Differentiable Systematic Resampling for Variational Sequential Monte Carlo

Fredrik Cumlin
Dept. of Information Science and Engineering
KTH Royal Institute of Technology
Stockholm, Sweden
fcumlin@kth.se

Saikat Chatterjee
Dept. of Information Science and Engineering
KTH Royal Institute of Technology
Stockholm, Sweden
sach@kth.se

Abstract

Particle filters are a standard tool for nonlinear state estimation, but their resampling step is discrete, preventing gradient-based learning in variational sequential Monte Carlo. We introduce Differentiable Systematic Resampling (DSR), a temperature-controlled relaxation of systematic resampling, that preserves the CDF-ordered, banded structure of systematic resampling while enabling full gradient flow. DSR converges to exact systematic resampling as the temperature vanishes, and we prove a pointwise exponential convergence rate for the induced bias. Compared to optimal-transport-based differentiable resampling, DSR avoids iterative solvers and has substantially lower computational overhead. Experiments on stochastic dynamical systems and real-world handwriting data show that DSR achieves comparable or superior filtering and dynamics learning performance.

1 Introduction

Sequential Monte Carlo (SMC) methods, commonly known as particle filters (PFs) [14, 9], are standard tools for Bayesian state estimation in nonlinear and non-Gaussian dynamical systems, with applications in robotics, econometrics, computer vision, and epidemiology [29, 27, 4]. Particle filters approximate the filtering posterior by propagating and reweighting a set of particles using the system dynamics and incoming observations. In many applications, however, the state dynamics are unknown or only partially specified; that is, the transition model (system dynamics) is not available to the practitioner. One line of work addresses this by learning a state estimator directly from observations without an explicit particle representation [13, 12, 28, 7]; here we instead focus on variational sequential Monte Carlo (VSMC) [26, 23, 22], which jointly learns a parameterized transition model and proposal model by maximizing a variational lower bound on the marginal likelihood using gradient-based optimization. Importantly, VSMC requires the entire filtering recursion to be differentiable.

An essential component of SMC, and hence of VSMC, is resampling. During filtering, particle weights tend to degenerate, meaning that only one or a few particles carry most of the weight [10]. Resampling mitigates this by replacing low-weight particles with copies of high-weight ones [8, 20]. Among standard resampling schemes, systematic resampling is widely adopted for its simplicity and low variance in practice [8, 20, 10]. However, resampling relies on discrete ancestor assignments and is therefore inherently non-differentiable. While particle propagation and importance weighting can be differentiated via the reparameterization trick [19], the resampling step blocks gradient flow and produces biased or high-variance gradient estimators [26]. This makes differentiable resampling a central challenge in VSMC. A standard approach for discrete operations in gradient-based learning is to replace them with temperature-controlled continuous relaxations that recover the discrete operation in the zero-temperature limit [15, 24]. We take this approach for the resampling step of VSMC, but rather than relaxing the ancestor categorical directly, we relax the assignment of systematic

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

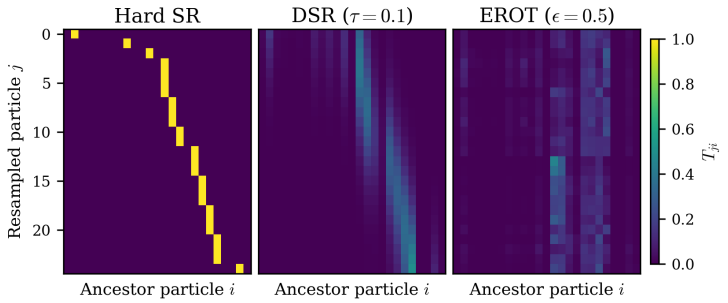


Figure 1: Transportation maps for systematic resampling (left), DSR (middle), and EROT (right), for a sequential importance resampling PF of the Lorenz-63 system. The transportation maps are calculated at step 150 using $N = 25$ particles, and DSR and EROT use the training relaxation parameters.

resampling, based on the cumulative distribution function (CDF) of the weights, hence preserving its CDF-ordered, banded structure.

Several differentiable relaxations of resampling have been proposed. Soft multinomial resampling [17] keeps ancestor sampling discrete but draws ancestors from a smoothed distribution that mixes the importance weights with a uniform distribution, and propagates gradients through the corresponding importance-weight correction. A drawback is that the underlying scheme is multinomial resampling, which has higher variance than residual and stratified resampling and, in practice, than systematic resampling [8]; hence, the method inherits an inferior starting point. Entropy-regularized optimal transport (EROT) [5] instead formulates resampling as an optimal transport problem. However, EROT relies on recursive Sinkhorn updates that scale as $\mathcal{O}(N^2)$ per iteration, is sensitive to the choice of regularization parameter, and, at the regularization levels needed for stable training, produces diffuse transport plans that bear little resemblance to the sparse, CDF-ordered structure of systematic resampling (see Fig. 1). However, neither approach preserves the structure of systematic resampling, which is the de facto standard at inference time due to its favorable variance properties [10].

In this paper, we introduce Differentiable Systematic Resampling (DSR), a temperature-controlled relaxation that directly relaxes the CDF-level ancestor assignment of systematic resampling using sigmoid functions. To further ensure that neighbours in the CDF correspond to spatially nearby particles, we equip DSR with Morton space-filling curve sorting, which reduces approximation error to systematic resampling. The relaxation introduces a bias that we prove vanishes exponentially as the temperature $\tau \to 0^+$. Our main contributions are as follows:

- We propose DSR, a differentiable relaxation that preserves the CDF-ordered, banded structure of systematic resampling while enabling full gradient flow.
- We establish theoretical guarantees showing that DSR converges to exact systematic resampling, with a pointwise exponential rate under a positive-margin condition, as $\tau \to 0^+$.
- We demonstrate that DSR achieves competitive or superior state estimation and dynamics learning in VSMC while training more than $2\times$ faster than EROT.

2 Background

2.1 State space models and particle filters

We consider state space models $\mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1})$, $\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t)$, where $\mathbf{x}_t \in \mathbb{R}^{d_x}$ is the latent state and $\mathbf{y}_t \in \mathbb{R}^{d_y}$ the observation. Particle filters approximate the filtering posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^N w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)})$ using N weighted particles $\{(\mathbf{x}_t^{(i)}, w_t^{(i)})\}_{i=1}^N$ at time t. During inference, particles are propagated through the dynamics, reweighted by the observation likelihood, and resampled to avoid weight degeneracy. Resampling replaces the weighted particle set with an equal-weight set by sampling ancestor indices according to the weights.

Algorithm 1 Variational Sequential Monte Carlo (VSMC)

Require: Initial particles $\{\tilde{\mathbf{x}}_0^{(i)}\}_{i=1}^N$, observations $\{\mathbf{y}_t\}_{t=1}^T$
1: **for** $t = 1$ to T **do**
2: Sample $\mathbf{x}_t^{(i)} \sim q_{\boldsymbol{\theta}}(\mathbf{x}_t|\tilde{\mathbf{x}}_{t-1}^{(i)}, \mathbf{y}_t)$ for $i = 1, \ldots, N$
3: Compute weights $w_t^{(i)} = \frac{p_{\boldsymbol{\phi}}(\mathbf{x}_t^{(i)}|\tilde{\mathbf{x}}_{t-1}^{(i)})p(\mathbf{y}_t|\mathbf{x}_t^{(i)})}{q_{\boldsymbol{\theta}}(\mathbf{x}_t^{(i)}|\tilde{\mathbf{x}}_{t-1}^{(i)}, \mathbf{y}_t)}$; normalize $\sum_i w_t^{(i)} = 1$
4: $\{\tilde{\mathbf{x}}_t^{(i)}\}_{i=1}^N \leftarrow \texttt{Resample}\big(\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N\big)$ *(discrete, non-differentiable)*
5: **end for**

2.2 Variational Sequential Monte Carlo

Variational Sequential Monte Carlo (VSMC) [26] provides a framework for jointly learning a transition model $p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$ and proposal distribution $q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ by maximizing the so-called surrogate evidence lower-bound (ELBO):

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}\left[\log \hat{p}(\mathbf{y}_{1:T})\right] = \mathbb{E}\left[\sum_{t=1}^{T} \log\left(\frac{1}{N}\sum_{i=1}^{N} w_t^{(i)}\right)\right], \tag{1}$$

where the expectation is over all sampling and resampling operations in the particle filter (the weights before normalization are used). Algorithm 1 summarizes the VSMC recursion.

Optimizing $\mathcal{L}$ requires differentiating through the entire filtering recursion. As shown by [26], the gradient decomposes into two terms:

$$\nabla\mathcal{L} = \underbrace{\mathbb{E}[\nabla \log \hat{p}(\mathbf{y}_{1:T})]}_{g_{\text{rep}}\text{ (reparameterization)}} + \underbrace{\mathbb{E}\big[\log \hat{p}(\mathbf{y}_{1:T})\ \nabla \log \tilde{\varphi}(\mathbf{a}_{1:T-1}^{1:N}|\boldsymbol{\varepsilon}_{1:T}^{1:N})\big]}_{g_{\text{score}}\text{ (score function)}}, \tag{2}$$

where $\tilde{\varphi}$ is the marginal distribution over the discrete ancestor variables $\mathbf{a}_{1:T-1}^{1:N}$ given the reparameterized noise $\boldsymbol{\varepsilon}_{1:T}^{1:N}$. The reparameterization term g_{rep} differentiates through the continuous particle propagation and weighting steps via the reparameterization trick [19]. The score-function term g_{score} accounts for the discrete ancestor assignments introduced by resampling, which are not reparameterizable and can have high variance [26, cf. Fig. 3].

In practice, g_{score} is dropped, yielding the biased gradient $\nabla\mathcal{L} \approx g_{\text{rep}}$ [26]. This treats ancestor indices as fixed during backpropagation: gradients flow through the surviving particle values, but not through the ancestor selection itself, a limitation that becomes severe when weights are concentrated and resampling decisions dominate filtering performance (see Appendix F.3). Differentiable resampling methods address this by replacing the discrete ancestor selection with a continuous relaxation, so that g_{rep} flows through the full resampling mechanism. This yields a relaxed objective in place of the surrogate ELBO. At finite temperature the relaxation does not preserve expected replication counts, so the resulting likelihood estimator $\hat{p}(\mathbf{y}_{1:T})$ is no longer unbiased and the relaxed objective is a differentiable surrogate rather than a lower bound on $\log p(\mathbf{y}_{1:T})$ (see Section 3.1). The relaxed objectives have been shown to provide a richer learning signal for the transition model and proposal with improved state estimation and system identification across tasks [5, 31, 3, 21, 6].

3 Differentiable Systematic Resampling

Traditional resampling methods are discrete and non-differentiable, creating a bottleneck for gradient-based learning. We propose differentiable systematic resampling (DSR): a soft relaxation that converges to systematic resampling in the limit, while allowing gradients to flow through the resampling step. For ease of notation, we omit time subscripts in Sections 3 and 4; particle indices are instead denoted as subscripts.

3.1 DSR formulation

Recall that systematic resampling computes the CDF $F_i = \sum_{k=1}^{i} w_k$, draws a single offset $u_0 \sim \mathcal{U}(0, 1/N)$, and assigns $a_j = \min\{i : F_i \geq u_j\}$ where $u_j = u_0 + (j-1)/N$. Equivalently, a_j is

the unique index i for which $F_{i-1} < u_j \leq F_i$, so the discrete assignment can be written as a matrix $A_{ij} = \mathbb{1}(F_{i-1} < u_j \leq F_i)$. This indicator is non-differentiable with respect to the weights, blocking gradient flow.

Consider $N = 3$ particles with weights $\mathbf{w} = (0.1, 0.7, 0.2)$, giving the CDF $\mathbf{F} = (0.1, 0.8, 1.0)$ and $F_0 = 0$. Systematic resampling draws a single offset $u_0 \sim \mathcal{U}(0, \frac{1}{3})$ and places the comb $u_j = u_0 + (j-1)/3$. For $u_0 = 0.05$ the comb points are $u_1 = 0.05$, $u_2 = 0.38$, $u_3 = 0.72$, and thresholding each against the CDF gives ancestors $a_1 = 1$ (since $F_0 < u_1 \leq F_1$), $a_2 = 2$ (since $F_1 < u_2 \leq F_2$), and $a_3 = 2$ (since $F_1 < u_3 \leq F_2$): the high-weight particle 2 is replicated, the low-weight particle 3 is dropped, and the assignment matrix is $A_{ij} = \mathbb{1}(F_{i-1} < u_j \leq F_i)$. The only randomness is the single draw u_0; given u_0, the assignment is a deterministic thresholding of the comb against the CDF. The idea with DSR is to relax only this thresholding step and to leave the draw u_0 untouched.

Our strategy is to replace the indicator with a smooth surrogate that recovers it in a limiting sense, following the temperature-relaxation approach common for discrete operations in gradient-based learning [15, 24]. Writing the indicator as a telescoping difference,

$$\mathbb{1}(F_{i-1} < u \leq F_i) = \mathbb{1}(u > F_{i-1}) - \mathbb{1}(u > F_i), \tag{3}$$

reduces the problem to relaxing each step indicator $\mathbb{1}(u > F)$ separately. The sigmoid $\sigma(x) = (1 + e^{-x})^{-1}$ provides a natural smooth surrogate: $\sigma((u - F)/\tau) \to \mathbb{1}(u > F)$ pointwise as the temperature $\tau \to 0^+$. Substituting into the telescoping identity with boundary conventions $F_0 = 0$, $F_N = 1$ gives the soft assignment:

$$s_i(u; \tau) = \sigma\left(\frac{u - F_{i-1}}{\tau}\right) - \sigma\left(\frac{u - F_i}{\tau}\right), \quad i = 1, \ldots, N \tag{4}$$

Each $s_i(u; \tau) \in [0, 1]$ is differentiable in u, F_{i-1}, and F_i, and $\sum_i s_i(u; \tau) \to 1$ as $\tau \to 0^+$. At finite τ, the sum $\sum_i s_i(u; \tau)$ is not necessarily one, so we normalize to obtain a proper convex combination. The transportation matrix and resampled particles are:

$$T_{ij}^{(\tau)} = \frac{s_i(u_j; \tau)}{\sum_{k=1}^{N} s_k(u_j; \tau)}, \qquad \tilde{\mathbf{x}}_j^{(\tau)} = \sum_{i=1}^{N} T_{ij}^{(\tau)} \mathbf{x}_i. \tag{5}$$

Each column of $T^{(\tau)}$ sums to one, so $\tilde{\mathbf{x}}_j^{(\tau)}$ is a convex combination of the ancestor particles. The row sums, however, satisfy $\sum_{j=1}^{N} T_{ij}^{(\tau)} \neq N w_i$ in general, so DSR does not preserve expected replication counts at finite τ. The temperature τ controls the relaxation between gradient flow and approximation fidelity: small τ gives a sharp transport plan close to systematic resampling but produces large gradient magnitudes, while large τ gives a diffuse transport plan with smoother gradients. We show in Section 4 that the approximation error of each resampled particle decays exponentially in $1/\tau$, and demonstrate in Section 5 that $\tau = 0.1$ is a robust choice across all our experiments. Forming the full transport matrix $T^{(\tau)} \in \mathbb{R}^{N \times N}$ makes DSR $\mathcal{O}(N^2)$ per resampling step, compared with $\mathcal{O}(N)$ for hard systematic resampling; the resulting overhead is nonetheless small in practice due to batch processing on GPU (Table 5)[1]. An algorithmic overview of DSR is given in Appendix A.

3.2 Particle ordering

Since DSR operates on the empirical CDF, the particle ordering determines which particles interact through the soft assignments: particles adjacent in the CDF share mass in the rows of $T^{(\tau)}$. It is therefore desirable to order particles so that CDF neighbours are also spatial neighbours, which keeps the soft mixture more local. Thus, we sort particles along a Morton space-filling curve [25] before constructing the CDF; an algorithmic overview of this sorting is provided in Appendix A. Locality-preserving orderings have been shown to improve convergence properties of systematic resampling [11]; in DSR, they additionally ensure the soft mixture blends spatially nearby ancestors rather than distant particles (for an analysis of this, see Appendix E). The sorting costs $\mathcal{O}(N \log N)$ per resampling step, negligible compared to the overall resampling cost of $\mathcal{O}(N^2)$.

[1] Note that the $\mathcal{O}(N^2)$ cost could be reduced to $\mathcal{O}(NK)$ by keeping, for each resampled particle j, only the K ancestors around its hard ancestor a_j, where the soft assignments concentrate. However, on a GPU the full matrix is cheap, hence, we keep it.

Morton sorting significantly improves the approximation to systematic resampling: at $N = 200$, it reduces the L2 distance between DSR and hard systematic resampled particles by 35% at $\tau = 0.01$ and by 15% at $\tau = 0.1$ (default used in experiments), compared to unsorted ordering (Appendix E). For filtering and transition model quality, Morton sorting offers a modest advantage at low SNR (Appendix F.4).

4 Theoretical Analysis

We show that DSR is a consistent relaxation whose bias vanishes exponentially fast in $1/\tau$. Let $\{(\mathbf{x}_i, w_i)\}_{i=1}^N$ be particles with normalized weights $w_i > 0$, $F_i = \sum_{k=1}^i w_k$ the empirical CDF, and $T_{ij}^{(\tau)}$, $\tilde{\mathbf{x}}_j^{(\tau)}$ as defined in Eq. (5). The first result establishes that DSR recovers systematic resampling in the limit.

Proposition 4.1 (Asymptotic exactness)**.** *Fix $j \in [N]$ and let $a_j^* = \min\{i \in [N] : F_i \geq u_j\}$ denote the hard systematic resampling index. Then $\lim_{\tau \to 0^+} T_{ij}^{(\tau)} = \mathbb{1}(i = a_j^*)$ and $\lim_{\tau \to 0^+} \tilde{\mathbf{x}}_j^{(\tau)} = \mathbf{x}_{a_j^*}$ almost surely.*

The second result quantifies the approximation error for finite τ.

Proposition 4.2 (Exponential convergence rate)**.** *Let $D = \sup_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|_2$ denote the particle diameter and*

$$\Delta_j := \min\{u_j - F_{a_j^*-1},\ F_{a_j^*} - u_j\} > 0 \text{ a.s.} \tag{6}$$

the margin between u_j and the nearest CDF boundary. Then there exists $\tau_0 > 0$ such that for all $\tau \leq \tau_0$,

$$\|\tilde{\mathbf{x}}_j^{(\tau)} - \mathbf{x}_{a_j^*}\|_2 \leq 2DN \exp\left(-\frac{\Delta_j}{\tau}\right). \tag{7}$$

The proofs are in Appendix B.

Proposition 4.2 is primarily a qualitative guarantee: the bound depends on the particle diameter D, the count N, and the margin Δ_j between u_j and the nearest CDF boundary. The dominant term is $\exp(-\Delta_j/\tau)$, so concentrated weights (large Δ_j) yield fast convergence while near-uniform weights, where Δ_j scales as $\mathcal{O}(1/N)$, bound the worst-case rate. The bound holds pointwise in j under the positive-margin event $\Delta_j > 0$ (which holds almost surely); it is not uniform over offsets, since Δ_j can be made arbitrarily small by the weight and offset configuration, so no uniform exponential rate holds. Empirically, the per-particle L2 distance between DSR and hard systematic resampling decreases as $\tau \to 0^+$, from 0.36 at $\tau = 0.1$ to 0.10 at $\tau = 0.01$ for $N = 25$ particles (Appendix E).

5 Experiments

We evaluate DSR on synthetic and real-world state estimation and dynamics learning tasks. The experiments assess four aspects: (1) training-time filtering posterior quality $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$; (2) state estimation accuracy at inference time in standard SMC; (3) transition model quality $\hat{p}(\mathbf{x}_t|\mathbf{x}_{t-1})$; and (4) computational efficiency.

5.1 Experimental Setup

We consider nonlinear state-space models where the transition and proposal models are parameterized by fully connected neural networks and learned jointly under the VSMC objective. All methods use $N = 25$ particles unless stated otherwise (Appendix F.3 reports a sweep over $N \in \{25, 50, 100, 200\}$), the same Adam [18] configuration, identical network architectures, and gradient clipping at norm 5.0. We clip the estimated log-likelihood below to avoid extreme instability in training, see Appendix C. For DSR and EROT, the gradient with respect to the transport plan is clipped elementwise to $[-1, 1]$, as in the implementation [5]. Results are averaged over 10 independent runs (3 for the linear Gaussian SSM); we report mean $\pm$ standard deviation. Training configuration can be found in the Appendix C.

We evaluate on three settings: (1) a 4-dimensional *Linear Gaussian SSM*, where the Kalman filter gives the optimal posterior; (2) *Lorenz-63*, a 3D chaotic attractor with a nonlinear camera observation

map; and (3) *CharacterTrajectories*, real-world handwriting dynamics with 3D position tracking. We use different signal-to-measurement noise (SMNR) ratios to experiment with different noise levels (definition of SMNR can be found in Appendix D.1).

We report NMSE (dB) for state estimation, one-step MSE and five-step MSE for transition model quality, log-likelihood (LL) and Kullback–Leibler divergence (KLD) for distribution matching, and wall-clock training time. Full definitions are in Appendix D.

We compare DSR to three resampling baselines: *Hard*, standard systematic resampling [16], which corresponds to estimating the ELBO gradient via g_{rep} in Eq. (2); *Soft*, relaxation of multinomial resampling with $\alpha = 0.5$ [17]; and *EROT*, entropy-regularized optimal-transport resampling with $\varepsilon = 0.5$ and at most 100 Sinkhorn iterations, with early stopping [5]. All methods produce biased estimates of the surrogate ELBO gradient. Our code is available at `https://github.com/fcumlin/differentiable-systematic-resampling.git` and https://github.com/SachLab/DSR/.

5.2 Linear Gaussian State Space Model

To evaluate DSR in a controlled setting where the ground truth is analytically available, we consider a 4-dimensional linear Gaussian state space model

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{e}_t, \quad \mathbf{y}_t = \mathbf{x}_t + \mathbf{w}_t, \tag{8}$$

with $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I}_4)$, $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}_4)$, process noise at -10 dB and observation noise set by the SMNR. The true transition matrix is a damped block-diagonal rotation $\mathbf{A} = 0.9 \cdot \text{blockdiag}(\mathbf{R}(\theta_1), \mathbf{R}(\theta_2))$ with $\theta_1 = 0.3$, $\theta_2 = 0.5$ and $\mathbf{R}(\theta)$ the 2×2 rotation matrix. Since the model is linear Gaussian, the Kalman filter gives the optimal posterior at each t, which lets us compare the training-time particle distribution against a Bayesian oracle via KLD. We parameterize the transition model as $\hat{\mathbf{A}} \in \mathbb{R}^{4\times4}$ with a learnable diagonal covariance. We train with $N = 25$ particles on 1000 trajectories of length $T = 100$ at SMNR $= 10$ dB.

Table 1: Performance on the linear Gaussian SSM ($d_x = 4$, SMNR $= 10$ dB). KLD KF is the training-time divergence to the Kalman posterior; NMSE is evaluated with $N = 1000$ particles and hard systematic resampling; $\|A - \hat{A}\|_F$ measures recovery of the true transition matrix. Results show mean $\pm$ std over 3 runs.

Method	KLD KF	NMSE (dB)	$\|A - \hat{A}\|_F$
Hard	$\mathbf{8.64}_{\pm 0.02}$	$\underline{-11.33}_{\pm 0.06}$	$0.035_{\pm 0.001}$
Soft	$\underline{9.72}_{\pm 0.05}$	$-10.33_{\pm 0.07}$	$0.025_{\pm 0.002}$
EROT	$12.55_{\pm 0.03}$	$-11.31_{\pm 0.02}$	$\underline{0.022}_{\pm 0.004}$
DSR	$12.30_{\pm 0.03}$	$\mathbf{-11.34}_{\pm 0.01}$	$\mathbf{0.019}_{\pm 0.001}$

Hard achieves the best training-time posterior (KLD 8.64 vs. 12.30 for DSR), yet DSR recovers the true transition matrix most accurately ($\|A - \hat{A}\|_F$ of 0.019 vs. 0.035). A better training-time posterior therefore does not imply a better gradient signal for parameter learning, which is consistent with the gradient-norm analysis on Lorenz-63 (Section 5.3.1), where DSR's smoother gradients lead to better downstream performance despite comparable point-wise posterior quality. Appendix H reports two further linear Gaussian studies: an evaluation-time decomposition of the marginal likelihood, and a comparison of the DSR gradient against an unbiased VSMC gradient.

5.3 Lorenz-63

We evaluate all methods on the (stochastic) Lorenz-63 attractor with a nonlinear camera measurement map, following [31]. The Lorenz-63 system is a 3-dimensional chaotic attractor given by the discretized dynamics

$$\mathbf{x}_{t+1} = F(\mathbf{x}_t)\mathbf{x}_t + \mathbf{e}_t, \quad \mathbf{e}_t \sim \mathcal{N}(0, \sigma_e^2 I), \tag{9}$$

where $F(\mathbf{x}_t) = \exp\left(\begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -x_{t,1} \\ 0 & x_{t,1} & -8/3 \end{bmatrix} \Delta\tau\right)$, with $\Delta\tau = 0.02$ and $\sigma_e^2 = 0.1$ corresponding to -10 dB process noise. The values follow [13, 12]. We use a 2D camera model with Gaussian point

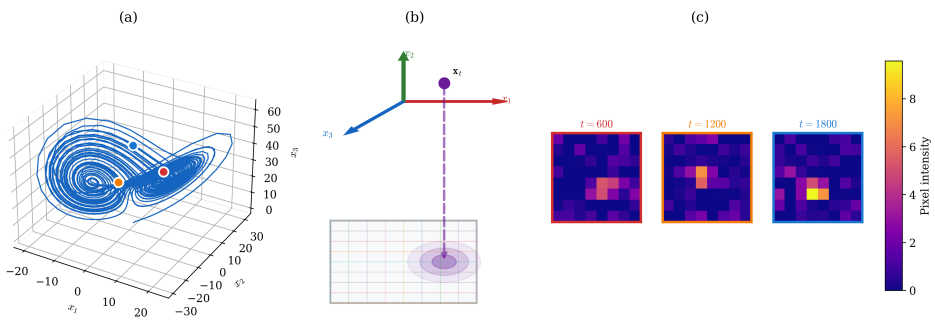


Figure 2: **Lorenz-63 system with nonlinear camera measurement model.** (a) Sample state trajectory of Lorenz-63. Colored points indicate three observation times that are shown in the rightmost figure. (b) Camera projection geometry: the 3D state $\mathbf{x}_t = (x_1, x_2, x_3)$ projects onto an 8×8 pixel grid via a depth-dependent Gaussian point spread function (Eq. 10). The first two coordinates determine the 2D position, while the third coordinate (depth) controls the spread of the Gaussian. (c) Three sample camera observations at SMNR = 0 dB.

Table 2: Performance comparison on the Lorenz-63 system at SMNR = 10 and 0 dB. Results show mean ± std over 10 runs. Best in **bold**, second-best underlined.

	State Estimation	Transition Model Quality			
Method	NMSE (dB)	MSE	MSE-5-step	LL	KLD
SMNR = 10 dB					
Hard	$\underline{-28.86}_{\pm 0.41}$	$\underline{0.55}_{\pm 0.19}$	$5.50_{\pm 3.68}$	$\underline{-2.26}_{\pm 0.25}$	$\underline{5.41}_{\pm 1.47}$
Soft	$-27.82_{\pm 0.45}$	$0.86_{\pm 0.49}$	$6.45_{\pm 3.92}$	$-2.55_{\pm 0.48}$	$6.90_{\pm 2.43}$
EROT	$-28.76_{\pm 0.31}$	$0.56_{\pm 0.24}$	$\underline{3.39}_{\pm 2.59}$	$-2.32_{\pm 0.20}$	$5.74_{\pm 1.03}$
DSR	$\mathbf{-29.94}_{\pm 0.28}$	$\mathbf{0.47}_{\pm 0.24}$	$\mathbf{3.38}_{\pm 2.11}$	$\mathbf{-1.98}_{\pm 0.27}$	$\mathbf{3.60}_{\pm 1.04}$
SMNR = 0 dB					
Hard	$-11.73_{\pm 1.75}$	$0.96_{\pm 0.88}$	$9.64_{\pm 13.71}$	$\mathbf{-3.04}_{\pm 0.40}$	$\mathbf{12.40}_{\pm 3.95}$
Soft	$-11.69_{\pm 1.65}$	$\underline{0.92}_{\pm 0.08}$	$\underline{8.19}_{\pm 6.47}$	$-3.17_{\pm 0.14}$	$13.87_{\pm 2.32}$
EROT	$\underline{-12.57}_{\pm 1.64}$	$4.79_{\pm 5.65}$	$48.41_{\pm 60.72}$	$-4.28_{\pm 1.54}$	$33.82_{\pm 29.79}$
DSR	$\mathbf{-18.28}_{\pm 0.46}$	$\mathbf{0.54}_{\pm 0.33}$	$\mathbf{3.70}_{\pm 2.87}$	$\underline{-3.06}_{\pm 0.24}$	$\underline{13.41}_{\pm 1.94}$

spread functions over an 8×8 pixel grid, following [2, 31]. The grid points $\{g_i\}_{i=1}^{64}$ are uniformly distributed in the rectangle $[-30, 30] \times [-40, 40]$. The state is projected into the camera plane according to its first two coordinates, with the third coordinate representing depth. The measurement function $\mathbf{h} : \mathbb{R}^3 \to \mathbb{R}^{64}$ is given by

$$\mathbf{h}(\mathbf{x}_t)_i = 10 \exp\left(-\frac{0.5}{7 + x_{t,3}} \left\| g_i - \begin{bmatrix} x_{t,1} \\ x_{t,2} \end{bmatrix} \right\|_2^2 \right), \tag{10}$$

where $\mathbf{h}(\mathbf{x}_t)_i$ represents the intensity at pixel i. The observations are given by $\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{w}_t$ with $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I})$, where σ_w^2 is determined by the signal-to-measurement noise ratio (SMNR). For training, we use 1000 trajectories of observations $\mathbf{y}_{1:T}$ with $T = 100$; we experiment with two SMNR levels: 10 dB and 0 dB. A visualization of the Lorenz-63 system under the camera mapping is shown in Fig. 2.

We evaluate both state estimation and transition model quality (Table 2). State estimation uses 10 held-out trajectories with $T = 1000$ and the systematic resampling at inference; the transition model f_θ is evaluated directly on held-out trajectories.

At 10 dB SMNR, DSR achieves the best NMSE (-29.94 dB) with low run-to-run variance, and the best MSE, LL, and KLD on the transition model, matching EROT on multi-step prediction. Hard and Soft struggle more on longer horizons.

The gap widens at 0 dB SMNR, where DSR improves NMSE by 5.7 dB over the second-best method (EROT at -12.57 dB). At low SNR, particle weights become highly concentrated and amplify noisy

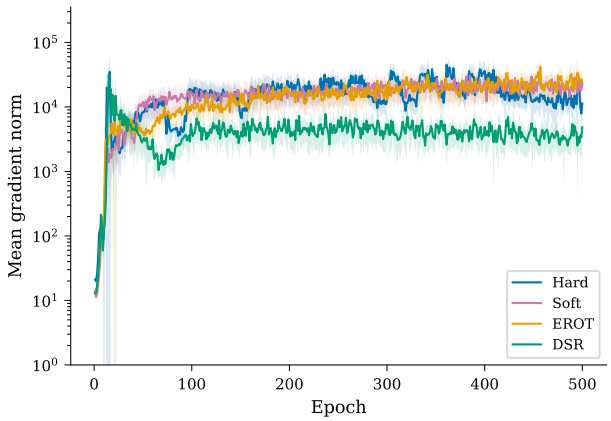


Figure 3: Norm of the proposal network's gradient before clipping during training on Lorenz-63 (SMNR $= 10$ dB). Solid: per-epoch mean over batches, averaged over the 10 runs; shaded: ± 1 std over runs.

gradients; DSR's smaller and steadier gradients (Table 3) yield more stable parameter learning. The resulting transition model is not best on every one-step metric (LL, KLD), but produces particles that filter more effectively at inference time, since one-step metrics from true states do not capture error compounding during sequential filtering.

The large MSE-5-step standard deviation of EROT at 0 dB (60.72) raises the question of whether this reflects the method or the choice of ε. We therefore evaluate EROT at $\varepsilon = 1$ as well (Appendix F.5). Increasing ε stabilizes training (MSE-5-step std drops to 7.58) and improves NMSE to -13.47 dB, but DSR still outperforms EROT on every metric.

5.3.1 Gradient norm

To assess training stability, we study the norm of the proposal network's gradient, before clipping, under each resampling method on Lorenz-63 at SMNR $= 10$ dB. Figure 3 shows the mean gradient norm per epoch. DSR maintains gradient norms around 4×10^3 throughout training, whereas Hard and EROT exhibit norms in the 10^4–10^5 range with substantially larger batch-to-batch fluctuations. Table 3 confirms this across runs: DSR produces gradient norms roughly $3\times$ lower than Hard and EROT, with a $2.5-6\times$ lower within-epoch standard deviation. Lower gradient norms do not in themselves imply a better gradient signal, but combined with the improved performance of DSR, the pattern indicates more stable learning.

Table 3: Proposal gradient norm before clipping on Lorenz-63 (SMNR $= 10$ dB), averaged over runs and epochs.

Method	Mean	Within-epoch std
Hard	17399	17253
Soft	17013	8314
EROT	15771	6918
DSR	**4504**	**2720**

5.3.2 Scaling with particle count

To assess how each method scales with the number of particles, we train on Lorenz-63 at SMNR $= 10$ dB with $N \in \{25, 50, 100, 200\}$, keeping all other hyperparameters fixed. DSR is the only method that trains stably at every tested N: its NMSE improves from -29.94 dB at $N = 25$ to -30.35 dB at $N = 200$ with run-to-run standard deviation below 0.3 dB throughout. Hard and Soft are competitive at $N \leq 100$ but collapse at $N = 200$, with NMSE degrading to -20.42 and -21.52 dB and standard deviations above 7 and 12 dB, respectively, which is caused by training instability. EROT does not complete at $N = 200$: the $\mathcal{O}(N^2)$ transport matrix combined with at most 100 Sinkhorn iterations per resampling step and the autodiff tape over $T = 100$ time steps exceeds available GPU memory. Results and a discussion are reported in Appendix F.3.

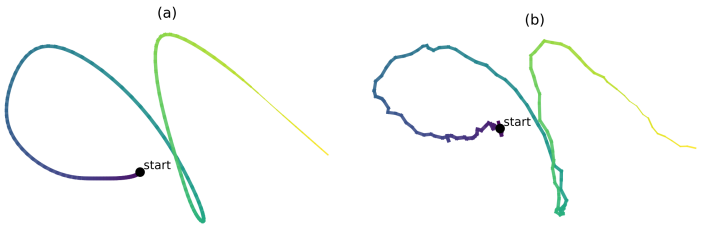


Figure 4: Handwriting dynamics from the CharacterTrajectories dataset. Sample trajectory of the letter 'a' captured at 200 Hz on a WACOM tablet. (a) pen-tip path obtained by integrating the recorded x and y velocities; line thickness shows pen force and colour shows time from start (black dot) to finish. (b) the same after adding noise to the recorded channels at SMNR $= 10$ dB.

5.4 Character typing

To demonstrate applicability to real-world data, we evaluate on the CharacterTrajectories dataset [33] from the UEA multivariate time series archive [1], $2,858$ pen-tip recordings of 20 lower-case characters captured on a WACOM tablet at 200 Hz.

The raw observations $\mathbf{y}_t \in \mathbb{R}^3$ are the pen-tip x and y velocities and the pen force. To give the learned dynamics room to encode temporal information not directly observed, we augment the state to $\mathbf{x}_t = [\mathbf{s}_t^{(1)}, \mathbf{s}_t^{(2)}, \mathbf{s}_t^{(3)}]^\top \in \mathbb{R}^9$ with three 3D latent components, and observe only the last:

$$\mathbf{y}_t = [0_3 \quad 0_3 \quad I_3]\,\mathbf{x}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(0, \mathrm{diag}(\sigma_{w,1}^2, \sigma_{w,2}^2, \sigma_{w,3}^2)), \tag{11}$$

each $\sigma_{w,k}^2$ set by the SMNR of channel k in that recording. The remaining six dimensions are latent and learned end-to-end. We use windows of 178 samples. All recordings of a character have the same length in this release (109–205 samples), so only six characters are long enough (a, m, o, q, w, z): we train on the 526 such recordings among the first $2,000$ and report NMSE on the next 10, all of the character m, with $N = 25$ particles for training and $N = 1000$ for evaluation. A sample trajectory is shown in Fig. 4.

Table 4: State-estimation NMSE on CharacterTrajectories (10 held-out recordings of the character m; $N = 1000$, hard systematic resampling). Results show mean $\pm$ std over 10 runs.

	NMSE (dB)	
Method	SMNR = 10 dB	SMNR = 0 dB
Hard	$-14.65_{\pm 0.20}$	$-7.66_{\pm 0.35}$
Soft	$-14.69_{\pm 0.29}$	$-7.08_{\pm 1.12}$
EROT	$\underline{-14.88}_{\pm 0.17}$	$\underline{-7.77}_{\pm 1.26}$
DSR	$\mathbf{-15.21}_{\pm 0.15}$	$\mathbf{-8.59}_{\pm 0.22}$

The results can be found in Table 4. At 10 dB SMNR, all differentiable methods achieve comparable performance, with DSR obtaining marginally better NMSE (-15.21 dB) than EROT (-14.88 dB) and Soft (-14.69 dB). The non-differentiable Hard baseline (-14.65 dB) is within one standard deviation of Soft, so at SMNR 10 dB, gradient flow through resampling gives at most a small gain here (0.56 dB for DSR over Hard). However, we can observe a larger difference at 0 dB SMNR. DSR achieves -8.59 dB NMSE, which is a 0.82 dB improvement over EROT (-7.77 dB), which corresponds to a 17% lower MSE.

5.5 Training time comparison

Table 5 reports wall-clock training times on Lorenz-63, using NVIDIA T4 GPU. DSR adds only 2.4% over Hard (2.55 h vs. 2.49 h), while EROT is more than 2× slower (5.96 h) because of its Sinkhorn iterations. Although DSR shares the $\mathcal{O}(N^2)$ transport matrix cost with EROT, it avoids the iterative solver: the soft assignment in Eq. (5) is a single pass of sigmoid evaluations, implemented as batched matrix operations, hence the nested loops in Algorithm 3 do not appear in practice. This brings DSR's per-resampling cost close to that of a discrete systematic resampling step.

Table 5: Training time on Lorenz-63.

Method	Total (h)	Time/epoch (s)
Hard	$\mathbf{2.49}_{\pm 0.15}$	$\mathbf{17.94}_{\pm 1.11}$
Soft	$4.33_{\pm 0.05}$	$31.16_{\pm 0.39}$
EROT	$5.96_{\pm 0.11}$	$42.94_{\pm 0.77}$
DSR	$\underline{2.55}_{\pm 0.02}$	$\underline{18.39}_{\pm 0.16}$

The convex combination limitation. At any finite τ, DSR produces convex combinations of ancestor particles rather than exact copies, which may place mass between modes when the posterior is multimodal; the same property applies to EROT [5]. DSR's CDF-ordered structure partially mitigates this: Morton sorting makes soft assignments mix mostly nearby particles, which limits cross-mode mixing when the modes differ in the leading coordinates of the Morton key. We quantify this on a static bimodal Gaussian in Appendix G (separating distributions along first dimension): without Morton sorting every resampled particle lands between the modes, whereas with Morton sorting the inter-modal mass drops to ≈ 0.28 and remains stable up to dimension 10, while EROT places most mass between the modes for $d \geq 3$. Nonetheless, multimodal posteriors remain challenging for particle filters in general [32, 30], and a fuller treatment of differentiable resampling under such posteriors is an interesting direction for future work.

6 Conclusion

In this paper we have introduced Differentiable Systematic Resampling (DSR), a principled and computationally efficient method for enabling gradient-based learning in particle filters. DSR is a soft relaxation of systematic resampling, where a parameter is introduced that controls the tradeoff between gradient flow and exactness of systematic resampling. Unlike existing differentiable resampling methods, DSR preserves the CDF-ordered, banded structure of systematic resampling while enabling full gradient flow through the particle filtering recursion. Our experimental results on both synthetic chaotic systems (Lorenz-63) and real-world handwriting dynamics (CharacterTrajectories) demonstrate that DSR achieves the best state estimation accuracy in every setting we tested and the best or near-best dynamics learning quality, with the largest gains at low signal-to-noise ratios.

7 Acknowledgments

The research is supported by funding and activities from Digital Futures Center, European Defence Fund REACT II project, Saab, Ericsson, VINNOVA and partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP). Computational resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), funded by the Swedish Research Council.

References

[1] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018.

[2] Itay Buchnik, Guy Revach, Damiano Steger, Ruud J. G. van Sloun, Tirza Routtenberg, and Nir Shlezinger. Latent-kalmannet: Learned kalman filtering for tracking from high-dimensional signals. *Trans. Sig. Proc.*, 72:352–367, January 2024.

[3] Xiongjie Chen and Yunpeng Li. Normalizing flow-based differentiable particle filters. *IEEE Transactions on Signal Processing*, 73:493–507, 2025.

[4] Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer Series in Statistics. Springer, 2020.

[5] Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *International Conference on Machine Learning*, pages 2100–2111. PMLR, 2021.

[6] Domonkos Csuzdi, Olivér Törő, and Tamás Bécsi. Differentiable particle filtering using optimal placement resampling. In *2024 IEEE 18th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 000183–000188, 2024.

[7] Fredrik Cumlin, Anubhab Ghosh, and Saikat Chatterjee. Dns: Data-driven nonlinear smoother for complex model-free process. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 356–360, 2026.

[8] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69. IEEE, 2005.

[9] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *An Introduction to Sequential Monte Carlo Methods*, pages 3–14. Springer New York, New York, NY, 2001.

[10] Arnaud Doucet and Adam M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In Dan Crisan and Boris Rozovskii, editors, *The Oxford handbook of nonlinear filtering*, Oxford handbooks in mathematics, pages 656–705. Oxford University Press, Oxford ; N.Y., 2011.

[11] Mathieu Gerber, Nicolas Chopin, and Nick Whiteley. Negative association, ordering and convergence of resampling methods. *The Annals of Statistics*, 47(4):2236 – 2260, 2019.

[12] Anubhab Ghosh, Yonina C. Eldar, and Saikat Chatterjee. Particle-based data-driven nonlinear state estimation of model-free process from nonlinear measurements. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025.

[13] Anubhab Ghosh, Antoine Honoré, and Saikat Chatterjee. DANSE: Data-Driven Non-Linear State Estimation of Model-Free Process in Unsupervised Learning Setup. *IEEE Transactions on Signal Processing*, 72:1824–1838, 2024.

[14] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113, 1993.

[15] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[16] Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. *Robotics: Science and Systems XIV*, 2018.

[17] Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter networks with application to visual localization. In *Proceedings of The 2nd Conference on Robot Learning*, pages 169–178. PMLR, 2018.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations (ICLR)*, 2014.

[20] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[21] Alina Kloss, Georg Martius, and Jeannette Bohg. How to train your differentiable filter. *Autonomous Robots*, 45:561–578, 2021.

[22] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential Monte Carlo. In *International Conference on Learning Representations*, 2018.

[23] Chris J. Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh. Filtering variational objectives. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6576–6586, Red Hook, NY, USA, 2017. Curran Associates Inc.

[24] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[25] G. M. Morton. A computer oriented geodetic data base and a new technique in file sequencing. Technical report, International Business Machines Company, 1966.

[26] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International conference on artificial intelligence and statistics*, pages 968–977. PMLR, 2018.

[27] Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön. Elements of sequential monte carlo. *Found. Trends Mach. Learn.*, 12(3):307–392, November 2019.

[28] Gustav Norén, Anubhab Ghosh, Fredrik Cumlin, and Saikat Chatterjee. Vse: Variational state estimation of complex model-free process. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4921–4925, 2026.

[29] Simo Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[30] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1):99–141, 2001.

[31] Wessel L. van Nierop, Nir Shlezinger, and Ruud J.G. van Sloun. Deep variational sequential monte carlo for high-dimensional observations. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025.

[32] Vermaak, Doucet, and Perez. Maintaining multimodality through mixture tracking. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1110–1116 vol.2, 2003.

[33] Ben Williams. Character Trajectories. UCI Machine Learning Repository, `https://doi.org/10.24432/C58G7V`, 2006.

A Differentiable Systematic Resampling Algorithm

In this section, we explain the DSR algorithm. Note that `MortonSort` is used by default, but is typically optional when the filtering posterior is unimodal, as particle ordering has less effect on the soft assignments in this regime (see Appendix F.4). Also note that DSR in Alg. 3 can be implemented using batched matrix operations, hence avoiding explicit loops over particles.

Algorithm 2 `MortonSort`: Morton Space-Filling Curve Sort.

Require: Particles $\{\mathbf{x}_i\}_{i=1}^N$ with $\mathbf{x}_i \in \mathbb{R}^{d_x}$, resolution b bits

for $i = 1$ to N **do**
  for $d = 1$ to d_x **do**
    $c_i^{(d)} = \left\lfloor \frac{x_i^{(d)} - x_{\min}^{(d)}}{x_{\max}^{(d)} - x_{\min}^{(d)}} \cdot (2^b - 1) \right\rfloor$ ▷ normalize to integer grid
  end for
  Write each $c_i^{(d)}$ in binary: $c_i^{(d)} = \sum_{k=0}^{b-1} c_{i,k}^{(d)}\, 2^k$ with $c_{i,k}^{(d)} \in \{0, 1\}$
  $z_i = \sum_{k=0}^{b-1} \sum_{d=1}^{d_x} c_{i,k}^{(d)} \cdot 2^{k \cdot d_x + (d_x - d)}$ ▷ interleave bits across dimensions
end for
$\pi = \mathrm{argsort}(z_1, \ldots, z_N)$
return $\{\mathbf{x}_{\pi(i)}\}_{i=1}^N, \pi$

Algorithm 3 Differentiable Systematic Resampling (DSR). Equation marked in green is the DSR assignment step, and equation marked in red is the SR assignment step.

Require: Weights $\{w_i\}_{i=1}^N$, particles $\{\mathbf{x}_i\}_{i=1}^N$, temperature $\tau > 0$, resolution b

$\{\mathbf{x}_i\}_{i=1}^N, \pi \leftarrow$ `MortonSort`$(\{\mathbf{x}_i\}_{i=1}^N, b)$; reorder $\{w_i\}$ accordingly
Compute CDF: $F_0 = 0$, $F_i = \sum_{k=1}^{i} w_k$ for $i = 1, \ldots, N$
Draw $u_0 \sim \mathrm{Uniform}(0, \frac{1}{N})$
for $j = 1$ to N **do**
  $u_j = u_0 + \frac{j-1}{N}$
  for $i = 1$ to N **do**
    $s_i = \mathbb{1}(F_{i-1} < u_j \leq F_i)$ ▷ systematic resampling
    $s_i(u_j; \tau) = \sigma\left(\frac{u_j - F_{i-1}}{\tau}\right) - \sigma\left(\frac{u_j - F_i}{\tau}\right)$
  end for
  $T_{ij} = s_i(u_j; \tau) \big/ \sum_{k=1}^N s_k(u_j; \tau)$
  $\tilde{\mathbf{x}}_j = \sum_{i=1}^N T_{ij} \mathbf{x}_i$
end for
return $\{\tilde{\mathbf{x}}_j\}_{j=1}^N$

B Proofs of propositions

Proof of Proposition 4.1. We prove both parts of the proposition.

Part 1: Convergence of the soft transport plan.

Fix a column index $j \in [N]$ and define the hard systematic resampling index

$$a_j^* := \min\{i \in [N] : F_i \geq u_j\}.$$

By construction, we have

$$F_{a_j^* - 1} < u_j \leq F_{a_j^*}.$$

Since we can view u_j as a realization from a continuous distribution, the event $u_j = F_i$ occurs with probability zero and is ignored in the almost sure sense.

Recall that

$$s_i(u_j; \tau) = \sigma\left(\frac{u_j - F_{i-1}}{\tau}\right) - \sigma\left(\frac{u_j - F_i}{\tau}\right),$$

where $\sigma(x) = (1 + e^{-x})^{-1}$.

We analyze the limit as $\tau \to 0^+$.

Case 1: $i = a_j^*$. Since $u_j > F_{a_j^*-1}$ and $u_j < F_{a_j^*}$ almost surely, we have

$$\frac{u_j - F_{a_j^*-1}}{\tau} \to +\infty, \qquad \frac{u_j - F_{a_j^*}}{\tau} \to -\infty.$$

Therefore,

$$\lim_{\tau \to 0^+} s_{a_j^*}(u_j; \tau) = 1 - 0 = 1 \quad \text{a.s.}$$

Case 2: $i < a_j^*$. Then $F_i < u_j$ and $F_{i-1} < u_j$, implying both arguments of σ tend to $+\infty$ as $\tau \to 0^+$. Hence,

$$\lim_{\tau \to 0^+} s_i(u_j; \tau) = 1 - 1 = 0.$$

Case 3: $i > a_j^*$. Then $F_{i-1} \geq F_{a_j^*} > u_j$, so both arguments tend to $-\infty$, yielding

$$\lim_{\tau \to 0^+} s_i(u_j; \tau) = 0 - 0 = 0.$$

Combining all cases, we obtain

$$\lim_{\tau \to 0^+} s_i(u_j; \tau) = \begin{cases} 1, & i = a_j^*, \\ 0, & i \neq a_j^*, \end{cases} \quad \text{a.s.}$$

Since $\sum_{k=1}^N s_k(u_j; \tau) \to 1$ almost surely, it follows that

$$\lim_{\tau \to 0^+} T_{ij}^{(\tau)} = \lim_{\tau \to 0^+} \frac{s_i(u_j; \tau)}{\sum_{k=1}^N s_k(u_j; \tau)} = \mathbb{1}(i = a_j^*) \quad \text{a.s.}$$

Part 2: Convergence of the resampled particles.

Using the result above,

$$\begin{aligned} \lim_{\tau \to 0^+} \tilde{\mathbf{x}}_j^{(\tau)} &= \lim_{\tau \to 0^+} \sum_{i=1}^N T_{ij}^{(\tau)} \mathbf{x}_i \\ &= \sum_{i=1}^N \left(\lim_{\tau \to 0^+} T_{ij}^{(\tau)} \right) \mathbf{x}_i \\ &= \mathbf{x}_{a_j^*} \\ &= \tilde{\mathbf{x}}_j^*, \end{aligned}$$

where the exchange of limit and summation is justified since the sum is finite and $T_{ij}^{(\tau)} \in [0, 1]$ for all $\tau > 0$. This completes the proof. □

Proof of Proposition 4.2. Recall that

$$\tilde{\mathbf{x}}_j^{(\tau)} = \sum_{i=1}^N T_{ij}^{(\tau)} \mathbf{x}_i, \qquad T_{ij}^{(\tau)} = \frac{s_i(u_j; \tau)}{\sum_{k=1}^N s_k(u_j; \tau)}.$$

Let

$$\Delta_j := \min\{u_j - F_{a_j^*-1},\, F_{a_j^*} - u_j\} > 0 \quad \text{a.s.}$$

Since $\sum_{i=1}^N T_{ij}^{(\tau)} = 1$, we have $\mathbf{x}_{a_j^*} = \sum_{i=1}^N T_{ij}^{(\tau)} \mathbf{x}_{a_j^*}$. Thus

$$\tilde{\mathbf{x}}_j^{(\tau)} - \mathbf{x}_{a_j^*} = \sum_{i=1}^N T_{ij}^{(\tau)} \mathbf{x}_i - \sum_{i=1}^N T_{ij}^{(\tau)} \mathbf{x}_{a_j^*} = \sum_{i=1}^N T_{ij}^{(\tau)} (\mathbf{x}_i - \mathbf{x}_{a_j^*}).$$

Taking norms and using the triangle inequality yields

$$\|\tilde{\mathbf{x}}_j^{(\tau)} - \mathbf{x}_{a_j^*}\|_2 \leq \sum_{i=1}^{N} T_{ij}^{(\tau)} \|\mathbf{x}_i - \mathbf{x}_{a_j^*}\|_2 \leq D \sum_{i \neq a_j^*} T_{ij}^{(\tau)},$$

where D is the diameter, $\sup_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq D$, and we have used the obvious equality $T_{a_j^* j}\|\mathbf{x}_{a_j^*} - \mathbf{x}_{a_j^*}\|_2 = 0$.

It therefore suffices to bound $\sum_{i \neq a_j^*} T_{ij}^{(\tau)}$.

Lower bound on the dominant term. Using the inequalities $\sigma(x) \geq 1 - e^{-x}$ for $x \geq 0$ and $-\sigma(x) \geq -e^{x}$ for $x \leq 0$, we obtain

$$s_{a_j^*}(u_j; \tau) = \sigma\left(\frac{u_j - F_{a_j^*-1}}{\tau}\right) - \sigma\left(\frac{u_j - F_{a_j^*}}{\tau}\right) \geq 1 - e^{-\Delta_j/\tau} - e^{-\Delta_j/\tau}.$$

Thus, for all $\tau \leq \Delta_j / \log 4$,

$$s_{a_j^*}(u_j; \tau) \geq \tfrac{1}{2}.$$

Upper bound on the remaining terms. Fix $i \neq a_j^*$. If $i < a_j^*$, then $F_i \leq F_{a_j^*-1} < u_j$, so both arguments of the sigmoids are positive and

$$s_i(u_j; \tau) = \sigma\left(\frac{u_j - F_{i-1}}{\tau}\right) - \sigma\left(\frac{u_j - F_i}{\tau}\right) \leq 1 - \sigma\left(\frac{u_j - F_{a_j^*-1}}{\tau}\right).$$

Using $1 - \sigma(x) \leq e^{-x}$ for $x \geq 0$, we obtain

$$s_i(u_j; \tau) \leq \exp\left(-\frac{u_j - F_{a_j^*-1}}{\tau}\right).$$

If $i > a_j^*$, then $F_{i-1} \geq F_{a_j^*} > u_j$, so both arguments are negative and

$$s_i(u_j; \tau) \leq \sigma\left(\frac{u_j - F_{a_j^*}}{\tau}\right) \leq \exp\left(-\frac{F_{a_j^*} - u_j}{\tau}\right),$$

where we used $\sigma(x) \leq e^{x}$ for $x \leq 0$.

In both cases, we obtain the uniform bound

$$s_i(u_j; \tau) \leq \exp\left(-\frac{\Delta_j}{\tau}\right).$$

Hence,

$$\sum_{i \neq a_j^*} s_i(u_j; \tau) \leq N \exp\left(-\frac{\Delta_j}{\tau}\right).$$

As $s_{a_j^*}(u_j; \tau) \geq \frac{1}{2}$, it follows that $\sum_{k=1}^{N} s_k(u_j; \tau) \geq \frac{1}{2}$. Combining the bounds yields

$$\sum_{i \neq a_j^*} T_{ij}^{(\tau)} = \frac{\sum_{i \neq a_j^*} s_i(u_j; \tau)}{\sum_{k=1}^{N} s_k(u_j; \tau)} \leq \frac{N e^{-\Delta_j/\tau}}{1/2} = 2N \exp\left(-\frac{\Delta_j}{\tau}\right).$$

Finally,

$$\|\tilde{\mathbf{x}}_j^{(\tau)} - \mathbf{x}_{a_j^*}\|_2 \leq 2DN \exp\left(-\frac{\Delta_j}{\tau}\right),$$

which establishes exponential convergence as $\tau \to 0^+$. □

C Architecture and training configuration

C.1 Network Architectures

All experiments use multilayer perceptrons (MLPs) with ReLU activations to parameterize the transition and proposal models. Both models predict Gaussian distributions with diagonal covariance matrices.

Transition Model. The transition model $f_{\boldsymbol{\phi}}(\mathbf{x}_t)$ predicts $p_{\boldsymbol{\phi}}(\mathbf{x}_{t+1}|\mathbf{x}_t)$ and consists of:

- Shared trunk: Two fully-connected layers with hidden dimension h and ReLU activations
- Mean head: Linear layer mapping from hidden dimension to state dimension d_x
- Covariance head: Linear layer mapping from hidden dimension to d_x, followed by softplus activation to ensure positive diagonal entries

The model outputs $(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_t), \boldsymbol{\Lambda}_{\boldsymbol{\phi}}(\mathbf{x}_t))$ where $\boldsymbol{\Lambda}_{\boldsymbol{\phi}}$ is a diagonal covariance matrix.

Proposal Model. For simplicity, the proposal model $q_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ has the same architecture as the transition model, but takes concatenated inputs $[\mathbf{x}_{t-1}, \mathbf{y}_t]$.

C.2 Hyperparameters

Linear Gaussian SSM. The transition model is parameterized as a learnable matrix $\hat{\mathbf{A}} \in \mathbb{R}^{4\times4}$ with a learnable diagonal covariance, and the proposal is a second, separately learned linear-Gaussian model of the same form. The true transition matrix is $\mathbf{A} = 0.9 \cdot \mathrm{blockdiag}(\mathbf{R}(\theta_1), \mathbf{R}(\theta_2))$ with $\theta_1 = 0.3$ and $\theta_2 = 0.5$, where $\mathbf{R}(\theta)$ is the 2×2 rotation matrix. The process noise covariance is $\mathbf{C}_e = 0.1 \cdot \mathbf{I}_4$ and observations are full-state with $\mathbf{H} = \mathbf{I}_4$. Training uses the Adam optimizer with learning rate 10^{-3}, batch size 64, weight decay 0, and gradient clipping at norm 5.0. We train with $N = 25$ particles for 500 epochs on 1000 trajectories of length $T = 100$ at SMNR $= 10$ dB. For DSR, we set $\tau = 0.1$. We clamp the log-likelihood values below by -50 and -100 for 10 dB and 0 dB SMNR, respectively. This avoids extreme instability during training; none of the methods converges without this. For DSR and EROT, the gradient is clipped elementwise to $[-1, 1]$ per resampling step. EROT uses Sinkhorn with 100 iterations, with early stopping when the difference is less than 10^{-3}

Lorenz-63. We use $h = 32$ for both models. The transition model has input dimension 3 (state dimension), while the proposal model has input dimension $3 + 64 = 67$ (state plus observation). Training uses the Adam optimizer with learning rate 5×10^{-4}, batch size 64, weight decay 0, and gradient clipping at norm 5.0. We train with $N = 25$ particles for 500 epochs on 1000 trajectories of length $T = 100$. For DSR, we set $\tau = 0.1$ across all experiments, and use 20 bits for the Morton curves.

Character Typing. We use a hidden dimension of $h = 32$. The augmented state dimension is $d_x = 9$, with direct observations of the last 3 dimensions ($d_y = 3$). Training uses 25 particles, learning rate 1×10^{-3}, and batch size 64 on 526 training sequences of length $T = 178$. All other hyperparameters match Lorenz-63.

Evaluation. We evaluate the state estimation performance using the same training setup except for the resampling method, which uses hard systematic resampling (as this resampling method is typically used in practice). We use $N = 1000$ particles, $T = 1000$ (linear and Lorenz-63) or 10 trajectories at $T = 178$ (CharacterTrajectories), and report the values after the last epoch.

D SMNR and Evaluation Measures

D.1 SMNR

The signal-to-measurement noise ratio is given by

$$\mathrm{SMNR} = 10 \log_{10} \left(\frac{\sum_{t=1}^{T} \mathbb{E}\left[\left\| \mathbf{h}(\mathbf{x}_t^{(j)}) - \mathbb{E}[\mathbf{h}(\mathbf{x}_t^{(j)})] \right\|_2^2 \right]}{T d_y \sigma_w^2} \right), \tag{12}$$

where $\mathbf{h}(\mathbf{x}_t^{(j)})$ is the noiseless observation, σ_w^2 is the observation noise variance per dimension, and d_y is the observation dimension. A low SMNR corresponds to high measurement noise.

D.2 State Estimation

Normalized Mean Squared Error (NMSE). We evaluate state estimation accuracy using NMSE in dB:

$$\text{NMSE} = 10 \log_{10} \left(\frac{\frac{1}{T} \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2}{\frac{1}{T} \sum_{t=1}^{T} \|\mathbf{x}_t\|_2^2} \right), \tag{13}$$

where $\mathbf{x}_t$ is the true state, $\hat{\mathbf{x}}_t$ is the estimated state (the posterior mean of the particle filter), and T is the trajectory length. Lower values indicate better performance.

D.3 Transition Model Quality

One-step Mean Squared Error (MSE). Measures the quality of the learned dynamics f_θ by predicting one step ahead from true states:

$$\text{MSE} = \frac{1}{T-1} \sum_{t=1}^{T-1} \|\mathbf{x}_{t+1} - f_\theta(\mathbf{x}_t)\|_2^2. \tag{14}$$

Lower values indicate better transition model quality.

Multi-step Mean Squared Error (MSE-5-step). Evaluates prediction over a longer horizon by rolling out the learned dynamics for 5 steps:

$$\text{MSE-5-step} = \frac{1}{T-5} \sum_{t=1}^{T-5} \|\mathbf{x}_{t+5} - f_\theta^{(5)}(\mathbf{x}_t)\|_2^2, \tag{15}$$

where $f_\theta^{(5)}(\mathbf{x}_t) = f_\theta(f_\theta(f_\theta(f_\theta(f_\theta(\mathbf{x}_t)))))$ denotes 5 iterative applications of the learned dynamics. For each use of the transition model, the mean is used as input to the subsequent call of the model.

Log-Likelihood (LL). Evaluates the probabilistic quality of the learned transition model:

$$\text{LL} = \frac{1}{T-1} \sum_{t=1}^{T-1} \log p_\theta(\mathbf{x}_{t+1}|\mathbf{x}_t), \tag{16}$$

where $p_\theta(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t), \boldsymbol{\Lambda}_\theta(\mathbf{x}_t))$ is the learned transition distribution, where $f_{\boldsymbol{\theta}}(\mathbf{x}_t) = (\boldsymbol{\mu}_\theta(\mathbf{x}_t), \boldsymbol{\Lambda}_\theta(\mathbf{x}_t))$. Higher values indicate better probabilistic modeling.

KL Divergence (KLD). Measures the distributional mismatch between the learned dynamics and true dynamics:

$$\text{KLD} = \frac{1}{T-1} \sum_{t=1}^{T-1} D_{\text{KL}}\big(p_\phi(\mathbf{x}_{t+1} \mid \mathbf{x}_t) \,\|\, p_{\text{true}}(\mathbf{x}_{t+1} \mid \mathbf{x}_t)\big). \tag{17}$$

Lower values indicate a closer match to the true transition distribution.

E Empirical Analysis of the DSR Relaxation

The theoretical results in Section 4 establish that DSR converges to systematic resampling as $\tau \to 0^+$, with pointwise exponential rates. In this section, we complement the theory with an empirical study of the DSR transport plan and the resampled particles. We use a sequential importance resampling (SIR) particle filter with systematic resampling [10] on the Lorenz-63 system with oracle (true) dynamics, at 10 dB SMNR with the Gaussian point spread observation map. This isolates the resampling relaxation from the dynamics learning, so we can study DSR as an approximation to systematic resampling directly. We vary $\tau \in \{0.01, 0.05, 0.1, 0.5, 1.0\}$ and $N \in \{25, 50, 100, 200\}$, and report results averaged over 10 sequences of length $T = 100$.

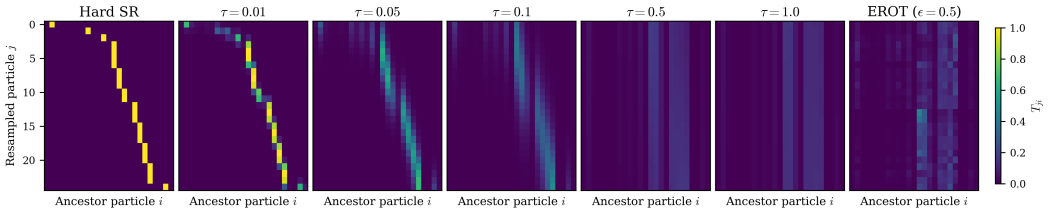


Figure 5: Transportation maps for hard systematic resampling and DSR at varying temperatures $\tau \in \{0.01, 0.05, 0.1, 0.5, 1.0\}$, alongside EROT ($\varepsilon = 0.5$), for a SIR particle filter of the Lorenz-63 system with $N = 25$ particles. DSR is shown without Morton sorting (the filter's particle order).

E.1 Transport Plan Visualization

Figure 5 shows the DSR transportation matrices $T^{(\tau)}$ for varying τ, alongside hard systematic resampling and EROT ($\varepsilon = 0.5$), at a representative time step with $N = 25$ particles. At $\tau = 0.01$, the DSR transport plan closely resembles the sparse, block-diagonal structure of hard systematic resampling, with mass concentrated on a single ancestor per resampled particle. As τ grows, the plan becomes more diffuse and spreads mass across neighbouring ancestors. At $\tau = 0.5$, the plan becomes visibly banded: a high-weight ancestor contributes to many resampled particles, so DSR becomes less restrictive about spatial locality. This brings DSR qualitatively closer to EROT. A main difference is that at $\tau = 0.5$ DSR's columns are flatter: each resampled particle mixes more ancestors than under EROT ($N_{eff} = 11.2$ vs. 9.8 for $N = 25$, Appendix E.2), which makes DSR less local. A lower column-wise spread implies that the resampling linear combination has more even weights, which in turn becomes less restrictive to spatial locality. On the other hand, EROT's transport plan bears little resemblance to systematic resampling overall, whereas DSR retains the CDF-ordered banded structure even at high τ.

E.2 Effective Number of Ancestors

To quantify the sharpness of the transport plan, we compute the effective number of ancestors for each resampled particle j:

$$N_{\text{eff},j} = \exp\left(-\sum_{i=1}^{N} T_{ij} \log T_{ij}\right),$$

averaged over particles and time steps. For hard systematic resampling, $N_{\text{eff},j} = 1$, as each resampled particle copies exactly one ancestor.

Figure 6a shows N_{eff} as a function of τ for different particle counts. At low temperatures, the DSR transport plan is sharply concentrated: $N_{\text{eff}} = 1.70$ at $\tau = 0.01$ with $N = 25$, close to the hard SR limit. N_{eff} grows with τ up to $\tau = 0.5$ and then levels off ($N = 25$: 11.2 at $\tau = 0.5$, 10.6 at $\tau = 1.0$); for fixed τ it scales with N. This is consistent with the N-dependent factor in Proposition 4.2: as N grows, the CDF intervals $[F_{i-1}, F_i]$ become narrower, so more ancestors fall within the sigmoid bandwidth τ and contribute to each resampled particle.

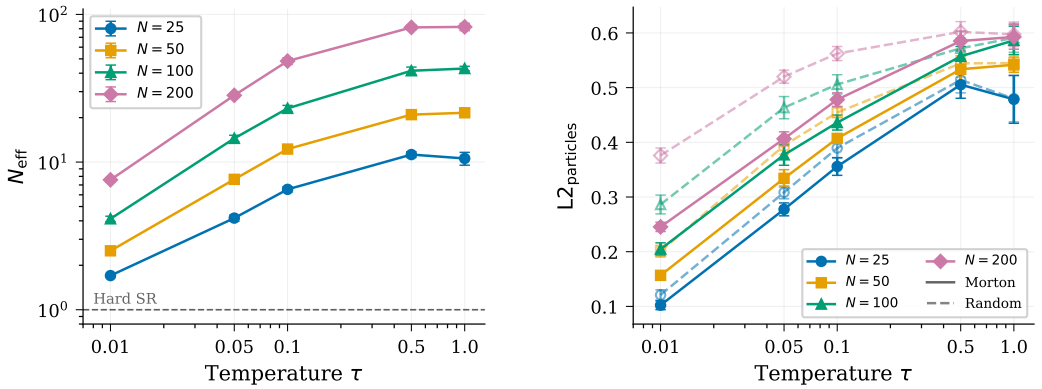
For comparison, EROT ($\varepsilon = 0.5$) yields N_{eff} of 9.83, 18.36, 36.41, and 71.77 for $N = 25, 50, 100, 200$. For DSR using the default value $\tau = 0.1$, DSR is substantially sparser than EROT across all particle counts (e.g., 6.54 vs. 9.83 for $N = 25$; 23.11 vs. 36.41 for $N = 100$).

E.3 L2 Distance and Effect of Morton Sorting

We measure the per-particle L2 distance between DSR and hard SR resampled particles:

$$\text{L2}_{\text{particles}} = \frac{1}{N} \sum_{j=1}^{N} \|\tilde{\mathbf{x}}_j^{(\tau)} - \mathbf{x}_{a_j^*}\|_2,$$

where $\mathbf{x}_{a_j^*}$ is the resampled particle by systematic resampling. This directly relates to the bound in Proposition 4.2. Further, to assess the role of particle ordering, we compare Morton space-filling



(a) Effective number of ancestors N_{eff} as a function of temperature τ for varying particle counts N. The dashed line indicates hard systematic resampling ($N_{\text{eff}} = 1$).

(b) Per-particle L2 distance between DSR and hard SR resampled particles. Solid lines use Morton sorting; dashed lines use the filter's unsorted particle order.

Figure 6: Empirical analysis of the DSR relaxation on the Lorenz-63 system with oracle dynamics at SMNR $= 10$ dB. Error bars show ± 1 standard deviation over 10 sequences.

curve sorting (used in all VSMC experiments) against the unsorted order the filter produces (inherited from the previous resampling step).

Figure 6b shows the L2 distance as a function of τ for both orderings. As suggested in Proposition 4.2, the distance grows with τ under Morton sorting: for $N = 25$, it increases from 0.10 at $\tau = 0.01$ to 0.51 at $\tau = 0.5$. Morton sorting consistently reduces the L2 distance relative to the unsorted order, and the gap is largest at low τ and high N. At $\tau = 0.01$ with $N = 200$, Morton sorting reduces the distance from 0.38 to 0.25, a 35% reduction. The reason is that at low τ, each resampled particle draws mass primarily from its CDF neighbours, and Morton sorting ensures those neighbours are also spatially close, keeping the mixture near the hard ancestor. At higher temperatures ($\tau \geq 0.5$), the plan is diffuse enough that ordering matters less.

E.4 Bias and Variance of DSR Estimates

Here we study the bias and variance of DSR. For resampling schemes, a typical study is the study of how $I = \sum_i w_i f(\mathbf{x}_i)$ (f is an arbitrary function) changes after resampling, in particular, studying how I behaves after resampling. To quantify the effect, we measure the bias, variance, and MSE of a test-function f estimate under each scheme. Fixing a particle set and weights, over 20000 seeds we compute $\hat{I} = \sum_j \tilde{w}_j f(\tilde{\mathbf{x}}_j)$ with equal weights $\tilde{w}_j = 1/N$ after resampling, and compare against the weighted estimate $I = \sum_i w_i f(\mathbf{x}_i)$. We use $N = 100$ particles with $\mathbf{x}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_3)$ and w_j the likelihood, and two test functions, $f(\mathbf{x}) = \mathbf{x}$ and $f(\mathbf{x}) = \mathbf{x}^2$ (elementwise). Hard systematic and multinomial resampling are unbiased, so their reported bias is Monte-Carlo error. Results are in Table 6.

For the linear function $f(\mathbf{x}) = \mathbf{x}$, DSR trades variance for bias at every τ: the variance is 1.6 to 5 orders of magnitude below hard systematic resampling, since a convex combination is an average, and DSR attains lower MSE at every τ. Here the bias is non-monotone in τ, because both limits are favourable for a linear function: $\tau \to 0^+$ recovers systematic resampling, while large τ drives every output toward the weighted mean, which estimates a linear function exactly (here $I \approx 0$, so the contraction at $\tau = 1.0$ is favourable and yields the lowest MSE). This should not be read as $\tau = 1.0$ being preferable in general, since the same contraction degrades nonlinear functions: for $f(\mathbf{x}) = \mathbf{x}^2$ the bias increases monotonically with τ. The variance reduction comes from averaging: every output is a convex combination of ancestors.

Table 6: Bias, variance, and MSE of resampling estimates against the weighted estimate I, over 20000 seeds with $N = 100$. Hard SR and multinomial are unbiased (their bias is Monte-Carlo error).

	$f(\mathbf{x}) = \mathbf{x}$			$f(\mathbf{x}) = \mathbf{x}^2$		
Method	bias	var	MSE	bias	var	MSE
Hard SR	2.3×10^{-4}	1.2×10^{-3}	1.2×10^{-3}	1.2×10^{-4}	7.0×10^{-4}	7.0×10^{-4}
Multinomial	4.9×10^{-4}	5.5×10^{-3}	5.5×10^{-3}	2.8×10^{-4}	2.6×10^{-3}	2.6×10^{-3}
DSR ($\tau = 0.01$)	2.4×10^{-3}	2.9×10^{-5}	3.4×10^{-5}	1.2×10^{-1}	3.8×10^{-7}	1.4×10^{-2}
DSR ($\tau = 0.05$)	1.2×10^{-2}	8.6×10^{-6}	1.6×10^{-4}	2.3×10^{-1}	3.1×10^{-7}	5.3×10^{-2}
DSR ($\tau = 0.1$)	2.0×10^{-2}	4.7×10^{-6}	3.9×10^{-4}	2.7×10^{-1}	2.0×10^{-7}	7.3×10^{-2}
DSR ($\tau = 0.5$)	1.2×10^{-2}	1.6×10^{-7}	1.4×10^{-4}	3.1×10^{-1}	2.2×10^{-9}	9.9×10^{-2}
DSR ($\tau = 1.0$)	3.6×10^{-3}	1.3×10^{-8}	1.3×10^{-5}	3.2×10^{-1}	2.0×10^{-10}	1.0×10^{-1}

E.5 Discussion

The empirical analysis supports the theoretical predictions of Section 4. The DSR transport plan converges to the sparse structure of systematic resampling as $\tau \to 0^+$, with $N_{\text{eff}} \to 1$ at low temperatures. At the default value $\tau = 0.1$, DSR is substantially sparser than EROT (N_{eff} 6.5 vs. 9.8 for $N = 25$) and puts on average 35% of each resampled particle's weight on its hard SR ancestor ($N = 25$). Morton sorting reduces the approximation error, particularly at low τ and high N, confirming that locality-preserving orderings complement the DSR relaxation. Together, these results show that DSR approximates systematic resampling closely at small τ, and how the approximation loosens as τ and N grow.

F Ablation Studies

F.1 Dependence on τ

We study the effect of the temperature parameter τ on filtering performance in the Lorenz-63 dataset at SMNR = 10 dB. Table 7 reports NMSE results for different temperature values.

Table 7: Ablation study of the temperature parameter τ on Lorenz-63 (SMNR = 10 dB).

Temperature τ	NMSE (dB)
0.01	-28.31 ± 0.15
0.05	-29.56 ± 0.34
0.10	-29.94 ± 0.28
0.50	$\mathbf{-30.21 \pm 0.29}$
1.00	-29.91 ± 0.28

At $\tau = 0.01$, NMSE is notably worse (-28.31 dB) than at larger temperatures. This is consistent with the $\mathcal{O}(1/\tau)$ scaling of the sigmoid derivative,

$$\frac{d}{dx}\sigma(x) = \frac{1}{\tau}\sigma(x/\tau)\big(1 - \sigma(x/\tau)\big),$$

which causes gradient magnitudes to explode during backpropagation at small τ, leading to unstable optimization. As τ grows, gradient magnitudes become more moderate and training stabilizes. Performance plateaus for $\tau \in [0.05, 1.0]$, with NMSE varying by less than 0.7 dB across this range and consistently low run-to-run variance. Based on these results, we fix $\tau = 0.1$ for all experiments; this is a conservative choice within the stable range that keeps the soft transport plan close to hard systematic resampling (see Appendix E).

Beyond NMSE, the temperature τ governs a bias–variance tradeoff in the gradient itself. Table 8 reports, for the same setup, the gradient-norm statistics and the per-particle L_2 distance of the DSR transport to hard systematic resampling (Appendix E) as a function of τ. The two quantities move in opposite directions: as τ shrinks, the approximation error L_2 decreases ($0.505 \to 0.102$) while

the median gradient norm and within-epoch standard deviation increase. The ratio of median within-epoch std to median gradient norm stays close to 0.45 for $\tau \geq 0.05$, so gradients grow but remain comparably well behaved; at $\tau = 0.01$ the ratio rises to 0.55, indicating larger variance relative to norm and less stable training, consistent with the degraded NMSE at $\tau = 0.01$. We read $\tau = 0.01$ as a change of regime rather than of scale.

Table 8: Gradient behaviour and approximation error as a function of τ on Lorenz-63 (SMNR $= 10$ dB). The L_2 column is the per-particle distance of the DSR transport to hard systematic resampling from the same particle set and offset (Appendix E).

		Gradient norm		Within-epoch std		
τ	NMSE (dB)	Median	Mean	Median	Mean	L_2 to hard SR
0.01	$-28.31_{\pm 0.15}$	14842	2.39×10^5	8163	4.35×10^5	$0.102_{\pm 0.009}$
0.05	$-29.56_{\pm 0.34}$	6457	7228	3009	3676	$0.277_{\pm 0.012}$
0.10	$-29.94_{\pm 0.28}$	4174	4504	1885	2720	$0.356_{\pm 0.017}$
0.50	$\mathbf{-30.21}_{\pm 0.29}$	2546	2951	1155	1985	$0.505_{\pm 0.026}$
1.00	$-29.91_{\pm 0.28}$	2306	3041	1027	3453	$0.479_{\pm 0.046}$

F.2 Annealing τ during training

The temperature is fixed at $\tau = 0.1$ in all main experiments. Here we test annealing τ over the course of training: starting at a larger τ for smoother gradients and decreasing it toward a sharper, more faithful assignment. We anneal τ linearly from 1.0 to 0.1 over 300 epochs (training is done for 500 epochs), and evaluate on Lorenz-63 at SMNR $= 10$ dB under the camera measurement map, with all other parameters at default. Table 9 compares the annealed schedule against the fixed $\tau = 0.1$ setting.

Table 9: Fixed versus annealed τ on Lorenz-63 (SMNR $= 10$ dB). Results show mean $\pm$ std over 10 runs.

	State Est.	Transition Model Quality			
Setting	NMSE (dB)	MSE	MSE-5-step	LL	KLD
Fixed ($\tau = 0.1$)	$-29.94_{\pm 0.28}$	$0.47_{\pm 0.24}$	$3.38_{\pm 2.11}$	$-1.98_{\pm 0.27}$	$3.60_{\pm 1.04}$
Annealed	$-30.33_{\pm 0.10}$	$0.40_{\pm 0.11}$	$3.01_{\pm 0.63}$	$-1.86_{\pm 0.13}$	$3.15_{\pm 0.32}$

Annealing gives a small but consistent improvement over the fixed setting across all metrics, and notably reduces run-to-run variance (e.g. NMSE std 0.10 versus 0.28 for fixed τ). This indicates that annealing can benefit learning; we leave a systematic study of annealing schedules for future work.

F.3 Dependence on N

We study the effect of the number of particles N on filtering and dynamics learning performance for all resampling methods on the Lorenz-63 system at SMNR $= 10$ dB using the Gaussian point spread observation map. All other hyperparameters are kept fixed. Results are shown in Table 10.

DSR is the only method that scales gracefully across all tested particle counts. Its NMSE improves from -29.94 dB at $N = 25$ to around -30.4 dB from $N = 50$ onwards and plateaus there, with consistently low variance (≤ 0.28 dB std) across runs. The transition model metrics show the same trend: MSE, LL, and KLD all improve with N and reach their best values at $N = 200$ (except MSE-5-step, which generally has a high variance).

By comparison, the other methods either fail to train stably or run out of memory. Hard and Soft are competitive at $N \leq 100$, but at $N = 200$ their performance collapses, with NMSE degrading to -20.42 and -21.52 dB and standard deviations above 7 and 12 dB across runs. These large variances are the result of training instability; there is a large variance across runs; for example, one run for Soft achieved an NMSE of 0.23 dB. A likely cause is that, at large N, the variance of the biased estimator $\nabla \mathcal{L} \approx g_{\text{rep}}$ (used by Hard) grows because more ancestor decisions are dropped from the gradient graph, and Soft's importance-weight correction becomes numerically less stable as the uniform component of its sampling distribution $(1 - \alpha)/N$ shrinks. EROT does not complete at $N = 200$

Table 10: Effect of particle count N on resampling methods for Lorenz-63 at SMNR $= 10$ dB. Results show mean $\pm$ std over 10 runs. OOM indicates that training exceeded available GPU memory.

		State Est.	Transition Model Quality			
N	Method	NMSE (dB)	MSE	MSE-5-step	LL	KLD
25	Hard	$\underline{-28.86}\pm0.41$	$\underline{0.55}\pm0.19$	5.50 ± 3.68	$\underline{-2.26}\pm0.25$	$\underline{5.41}\pm1.47$
	Soft	-27.82 ± 0.45	0.86 ± 0.49	6.45 ± 3.92	-2.55 ± 0.48	6.90 ± 2.43
	EROT	-28.76 ± 0.31	0.56 ± 0.24	$\underline{3.39}\pm2.59$	-2.32 ± 0.20	5.74 ± 1.03
	DSR	$\mathbf{-29.94}\pm0.28$	$\mathbf{0.47}\pm0.24$	$\mathbf{3.38}\pm2.11$	$\mathbf{-1.98}\pm0.27$	$\mathbf{3.60}\pm1.04$
50	Hard	-28.30 ± 0.88	$\mathbf{0.39}\pm0.08$	$\mathbf{1.73}\pm0.61$	$\underline{-1.98}\pm0.09$	$\underline{4.30}\pm0.66$
	Soft	-28.30 ± 0.97	0.59 ± 0.35	5.78 ± 3.19	-2.33 ± 0.45	6.27 ± 2.71
	EROT	-27.91 ± 0.39	0.59 ± 0.21	5.22 ± 2.95	-2.49 ± 0.21	6.88 ± 1.44
	DSR	$\mathbf{-30.34}\pm0.28$	$\underline{0.41}\pm0.13$	$\underline{1.95}\pm1.04$	$\mathbf{-1.77}\pm0.15$	$\mathbf{2.86}\pm0.67$
100	Hard	-28.34 ± 0.48	0.79 ± 0.21	4.65 ± 1.52	-2.61 ± 0.30	7.20 ± 3.42
	Soft	-28.93 ± 0.30	0.50 ± 0.24	3.35 ± 1.74	-2.00 ± 0.30	3.61 ± 0.75
	EROT	-27.55 ± 0.92	$\mathbf{0.39}\pm0.19$	$\mathbf{1.86}\pm1.20$	-2.03 ± 0.25	4.29 ± 0.96
	DSR	$\mathbf{-30.42}\pm0.20$	$\underline{0.44}\pm0.19$	$\underline{2.38}\pm0.83$	$\mathbf{-1.80}\pm0.28$	$\mathbf{2.82}\pm0.52$
200	Hard	$\underline{-20.42}\pm7.29$	0.75 ± 0.32	8.92 ± 4.96	$\underline{-2.54}\pm0.30$	$\underline{8.27}\pm2.74$
	Soft	-21.52 ± 12.20	$\underline{0.66}\pm0.43$	$\underline{5.39}\pm4.65$	$\underline{-2.54}\pm0.66$	9.62 ± 9.44
	EROT	OOM	OOM	OOM	OOM	OOM
	DSR	$\mathbf{-30.35}\pm0.22$	$\mathbf{0.37}\pm0.17$	$\mathbf{3.27}\pm2.23$	$\mathbf{-1.73}\pm0.23$	$\mathbf{2.52}\pm0.41$

because the $\mathcal{O}(N^2)$ transport matrix, combined with 100 Sinkhorn iterations per resampling step and the autodiff tape over $T = 100$ time steps, exceeds the available GPU memory.

These results illustrate two distinct scaling properties of DSR. First, it provides a stable gradient signal across particle counts, whereas Hard and Soft degrade. Second, it needs about a quarter of EROT's memory per resampling step (no Sinkhorn iterations, float32), so it still fits at $N = 200$, whereas EROT does not. The relaxation parameter (τ) of DSR is kept fixed at 0.1 across all N.

F.4 Dependence on spatial sorting of CDF

We now study the effect of Morton sorting on VSMC performance. As discussed, DSR operates on the empirical CDF, and the particle ordering determines which ancestors interact through the soft transport plan. Morton space-filling curve sorting places CDF neighbors close in state space, producing a transport plan that more closely approximates hard systematic resampling (Appendix E). The alternative is the unsorted order the filter produces (inherited from the previous resampling step), which allows soft assignments to mix spatially distant particles. We compare the two orderings with DSR on Lorenz-63 under the Gaussian point spread observation map (10).

Table 11: Effect of particle ordering on DSR performance for Lorenz-63 at SMNR $= 10$ and 0 dB. Results show mean $\pm$ std over 10 runs.

	State Est.	Transition Model Quality			
Ordering	NMSE (dB)	MSE	MSE-5-step	LL	KLD
		SMNR $= 10$ dB			
Morton	-29.94 ± 0.28	0.47 ± 0.24	3.38 ± 2.11	-1.98 ± 0.27	3.60 ± 1.04
Unsorted	-29.96 ± 0.34	0.44 ± 0.20	2.31 ± 0.80	-1.92 ± 0.21	3.59 ± 0.80
		SMNR $= 0$ dB			
Morton	-18.28 ± 0.46	0.54 ± 0.33	3.70 ± 2.87	-3.06 ± 0.24	13.41 ± 1.94
Unsorted	-18.43 ± 0.59	0.68 ± 0.18	6.55 ± 4.64	-3.19 ± 0.15	14.43 ± 2.23

The results can be seen in Table 11. At 10 dB, the two orderings are essentially indistinguishable, with overlapping error bars on every metric. At 0 dB, the picture is mixed: Morton sorting gives

a better transition model (MSE-5-step of 3.70 vs. 6.55, but within one standard deviation), while unsorted ordering gives marginally better state estimation (-18.43 vs. -18.28 dB NMSE), though this is within one standard deviation. A plausible reading is that the two orderings trade off along the dual role of the soft transport plan. Morton sorting produces a closer approximation to hard systematic resampling, preserving the spatial coherence that helps dynamics learning when weights become concentrated. Unsorted ordering mixes more distant particles, which softens the relaxation further; this extra diversity may help state estimation through broader particle coverage, but at the cost of a less structured gradient signal for the transition model. Overall, DSR is relatively robust to the choice of ordering, with Morton sorting offering a modest advantage for transition model learning at low SNR. Studying the interaction between particle ordering and posterior geometry, for instance, under multimodal or heavy-tailed posteriors, is an interesting direction for future work.

F.5 EROT regularization sensitivity at low SNR

Table 12 reports EROT performance on Lorenz-63 at 0 dB SMNR for $\varepsilon \in \{0.5, 1.0\}$, alongside DSR for reference. The default $\varepsilon = 0.5$, as used in [5, 31], exhibits severe training instability at this noise level, with MSE-5-step standard deviation exceeding the mean. Increasing ε to 1.0 stabilizes training and improves mean performance across all metrics, but DSR still outperforms EROT on every measure. Training time drops slightly with $\varepsilon = 1.0$ (40.35 s vs. 42.94 s per epoch), consistent with the Sinkhorn iterations stopping earlier at the larger ε.

Table 12: EROT regularization sensitivity on Lorenz-63 at SMNR = 0 dB. Results show mean $\pm$ std over 10 runs. DSR with $\tau = 0.1$ is included for reference. Best in **bold**.

	State Est.	Transition Model Quality				
Method	NMSE (dB)	MSE	MSE-5-step	LL	KLD	Time/epoch (s)
EROT ($\varepsilon = 0.5$)	$-12.57_{\pm 1.64}$	$4.79_{\pm 5.65}$	$48.41_{\pm 60.72}$	$-4.28_{\pm 1.54}$	$33.82_{\pm 29.79}$	$42.94_{\pm 0.77}$
EROT ($\varepsilon = 1.0$)	$-13.47_{\pm 0.95}$	$0.57_{\pm 0.43}$	$9.19_{\pm 7.58}$	$-3.49_{\pm 0.09}$	$20.20_{\pm 2.33}$	$40.35_{\pm 0.39}$
DSR ($\tau = 0.1$)	$\mathbf{-18.28}_{\pm 0.46}$	$\mathbf{0.54}_{\pm 0.33}$	$\mathbf{3.70}_{\pm 2.87}$	$\mathbf{-3.06}_{\pm 0.24}$	$\mathbf{13.41}_{\pm 1.94}$	$\mathbf{18.39}_{\pm 0.16}$

G Multimodal Posteriors and Inter-modal Mass

At finite τ, DSR forms convex combinations of ancestors, which can place mass between well-separated modes of a multimodal posterior. To study this phenomenon, we study a static two-component Gaussian (i.e., a mixture of two Gaussians). We draw particles from this two-component Gaussian, perform resampling, and study the intermodal mass. The two-component Gaussian is given by

$$\tfrac{1}{2}\mathcal{N}\big(-\tfrac{\Delta}{2}\mathbf{e}_0,\, \sigma^2\mathbf{I}_d\big) + \tfrac{1}{2}\mathcal{N}\big(+\tfrac{\Delta}{2}\mathbf{e}_0,\, \sigma^2\mathbf{I}_d\big), \tag{18}$$

where Δ is the mode separation, $\mathbf{e}_0 = (1, 0, \ldots, 0)^\top$, d is the dimension, and σ^2 is a scalar for the covariance. The modes are separated along coordinate 0 only, while coordinates $1, \ldots, d-1$ carry isotropic noise and no mode information. We use $N = 100$, $\Delta = 10$, $\sigma = 0.3$ (a separation of ≈ 33 within-mode standard deviations), and measure the resampled mass in the inter-modal region $|x_0| < 3$. Particles are randomly permuted before resampling, and we use the manuscript defaults ($\alpha = 0.5$; $\tau = 0.1$; $\varepsilon = 0.5$ with at most 100 Sinkhorn iterations). We report the inter-modal mass as mean $\pm$ std over 10^4 simulations.

Any scheme that copies ancestors places zero mass in the region by construction, since no particle lies there; this holds for hard systematic and soft multinomial resampling. Any inter-modal mass under DSR or EROT is therefore attributable to the relaxation. Table 13 reports the results.

The results support the use of Morton sorting in this setting. Without Morton sorting, every resampled particle lands between the modes; with Morton sorting the inter-modal mass drops to ≈ 0.28, and it does not degrade over this range of d. EROT, by contrast, is sensitive to dimension: for $d \geq 3$ it places most resampled mass between the modes. Morton sorting thus substantially reduces inter-modal mass relative to no sorting for dimensions up to 10. Multimodal posteriors remain challenging for particle filters in general [32, 30], and investigating differentiable resampling under such posteriors is an interesting direction for future work.

Table 13: Inter-modal mass (resampled mass in $|x_0| < 3$) for a static two-component Gaussian as a function of dimension d. Lower is better; 0 means no mass placed between the modes. Results show mean $\pm$ std over 10^4 simulations.

Method	$d = 1$	$d = 2$	$d = 3$	$d = 5$	$d = 10$
Hard SR	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
Soft	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
EROT	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.747_{\pm 0.052}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$
DSR (Morton)	$0.287_{\pm 0.005}$	$0.283_{\pm 0.005}$	$0.282_{\pm 0.005}$	$0.282_{\pm 0.005}$	$0.282_{\pm 0.005}$
DSR (w/o Morton)	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$

H Additional Linear Gaussian Experiments

The linear Gaussian SSM of Section 5, as the state transition model is known, admits optimal filtering recursion via the Kalman filter, which lets us approach two questions not accessible in the chaotic or real-world settings: how the marginal likelihood at evaluation time decomposes across methods, and how the DSR gradient compares to an unbiased VSMC gradient.

H.1 Marginal likelihood at evaluation time

At evaluation time all methods use hard systematic resampling, so $\log \hat{p}(\mathbf{y}_{1:T})$ is directly comparable across methods, and on the linear Gaussian model the exact log-evidence is available from the Kalman filter. This yields the decomposition

$$\underbrace{\log p_{\mathrm{true}}(\mathbf{y}_{1:T}) - \log p_{\hat{\boldsymbol{\phi}}}(\mathbf{y}_{1:T})}_{\text{model error}} + \underbrace{\log p_{\hat{\boldsymbol{\phi}}}(\mathbf{y}_{1:T}) - \mathbb{E}[\log \hat{p}_N(\mathbf{y}_{1:T})]}_{\text{Jensen's gap}}, \tag{19}$$

where the model error measures how close the learned transition model is to the data-generating model, and the Jensen's gap measures how well an N-particle filter approximates the evidence of that learned model. Using the trained models from Section 5 on the linear Gaussian SSM with default parameters ($T = 100$, SMNR $= 10$ dB, mean $\pm$ std over model seeds on 10 shared sequences), we obtain Table 14.

Table 14: Evaluation-time evidence decomposition on the linear Gaussian SSM ($T = 100$, SMNR $= 10$ dB). The true log-evidence is -232.367. Results show mean $\pm$ std over model seeds.

Method	$\log p_{\hat{\boldsymbol{\phi}}}$	$\mathbb{E}[\log \hat{p}_N]$	model error	Jensen's gap
DSR	$-232.942_{\pm 0.087}$	$\mathbf{-235.138}_{\pm 0.385}$	$0.575_{\pm 0.087}$	$\mathbf{2.196}_{\pm 0.302}$
EROT	$-233.065_{\pm 0.104}$	$-235.851_{\pm 0.414}$	$0.699_{\pm 0.104}$	$2.786_{\pm 0.467}$
Hard	$\mathbf{-232.795}_{\pm 0.038}$	$-236.147_{\pm 0.924}$	$\mathbf{0.428}_{\pm 0.038}$	$3.352_{\pm 0.901}$
Soft	$-232.996_{\pm 0.138}$	$-237.215_{\pm 0.749}$	$0.630_{\pm 0.138}$	$4.218_{\pm 0.669}$

All four methods are within 0.7 of the true evidence. Hard SR attains the smallest model error (0.428 versus 0.575 for DSR, a gap outside the spread over seeds), but the methods separate on the Jensen's gap, where DSR attains 2.196 against 3.352 for hard SR and with roughly a third of the spread ($\pm.302$ versus $\pm.901$). The two columns measure different things: since the optimised quantity is $\mathbb{E}[\log \hat{p}_N]$ rather than $\log p$, these results indicate that DSR learns parameters whose evidence the finite-particle filter approximates more accurately.

H.2 Comparison against unbiased VSMC gradients

We compare the DSR gradient against an unbiased VSMC gradient. Here DSR is the plain relaxation of Eq. 5, without Morton sorting or the gradient clip. This is to study the gradient bias induced by DSR. We follow the linear Gaussian experiment of [26] adapted to our setting. We use the linear Gaussian SSM introduced in Sec. 5.2, using SMNR $= 10$ dB with $N = 4$, $T = 4$, and additionally a two-dimensional version (a single rotation block) at the same N and T. For each we generate one observation sequence $\mathbf{y}_{1:T}$ that every run below targets. Gradients are taken with respect to all d^2 entries of $\mathbf{A}$ and the d diagonal entries of $\mathbf{Q} = \mathrm{diag}(\mathrm{softplus}(\boldsymbol{\ell}))$, evaluated at the data-generating

parameters. The unbiased gradient $\nabla\mathcal{L}$ is estimated by averaging over 1000 bootstrap noise draws and exhaustively summing over ancestor vectors (the vectors are the systematic combs induced by a single uniform draw). Let $\hat{\mathbf{g}}$ denote an estimate obtained by averaging $n = 1600\,(= 100/4 \times 64)$ independent SMC runs, matching the number of observations consumed in our training configuration (Appendix C), each run random through its bootstrap noise and resampling offsets. We report

$$\text{Bias} = \left\|\mathbb{E}[\hat{\mathbf{g}}] - \nabla\mathcal{L}\right\|, \quad \text{Variance} = \mathbb{E}\left\|\hat{\mathbf{g}} - \mathbb{E}[\hat{\mathbf{g}}]\right\|^2, \quad \text{MSE} = \mathbb{E}\left\|\hat{\mathbf{g}} - \nabla\mathcal{L}\right\|^2 = \text{Bias}^2 + \text{Variance}. \tag{20}$$

We estimate $\mathbb{E}[\hat{\mathbf{g}}]$ from the same 1000 bootstrap draws with 500 offset sequences (5×10^5 runs), and the variance separately from 200 independent realisations of $\hat{\mathbf{g}}$ at $n = 1600$. Results are in Table 15.

Table 15: Gradient bias, variance, and MSE against the unbiased VSMC gradient on the linear Gaussian SSM. Hard SR drops the score-function term. Results show mean $\pm$ standard errors.

Method	Bias	Variance	MSE
(a) $d = 2, N = 4, T = 4;\ \|\nabla\mathcal{L}\| = 9.64$			
Hard SR	$2.324_{\pm 0.014}$	$0.308_{\pm 0.016}$	$5.707_{\pm 0.067}$
DSR ($\tau = 1.0$)	$1.256_{\pm 0.000}$	$0.241_{\pm 0.014}$	$1.818_{\pm 0.014}$
DSR ($\tau = 0.5$)	$1.257_{\pm 0.001}$	$0.245_{\pm 0.014}$	$1.825_{\pm 0.014}$
DSR ($\tau = 0.1$)	$0.764_{\pm 0.020}$	$0.389_{\pm 0.031}$	$0.973_{\pm 0.043}$
DSR ($\tau = 0.05$)	$0.370_{\pm 0.050}$	$0.723_{\pm 0.037}$	$0.860_{\pm 0.052}$
DSR ($\tau = 0.01$)	$0.343_{\pm 0.227}$	$13.409_{\pm 2.583}$	$13.526_{\pm 2.587}$
(b) $d = 4, N = 4, T = 4;\ \|\nabla\mathcal{L}\| = 20.50$			
Hard SR	$3.627_{\pm 0.028}$	$2.092_{\pm 0.058}$	$15.248_{\pm 0.212}$
DSR ($\tau = 1.0$)	$2.869_{\pm 0.001}$	$2.073_{\pm 0.057}$	$10.304_{\pm 0.057}$
DSR ($\tau = 0.5$)	$2.914_{\pm 0.005}$	$2.127_{\pm 0.058}$	$10.615_{\pm 0.064}$
DSR ($\tau = 0.1$)	$2.131_{\pm 0.088}$	$4.895_{\pm 0.192}$	$9.437_{\pm 0.420}$
DSR ($\tau = 0.05$)	$1.720_{\pm 0.190}$	$16.382_{\pm 1.864}$	$19.340_{\pm 1.975}$
DSR ($\tau = 0.01$)	$1.412_{\pm 1.216}$	$384.805_{\pm 102.564}$	$386.799_{\pm 102.621}$

DSR has a smaller gradient bias than hard SR (which drops g_{score}) at every temperature and in both settings, and the bias decreases as τ decreases: at $\tau = 1.0$ the bias drops from 2.324 to 1.256 ($d = 2$) and from 3.627 to 2.869 ($d = 4$) with essentially unchanged variance. Decreasing τ further keeps reducing the bias but increases the variance sharply. At the operating temperature $\tau = 0.1$, DSR is competitive on MSE with hard SR and the other τ values, attaining the lowest MSE at $d = 4$ and the second lowest at $d = 2$. This experiment makes the bias–variance tradeoff explicit and motivates the annealing strategy studied in Appendix F.2.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer:

   Answer: [Yes]

   Justification: The abstract and Introduction make three main claims; see the enumerated section, lines 55–60. These are supported respectively by Section 3 (method), Section 4 (theory) and Appendix B (Propositions 4.1 and 4.2 with proofs), and Section 5 (experiments on Linear Gaussian SSM, Lorenz-63, and CharacterTrajectories, including Table 5 on training time).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer:

   Answer: [Yes]

   Justification: Section 5.5 explicitly discusses the convex combination limitation in multimodal posterior scenarios. The bound in Proposition 4.2 is also discussed as primarily qualitative, with worst-case scaling $\Delta_j = \mathcal{O}(1/N)$ for near-uniform weights (Section 4). Sensitivity to the temperature parameter is studied in Appendix F.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Propositions 4.1 (asymptotic exactness) and 4.2 (exponential convergence rate) are stated in Section 4 with all required quantities defined. Complete proofs are provided in Appendix B. All theoretical results are numbered and cross-referenced. The proofs have been verified by the authors.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The DSR algorithm is fully specified in Algorithms 2 and 3 (Appendix A). Architecture and training hyperparameters are detailed in Appendix C for the experiments. Code will be released upon acceptance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code will be released upon acceptance, and the datasets used are referenced and all are open sourced.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix C specifies all training and test details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All reported results are mean ± standard deviation over 10 independent runs with different random seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Wall-clock training times are reported in Table 5, trained on T4 GPU as specified.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors and the study conform to the NeurIPS Code of Ethics. The work uses only synthetic data and a publicly available open-source dataset.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Societal impacts are indirectly discussed in the introduction.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The authors do not see such risks. Open-source data has been used, and the algorithm derived is for state estimation and dynamics learning, which the authors deem to pose small risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The CharacterTrajectories dataset from the UEA multivariate time series archive is used and properly cited. All baseline methods are also properly cited.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No new assets are introduced. Code will be released upon acceptance.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Core method development does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.