cdot \underbrace{\frac{\tilde{\gamma}_t(z_{1:t})}{\tilde{\gamma}_{t-1}(z_{1:t-1})\, q_t(z_t \mid z_{1:t-1})}}_{\textbf{incremental importance weight } \alpha_t(z_{1:t})}$$

# Sequential Importance Sampling (SIS)

**Special case: State Space Models**

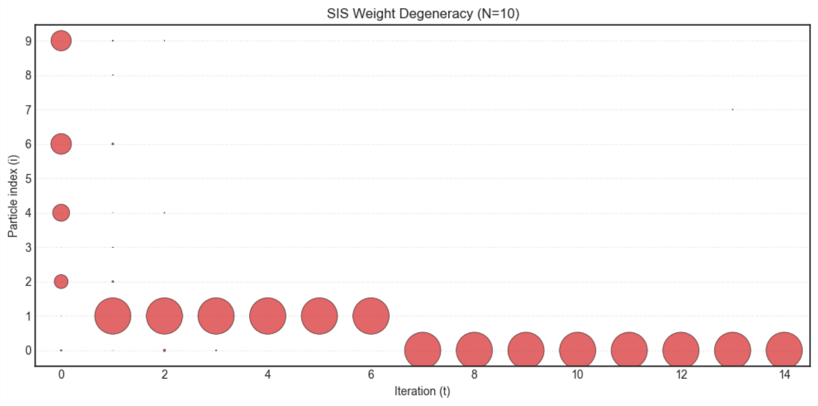
$$\tilde{\gamma}_t(z_{1:t}) = p(z_{1:t}, y_{1:t}) = p(y_t \mid z_{1:t})\, p(z_t \mid z_{1:t-1})\, \tilde{\gamma}_{t-1}(z_{1:t-1})$$

Hence,

$$\alpha_t(z_{1:t}) = \frac{p(y_t \mid z_{1:t})\, p(z_t \mid z_{1:t-1})}{q_t(z_t \mid z_{1:t-1})}.$$

# Sequential Importance Sampling (SIS)

**Limitation:** Weight degeneracy



# Sequential Importance Sampling with Resampling (SISR)

**SIS proposal:**

$$q_t^{\mathsf{SIS}}(z_{1:t}) = q_{t-1}(z_{1:t-1})\, q_t(z_t \mid z_{1:t-1}),$$

**SISR proposal:**

$$q_t^{\mathsf{SISR}}(z_{1:t}) = \hat{\pi}_{t-1}(z_{1:t-1})\, q_t(z_t \mid z_{1:t-1}),$$

where $\hat{\pi}_{t-1}(z_{1:t-1}) = \sum_i W_{t-1}^{(i)} \delta(z_{1:t-1} - z_{1:t-1}^{(i)})$.

## Sequential Importance Sampling with Resampling (SISR)

**SIS proposal:**

$$q_t^{\mathsf{SIS}}(z_{1:t}) = q_{t-1}(z_{1:t-1})\, q_t(z_t \mid z_{1:t-1}),$$

**SISR proposal:**

$$q_t^{\mathsf{SISR}}(z_{1:t}) = \hat{\pi}_{t-1}(z_{1:t-1})\, q_t(z_t \mid z_{1:t-1}),$$

where $\hat{\pi}_{t-1}(z_{1:t-1}) = \sum_i W_{t-1}^{(i)} \delta(z_{1:t-1} - z_{1:t-1}^{(i)})$.

**How resampling works (selection step):**

- Resample $N_s$ samples from $z_{1:t-1}^{(a_i)} \sim \hat{\pi}_{t-1}(z_{1_t-1})$.
- Reset weights: $\tilde{w}_{t-1}^{(i)} = 1$.

**Propagation:**

$$z_t^{(i)} \sim q_t(z_t \mid z_{1:t-1}^{(a_i)}), \qquad z_{1:t}^{(i)} = (z_{1:t-1}^{(a_i)}, z_t^{(i)}).$$

# Sequential Importance Sampling with Resampling (SISR)

**Weight update:**

$$\tilde{w}_t^{(i)} = \alpha_t(z_{1:t}^{(i)}) = \frac{\tilde{\gamma}_t(z_{1:t}^{(i)})}{\tilde{\gamma}_{t-1}(z_{1:t-1}^{(a_i)})\, q_t(z_t^{(i)} \mid z_{1:t-1}^{(a_i)})}.$$

(Same incremental weights as SIS.)

**Resampling methods:** Inverse cdf, multinomial resampling, stratified resampling, systematic resampling.

# Bootstrap filter

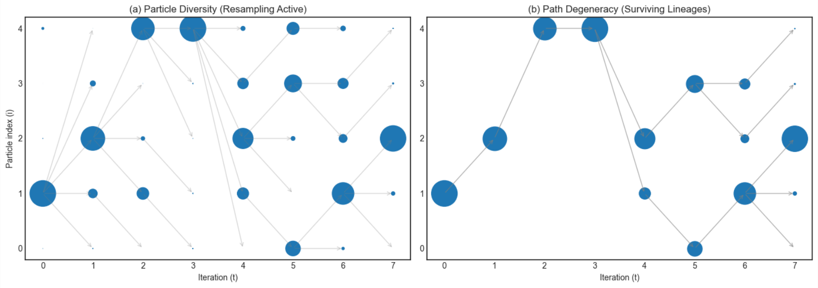
**Bootstrap filter:** Special case where the model is an SSM and the proposal equal to the dynamical prior:

$$q_t(z_t|z_{1:t-1}) = p(z_t|z_{1:t-1})$$

Hence,

$$\alpha_t(z_{1:t}) = \frac{p(y_t \mid z_{1:t})p(z_t \mid z_{1:t-1})}{q_t(z_t \mid z_{1:t-1})} = p(y_t|z_{1:t})$$

**Limitation**: Path degeneracy



# Adaptive Resampling

**Key trade-off:**

- **No resampling** $\Rightarrow$ SIS:
  - severe *weight degeneracy*
  - few particles effectively contribute
- **Resample at every step**:
  - avoids weight collapse
  - increases *path degeneracy* (loss of diversity in ancestry)

# Adaptive Resampling

**Key trade-off:**

- **No resampling** $\Rightarrow$ SIS:
  - severe *weight degeneracy*
  - few particles effectively contribute
- **Resample at every step**:
  - avoids weight collapse
  - increases *path degeneracy* (loss of diversity in ancestry)

**Adaptive resampling:** resample only when particle diversity becomes too low.

**Effective Sample Size (ESS):**

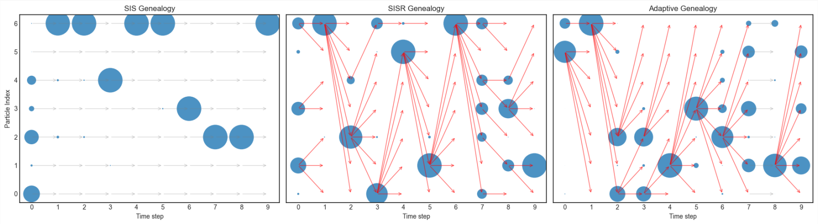
$$\mathrm{ESS}(W_{1:N}) = \frac{1}{\sum_{n=1}^{N} W_n^2}, \qquad \mathrm{ESS}(\tilde{w}_{1:N}) = \frac{\left(\sum_{n=1}^{N} \tilde{w}_n\right)^2}{\sum_{n=1}^{N} \tilde{w}_n^2}.$$

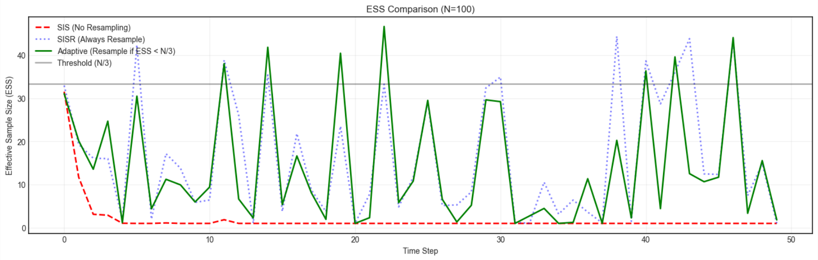
# Algorithm: SISR with Adaptive Resampling

**Algorithm 1** SISR with adaptive resampling (generic SMC)

**Initialization:** $\tilde{w}_0^{1:N_s} = 1$,
**for** $t = 1 : T$ **do**
  **for** $i = 1 : N_s$ **do**
    Sample particle: $z_t^{(i)} \sim q_t(z_t \mid z_{1:t-1}^{(i)})$
    Compute incremental weight: $\alpha_t^{(i)} = \frac{\tilde{\gamma}_t(z_{1:t}^{(i)})}{\tilde{\gamma}_{t-1}(z_{1:t-1}^{(i)})\, q_t(z_t^{(i)} | z_{1:t-1}^{(i)})}$
    Compute unnormalized weight: $\tilde{w}_t^{(i)} = \tilde{w}_{t-1}^{(i)} \alpha_t^{(i)}$
  **end for**
  **if** $\mathrm{ESS}(\tilde{w}_{t-1}^{1:N_s}) < \mathrm{ESS}_{\min}$ **then**
    Compute ancestors: $a_{1:N_s} = \text{resample}(\tilde{w}_{t-1}^{1:N_s})$
    Select particles: $z_{1:t}^{(i)} \leftarrow z_{1:t}^{(a_i)}$
    Reset weights: $\tilde{w}_t^{(i)} = 1/N_s$
  **end if**
  Normalize weights: $W_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}}$
  Posterior approximation: $\hat{\pi}_t(z_{1:t}) = \sum_{i=1}^{N_s} W_t^{(i)} \delta(z_{1:t} - z_{1:t}^{(i)})$
**end for**

# Comparing SIS and SISR



## Proposal distributions

The locally optimal proposal distribution $q^*(z_t|z_{1:t-1})$ is the one that minimizes:

$$D_{KL}(\pi_{t-1}(z_{1:t-1})q_t(z_t|z_{1:t-1})\|\pi_t(z_{1:t})),$$

which is:

$$q^*(z_t|z_{1:t-1}) = \pi_t(z_t|z_{1:t-1}) = \frac{\tilde{\gamma}(z_{1:t})}{\tilde{\gamma}(z_{1:t-1})}.$$

Usually intractable. Some approximations are:

- Proposals based on extended and unscented Kalman filter.
- Laplace approximations.
- Nested SMC.

# Contents

## SMC Samplers

**Goal:** sample from a *generic* target distribution

$$\pi(z) = \frac{\tilde{\gamma}(z)}{Z},$$

**SMC samplers** provide an alternative to MCMC, with:

- estimation of the normalizing constant $Z$,
- natural adaptivity,
- easy parallelization.

**Key idea:** define a sequence of intermediate distributions

$$\pi_t(z_t) = \frac{\tilde{\gamma}_t(z_t)}{Z_t}, \qquad t = 0, \ldots, T,$$

interpolating between an easy distribution $\pi_0$ and the target $\pi_T = \pi$.

**Path construction via forward and backward kernels:**

- **Forward kernel** $M_t(z_t \mid z_{t-1})$: a Markov kernel that leaves $\pi_t$ invariant. Used to propagate particles forward.
- **Backward kernel** $L_{t-1}(z_{t-1} \mid z_t)$: defines a joint path distribution

$$\pi_t(z_{1:t}) = \pi_t(z_t) \prod_{s=1}^{t-1} L_s(z_s \mid z_{s+1}),$$

satisfying $\sum_{z_{1:t-1}} \pi_t(z_{1:t}) = \pi_t(z_t)$.

## SMC samplers

**Incremental importance weight:**

$$\alpha_t = \frac{\pi_t(z_{1:t})}{\pi_{t-1}(z_{1:t-1})\, M_t(z_t \mid z_{t-1})} \;\propto\; \frac{\tilde{\gamma}_t(z_t)}{\tilde{\gamma}_{t-1}(z_{t-1})} \frac{L_{t-1}(z_{t-1} \mid z_t)}{M_t(z_t \mid z_{t-1})}.$$

**Key condition (time-reversal):** choose $L_{t-1}$ such that

$$\pi_t(z_t)\, L_{t-1}(z_{t-1} \mid z_t) = \pi_t(z_{t-1})\, M_t(z_t \mid z_{t-1}),$$

which simplifies the weight update to

$$\alpha_t \;\propto\; \frac{\tilde{\gamma}_t(z_t)}{\tilde{\gamma}_{t-1}(z_{t-1})}$$

# Likelihood Tempering (Geometric Path)

### Geometric path of targets

$$\tilde{\gamma}_t(z) = \tilde{\gamma}_0(z)^{1-\lambda_t}\, \tilde{\gamma}(z)^{\lambda_t}, \qquad 0 = \lambda_0 < \cdots < \lambda_T = 1.$$

### Bayesian parameter inference

$$\tilde{\gamma}_0(\theta) \propto \pi_0(\theta), \qquad \tilde{\gamma}(\theta) = \pi_0(\theta)\, p(D \mid \theta),$$

$$\tilde{\gamma}_t(\theta) = \pi_0(\theta)\, p(D \mid \theta)^{\lambda_t} = \pi_0(\theta)\, \exp[-\lambda_t E(\theta)], \qquad E(\theta) = -\log p(D, \theta)$$

### Incremental importance weights

Let $\lambda_t = \lambda_{t-1} + \delta_t$. Then

$$\alpha_t(\theta) = \frac{\tilde{\gamma}_t(\theta)}{\tilde{\gamma}_{t-1}(\theta)} = \exp[-\delta_t E(\theta)].$$

# Likelihood Tempering

**Adaptive choice of $\delta_t$**

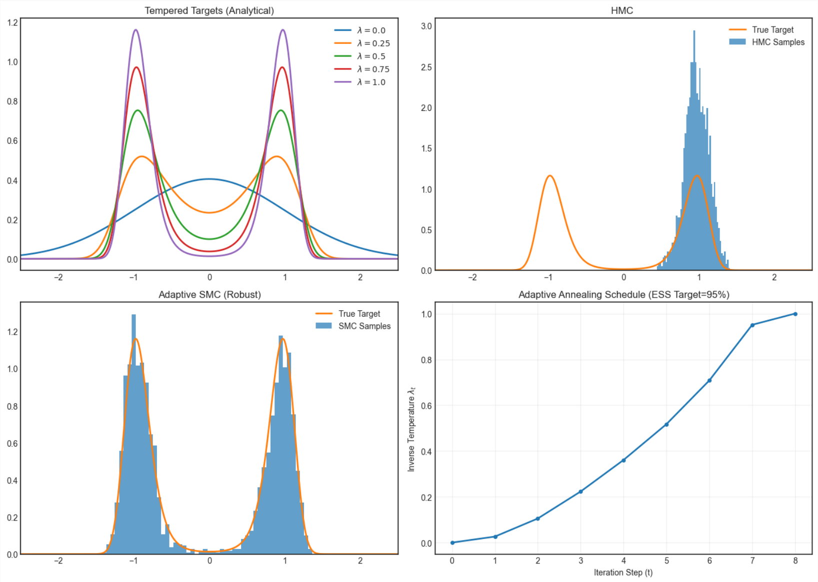
Choose $\delta_t$ such that

$$\delta_t = \operatorname{argmin}_{\delta \in [0, 1-\lambda_{t-1}]} \left(\mathrm{ESSLW}\left(\{-\delta\, E(\theta_t^n)\}\right) - \mathrm{ESS}_{\min}\right)$$

typically $\mathrm{ESS}_{\min} \approx 0.5\, N$.

- Ensures successive targets are approximately equidistant
- Prevents weight degeneracy
- If there is no such $\delta_t$, set $\delta_t = 1 - \lambda_{t-1}$

# Example



Tempered Targets (Analytical)
$\lambda = 0.0$
$\lambda = 0.25$
$\lambda = 0.5$
$\lambda = 0.75$
$\lambda = 1.0$
HMC
True Target
HMC Samples
Adaptive SMC (Robust)
True Target
SMC Samples
Adaptive Annealing Schedule (ESS Target=95%)
Inverse Temperature $\lambda_t$
Iteration Step (t)

# Data Tempering

**Intermediate target distributions**

Given iid observations $y_{1:T}$, define

$$\tilde{\gamma}_t(\theta) = p(\theta)\, p(y_{1:t} \mid \theta), \qquad t = 1, \ldots, T.$$

- Prior: $\tilde{\gamma}_0(\theta) = p(\theta)$
- Final target: $\tilde{\gamma}_T(\theta) = p(\theta \mid y_{1:T})$

**Incremental importance weights** Using the SMC sampler weight update,

$$\alpha_t(\theta) = \frac{\tilde{\gamma}_t(\theta)}{\tilde{\gamma}_{t-1}(\theta)} = \frac{p(\theta)p(y_{1:t} \mid \theta)}{p(\theta)p(y_{1:t-1} \mid \theta)} = p(y_t \mid y_{1:t-1}, \theta).$$

# Data tempering

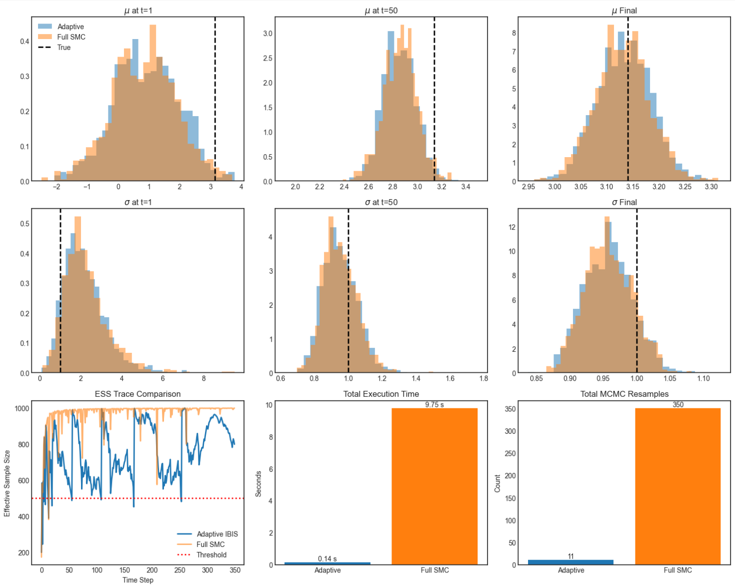
**Computational considerations**

- Evaluating MCMC at every step is expensive.

**IBIS (Iterated Batch Importance Sampling)**

- Only apply MCMC step when ESS drops below a threshold
- Otherwise, propagate particles deterministically ($\theta^{(i)} = \theta^{(i-1)}$)

# Example



$\mu$ at t=1
Adaptive
Full SMC
True
$\mu$ at t=50
$\mu$ Final
$\sigma$ at t=1
$\sigma$ at t=50
$\sigma$ Final
ESS Trace Comparison
Effective Sample Size
Time Step
Adaptive IBIS
Full SMC
Threshold
Total Execution Time
Seconds
0.14 s
9.75 s
Adaptive
Full SMC
Total MCMC Resamples
Count
11
350
Adaptive
Full SMC

**Sampling Rare Events:**

- Goal: sample from $\pi_0(\theta)$ conditioned on a rare event $S(\theta) > \lambda^*$, where $S(\theta)$ is a score or fitness function.
- Approach: SMC with gradually increasing thresholds:

$$\pi_t(\theta) = \frac{1}{Z_t}\,\mathbb{I}(S(\theta) \geq \lambda_t)\,\pi_0(\theta), \quad \lambda_0 < \cdots < \lambda_T = \lambda^*.$$

- We may use likelihood tempering with the "likelihood" at each step being:

$$G_t(\theta) = \mathbb{I}(S(\theta) \geq \lambda_t)$$

# Likelihood-Free inference via SMC

**Likelihood-Free Inference (SMC-ABC):**

- Consider models where the likelihood $p(y|\theta)$ is **intractable** but we can **simulate data** $y \sim p(\cdot|\theta)$.
- Approximate Bayesian Computation (ABC) samples $(\theta, y)$ such that simulated data is close to observed data:

$$d(y, y^*) < \epsilon$$

or for summary statistics $s(y)$: $d(s(y), s(y^*)) < \epsilon$.

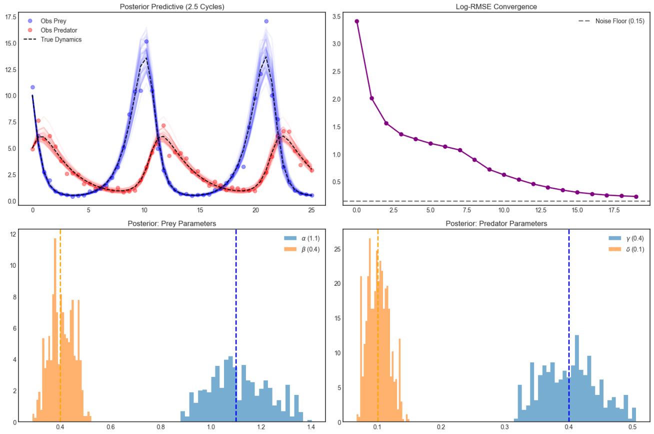
# Likelihood-Free inference via SMC

- SMC-ABC gradually decreases $\epsilon$ over iterations:

$$\pi_t(\theta, y) = \frac{1}{Z_t}\, \pi_0(\theta)\, p(y|\theta)\, \mathbb{I}(d(y, y^*) < \epsilon_t), \quad \epsilon_0 > \cdots > \epsilon_T$$

- This is analogous to rare-event SMC, but the "likelihood" is evaluated in **data space**.

# Example



Posterior Predictive (2.5 Cycles)
Obs Prey
Obs Predator
True Dynamics
Log-RMSE Convergence
Noise Floor (0.15)
Posterior: Prey Parameters
$\alpha$ (1.1)
$\beta$ (0.4)
Posterior: Predator Parameters
$\gamma$ (0.4)
$\delta$ (0.1)

# Contents

# Conclusions

- We motivated **Sequential Monte Carlo (SMC)** as a general framework for approximating sequences of complex distributions. Specifically, we
- Studied SMC for **State Space Models**, including:
  - Sequential Importance Sampling (SIS).
  - Sequential Importance Sampling with Resampling (SISR).
  - Adaptive resampling.
- Presented **SMC samplers** as a flexible alternative to MCMC:
  - likelihood tempering.
  - Data tempering.
  - likelihood free inference.

# References i


[DGA00] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. **"On sequential Monte Carlo sampling methods for Bayesian filtering".** In: *Statistics and Computing* (2000).

[DJ09] Arnaud Doucet and Adam Johansen. **"A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later".** In: *Handbook of Nonlinear Filtering* 12 (Jan. 2009).

[DMDJ12] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. **"An Adaptive Sequential Monte Carlo Method for Approximate Bayesian Computation".** In: *Statistics and Computing* (2012).

# Questions?

# Upcoming sessions

- Next session:
  - **Special session: VI + MCMC**
  - January 14th, 2026
  - *Max Hird (PostDoc @ University of Waterloo)*
- Following section:
  - **Bayesian Neural Networks (Ch. 17)**
  - January 21st, 2026
  - *Daniel Corrales (PhD @ Institute of Mathematical Sciences)*
- After that:
  - Every other week (so 4/2, 18/2, 4/3, ...)