Particle-MALA and Particle-mGRAD

Gradient-based MCMC methods for high-dimensional state-space models[1]

Adrien Corenflos* Axel Finke†

*The University of Warwick, UK

†Loughborough University, UK

14th June 2024

[1] https://arxiv.org/pdf/2401.14868

Talk outline

1. State-space models/Feynman–Kac representation
2. Existing methods
3. Particle extensions of MALA and aMALA
4. Particle extensions of mGRAD and aGRAD
5. Numerical illustration
6. Summary

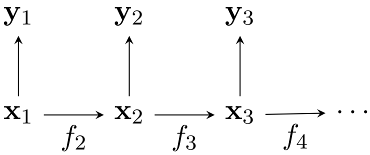
Motivation: State-space model

$$\mathbf{x}_1 \longrightarrow \mathbf{x}_2 \longrightarrow \mathbf{x}_3 \longrightarrow \cdots$$

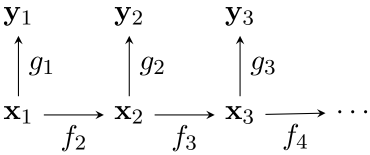
Motivation: State-space model

$$\mathbf{x}_1 \xrightarrow[f_2]{} \mathbf{x}_2 \xrightarrow[f_3]{} \mathbf{x}_3 \xrightarrow[f_4]{} \cdots$$

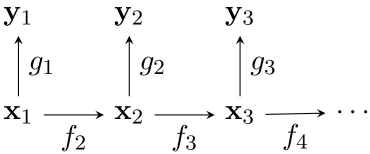
Motivation: State-space model



Motivation: State-space model

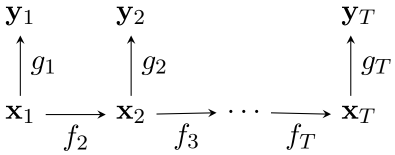


Motivation: State-space model

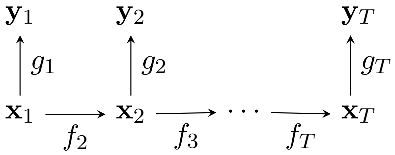


- Examples:
  - econometrics/finance,
  - ecology,
  - engineering,
  - epidemiology,
  - weather forcasting,
  - ...

Motivation: State-space model, continued

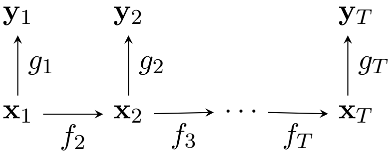


Motivation: State-space model, continued



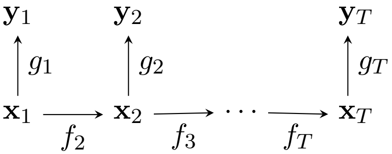
- T observations: $\mathbf{y}_1, \ldots, \mathbf{y}_T$.

Motivation: State-space model, continued



- T observations: $\mathbf{y}_1, \ldots, \mathbf{y}_T$.
- D-dimensional latent states: $\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,D} \end{bmatrix} \in \mathcal{X} := \mathbb{R}^D$,

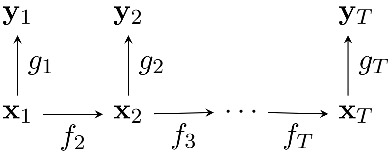
Motivation: State-space model, continued



- T observations: $\mathbf{y}_1, \ldots, \mathbf{y}_T$.
- D-dimensional latent states: $\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,D} \end{bmatrix} \in \mathcal{X} := \mathbb{R}^D$,
- **Joint smoothing distribution:**

$$\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \propto \prod_{t=1}^{T} f_t(\mathbf{x}_t|\mathbf{x}_{t-1}) g_t(\mathbf{y}_t|\mathbf{x}_t).$$

Motivation: State-space model, continued

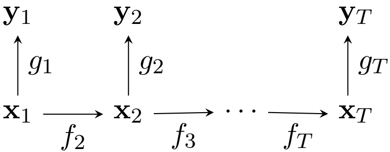


- T observations: $\mathbf{y}_1, \ldots, \mathbf{y}_T$.
- D-dimensional latent states: $\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,D} \end{bmatrix} \in \mathcal{X} := \mathbb{R}^D$,
- **Joint smoothing distribution:**

$$\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \propto \prod_{t=1}^{T} f_t(\mathbf{x}_t|\mathbf{x}_{t-1}) g_t(\mathbf{y}_t|\mathbf{x}_t).$$

- **Assumption:** densities f_t and g_t are differentiable (in the states); densities/gradients can be evaluated pointwise.

Motivation: State-space model, continued

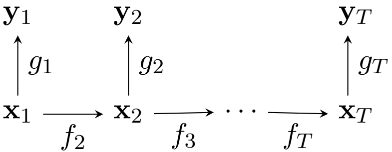


- T observations: $\mathbf{y}_1, \ldots, \mathbf{y}_T$.
- D-dimensional latent states: $\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,D} \end{bmatrix} \in \mathcal{X} := \mathbb{R}^D$,
- **Joint smoothing distribution:**

$$\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \propto \prod_{t=1}^{T} f_t(\mathbf{x}_t|\mathbf{x}_{t-1}) g_t(\mathbf{y}_t|\mathbf{x}_t).$$

- **Assumption:** densities f_t and g_t are differentiable (in the states); densities/gradients can be evaluated pointwise.
- **Goal:** find efficient MCMC algorithms targetting $\pi_T(\mathbf{x}_{1:T})$.

Motivation: State-space model, continued



- T observations: $\mathbf{y}_1, \ldots, \mathbf{y}_T$.
- D-dimensional latent states: $\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,D} \end{bmatrix} \in \mathcal{X} := \mathbb{R}^D$,
- **Joint smoothing distribution:**

$$\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \propto \prod_{t=1}^{T} f_t(\mathbf{x}_t|\mathbf{x}_{t-1}) g_t(\mathbf{y}_t|\mathbf{x}_t).$$

- **Assumption:** densities f_t and g_t are differentiable (in the states); densities/gradients can be evaluated pointwise.
- **Goal:** find efficient MCMC algorithms targetting $\pi_T(\mathbf{x}_{1:T})$.
- **Problem:** $\pi_T(\mathbf{x}_{1:T})$ may be high dimensional (T or D large).

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution
$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$
on $\mathcal{X}^T$ (with $\mathcal{X} \coloneqq \mathbb{R}^D$), where

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution
$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1})G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$
on $\mathcal{X}^T$ (with $\mathcal{X} := \mathbb{R}^D$), where
  - $M_t(\,\cdot\,|\mathbf{x}_{t-1})$ is a density of a *mutation kernel*;

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution
$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1})G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$
on $\mathcal{X}^T$ (with $\mathcal{X} := \mathbb{R}^D$), where
  - $M_t(\,\cdot\,|\mathbf{x}_{t-1})$ is a density of a *mutation kernel*;
  - $G_t(\mathbf{x}_{t-1:t}) > 0$ is called *potential function*.

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution
$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$
on $\mathcal{X}^T$ (with $\mathcal{X} := \mathbb{R}^D$), where
  - $M_t(\,\cdot\,|\mathbf{x}_{t-1})$ is a density of a *mutation kernel*;
  - $G_t(\mathbf{x}_{t-1:t}) > 0$ is called *potential function*.
- **Assumption:** M_t and G_t are differentiable; both functions and their gradients can be evaluated point-wise.

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution
$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$
on $\mathcal{X}^T$ (with $\mathcal{X} \coloneqq \mathbb{R}^D$), where
  - $M_t(\,\cdot\,|\mathbf{x}_{t-1})$ is a density of a *mutation kernel*;
  - $G_t(\mathbf{x}_{t-1:t}) > 0$ is called *potential function*.
- **Assumption:** M_t and G_t are differentiable; both functions and their gradients can be evaluated point-wise.
- For $t \leq T$, define the *filters*: $\pi_t(\mathbf{x}_{1:t}) \propto \prod_{s=1}^{t} Q_s(\mathbf{x}_{s-1:s})$.

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution
$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$
on $\mathcal{X}^T$ (with $\mathcal{X} \coloneqq \mathbb{R}^D$), where
  - $M_t(\,\cdot\,|\mathbf{x}_{t-1})$ is a density of a *mutation kernel*;
  - $G_t(\mathbf{x}_{t-1:t}) > 0$ is called *potential function*.
- **Assumption:** M_t and G_t are differentiable; both functions and their gradients can be evaluated point-wise.
- For $t \leq T$, define the *filters*: $\pi_t(\mathbf{x}_{1:t}) \propto \prod_{s=1}^{t} Q_s(\mathbf{x}_{s-1:s})$.
- **Example:** For state-space models, *one* possible Feynman–Kac representation of $\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ is

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution
$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$
on $\mathcal{X}^T$ (with $\mathcal{X} \coloneqq \mathbb{R}^D$), where
  - $M_t(\,\cdot\,|\mathbf{x}_{t-1})$ is a density of a *mutation kernel*;
  - $G_t(\mathbf{x}_{t-1:t}) > 0$ is called *potential function*.
- **Assumption:** M_t and G_t are differentiable; both functions and their gradients can be evaluated point-wise.
- For $t \leq T$, define the *filters*: $\pi_t(\mathbf{x}_{1:t}) \propto \prod_{s=1}^{t} Q_s(\mathbf{x}_{s-1:s})$.
- **Example:** For state-space models, *one* possible Feynman–Kac representation of $\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ is
  - $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = f_t(\mathbf{x}_t|\mathbf{x}_{t-1})$; and

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution

$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1})G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$

  on $\mathcal{X}^T$ (with $\mathcal{X} \coloneqq \mathbb{R}^D$), where
  - $M_t(\,\cdot\,|\mathbf{x}_{t-1})$ is a density of a *mutation kernel*;
  - $G_t(\mathbf{x}_{t-1:t}) > 0$ is called *potential function*.
- **Assumption:** M_t and G_t are differentiable; both functions and their gradients can be evaluated point-wise.
- For $t \leq T$, define the *filters*: $\pi_t(\mathbf{x}_{1:t}) \propto \prod_{s=1}^{t} Q_s(\mathbf{x}_{s-1:s})$.
- **Example:** For state-space models, *one* possible Feynman–Kac representation of $\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ is
  - $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = f_t(\mathbf{x}_t|\mathbf{x}_{t-1})$; and
  - $G_t(\mathbf{x}_{t-1:t}) = g_t(\mathbf{y}_t|\mathbf{x}_t)$

Generic Feynman–Kac representation

- **More generally:** we are interested in a distribution

$$\pi_T(\mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t}) = \prod_{t=1}^{T} Q_t(\mathbf{x}_{t-1:t}),$$

  on $\mathcal{X}^T$ (with $\mathcal{X} \coloneqq \mathbb{R}^D$), where
  - $M_t(\,\cdot\,|\mathbf{x}_{t-1})$ is a density of a *mutation kernel*;
  - $G_t(\mathbf{x}_{t-1:t}) > 0$ is called *potential function*.
- **Assumption:** M_t and G_t are differentiable; both functions and their gradients can be evaluated point-wise.
- For $t \leq T$, define the *filters*: $\pi_t(\mathbf{x}_{1:t}) \propto \prod_{s=1}^{t} Q_s(\mathbf{x}_{s-1:s})$.
- **Example:** For state-space models, *one* possible Feynman–Kac representation of $\pi_T(\mathbf{x}_{1:T}) = p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ is
  - $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = f_t(\mathbf{x}_t|\mathbf{x}_{t-1})$; and
  - $G_t(\mathbf{x}_{t-1:t}) = g_t(\mathbf{y}_t|\mathbf{x}_t)$

  Then, $Q_t(\mathbf{x}_{t-1:t}) = p(\mathbf{x}_t, \mathbf{y}_t|\mathbf{x}_{t-1})$ and $\pi_t(\mathbf{x}_{1:t}) = p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$.

Talk outline

1. State-space models/Feynman–Kac representation
2. Existing methods
3. Particle extensions of MALA and aMALA
4. Particle extensions of mGRAD and aGRAD
5. Numerical illustration
6. Summary

Talk outline

MCMC methods

- 'Classical' MCMC methods are agnostic to the state-space model structure.

MCMC methods

- 'Classical' MCMC methods are agnostic to the state-space model structure.
- For the moment, write $\mathbf{x} \coloneqq \mathbf{x}_{1:T}$, so that

$$\pi(\mathbf{x}) \coloneqq \pi_T(\mathbf{x}) \propto M(\mathbf{x})G(\mathbf{x}),$$

where

MCMC methods

- 'Classical' MCMC methods are agnostic to the state-space model structure.
- For the moment, write $\mathbf{x} \coloneqq \mathbf{x}_{1:T}$, so that

$$\pi(\mathbf{x}) \coloneqq \pi_T(\mathbf{x}) \propto M(\mathbf{x})G(\mathbf{x}),$$

  where
  - $M(\mathbf{x}) \coloneqq \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1})$ ('prior');

MCMC methods

- 'Classical' MCMC methods are agnostic to the state-space model structure.
- For the moment, write $\mathbf{x} := \mathbf{x}_{1:T}$, so that

$$\pi(\mathbf{x}) := \pi_T(\mathbf{x}) \propto M(\mathbf{x})G(\mathbf{x}),$$

  where
  - $M(\mathbf{x}) := \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1})$ ('prior');
  - $G(\mathbf{x}) := \prod_{t=1}^{T} G_t(\mathbf{x}_{t-1:t})$ ('likelihood').

MCMC methods

- 'Classical' MCMC methods are agnostic to the state-space model structure.
- For the moment, write $\mathbf{x} \coloneqq \mathbf{x}_{1:T}$, so that

$$\pi(\mathbf{x}) \coloneqq \pi_T(\mathbf{x}) \propto M(\mathbf{x})G(\mathbf{x}),$$

  where
  - $M(\mathbf{x}) \coloneqq \prod_{t=1}^{T} M_t(\mathbf{x}_t|\mathbf{x}_{t-1})$ ('prior');
  - $G(\mathbf{x}) \coloneqq \prod_{t=1}^{T} G_t(\mathbf{x}_{t-1:t})$ ('likelihood').
- **Note:** $\mathbf{x}$ is thus (TD)-dimensional.

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability $\alpha(\mathbf{x}, \tilde{\mathbf{x}}) := 1 \wedge \dfrac{\pi(\tilde{\mathbf{x}})q(\mathbf{x}|\tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}$.

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability $\alpha(\mathbf{x}, \tilde{\mathbf{x}}) := 1 \wedge \dfrac{\pi(\tilde{\mathbf{x}})q(\mathbf{x}|\tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}$.
- Assume $q(\tilde{\mathbf{x}}|\mathbf{x}) = \int q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})q(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}$.

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability $\alpha(\mathbf{x}, \tilde{\mathbf{x}}) := 1 \wedge \frac{\pi(\tilde{\mathbf{x}})q(\mathbf{x}|\tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}$.
- Assume $q(\tilde{\mathbf{x}}|\mathbf{x}) = \int q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})q(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}$.
- **[Auxiliary sampler]**

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability $\alpha(\mathbf{x}, \tilde{\mathbf{x}}) := 1 \wedge \dfrac{\pi(\tilde{\mathbf{x}})q(\mathbf{x}|\tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}$.
- Assume $q(\tilde{\mathbf{x}}|\mathbf{x}) = \int q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})q(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}$.
- **[Auxiliary sampler]**
  1. propose $\mathbf{u} \sim q(\mathbf{u}|\mathbf{x})$ and $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})$;

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability $\alpha(\mathbf{x}, \tilde{\mathbf{x}}) := 1 \wedge \frac{\pi(\tilde{\mathbf{x}})q(\mathbf{x}|\tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}$.
- Assume $q(\tilde{\mathbf{x}}|\mathbf{x}) = \int q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})q(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}$.
- **[Auxiliary sampler]**
  1. propose $\mathbf{u} \sim q(\mathbf{u}|\mathbf{x})$ and $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability

$$1 \wedge \frac{\pi(\tilde{\mathbf{x}})q(\mathbf{u}|\tilde{\mathbf{x}})q(\mathbf{x}|\mathbf{u}, \tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\mathbf{u}|\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})} = \alpha(\mathbf{x}, \tilde{\mathbf{x}}) \overbrace{\frac{q(\mathbf{u}|\mathbf{x}, \tilde{\mathbf{x}})}{q(\mathbf{u}|\tilde{\mathbf{x}}, \mathbf{x})}}^{=:h(\mathbf{u})}.$$

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability $\alpha(\mathbf{x}, \tilde{\mathbf{x}}) := 1 \wedge \dfrac{\pi(\tilde{\mathbf{x}})q(\mathbf{x}|\tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}$.
- Assume $q(\tilde{\mathbf{x}}|\mathbf{x}) = \int q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})q(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}$.
- **[Auxiliary sampler]**
  1. propose $\mathbf{u} \sim q(\mathbf{u}|\mathbf{x})$ and $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability

$$1 \wedge \frac{\pi(\tilde{\mathbf{x}})q(\mathbf{u}|\tilde{\mathbf{x}})q(\mathbf{x}|\mathbf{u}, \tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\mathbf{u}|\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})} = \alpha(\mathbf{x}, \tilde{\mathbf{x}}) \overbrace{\frac{q(\mathbf{u}|\mathbf{x}, \tilde{\mathbf{x}})}{q(\mathbf{u}|\tilde{\mathbf{x}}, \mathbf{x})}}^{=: h(\mathbf{u})}.$$

- Two interpretations of the auxiliary sampler:

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability $\alpha(\mathbf{x}, \tilde{\mathbf{x}}) := 1 \wedge \dfrac{\pi(\tilde{\mathbf{x}})q(\mathbf{x}|\tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}$.
- Assume $q(\tilde{\mathbf{x}}|\mathbf{x}) = \int q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})q(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}$.
- **[Auxiliary sampler]**
  1. propose $\mathbf{u} \sim q(\mathbf{u}|\mathbf{x})$ and $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability

$$1 \wedge \frac{\pi(\tilde{\mathbf{x}})q(\mathbf{u}|\tilde{\mathbf{x}})q(\mathbf{x}|\mathbf{u}, \tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\mathbf{u}|\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})} = \alpha(\mathbf{x}, \tilde{\mathbf{x}}) \overbrace{\frac{q(\mathbf{u}|\mathbf{x}, \tilde{\mathbf{x}})}{q(\mathbf{u}|\tilde{\mathbf{x}}, \mathbf{x})}}^{=:h(\mathbf{u})}.$$

- Two interpretations of the auxiliary sampler:
  1. Standard MH conditional on $\mathbf{u}$, i.e. targetting $\pi(\mathbf{x}; \mathbf{u}) = \pi(\mathbf{x})q(\mathbf{u}|\mathbf{x})$.
  2. MH with randomised acceptance ratio[3] (since $\mathbb{E}[h(\mathbf{u})|\mathbf{x}, \tilde{\mathbf{x}}] = 1$).

---

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

MCMC methods

- **[Marginal sampler]** Metropolis–Hastings (MH)[2] algorithm:
  1. propose $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability $\alpha(\mathbf{x}, \tilde{\mathbf{x}}) := 1 \wedge \dfrac{\pi(\tilde{\mathbf{x}})q(\mathbf{x}|\tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}$.
- Assume $q(\tilde{\mathbf{x}}|\mathbf{x}) = \int q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x}) q(\mathbf{u}|\mathbf{x}) \, \mathrm{d}\mathbf{u}$.
- **[Auxiliary sampler]**
  1. propose $\mathbf{u} \sim q(\mathbf{u}|\mathbf{x})$ and $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})$;
  2. accept $\tilde{\mathbf{x}}$ with probability

$$1 \wedge \frac{\pi(\tilde{\mathbf{x}})q(\mathbf{u}|\tilde{\mathbf{x}})q(\mathbf{x}|\mathbf{u}, \tilde{\mathbf{x}})}{\pi(\mathbf{x})q(\mathbf{u}|\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{u}, \mathbf{x})} = \alpha(\mathbf{x}, \tilde{\mathbf{x}}) \overbrace{\frac{q(\mathbf{u}|\mathbf{x}, \tilde{\mathbf{x}})}{q(\mathbf{u}|\tilde{\mathbf{x}}, \mathbf{x})}}^{=:h(\mathbf{u})}.$$

- Two interpretations of the auxiliary sampler:
  1. Standard MH conditional on $\mathbf{u}$, i.e. targetting $\pi(\mathbf{x}; \mathbf{u}) = \pi(\mathbf{x})q(\mathbf{u}|\mathbf{x})$.
  2. MH with randomised acceptance ratio[3] (since $\mathbb{E}[h(\mathbf{u})|\mathbf{x}, \tilde{\mathbf{x}}] = 1$).
- Efficiency of auxiliary sampler $\leq$ efficiency of marginal sampler.[4]

[2] Metropolis et al. (1953); Hastings (1970)
[3] Ceperley and Dewing (1999)
[4] Andrieu and Vihola (2016)

A simple MCMC algorithm

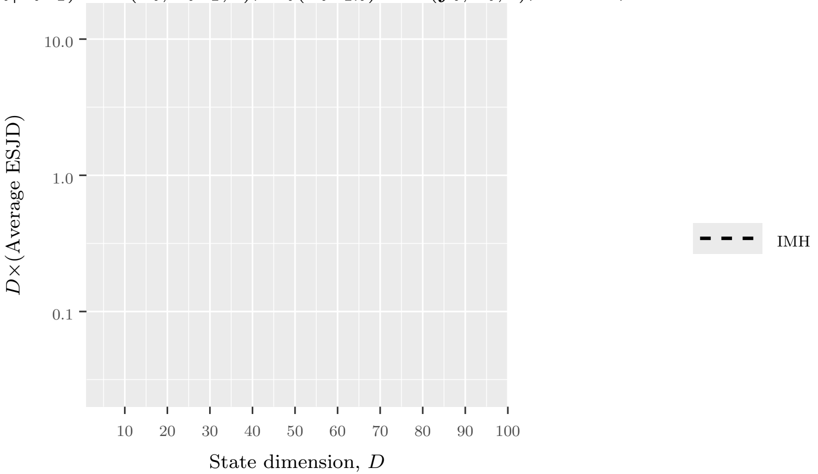
- Independent Metropolis–Hastings (IMH)[5]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = M(\tilde{\mathbf{x}}).$$

[5] Hastings (1970)

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \Longrightarrow$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Proposing local moves

- **[Marginal sampler]** Random-walk Metropolis (RWM)[6]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x}, \delta \mathbf{I}).$$

[6] Metropolis et al. (1953)

Proposing local moves

- **[Marginal sampler]** Random-walk Metropolis (RWM)[6]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x}, \delta\mathbf{I}).$$

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling

[6] Metropolis et al. (1953)

Proposing local moves

- **[Marginal sampler]** Random-walk Metropolis (RWM)[6]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x}, \delta\mathbf{I}).$$

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x}, \frac{\delta}{2}\mathbf{I})$;

[6] Metropolis et al. (1953)

Proposing local moves

- **[Marginal sampler]** Random-walk Metropolis (RWM)[6]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x}, \delta\mathbf{I}).$$

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x}, \frac{\delta}{2}\mathbf{I})$;
  2. $\tilde{\mathbf{x}} \sim \mathrm{N}(\mathbf{u}, \frac{\delta}{2}\mathbf{I})$.

[6] Metropolis et al. (1953)

Proposing local moves

- **[Marginal sampler]** Random-walk Metropolis (RWM)[6]:

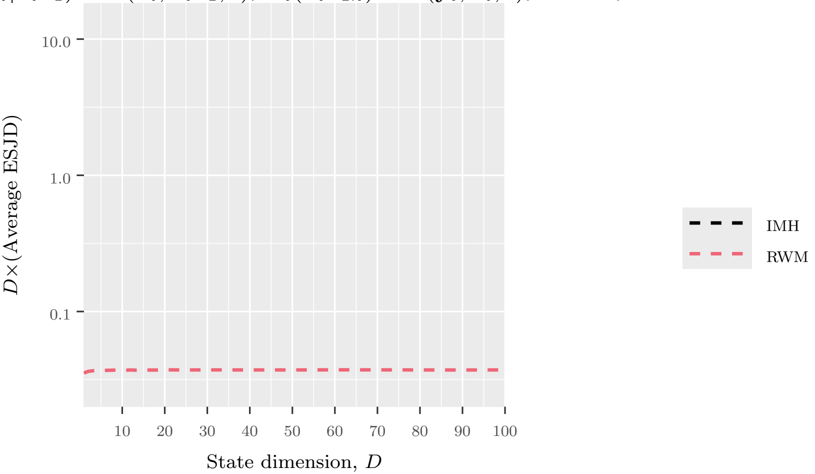
$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x}, \delta\mathbf{I}).$$

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x}, \frac{\delta}{2}\mathbf{I})$;
  2. $\tilde{\mathbf{x}} \sim \mathrm{N}(\mathbf{u}, \frac{\delta}{2}\mathbf{I})$.
- **[Auxiliary sampler]** Not integrating out $\mathbf{u}$ in the acceptance ratio is statistically equivalent to the marginal sampler.

[6] Metropolis et al. (1953)

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\mathrm{new}} - x_{t,d}^{\mathrm{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Exploiting gradient information

- **[Marginal sampler]** Metropolis-adjusted Langevin algorithm (MALA)[7]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x} + \tfrac{\delta}{2}\nabla \log \pi(\mathbf{x}), \delta \mathbf{I}).$$

[7] Besag (1994)
[8] Titsias and Papaspiliopoulos (2018)

Exploiting gradient information

- **[Marginal sampler]** Metropolis-adjusted Langevin algorithm (MALA)[7]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x} + \frac{\delta}{2}\nabla \log \pi(\mathbf{x}), \delta \mathbf{I}).$$

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling

[7] Besag (1994)

[8] Titsias and Papaspiliopoulos (2018)

Exploiting gradient information

- **[Marginal sampler]** Metropolis-adjusted Langevin algorithm (MALA)[7]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x} + \tfrac{\delta}{2}\nabla \log \pi(\mathbf{x}), \delta \mathbf{I}).$$

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \tfrac{\delta}{2}\nabla \log \pi(\mathbf{x}), \tfrac{\delta}{2}\mathbf{I})$;

[7] Besag (1994)
[8] Titsias and Papaspiliopoulos (2018)

Exploiting gradient information

- **[Marginal sampler]** Metropolis-adjusted Langevin algorithm (MALA)[7]:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x} + \tfrac{\delta}{2}\nabla \log \pi(\mathbf{x}), \delta \mathbf{I}).$$

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \tfrac{\delta}{2}\nabla \log \pi(\mathbf{x}), \tfrac{\delta}{2}\mathbf{I})$;
  2. $\tilde{\mathbf{x}} \sim \mathrm{N}(\mathbf{u}, \tfrac{\delta}{2}\mathbf{I})$.

[7] Besag (1994)
[8] Titsias and Papaspiliopoulos (2018)

Exploiting gradient information

- **[Marginal sampler]** Metropolis-adjusted Langevin algorithm (MALA)[7]:

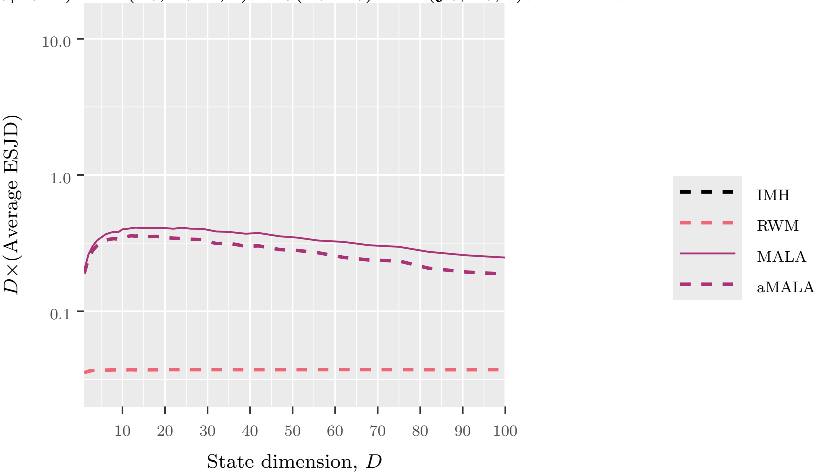
$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; \mathbf{x} + \tfrac{\delta}{2}\nabla \log \pi(\mathbf{x}), \delta \mathbf{I}).$$

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \frac{\delta}{2}\nabla \log \pi(\mathbf{x}), \frac{\delta}{2}\mathbf{I})$;
  2. $\tilde{\mathbf{x}} \sim \mathrm{N}(\mathbf{u}, \frac{\delta}{2}\mathbf{I})$.
- **[Auxiliary sampler]** Not integrating out $\mathbf{u}$ in the acceptance ratio gives the auxiliary MALA (aMALA)[8].

[7] Besag (1994)
[8] Titsias and Papaspiliopoulos (2018)

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$

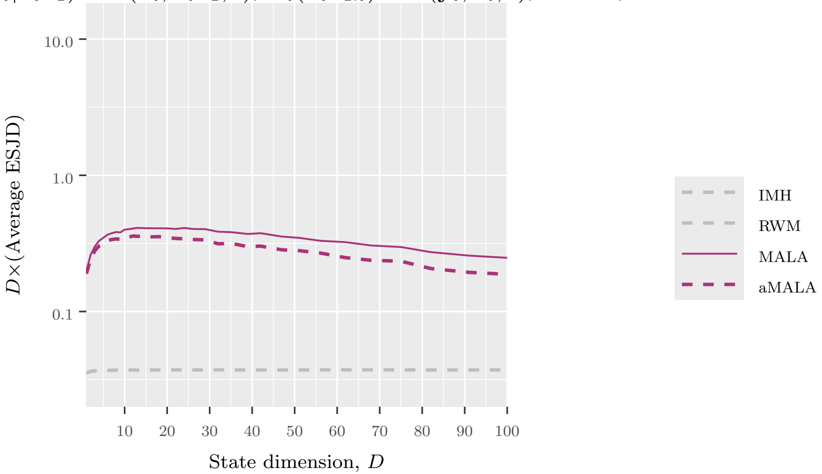


$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Exploiting Gaussian priors (and gradient info)

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Preconditioned Crank–Nicolson–Langevin (PCNL)[9] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (1-\beta)\mathbf{m} + \beta[\mathbf{x} + \tfrac{\delta}{2}\mathbf{C}\nabla \log G(\mathbf{x})], (1-\beta^2)\mathbf{C}),$$

where $\beta \coloneqq 2/(2+\delta)$.

[9] Cotter et al. (2013)

Exploiting Gaussian priors (and gradient info)

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Preconditioned Crank–Nicolson–Langevin (PCNL)[9] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (1-\beta)\mathbf{m} + \beta[\mathbf{x} + \tfrac{\delta}{2}\mathbf{C}\nabla \log G(\mathbf{x})], (1-\beta^2)\mathbf{C}),$$

  where $\beta := 2/(2+\delta)$.
- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling

[9] Cotter et al. (2013)

Exploiting Gaussian priors (and gradient info)

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Preconditioned Crank–Nicolson–Langevin (PCNL)[9] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (1-\beta)\mathbf{m} + \beta[\mathbf{x} + \tfrac{\delta}{2}\mathbf{C}\nabla \log G(\mathbf{x})], (1-\beta^2)\mathbf{C}),$$

  where $\beta := 2/(2+\delta)$.

- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \frac{\delta}{2}\mathbf{C}\nabla \log G(\mathbf{x}), \frac{\delta}{2}\mathbf{C})$;

[9] Cotter et al. (2013)

Exploiting Gaussian priors (and gradient info)

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Preconditioned Crank–Nicolson–Langevin (PCNL)[9] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (1-\beta)\mathbf{m} + \beta[\mathbf{x} + \tfrac{\delta}{2}\mathbf{C}\nabla \log G(\mathbf{x})], (1-\beta^2)\mathbf{C}),$$

  where $\beta \coloneqq 2/(2+\delta)$.
- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \frac{\delta}{2}\mathbf{C}\nabla \log G(\mathbf{x}), \frac{\delta}{2}\mathbf{C})$;
  2. $\tilde{\mathbf{x}} \sim \mathrm{N}((1-\beta)\mathbf{m} + \beta\mathbf{u}, (1-\beta)\mathbf{C})$.

[9] Cotter et al. (2013)

Exploiting Gaussian priors (and gradient info)

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Preconditioned Crank–Nicolson–Langevin (PCNL)[9] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (1-\beta)\mathbf{m} + \beta[\mathbf{x} + \tfrac{\delta}{2}\mathbf{C}\nabla \log G(\mathbf{x})], (1-\beta^2)\mathbf{C}),$$

  where $\beta := 2/(2+\delta)$.
- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \frac{\delta}{2}\mathbf{C}\nabla \log G(\mathbf{x}), \frac{\delta}{2}\mathbf{C})$;
  2. $\tilde{\mathbf{x}} \sim \mathrm{N}((1-\beta)\mathbf{m} + \beta\mathbf{u}, (1-\beta)\mathbf{C})$.
- **[Auxiliary sampler]** Not integrating out $\mathbf{u}$ in the acceptance ratio gives an auxiliary PCNL (aPCNL) algorithm.

[9] Cotter et al. (2013)

Exploiting Gaussian priors (and gradient info), continued

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Marginal gradient (mGRAD)[10] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (\mathbf{I} - \mathbf{A})\mathbf{m} + \mathbf{A}[\mathbf{x} + \tfrac{\delta}{2}\nabla \log G(\mathbf{x})], \mathbf{B}),$$

where $\mathbf{B} \coloneqq \frac{\delta}{2}\mathbf{A}^2 + \mathbf{A}$ and $\mathbf{A} = (\mathbf{C} + \frac{\delta}{2}\mathbf{I})^{-1}\mathbf{C}$.

[10] Titsias and Papaspiliopoulos (2018)

Exploiting Gaussian priors (and gradient info), continued

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Marginal gradient (mGRAD)[10] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (\mathbf{I} - \mathbf{A})\mathbf{m} + \mathbf{A}[\mathbf{x} + \tfrac{\delta}{2}\nabla \log G(\mathbf{x})], \mathbf{B}),$$

  where $\mathbf{B} := \frac{\delta}{2}\mathbf{A}^2 + \mathbf{A}$ and $\mathbf{A} = (\mathbf{C} + \frac{\delta}{2}\mathbf{I})^{-1}\mathbf{C}$.
- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling

[10] Titsias and Papaspiliopoulos (2018)

Exploiting Gaussian priors (and gradient info), continued

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Marginal gradient (mGRAD)[10] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (\mathbf{I} - \mathbf{A})\mathbf{m} + \mathbf{A}[\mathbf{x} + \tfrac{\delta}{2}\nabla \log G(\mathbf{x})], \mathbf{B}),$$

  where $\mathbf{B} := \frac{\delta}{2}\mathbf{A}^2 + \mathbf{A}$ and $\mathbf{A} = (\mathbf{C} + \frac{\delta}{2}\mathbf{I})^{-1}\mathbf{C}$.
- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \frac{\delta}{2}\nabla \log G(\mathbf{x}), \frac{\delta}{2}\mathbf{I})$;

[10] Titsias and Papaspiliopoulos (2018)

Exploiting Gaussian priors (and gradient info), continued

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Marginal gradient (mGRAD)[10] algorithm:

$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (\mathbf{I} - \mathbf{A})\mathbf{m} + \mathbf{A}[\mathbf{x} + \tfrac{\delta}{2}\nabla \log G(\mathbf{x})], \mathbf{B}),$$

  where $\mathbf{B} \coloneqq \frac{\delta}{2}\mathbf{A}^2 + \mathbf{A}$ and $\mathbf{A} = (\mathbf{C} + \frac{\delta}{2}\mathbf{I})^{-1}\mathbf{C}$.
- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \frac{\delta}{2}\nabla \log G(\mathbf{x}), \frac{\delta}{2}\mathbf{I})$;
  2. $\tilde{\mathbf{x}} \sim \mathrm{N}((\mathbf{I} - \mathbf{A})\mathbf{m} + \mathbf{A}\mathbf{u}, \frac{\delta}{2}\mathbf{A})$.

[10] Titsias and Papaspiliopoulos (2018)

Exploiting Gaussian priors (and gradient info), continued

Assuming $M(\mathbf{x}) = \mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$

- **[Marginal sampler]** Marginal gradient (mGRAD)[10] algorithm:

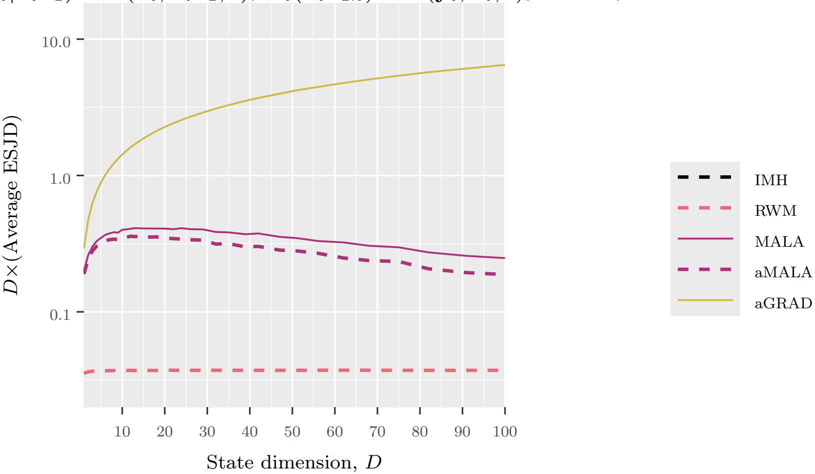
$$q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathrm{N}(\tilde{\mathbf{x}}; (\mathbf{I} - \mathbf{A})\mathbf{m} + \mathbf{A}[\mathbf{x} + \tfrac{\delta}{2}\nabla \log G(\mathbf{x})], \mathbf{B}),$$

  where $\mathbf{B} := \frac{\delta}{2}\mathbf{A}^2 + \mathbf{A}$ and $\mathbf{A} = (\mathbf{C} + \frac{\delta}{2}\mathbf{I})^{-1}\mathbf{C}$.
- Can sample from $q(\tilde{\mathbf{x}}|\mathbf{x})$ by sampling
  1. $\mathbf{u} \sim \mathrm{N}(\mathbf{x} + \frac{\delta}{2}\nabla \log G(\mathbf{x}), \frac{\delta}{2}\mathbf{I})$;
  2. $\tilde{\mathbf{x}} \sim \mathrm{N}((\mathbf{I} - \mathbf{A})\mathbf{m} + \mathbf{A}\mathbf{u}, \frac{\delta}{2}\mathbf{A})$.
- **[Auxiliary sampler]** Not integrating out $\mathbf{u}$ in the acceptance ratio gives the auxiliary gradient (aGRAD)[10] algorithm.

[10] Titsias and Papaspiliopoulos (2018)

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t;\mathbf{x}_{t-1},\mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t;\mathbf{x}_t,\mathbf{I})$; $T = 25$, $N = 31$

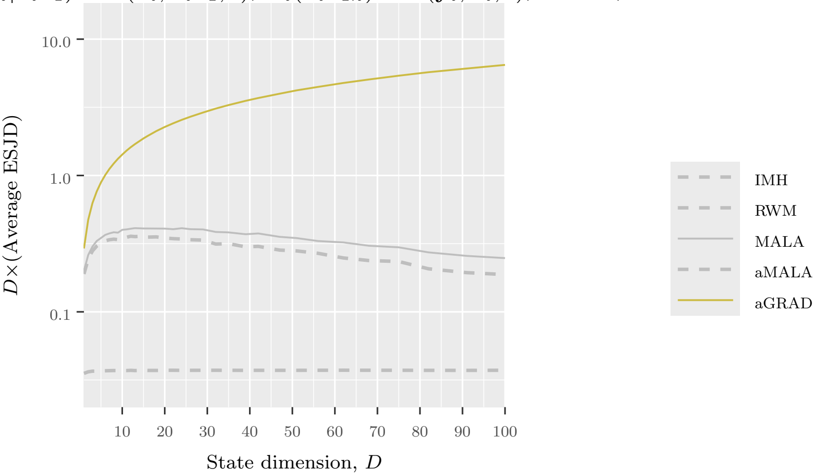


$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Summary of ‘classical’ MCMC methods

- **Summary:** ‘Classical’ MCMC methods can use ‘local’ moves guided by:

Summary of ‘classical’ MCMC methods

- **Summary:** ‘Classical’ MCMC methods can use ‘local’ moves guided by:
  - gradient information (e.g., as in MALA);

Summary of ‘classical’ MCMC methods

- **Summary:** ‘Classical’ MCMC methods can use ‘local’ moves guided by:
  - gradient information (e.g., as in MALA);
  - Gaussian prior information (e.g., as in Crank–Nicholson type methods and mGRAD/aGRAD).

Summary of ‘classical’ MCMC methods

- **Summary:** ‘Classical’ MCMC methods can use ‘local’ moves guided by:
  - gradient information (e.g., as in MALA);
  - Gaussian prior information (e.g., as in Crank–Nicholson type methods and mGRAD/aGRAD).

$\rightsquigarrow$ favourable scaling with D (for small, fixed T).

Summary of ‘classical’ MCMC methods

- **Summary:** ‘Classical’ MCMC methods can use ‘local’ moves guided by:
  - gradient information (e.g., as in MALA);
  - Gaussian prior information (e.g., as in Crank–Nicholson type methods and mGRAD/aGRAD).

  $\rightsquigarrow$ favourable scaling with D (for small, fixed T).
- **Problem:** ‘Classical’ MCMC methods do not exploit the ‘decorrelation-over-time’ property the state-space model.

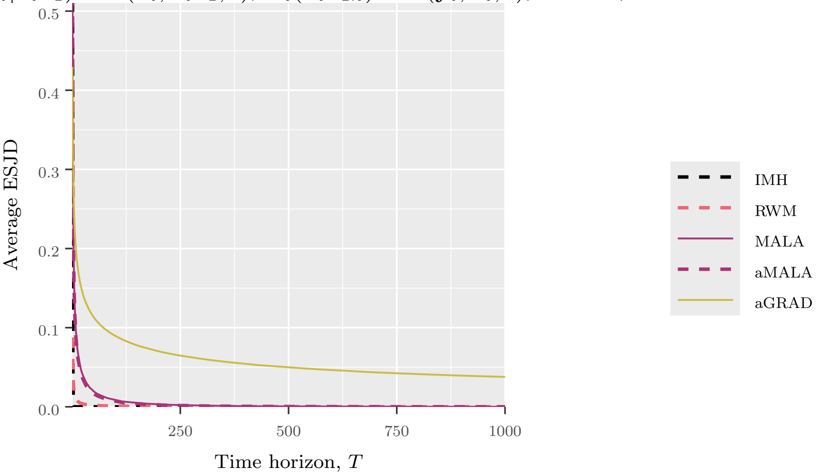
Summary of 'classical' MCMC methods

- **Summary:** 'Classical' MCMC methods can use 'local' moves guided by:
  - gradient information (e.g., as in MALA);
  - Gaussian prior information (e.g., as in Crank–Nicholson type methods and mGRAD/aGRAD).

  $\rightsquigarrow$ favourable scaling with D (for small, fixed T).
- **Problem:** 'Classical' MCMC methods do not exploit the 'decorrelation-over-time' property the state-space model.
  $\rightsquigarrow$ suboptimal scaling with T (for fixed D).

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$



$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in $T \rightsquigarrow$ horizontal line;
- must **increase** in $T \rightsquigarrow$ decreasing line.

Talk outline

Conditional sequential Monte Carlo (CSMC) algorithm

- For the moment: D small.

[11] Andrieu et al. (2010)

Conditional sequential Monte Carlo (CSMC) algorithm

- For the moment: D small.
- CSMC algorithm[11].

[11] Andrieu et al. (2010)

Conditional sequential Monte Carlo (CSMC) algorithm

- For the moment: D small.
- CSMC algorithm[11].
  - Induces π_T-invariant MCMC kernel.

[11] Andrieu et al. (2010)

Conditional sequential Monte Carlo (CSMC) algorithm

- For the moment: D small.
- CSMC algorithm[11].
  - Induces π_T-invariant MCMC kernel.
- Sequentially builds proposal in the 'time'-direction:

[11] Andrieu et al. (2010)

Conditional sequential Monte Carlo (CSMC) algorithm

- For the moment: D small.
- CSMC algorithm[11].
  - Induces π_T-invariant MCMC kernel.
- Sequentially builds proposal in the 'time'-direction:
  - using $N+1$ interacting samples ('particles'),

[11] Andrieu et al. (2010)

Conditional sequential Monte Carlo (CSMC) algorithm

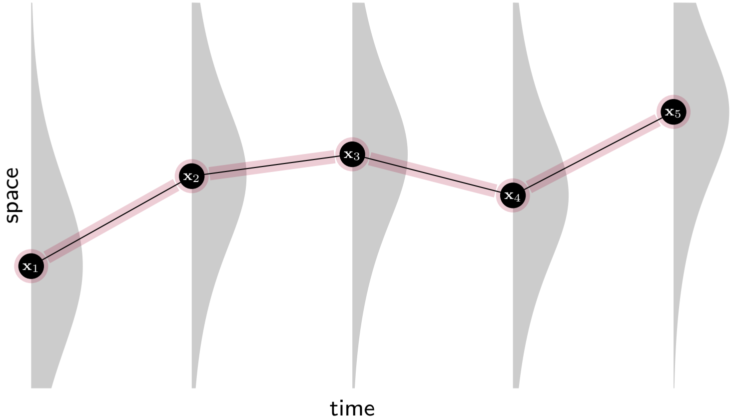
- For the moment: D small.
- CSMC algorithm[11].
  - Induces π_T-invariant MCMC kernel.
- Sequentially builds proposal in the 'time'-direction:
  - using $N+1$ interacting samples ('particles'),
  - avoids curse of dimensionality in T (for fixed D).

[11] Andrieu et al. (2010)

Algorithm 1 (CSMC). Given $\mathbf{x}_{1:T} \in \mathcal{X}^T$:

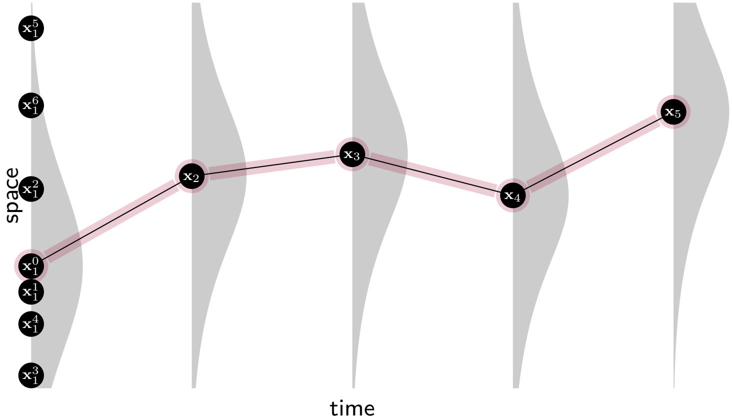
1. for $t = 1, \ldots, T$,
   1.1 set $\mathbf{x}_t^0 := \mathbf{x}_t$,
   1.2 **[resampling]** if $t > 1$, set $a_{t-1}^0 := 0$; sample $a_{t-1}^n = i$ w.p. W_{t-1}^i, for $n \in [N]$,
   1.3 **[sampling]** isample $\mathbf{x}_t^n \sim M_t(\cdot|\mathbf{x}_{t-1}^{a_{t-1}^n})$ for $n \in [N]$,
   1.4 **[weighting]** for $n \in [N]_0$, set $w_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.
   1.5 for $n \in [N]_0$, set $W_t^n := w_t^n / \sum_{m=0}^N w_t^m$;
2. sample $l_T = i \in [N]_0$ w.p. W_T^i.
3. **[ancestral tracing]** for $t = T-1, \ldots, 1$, set $l_t := a_t^{l_{t+1}}$.
4. return $\mathbf{x}'_{1:T} := (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_t^{l_T})$.

Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

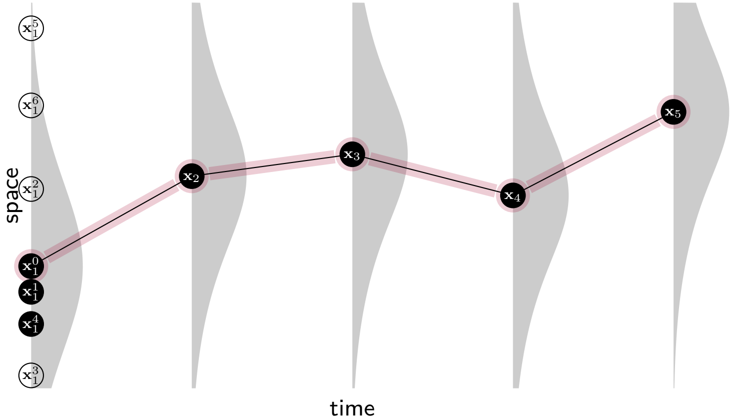
Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_1^0 := \mathbf{x}_1$.
- Sample $\mathbf{x}_1^{1:N} \sim \prod_{n=1}^N M_1(\mathbf{x}_1^n)$.

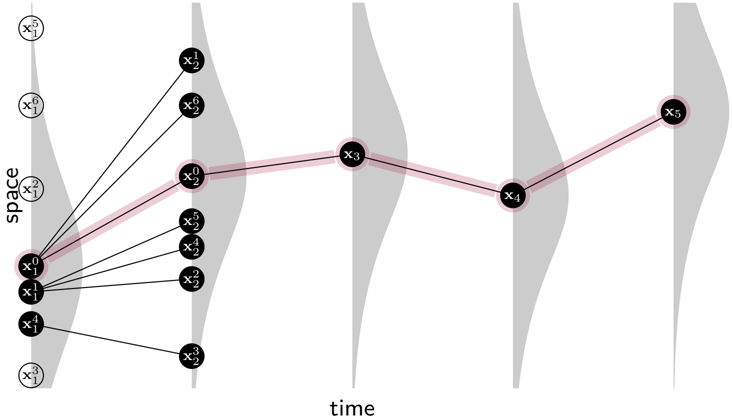
Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} M_t(\mathbf{x}_t^n | \mathbf{x}_{t-1}^{a_{t-1}^n})$,
  - where $W_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

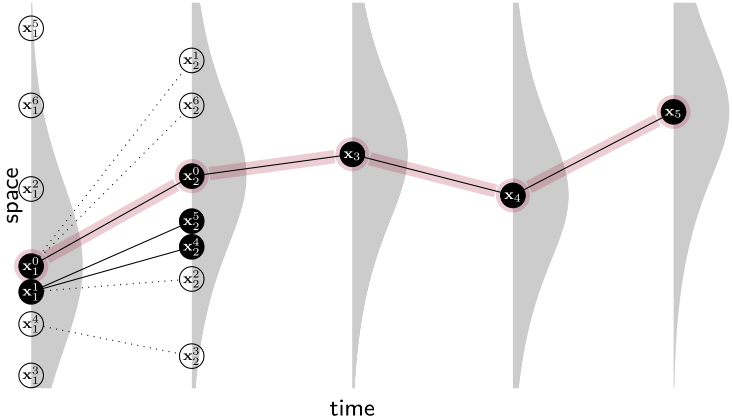
Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} M_t(\mathbf{x}_t^n | \mathbf{x}_{t-1}^{a_{t-1}^n})$,
  - where $W_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

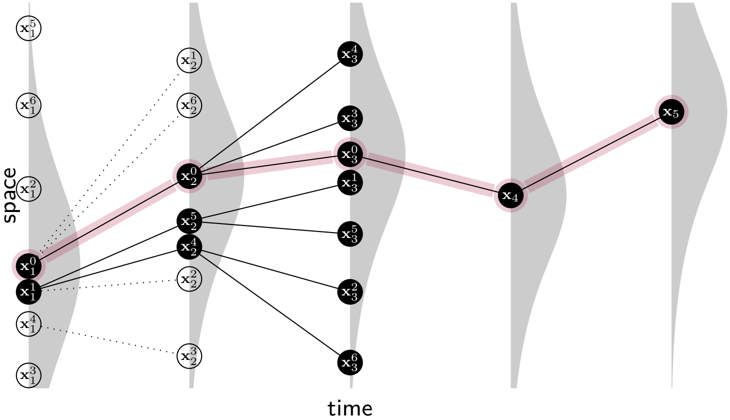
Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} M_t(\mathbf{x}_t^n | \mathbf{x}_{t-1}^{a_{t-1}^n})$,
  - where $W_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

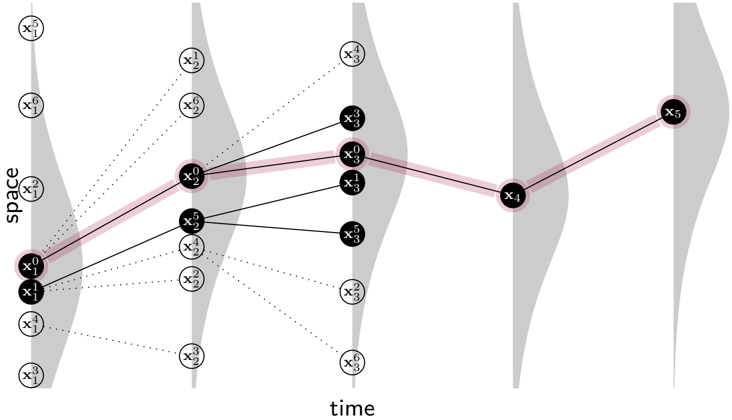
Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} M_t(\mathbf{x}_t^n | \mathbf{x}_{t-1}^{a_{t-1}^n})$,
  - where $W_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

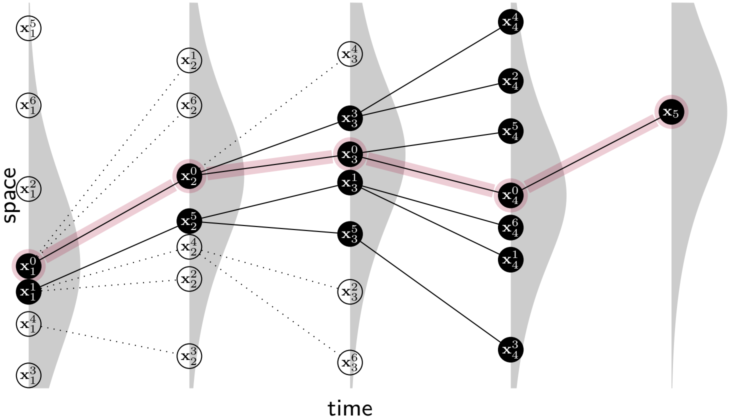
Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} M_t(\mathbf{x}_t^n | \mathbf{x}_{t-1}^{a_{t-1}^n})$,
  - where $W_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

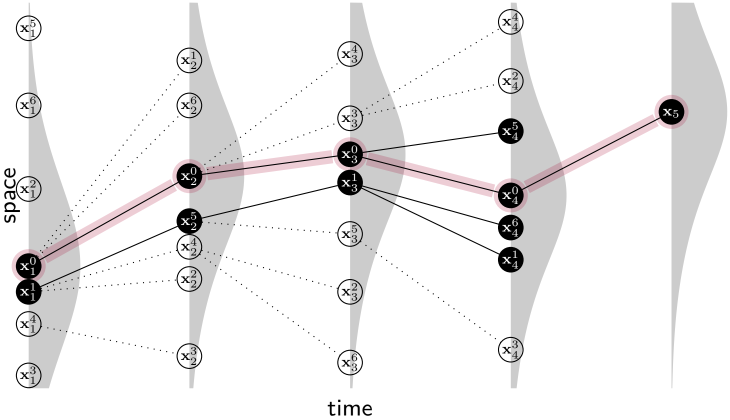
Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} M_t(\mathbf{x}_t^n | \mathbf{x}_{t-1}^{a_{t-1}^n})$,
  - where $W_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

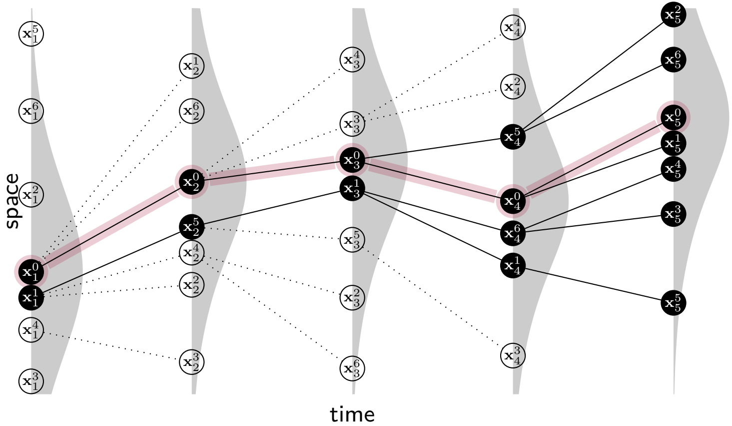
Proposal



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} M_t(\mathbf{x}_t^n | \mathbf{x}_{t-1}^{a_{t-1}^n})$,
  - where $W_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

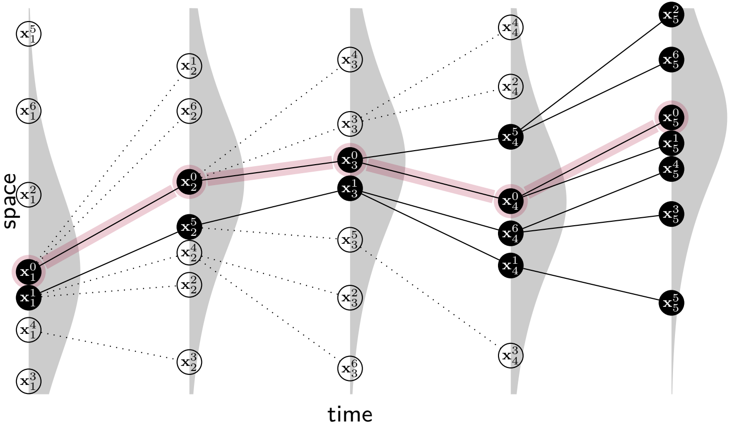
Proposal



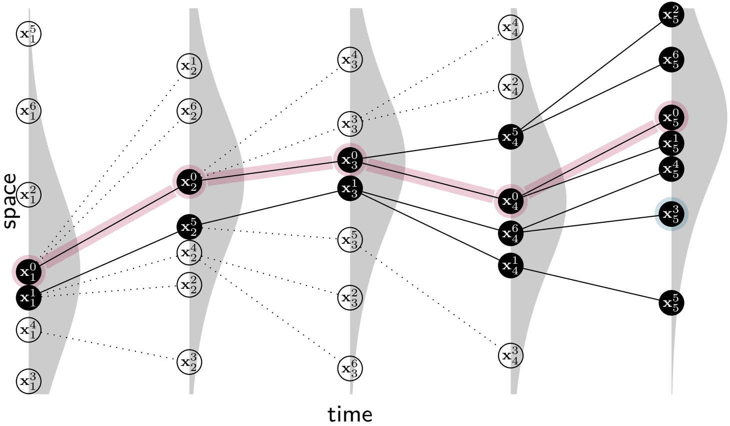
Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} M_t(\mathbf{x}_t^n | \mathbf{x}_{t-1}^{a_{t-1}^n})$,
  - where $W_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

Selecting new reference path

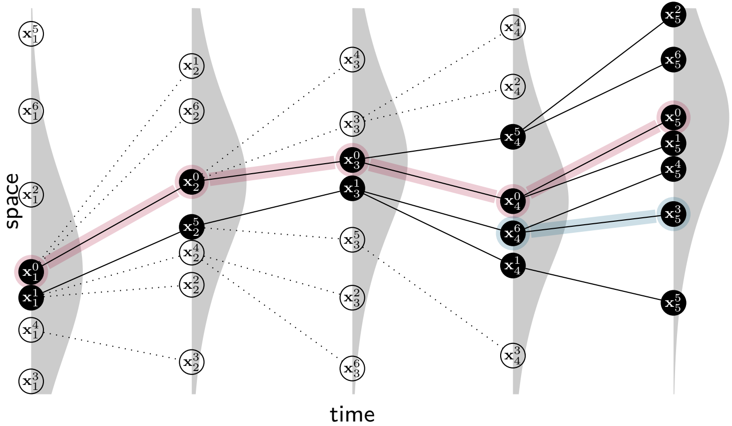


Selecting new reference path



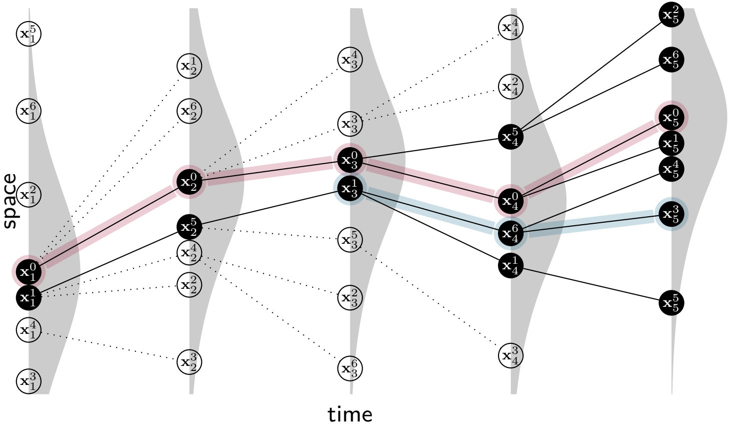
1. Sample $l_T \sim W_T^{l_T}$.

Selecting new reference path



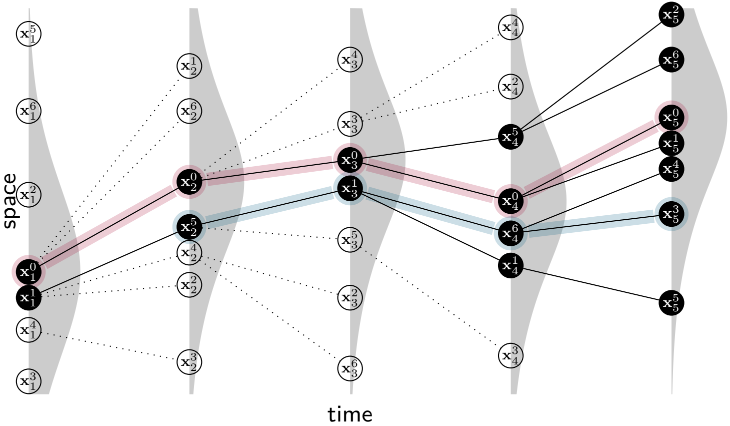
1. Sample $l_T \sim W_T^{l_T}$.
2. Set $l_t := a_t^{l_{t+1}}$, for $t = T-1, \ldots, 1$.

Selecting new reference path



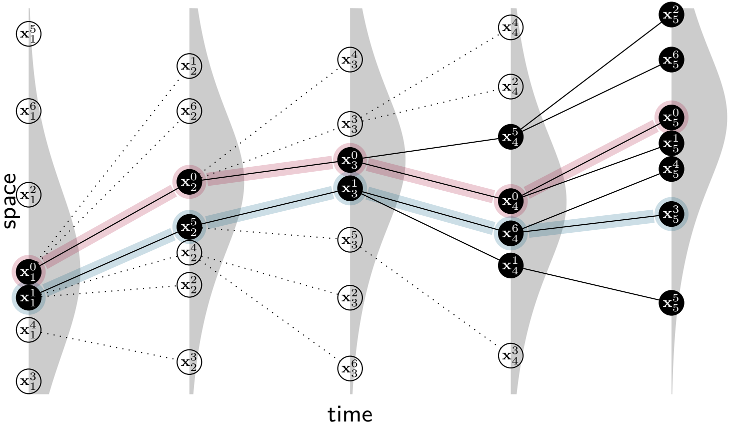
1. Sample $l_T \sim W_T^{l_T}$.
2. Set $l_t := a_t^{l_{t+1}}$, for $t = T-1, \ldots, 1$.

Selecting new reference path



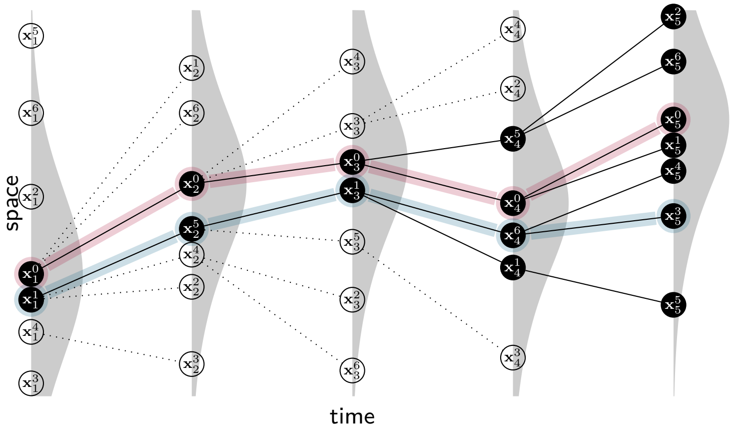
1. Sample $l_T \sim W_T^{l_T}$.
2. Set $l_t := a_t^{l_{t+1}}$, for $t = T-1, \ldots, 1$.

Selecting new reference path



1. Sample $l_T \sim W_T^{l_T}$.
2. Set $l_t := a_t^{l_{t+1}}$, for $t = T-1, \ldots, 1$.

Selecting new reference path

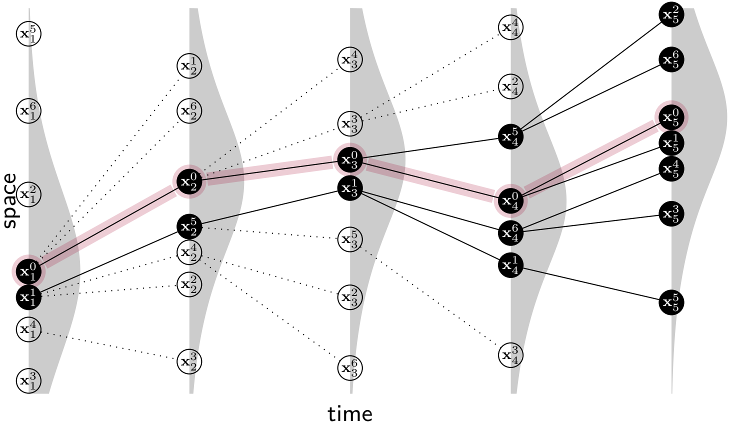


1. Sample $l_T \sim W_T^{l_T}$.
2. Set $l_t := a_t^{l_{t+1}}$, for $t = T-1, \dots, 1$.
3. Return $\mathbf{x}'_{1:T} := (\mathbf{x}_1^{l_1}, \dots, \mathbf{x}_T^{l_T})$ (new state of MCMC chain).

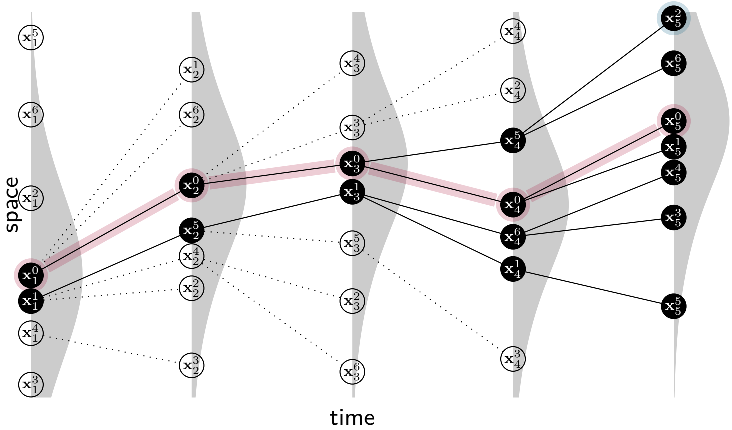
- induces π_T-invariant MCMC kernel $P_{\mathsf{CSMC}}(\mathbf{x}'_{1:T}|\mathbf{x}_{1:T})$.

- induces π_T-invariant MCMC kernel $P_{\mathsf{CSMC}}(\mathbf{x}'_{1:T}|\mathbf{x}_{1:T})$.
- T "accept-reject decisions".

Mixing

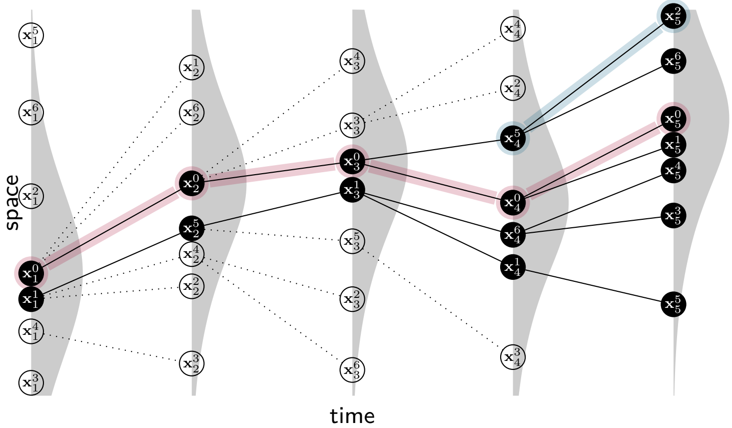


Mixing



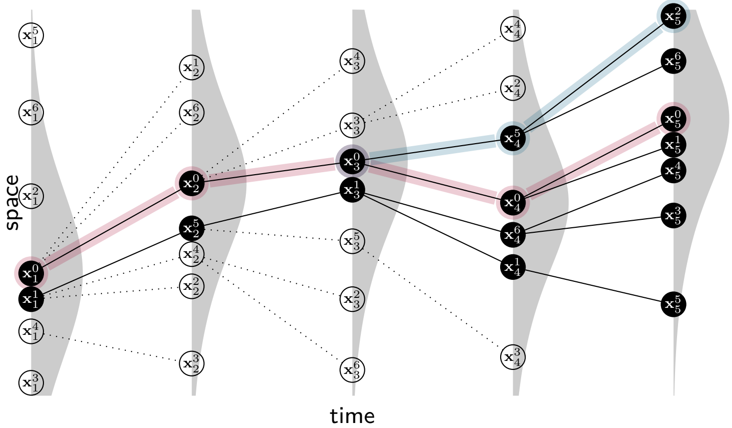
x_1^5
x_1^6
x_1^2
x_1^0
x_1^1
x_1^4
x_1^3
x_2^1
x_2^6
x_2^0
x_2^5
x_2^4
x_2^2
x_2^3
x_3^4
x_3^3
x_3^0
x_3^1
x_3^5
x_3^2
x_3^6
x_4^4
x_4^2
x_4^5
x_4^0
x_4^6
x_4^1
x_4^3
x_5^2
x_5^6
x_5^0
x_5^1
x_5^4
x_5^3
x_5^5
space
time

Mixing

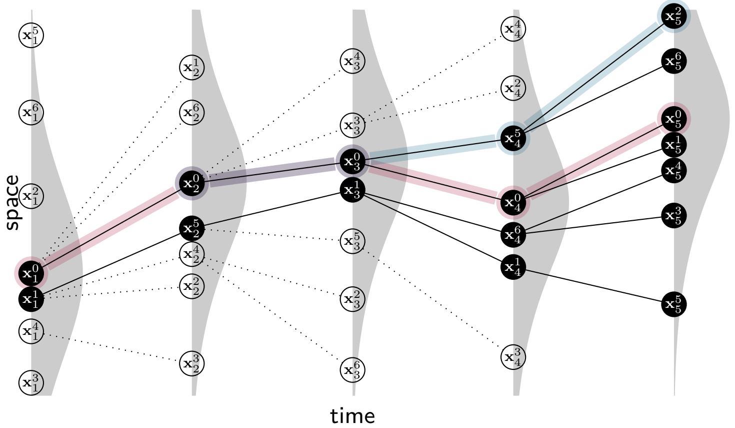


$\mathbf{x}_1^5$
$\mathbf{x}_1^6$
$\mathbf{x}_1^2$
$\mathbf{x}_1^0$
$\mathbf{x}_1^1$
$\mathbf{x}_1^4$
$\mathbf{x}_1^3$
$\mathbf{x}_2^1$
$\mathbf{x}_2^6$
$\mathbf{x}_2^0$
$\mathbf{x}_2^5$
$\mathbf{x}_2^4$
$\mathbf{x}_2^2$
$\mathbf{x}_2^3$
$\mathbf{x}_3^4$
$\mathbf{x}_3^3$
$\mathbf{x}_3^0$
$\mathbf{x}_3^1$
$\mathbf{x}_3^5$
$\mathbf{x}_3^2$
$\mathbf{x}_3^6$
$\mathbf{x}_4^4$
$\mathbf{x}_4^2$
$\mathbf{x}_4^5$
$\mathbf{x}_4^0$
$\mathbf{x}_4^6$
$\mathbf{x}_4^1$
$\mathbf{x}_4^3$
$\mathbf{x}_5^2$
$\mathbf{x}_5^6$
$\mathbf{x}_5^0$
$\mathbf{x}_5^1$
$\mathbf{x}_5^4$
$\mathbf{x}_5^3$
$\mathbf{x}_5^5$
space
time

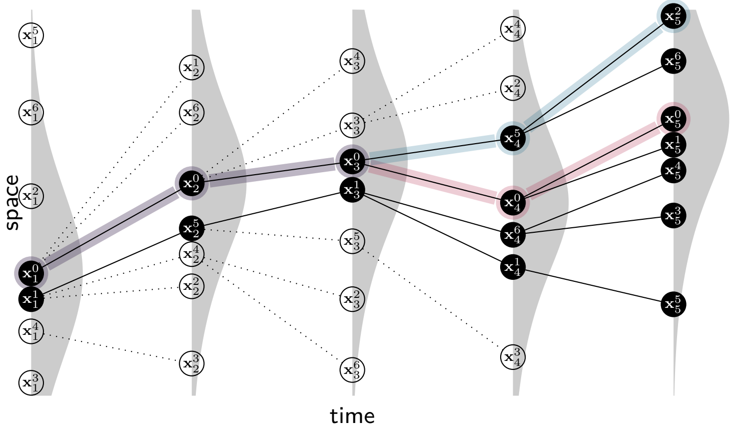
Mixing



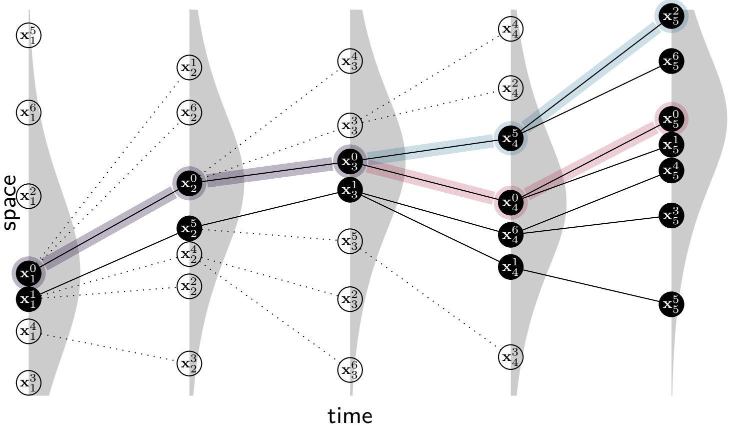
Mixing



Mixing

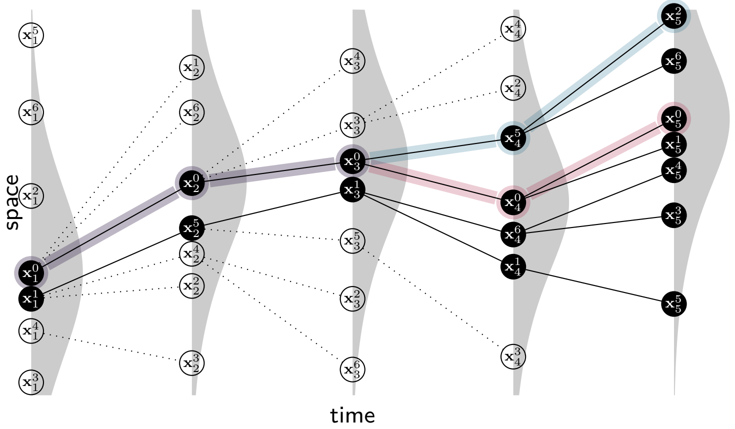


Mixing



Problem: $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$ & $\mathbf{x}_{1:T} = (\mathbf{x}_1^{0}, \ldots, \mathbf{x}_T^{0})$ coalesce

Mixing



Problem: $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$ & $\mathbf{x}_{1:T} = (\mathbf{x}_1^0, \ldots, \mathbf{x}_T^0)$ coalesce

- controlling the 'acceptance rates' requires $N \sim T$ (Andrieu et al., 2018; Koskela et al., 2020)

Algorithm 2 (CSMC). Given $\mathbf{x}_{1:T} \in \mathcal{X}^T$:

1. for $t = 1, \ldots, T$,
   1.1 set $\mathbf{x}_t^0 := \mathbf{x}_t$,
   1.2 **[resampling]** if $t > 1$, set $a_{t-1}^0 := 0$; sample $a_{t-1}^n = i$ w.p. W_{t-1}^i, for $n \in [N]$,
   1.3 **[sampling]** sample $\mathbf{x}_t^n \sim M_t(\cdot \,|\mathbf{x}_{t-1}^{a_{t-1}^n})$ for $n \in [N]$,
   1.4 **[weighting]** for $n \in [N]_0$, set $w_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.
   1.5 for $n \in [N]_0$, set $W_t^n := w_t^n / \sum_{m=0}^N w_t^m$;
2. sample $l_T = i \in [N]_0$ w.p. W_T^i.
3. **[ancestral tracing]** for $t = T-1, \ldots, 1$, set $l_t := a_t^{l_{t+1}}$.
4. return $\mathbf{x}'_{1:T} := (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_t^{l_T})$.

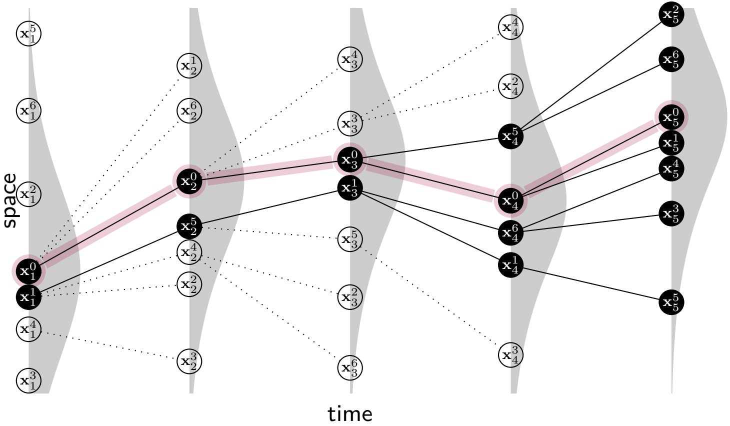
Algorithm 2 (CSMC). Given $\mathbf{x}_{1:T} \in \mathcal{X}^T$:

1. for $t = 1, \ldots, T$,
   1.1 set $\mathbf{x}_t^0 := \mathbf{x}_t$,
   1.2 **[resampling]** if $t > 1$, set $a_{t-1}^0 := 0$; sample $a_{t-1}^n = i$ w.p. W_{t-1}^i, for $n \in [N]$,
   1.3 **[sampling]** sample $\mathbf{x}_t^n \sim M_t(\cdot \,|\mathbf{x}_{t-1}^{a_{t-1}^n})$ for $n \in [N]$,
   1.4 **[weighting]** for $n \in [N]_0$, set $w_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.
   1.5 for $n \in [N]_0$, set $W_t^n := w_t^n / \sum_{m=0}^N w_t^m$;
2. sample $i \in [N]$ w.p. $\dfrac{W_T^i}{1 - W_T^0}$; set $l_T := i$ w.p. $1 \wedge \dfrac{1 - W_T^0}{1 - W_T^i}$; otherwise, set $l_T := 0$;
3. **[ancestral tracing]** for $t = T-1, \ldots, 1$, set $l_t := a_t^{l_{t+1}}$.
4. return $\mathbf{x}'_{1:T} := (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_t^{l_T})$.

Algorithm 2 (CSMC). Given $\mathbf{x}_{1:T} \in \mathcal{X}^T$:

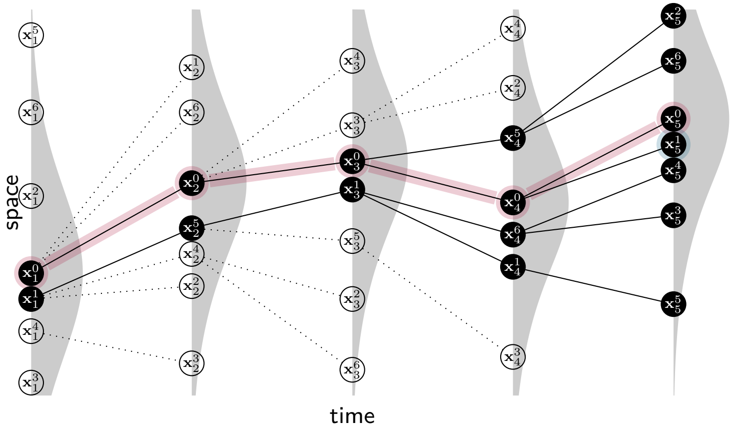
1. for $t = 1, \ldots, T$,
   1.1 set $\mathbf{x}_t^0 := \mathbf{x}_t$,
   1.2 **[resampling]** if $t > 1$, set $a_{t-1}^0 := 0$; sample $a_{t-1}^n = i$ w.p. W_{t-1}^i, for $n \in [N]$,
   1.3 **[sampling]** sample $\mathbf{x}_t^n \sim M_t(\,\cdot\,|\mathbf{x}_{t-1}^{a_{t-1}^n})$ for $n \in [N]$,
   1.4 **[weighting]** for $n \in [N]_0$, set $w_t^n \propto G_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.
   1.5 for $n \in [N]_0$, set $W_t^n := w_t^n / \sum_{m=0}^N w_t^m$;
2. sample $i \in [N]$ w.p. $\dfrac{W_T^i}{1 - W_T^0}$; set $l_T := i$ w.p. $1 \wedge \dfrac{1 - W_T^0}{1 - W_T^i}$; otherwise, set $l_T := 0$;
3. **[backward sampling]** for $t = T-1, \ldots, 1$, sample $l_t = i \in [N]_0$ w.p. $\dfrac{W_t^i Q_{t+1}(\mathbf{x}_t^i, \mathbf{x}_{t+1}^{l_{t+1}})}{\sum_{n=0}^N W_t^n Q_{t+1}(\mathbf{x}_t^n, \mathbf{x}_{t+1}^{l_{t+1}})}$;
4. return $\mathbf{x}'_{1:T} := (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_t^{l_T})$.

Backward-sampling extension



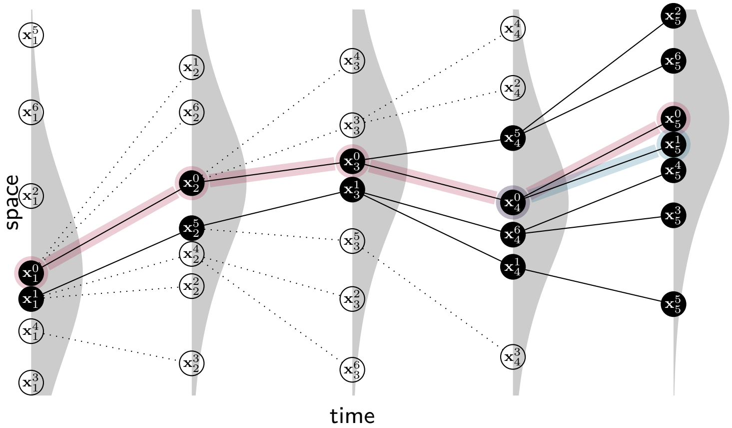
- Forms new lineage $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$.

Backward-sampling extension



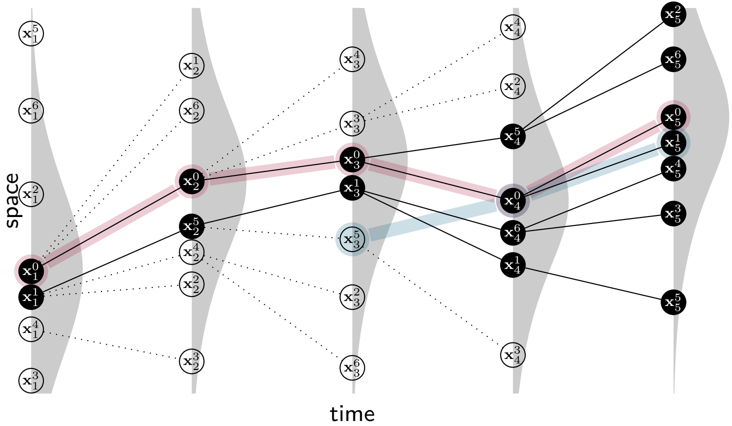
- Forms new lineage $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$.

Backward-sampling extension



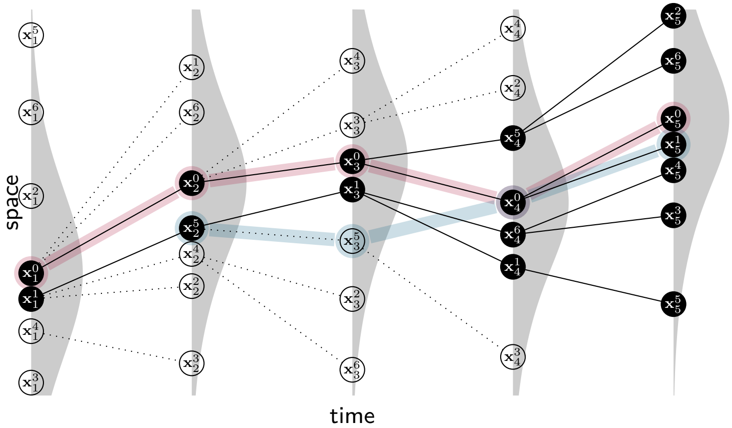
- Forms new lineage $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$.

Backward-sampling extension



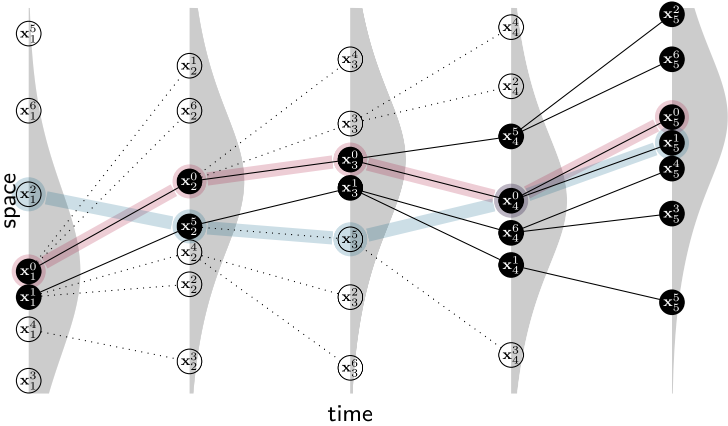
- Forms new lineage $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$.

Backward-sampling extension



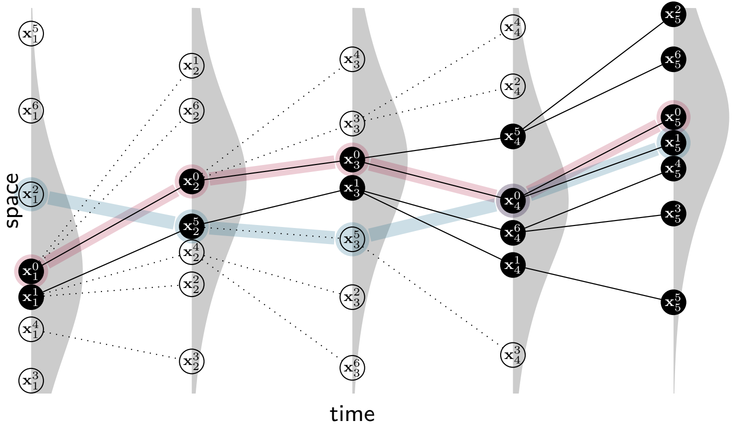
- Forms new lineage $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$.

Backward-sampling extension



- Forms new lineage $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$.

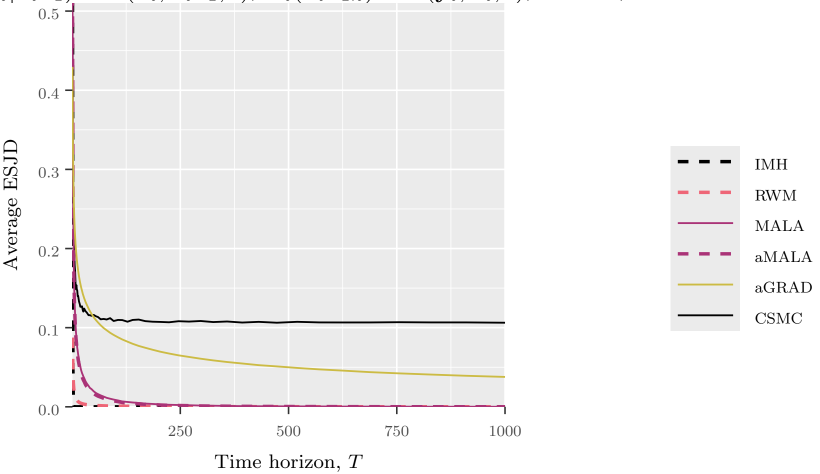
Backward-sampling extension



- Forms new lineage $\mathbf{x}'_{1:T} = (\mathbf{x}_1^{l_1}, \ldots, \mathbf{x}_T^{l_T})$.
- Frees us from having to grow N with T (Lee et al., 2020).

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

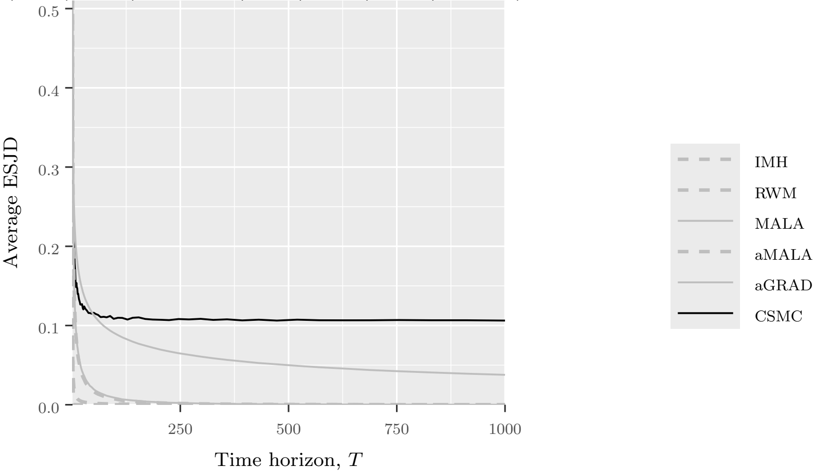


(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in $T \rightsquigarrow$ horizontal line;
- must **increase** in $T \rightsquigarrow$ decreasing line.

Scaling with T

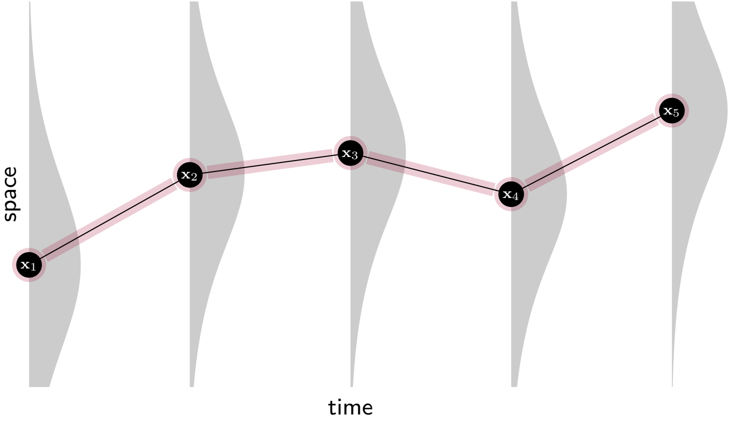
$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$



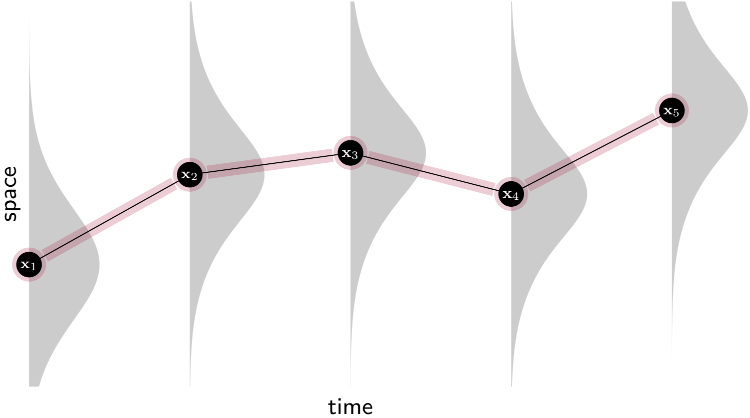
(Average ESJD) $= \frac{1}{TD} \sum_{t=1}^{T} \sum_{d=1}^{D} (x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in $T \rightsquigarrow$ horizontal line;
- must **increase** in $T \rightsquigarrow$ decreasing line.

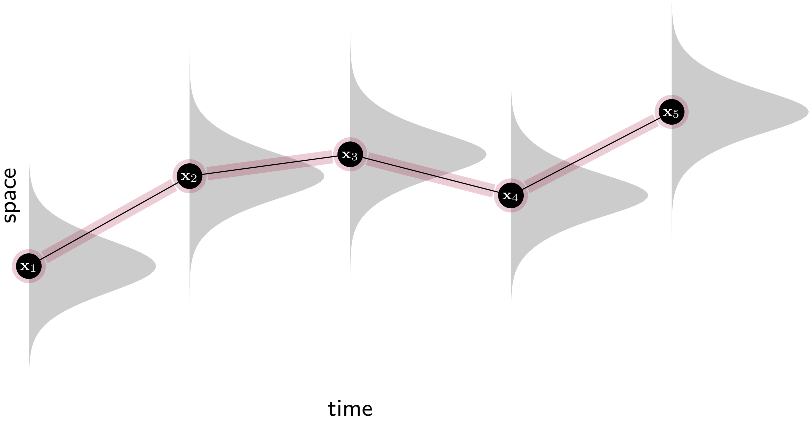
Breakdown of CSMC as $D \to \infty$



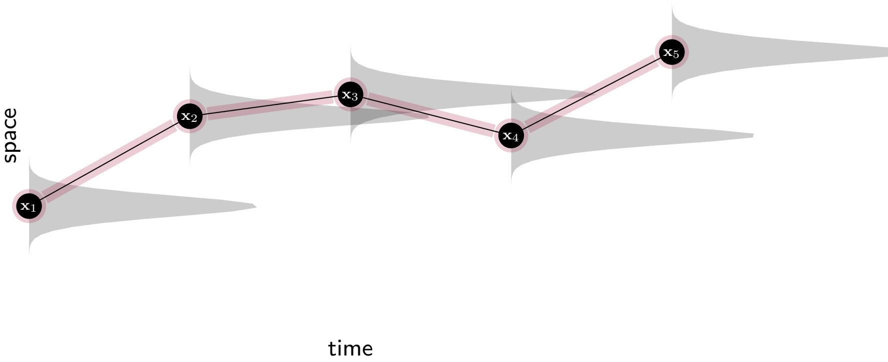
Breakdown of CSMC as $D \to \infty$



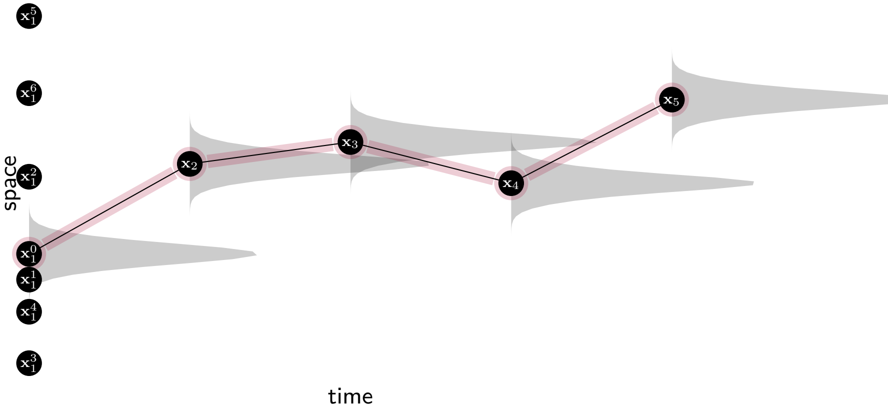
Breakdown of CSMC as $D \to \infty$



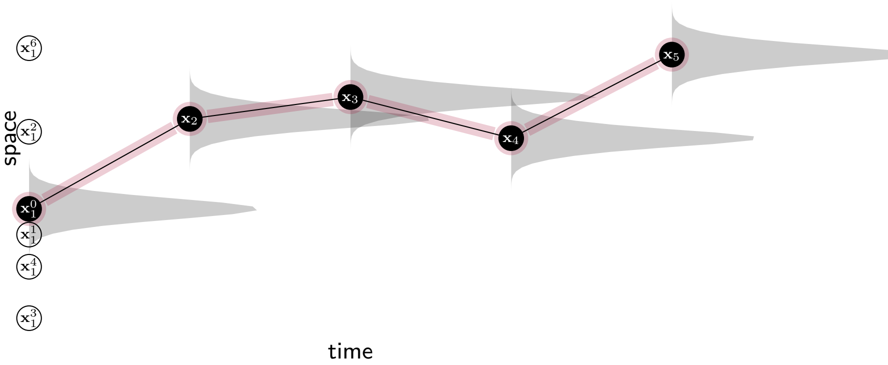
Breakdown of CSMC as $D \to \infty$



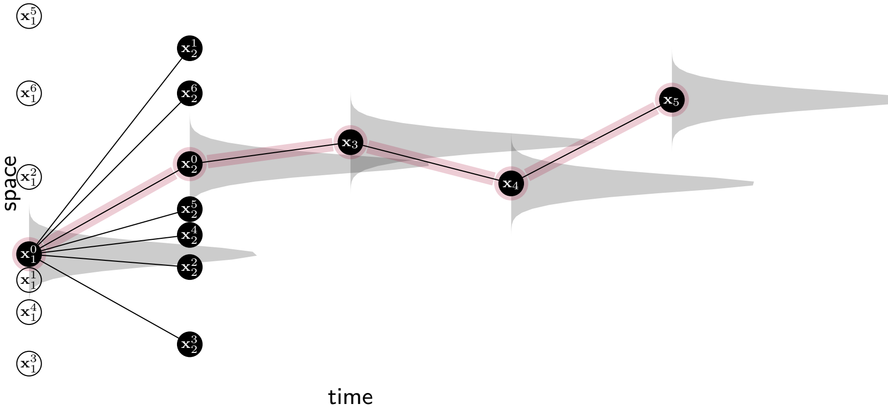
Breakdown of CSMC as $D \to \infty$



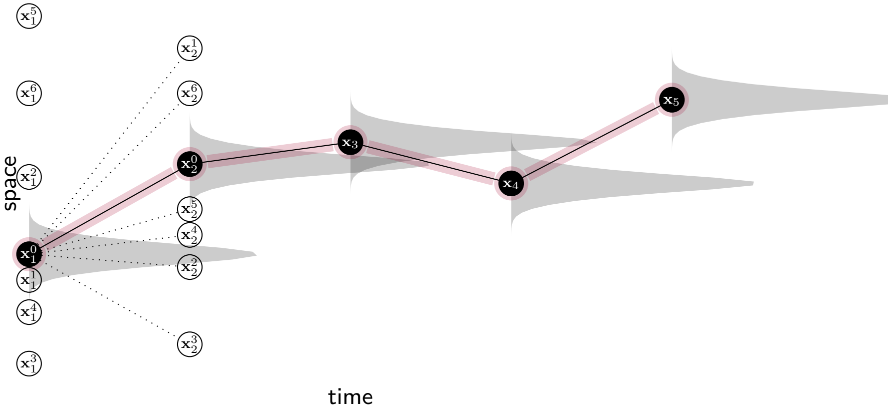
Breakdown of CSMC as $D \to \infty$



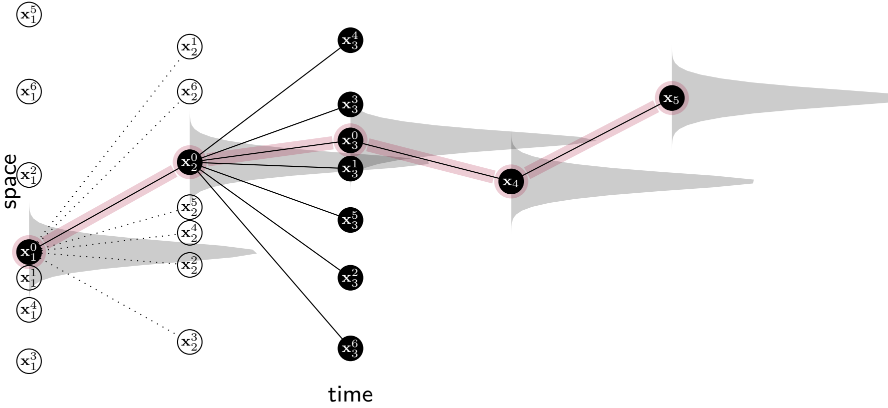
Breakdown of CSMC as $D \to \infty$



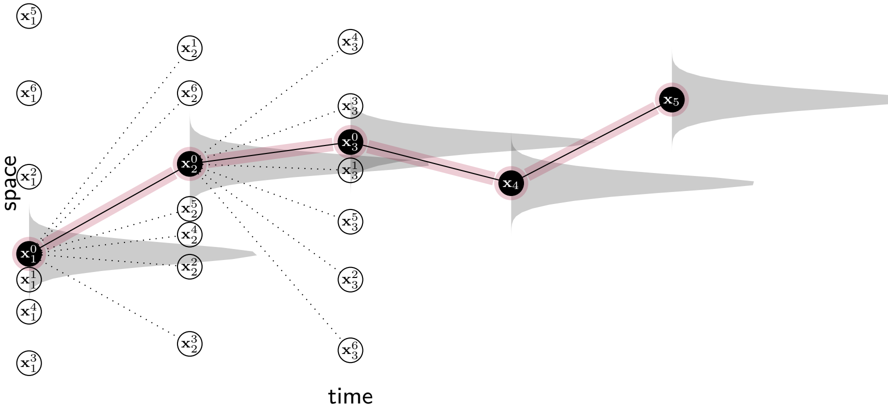
Breakdown of CSMC as $D \to \infty$



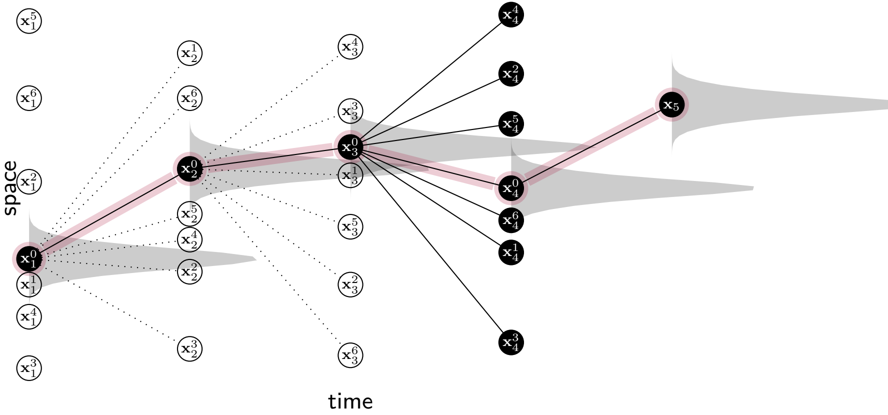
Breakdown of CSMC as $D \to \infty$



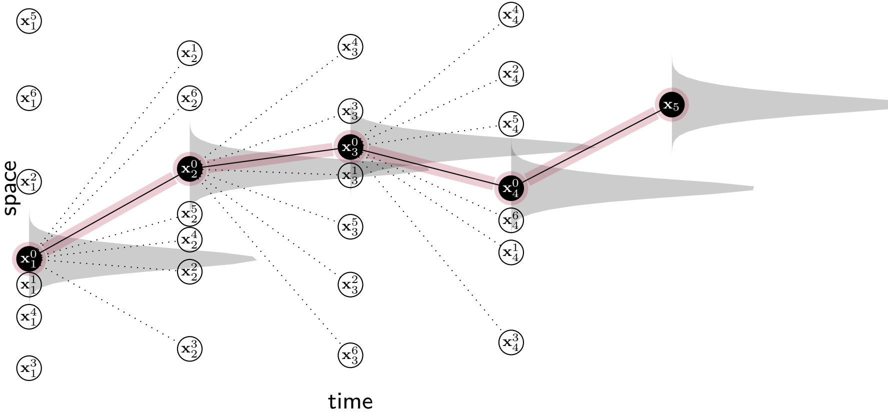
Breakdown of CSMC as $D \to \infty$



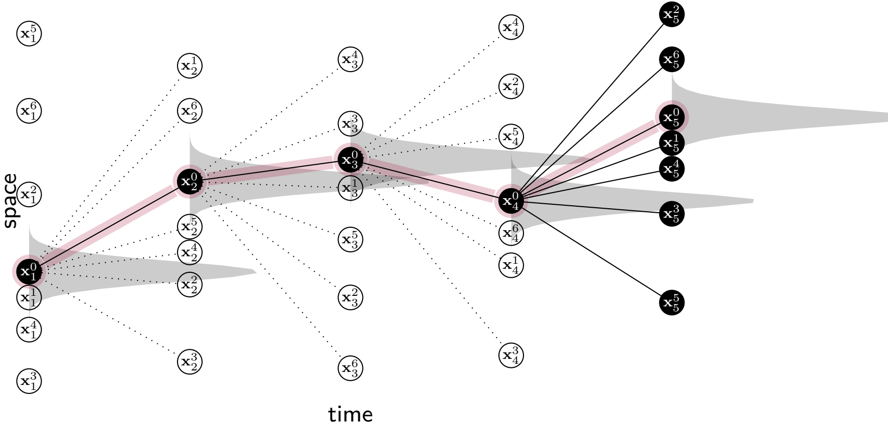
Breakdown of CSMC as $D \to \infty$



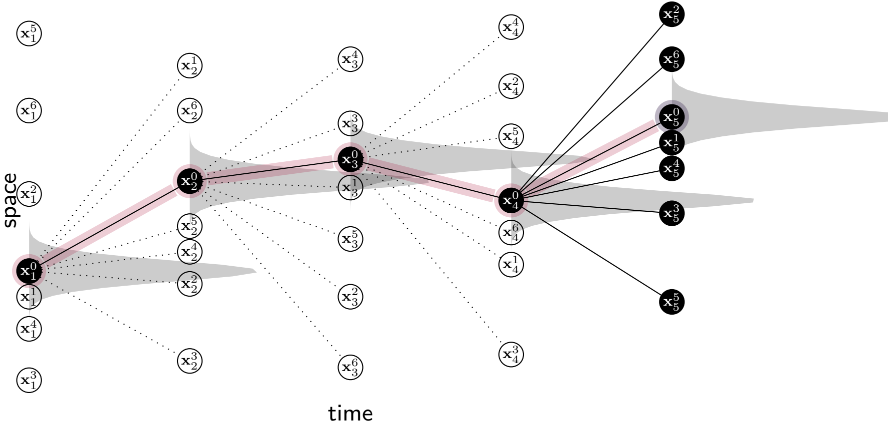
Breakdown of CSMC as $D \to \infty$



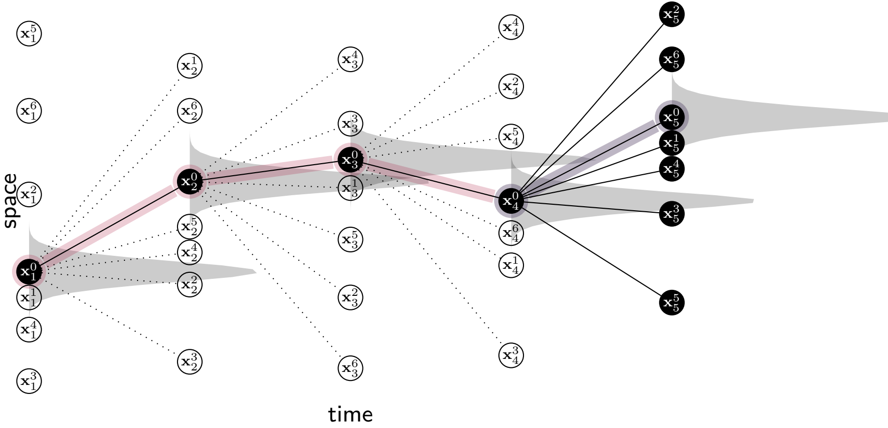
Breakdown of CSMC as $D \to \infty$



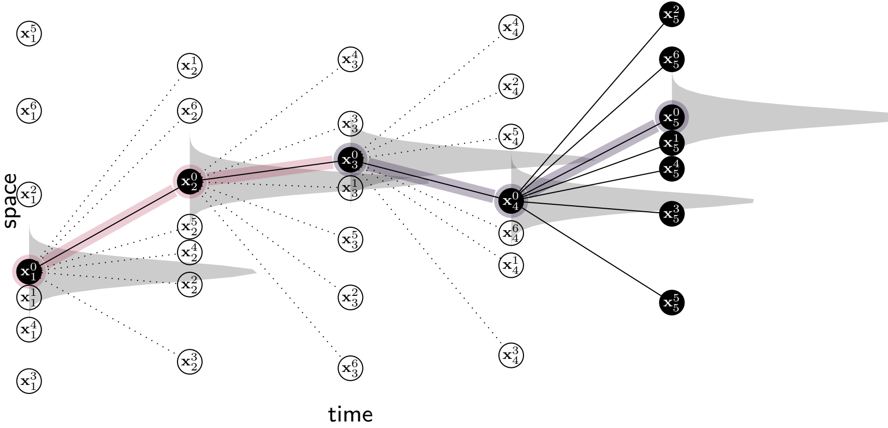
Breakdown of CSMC as $D \to \infty$



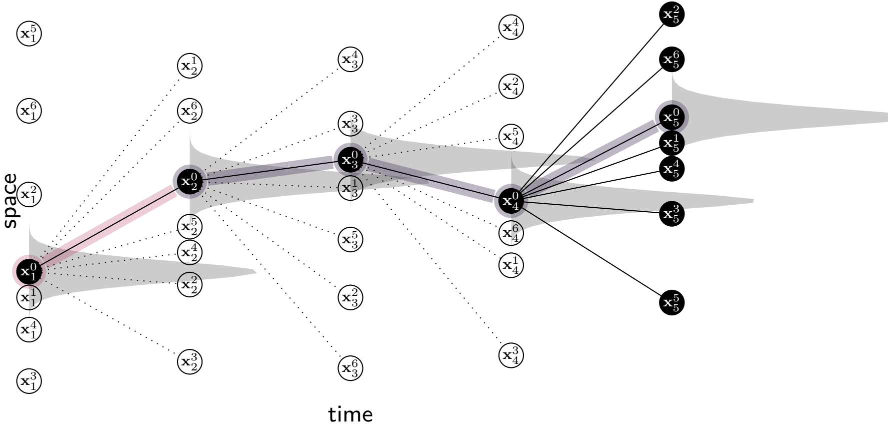
Breakdown of CSMC as $D \to \infty$



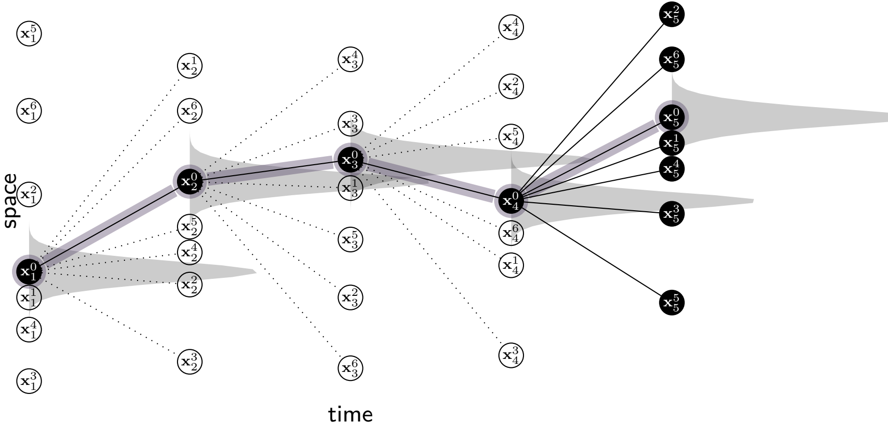
Breakdown of CSMC as $D \to \infty$



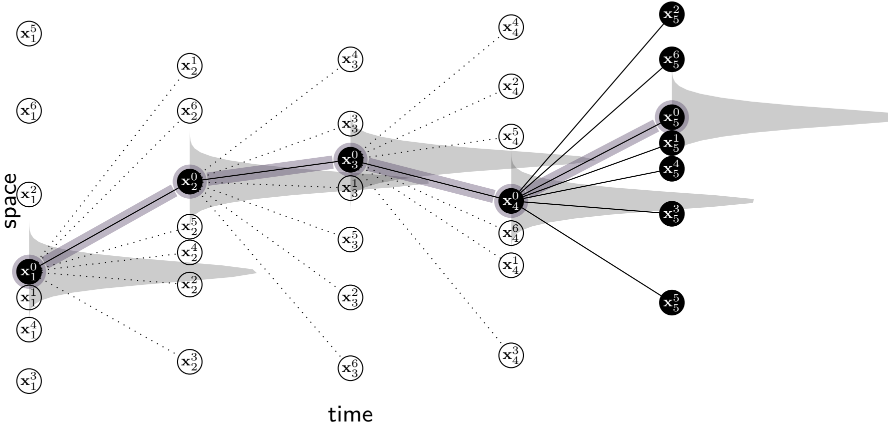
Breakdown of CSMC as $D \to \infty$



Breakdown of CSMC as $D \to \infty$

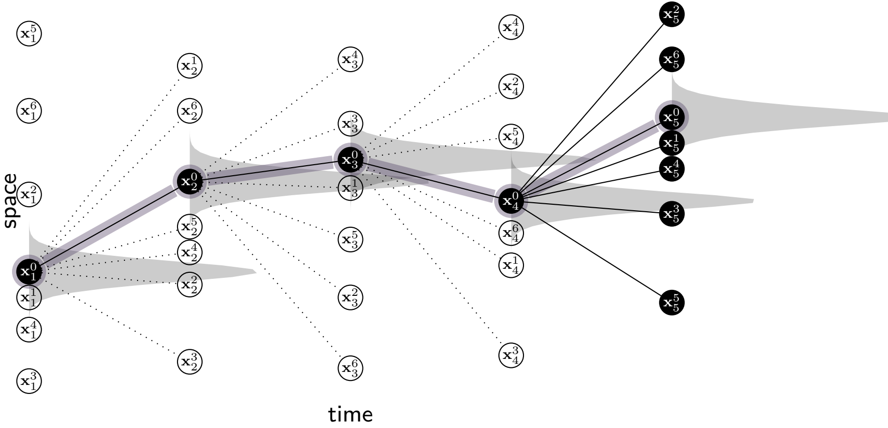


Breakdown of CSMC as $D \to \infty$



- all acceptance rates $\to 0$ (Finke and Thiery, 2023);

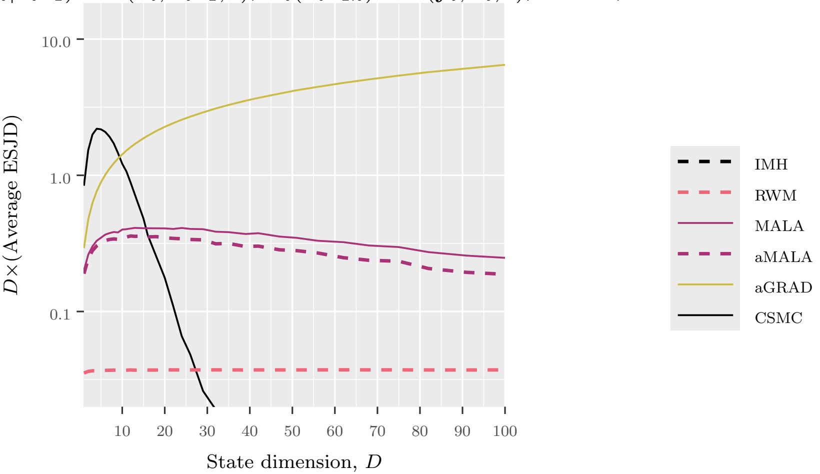
Breakdown of CSMC as $D \to \infty$



- all acceptance rates $\to 0$ (Finke and Thiery, 2023);
- even with backward sampling.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$

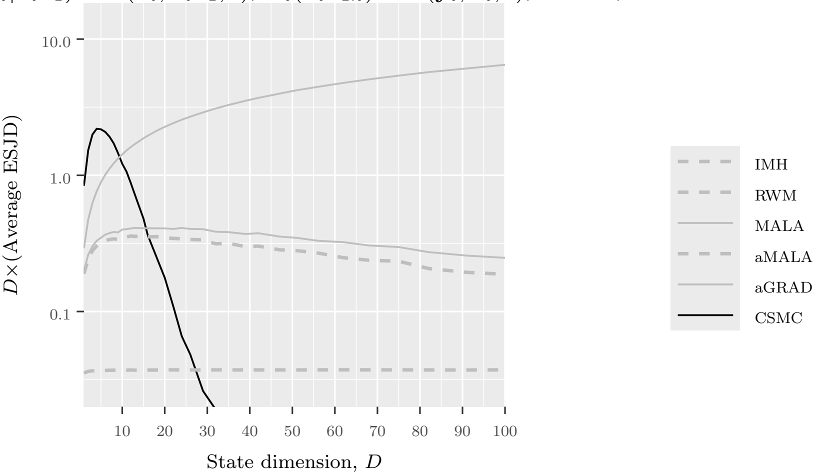


$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Summary of the CSMC algorithm

- **Summary:** The CSMC algorithm exploits the 'decorrelation-over-time' property the state-space model.

Summary of the CSMC algorithm

- **Summary:** The CSMC algorithm exploits the 'decorrelation-over-time' property the state-space model.
  $\rightsquigarrow$ favourable scaling with T (for small, fixed D).

Summary of the CSMC algorithm

- **Summary:** The CSMC algorithm exploits the 'decorrelation-over-time' property the state-space model.
  $\rightsquigarrow$ favourable scaling with T (for small, fixed D).
- **Problem:** The CSMC algorithm cannot use 'local' moves.

Summary of the CSMC algorithm

- **Summary:** The CSMC algorithm exploits the 'decorrelation-over-time' property the state-space model.
  $\rightsquigarrow$ favourable scaling with T (for small, fixed D).
- **Problem:** The CSMC algorithm cannot use 'local' moves.
  $\rightsquigarrow$ curse of dimension in D (for fixed T).

Talk outline

Particle random-walk Metropolis (Particle-RWM)

Finke and Thiery (2023)

Algorithm 3 (Particle-RWM). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t, \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** for $n \in [N]_0$, set $w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

---

[12] Finke and Thiery (2023); see also Malory (2021)

Particle random-walk Metropolis (Particle-RWM)

Finke and Thiery (2023)

Algorithm 3 (Particle-RWM). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t, \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** for $n \in [N]_0$, set $w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t, \delta_t \mathbf{I}).$$

[12] Finke and Thiery (2023); see also Malory (2021)

Particle random-walk Metropolis (Particle-RWM)

Finke and Thiery (2023)

Algorithm 3 (Particle-RWM). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t, \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** for $n \in [N]_0$, set $w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t, \delta_t \mathbf{I}).$$

- Reduces to RWM if $N = T = 1$.

[12] Finke and Thiery (2023); see also Malory (2021)

Particle random-walk Metropolis (Particle-RWM)

Finke and Thiery (2023)

Algorithm 3 (Particle-RWM). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t, \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** for $n \in [N]_0$, set $w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

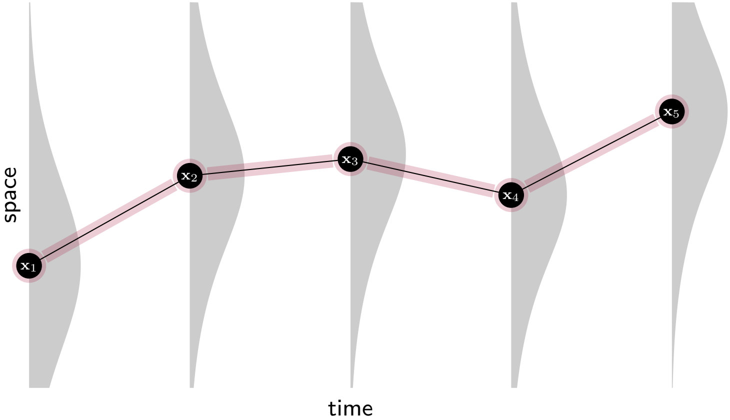
- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t, \delta_t \mathbf{I}).$$

- Reduces to RWM if $N = T = 1$.
- Dimensionally stable if $\delta_t = \mathrm{O}(D^{-1})$.[12]

[12] Finke and Thiery (2023); see also Malory (2021)

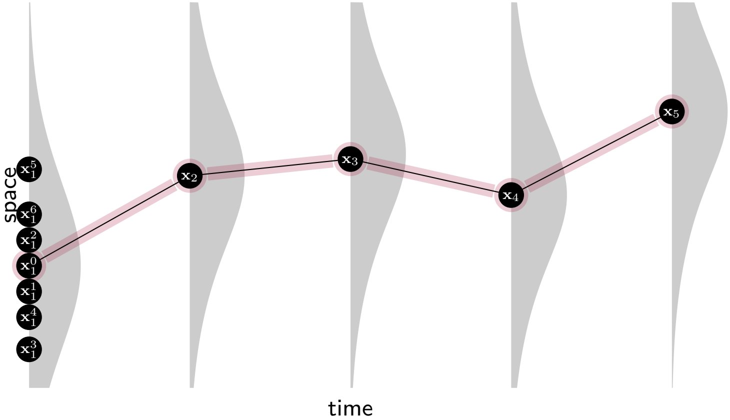
Particle-RWM ($D \to \infty$)



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_1^0 := \mathbf{x}_1$.
- Sample $(\mathbf{u}_1, \mathbf{x}_1^{1:N}) \sim \mathrm{N}(\mathbf{u}_1; \mathbf{x}_1^0, \frac{\delta_1}{2}\mathbf{I}) \prod_{n=1}^N \mathrm{N}(\mathbf{x}_1^n; \mathbf{u}_1, \frac{\delta_1}{2}\mathbf{I})$.

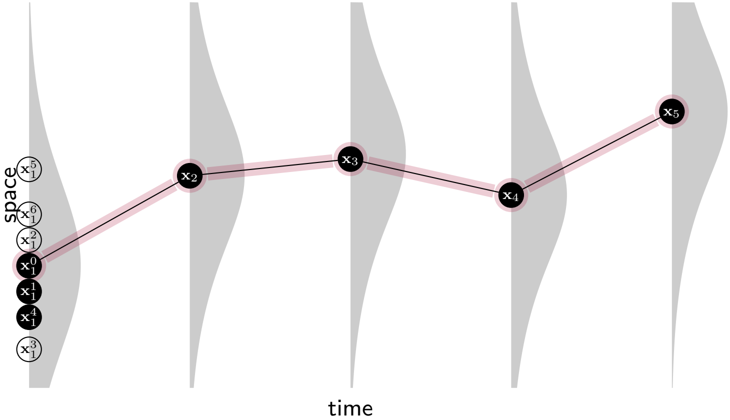
Particle-RWM ($D \to \infty$)



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_1^0 := \mathbf{x}_1$.
- Sample $(\mathbf{u}_1, \mathbf{x}_1^{1:N}) \sim \mathrm{N}(\mathbf{u}_1; \mathbf{x}_1^0, \frac{\delta_1}{2}\mathbf{I}) \prod_{n=1}^N \mathrm{N}(\mathbf{x}_1^n; \mathbf{u}_1, \frac{\delta_1}{2}\mathbf{I})$.

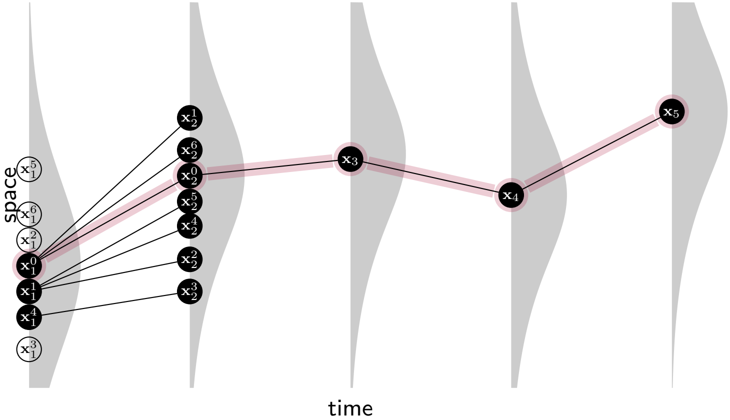
Particle-RWM $(D \to \infty)$



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_1^0 := \mathbf{x}_1$.
- Sample $(\mathbf{u}_1, \mathbf{x}_1^{1:N}) \sim \mathrm{N}(\mathbf{u}_1; \mathbf{x}_1^0, \frac{\delta_1}{2}\mathbf{I}) \prod_{n=1}^N \mathrm{N}(\mathbf{x}_1^n; \mathbf{u}_1, \frac{\delta_1}{2}\mathbf{I})$.

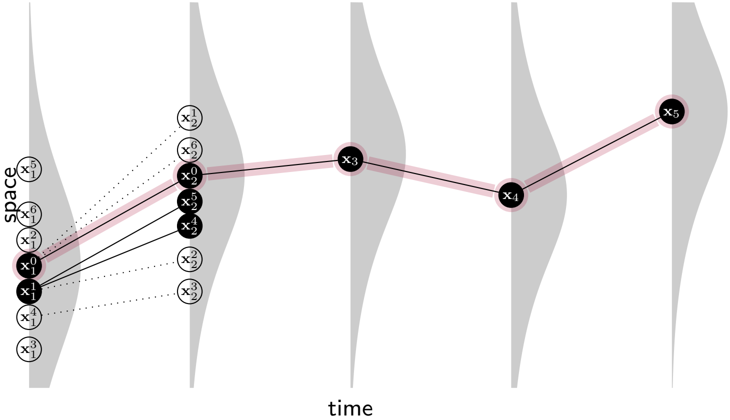
Particle-RWM $(D \to \infty)$



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

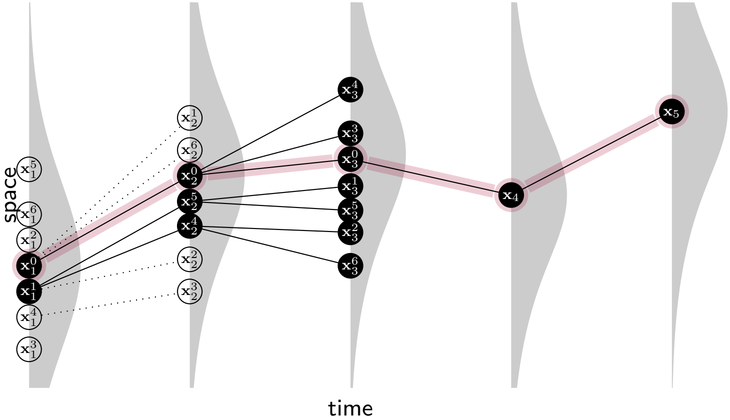
Particle-RWM $(D \to \infty)$



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

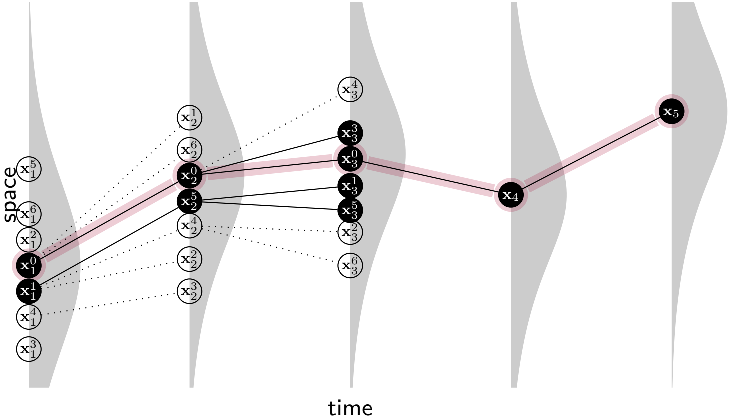
Particle-RWM ($D \to \infty$)



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

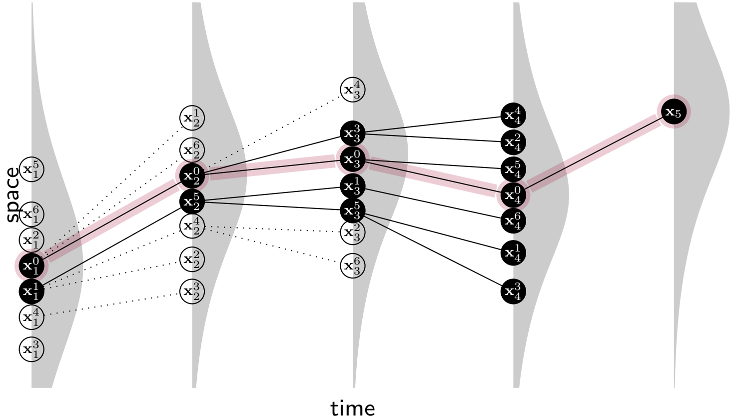
Particle-RWM $(D \to \infty)$



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

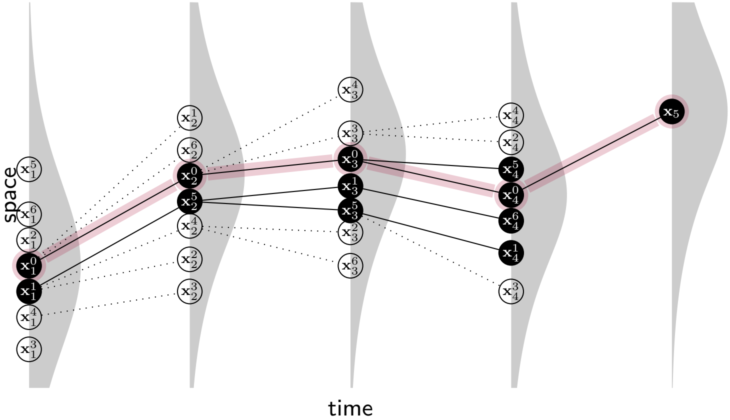
Particle-RWM ($D \to \infty$)



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^{N} W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

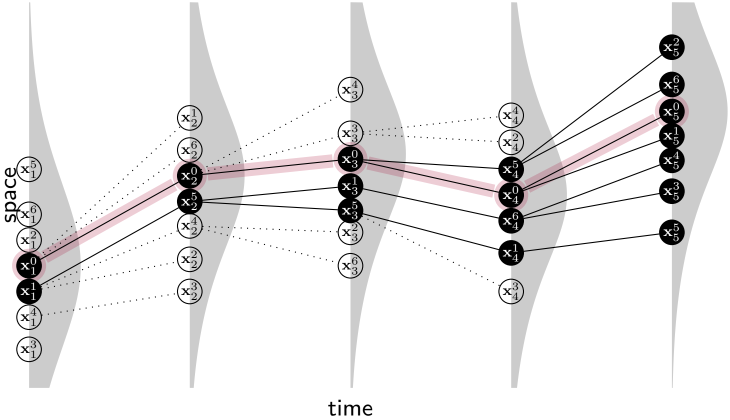
Particle-RWM $(D \to \infty)$



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

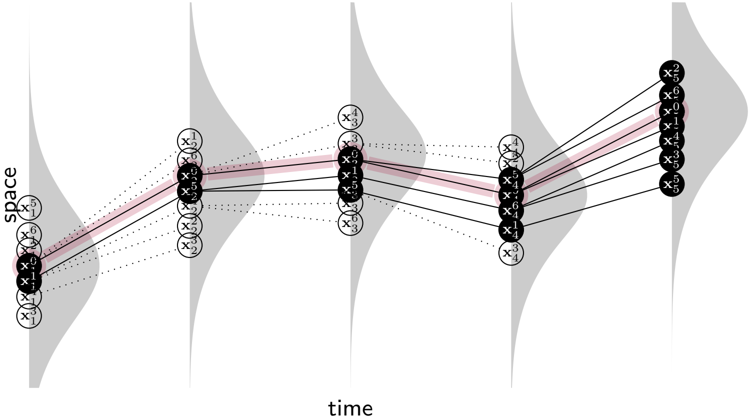
Particle-RWM ($D \to \infty$)



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

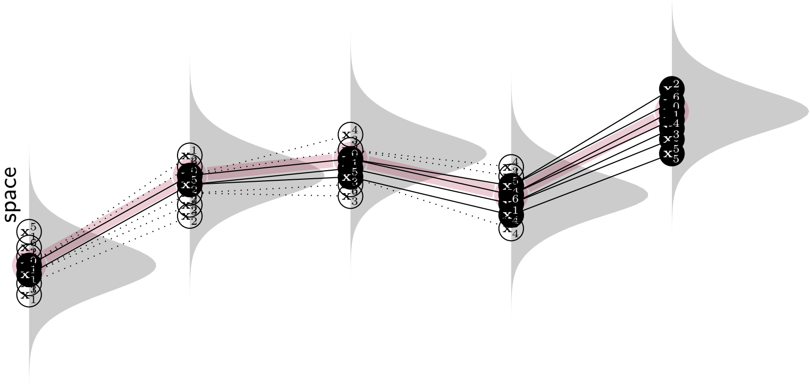
Particle-RWM ($D \to \infty$)



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

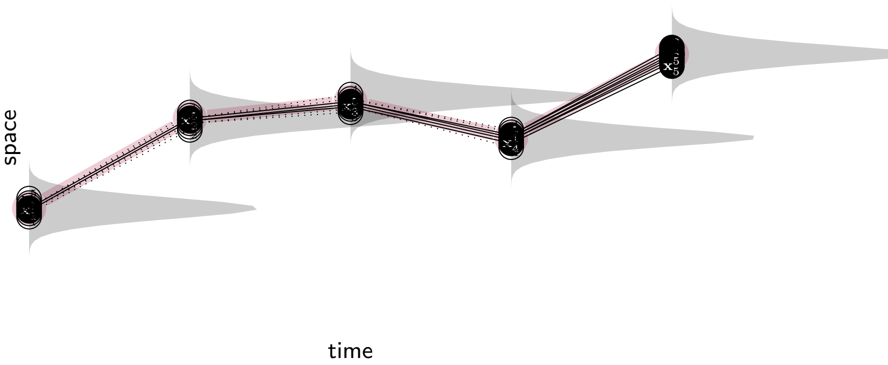
Particle-RWM ($D \to \infty$)



Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

Particle-RWM ($D \to \infty$)

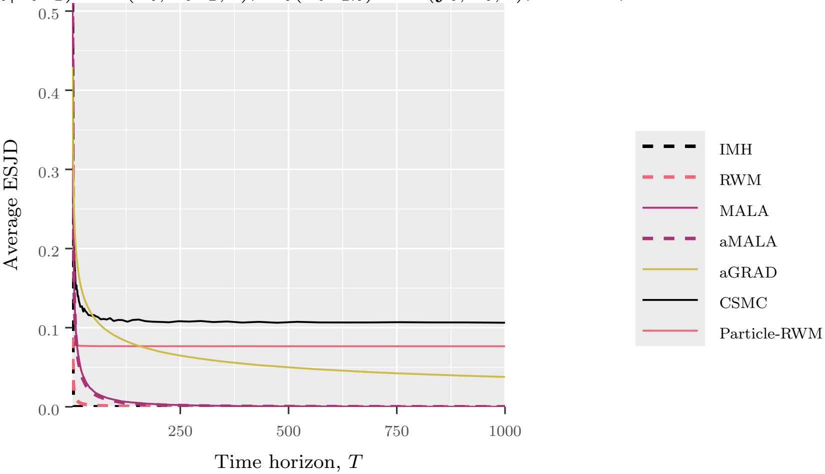


Given reference path $\mathbf{x}_{1:T}$ (current state of MCMC chain):

- Set $\mathbf{x}_t^0 := \mathbf{x}_t$, $a_{t-1}^0 := 0$.
- Sample $(\mathbf{u}_t, \mathbf{x}_t^{1:N}, a_{t-1}^{1:N}) \sim \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t^0, \frac{\delta_t}{2}\mathbf{I}) \prod_{n=1}^N W_{t-1}^{a_{t-1}^n} \mathrm{N}(\mathbf{x}_t^n; \mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$,
  - where $W_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n)$.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

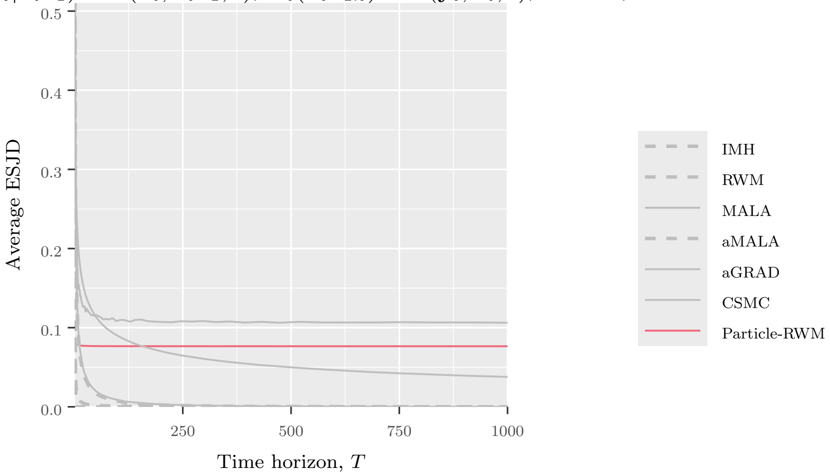


(Average ESJD) $= \frac{1}{TD} \sum_{t=1}^{T} \sum_{d=1}^{D} (x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in T $\rightsquigarrow$ horizontal line;
- must **increase** in T $\rightsquigarrow$ decreasing line.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

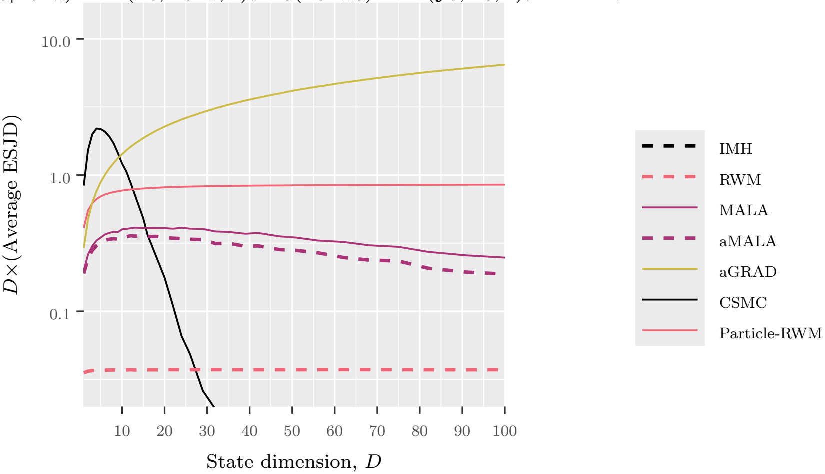


(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in $T \rightsquigarrow$ horizontal line;
- must **increase** in $T \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$

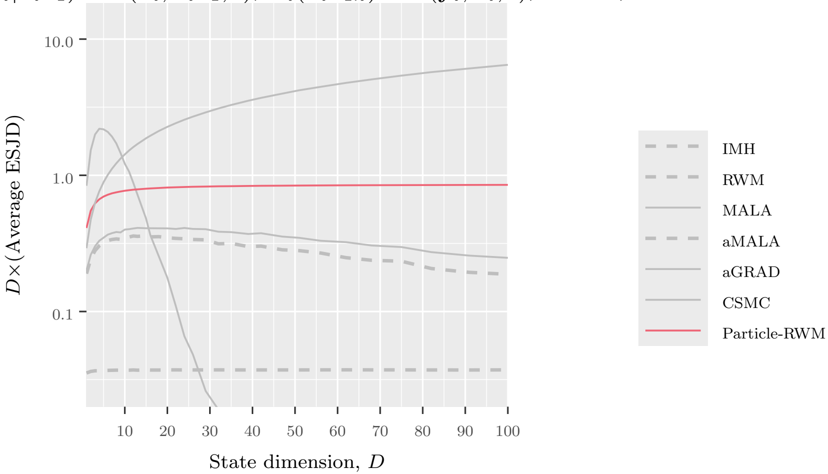


$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in D $\rightsquigarrow$ horizontal line;
- can grow **sublinearly** in D $\rightsquigarrow$ increasing line;
- must grow **superlinearly** in D $\rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Extended state-space view of Particle-RWM

Corenflos and Särkkä (2023)

Extended target distribution (admits $\pi_T(\mathbf{x}_{1:T})$ as a marginal!):

$$\begin{aligned}\pi'_T(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}) &:= \pi_T(\mathbf{x}_{1:T}) \prod_{t=1}^{T} \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t, \tfrac{\delta_t}{2}\mathbf{I}) \\ &\propto \prod_{t=1}^{T} \underbrace{\mathrm{N}(\mathbf{x}_t; \mathbf{u}_t, \tfrac{\delta_t}{2}\mathbf{I})}_{=: M'_t(\mathbf{x}_t \mid \mathbf{x}_{t-1}; \mathbf{u}_t)} \underbrace{M_t(\mathbf{x}_t \mid \mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t})}_{=: G'_t(\mathbf{x}_{t-1:t})},\end{aligned}$$

Equivalent formulation of Particle-RWM:

Extended state-space view of Particle-RWM

Corenflos and Särkkä (2023)

Extended target distribution (admits $\pi_T(\mathbf{x}_{1:T})$ as a marginal!):

$$\begin{aligned}\pi'_T(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}) &:= \pi_T(\mathbf{x}_{1:T}) \prod_{t=1}^{T} \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t, \tfrac{\delta_t}{2}\mathbf{I}) \\ &\propto \prod_{t=1}^{T} \underbrace{\mathrm{N}(\mathbf{x}_t; \mathbf{u}_t, \tfrac{\delta_t}{2}\mathbf{I})}_{=: M'_t(\mathbf{x}_t|\mathbf{x}_{t-1};\mathbf{u}_t)} \underbrace{M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t})}_{=: G'_t(\mathbf{x}_{t-1:t})},\end{aligned}$$

Equivalent formulation of Particle-RWM:

1. sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t, \frac{\delta_t}{2}\mathbf{I})$, for $t = 1, \ldots, T$;

Extended state-space view of Particle-RWM

Corenflos and Särkkä (2023)

Extended target distribution (admits $\pi_T(\mathbf{x}_{1:T})$ as a marginal!):

$$\begin{aligned}\pi'_T(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}) &:= \pi_T(\mathbf{x}_{1:T}) \prod_{t=1}^{T} \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t, \tfrac{\delta_t}{2}\mathbf{I}) \\ &\propto \prod_{t=1}^{T} \underbrace{\mathrm{N}(\mathbf{x}_t; \mathbf{u}_t, \tfrac{\delta_t}{2}\mathbf{I})}_{=: M'_t(\mathbf{x}_t | \mathbf{x}_{t-1}; \mathbf{u}_t)} \underbrace{M_t(\mathbf{x}_t | \mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t})}_{=: G'_t(\mathbf{x}_{t-1:t})},\end{aligned}$$

Equivalent formulation of Particle-RWM:

1. sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t, \frac{\delta_t}{2}\mathbf{I})$, for $t = 1, \ldots, T$;
2. run standard CSMC algorithm but replace

Extended state-space view of Particle-RWM

Corenflos and Särkkä (2023)

Extended target distribution (admits $\pi_T(\mathbf{x}_{1:T})$ as a marginal!):

$$\begin{aligned}\pi'_T(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}) &:= \pi_T(\mathbf{x}_{1:T}) \prod_{t=1}^{T} \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t, \tfrac{\delta_t}{2}\mathbf{I}) \\ &\propto \prod_{t=1}^{T} \underbrace{\mathrm{N}(\mathbf{x}_t; \mathbf{u}_t, \tfrac{\delta_t}{2}\mathbf{I})}_{=:M'_t(\mathbf{x}_t|\mathbf{x}_{t-1};\mathbf{u}_t)} \underbrace{M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t})}_{=:G'_t(\mathbf{x}_{t-1:t})},\end{aligned}$$

Equivalent formulation of Particle-RWM:

1. sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t, \frac{\delta_t}{2}\mathbf{I})$, for $t = 1, \ldots, T$;
2. run standard CSMC algorithm but replace
   - $M_t(\mathbf{x}_t|\mathbf{x}_{t-1})$ by $M'_t(\mathbf{x}_t|\mathbf{x}_{t-1}; \mathbf{u}_t)$;

Extended state-space view of Particle-RWM

Corenflos and Särkkä (2023)

Extended target distribution (admits $\pi_T(\mathbf{x}_{1:T})$ as a marginal!):

$$\begin{aligned}\pi'_T(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}) &:= \pi_T(\mathbf{x}_{1:T}) \prod_{t=1}^{T} \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t, \tfrac{\delta_t}{2}\mathbf{I}) \\ &\propto \prod_{t=1}^{T} \underbrace{\mathrm{N}(\mathbf{x}_t; \mathbf{u}_t, \tfrac{\delta_t}{2}\mathbf{I})}_{=: M'_t(\mathbf{x}_t|\mathbf{x}_{t-1};\mathbf{u}_t)} \underbrace{M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) G_t(\mathbf{x}_{t-1:t})}_{=: G'_t(\mathbf{x}_{t-1:t})},\end{aligned}$$

Equivalent formulation of Particle-RWM:

1. sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t, \frac{\delta_t}{2}\mathbf{I})$, for $t = 1, \ldots, T$;
2. run standard CSMC algorithm but replace
   - $M_t(\mathbf{x}_t|\mathbf{x}_{t-1})$ by $M'_t(\mathbf{x}_t|\mathbf{x}_{t-1}; \mathbf{u}_t)$;
   - $G_t(\mathbf{x}_{t-1:t})$ by $G'_t(\mathbf{x}_{t-1:t})$.

Summary of the Particle-RWM algorithm

- **Summary:** Particle-RWM exploits ‘decorrelation-over-time’ property of the state-space model and also uses ‘local’ moves.

Summary of the Particle-RWM algorithm

- **Summary:** Particle-RWM exploits 'decorrelation-over-time' property of the state-space model and also uses 'local' moves.
$\rightsquigarrow$ favourable scaling in T & dimensional stability in D.

Summary of the Particle-RWM algorithm

- **Summary:** Particle-RWM exploits 'decorrelation-over-time' property of the state-space model and also uses 'local' moves. $\rightsquigarrow$ favourable scaling in T & dimensional stability in D.
- **Problem:** Particle-RWM does not utilise

Summary of the Particle-RWM algorithm

- **Summary:** Particle-RWM exploits 'decorrelation-over-time' property of the state-space model and also uses 'local' moves.
  $\rightsquigarrow$ favourable scaling in T & dimensional stability in D.
- **Problem:** Particle-RWM does not utilise
  - gradient information (e.g., as in MALA);

Summary of the Particle-RWM algorithm

- **Summary:** Particle-RWM exploits 'decorrelation-over-time' property of the state-space model and also uses 'local' moves.
  $\rightsquigarrow$ favourable scaling in T & dimensional stability in D.
- **Problem:** Particle-RWM does not utilise
  - gradient information (e.g., as in MALA);
  - Gaussian prior information (e.g., as in Crank–Nicholson type methods and mGRAD/aGRAD).

Talk outline

1. State-space models/Feynman–Kac representation
2. Existing methods
3. Particle extensions of MALA and aMALA
4. Particle extensions of mGRAD and aGRAD
5. Numerical illustration
6. Summary

Talk outline

3. Particle extensions of MALA and aMALA
3.1 Exploiting filter gradients (gradients w.r.t. $\log \pi_t$)
3.2 Exploiting smoothing gradients (gradients w.r.t. $\log \pi_T$)

Particle-MALA and Particle-aMALA

Algorithm 4 (Particle-MALA). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** set $\bar{\mathbf{x}}_t := \frac{1}{N+1}\sum_{n=0}^{N} \mathbf{x}_t^n$ and, for $n \in [N]_0$,

$$w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n) F_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n, \bar{\mathbf{x}}_t).$$

Particle-MALA and Particle-aMALA

Algorithm 4 (Particle-MALA). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** set $\bar{\mathbf{x}}_t := \frac{1}{N+1}\sum_{n=0}^{N} \mathbf{x}_t^n$ and, for $n \in [N]_0$,

$$w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n) F_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n, \bar{\mathbf{x}}_t).$$

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \delta_t \mathbf{I}).$$

Particle-MALA and Particle-aMALA

Algorithm 4 (Particle-MALA). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** set $\bar{\mathbf{x}}_t \coloneqq \frac{1}{N+1}\sum_{n=0}^{N} \mathbf{x}_t^n$ and, for $n \in [N]_0$,

$$w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n) F_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n, \bar{\mathbf{x}}_t).$$

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \delta_t \mathbf{I}).$$

- Reduces to MALA if $N = T = 1$.

Particle-MALA and Particle-aMALA

Algorithm 4 (Particle-MALA). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** set $\bar{\mathbf{x}}_t := \frac{1}{N+1}\sum_{n=0}^{N} \mathbf{x}_t^n$ and, for $n \in [N]_0$,

$$w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n) F_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n, \bar{\mathbf{x}}_t).$$

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \delta_t \mathbf{I}).$$

- Reduces to MALA if $N = T = 1$.
- Not integrating out the auxiliary variable $\mathbf{u}_t$ in the weights (and in the backward kernel) gives the Particle-aMALA.

Particle-MALA and Particle-aMALA

Algorithm 4 (Particle-MALA). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** set $\bar{\mathbf{x}}_t := \frac{1}{N+1}\sum_{n=0}^{N} \mathbf{x}_t^n$ and, for $n \in [N]_0$,

$$w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n) F_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n, \bar{\mathbf{x}}_t).$$

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \delta_t \mathbf{I}).$$

- Reduces to MALA if $N = T = 1$.
- Not integrating out the auxiliary variable $\mathbf{u}_t$ in the weights (and in the backward kernel) gives the Particle-aMALA.
  - 'random-weight' version of Particle-MALA;

Particle-MALA and Particle-aMALA

Algorithm 4 (Particle-MALA). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** set $\bar{\mathbf{x}}_t := \frac{1}{N+1}\sum_{n=0}^{N} \mathbf{x}_t^n$ and, for $n \in [N]_0$,

$$w_t^n \propto Q_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n) F_t(\mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n, \bar{\mathbf{x}}_t).$$

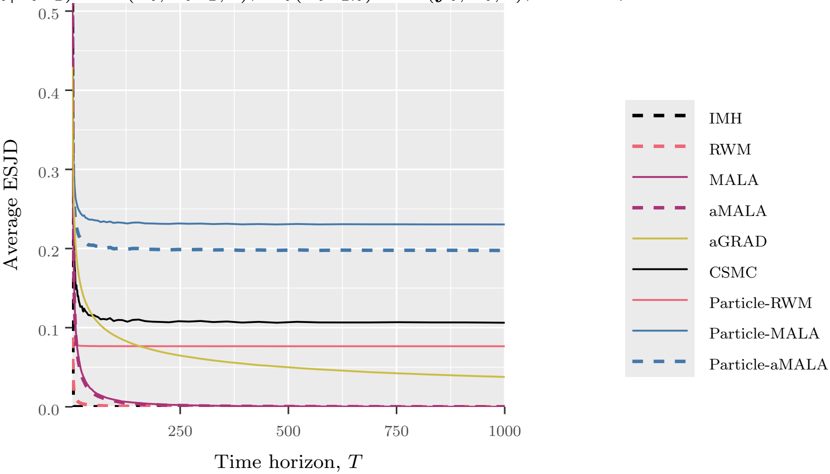
- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \delta_t \mathbf{I}).$$

- Reduces to MALA if $N = T = 1$.
- Not integrating out the auxiliary variable $\mathbf{u}_t$ in the weights (and in the backward kernel) gives the Particle-aMALA.
  - ‘random-weight’ version of Particle-MALA;
  - reduces to aMALA if $N = T = 1$.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

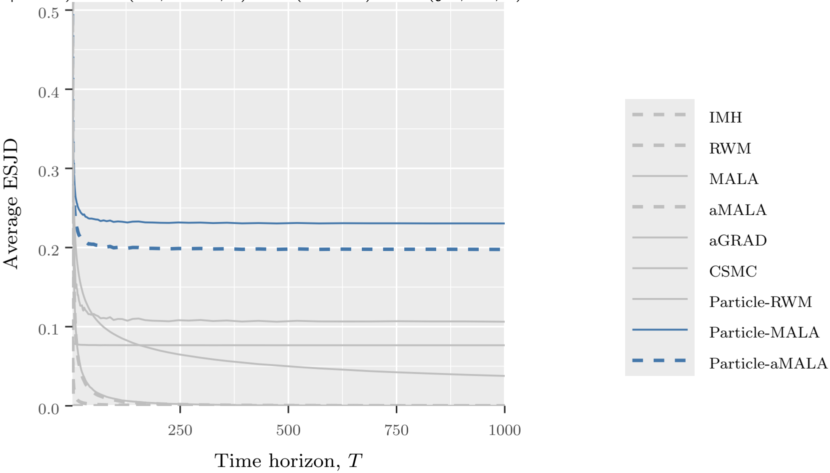


(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in T $\rightsquigarrow$ horizontal line;
- must **increase** in T $\rightsquigarrow$ decreasing line.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

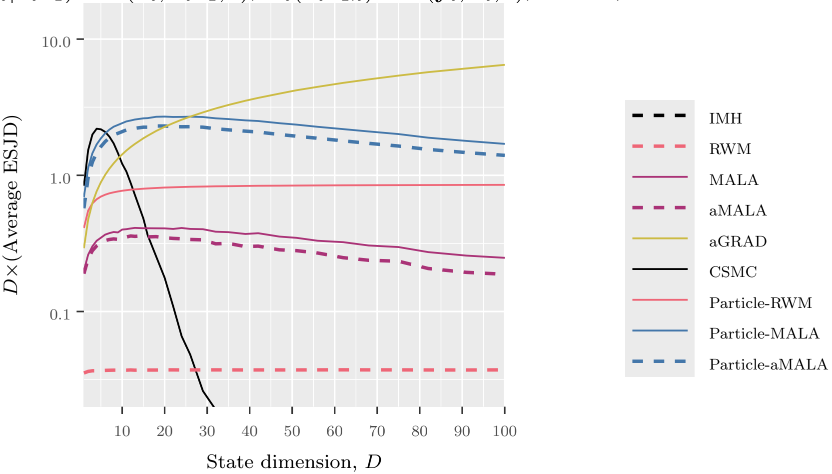


$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in T $\rightsquigarrow$ horizontal line;
- must **increase** in T $\rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$

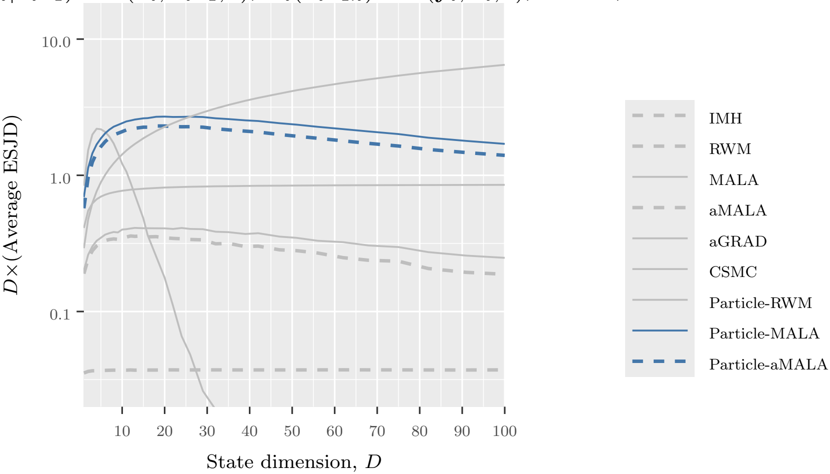


(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Talk outline

3. Particle extensions of MALA and aMALA

3.1 Exploiting filter gradients (gradients w.r.t. $\log \pi_t$)

3.2 Exploiting smoothing gradients (gradients w.r.t. $\log \pi_T$)

Particle-aMALA+

Algorithm 5 (Particle-aMALA+). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_T(\mathbf{x}_{1:T}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** for $n \in [N]_0$, set $w_t^n \propto G'_t(\mathbf{x}_{t-2}^{a_{t-2}^{a_{t-1}^n}}, \mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n; \mathbf{u}_{t-1:t})$,

3. **[backward sampling]** (*omitted*)

Particle-aMALA+

Algorithm 5 (Particle-aMALA+). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_T(\mathbf{x}_{1:T}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** for $n \in [N]_0$, set $w_t^n \propto G_t'(\mathbf{x}_{t-2}^{a_{t-2}^{a_{t-1}^n}}, \mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n; \mathbf{u}_{t-1:t})$,

3. **[backward sampling]** (*omitted*)

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_T(\mathbf{x}_{1:T}), \delta_t \mathbf{I}).$$

Particle-aMALA+

Algorithm 5 (Particle-aMALA+). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_T(\mathbf{x}_{1:T}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** for $n \in [N]_0$, set $w_t^n \propto G'_t(\mathbf{x}_{t-2}^{a_{t-2}^{a_{t-1}^n}}, \mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n; \mathbf{u}_{t-1:t})$,

3. **[backward sampling]** (*omitted*)

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_T(\mathbf{x}_{1:T}), \delta_t\mathbf{I}).$$

- Using gradients w.r.t. $\log \pi_T(\mathbf{x}_{1:T})$ (rather than $\log \pi_t(\mathbf{x}_{1:t})$) comes at cost of having only 2nd-order Markovianity.

Particle-aMALA+

Algorithm 5 (Particle-aMALA+). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_T(\mathbf{x}_{1:T}), \frac{\delta_t}{2}\mathbf{I})$, and $\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{u}_t, \frac{\delta_t}{2}\mathbf{I})$, for $n \in [N]$,

1d. **[weighting]** for $n \in [N]_0$, set $w_t^n \propto G'_t(\mathbf{x}_{t-2}^{a_{t-2}^{a_{t-1}^n}}, \mathbf{x}_{t-1}^{a_{t-1}^n}, \mathbf{x}_t^n; \mathbf{u}_{t-1:t})$,

3. **[backward sampling]** (*omitted*)

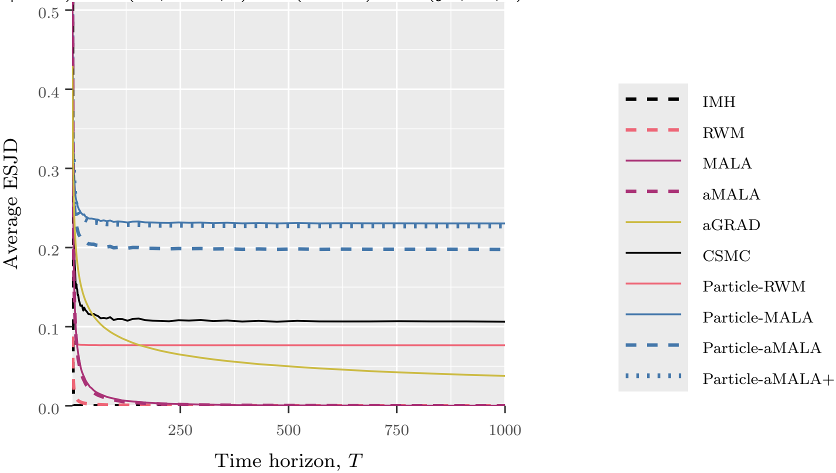
- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_T(\mathbf{x}_{1:T}), \delta_t\mathbf{I}).$$

- Using gradients w.r.t. $\log \pi_T(\mathbf{x}_{1:T})$ (rather than $\log \pi_t(\mathbf{x}_{1:t})$) comes at cost of having only 2nd-order Markovianity.
- Again reduces to aMALA if $N = T = 1$.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

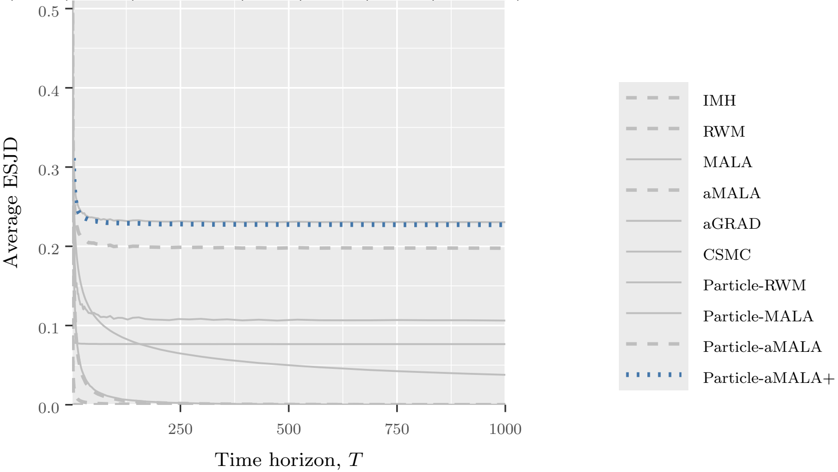


$(\text{Average ESJD}) = \frac{1}{TD} \sum_{t=1}^{T} \sum_{d=1}^{D} (x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in $T \rightsquigarrow$ horizontal line;
- must **increase** in $T \rightsquigarrow$ decreasing line.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

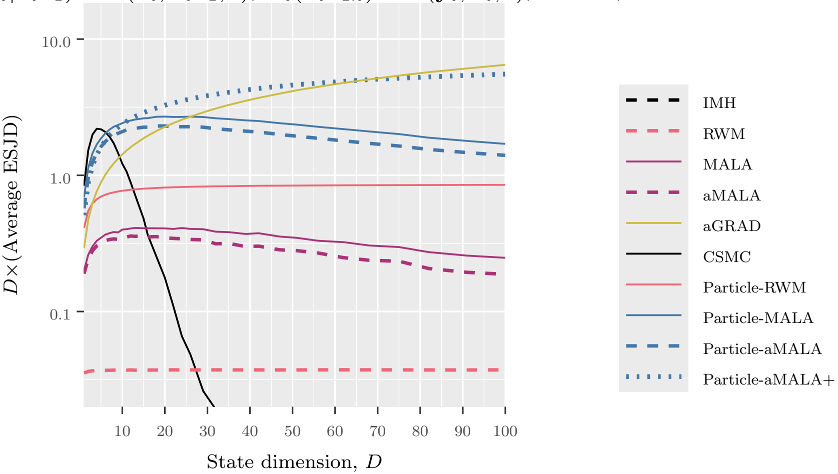


(Average ESJD) $= \frac{1}{TD} \sum_{t=1}^{T} \sum_{d=1}^{D} (x_{t,d}^{\mathrm{new}} - x_{t,d}^{\mathrm{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in $T \rightsquigarrow$ horizontal line;
- must **increase** in $T \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$

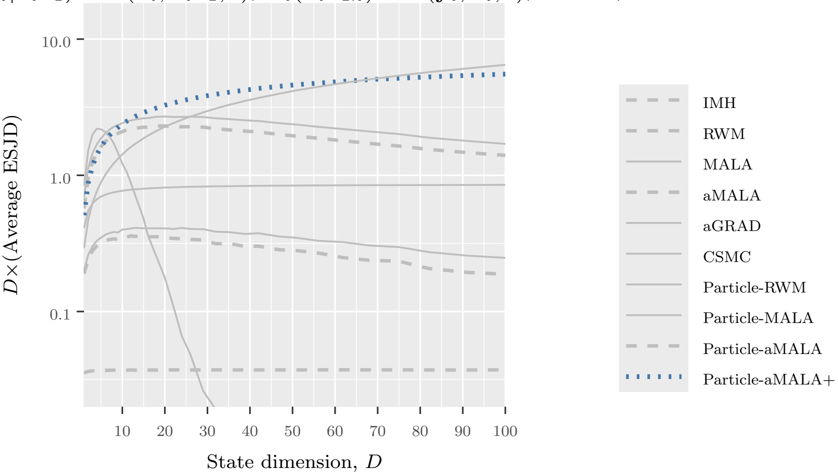


$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Talk outline

1. State-space models/Feynman–Kac representation
2. Existing methods
3. Particle extensions of MALA and aMALA
4. Particle extensions of mGRAD and aGRAD
5. Numerical illustration
6. Summary

Talk outline

Conditionally Gaussian prior dynamics

- For the moment, assume that

$$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t).$$

Particle-mGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t)$

Algorithm 6 (Particle-mGRAD). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t}), \frac{\delta_t}{2}\mathbf{I})$ and $\mathbf{x}_t^n \sim M_t'(\,\cdot\,|\mathbf{x}_{t-1}^{a_{t-1}^n}; \mathbf{u}_t)$, for $n \in [N]$,

1d. **[weighting]** (*omitted*)

Particle-mGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t)$

Algorithm 6 (Particle-mGRAD). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t}), \frac{\delta_t}{2}\mathbf{I})$ and $\mathbf{x}_t^n \sim M_t'(\,\cdot\,|\mathbf{x}_{t-1}^{a_{t-1}^n}; \mathbf{u}_t)$, for $n \in [N]$,

1d. **[weighting]** (*omitted*)

- Here,

$$M_t'(\mathbf{x}_t|\mathbf{x}_{t-1}; \mathbf{u}_t) \propto \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t)\,\mathrm{N}(\mathbf{u}_t; \mathbf{x}_t, \tfrac{\delta_t}{2}\mathbf{I}),$$

  is the **'fully-adapted auxiliary-particle filter'** proposal for the pseudo observation $\mathbf{u}_t$:

Particle-mGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t)$

Algorithm 6 (Particle-mGRAD). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t}), \frac{\delta_t}{2}\mathbf{I})$ and $\mathbf{x}_t^n \sim M_t'(\,\cdot\,|\mathbf{x}_{t-1}^{a_{t-1}^n}; \mathbf{u}_t)$, for $n \in [N]$,

1d. **[weighting]** (*omitted*)

- Here,

$$M_t'(\mathbf{x}_t|\mathbf{x}_{t-1}; \mathbf{u}_t) \propto \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t)\, \mathrm{N}(\mathbf{u}_t; \mathbf{x}_t, \tfrac{\delta_t}{2}\mathbf{I}),$$

  is the **'fully-adapted auxiliary-particle filter'** proposal for the pseudo observation $\mathbf{u}_t$:

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}((\mathbf{I} - \mathbf{A}_t)\mathbf{m}_t(\mathbf{x}_{t-1}^{a_{t-1}^n}) + \mathbf{A}_t[\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t})], \mathbf{B}_t),$$

  where $\mathbf{B}_t := \frac{\delta_t}{2}\mathbf{A}_t^2 + \mathbf{A}_t$ and $\mathbf{A}_t = (\mathbf{C}_t + \frac{\delta_t}{2}\mathbf{I})^{-1}\mathbf{C}_t$.

Particle-mGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t)$

Algorithm 6 (Particle-mGRAD). Modify CSMC as follows:

1c. **[sampling]** sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t}), \frac{\delta_t}{2}\mathbf{I})$ and $\mathbf{x}_t^n \sim M_t'(\,\cdot\,|\mathbf{x}_{t-1}^{a_{t-1}^n}; \mathbf{u}_t)$, for $n \in [N]$,

1d. **[weighting]** (*omitted*)

- Here,

$$M_t'(\mathbf{x}_t|\mathbf{x}_{t-1}; \mathbf{u}_t) \propto \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t)\,\mathrm{N}(\mathbf{u}_t; \mathbf{x}_t, \tfrac{\delta_t}{2}\mathbf{I}),$$

  is the **'fully-adapted auxiliary-particle filter'** proposal for the pseudo observation $\mathbf{u}_t$:

- Step 1c *marginally* proposes (for $n \neq 0$):

$$\mathbf{x}_t^n \sim \mathrm{N}((\mathbf{I} - \mathbf{A}_t)\mathbf{m}_t(\mathbf{x}_{t-1}^{a_{t-1}^n}) + \mathbf{A}_t[\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t})], \mathbf{B}_t),$$

  where $\mathbf{B}_t := \frac{\delta_t}{2}\mathbf{A}_t^2 + \mathbf{A}_t$ and $\mathbf{A}_t = (\mathbf{C}_t + \frac{\delta_t}{2}\mathbf{I})^{-1}\mathbf{C}_t$.

- Reduces to mGRAD if $N = T = 1$.

Particle-aGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t(\mathbf{x}_{t-1}))$

- Not integrating out the auxiliary variable $\mathbf{u}_t$ in the weights/backward kernel of Particle-mGRAD gives the Particle-aGRAD algorithm:

Particle-aGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t(\mathbf{x}_{t-1}))$

- Not integrating out the auxiliary variable $\mathbf{u}_t$ in the weights/backward kernel of Particle-mGRAD gives the Particle-aGRAD algorithm:
  - 'random-weight' version of Particle-mGRAD;

Particle-aGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t(\mathbf{x}_{t-1}))$

- Not integrating out the auxiliary variable $\mathbf{u}_t$ in the weights/backward kernel of Particle-mGRAD gives the Particle-aGRAD algorithm:
  - 'random-weight' version of Particle-mGRAD;
  - implementable even if $\mathbf{C}_t = \mathbf{C}_t(\mathbf{x}_{t-1})$ depends on $\mathbf{x}_{t-1}$;

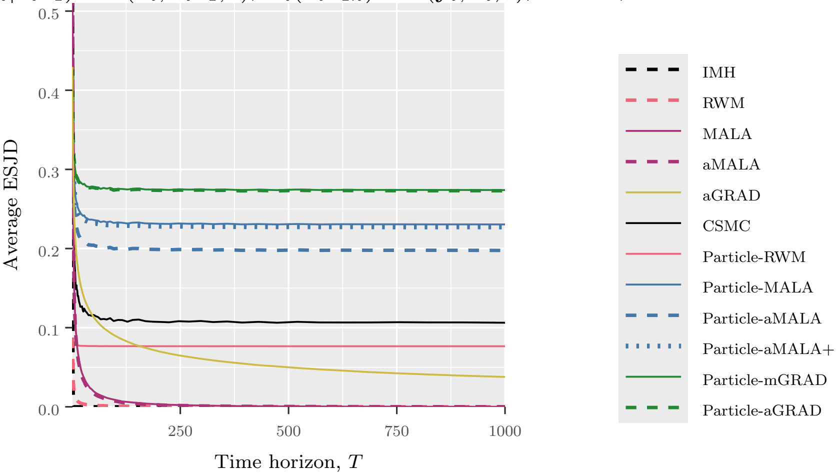
Particle-aGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t(\mathbf{x}_{t-1}))$

- Not integrating out the auxiliary variable $\mathbf{u}_t$ in the weights/backward kernel of Particle-mGRAD gives the Particle-aGRAD algorithm:
  - 'random-weight' version of Particle-mGRAD;
  - implementable even if $\mathbf{C}_t = \mathbf{C}_t(\mathbf{x}_{t-1})$ depends on $\mathbf{x}_{t-1}$;
  - reduces to aGRAD if $N = T = 1$.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

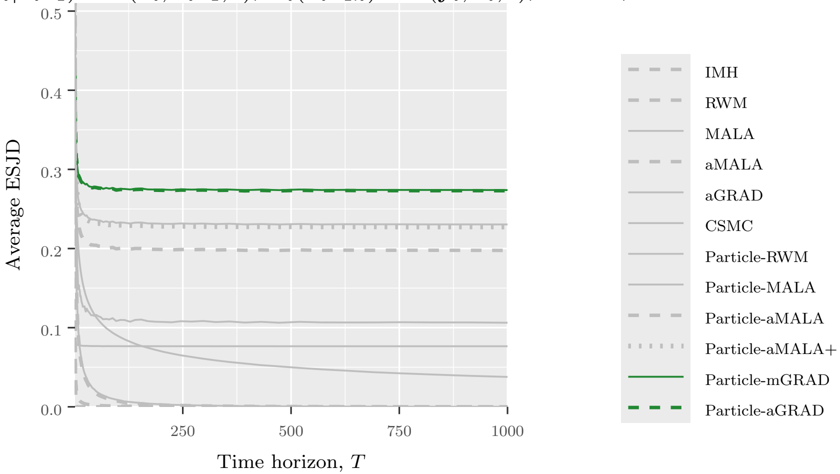


(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in T $\rightsquigarrow$ horizontal line;
- must **increase** in T $\rightsquigarrow$ decreasing line.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

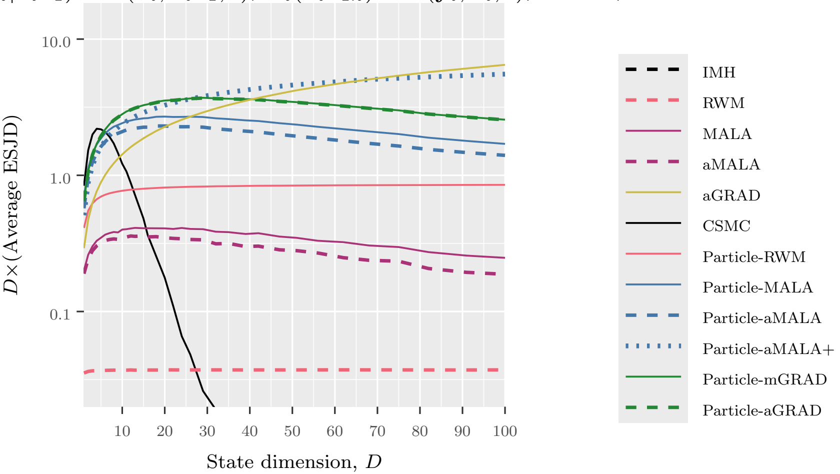


$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in T $\rightsquigarrow$ horizontal line;
- must **increase** in T $\rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$

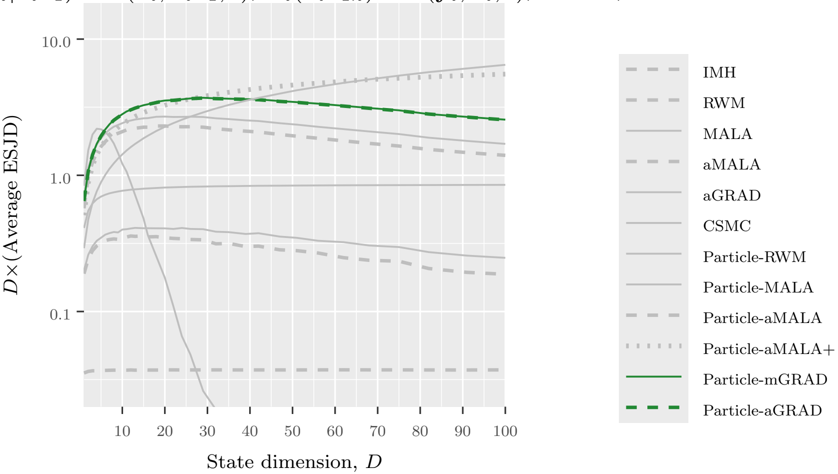


(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\mathrm{new}} - x_{t,d}^{\mathrm{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Talk outline

Gaussian prior dynamics

- Now assume $\mathbf{m}_t(\mathbf{x}_{t-1}) = \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{b}_t$, i.e.:

$$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{b}_t, \mathbf{C}_t).$$

Twisted Particle-aGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{b}_t, \mathbf{C}_t)$

Algorithm 7 (Twisted Particle-aGRAD). For $t \in [T]$, sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t}), \frac{\delta_t}{2}\mathbf{I})$. Then, run the CSMC algorithm with the following modifications.

1c. **[sampling]** $\mathbf{x}_t^n \sim M_t'(\,\cdot\,|\mathbf{x}_{t-1}^{a_{t-1}^n}; \mathbf{u}_{t:T})$, for $n \in [N]$,

1d. **[weighting]** (*omitted*)

3. **[backward sampling]** (*omitted*)

Twisted Particle-aGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{b}_t, \mathbf{C}_t)$

Algorithm 7 (Twisted Particle-aGRAD). For $t \in [T]$, sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t}), \frac{\delta_t}{2}\mathbf{I})$. Then, run the CSMC algorithm with the following modifications.

1c. **[sampling]** $\mathbf{x}_t^n \sim M_t'(\,\cdot\,|\mathbf{x}_{t-1}^{a_{t-1}^n}; \mathbf{u}_{t:T})$, for $n \in [N]$,

1d. **[weighting]** (*omitted*)

3. **[backward sampling]** (*omitted*)

- Here,

$$M_t'(\mathbf{x}_t|\mathbf{x}_{t-1}; \mathbf{u}_{t:T})$$

$$\propto \int_{\mathcal{X}^{T-t}} \left[\prod_{s=t}^{T} \mathrm{N}(\mathbf{x}_s; \mathbf{F}_s\mathbf{x}_{s-1} + \mathbf{b}_s, \mathbf{C}_s)\, \mathrm{N}(\mathbf{u}_s; \mathbf{x}_s, \tfrac{\delta_s}{2}\mathbf{I}) \right] \mathrm{d}\mathbf{x}_{t+1:T},$$

is the **'fully-twisted particle filter'** proposal for the pseudo observations $\mathbf{u}_{t:T}$:

Twisted Particle-aGRAD

Assuming $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{b}_t, \mathbf{C}_t)$

Algorithm 7 (Twisted Particle-aGRAD). For $t \in [T]$, sample $\mathbf{u}_t \sim \mathrm{N}(\mathbf{x}_t + \frac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t}), \frac{\delta_t}{2}\mathbf{I})$. Then, run the CSMC algorithm with the following modifications.

1c. **[sampling]** $\mathbf{x}_t^n \sim M'_t(\,\cdot\,|\mathbf{x}_{t-1}^{a_{t-1}^n}; \mathbf{u}_{t:T})$, for $n \in [N]$,

1d. **[weighting]** (*omitted*)

3. **[backward sampling]** (*omitted*)

- Here,

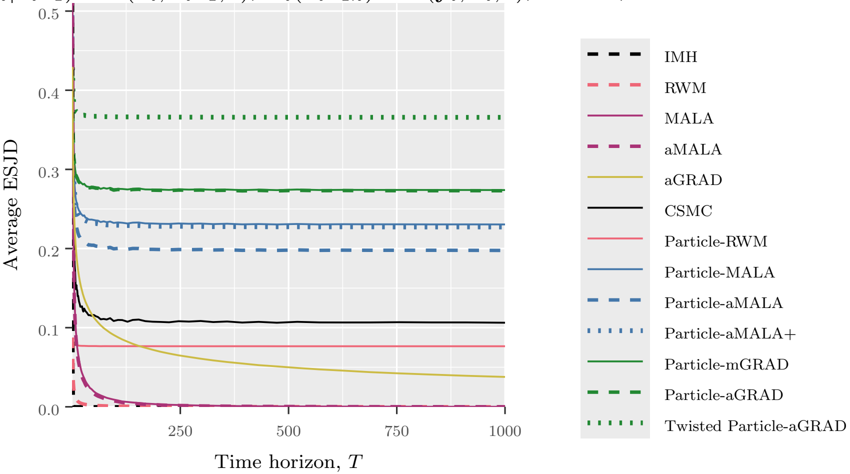
$$M'_t(\mathbf{x}_t|\mathbf{x}_{t-1}; \mathbf{u}_{t:T})$$

$$\propto \int_{\mathcal{X}^{T-t}} \left[\prod_{s=t}^{T} \mathrm{N}(\mathbf{x}_s; \mathbf{F}_s\mathbf{x}_{s-1} + \mathbf{b}_s, \mathbf{C}_s)\,\mathrm{N}(\mathbf{u}_s; \mathbf{x}_s, \tfrac{\delta_s}{2}\mathbf{I})\right] \mathrm{d}\mathbf{x}_{t+1:T},$$

  is the **'fully-twisted particle filter'** proposal for the pseudo observations $\mathbf{u}_{t:T}$:
- Reduces to aGRAD if $N = T = 1$.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

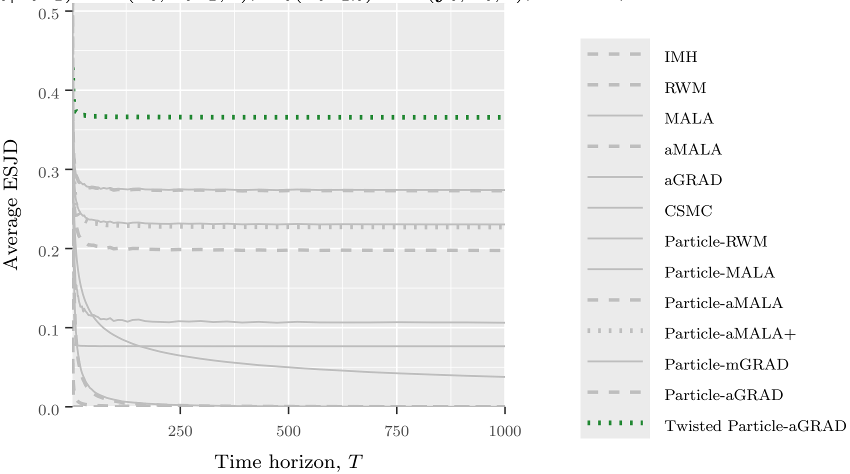


$(\text{Average ESJD}) = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in $T \rightsquigarrow$ horizontal line;
- must **increase** in $T \rightsquigarrow$ decreasing line.

Scaling with T

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $D = 10$, $N = 31$

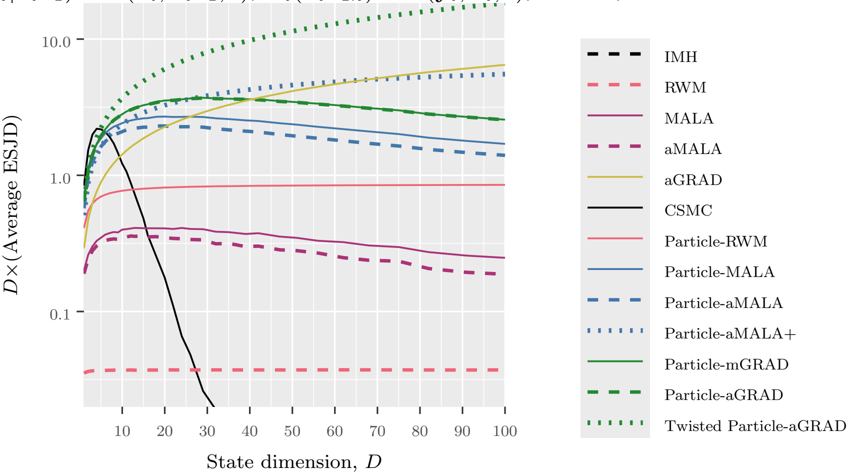


(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\mathrm{new}} - x_{t,d}^{\mathrm{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- can be **constant** in $T \rightsquigarrow$ horizontal line;
- must **increase** in $T \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$

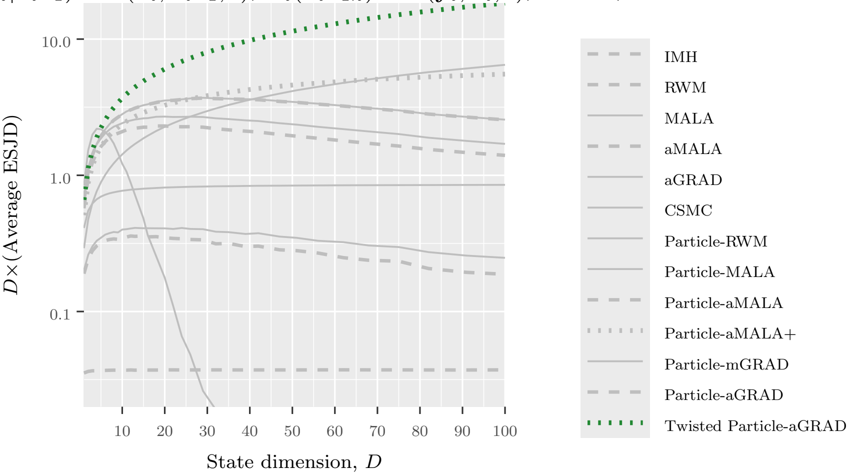


(Average ESJD) $= \frac{1}{TD} \sum_{t=1}^{T} \sum_{d=1}^{D} (x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Scaling with D

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = 25$, $N = 31$



(Average ESJD) $= \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}(x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2 \implies$ Informally, to stably approximate marginals, the number of iterations

- must grow **linearly** in $D \rightsquigarrow$ horizontal line;
- can grow **sublinearly** in $D \rightsquigarrow$ increasing line;
- must grow **superlinearly** in $D \rightsquigarrow$ decreasing line.

Talk outline

Intuition

- Assume $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{C}_t)$.

Intuition

- Assume $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{C}_t)$.
- **Recall:** Particle-mGRAD/Particle-aGRAD *marginally* propose:

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{a}_t, \mathbf{B}_t), \quad \text{for } n \neq 0,$$

where (with $\mathbf{A}_t = (\mathbf{C}_t + \frac{\delta_t}{2}\mathbf{I})^{-1}\mathbf{C}_t$),

$$\mathbf{a}_t \coloneqq (\mathbf{I} - \mathbf{A}_t)\mathbf{m}_t + \mathbf{A}_t[\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t})],$$
$$\mathbf{B}_t \coloneqq \tfrac{\delta_t}{2}\mathbf{A}_t^2 + \mathbf{A}_t,$$

Intuition

- Assume $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{C}_t)$.
- **Recall:** Particle-mGRAD/Particle-aGRAD *marginally* propose:

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{a}_t, \mathbf{B}_t), \quad \text{for } n \neq 0,$$

where (with $\mathbf{A}_t = (\mathbf{C}_t + \frac{\delta_t}{2}\mathbf{I})^{-1}\mathbf{C}_t$),

$$\mathbf{a}_t := (\mathbf{I} - \mathbf{A}_t)\mathbf{m}_t + \mathbf{A}_t[\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t})],$$
$$\mathbf{B}_t := \tfrac{\delta_t}{2}\mathbf{A}_t^2 + \mathbf{A}_t,$$

- If prior is highly **informative** (all eigenvalues of $\mathbf{C}_t$ small) then $\mathbf{A}_t \approx \mathbf{0}$ and

Intuition

- Assume $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{C}_t)$.
- **Recall:** Particle-mGRAD/Particle-aGRAD *marginally* propose:

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{a}_t, \mathbf{B}_t), \quad \text{for } n \neq 0,$$

where (with $\mathbf{A}_t = (\mathbf{C}_t + \frac{\delta_t}{2}\mathbf{I})^{-1}\mathbf{C}_t$),

$$\mathbf{a}_t := (\mathbf{I} - \mathbf{A}_t)\mathbf{m}_t + \mathbf{A}_t[\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t})],$$
$$\mathbf{B}_t := \tfrac{\delta_t}{2}\mathbf{A}_t^2 + \mathbf{A}_t,$$

- If prior is highly **informative** (all eigenvalues of $\mathbf{C}_t$ small) then $\mathbf{A}_t \approx \mathbf{0}$ and

$$\mathrm{N}(\mathbf{a}_t, \mathbf{B}_t) \approx \overbrace{\mathrm{N}(\mathbf{m}_t, \mathbf{C}_t)}^{\text{CSMC proposal}}.$$

Intuition

- Assume $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{C}_t)$.
- **Recall:** Particle-mGRAD/Particle-aGRAD *marginally* propose:

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{a}_t, \mathbf{B}_t), \quad \text{for } n \neq 0,$$

  where (with $\mathbf{A}_t = (\mathbf{C}_t + \frac{\delta_t}{2}\mathbf{I})^{-1}\mathbf{C}_t$),

$$\mathbf{a}_t := (\mathbf{I} - \mathbf{A}_t)\mathbf{m}_t + \mathbf{A}_t[\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t})],$$
$$\mathbf{B}_t := \tfrac{\delta_t}{2}\mathbf{A}_t^2 + \mathbf{A}_t,$$

- If prior is highly **informative** (all eigenvalues of $\mathbf{C}_t$ small) then $\mathbf{A}_t \approx \mathbf{0}$ and

$$\mathrm{N}(\mathbf{a}_t, \mathbf{B}_t) \approx \overbrace{\mathrm{N}(\mathbf{m}_t, \mathbf{C}_t)}^{\text{CSMC proposal}}.$$

- If prior is highly **uninformative** (all eigenvalues of $\mathbf{C}_t$ large) then $\mathbf{A}_t \approx \mathbf{I}$ and

Intuition

- Assume $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{C}_t)$.
- **Recall:** Particle-mGRAD/Particle-aGRAD *marginally* propose:

$$\mathbf{x}_t^n \sim \mathrm{N}(\mathbf{a}_t, \mathbf{B}_t), \quad \text{for } n \neq 0,$$

where (with $\mathbf{A}_t = (\mathbf{C}_t + \frac{\delta_t}{2}\mathbf{I})^{-1}\mathbf{C}_t$),

$$\begin{aligned}\mathbf{a}_t &:= (\mathbf{I} - \mathbf{A}_t)\mathbf{m}_t + \mathbf{A}_t[\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log G_t(\mathbf{x}_{t-1:t})],\\ \mathbf{B}_t &:= \tfrac{\delta_t}{2}\mathbf{A}_t^2 + \mathbf{A}_t,\end{aligned}$$

- If prior is highly **informative** (all eigenvalues of $\mathbf{C}_t$ small) then $\mathbf{A}_t \approx \mathbf{0}$ and

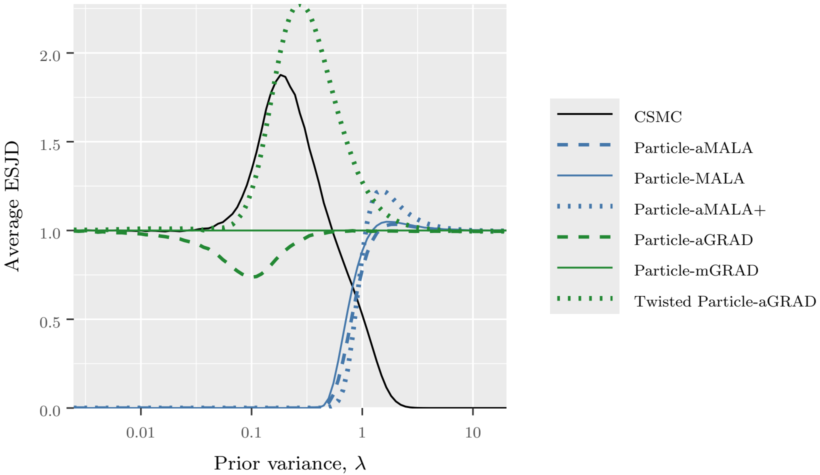
$$\mathrm{N}(\mathbf{a}_t, \mathbf{B}_t) \approx \overbrace{\mathrm{N}(\mathbf{m}_t, \mathbf{C}_t)}^{\text{CSMC proposal}}.$$

- If prior is highly **uninformative** (all eigenvalues of $\mathbf{C}_t$ large) then $\mathbf{A}_t \approx \mathbf{I}$ and

$$\mathrm{N}(\mathbf{a}_t, \mathbf{B}_t) \approx \underbrace{\mathrm{N}(\mathbf{x}_t + \tfrac{\delta_t}{2}\nabla_{\mathbf{x}_t} \log \pi_t(\mathbf{x}_{1:t}), \delta_t \mathbf{I})}_{\text{(marginal) Particle-MALA/Particle-aMALA proposal}}.$$

Scaling with prior informativeness

$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \lambda \mathbf{I})$, $G_t(\mathbf{x}_{t-1:t}) = \mathrm{N}(\mathbf{y}_t; \mathbf{x}_t, \mathbf{I})$; $T = D = 10$, $N = 31$



$(\text{Average ESJD}) = \frac{1}{TD} \sum_{t=1}^{T} \sum_{d=1}^{D} (x_{t,d}^{\text{new}} - x_{t,d}^{\text{old}})^2.$

Convergence to CSMC for highly informative priors

A1 $M_t(\cdot|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{m}_t, \mathbf{C}_t)$, with G_t bounded and $\mathbf{C}_t$ invertible.

A2 $\exists\, C_0, C_1 \geq 0$ such that $\|\nabla \log G_t(\mathbf{x}_t)\|_2 \leq C_0 + C_1\|\mathbf{x}_t\|_2$.

Proposition 1. *For some $D, T, N \geq 1$, assume* **A1–A2***, and assume that there exists a sequence $(\lambda_k)_{k\geq 1}$ in $(0,\infty)$ with $\max_{t\in[T]} \max \mathrm{eigenval}(\mathbf{C}_{t,k}) \leq \lambda_k \to 0$ as $k \to \infty$. Then for any $\varepsilon > 0$, there exists a sequence $(F_{T,k})_{k\geq 1}$ of subsets of $\mathcal{X}^T$ with $\lim_{k\to\infty} \pi_{T,k}(F_{T,k}) = 1$ such that*

$$\sup_{\mathbf{x}_{1:T}\in F_{T,k}} \|P_{\text{Particle-mGRAD},k}(\cdot|\mathbf{x}_{1:T}) - P_{\text{CSMC},k}(\cdot|\mathbf{x}_{1:T})\|_{\mathrm{TV}} \in \mathrm{O}(\lambda_k^{(1-\varepsilon)/4});$$

$$\sup_{\mathbf{x}_{1:T}\in F_{T,k}} \|P_{\text{Particle-aGRAD},k}(\cdot|\mathbf{x}_{1:T}) - P_{\text{CSMC},k}(\cdot|\mathbf{x}_{1:T})\|_{\mathrm{TV}} \in \mathrm{O}(\lambda_k^{(1-\varepsilon)/4}).$$

Convergence to Particle-MALA for uninformative priors

A3 $\max_{d\in[D]} \int_{\mathcal{X}} x_{t,d}^2 G_t(\mathbf{x}_t)\,\mathrm{d}\mathbf{x}_t < \infty$, where $x_{t,d}$ is the dth component of $\mathbf{x}_t$.

Proposition 2. *For some $D, T, N \geq 1$, assume* **A1–A3**, *and assume that there exists a sequence $(\lambda_k)_{k\geq 1}$ in $(0,\infty)$ with $\min_{t\in[T]} \min \operatorname{eigenval}(\mathbf{C}_{t,k}) \geq \lambda_k \to \infty$ as $k \to \infty$. Then for any $\varepsilon > 0$, there exists a sequence $(F_{T,k})_{k\geq 1}$ of subsets of $\mathcal{X}^T$ with $\lim_{k\to\infty} \pi_{T,k}(F_{T,k}) = 1$ such that*

$$\sup_{\mathbf{x}_{1:T}\in F_{T,k}} \|P_{\text{Particle-mGRAD},k}(\,\cdot\,|\mathbf{x}_{1:T}) - P_{\text{Particle-MALA},k}(\,\cdot\,|\mathbf{x}_{1:T})\|_{\mathrm{TV}} \in \mathrm{O}(\lambda_k^{-(1-\varepsilon)/4});$$

$$\sup_{\mathbf{x}_{1:T}\in F_{T,k}} \|P_{\text{Particle-aGRAD},k}(\,\cdot\,|\mathbf{x}_{1:T}) - P_{\text{Particle-aMALA},k}(\,\cdot\,|\mathbf{x}_{1:T})\|_{\mathrm{TV}} \in \mathrm{O}(\lambda_k^{-(1-\varepsilon)/4}).$$

Talk outline

1. State-space models/Feynman–Kac representation
2. Existing methods
3. Particle extensions of MALA and aMALA
4. Particle extensions of mGRAD and aGRAD
5. Numerical illustration
6. Summary

Multivariate stochastic volatility model

- Potential function/observation density:

$$G_t(\mathbf{x}_{t-1:t}) = g_t(\mathbf{y}_t|\mathbf{x}_t) = \mathrm{N}(\mathbf{y}_t; \mathbf{0}, \mathrm{diag}(\exp \mathbf{x}_t)).$$

Multivariate stochastic volatility model

- Potential function/observation density:

$$G_t(\mathbf{x}_{t-1:t}) = g_t(\mathbf{y}_t|\mathbf{x}_t) = \mathrm{N}(\mathbf{y}_t; \mathbf{0}, \mathrm{diag}(\exp \mathbf{x}_t)).$$

- Mutation kernel/transition density:

$$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = f_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; 0.9\mathbf{x}_{t-1}, \tau\mathbf{H}),$$

where, with $\rho = 0.25$,

$$\mathbf{H} = \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & 1 \end{bmatrix}.$$

Multivariate stochastic volatility model

- Potential function/observation density:

$$G_t(\mathbf{x}_{t-1:t}) = g_t(\mathbf{y}_t|\mathbf{x}_t) = \mathrm{N}(\mathbf{y}_t; \mathbf{0}, \mathrm{diag}(\exp \mathbf{x}_t)).$$

- Mutation kernel/transition density:

$$M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = f_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; 0.9\mathbf{x}_{t-1}, \tau\mathbf{H}),$$

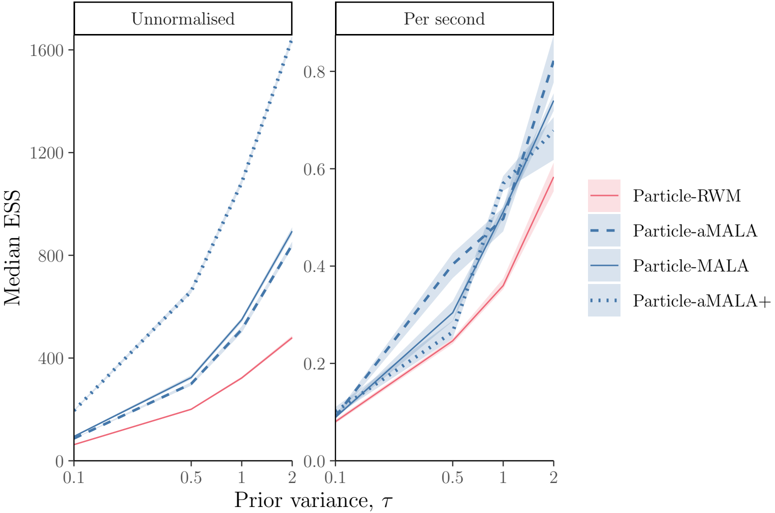
  where, with $\rho = 0.25$,

$$\mathbf{H} = \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & 1 \end{bmatrix}.$$

- The prior variance $\tau > 0$ controls 'prior informativeness'.

Multivariate stochastic volatility model, continued

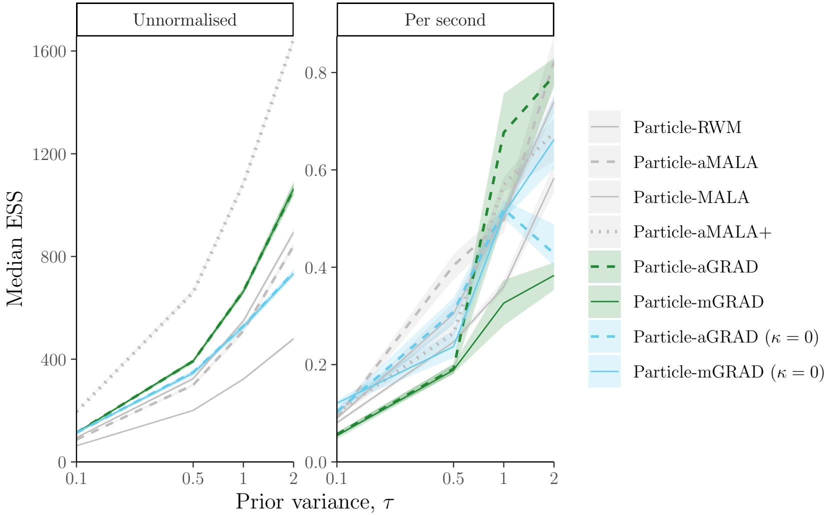
$T = 128$, $D = 30$, $N = 31$; δ_t tuned to achieve 75 % acceptance rate



Proposed methods which do not require (conditionally or unconditionally) Gaussian dynamics compared with Particle-RWM as a baseline.

Multivariate stochastic volatility model, continued

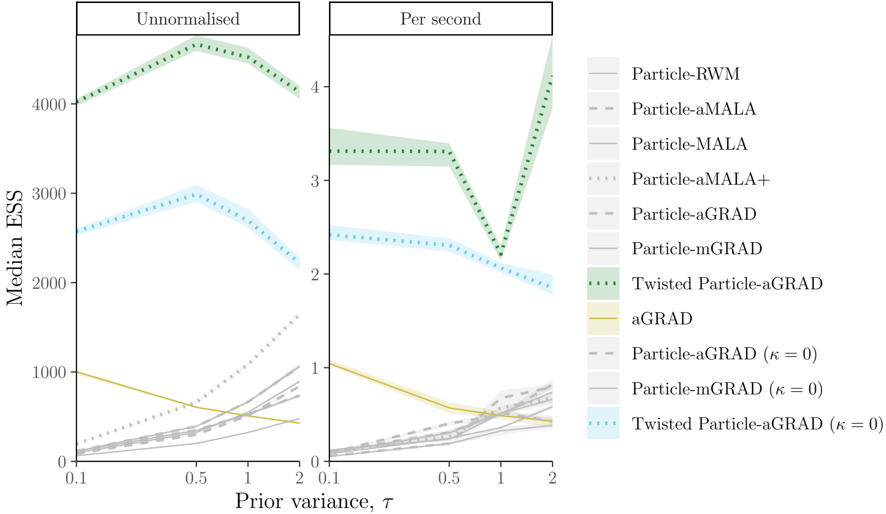
$T = 128$, $D = 30$, $N = 31$; δ_t tuned to achieve 75 % acceptance rate



Proposed methods which require only *conditionally* Gaussian dynamics, i.e., $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{m}_t(\mathbf{x}_{t-1}), \mathbf{C}_t(\mathbf{x}_{t-1}))$ (Particle-mGRAD algorithm also needs $\mathbf{C}_t(\mathbf{x}_{t-1}) = \mathbf{C}_t$). '$(\kappa = 0)$' indicates no gradient usage.

Multivariate stochastic volatility model, continued

$T = 128$, $D = 30$, $N = 31$; δ_t tuned to achieve 75 % acceptance rate



Proposed methods which require *unconditionally* Gaussian dynamics i.e., $M_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t; \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{b}_t, \mathbf{C}_t)$, compared with aGRAD (which also makes this assumption) as baseline. '($\kappa = 0$)' indicates no gradient usage.

Talk outline

Summary

- We have proposed MCMC methods for state-inference in high-dimensional state-space models.

Summary

- We have proposed MCMC methods for state-inference in high-dimensional state-space models.
- From 'classical' MCMC (e.g., MALA or mGRAD) our methods borrow the ability to make local moves guided by gradient or Gaussian-prior information:

Summary

- We have proposed MCMC methods for state-inference in high-dimensional state-space models.
- From 'classical' MCMC (e.g., MALA or mGRAD) our methods borrow the ability to make local moves guided by gradient or Gaussian-prior information:
  $\rightsquigarrow$ favourable scaling with the state dimension, D.

Summary

- We have proposed MCMC methods for state-inference in high-dimensional state-space models.
- From 'classical' MCMC (e.g., MALA or mGRAD) our methods borrow the ability to make local moves guided by gradient or Gaussian-prior information:
  $\rightsquigarrow$ favourable scaling with the state dimension, D.
- From CSMC, our methods borrow the ability to exploit the 'decorrelation-over-time' model property:

Summary

- We have proposed MCMC methods for state-inference in high-dimensional state-space models.
- From 'classical' MCMC (e.g., MALA or mGRAD) our methods borrow the ability to make local moves guided by gradient or Gaussian-prior information:
  $\rightsquigarrow$ favourable scaling with the state dimension, D.
- From CSMC, our methods borrow the ability to exploit the 'decorrelation-over-time' model property:
  $\rightsquigarrow$ favourable scaling with the time horizon, T.

Summary

- We have proposed MCMC methods for state-inference in high-dimensional state-space models.
- From 'classical' MCMC (e.g., MALA or mGRAD) our methods borrow the ability to make local moves guided by gradient or Gaussian-prior information:
  $\rightsquigarrow$ favourable scaling with the state dimension, D.
- From CSMC, our methods borrow the ability to exploit the 'decorrelation-over-time' model property:
  $\rightsquigarrow$ favourable scaling with the time horizon, T.
- All our methods can be implemented in $\mathrm{O}(NT)$ operations per iteration (for fixed D).

Summary

- We have proposed MCMC methods for state-inference in high-dimensional state-space models.
- From 'classical' MCMC (e.g., MALA or mGRAD) our methods borrow the ability to make local moves guided by gradient or Gaussian-prior information:
  $\rightsquigarrow$ favourable scaling with the state dimension, D.
- From CSMC, our methods borrow the ability to exploit the 'decorrelation-over-time' model property:
  $\rightsquigarrow$ favourable scaling with the time horizon, T.
- All our methods can be implemented in $\mathrm{O}(NT)$ operations per iteration (for fixed D).
- All our methods are exact (they leave $\pi_T(\mathbf{x}_{1:T})$ invariant).

Summary, continued

The methods mentioned in this work (new methods are in *italic*).

Method	Special case if $N = T = 1$
CSMC[†]	IMH
Particle-RWM	RWM
Particle-aMALA	aMALA
Particle-MALA	MALA
Particle-aMALA+	aMALA
Particle-aGRAD	aGRAD
Particle-mGRAD	mGRAD
Particle-aGRAD+	aGRAD
Twisted Particle-aGRAD(+)	aGRAD
Particle-PCNL & more[‡]	PCNL

[†] In our taxonomy, CSMC could be called 'Particle-IMH'. However, the latter already refers to a different algorithm in Andrieu et al. (2010).

[‡] auxiliary, smoothing-gradient ('+') and twisted versions.

- For $T = 1$ and $N = 1$, our methods reduce to well known 'classical' MCMC algorithms.

Summary, continued

The methods mentioned in this work (new methods are in *italic*).

Method	**Special case if $N = T = 1$**
CSMC[†]	IMH
Particle-RWM	RWM
Particle-aMALA	aMALA
Particle-MALA	MALA
Particle-aMALA+	aMALA
Particle-aGRAD	aGRAD
Particle-mGRAD	mGRAD
Particle-aGRAD+	aGRAD
Twisted Particle-aGRAD(+)	aGRAD
Particle-PCNL & more[‡]	PCNL

[†] In our taxonomy, CSMC could be called 'Particle-IMH'. However, the latter already refers to a different algorithm in Andrieu et al. (2010).

[‡] auxiliary, smoothing-gradient ('+') and twisted versions.

- For $T = 1$ and $N = 1$, our methods reduce to well known 'classical' MCMC algorithms.
- For $T = 1$ and $N > 1$, our methods are novel 'multi-proposal' variants of these 'classical' MCMC algorithms.

Summary, continued

The methods mentioned in this work (new methods are in *italic*).

Method	**Special case if $N = T = 1$**
CSMC[†]	IMH
Particle-RWM	RWM
Particle-aMALA	aMALA
Particle-MALA	MALA
Particle-aMALA+	aMALA
Particle-aGRAD	aGRAD
Particle-mGRAD	mGRAD
Particle-aGRAD+	aGRAD
Twisted Particle-aGRAD(+)	aGRAD
Particle-PCNL & more[‡]	PCNL

[†] In our taxonomy, CSMC could be called 'Particle-IMH'. However, the latter already refers to a different algorithm in Andrieu et al. (2010).

[‡] auxiliary, smoothing-gradient ('+') and twisted versions.

- For $T = 1$ and $N = 1$, our methods reduce to well known 'classical' MCMC algorithms.
- For $T = 1$ and $N > 1$, our methods are novel 'multi-proposal' variants of these 'classical' MCMC algorithms.
- More details: https://arxiv.org/pdf/2401.14868

Literature I

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342. With discussion.

Andrieu, C., Lee, A., and Vihola, M. (2018). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2):842–872.

Andrieu, C. and Vihola, M. (2016). Establishing some order amongst exact approximations of MCMCs. *Annals of Applied Probability*, 26(5):2661–2696.

Besag, J. E. (1994). Contribution to the discussion on 'Representations of knowledge in complex systems' by Grenander, U and Miller, M. I.. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 56(4):549–581.

Ceperley, D. M. and Dewing, M. (1999). The penalty method for random walks with uncertain energies. *The Journal of Chemical Physics*, 110(20):9812–9820.

Corenflos, A. and Särkkä, S. (2023). Auxiliary MCMC and particle Gibbs samplers for parallelisable inference in latent dynamical systems. *arXiv preprint arXiv:2303.00301*.

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013). MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, 28(3):424–446.

Finke, A. and Thiery, A. H. (2023). Conditional sequential Monte Carlo in high dimensions. *The Annals of Statistics*, 51(2):437–463.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Koskela, J., Jenkins, P. A., Johansen, A. M., and Spano, D. (2020). Asymptotic genealogies of interacting particle systems with an application to sequential Monte Carlo. *The Annals of Statistics*, 48(1):560–583.

Lee, A., Singh, S. S., and Vihola, M. (2020). Coupled conditional backward sampling particle filter. *Annals of Statistics*, 48(5):3066–3089.

Malory, S. (2021). *Bayesian inference for stochastic processes*. PhD thesis, Lancaster University.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Titsias, M. K. and Papaspiliopoulos, O. (2018). Auxiliary gradient-based sampling algorithms. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(4):749–767.