Some Initial Explorations of Differentiable Particle Filters

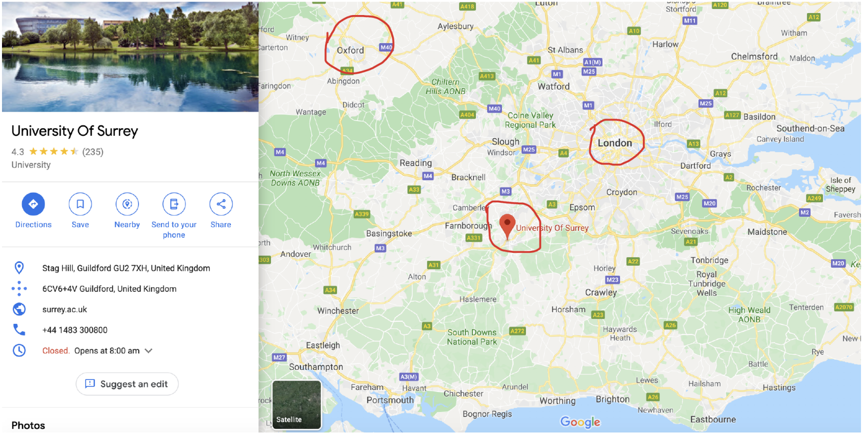
Yunpeng Li

Collaborators: Xiongjie Chen, Hao Wen, Georgios Papagiannis, Conghui Hu

Department of Computer Science

University of Surrey, UK

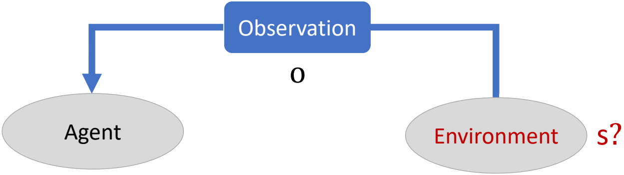
Bellairs Workshop 2021


University Of Surrey
4.3 (235)
University
Directions
Save
Nearby
Send to your phone
Share
Stag Hill, Guildford GU2 7XH, United Kingdom
6CV6+4V Guildford, United Kingdom
surrey.ac.uk
+44 1483 300800
Closed. Opens at 8:00 am
Suggest an edit
Photos
Oxford
London
University Of Surrey
Aylesbury
St Albans
Harlow
Chelmsford
Witney
Abingdon
Didcot
Wantage
Watford
Slough
Windsor
Reading
Bracknell
Croydon
Epsom
Camberley
Farnborough
Basingstoke
Andover
Whitchurch
Alton
Haslemere
Winchester
Eastleigh
Southampton
Fareham
Havant
Portsmouth
Chichester
Bognor Regis
Arundel
Worthing
Brighton
Horsham
Crawley
Horley
Haywards Heath
Lewes
Newhaven
Eastbourne
Hailsham
Hastings
Battle
Rye
Tonbridge
Royal Tunbridge Wells
Sevenoaks
Maidstone
Rochester
Dartford
Grays
Basildon
Southend-on-Sea
Canvey Island
Isle of Sheppey
Ilford
Witham
Maldon
Tenterden
High Weald AONB
South Downs National Park
Chiltern Hills AONB
Colne Valley Regional Park
North Wessex Downs AONB
Satellite
Google

Outline

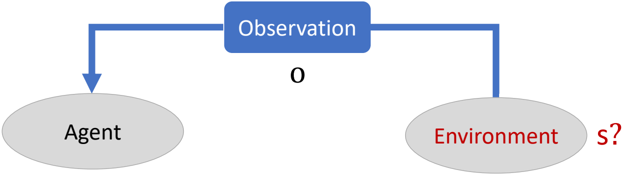
- Motivation
- Semi-supervised differentiable particle filters
- Differentiable particle filters through normalizing flow
- Future directions

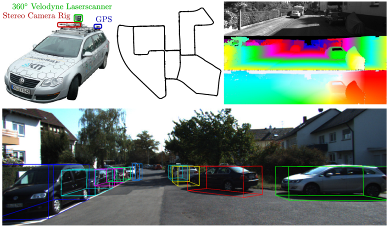
Background



Motivating Examples: Autonomous driving[1]

Radar, Lidar, GPS, Camera measurements

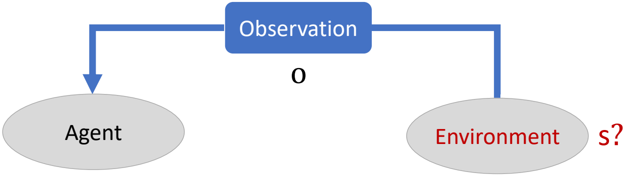


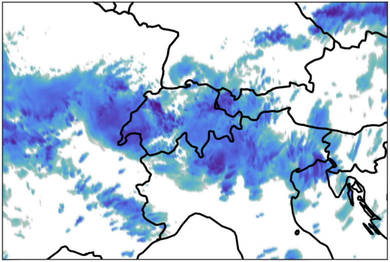


[1] Geiger et al., "Are we ready for autonomous driving? The KITTI vision benchmark suite", CVPR, 2012

Motivating Examples: Weather forecasting[2]

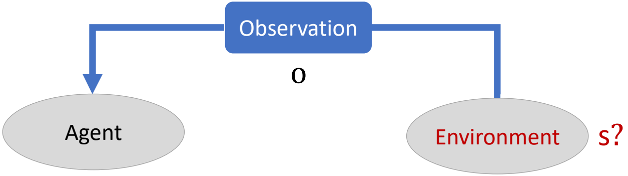
Weather station measurements
(Thermometer, Barometer, Hygrometer, Anemometer, etc.)





[2] Robert et al., "A local ensemble transform Kalman particle filter for convective scale data assimilation", J. Royal Meteorological Society 2018

Bayesian Learning



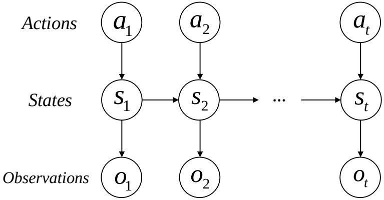
Prior: $p(s)$

Likelihood: $p(o|s)$

Posterior: $p(s|o) = \frac{p(s)p(o|s)}{\int p(o|s')p(s')\,\mathrm{d}s'}$

Filtering/Target Tracking Problem Formulation

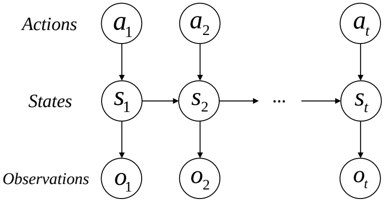
Recursive Bayesian Filtering: when the state and observation are sequence data.



- Dynamic model $p_\theta(s_t|s_{t-1}, a_t)$: transition of hidden state.
- Measurements model $p_\theta(o_t|s_t)$: likelihood of the observation given the state.

Filtering/Target Tracking Problem Formulation

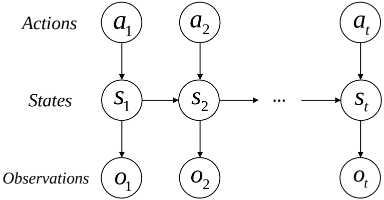
Recursive Bayesian Filtering: when the state and observation are sequence data.



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

Filtering/Target Tracking Problem Formulation

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

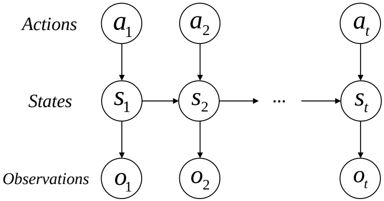


- Dynamic model $p_\theta(s_t|s_{t-1}, a_t)$: transition of hidden state.
- Measurements model $p_\theta(o_t|s_t)$: likelihood of the observation given the state.
- Goal: obtain marginal posterior $p_\theta(s_t|o_{1:t}, a_{1:t})$ or joint posterior $p_\theta(s_{1:t}|o_{1:t}, a_{1:t})$.

Linear Gaussian models: Kalman filters.

Filtering/Target Tracking Problem Formulation

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



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

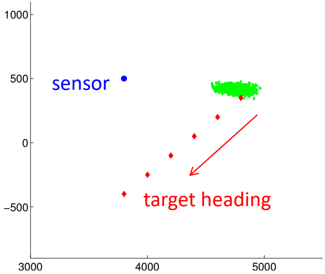
Linear Gaussian models: Kalman filters.
Non-linear non-Gaussian models: Particle filters.

(Bootstrap) Particle Filters in One Slide

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

(Bootstrap) Particle Filters in One Slide

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

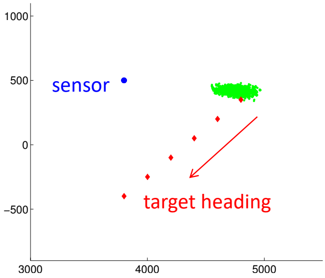


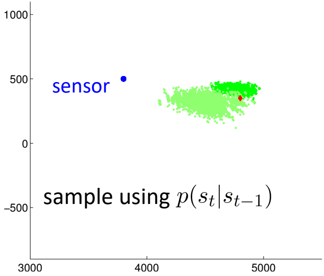
Use particle approximation of target state posterior

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

(Bootstrap) Particle Filters in One Slide

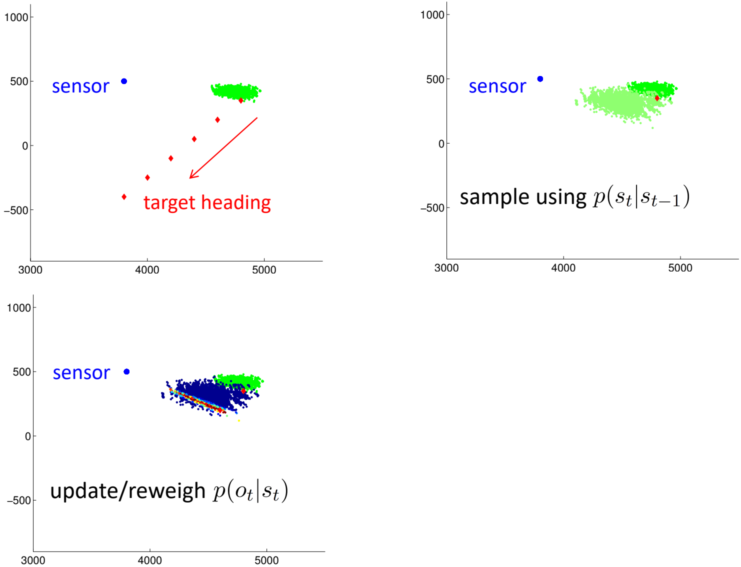
- Particle filters, a.k.a. sequential Monte Carlo (SMC) methods: Weighted samples to sequentially approximate target distribution.





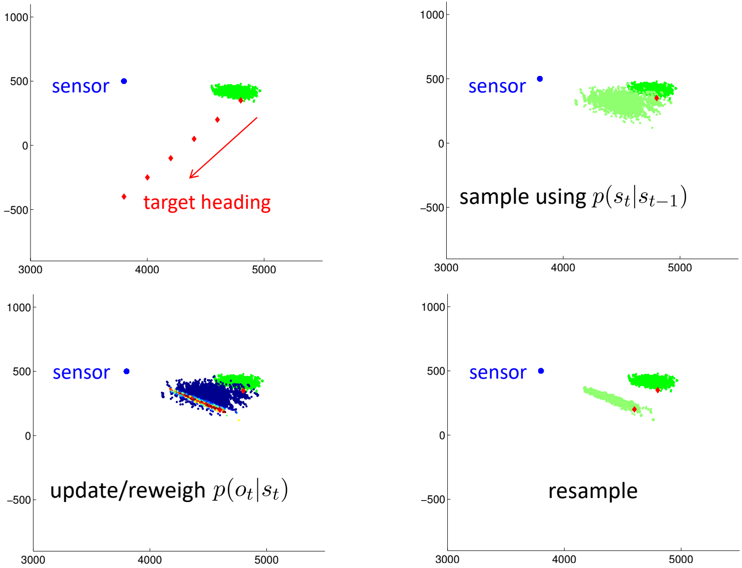
(Bootstrap) Particle Filters in One Slide

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

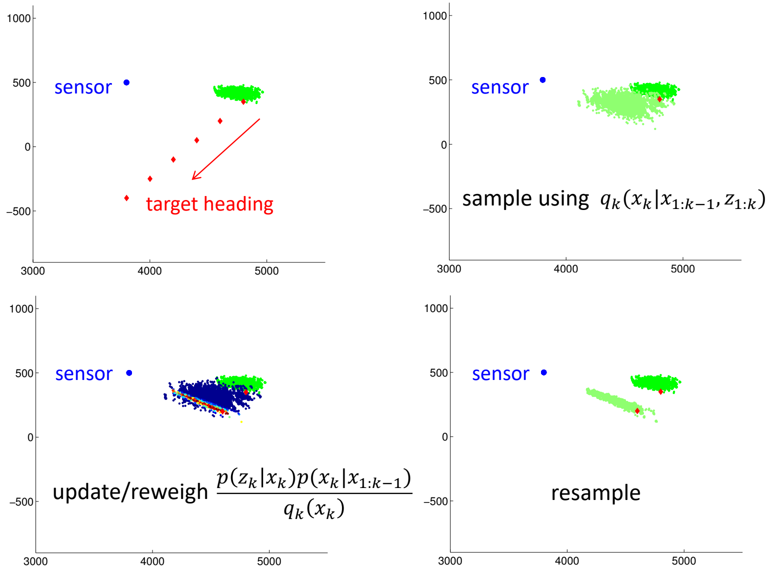


(Bootstrap) Particle Filters in One Slide

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



Particle Filters: more generally



Parameter Estimation for Particle Filtering

Can we learn the parameters of particle filters from data?

- Maximum likelihood (ML) estimation[3]
- Bayesian estimation[4]

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

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

Parameter Estimation for Particle Filtering

Can we learn the parameters of particle filters from data?

- Maximum likelihood (ML) estimation[3]
- Bayesian estimation[4]

Can be effective, but ...

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

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

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

Basic Idea of Differentiable Particle Filters (DPFs)

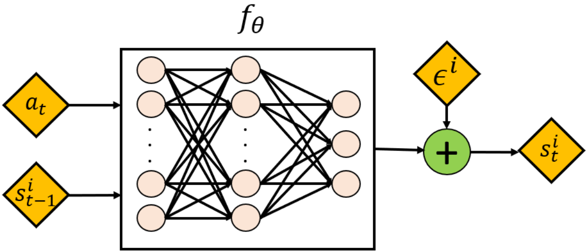
Combining particle filters with deep learning tools: Differentiable particle filters[5].

- Build dynamic model and measurement model with neural networks;
- Optimize the networks with gradient descent.

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

Differentiable Particle Filters: How?

Parameterise the dynamic and measurement model with neural networks.



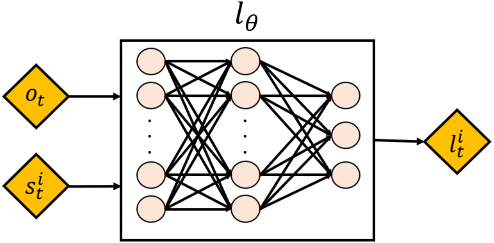
Dynamic model

$$s_t \sim p_\theta(s_t|s_{t-1}^i, a_t), s_t^i = f_\theta(s_{t-1}^i, a_t) + \epsilon^i$$

[5] Jonschkowski et al., "Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors", RSS, 2018.
[6] Karkus et al., "Particle Filter Networks with Application to Visual Localization", CoRL, 2018.

Differentiable Particle Filters: How?

Parameterise the dynamic and measurement model with neural networks.



Measurement model

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

[5] Jonschkowski et al., "Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors", RSS, 2018.
[6] Karkus et al., "Particle Filter Networks with Application to Visual Localization", CoRL, 2018.

Differentiable Particle Filters: How?

Loss function[7]:

- The mean squared error (MSE)[6]:

$$L_{MSE}(\theta) = \frac{1}{T}\sum_{t=1}^{T}(s_t^* - \hat{s}_t)^T(s_t^* - \hat{s}_t)\,,$$

- The negative log likelihood (NLL)[5]:

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

where s_t^* is the ground truth state, $\hat{s}_t$ is the estimated state.

[5] Jonschkowski et al., "Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors", RSS, 2018.
[6] Karkus et al., "Particle Filter Networks with Application to Visual Localization", CoRL, 2018.
[7] Kloss et al., "How to Train Your Differentiable Filter", arXiv:2012.14313, 2020.

Existing DPFs and Research Questions

- Require ground truth state information.

Existing DPFs and Research Questions

- Require ground truth state information.

  Q1: Can we train DPFs with a reduced demand for labelled data?

Existing DPFs and Research Questions

- Require ground truth state information.
  Q1: Can we train DPFs with a reduced demand for labelled data?
- Only able to generate Gaussian prior or intractable non-Gaussian prior[5].

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

Existing DPFs and Research Questions

- Require ground truth state information.

  Q1: Can we train DPFs with a reduced demand for labelled data?

- Only able to generate Gaussian prior or intractable non-Gaussian prior[5].

  Q2: Can we build flexible and tractable priors other than Gaussian?

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

Existing DPFs and Research Questions

- Require ground truth state information.

  Q1: Can we train DPFs with a reduced demand for labelled data?

- Only able to generate Gaussian prior or intractable non-Gaussian prior[5].

  Q2: Can we build flexible and tractable priors other than Gaussian?

- Bootstrap Particle Filtering framework or particle proposal schemes that use latest observation but ignore state[5].

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

Existing DPFs and Research Questions

- Require ground truth state information.

  Q1: Can we train DPFs with a reduced demand for labelled data?

- Only able to generate Gaussian prior or intractable non-Gaussian prior[5].

  Q2: Can we build flexible and tractable priors other than Gaussian?

- Bootstrap Particle Filtering framework or particle proposal schemes that use latest observation but ignore state[5].

  Q3: Can we construct flexible and tractable proposals based on latest observations?

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

Existing DPFs and Research Questions

- Require ground truth state information.

  Q1: Can we train DPFs with a reduced demand for labelled data?

- Only able to generate Gaussian prior or intractable non-Gaussian prior[5].

  Q2: Can we build flexible and tractable priors other than Gaussian?

- Bootstrap Particle Filtering framework or particle proposal schemes that use latest observation but ignore state[5].

  Q3: Can we construct flexible and tractable proposals based on latest observations?

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

H. Wen, X. Chen, G. Papagiannis, C. Hu, and Y. Li, "End-to-end semi-supervised learning for differentiable particle filters," ICRA 2021.

Maximum likelihood estimation

- ML estimation: recursively maximise the series of likelihoods $p_\theta(o_{1:t}|a_{1:t})$
- However ...

Maximum likelihood estimation

- ML estimation: recursively maximise the series of likelihoods $p_\theta(o_{1:t}|a_{1:t})$
- However ...
- The dimension of $p_\theta(o_{1:t}|a_{1:t})$ will increase over time.

Pseudo-likelihood

- "Divide" the log-likelihoods into blocks.

$$\log p_\theta(o_{1:t}|a_{1:t}) \longrightarrow \sum_{b=0}^{m-1} \log p_\theta(O_b|A_b)$$
$$O_b = o_{bL+1:(b+1)L} \text{ and } A_b = a_{bL+1:(b+1)L}$$
m: number of blocks, b: block index, L: block length

[8] Andrieu et al. "On-line parameter estimation in general state-space models", CDC, 2005.

Pseudo-likelihood

- "Divide" the log-likelihoods into blocks.

$$\log p_\theta(o_{1:t}|a_{1:t}) \longrightarrow \sum_{b=0}^{m-1} \log p_\theta(O_b|A_b)$$
$$O_b = o_{bL+1:(b+1)L} \text{ and } A_b = a_{bL+1:(b+1)L}$$
m: number of blocks, b: block index, L: block length

The log pseudo-likelihood for a block $\log p_\theta(O|A)$:

- Marginalise the joint distribution $p_\theta(S, O|A)$

$$\log p_\theta(O|A) = \log \int_{S^L} p_\theta(S, O|A) dS$$

[8] Andrieu et al. "On-line parameter estimation in general state-space models", CDC, 2005.

Pseudo-likelihood

- If all S observed, learning is relatively easy

$$p_\theta(S,O|A) = \underbrace{p_\theta(S|A)}_{\text{Dynamic model}} \cdot \underbrace{p_\theta(O|S,A)}_{\text{Measurement model}}$$

Pseudo-likelihood

- If all S observed, learning is relatively easy

$$p_\theta(S,O|A) = \underbrace{p_\theta(S|A)}_{\text{Dynamic model}} \cdot \underbrace{p_\theta(O|S,A)}_{\text{Measurement model}}$$

- If S not observed, use the θ_b to get the posterior of S at current block $p_{\theta_b}(S|O,A)$

$$\int_{S^L} \log(p_\theta(S,O|A))p_{\theta_b}(S|O,A)dS$$

Semi-supervised differentiable particle filters

- Optimisation objective for samples without true labels

$$\hat{Q}(\theta, \theta_b) = \sum_{i=1}^{N_p} w_b^i \log p_\theta(S_b^i, O_b | A_b)$$

Semi-supervised differentiable particle filters

- Optimisation objective for samples without true labels

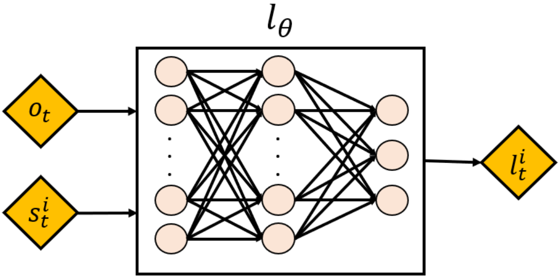
$$\hat{Q}(\theta, \theta_b) = \sum_{i=1}^{N_p} w_b^i \log p_\theta(S_b^i, O_b | A_b)$$

- Learning objective for semi-supervised learning:

$$\theta = \underset{\theta \in \Theta}{\arg\min}\, \lambda_1 L(\theta) - \lambda_2 Q(\theta)$$

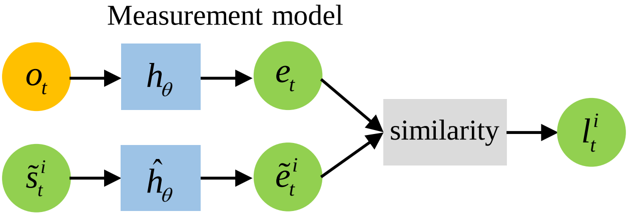
$$Q(\theta) = \frac{1}{m} \sum_{b=0}^{m-1} \hat{Q}(\theta, \theta_b)$$

Recall the measurement model



[Credit: Dutch Creatives]

Solution

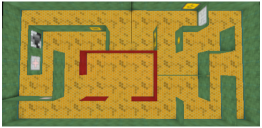


$l_t^i = l_\theta(o_t|\tilde{s}_t^i)$ $\quad$ $w_t^i = l_t^i w_{t-1}^i$

Maze environment[9]

Robot localisation.

- Top-down view of Maze 1.

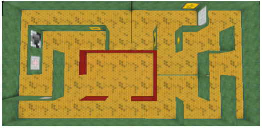


[9] Beattie et al. DeepMind Lab, 2018.

Maze environment[9]

Robot localisation.

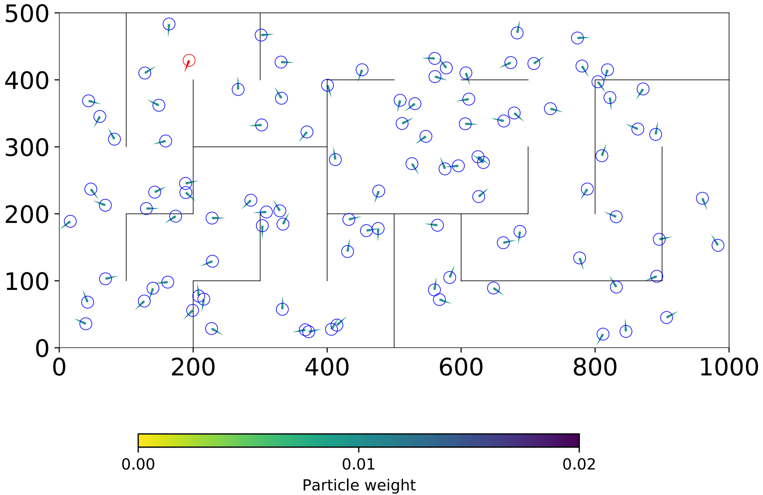
- Top-down view of Maze 1.



- Example observation images.

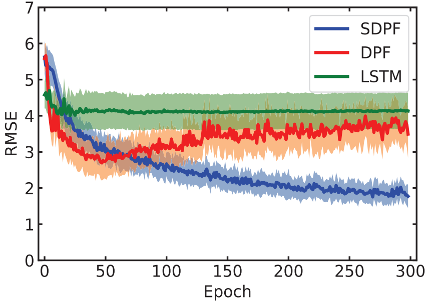
[9] Beattie et al. DeepMind Lab, 2018.

Tracking Demo (100 Particles)



Maze environment

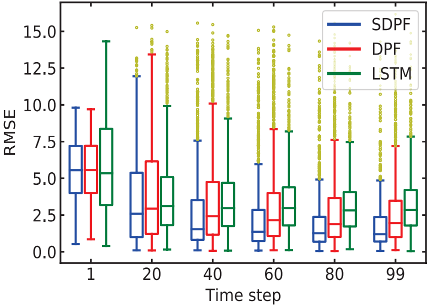
- SDPF converges to the lowest RMSE during training process.



Maze 1

Maze environment

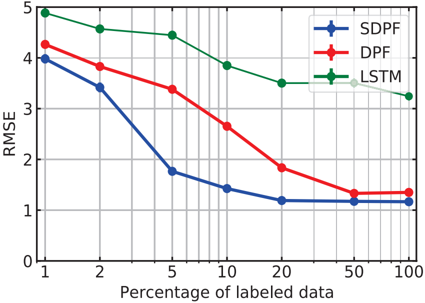
- SDPF improves tracking performance on testing trajectories.



Maze 1

Maze environment

- SDPF is robust to a wide range of percentage of labelled data.



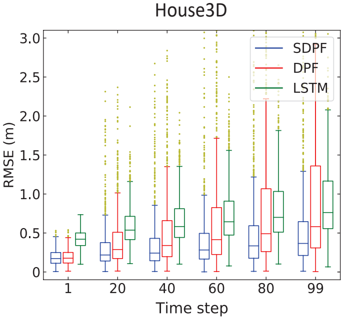
Maze 1

House3D environment[10]

[10] Yi et al. Building generalisable agents with a realistic and rich 3D environment, 2018

House3D environment

- SDPF can generalise to different environments.



Research Questions

1. Can we train DPFs with a reduced demand for labelled data?
2. Can we build flexible and tractable priors other than Gaussian?
3. Can we construct flexible and tractable proposals based on latest observations?

Research Questions

1. Can we train DPFs with a reduced demand for labelled data?
2. Can we build flexible and tractable priors other than Gaussian?
3. Can we construct flexible and tractable proposals based on latest observations?

Challenges:

- Vanilla neural networks do not allow density estimation.

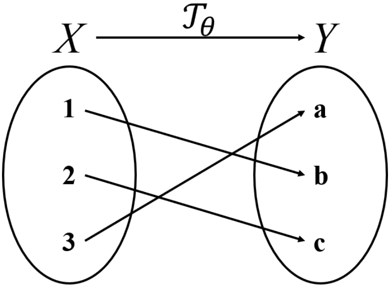
X. Chen, H. Wen, and Y. Li, "Differentiable Particle Filters through Conditional Normalizing Flow," FUSION 2021.

Normalizing Flows

Definition of normalizing flows:

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

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



Normalizing Flows

Definition of normalizing flows:

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

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

Why invertible transformations?

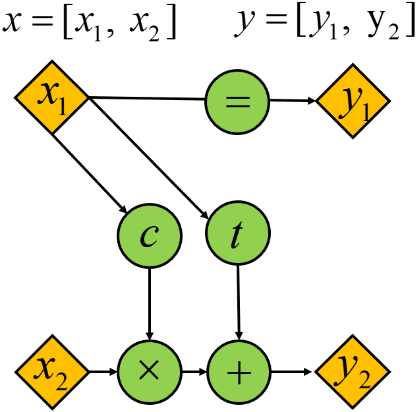
- Invertibility allows density estimation (change of variable):

$$p(y) = p(x)\left|\mathsf{det}\frac{dy}{dx}\right|^{-1}$$

An Example of Normalizing Flow: Coupling Layer

Real-NVP[11]

- Coupling layers.



[11] Dinh et al. "Density Estimation Using Real NVP", ICLR, 2017.

An Example of Normalizing Flow: Coupling Layer

Real-NVP[11]

- Coupling layers.

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

$$\underset{1:d}{y} = \underset{1:d}{x}$$

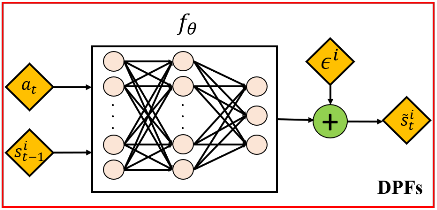
$$\underset{d+1:D}{y} = \underset{d+1:D}{x} \odot \exp(c(\underset{1:d}{x})) + t(\underset{1:d}{x})$$

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

[11] Dinh et al. "Density Estimation Using Real NVP", ICLR, 2017.

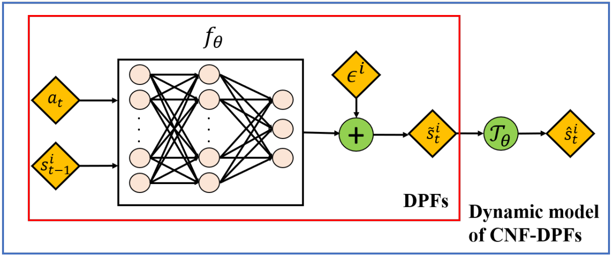
Construct Flexible Dynamic Model through Normalizing Flow

Solution to Question 2



Construct Flexible Dynamic Model through Normalizing Flow

Solution to Question 2



- Normalizing flow $\mathcal{T}_\theta(\cdot)$: construct flexible dynamic models.

Research Questions

2. Can we build flexible and tractable priors other than Gaussian?

Challenge:

- Vanilla neural networks do not allow density estimation. (Resolved)

Research Questions

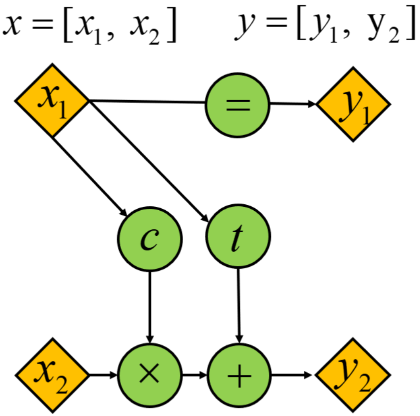
2. Can we build flexible and tractable priors other than Gaussian?
3. Can we construct flexible and tractable proposals based on latest observations?

Challenge:

- Vanilla neural networks do not allow density estimation. (Resolved)
- Normalizing flows allow density estimation but require the input and output to have the same dimensionality. (?)

Conditional Coupling Layer

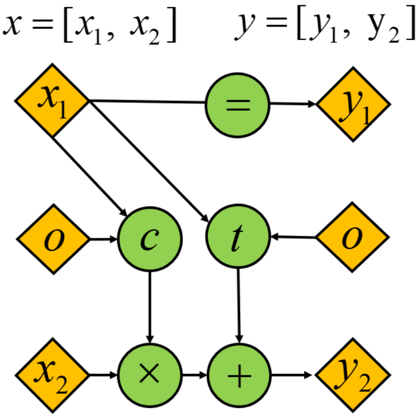
We use conditional coupling layer to construct conditional Real-NVP:



Standard coupling layer

Conditional Coupling Layer

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



Conditional coupling layer

Conditional Coupling Layer: Solution to Challenge 2

- Conditional coupling layer:

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

- Standard coupling layer:

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

Conditional Coupling Layer: Solution to Challenge 2

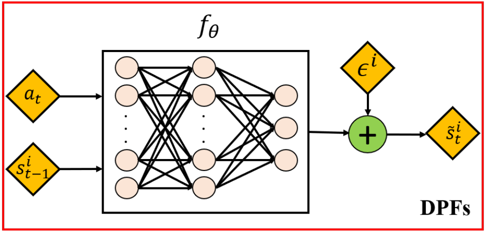
- Conditional coupling layer:

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

Still invertible and lead to triangular Jacobian matrix:

$$\frac{\partial s}{\partial \hat{s}} = \begin{bmatrix} \mathbb{I} & 0 \\ \frac{\partial s_{d+1:D}}{\partial \hat{s}_{1:d}} & \mathsf{diag}(\exp[c(\hat{s}_{1:d}, o)]) \end{bmatrix}$$

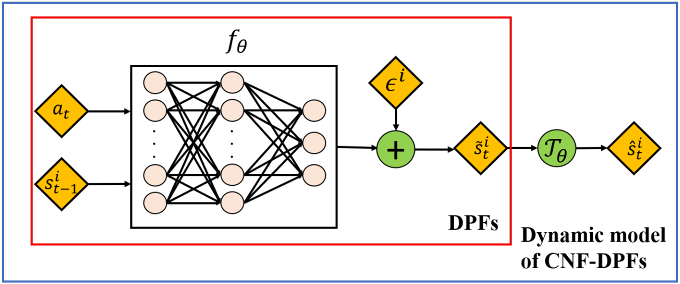
The Structure of CNF-DPFs



Challenges:

1. Vanilla neural networks do not allow density estimation. (Resolved)

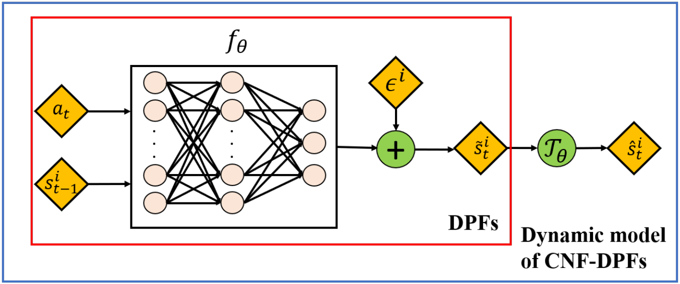
The Structure of CNF-DPFs



Challenges:

1. Vanilla neural networks do not allow density estimation. (Resolved)
   Solution: normalizing flows.

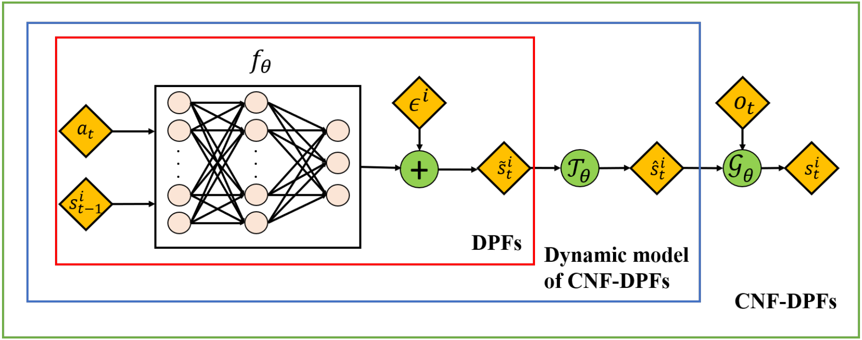
The Structure of CNF-DPFs



Challenges:

1. Vanilla neural networks do not allow density estimation. (Resolved)
   Solution: normalizing flows.
2. Normalizing flows allow density estimation but require the input and output to have the same dimensionality. (Resolved)

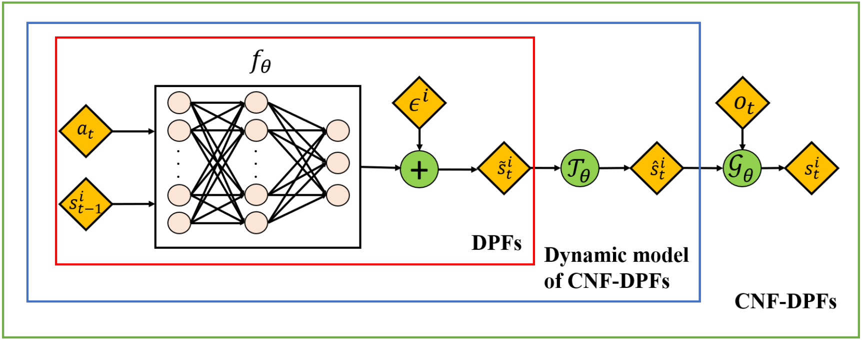
The Structure of CNF-DPFs



Challenges:

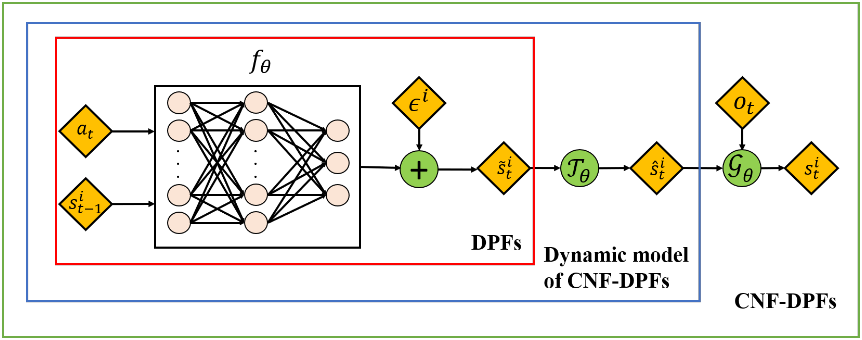
1. Vanilla neural networks do not allow density estimation. (Resolved)
   Solution: normalizing flows.
2. Normalizing flows allow density estimation but require the input and output to have the same dimensionality. (Resolved)
   Solution: conditional normalizing flows.

The Structure of CNF-DPFs



1. Normalizing flow $\mathcal{T}_\theta(\cdot)$: construct flexible dynamic models.

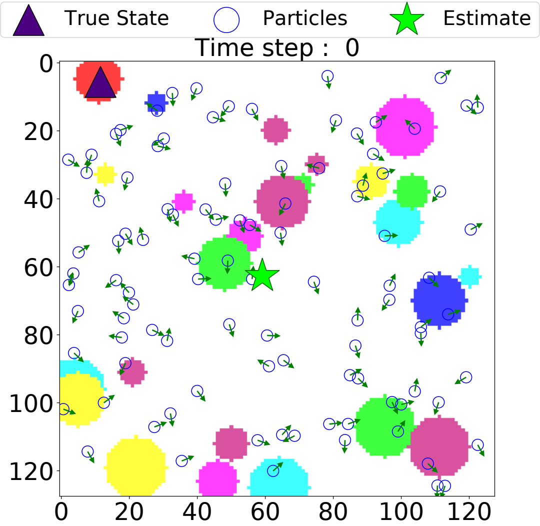
The Structure of CNF-DPFs



1. Normalizing flow $\mathcal{T}_\theta(\cdot)$: construct flexible dynamic models.
2. Conditional normalizing flow $\mathcal{G}_\theta(\cdot)$: move particles to areas closer to posterior by utilizing information from observations.

Numerical Experiment

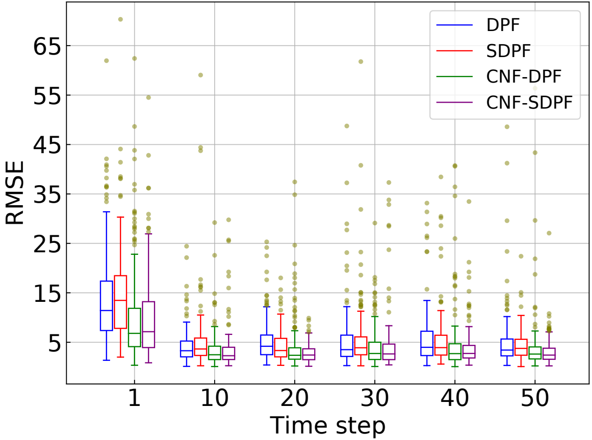
Disk tracking experiment[12,7]:



[12] Haarnoja et al., "Backprop KF: Learning Discriminative Deterministic State Estimators", NeurIPS 2016.
[7] Kloss et al., "How to Train Your Differentiable Filter", arXiv:2012.14313, 2020.

Numerical Experiment

Test RMSE between prediction and true state, particles are initialized uniformly:



DPF: differentiable particle filter

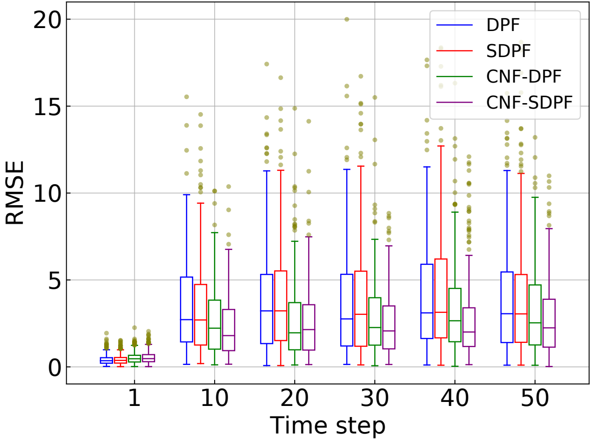
SDPF: semi-supervised DPF

CNF-DPF: conditional normalizing flow DPF

CNF-SDPF: conditional normalizing flow semi-supervised DPF

Numerical Experiment

Test RMSE between prediction and true state, particles are initialized around the true state:



DPF: differentiable particle filter

SDPF: semi-supervised DPF

CNF-DPF: conditional normalizing flow DPF

CNF-SDPF: conditional normalizing flow semi-supervised DPF

Summary

- A learning objective based upon the maximisation of a pseudo-likelihood function to use unlabelled observations.
- A mechanism to incorporate normalizing flows into DPFs to construct flexible and tractable prior and proposal.
- Can serve as "plug-in" modules in existing DPF pipelines.
- Improved performance through numerical experiments.

Future directions

- Online learning.

Future directions

- Online learning.
- Better dynamic models – e.g. mixture models?

Future directions

- Online learning.
- Better dynamic models – e.g. mixture models?
- Better measurement models.

Future directions

- Online learning.
- Better dynamic models – e.g. mixture models?
- Better measurement models.
- Better proposal distributions – e.g. APF with normalizing flow, physics-inspired NNs?

Future directions

- Online learning.
- Better dynamic models – e.g. mixture models?
- Better measurement models.
- Better proposal distributions – e.g. APF with normalizing flow, physics-inspired NNs?
- Better optimisation objectives – existing VI-based, self-supervised methods?

Future directions

- Online learning.
- Better dynamic models – e.g. mixture models?
- Better measurement models.
- Better proposal distributions – e.g. APF with normalizing flow, physics-inspired NNs?
- Better optimisation objectives – existing VI-based, self-supervised methods?
- Differentiable resampling schemes[13].

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

Future directions

- Online learning.
- Better dynamic models – e.g. mixture models?
- Better measurement models.
- Better proposal distributions – e.g. APF with normalizing flow, physics-inspired NNs?
- Better optimisation objectives – existing VI-based, self-supervised methods?
- Differentiable resampling schemes[13].
- Distributed learning and inference.

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

Future directions

- Online learning.
- Better dynamic models – e.g. mixture models?
- Better measurement models.
- Better proposal distributions – e.g. APF with normalizing flow, physics-inspired NNs?
- Better optimisation objectives – existing VI-based, self-supervised methods?
- Differentiable resampling schemes[13].
- Distributed learning and inference.
- Continuous-time filtering.

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

Future directions

- Online learning.
- Better dynamic models – e.g. mixture models?
- Better measurement models.
- Better proposal distributions – e.g. APF with normalizing flow, physics-inspired NNs?
- Better optimisation objectives – existing VI-based, self-supervised methods?
- Differentiable resampling schemes[13].
- Distributed learning and inference.
- Continuous-time filtering.

Thank you!

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