Normalising Flow-based Differentiable Particle Filters

Yunpeng Li

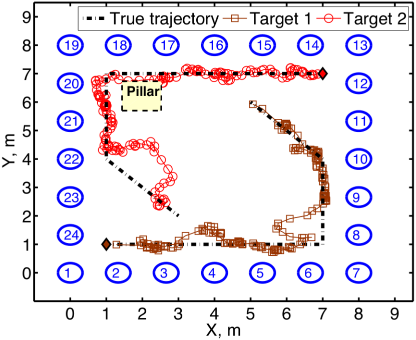
Centre for Oral, Clinicial & Translational Sciences
King's College London, UK

Major contributor: Xiongjie Chen

KCL Statistics Seminar

21st November, 2024

Motivating Examples: RF tomographic tracking[1]



[1] Li et al., "Sequential Monte Carlo radio-frequency tomographic tracking", ICASSP, 2011.

More complex scenario: autonomous driving[2]

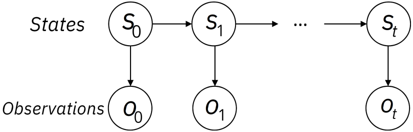
Multiple sensors

Multiple targets

[2] Redmon and Farhadi, "YOLO9000: better, faster, stronger", CVPR, 2017.

Filtering problem formulation

Recursive Bayesian Filtering: when the state and observation are sequence data.



- Dynamic model $p_\theta(s_t|s_{t-1})$: transition of hidden state.
- Measurements model $p_\theta(o_t|s_t)$: likelihood of the observation given the state.
- Goal: sequentially obtain marginal posterior $p_\theta(s_t|o_{0:t})$ or joint posterior $p_\theta(s_{1:t}|o_{0:t})$.

Filtering (non-linear models)

Particle filters:
sequential approximation of marginal posterior $p_\theta(s_t|o_{1:t})$ or joint posterior $p_\theta(s_{1:t}|o_{1:t})$ with particles i.e. weighted samples.

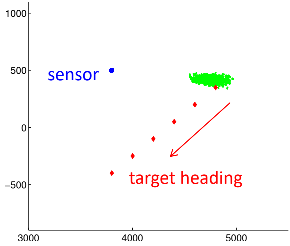
(Bootstrap) Particle Filters[3] in one slide

- Particle filters, a.k.a. sequential Monte Carlo (SMC) methods: Weighted samples to sequentially approximate target distribution.

[3] Gordon et al., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", in IEE Proc. FRSP, 1993

(Bootstrap) Particle Filters[3] in one slide

- Particle filters, a.k.a. sequential Monte Carlo (SMC) methods: Weighted samples to sequentially approximate target distribution.



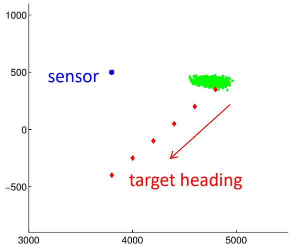
Use particle approximation of target state posterior

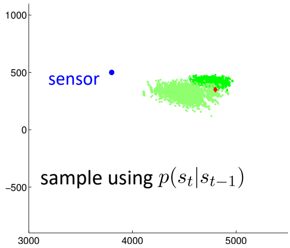
$\hat{p}(s_{t-1}|o_{1:t-1}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{s_{t-1}^i}(s_{t-1})$

[3] Gordon et al., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", in IEE Proc. FRSP, 1993

(Bootstrap) Particle Filters[3] in one slide

- Particle filters, a.k.a. sequential Monte Carlo (SMC) methods: Weighted samples to sequentially approximate target distribution.

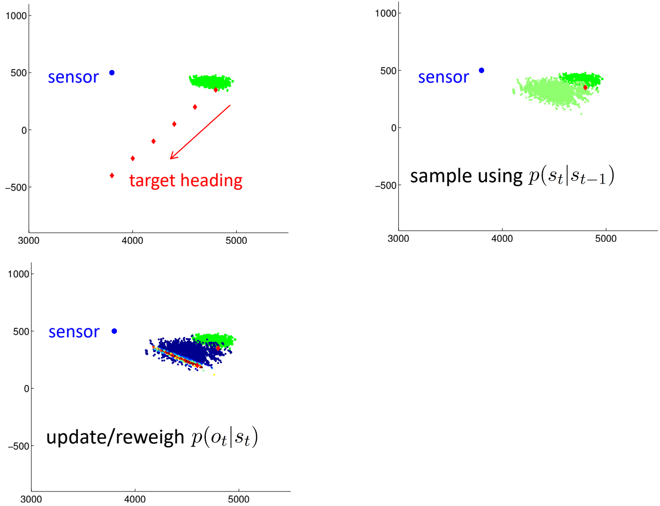




[3] Gordon et al., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", in IEE Proc. FRSP, 1993

(Bootstrap) Particle Filters[3] in one slide

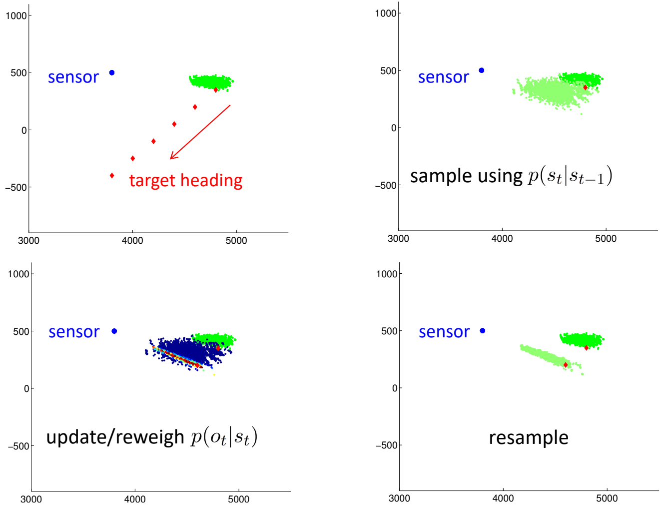
- Particle filters, a.k.a. sequential Monte Carlo (SMC) methods: Weighted samples to sequentially approximate target distribution.



[3] Gordon et al., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", in IEE Proc. FRSP, 1993

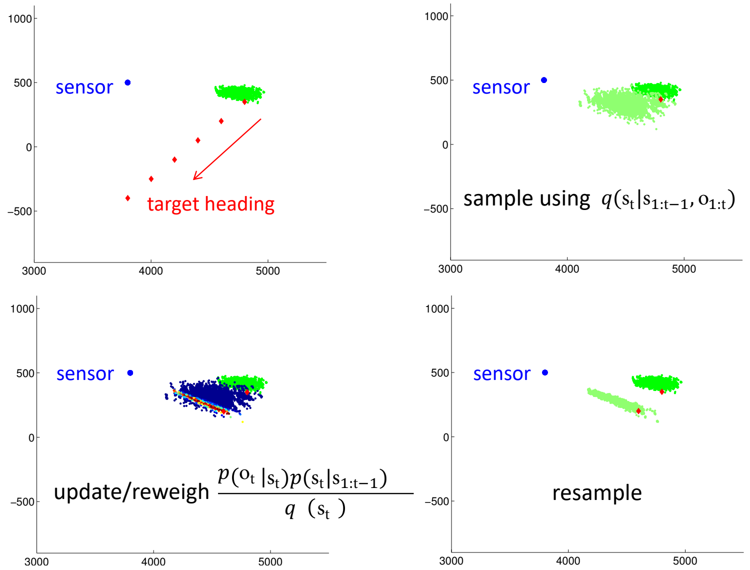
(Bootstrap) Particle Filters[3] in one slide

- Particle filters, a.k.a. sequential Monte Carlo (SMC) methods: Weighted samples to sequentially approximate target distribution.



[3] Gordon et al., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", in IEE Proc. FRSP, 1993

Particle filters: more generally



Parameter estimation for particle filtering

- Components of particle filters are usually parametrised by some parameter θ.
- Can we learn these parameters from data?
  - Maximum likelihood (ML) estimation[4]
  - Bayesian estimation[5]

[4] Kantas et al., "An overview of sequential Monte Carlo methods for parameter estimation in general state-space models", IFAC, 2009

[5] Kantas et al., "On particle methods for parameter estimation in state-space models", Statistical Science, 2015

Parameter estimation for particle filtering

- Components of particle filters are usually parametrised by some parameter θ.
- Can we learn these parameters from data?
  - Maximum likelihood (ML) estimation[4]
  - Bayesian estimation[5]

Can be effective, but ...

- Assume that the structures or part of parameters of the dynamic and measurement models are known.

[4] Kantas et al., "An overview of sequential Monte Carlo methods for parameter estimation in general state-space models", IFAC, 2009

[5] Kantas et al., "On particle methods for parameter estimation in state-space models", Statistical Science, 2015

Real-world scenarios?

High-dimensional unstructured observations, e.g. images[6].

[6]Geiger et al., "Are we ready for autonomous driving? The KITTI vision benchmark suite", CVPR, 2012

Real-world scenarios?

High-dimensional unstructured observations, e.g. images[6].

Designing particle filters can be complicated in complex environments:

- Dynamic model — How does the hidden state evolve?
  1. Which distribution family to use?
  2. How to optimise distribution parameters?

[6] Geiger et al., "Are we ready for autonomous driving? The KITTI vision benchmark suite", CVPR, 2012

Real-world scenarios?

High-dimensional unstructured observations, e.g. images[6].

Designing particle filters can be complicated in complex environments:

- Dynamic model — How does the hidden state evolve?
  1. Which distribution family to use?
  2. How to optimise distribution parameters?
- Measurement model — How to model the relationship between observations and hidden states?

[6] Geiger et al., "Are we ready for autonomous driving? The KITTI vision benchmark suite", CVPR, 2012

Real-world scenarios?

High-dimensional unstructured observations, e.g. images[6].

Designing particle filters can be complicated in complex environments:

- Dynamic model — How does the hidden state evolve?
  1. Which distribution family to use?
  2. How to optimise distribution parameters?
- Measurement model — How to model the relationship between observations and hidden states?
- Proposal distribution — How to use information from observations to construct good proposal distributions?

[6] Geiger et al., "Are we ready for autonomous driving? The KITTI vision benchmark suite", CVPR, 2012

Basic idea of differentiable particle filters[7]

Combining particle filters with deep learning tools: Differentiable particle filters (DPFs).

- Build components of particle filters with neural networks.
- Optimise these components by gradient descent.

Components of differentiable particle filters:

- Dynamic model
- Measurement model
- Proposal distribution

} can be built with neural networks

- Differentiable resampling
- Loss function & gradient descent.

[7] Jonschkowski et al., "Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors", RSS, 2018.

What does differentiable mean?

Differentiable particle filters:

- All components need to be differentiable.

What does differentiable mean?

Differentiable particle filters:

- All components need to be differentiable.
- Parametrise differentiable components with θ (model parameters) and ϕ (proposal parameters).

What does differentiable mean?

Differentiable particle filters:

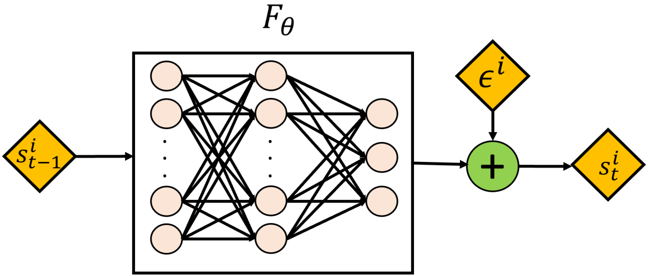
- All components need to be differentiable.
- Parametrise differentiable components with θ (model parameters) and ϕ (proposal parameters).
- Optimise by gradient descent with a loss function $\mathcal{L}$:

$$\theta \to \theta - \nabla_\theta \mathcal{L}\,,$$
$$\phi \to \phi - \nabla_\phi \mathcal{L}\,.$$

Differentiable particle filters: dynamic model

Reparameterisation trick.

- Adding noise to deterministic functions, e.g. neural networks.

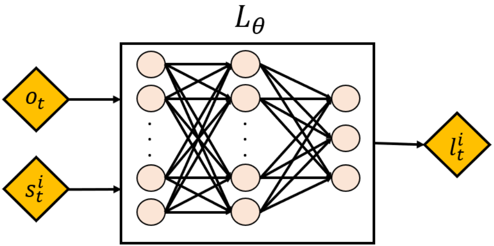


$$s_t^i = F_\theta(s_{t-1}^i) + \epsilon_t^i \sim p(s_t|s_{t-1}^i)$$

Differentiable particle filters: measurement models

Model the likelihood of observations given states with parametrised functions $L_\theta(\cdot)$:

- Compare feature vectors of observations and states given by neural networks.



$$l_t^i = p(o_t|s_t^i) = L_\theta(o_t, s_t^i), \qquad w_t^i = l_t^i w_{t-1}^i$$

[7] Jonschkowski et al., "Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors", RSS, 2018.
[8] Karkus et al., "Particle Filter Networks with Application to Visual localisation", CoRL, 2018.
[9] Wen et al., "End-To-End Semi-supervised Learning for Differentiable Particle Filters", ICRA, 2021.

Differentiable resampling

The standard multinomial resampling step is non-differentiable.

- Small changes in weights lead to discrete changes in output.

Different approaches, e.g.:

1. Soft resampling[8] (not really differentiable).
   - Resample with new weights $\dot{\tilde{w}}_t^i = \lambda \tilde{w}_t^i + (1 - \lambda)\frac{1}{N}$.
   - Non-zero gradients:

$$\hat{\tilde{w}}_t^i = \frac{\tilde{w}_t^i}{\dot{\tilde{w}}_t^i} = \frac{\tilde{w}_t^i}{\lambda \tilde{w}_t^i + (1 - \lambda)1/N} .$$

[8] Karkus et al., "Particle Filter Networks with Application to Visual Localization", CoRL, 2018.

[10] Corenflos et al., "Differentiable Particle Filtering via Entropy-regularized Optimal Transport." ICML, 2021.

Differentiable resampling

The standard multinomial resampling step is non-differentiable.

- Small changes in weights lead to discrete changes in output.

Different approaches, e.g.:

1. Soft resampling[8] (not really differentiable).
   - Resample with new weights $\dot{\tilde{w}}_t^i = \lambda \tilde{w}_t^i + (1-\lambda)\frac{1}{N}$.
   - Non-zero gradients:

$$\hat{\tilde{w}}_t^i = \frac{\tilde{w}_t^i}{\dot{\tilde{w}}_t^i} = \frac{\tilde{w}_t^i}{\lambda \tilde{w}_t^i + (1-\lambda)1/N} .$$

2. Entropy-regularised optimal transport resampling[10].
   - No multinomial resampling.
   - Consider the resampling step as an optimal transport problem.
   - Solve an entropy regularised OT problem via Sinkhorn iterations.
   - Resampled particles $\{\frac{1}{N}, \tilde{s}_t^i\}$ do not from ancestors.

[8] Karkus et al., "Particle Filter Networks with Application to Visual Localization", CoRL, 2018.

[10] Corenflos et al., "Differentiable Particle Filtering via Entropy-regularized Optimal Transport." ICML, 2021.

Differentiable Particle Filters: Training Objective

End-to-End learning by minimizing a given loss function:

1. Supervised losses (require ground-truth latent states)[7,8].
   - The mean squared error (MSE):

     $$L_{MSE}(\theta) = \sum_{t=0}^{T}(s_t^* - s_t)^T(s_t^* - s_t)\,,$$

   - The negative log likelihood (NLL):

     $$L_{NLL}(\theta) = -\sum_{t=0}^{T}\log\sum_{i=1}^{N}\frac{w_t^i}{\sqrt{|\Sigma|}}\exp(-\frac{1}{2}(s_t^* - s_t)^T\Sigma^{-1}(s_t^* - s_t))\,,$$

     where s_t^* is the ground truth state, s_t is the estimated state.

[7] Jonschkowski et al., "Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors", RSS, 2018.

[8] Karkus et al., "Particle Filter Networks with Application to Visual localisation", CoRL, 2018.

[9] Hao et al., "End-To-End Semi-supervised Learning for Differentiable Particle Filters", ICRA, 2021.

[11] Le et al., "Auto-Encoding Sequential Monte Carlo", ICLR, 2018.

Differentiable Particle Filters: Training Objective

End-to-End learning by minimizing a given loss function:

1. Supervised losses (require ground-truth latent states)[7,8].
   - The mean squared error (MSE):

     $$L_{MSE}(\theta) = \sum_{t=0}^{T}(s_t^* - s_t)^T(s_t^* - s_t)\,,$$

   - The negative log likelihood (NLL):

     $$L_{NLL}(\theta) = -\sum_{t=0}^{T}\log\sum_{i=1}^{N}\frac{w_t^i}{\sqrt{|\Sigma|}}\exp(-\tfrac{1}{2}(s_t^* - s_t)^T\Sigma^{-1}(s_t^* - s_t))\,,$$

     where s_t^* is the ground truth state, s_t is the estimated state.

2. Observation likelihood-based loss.
   - Pseudo-likelihood[9].
   - Evidence Lower Bound (ELBO)[11].

[7] Jonschkowski et al., "Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors", RSS, 2018.

[8] Karkus et al., "Particle Filter Networks with Application to Visual localisation", CoRL, 2018.

[9] Hao et al., "End-To-End Semi-supervised Learning for Differentiable Particle Filters", ICRA, 2021.

[11] Le et al., "Auto-Encoding Sequential Monte Carlo", ICLR, 2018.

Limitations in existing variants

- Only able to generate Gaussian prior.
- Bootstrap particle filtering framework or proposal distributions that only use latest observations while ignore states.
- Measurement models are either Gaussian or do not admit valid probability densities.

Xiongjie Chen and Yunpeng Li, "Normalising flow-based differentiable particle filters", arXiv:2403.01499, March 2024.

Normalising Flows

Definition of normalising flows:

$$y = \mathcal{T}_\theta(x),$$

where $\mathcal{T}_\theta$ is required to be an invertible transformation.

[12] Rezende et al., "Variational Inference with Normalizing Flows", ICML, 2015.

[13] Dinh et al., "Density Estimation using Real NVP" ICLR, 2017.

Normalising Flows

Definition of normalising flows:

$$y = \mathcal{T}_\theta(x),$$

where $\mathcal{T}_\theta$ is required to be an invertible transformation.

Why invertible transformations?

- Invertibility allows density estimation (change of variable):

$$p(y) = p(x)\left|\det\frac{d\mathcal{T}_\theta(x)}{dx}\right|^{-1}.$$

[12] Rezende et al., "Variational Inference with Normalizing Flows", ICML, 2015.

[13] Dinh et al., "Density Estimation using Real NVP" ICLR, 2017.

Normalising Flows

Definition of normalising flows:

$$y = \mathcal{T}_\theta(x),$$

where $\mathcal{T}_\theta$ is required to be an invertible transformation.

How about conditional probability densities?

- Given a condition u, we can build conditional normalising flows:

$$y = \mathcal{G}_\theta(x; u)\,.$$

Conditional probability of y given u:

$$p(y|u) = p(x)\left|\det \frac{d\mathcal{G}_\theta(x;u)}{dx}\right|^{-1}$$

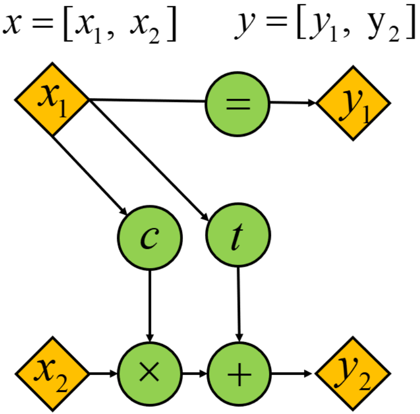
[12] Rezende et al., "Variational Inference with Normalizing Flows", ICML, 2015.

[13] Dinh et al., "Density Estimation using Real NVP" ICLR, 2017.

Examples of Normalising Flows: Coupling Layer

Real-NVP[13]

- Coupling layers.



[13] Ding et al., "Density Estimation Using Real NVP", ICLR, 2017.

Examples of Normalising Flows: Coupling Layer

Real-NVP[13]

- Coupling layers.

The special structure of coupling layers leads to triangular Jacobian matrix:

$$y_{1:d} = x_{1:d}$$

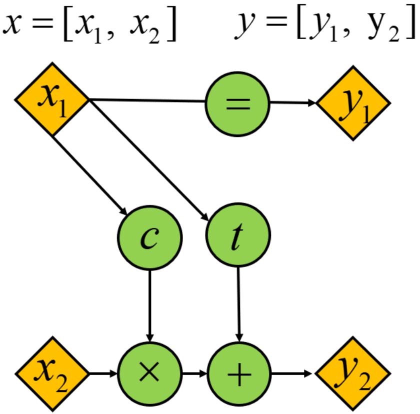
$$y_{d+1:D} = x_{d+1:D} \odot \exp(c(x_{1:d})) + t(x_{1:d})$$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \mathbb{I} & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \mathsf{diag}(\exp[c(x_{1:d})]) \end{bmatrix}$$

[13] Ding et al., "Density Estimation Using Real NVP", ICLR, 2017.

Conditional Coupling Layer

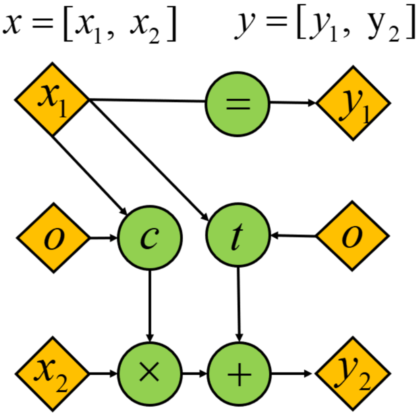
We use conditional coupling layer to construct conditional Real-NVP:



Standard coupling layer

Conditional Coupling Layer

We use conditional coupling layer to construct conditional Real-NVP:



Conditional coupling layer

Conditional Coupling Layer[14]

- Conditional coupling layer:

$$\underset{1:d}{y} = \underset{1:d}{x}$$
$$\underset{d+1:D}{y} = \underset{d+1:D}{x} \odot \exp(c(\underset{1:d}{x}, o)) + t(\underset{1:d}{x}, o)$$

- Standard coupling layer:

$$\underset{1:d}{y} = \underset{1:d}{x}$$
$$\underset{d+1:D}{y} = \underset{d+1:D}{x} \odot \exp(c(\underset{1:d}{x})) + t(\underset{1:d}{x})$$

[14] Winkler et al., "Learning Likelihoods with Conditional Normalizing Flows", arXiv, 2019.

Conditional Coupling Layer[14]

- Conditional coupling layer:

$$
\begin{aligned}
y_{1:d} &= x_{1:d} \\
y_{d+1:D} &= x_{d+1:D} \odot \exp(c(x_{1:d}, o)) + t(x_{1:d}, o)
\end{aligned}
$$

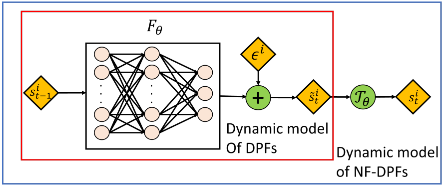
Still invertible and lead to triangular Jacobian matrix:

$$
\frac{\partial y}{\partial x} = \begin{bmatrix} \mathbb{I} & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}} & \mathsf{diag}(\exp[c(x_{1:d}, o)]) \end{bmatrix}
$$

[14] Winkler et al., "Learning Likelihoods with Conditional Normalizing Flows", arXiv, 2019.

NF-DPFs: Dynamic Model and Proposal[15]

Normalising flow-based differentiable particle filters (NF-DPFs).

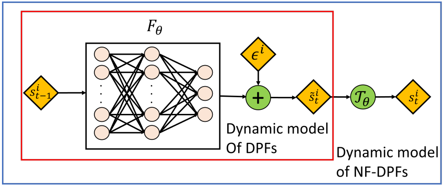


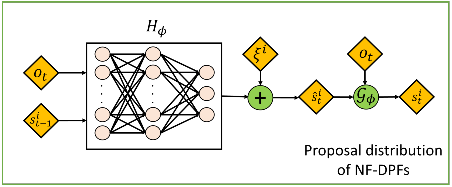
1. Dynamic normalising flow $\mathcal{T}_\theta(\cdot) : \mathcal{X} \rightarrow \mathcal{X}$: construct flexible dynamic models.

[15] Chen et al., "Differentiable particle filters through conditional normalising flow," FUSION, 2021.

NF-DPFs: Dynamic Model and Proposal[15]

Normalising flow-based differentiable particle filters (NF-DPFs).

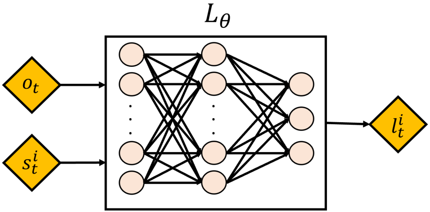




1. Dynamic normalising flow $\mathcal{T}_\theta(\cdot) : \mathcal{X} \to \mathcal{X}$: construct flexible dynamic models.
2. Conditional normalising flow $\mathcal{G}_\phi(\cdot) : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$: move particles to areas closer to posterior by utilising information from observations.

[15] Chen et al., "Differentiable particle filters through conditional normalising flow," FUSION, 2021.

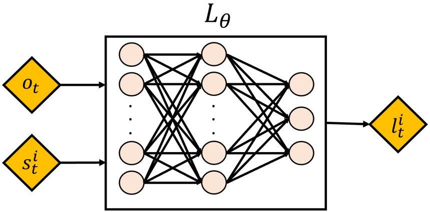
NF-DPFs: Measurement Model[16]



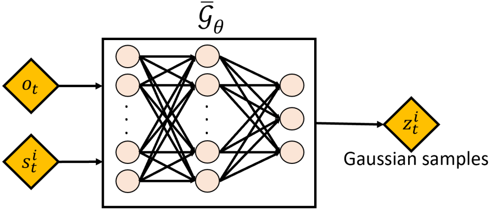
Existing measurement models

[16] Chen and Li, "Conditional Measurement Density Estimation in Sequential Monte Carlo via Normalising Flow," EUSIPCO, 2022.

NF-DPFs: Measurement Model[16]



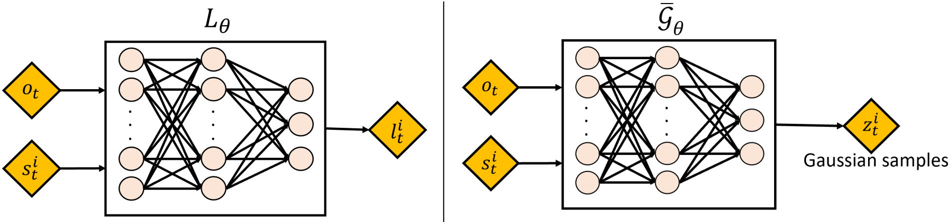
Existing measurement models



Proposed measurement model

[16] Chen and Li, "Conditional Measurement Density Estimation in Sequential Monte Carlo via Normalising Flow," EUSIPCO, 2022.

NF-DPFs: Measurement Model[16]

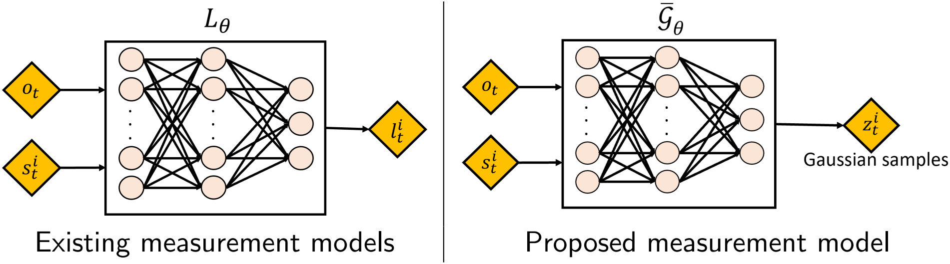


Existing measurement models | Proposed measurement model

1. Valid probability densities $p(o_t|s_t) = p(z_t)\left|\det \frac{d\bar{\mathcal{G}}_\theta(o_t;s_t)}{do_t}\right|$ with $\bar{\mathcal{G}}_\theta(\cdot) : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$.

[16] Chen and Li, "Conditional Measurement Density Estimation in Sequential Monte Carlo via Normalising Flow," EUSIPCO, 2022.

NF-DPFs: Measurement Model[16]



1. Valid probability densities $p(o_t|s_t) = p(z_t)\left|\det \frac{d\bar{\mathcal{G}}_\theta(o_t;s_t)}{do_t}\right|$ with $\bar{\mathcal{G}}_\theta(\cdot) : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$.
2. Can be trained with likelihood-based loss functions:

$$\log p(o_t|o_{0:t-1}) = \log \int p(s_{1:t}|o_{0:t-1})p(o_t|s_t)ds_{1:t} \,.$$

[16] Chen and Li, "Conditional Measurement Density Estimation in Sequential Monte Carlo via Normalising Flow," EUSIPCO, 2022.

Theoretical Results

How to establish convergence results for differentiable particle filters?

- Main difference: resamplers.
  - multinomial $\rightarrow$ entropy-regularised optimal transport.

[17] Crisan and Doucet, "A Survey of Convergence Results on Particle Filtering Methods for Practitioners", IEEE TSP, 2002.

Theoretical Results

How to establish convergence results for differentiable particle filters?

- Main difference: resamplers.
  - multinomial $\rightarrow$ entropy-regularised optimal transport.
- Standard PFs using multinomial resampling[17](not differentiable):

$$\mathbb{E}\left[\left(\alpha_N^{(t)}(\psi) - \beta^{(t-1)} f(\psi)\right)^2\right] \leq c_t \frac{||\psi||_\infty^2}{N}\,, \quad t \geq 1,$$

$$\mathbb{E}\left[\left(\beta_N^{(t)}(\psi) - \beta^{(t)}(\psi)\right)^2\right] \leq {c'}_t \frac{||\psi||_\infty^2}{N}\,, \quad t \geq 0.$$

- $\alpha^{(t)} := p(s_t|o_{0:t-1};\theta)$: predictive distributions at t.
- $\alpha_N^{(t)} := \frac{1}{N}\sum_{i=1}^N s_t^i$: approximations of $\alpha^{(t)}$.
- $\beta^{(t)} := p(s_t|o_{0:t};\theta)$: posterior distributions at t.
- $\beta_N^{(t)} := \sum_{i=1}^N \tilde{w}_t^i s_t^i$: approximations of $\beta^{(t)}$.
- $f(\cdot)$: a transition kernel defined by $p(s_t|s_{t-1};\theta)$.

[17] Crisan and Doucet, "A Survey of Convergence Results on Particle Filtering Methods for Practitioners", IEEE TSP, 2002.

Theoretical Results

Assumption 1: the state $s_t \in \mathcal{X}$ is defined on a compact set $\mathcal{X}$ with finite diameter $\mathfrak{d}$.

Assumption 2: the optimal transport plan between $\alpha^{(t)}$ and $\beta^{(t)}$ is unique and the corresponding transport map is λ-Lipschitz.

Assumption 3: For any two probability measures μ and ρ and k-Lipschitz function $\psi(\cdot)$, the transition kernel $f(\cdot)$ satisfy:

$$|\mu f(\psi) - \rho f(\psi)| \leq \eta |\mu(\psi) - \rho(\psi)| \, .$$

Assumption 4: For any probability measure μ and its empirical approximation μ_N, the conditional likelihood function $\omega_t(\cdot)$ satisfy:

$$\mathcal{W}_2(\mu_{N,\omega_t}, \mu_{\omega_t}) \leq \zeta \mathcal{W}_2(\mu_N, \mu) \, ,$$

where $\mathcal{W}_2$ denotes 2-Wasserstein distances, $\mu_{\omega_t} = \omega_t \mu / \mu(\omega_t)$ and $\mu_{N,\omega_t} = \omega \mu_N / \mu_N(\omega_t)$ are weighted probability measures.

[10] Corenflos et al., "Differentiable Particle Filtering via Entropy-regularized Optimal Transport." ICML, 2021.

Theoretical Results

- Standard PFs using multinomial resampling[17] (not differentiable):

$$\mathbb{E}\left[\left(\alpha_N^{(t)}(\psi) - \beta^{(t-1)} f(\psi)\right)^2\right] \leq c_t \frac{||\psi||_\infty^2}{N}, \quad t \geq 1,$$

$$\mathbb{E}\left[\left(\beta_N^{(t)}(\psi) - \beta^{(t)}(\psi)\right)^2\right] \leq c'_t \frac{||\psi||_\infty^2}{N}, \quad t \geq 0.$$

- What we derived for NF-DPFs[18]:

$$\mathbb{E}\left[\left(\alpha_N^{(t)}(\psi) - \beta^{(t-1)} f(\psi)\right)^2\right] \leq c_t \frac{||\psi||_\infty^2}{N^{1/2d_{\mathcal{X}}}}, \quad t \geq 1, \qquad (1)$$

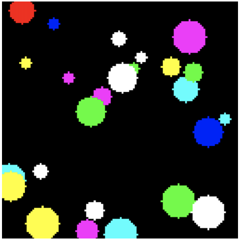
$$\mathbb{E}\left[\left(\beta_N^{(t)}(\psi) - \beta^{(t)}(\psi)\right)^2\right] \leq c'_t \frac{||\psi||_\infty^2}{N^{1/2d_{\mathcal{X}}}}, \quad t \geq 0. \qquad (2)$$

[17] Crisan and Doucet, "A Survey of Convergence Results on Particle Filtering Methods for Practitioners", IEEE TSP, 2002.

[18] Chen and Li, "Normalising Flow-based Differentiable Particle Filters", preprint, arXiv, 2403.01499, 2024.

Numerical experiments

Disk tracking experiment: localising the moving red disk[19].



- Observation o_t: an image that shows the location of disks at t.
- State s_t: the location of the red disk, $s_t = (x_t, y_t)$.

Challenges:

- High-dimensional, unstructured observations.
- Moving distractors.
- The target may disappear from the observation:
  - Occluded by distractors.
  - Out of boundaries.

[19] Kloss et al., "How to Train Your Differentiable Filter", Autonomous Robots, 2021.

Numerical experiments

Disk localisation experiment:

- Dynamic model:

$$\hat{a}_t = a_t + \epsilon_t, \quad \epsilon_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0},\ \sigma_\epsilon^2 \mathbb{I}) \ ,$$
$$s_{t+1} = s_t + \hat{a}_t + \alpha_t, \quad \alpha_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0},\ \sigma_\alpha^2 \mathbb{I}) \ .$$

- How to model the relationship between observations and states?
  - Encode o_t with neural networks: $e_t = E_\theta(o_t)$.
  - Estimate the conditional likelihood $p(o_t|s_t;\theta)$ in the encoded latent space - different methods to do this.
- Proposal: utilise the encoded feature e_t to draw samples.
- Loss function:
  - RMSE between predictions and ground truth locations: $\mathcal{L}_{\text{RMSE}}(\theta, \phi) := \sqrt{\frac{1}{T}\sum_{t=0}^{T} ||\bar{s}_t - s_t^*||_2^2}$.
  - Autoencoder loss: $\mathcal{L}_{\text{AE}}(\theta) = \frac{1}{T}\sum_{t=0}^{T} ||D_\theta(E_\theta(o_t)) - o_t||_2^2$.

Numerical experiments

Disk localisation experiment:

- Evaluated methods.
  - NF-DPF[18]: proposal and measurements constructed with normalising flows.
  - Particle filter network (PFNet)[8]: Boostrap, particle weights given by a neural network with scalar output, $p(o_t|s_t^i;\theta) \propto L_\theta(o_t, s_t^i)$.
  - AESMC-Bootstrap[11]: Bootstrap, Gaussian measurements, $o_t \sim \mathcal{N}(\mu_\theta(s_t), \sigma_\theta(s_t))$.
  - AESMC[11]: Gaussian proposal and measurement, $s_t \sim \mathcal{N}(\mu'_\theta(s_{t-1}, o_t), \sigma'_\theta(s_{t-1}, o_t))$.
  - Particle filter recurrent neural network (PFRNN)[20]: new samples and associated weights generated by RNNs, e.g. GRU and LSTM, $(s_t^i, w_t^i) = \textbf{RNN}(s_{t-1}^i, w_{t-1}^i, o_t)$.
- 500/50/50 trajectories for training/validation/testing, each trajectory has 50 time steps.
- 100 particles in both training and testing.

---

[8] Karkus et al., "Particle Filter Networks with Application to Visual localisation", CoRL, 2018.

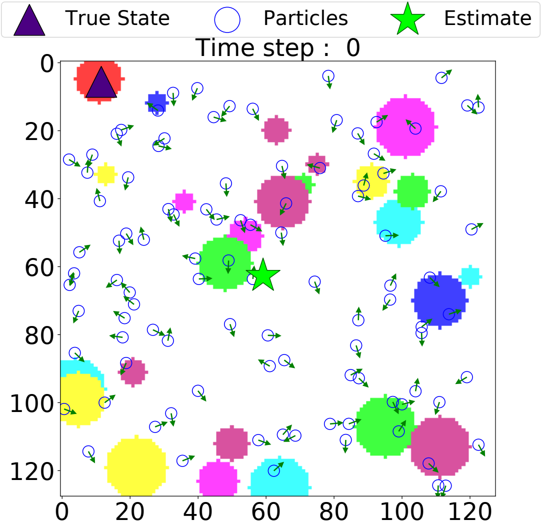
[11] Le et al., "Auto-Encoding Sequential Monte Carlo", ICLR, 2018.

[18] Chen and Li, "Normalising Flow-based Differentiable Particle Filters", preprint, arXiv, 2403.01499, 2024.

[20] Ma et al., "Particle Filter Recurrent Neural Networks", AAAI, 2020.

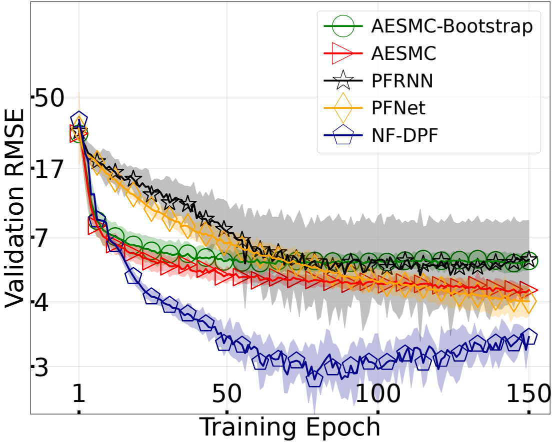
Numerical experiments

Disk localisation experiment:



Numerical experiments

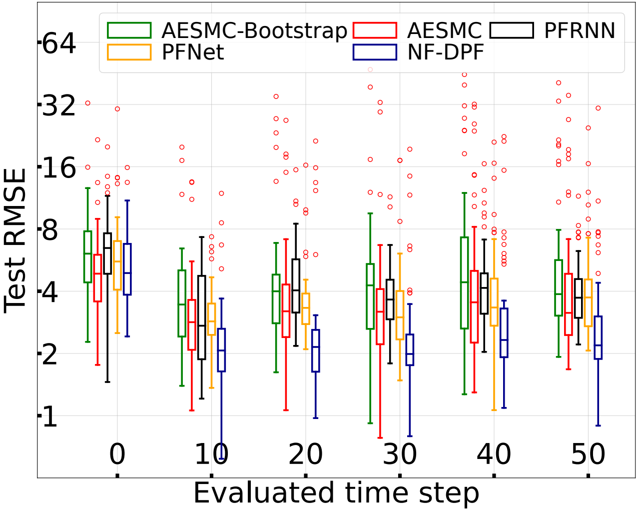
Disk localisation experiment:



Method	AESMC Bootstrap	AESMC	PFRNN	PFNet	NF-DPF
RMSE	6.35±1.15	5.85±1.34	6.12±1.23	5.34±1.27	**3.62±0.98**

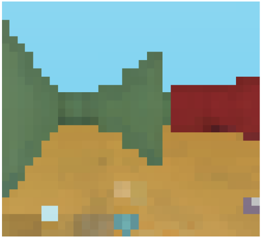
Numerical experiments

Disk tracking experiment:



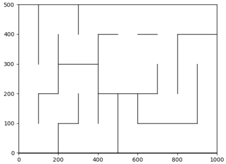
Numerical experiments

Robot localisation in a maze environment:



- Observations given by robot cameras in a simulated environment.
- State s_t: the location and the orientation of the robot, $s_t = (x_t, y_t, \varrho_t)$.

Map of the maze:



Numerical experiments

Robot localisation in a maze environment:

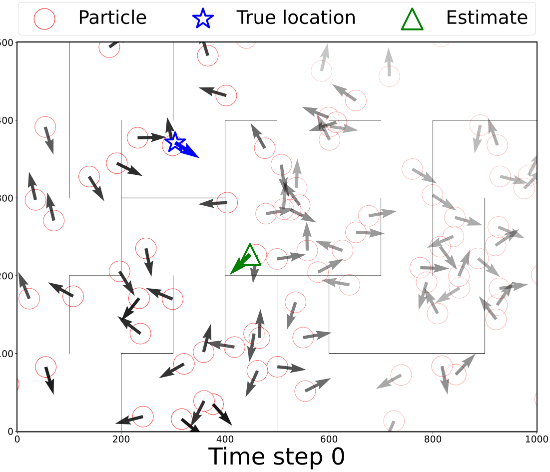
- Dynamic model:

$$s_{t+1} = \begin{bmatrix} x_t + \Delta x_t \cos(\varrho_t) + \Delta y_t \sin(\varrho_t) \\ y_t + \Delta x_t \sin(\varrho_t) - \Delta y_t \cos(\varrho_t) \\ \varrho_t + \Delta \varrho_t \end{bmatrix} + \varsigma_t$$

- Similar to the disk localisation experiment:
  - Encode observations into feature vectors to estimate $p(o_t|s_t;\theta)$ and construct proposal distributions.
  - Loss function: consist of RMSE loss and autoencoder loss.
- More difficult than disk localisation:
  - Uninformative observations.
  - Need to consider the orientation of the object.
- 900/100/100 trajectories for training/validation/testing, each trajectory has 100 time steps.
- 100 particles in both training and testing.

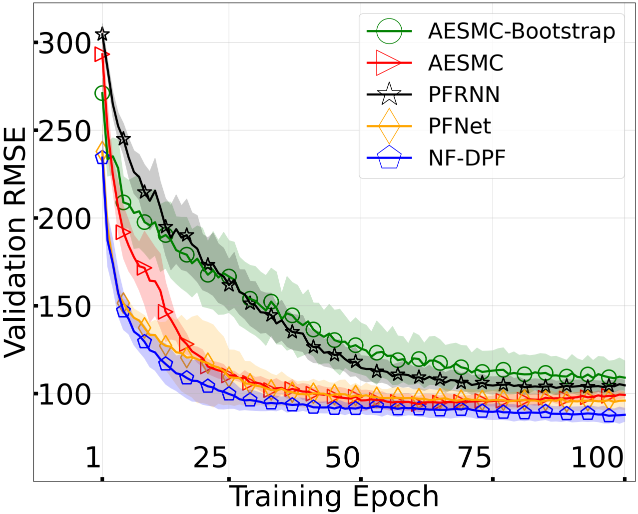
Numerical experiments

Robot localisation in a maze environment:



Numerical experiments

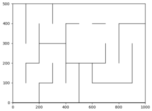
Robot localisation in a maze environment:



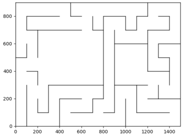
Numerical experiments

Robot localisation in a maze environment:

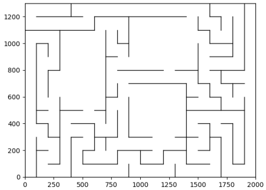
- We tested in three different maze environments.



Maze 1



Maze 2



Maze 3

	Method				
	AESMC Bootstrap	AESMC	PFNet	PFRNN	NF-DPF
Maze 1	56.5±11.5	52.1±7.5	51.4±8.7	54.1±8.9	**46.1±6.9**
Maze 2	115.6±6.8	109.2±11.7	120.3±8.0	125.1±8.2	**103.2±10.8**
Maze 3	220.6±11.1	201.3±14.7	212.1±15.3	210.5±10.8	**182.2±19.9**

Summary[18]

- Introduce a normalising flow-based differentiable particle filters that construct flexible, valid dynamic models and proposal distributions and measurement models.
- The proposed method can serve as a "plug-in" module in existing differentiable particle filter frameworks.
- NF-DPFs are differentiable and consistent.

[18] Chen and Li, "Normalising Flow-based Differentiable Particle Filters", preprint, arXiv, 2403.01499, 2024.