Hierarchical and Sequential Monte Carlo Fusion
A method of unifying distributed analyses

Ryan Chan [1,3], Murray Pollock [1,3], Gareth Roberts [1,3], Petros Dellaportas [2,3]
[1] University of Warwick, [2] University College London, [3] The Alan Turing Institute

Monte Carlo Fusion

Provides theory to carry out perfect inference for the target:

$$\pi(\boldsymbol{x}) \propto f_1(\boldsymbol{x}) \cdots f_C(\boldsymbol{x}) = \prod_{c=1}^{C} f_c(\boldsymbol{x}) \quad (1)$$

Uses rejection sampling on the extended target distribution:

$$g(\boldsymbol{x}^{(1:C)}, \boldsymbol{y}) \propto \prod_{c=1}^{C} \left[f_c^2(\boldsymbol{x}^{(c)}) p_c(\boldsymbol{y} \mid \boldsymbol{x}^{(c)}) \cdot \frac{1}{f_c(\boldsymbol{y})} \right] \quad (2)$$

Let $p_c(\boldsymbol{y} \mid \boldsymbol{x}^{(c)})$ is the transition density of a stochastic process with stationary distribution $f_c^2(\boldsymbol{x})$, then (2) admits π as a marginal for $\boldsymbol{y}$.

Proposal distribution:

$$h(\boldsymbol{x}^{(1:C)}, \boldsymbol{y}) \propto \prod_{c=1}^{C} \left[f_c\left(\boldsymbol{x}^{(c)}\right) \right] \cdot \exp\left(-\frac{C \cdot \|\boldsymbol{y} - \bar{\boldsymbol{x}}\|^2}{2T} \right) \quad (3)$$

Considering the transition probability of a d-dimensional double Langevin diffusion process, Dai et al. (2019) showed that under certain mild conditions,

$$\frac{g(\boldsymbol{x}^{(1:C)}, \boldsymbol{y})}{h(\boldsymbol{x}^{(1:C)}, \boldsymbol{y})} \propto \rho \cdot Q \quad (4)$$

where ρ and Q are two probability values, defined as

$$\rho := e^{-\frac{C\sigma^2}{2T}}, \qquad \sigma^2 = C^{-1} \sum_{c=1}^{C} \left\| \boldsymbol{x}^{(c)} - \bar{\boldsymbol{x}} \right\|^2 \quad (5)$$

$$Q := \mathbb{E}_{\bar{\mathbb{W}}} \left(\prod_{c=1}^{C} \left[\exp\left\{ -\int_0^T \left(\phi_c(\boldsymbol{x}_t^{(c)}) - \Phi_c \right) \mathrm{d}t \right\} \right] \right) \quad (6)$$

where $\bar{\mathbb{W}}$ denotes the law of C Brownian bridges with $\boldsymbol{x}_0^{(c)} = \boldsymbol{x}^{(c)}$ and $\boldsymbol{x}_T^{(c)} = \boldsymbol{y}$ in $[0, T]$.

Algorithm 1 Monte Carlo Fusion (Dai et al. 2019)

1. Initialise a value for $T > 0$
2. Simulate a proposal $\boldsymbol{y}$ from h:
   a) For $c = 1, \ldots, C$, simulate $\boldsymbol{x}_c \sim f_c(\boldsymbol{x})$ and calculate $\bar{\boldsymbol{x}}$
   b) Simulate $\boldsymbol{y} \sim \mathcal{N}_d(\bar{\boldsymbol{x}}, \frac{T\mathbb{I}_d}{C})$
3. Accept $\boldsymbol{y}$ as a sample from (1) with probability $\rho \cdot Q$

<u>Problem:</u> Fusion becomes inefficient as the number of sub-posteriors increases.

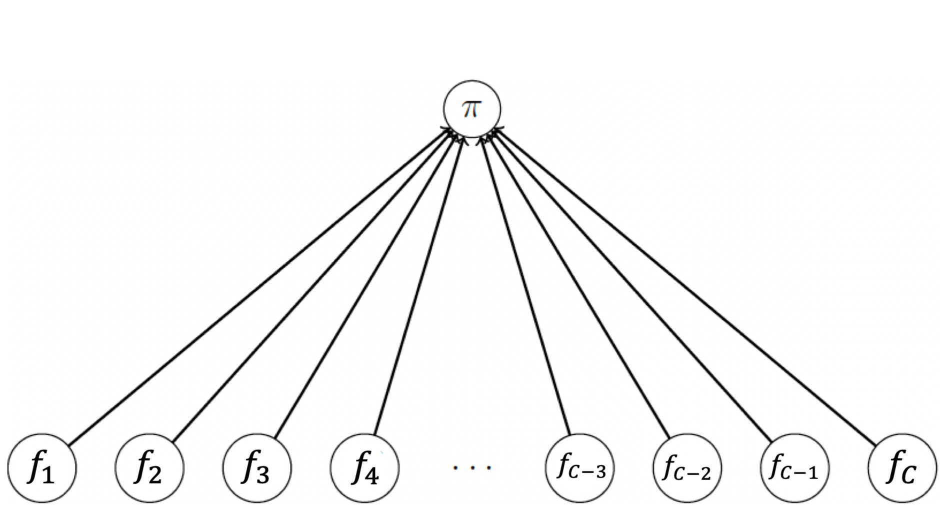

Figure 1: The 'fork-and-join' approach

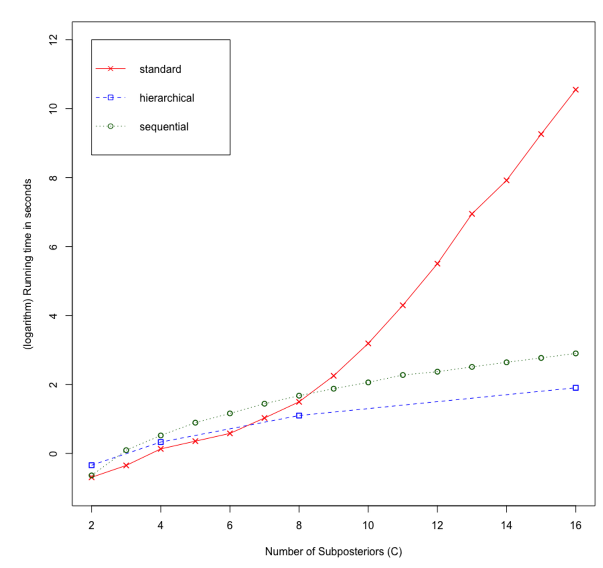

Figure 2: Run-times of MC fusion for $f_c \propto e^{-\frac{x^4}{2C}}$ for $c = 1, \ldots, C$ for varying C

Hierarchical and Sequential Monte Carlo Fusion

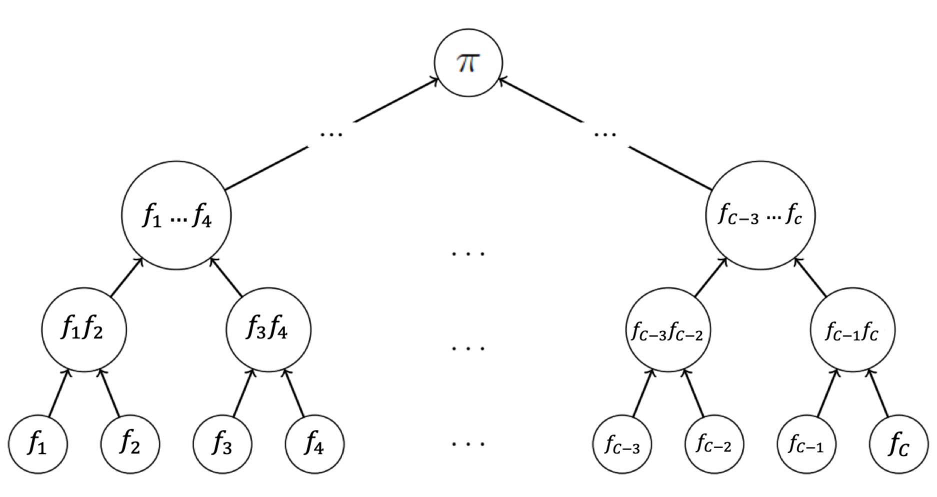

Figure 3: The hierarchical approach

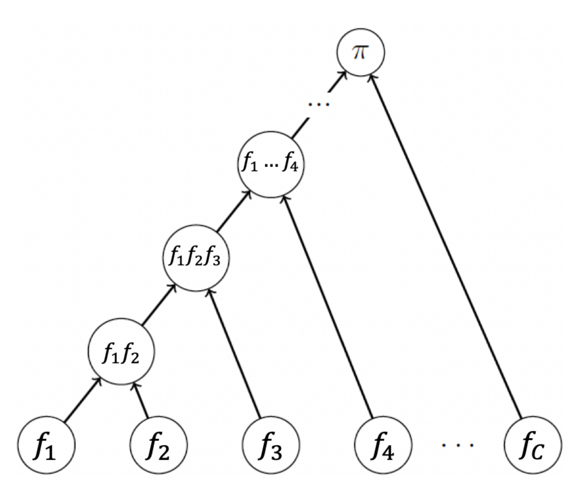

Figure 4: The sequential approach

Acknowledgements: RC is funded by The Alan Turing Institute Doctoral Studentship, under the EPSRC grant EP/N510129/1.

Time-adapting Monte Carlo Fusion

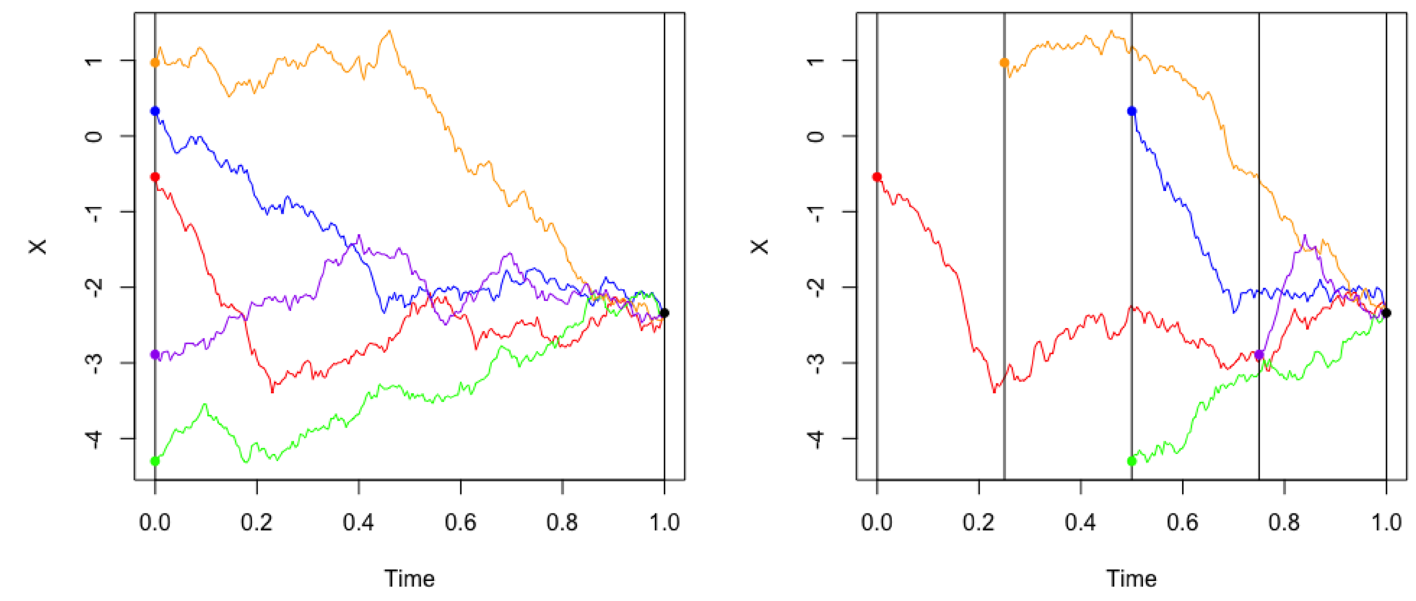

Figure 5: Comparison between regular fusion and time-adapting fusion

Tempering for Monte Carlo Fusion

<u>Problem:</u> Fusion is inefficient when the sub-posteriors conflict.

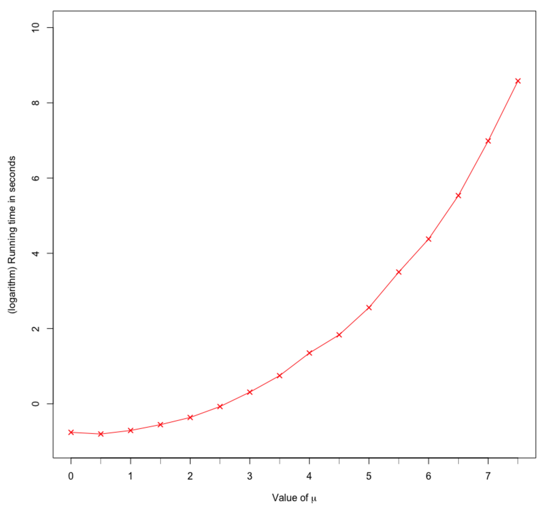

Figure 6: Run-times of MC fusion for $f_1 \sim \mathcal{N}(0,1)$ and $f_2 \sim \mathcal{N}(\mu, 1)$ for different μ

Let $f_c^\beta(\boldsymbol{x})$ be the *power-tempered sub-posterior*, for $\beta \in (0, 1]$, then

$$\pi(\boldsymbol{x}) \propto \prod_{i=1}^{\frac{1}{\beta}} \left[\prod_{c=1}^{C} f_c^\beta(\boldsymbol{x}) \right] \quad \text{where } \frac{1}{\beta} \in \mathbb{N} \quad (7)$$

Example. Target $\pi(x) \propto f_1 f_2$, where $f_1 \sim \mathcal{N}(-8, 1)$ and $f_2 \sim \mathcal{N}(2, 0.5)$.

1. Use Monte Carlo fusion to obtain samples for $\pi^\beta \propto f_1^\beta f_2^\beta$ with $\beta = \frac{1}{8}$:

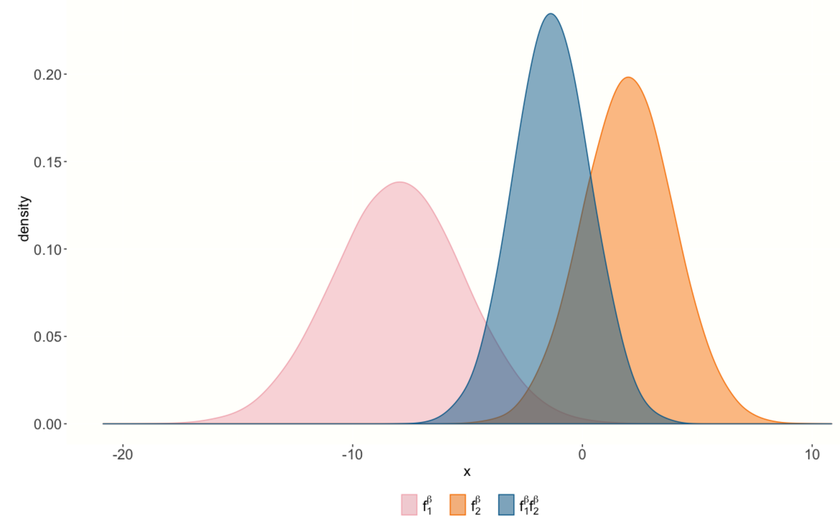

Figure 7: Kernel density fitting based on $10,000$ realisations for density proportional to $f_1^\beta f_2^\beta$ (blue) and the true density curves for f_1^β (pink) and f_2^β (orange)

2. Use hierarchical or sequential Monte Carlo fusion and samples for π^β to obtain samples for $\pi \propto f_1 f_2$

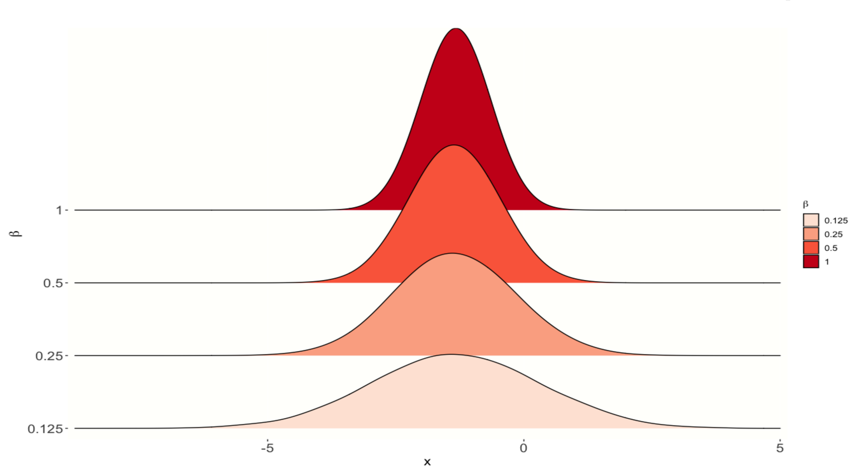

Figure 8: Kernel density fitting for π^β (hierarchical Monte Carlo fusion)

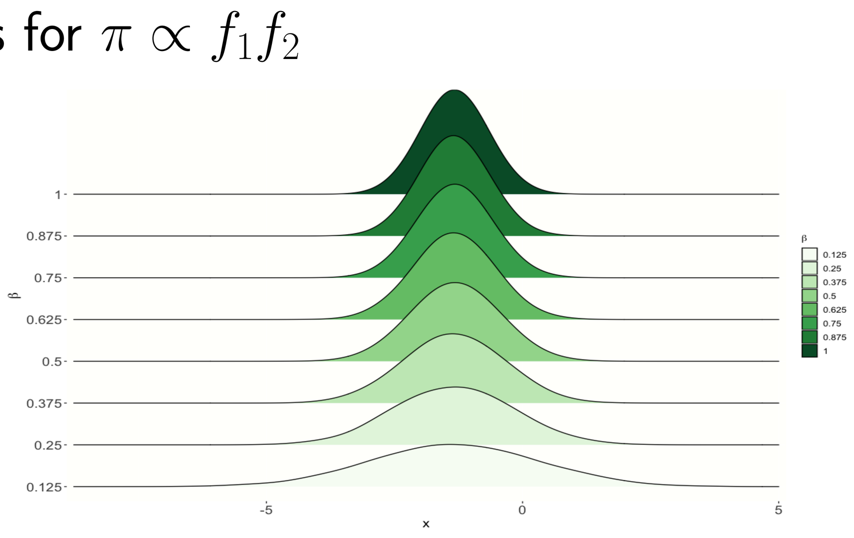

Figure 9: Kernel density fitting for π^β (sequential Monte Carlo fusion)

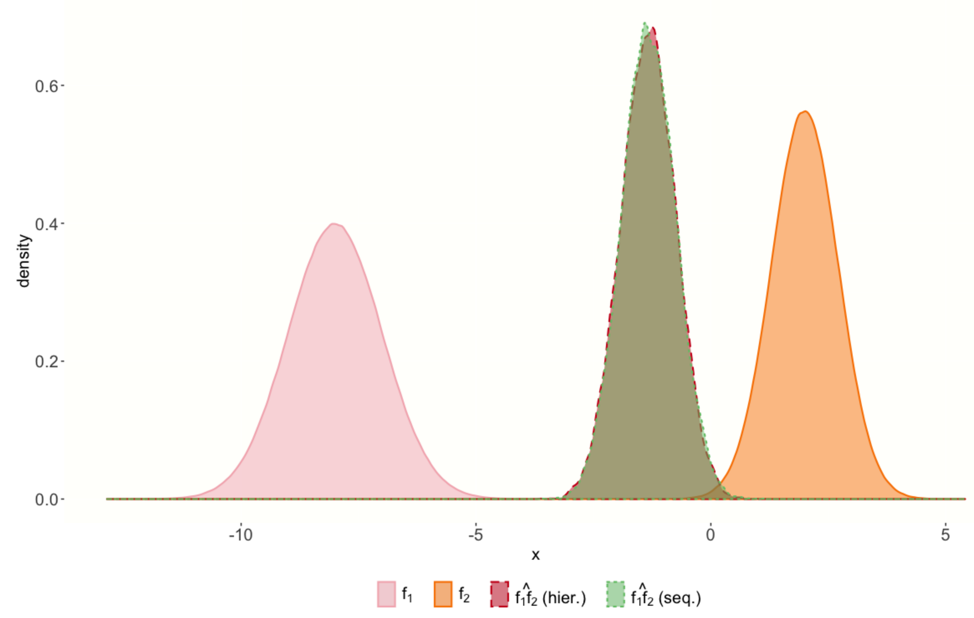

Figure 10: Kernel density fitting based on $10,000$ realisations for density proportional to π based on hierarchical fusion (red dashed), sequential fusion (green dotted), and the true density curves for f_1 (pink) and f_2 (orange)