Conditional Measurement Density Estimation in Sequential Monte Carlo via Normalizing Flow

Xiongjie Chen
Department of Computer Science
University of Surrey
Guildford, UK
xiongjie.chen@surrey.ac.uk

Yunpeng Li
Department of Computer Science
University of Surrey
Guildford, UK
yunpeng.li@surrey.ac.uk

***Abstract*—Tuning of measurement models is challenging in real-world applications of sequential Monte Carlo methods. Recent advances in differentiable particle filters have led to various efforts to learn measurement models through neural networks. But existing approaches in the differentiable particle filter framework do not admit valid probability densities in constructing measurement models, leading to incorrect quantification of the measurement uncertainty given state information. We propose to learn expressive and valid probability densities in measurement models through conditional normalizing flows, to capture the complex likelihood of measurements given states. We show that the proposed approach leads to improved estimation performance and faster training convergence in a visual tracking experiment.**

***Index Terms*—likelihood learning, conditional normalizing flow, sequential Monte Carlo methods**

I. Introduction

In this section we first provide a brief introduction to the conditional density estimation (CDE) problem and sequential Monte Carlo (SMC) methods. We then illustrate the role of conditional measurement density estimation in SMC methods, in particular differentiable particle filters (DPFs), and discuss their limitations. We summarize our contributions and outline the structure of this paper at the end of this section.

A. Conditional Density Estimation

Conditional density estimation (CDE) refers to the problem of modeling the conditional probability density $p(\mathbf{y}|\mathbf{x})$ for a dependent variable $\mathbf{y}$ given a conditional variable $\mathbf{x}$. In contrast to tasks targeting the conditional mean $\mathbb{E}_{p(\mathbf{y}|\mathbf{x})}[\mathbf{y}]$, CDE aims to model the full conditional density $p(\mathbf{y}|\mathbf{x})$. Applications of CDE can be found in various domains including computer vision [1], econometrics [2], and reinforcement learning [3].

Most CDE approaches can be classified into two categories, parametric CDE models and non-parametric CDE models. With parametric models, an important assumption is that the conditional probability density $p(\mathbf{y}|\mathbf{x})$ belongs to a family of probability distributions $\mathcal{F} := \{p(\cdot|\mathbf{x};\theta)|\theta \in \Theta\}$, where the distribution $p(\cdot|\mathbf{x};\theta)$ is uniquely determined by a parameter set θ and a conditional variable $\mathbf{x}$. In particular, $p(\mathbf{y}|\mathbf{x};\theta)$ is often formulated as a parameterized function $l_\theta(\mathbf{y},\ \mathbf{x})$ whose parameters θ are learned from data by minimizing the negative log likelihood (NLL) of training data. Examples of parametric CDE models include the conditional normalizing flow (CNF) [4], [5], the conditional variational autoencoder (CVAE) [1], the mixture density network (MDN) [6], and the bottleneck conditional density estimator (BCDE) [7]. In contrast, non-parametric CDE models do not impose any parametric restrictions on the distribution family that $p(\mathbf{y}|\mathbf{x})$ belongs to. So, theoretically they can approximate arbitrarily complex conditional densities [8]. Specifically, non-parametric CDE approaches often describe the distribution $p(\mathbf{y}|\mathbf{x})$ by specifying a kernel, e.g. the Gaussian kernel, to each training data sample. However, most non-parametric CDE models assume $p(\mathbf{y}|\mathbf{x})$ to be smooth, and require traversing the entire training set to make a single prediction [9]. Examples of non-parametric CDE models include the kernel mixture network (KMN) [8], the conditional kernel density estimation (CKDE) [10], and the Gaussian process conditional density estimation [11].

B. Likelihood Learning in Sequential Monte Carlo methods

While CDE models have been employed in many practical applications, in this work we focus on CDE models to describe the relation between hidden states and observations in sequential Monte Carlo methods, i.e. measurement models in particle filters.

We first introduce the problem setup. Sequential Monte Carlo methods, a.k.a. particle filters, are a set of powerful and flexible simulation-based methods designed to numerically solve sequential state estimation problems [12]. The sequential state estimation problem we consider here is characterized by an unobserved state $\{\mathbf{x}_t\}_{t\geq 0}$ defined on $\mathcal{X} \subseteq \mathbb{R}^{d_\mathbf{x}}$ and an observation $\{\mathbf{y}_t\}_{t\geq 1}$ defined on $\mathcal{Y} \subseteq \mathbb{R}^{d_\mathbf{y}}$. The evolution of $\mathbf{x}_t$ and the relationship between $\mathbf{y}_t$ and $\mathbf{x}_t$ can be described by the following Markov process:

$$\mathbf{x}_0 \sim \pi(\mathbf{x}_0), \tag{1}$$

$$\mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{a}_t;\theta) \text{ for } t \geq 1, \tag{2}$$

$$\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t;\theta) \text{ for } t \geq 1, \tag{3}$$

where $\theta \in \Theta$ denotes the parameter set of interest, $\pi(\cdot)$ is the initial distribution, $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{a}_t;\theta)$ is the dynamic model, $\mathbf{a}_t$ is the action, $p(\mathbf{y}_t|\mathbf{x}_t;\theta)$ is the measurement model estimating the likelihood [1] of $\mathbf{y}_t$ given $\mathbf{x}_t$. Denote by $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, \cdots, \mathbf{x}_t\}$,

[1] For brevity, we use the term "likelihood" and "conditional measurement density" interchangeably for the rest of the paper.

$\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, \cdots, \mathbf{y}_t\}$ and $\mathbf{a}_{1:t} \triangleq \{\mathbf{a}_1, \cdots, \mathbf{a}_t\}$ sequences of hidden states, observations and actions up to time step t, respectively. Our goal is to simultaneously track the joint posterior distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\mathbf{a}_{1:t};\theta)$ or the marginal posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t},\mathbf{a}_{1:t};\theta)$ and learn the parameter θ from data. However, except in some trivial cases such as linear Gaussian models, the posterior is usually analytically intractable.

Particle filters approximate the analytically intractable posteriors with empirical distributions consisting of sets of weighted samples:

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\mathbf{a}_{1:t};\theta) \approx \sum_{i=1}^{N_p} w_t^i \delta_{\mathbf{x}_{0:t}^i}(d\mathbf{x}_{0:t}), \tag{4}$$

where N_p is the number of particles, $\delta_{\mathbf{x}_{0:t}^i}(\cdot)$ denotes the Dirac measure located in $\mathbf{x}_{0:t}^i$, w_t^i is the importance weight of the i-th particle at time step t. Sampled from the initial distribution $\pi(\mathbf{x}_0)$ at time step $t=0$ and proposal distributions $q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_t,\mathbf{a}_t;\theta)$ at time step $t \geq 1$, the importance weight w_t^i of the i-th particle $\mathbf{x}_{0:t}^i$ can be computed recursively for each time step:

$$w_t^i \propto w_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{x}_t^i;\theta)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i,\mathbf{a}_t;\theta)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i,\mathbf{y}_t,\mathbf{a}_t;\theta)}, \text{ for } t \geq 1, \tag{5}$$

with $w_0^i = \frac{1}{N_p}$.

The evaluation of the likelihood $p(\mathbf{y}_t|\mathbf{x}_t^i; \theta)$ is required at every time step in SMC methods as shown in Eq. (5). Traditionally, the measurement model needs to be handcrafted. However, when observations are high-dimensional and unstructured, e.g. images, the specification of measurement models for SMC methods can be challenging and often relies on extensive domain knowledge. Estimation of model parameters in particle filters is an active research area [2], [13], [14]. Many existing approaches are restricted to models where the model structure or a subset of model parameters is known [15].

Differentiable particle filters (DPFs) were proposed to alleviate the challenges in handcrafting the dynamic models and measurement models in SMC methods [16]–[20]. Particularly, the measurement model in DPFs is learned from data by combining the algorithmic priors in SMC methods with the expressiveness of neural networks. For example, [17] considers the observation likelihoods as the scalar outputs of a fully-connected neural network whose input is the concatenation of states and encoded features of observations. In [16], feature maps of both observations and states are fed into a fully connected neural network whose outputs are used as the likelihood of the observations. The likelihood of observations in [19], [20] is constructed as the cosine similarity between the feature maps of observations and states. In [18], the likelihood is given by a multivariate Gaussian model outputting the Gaussian density of observation feature vectors conditioned on state feature vectors. In complex environments such as robot localization tasks with image observations, DPFs have shown promising results [16]–[20]. Nonetheless, to the best of our knowledge, existing DPFs do not admit valid probability densities in likelihood estimation, leading to incorrect quantification of the measurement uncertainty given state information.

In this paper, we present a novel approach to learn the conditional measurement density in DPFs. Our contributions are three-fold: 1) We developed a novel approach to construct measurement models capable of modeling complex likelihood with valid probability densities through conditional normalizing flows; 2) We show how to incorporate and train the conditional normalizing flow-based measurement model into existing DPF pipelines; 3) We show that the proposed method can improve the performance of state-of-the-art DPFs in a visual tracking experiment.

The remainder of the paper is organized as follows: Section II provides background knowledge and Section III describes the proposed conditional normalizing flow-based measurement model. We report the simulation setup and experiment results in Section IV and conclude the paper in Section V.

II. Background

A. Differentiable Particle Filters

In differentiable particle filters, both the dynamic model and the measurement model are often formulated as parameterized models. For example, the transition of states and the likelihood of observations can be modeled by the following parameterized functions, respectively:

$$\mathbf{x}_t^i = g_\theta(\mathbf{x}_{t-1}^i,\mathbf{a}_t,\alpha_t^i) \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^i,\mathbf{a}_t;\theta), \tag{6}$$

$$\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t^i;\theta) = l_\theta(\mathbf{y}_t,\mathbf{x}_t^i), \tag{7}$$

where α_t^i is the process noise. In [16]–[20], $g_\theta(\cdot)$ is parameterized by neural networks taking $\mathbf{x}_{t-1}$ and $\mathbf{a}_t$ as inputs. [18] adopted different dynamic models in different experiments, including multivariate Gaussian models whose mean and covariance matrices are functions of $\mathbf{x}_{t-1}$ and $\mathbf{a}_t$, and a pre-defined physical model designed based on prior knowledge of the system. For the measurement model, $l_\theta(\mathbf{y}_t,\mathbf{x}_t^i)$ is designed as neural networks with scalar outputs in [16], [17], [19], [20], and multivariate Gaussian models in [18]. In addition, proposal distributions can also be specified through a parameterized function $f_\theta(\cdot)$:

$$\mathbf{x}_t^i = f_\theta(\mathbf{x}_{t-1}^i,\mathbf{y}_t,\mathbf{a}_t,\beta_t^i) \sim q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i,\mathbf{y}_t,\mathbf{a}_t;\theta), \tag{8}$$

where β_t^i is the noise term used to generate proposed particles. In [20], $f_\theta(\cdot)$ is built with conditional normalizing flows to move particles to regions close to posterior distributions.

With the particle filtering framework being parameterized, DPFs then optimize their parameters by minimizing certain loss functions via gradient descent. Different optimization objectives have been proposed for training DPFs, which can be categorized as supervised loss [16], [17] and semi-supervised loss [19]. Supervised losses such as the root mean square error (RMSE) and the negative log-likelihood (NLL) require access to the ground truth states, and then compute the difference between the predicted state and the ground truth [16], [17]. In [19], a pseudo-likelihood loss was proposed to enable semi-supervised learning for DPFs, which aims to improve the performance of DPFs when only a small portion of data are labeled.

B. Conditional Density Estimation via Conditional Normalizing Flows

Normalizing flows are a family of invertible mappings transforming probability distributions. A simple base distribution can be transformed into arbitrarily complex distribution using normalizing flows under mild assumptions [21]. Since the transformation is invertible, it is straightforward to evaluate the density of the transformed distribution by applying the change of variable formula.

One application in which normalizing flows is particularly well suited is density estimation [5], [21], [22]. Denote by $p(\mathbf{y})$ a probability distribution defined on $\mathcal{Y} \subseteq \mathbb{R}^{d_\mathbf{y}}$ and $\mathcal{T}(\cdot): \mathbb{R}^{d_\mathbf{y}} \rightarrow \mathbb{R}^{d_\mathbf{y}}$ a normalizing flow. By applying the normalizing flow $\mathcal{T}(\mathbf{y})$, the probability density $p(\mathbf{y})$ can be evaluated as:

$$p(\mathbf{y}) = p_Z\big(\mathcal{T}(\mathbf{y})\big) \left| \det \frac{\partial \mathcal{T}(\mathbf{y})}{\partial \mathbf{y}} \right|, \tag{9}$$

where $p_Z(\cdot)$ is a base distribution [21], and $\left|\det \frac{\partial \mathcal{T}(\mathbf{y})}{\partial \mathbf{y}}\right|$ is the absolute value of the determinant of the Jacobian matrix $\frac{\partial \mathcal{T}(\mathbf{y})}{\partial \mathbf{y}}$ evaluated at $\mathbf{y}$. As a special case of density estimation, conditional density estimation (CDE) can be realized by using variants of normalizing flows, e.g. conditional normalizing flows (CNFs) [4], [5], [20]. Similar to Eq. (9) and with a slight abuse of notations, a conditional normalizing flow $\mathcal{T}(\mathbf{y}, \mathbf{x}): \mathbb{R}^{d_\mathbf{y}} \times \mathbb{R}^{d_\mathbf{x}} \rightarrow \mathbb{R}^{d_\mathbf{y}}$ can be constructed to evaluate the conditional probability density $p(\mathbf{y}|\mathbf{x})$:

$$p(\mathbf{y}|\mathbf{x}) = p_Z\big(\mathcal{T}(\mathbf{y}, \mathbf{x})\big) \left| \det \frac{\partial \mathcal{T}(\mathbf{y}, \mathbf{x})}{\partial \mathbf{y}} \right|, \tag{10}$$

where $\mathbf{x}$ is the variable on which $p(\mathbf{y}|\mathbf{x})$ is conditioned.

Compared with non-parametric and other parametric CDE models, CNFs are efficient and can model arbitrarily complex conditional densities. They have been employed to estimate complex and high-dimensional conditional probability densities [4], [5], [9], [23]. Therefore, we choose to use CNFs to construct the measurement model in DPFs.

III. Differentiable Particle Filter with Conditional Normalizing Flow Measurement model

We provide in this section the details of the proposed method and a generic DPF framework where our measurement model is incorporated. The proposed measurement model is built based on conditional normalizing flows, which can model expressive and valid probability densities and capture the complex likelihood of observations given state information.

A. Likelihood Estimation with Conditional Normalizing Flows

Consider an SMC model where we have an observation $\mathbf{y}_t$ and a set of particles $\{\mathbf{x}_t^i\}_{i=1}^{N_p}$ at the t-th time step. The proposed measurement model estimates the likelihood of $\mathbf{y}_t$ given $\mathbf{x}_t^i$ by:

$$p(\mathbf{y}_t|\mathbf{x}_t^i;\theta) = p_Z(\mathcal{T}_\theta(\mathbf{y}_t, \mathbf{x}_t^i)) \left| \det \frac{\partial \mathcal{T}_\theta(\mathbf{y}_t, \mathbf{x}_t^i)}{\partial \mathbf{y}_t} \right|, \tag{11}$$

where $\mathcal{T}_\theta(\mathbf{y}_t, \mathbf{x}_t^i)$ is a parameterized conditional normalizing flow. The base distribution $p_Z(\cdot)$ can be user-specified and is often chosen as a simple distribution such as isotropic Gaussian.

Algorithm 1 A generic differentiable particle filter with conditional normalizing flow measurement model (DPF-CM) framework. $[N_p]$ is used to denote the set $\{1, 2, \cdots, N_p\}$.

```
Initialize the parameter set $\theta$ and set learning rate $\gamma$;
while $\theta$ has not converged do
    Draw samples $\{\mathbf{x}_0^i\}_{i=1}^{N_p} \sim \pi(\mathbf{x}_0)$;
    Set importance weights $w_0^i = \frac{1}{N_p}$ for $i \in [N_p]$;
    Set $\tilde{\mathbf{x}}_0^i = \mathbf{x}_0^i$ for $i \in [N_p]$;
    for $t = 1,2,...,T$ do
        Sample $\mathbf{x}_t^i$ from proposal distribution for $i \in [N_p]$:
        $\mathbf{x}_t^i = f_\theta(\tilde{\mathbf{x}}_{t-1}^i, \mathbf{y}_t, \mathbf{a}_t, \beta_t^i) \sim q(\mathbf{x}_t^i|\tilde{\mathbf{x}}_{t-1}^i, \mathbf{y}_t, \mathbf{a}_t;\theta)$;
        [optional:] Encode observation: $\mathbf{y}_t := E_\theta(\mathbf{y}_t)$;
        Apply conditional normalizing flows: $\mathbf{z}_t^i = \mathcal{T}_\theta(\mathbf{y}_t, \mathbf{x}_t^i)$;
        Estimate observation likelihood for $i \in [N_p]$:
        $p(\mathbf{y}_t|\mathbf{x}_t^i;\theta) = p_Z(\mathbf{z}_t^i)\left|\det\frac{\partial \mathbf{z}_t^i}{\partial \mathbf{y}_t}\right|$;
        Calculate importance weights for $i \in [N_p]$:
        $w_t^i = w_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{x}_t^i;\theta)p(\mathbf{x}_t^i|\tilde{\mathbf{x}}_{t-1}^i, \mathbf{a}_t;\theta)}{q(\mathbf{x}_t^i|\tilde{\mathbf{x}}_{t-1}^i, \mathbf{y}_t, \mathbf{a}_t;\theta)}$;
        Normalize weights $w_t^i = \frac{w_t^i}{\sum_{m=1}^{N_p} w_t^m}$ for $i \in [N_p]$;
        Resample $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^{N_p}$ to obtain $\{\tilde{\mathbf{x}}_t^i, \frac{1}{N_p}\}_{i=1}^{N_p}$;
    end for
    Calculate the loss function $\mathcal{L}_{\text{overall}}$;
    Update $\theta$ by gradient descent: $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_{\text{overall}}$ .
end while
```

In scenarios where the observations are high-dimensional such as images, evaluating Eq. (11) with the raw observations $\mathbf{y}_t$ can be computationally expensive. As an alternative solution, we can map the observation $\mathbf{y}_t$ to a lower-dimensional space via $\mathbf{e}_t = E_\theta(\mathbf{y}_t) \in \mathbb{R}^{d_\mathbf{e}}$, where E_θ is a parameterized function $E_\theta(\cdot): \mathbb{R}^{d_\mathbf{y}} \rightarrow \mathbb{R}^{d_\mathbf{e}}$. Assume that the conditional probability density $p(\mathbf{e}_t|\mathbf{x}_t^i;\theta)$ of observation features given state is an approximation of the actual measurement likelihood $p(\mathbf{y}_t|\mathbf{x}_t^i;\theta)$, we then estimate $p(\mathbf{y}_t|\mathbf{x}_t^i;\theta)$ with $p(\mathbf{e}_t|\mathbf{x}_t^i;\theta)$:

$$\begin{aligned} p(\mathbf{y}_t|\mathbf{x}_t^i;\theta) &\approx p(\mathbf{e}_t|\mathbf{x}_t^i;\theta) & (12) \\ &= p_Z(\mathcal{T}_\theta(\mathbf{e}_t, \mathbf{x}_t^i)) \left| \det \frac{\partial \mathcal{T}_\theta(\mathbf{e}_t, \mathbf{x}_t^i)}{\partial \mathbf{e}_t} \right|. & (13) \end{aligned}$$

B. Numerical Implementation

We provide in Algorithm 1 a DPF framework where the proposed measurement model is applied, which we name as differentiable particle filter with conditional normalizing flow measurement model (DPF-CM).

IV. Experiment Results

In this section, we compare the performance of the proposed measurement model with other measurement models used in several state-of-the-art works on DPFs [16], [17], [19] in a synthetic visual tracking task introduced in [24], [25]. Two baselines are used in this experiment, including the vanilla differentiable particle filter (DPF) and the conditional normalizing flow differentiable particle filter (CNF-DPF) [20]. We name the two baselines equipped with the proposed measurement model as the DPF-CM and the CNF-DPF-CM, respectively[2].

A. Experiment Setup

The goal of this visual tracking problem is to track the location of a moving red disk. There are two key challenges in this experiment. Firstly, there are 25 distracting disks in observation images, and the distractors move as well while we try to locate the targeted red disk. Secondly, since collisions are not considered in this environment, the red disk may be occluded by distractors or move out of the boundary. The color of these distractors are randomly drawn from the set of $\{$green, blue, cyan, purple, yellow, white$\}$, and the radii of them are randomly sampled with replacement from $\{3, 4, \cdots, 10\}$. The radius of the tracking objective is set to be 7. Note that the locations of the disks, including the target and distractors, are initialized following a uniform distribution over the observation image. The initial velocity of the disks are sampled from a standard Gaussian distribution. Figure 1 shows an example of the observation image at time step $t=0$, which is an RGB image with the dimension 128×128×3.

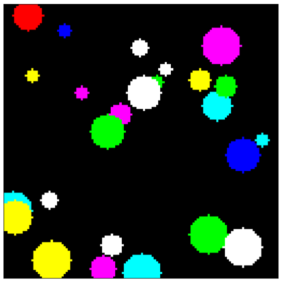

Fig. 1. An observation image in the disk tracking experiment. The goal is to track the red disk among distractors.

With a slight abuse of notations, in this example, we denote by $\mathbf{x}_t$ the location of the red disk at the t-th time step, and $\mathbf{a}_t$ the action. The velocity of the red disk at the t-th time step and the dynamic of the targeted red disk can be described as follows [24]:

$$\hat{\mathbf{a}}_t = \mathbf{a}_t + \epsilon_t,\ \epsilon_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbb{I}), \tag{14}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \hat{\mathbf{a}}_t + \alpha_t,\ \alpha_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0}, \sigma_\alpha^2 \mathbb{I}), \tag{15}$$

where $\mathbb{I}$ is the identity matrix, $\hat{\mathbf{a}}_t$ is the noisy action obtained by adding random action noise ϵ_t, $\sigma_\epsilon = 4$ is the standard deviation of the action noise, and α_t is the dynamic noise whose standard deviation is $\sigma_\alpha = 2$. The distractors follow the same dynamic presented above.

We model the prior distribution $p_Z(\cdot)$ as a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbb{I})$. We choose to use the conditional Real-NVP model [5] to construct conditional normalizing flow $\mathcal{T}_\theta(\cdot)$. An optimal transport map-based resampling approach [18] is adopted for the resampling step in Line 13 of Algorithm 1. For the proposal distribution $q(\mathbf{x}_t^i|\tilde{\mathbf{x}}_{t-1}^i, \mathbf{y}_t, \mathbf{a}_t; \theta)$, we follow the setup in [20] where the proposed particles are generated through conditional normalizing flows.

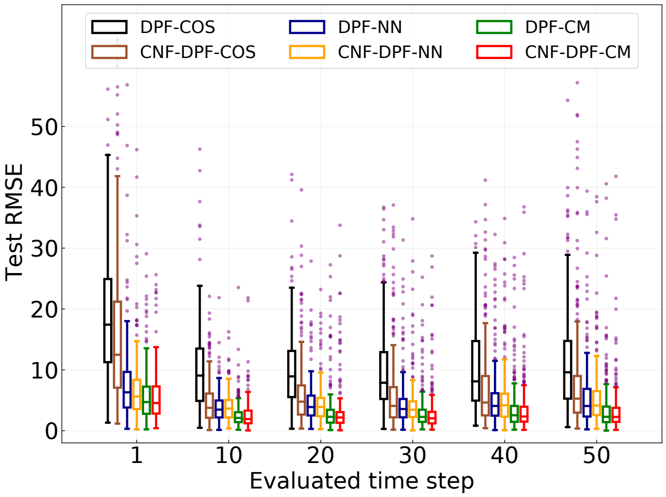

Fig. 2. A boxplot that shows test RMSEs of different methods evaluated at specified time steps. The test set consists of 50 different trajectories, each with 50 time steps. Data points in the boxplot are generated from 5 simulation runs with different random seeds.

We optimized the DPFs end-to-end by minimizing a loss function $\mathcal{L}_{\text{overall}}$ which consists of $\mathcal{L}_{\text{RMSE}}$, the root mean square error (RMSE) between the prediction and the ground truth, and $\mathcal{L}_{\text{AE}}$, the autoencoder reconstruction loss w.r.t observations:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{RMSE}} + \mathcal{L}_{\text{AE}}, \tag{16}$$

$$\mathcal{L}_{\text{RMSE}} = \sqrt{\frac{1}{T}\sum_{t=0}^{T} ||\hat{\mathbf{x}}_t - \mathbf{x}_t^*||_2^2}, \tag{17}$$

$$\mathcal{L}_{\text{AE}} = \frac{1}{T}\sum_{t=0}^{T} ||D_\theta(E_\theta(\mathbf{y}_t)) - \mathbf{y}_t||_2^2. \tag{18}$$

In Eqs. (16), (17), and (18), T is the number of time steps, $||\cdot||_2$ denotes the L^2 norm, x_t^* is the ground truth state at the t-th time step, $\hat{\mathbf{x}}_t = \sum_{i=1}^{N_p} w_t^i \mathbf{x}_t^i$ is an estimation of $\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\mathbf{x}_t]$, and $D_\theta(\cdot)$ and $E_\theta(\cdot)$ are respectively the decoder and the encoder of an autoencoder where the latent variable $\mathbf{e}_t = E_\theta(\mathbf{y}_t)$ is a 32-dimensional vector.

B. Experiment results

In reporting and discussing the experiment results with Fig. 2, Fig. 3 and Table I, we adopt the following naming convention to denote the proposed method and the compared algorithms – "DPF" refers to a baseline DPF [19]; the prefix "CNF-" refers to the approach introduced in [20] where conditional normalizing flows are used to construct dynamic models and proposal distributions; the suffix "-COS" represents the measurement model proposed in [19] which considers the cosine similarity between encoded features

[2] Code to reproduce the experiment results is available at: https://github.com/xiongjiechen/Normalizing-Flows-DPFs.

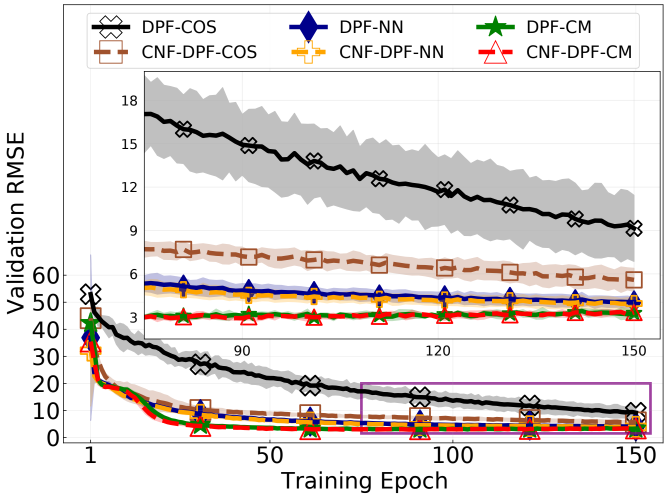

Fig. 3. Validation RMSE of different approaches during the training process. The validation set consisting of 50 different trajectories of length 50. The error bar represents the 95% confidence interval of validation RMSE among 5 simulation runs.

as the likelihood; the suffix "-NN" is used to denote the measurement model introduced in [16], [17], which outputs the likelihood via a neural network with encoded features as its input; "-CM" denotes the proposed measurement model.

Fig. 2 compares RMSEs of state predictions from different methods on the test set. It can be observed that the DPFs with the proposed measurement model, i.e. the DPF-CM and CNF-DPF-CM, exhibit the lowest RMSEs. The overall test RMSEs averaged across 50 time steps are reported in Table I, which shows that both the DPF-CM and CNF-DPF-CM produce the smallest overall test RMSEs. In addition, Fig. 3 plots the validation RMSEs recorded during the training phase. The proposed measurement model leads to faster convergence and the smallest validation RMSEs.

TABLE I
THE COMPARISON OF THE OVERALL AVERAGE RMSE ON THE TEST SET BETWEEN DIFFERENT APPROACHES, THE STANDARD DEVIATION IS CALCULATED THROUGH 5 SIMULATION RUNS.

Method	RMSE	Method	RMSE
DPF-COS	10.61±2.22	CNF-DPF-COS	5.87±0.83
DPF-NN	4.26±0.32	CNF-DPF-NN	4.19±0.31
DPF-CM	**2.99±0.13**	CNF-DPF-CM	**2.86±0.10**

V. Conclusion

We propose in this paper a novel measurement model for differentiable particle filters. The proposed measurement model employs conditional normalizing flows to construct flexible and valid probability densities for likelihood estimation that can be embedded into existing DPF frameworks. A numerical implementation of the proposed method is provided in the paper, and we evaluate the performance of the proposed method in a visual disk tracking task. Numerical results show that the proposed method can significantly improve the tracking performance compared with state-of-the-art DPF variants.

References

[1] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Adv. in Neural Inf. Process. Syst. (NeurIPS)*, Montreal, Canada, Dec. 2015.

[2] S. Malik and M. K. Pitt, "Particle filters for continuous likelihood evaluation and maximisation," *J. Econometrics*, vol. 165, no. 2, pp. 190–209, 2011.

[3] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Sydney, Australia, Aug. 2017.

[4] Y. Lu and B. Huang, "Structured output learning with conditional generative flows," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, New York, New York, Feb. 2020.

[5] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling, "Learning likelihoods with conditional normalizing flows," *arXiv preprint arXiv:1912.00042*, 2019.

[6] C. M. Bishop, "Mixture density networks," *Technical Report*, 1994.

[7] R. Shu, H. H. Bui, and M. Ghavamzadeh, "Bottleneck conditional density estimation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Sydney, Australia, Aug. 2017.

[8] L. Ambrogioni, U. Güçlü, M. A. van Gerven, and E. Maris, "The kernel mixture network: A nonparametric method for conditional density estimation of continuous random variables," *arXiv preprint arXiv:1705.07111*, 2017.

[9] B. L. Trippe and R. E. Turner, "Conditional density estimation with Bayesian normalising flows," *arXiv preprint arXiv:1802.04908*, 2018.

[10] D. M. Bashtannyk and R. J. Hyndman, "Bandwidth selection for kernel conditional density estimation," *J. Comput. Statist. & Data Anal.*, vol. 36, no. 3, pp. 279–298, 2001.

[11] V. Dutordoir, H. Salimbeni, J. Hensman, and M. Deisenroth, "Gaussian process conditional density estimation," in *Proc. Adv. in Neural Inf. Process. Syst. (NeurIPS)*, Montreal, Canada, Dec. 2018.

[12] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *Proc. IEE F-radar and Signal Process. (IEE-F)*, Apr. 1993.

[13] A. Doucet and V. B. Tadić, "Parameter estimation in general state-space models using particle methods," *Ann. Inst. Stat. Math.*, vol. 55, no. 2, pp. 409–422, 2003.

[14] C. Andrieu, A. Doucet, and V. B. Tadic, "On-line parameter estimation in general state-space models," in *Proc. IEEE Conf. Decision and Control (CDC)*, Seville, Spain, Dec. 2005.

[15] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On particle methods for parameter estimation in state-space models," *Statistical Science*, vol. 30, no. 3, pp. 328–351, 2015.

[16] P. Karkus, D. Hsu, and W. S. Lee, "Particle filter networks with application to visual localization," in *Proc. Conf. Robot Learn. (CoRL)*, Zurich, Switzerland, Oct 2018.

[17] R. Jonschkowski, D. Rastogi, and O. Brock, "Differentiable particle filters: end-to-end learning with algorithmic priors," in *Proc. Robot.: Sci. and Syst. (RSS)*, Pittsburgh, Pennsylvania, July 2018.

[18] A. Corenflos, J. Thornton, G. Deligiannidis, and A. Doucet, "Differentiable particle filtering via entropy-regularized optimal transport," in *Proc. Int. Conf. Mach. Learn. (ICML)*, July 2021.

[19] H. Wen, X. Chen, G. Papagiannis, C. Hu, and Y. Li, "End-to-end semi-supervised learning for differentiable particle filters," in *Proc. IEEE Int. Conf. Robot. Automat., (ICRA)*, Xi'an, China, May 2021.

[20] X. Chen, H. Wen, and Y. Li, "Differentiable particle filters through conditional normalizing flow," in *Proc. IEEE Int. Conf. Inf. Fusion (FUSION)*, Sun City, South Africa, Nov. 2021.

[21] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *J. Mach. Learn. Res. (JMLR)*, vol. 22, pp. 1–64, 2021.

[22] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Toulon, France, Apr. 2017.

[23] A. Abdelhamed, M. A. Brubaker, and M. S. Brown, "Noise flow: noise modeling with conditional normalizing flows," in *Proc. IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, Seoul, Korea, Oct. 2019.

[24] A. Kloss, G. Martius, and J. Bohg, "How to train your differentiable filter," *J. Auton. Robots*, vol. 45, no. 4, pp. 561–578, 2021.

[25] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, "Backprop KF: learning discriminative deterministic state estimators," in *Proc. Adv. in Neural Inf. Process. Syst. (NeurIPS)*, Barcelona, Spain, Dec. 2016.