Differentiable Nonparametric Belief Propagation

Anthony Opipari[1], Jana Pavlasek[1], Chao Chen[1],
Shoutian Wang[1], Karthik Desingh[2], and Odest Chadwicke Jenkins[1]

***Abstract*— We present a differentiable approach to learn the probabilistic factors used for inference by a nonparametric belief propagation algorithm. Existing nonparametric belief propagation methods rely on domain-specific features encoded in the probabilistic factors of a graphical model. In this work, we replace each crafted factor with a differentiable neural network enabling the factors to be learned end-to-end, using an efficient optimization routine from labeled data. By combining differentiable neural networks with an efficient belief propagation algorithm, our method learns to maintain a set of marginal posterior samples. We evaluate our differentiable nonparametric belief propagation (DNBP) method on a set of articulated pose tracking tasks and compare performance with convolutional neural networks. Results from this comparison demonstrate the effectiveness of using learned factors for tracking and suggest the practical advantage over hand-crafted approaches. The project webpage is available at: http://progress.eecs.umich.edu/projects/dnbp.**

I. Introduction

Perception of articulated object pose in noisy environments remains a significant challenge for robotic applications. Nonparametric belief propagation (NBP) algorithms [1], [2] have proven effective for inference in these environments [3], [4], [5]. Moreover, these algorithms are able to account for uncertainty in their estimates when environmental noise is high and show promising computational properties in practice [4], [6]. Their adaptability to new applications, however, is limited by the need to define hand-crafted functions that describe the distinct statistical relationships in a particular dataset. In this paper, we present a differentiable nonparametric belief propagation (DNBP) method, a hybrid approach which leverages neural networks to parameterize the NBP algorithm. Specifically, we develop a differentiable version of the efficient pull message passing nonparametric belief propagation (PMPNBP) algorithm presented by Desingh et al. [4]. We are inspired by the differentiable particle filter proposed by Jonschkowski et al. [7]. Similar to this approach, DNBP performs end-to-end learning of each probabilistic factor required for graphical model inference.

The effectiveness of DNBP is demonstrated on two challenging articulated tracking tasks in noisy environments. Results show that our approach can leverage the graph structure to report uncertainty about its estimates while significantly reducing the need for prior domain knowledge required by previous NBP methods. DNBP performs competitively in comparison to traditional learning-based approaches. Collectively these results indicate that DNBP has the potential to be successfully applied to robotic perception tasks, such as pose estimation, where uncertainty estimation is crucial.

II. Related Work

Probabilistic graphical models, such as the Markov Random Field (MRF), describe probability distributions as a collection of nodes and edges. The nodes of a graphical model correspond to random variables while the edges represent probabilistic relationships between the variables. Given a graphical model, belief propagation (BP) is a message passing algorithm for inferring the marginal distributions expressed within the graph. BP computes exact marginal distributions on trees [8], and has demonstrated empirical success on loopy graphs [9], [10], [11], [12]. In continuous, high-dimensional spaces, BP becomes intractable due to its demand for integral computation. Nonparametric belief propagation (NBP) methods approximate the marginal distributions as mixtures of Gaussians and define efficient message passing schemes that perform inference [2], [1], [13], [3]. Particle belief propagation [14] approximates marginal distributions by a set of particles, yielding tractable algorithms for high-dimensional spaces [4], [5].

An important limitation of the existing NBP approaches is their assumption that the probabilistic relationships expressed in the graph model are provided as input. In these approaches, the factors must be modeled or trained using domain knowledge unique to each application. The potential for neural networks to learn the parameters used by alternative inference techniques has been demonstrated [15], [16], [17]. Other works have demonstrated learned observation likelihoods within nonparametric inference frameworks [18], [5]. For robotic perception applications, end-to-end differentiable Bayes filtering algorithms enable automatic learning of probabilistic factors and have been shown to outperform recurrent neural networks [19], [20], [7], [21], [22], [23], [24]. In contrast, the focus of this study on articulated object pose estimation motivates the use of NBP since its ability to factor high-dimensional spaces is associated with improved performance in the face of increased dimensionality [4], [5].

This study sets out to explore the potential for an end-to-end deep learning framework to be used to learn the probabilistic factors for nonparametric belief propagation.

III. Differentiable Nonparametric Belief Propagation

Consider an MRF model defined by the undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ denotes a set of nodes and $\mathcal{E}$ denotes a

[1]Robotics Institute, University of Michigan, Ann Arbor {topipari,pavlasek,joecc,shoutian,ocj}@umich.edu
[2]Department of Computer Science & Engineering, University of Washington kdesingh@cs.washington.edu

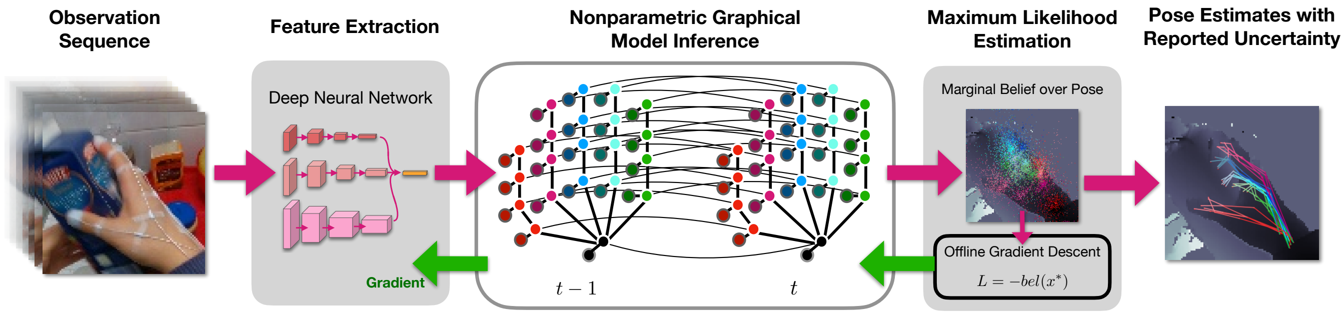

Fig. 1: Architecture diagram of differentiable nonparametric belief propagation. DNBP combines domain knowledge in the form of graphical models with differentiable neural networks for tractable inference in continuous spaces. Input features from a deep neural network and the probabilistic relationships encoded in a graphical model are learned jointly in an end-to-end fashion using backpropagation. Following offline training, DNBP can be applied to unseen data without hand-tuning.

set of edges. An example MRF model is shown in the center column of Fig. 1. Each node in $\mathcal{V}$ represents an observed or unobserved random variable while each edge in $\mathcal{E}$ represents a pairwise relationship between two random variables in $\mathcal{V}$. The MRF formulation specifies the joint probability distribution for this collection of random variables as:

$$p(\mathcal{X}, \mathcal{Y}) = \frac{1}{Z} \prod_{(s,d)\in\mathcal{E}} \psi_{s,d}(X_s, X_d) \prod_{d\in\mathcal{V}} \phi_d(X_d, Y_d) \tag{1}$$

where $\mathcal{X} = \{X_d \,|\, d \in \mathcal{V}\}$ is the set of unobserved variables and $\mathcal{Y} = \{Y_d \,|\, d \in \mathcal{V}\}$ is the set of corresponding observed variables. The scalar Z is a normalizing constant. For each edge, the function $\psi_{s,d}(\cdot)$ is the *pairwise potential*, describing the compatibility of neighboring variables X_s and X_d. For each node, the function $\phi_d(\cdot)$ is the *unary potential*, describing the compatibility of an unobserved variable X_d with a corresponding observed variable Y_d.

Given the factorization in Eq. (1), a nonparametric belief propagation method such as PMPNBP [4] may be used to infer the marginal posterior distributions, or 'beliefs', at each unobserved node. We propose a differentiable nonparametric belief propagation (DNBP) method. DNBP is a message-passing algorithm that maintains a representation of the uncertainty in its estimate by efficiently approximating the marginal posterior distributions encoded in an MRF. DNBP represents the belief and messages at iteration t by sets of N and M weighted particles respectively:

$$bel_d^t(X_d) = \left\{\left(\mu_d^{(i)}, w_d^{(i)}\right)\right\}_{i=1}^{N} \tag{2}$$

$$m_{s\to d}^t = \left\{\left(\mu_{sd}^{(i)}, w_{sd}^{(i)}\right)\right\}_{i=1}^{M} \tag{3}$$

To compute the beliefs at each node of the MRF, DNBP relies on an iterative "pull" message passing strategy similar to the one presented by [4]. In this strategy, each iteration of the algorithm is defined in terms of a message update step and a belief update step[1]. The message update generates a new set of message particles as a reweighted set of samples from the previous iteration's belief. Crucially, our method avoids the need to define hand-crafted functions for each domain by modeling the potentials needed for computing the distributions in Eq. (1) with neural networks that are trained end-to-end as opposed to hand-crafted ones. Following a message update, the belief update combines information that is incoming to each node from the newly generated messages. The final marginal posterior estimates result from the belief update.

The following sections describe the neural networks used to compute the message and belief updates.

Unary Potential Functions: According to the factorization of the MRF joint distribution in Eq. (1), each unobserved variable X_d, for $d \in \mathcal{V}$, is related to a corresponding observed variable Y_d by the unary potential function $\phi_d(X_d, Y_d)$. DNBP models each unary function with a feedforward neural network. The unary for a particle, x_d, given an observed image, y_d, is:

$$\phi_d(X_d = x_d, Y_d = y_d) = l_d\left(x_d \oplus f_d(y_d)\right) \tag{4}$$

where f_d is a convolutional neural network, l_d is a fully connected neural network, and the symbol $\oplus$ denotes concatenation of feature vectors.

Pairwise Potential Functions: For any pair of hidden variables, X_s and X_d, which are connected by an edge in $\mathcal{E}$, a pairwise potential function, $\psi_{s,d}(X_s, X_d)$, represents the probabilistic relationship between the two variables. DNBP models each pairwise potential using a pair of feedforward, fully connected neural networks, $\psi_{s,d}(X_s, X_d) = \{\psi_{sd}^{\rho}(\cdot), \psi_{sd}^{\sim}(\cdot)\}$. The pairwise *density* network, $\psi_{sd}^{\rho}(\cdot)$, evaluates the un-

NETWORK	UNIT LAYERS
f_s	5 x [conv(3x3, 10, stride=2, ReLU), maxpool(2x2, 2)]
l_s	2 x fc(64, ReLU), fc(1, Sigmoid scaled to [0.005, 1])
ψ_{sd}^{ρ}	4 x fc(32, ReLU), fc(1, Sigmoid scaled to [0.005, 1])
$\psi_{sd}^{\sim}$	2 x fc(64, ReLU), fc(2)
$\tau_s^{\sim}$	2 x fc(64, ReLU), fc(2)

TABLE I: Network parameters of learned DNBP potential functions used on both simulated articulated tracking tasks. Note $s, d \in \mathcal{V}$, and $(s, d) \in \mathcal{E}$. Unary potentials: $l_s(f_s(\cdot))$. Pairwise potentials: $\{\psi_{sd}^{\rho}, \psi_{sd}^{\sim}\}$. Particle diffusion: $\tau_s^{\sim}$.

[1] Implementation of message and belief update steps as well as experiment datasets are available at: https://github.com/opipari/diffBP

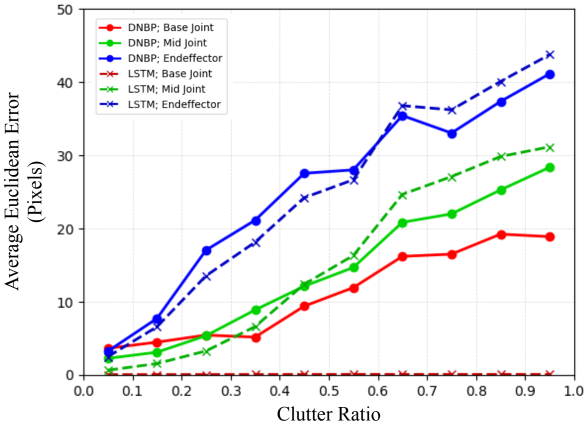

Fig. 2: Average error of DNBP and LSTM predictions as a function of clutter ratio and keypoint type for the double pendulum tracking problem.

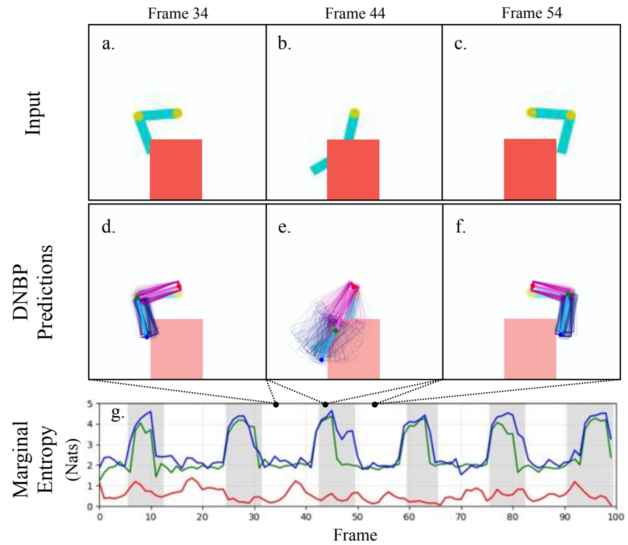

Fig. 3: Tracking of double pendulum by DNBP under partial occlusion (orange block). Uncertainty associated with predictions is shown as samples from the joint distribution in pink and blue (d,e,f). (g) Marginal entropy for each keypoint across test sequence; base keypoint (red), middle keypoint (green), end-effector keypoint (blue). Sequence steps highlighted by gray correspond to images in which $> 25\%$ of the pendulum is occluded.

normalized potential for a pair of particles. The pairwise *sampling* network, $\psi^{\sim}_{sd}(\cdot)$, is used to form samples of node s conditioned on node d and vice versa.

Particle Diffusion: DNBP uses a learned particle diffusion model for each hidden variable, modeled as distinct feedforward neural networks, $\tau^{\sim}_{d}(\cdot)$ for $d \in \mathcal{V}$. This diffusion model replaces the Gaussian diffusion models typically used by particle-based inference methods that encourage exploration of the state space. At the outset of message generation at iteration t, DNBP's belief particles from iteration $t-1$ are resampled then passed through the diffusion model at the beginning of iteration t to form the messages used to update the distributions at iteration t.

Particle Resampling: The final operation of the belief update algorithm in NBP is a weighted resampling of belief particles. This resampling operation is non-differentiable [21], [7]. It follows that the iterative belief update algorithm is non-differentiable due to the resampling step. DNBP addresses the non-differentiability of the belief update algorithm by relocating the resampling and diffusion operations to the beginning of the message update algorithm. The resulting algorithm is differentiable through one belief update and message passing update.

A. *Supervised Training*

DNBP's training approach is inspired by the work of [7] with modifications to enable learning the potential functions distinct to DNBP. During training, DNBP uses a set of observation sequences, and a corresponding set of ground truth sequences. Using the observation sequences, DNBP estimates belief of each unobserved variable at each sequence step. Then, by maximizing estimated belief at the ground truth label of each unobserved variable, DNBP learns its network parameters by maximum likelihood estimation.

Objective Function: Given a set of weighted particles representing the belief of X_d produced by the inference procedure at iteration t, the density of the belief can be expressed as a mixture of Gaussians, with a component centered at each particle. The density of a sample x_d can be computed as follows:

$$\overline{bel}^{t}_{d}(x_d) = \sum_{i=1}^{N} w^{(i)}_{d} \cdot \mathcal{N}(x_d; \mu^{(i)}_{d}, \Sigma) \tag{5}$$

DNBP defines a loss function on each hidden node $d \in \mathcal{G}$ as:

$$L^{t}_{d} = -\log(\overline{bel}^{t}_{d}(x^{t,*}_{d})) \tag{6}$$

where $x^{t,*}_{d}$ denotes the ground truth label for node d at sequence step t. The loss for each hidden node is computed and optimized separately. At each sequence step during training, DNBP iterates through the nodes of the graph, updating each node's incoming messages and belief followed by a single optimization step of Eq. (6) using stochastic gradient descent.

IV. Experiments

The capability of DNBP is demonstrated on two challenging articulated tracking tasks. The performance of DNBP is compared with a baseline LSTM neural network [25] and a convolutional neural network [26] on the two tasks respectively. In the first task, both DNBP and LSTM attempt to track the 2-dimensional position of each joint of a simulated double pendulum, as shown in Fig. 3, that swings under the effect of gravity. To emulate environments where occlusion is common, simulated clutter in the form of static and dynamic geometric shapes are rendered into the image sequences. The second task involves tracking the 3-dimensional position of each joint of a human hand. DNBP models the kinematic

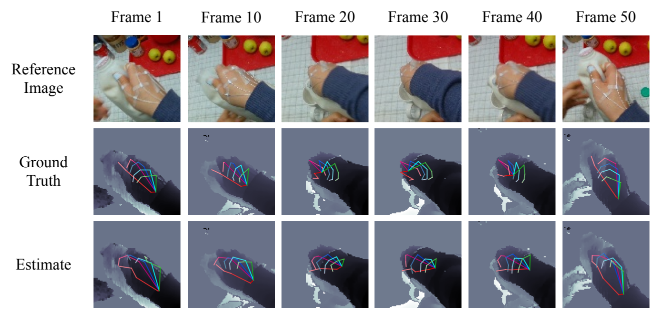

Fig. 4: Output from DNBP throughout a chosen sequence of hand tracking. DNBP maintains plausible estimates of the hand pose in cases of occlusion (Frames 20, 30, 40) and recovers with improved observability (Frame 50).

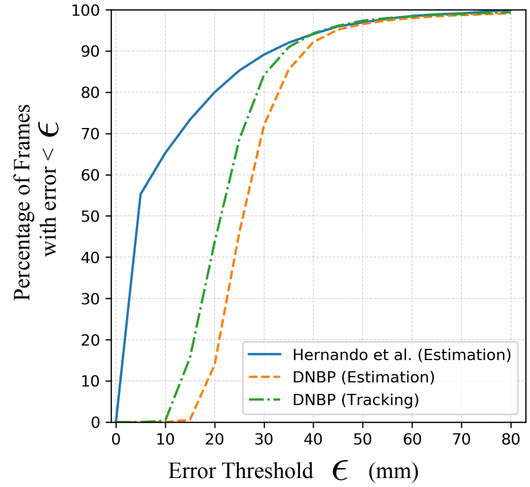

Fig. 5: Quantitative comparison between DNBP and neural network baseline on hand pose tracking task of the FPHAB dataset. For each model the percent of frames with predicted pose less than a set threshold is calculated as the threshold is varied from 0mm to 80mm.

structure in each task using an MRF model, in which each unobserved variable corresponds to the position of a particular joint keypoint and each observed variable corresponds to an observed image.

A. Double Pendulum Tracking

As shown in Fig. 2, the keypoint tracking error of DNBP is directly compared to that of the LSTM baseline on a held-out test set. Results from this comparison show that DNBP's average keypoint tracking error is comparable to the LSTM's corresponding error for both the mid joint and end effector keypoints, independent of clutter ratio. For the base joint keypoint, which is stationary at the center position of every image, the LSTM was able to memorize the correct position while DNBP registers a consistently larger error. The lack of memorization of the base joint position is likely due to the use of particle diffusion within DNBP's message passing scheme, which encourages exploration at the expense of memorization. Next tested was the hypothesis that the DNBP model would generate increased uncertainty under conditions in which occlusions are present. Results from this test, shown in Fig. 3, demonstrate that under optimal conditions, with minimal occlusion, the output of DNBP indicates a low level of uncertainty. In contrast, under conditions of occlusion, the model's output indicates relatively high levels of uncertainty precisely at frames in which a superimposed object occludes a portion of the double pendulum. These results demonstrate that DNBP has the ability to represent uncertainty along with its estimates thereby allowing DNBP to automatically identify cases in which it expects to be unreliable, which is a crucial necessity for robots acting in unstructured human environments.

B. Human Hand Tracking Results

To evaluate DNBP's capability for application to real-world tasks, the algorithm's state estimation and tracking performance was evaluated on the FPHAB dataset [26]. This is a challenging dataset with extreme occlusions where complete observations of all the finger joints are rare. Example output estimates from DNBP is included in Fig. 4. As a quantitative evaluation of estimation accuracy, Euclidean error between the estimated and ground truth pose is measured for every frame in the test set. For this first evaluation, DNBP is applied as a frame-by-frame estimator without maintaining its belief over time. The quantitative results from this experiment, are included in Fig. 5 with direct comparison to a pure neural network baseline. The results from this experiment indicate that for error thresholds below 50mm, DNBP will consistently have an accuracy of 95% and above.

Following the comparison against a state of the art baseline, it was hypothesized that DNBP's performance would improve when applied as a tracking method which maintains belief over time. To perform this test, DNBP was applied sequentially to each test sequence and evaluated under the same error metric. The result from this test, as shown in Fig. 5, demonstrates that DNBP does improve in terms of frame error when allowed to track its uncertainty over time. Qualitative examples (on frames from randomly chosen sequences) and tracking videos showing DNBP's estimates and belief are included on the project webpage: http://progress.eecs.umich.edu/projects/dnbp.

V. Discussion

In this work, we proposed a novel formulation of belief propagation which is differentiable and uses a nonparametric representation of belief. It was hypothesized that combining maximum likelihood estimation with the nonparametric inference approach would enable end-to-end learning of the probabilistic factors needed for inference. The hypothesis was tested on both qualitative and quantitative experiments. Results from this study demonstrate DNBP successfully learns to encode probabilistic factors which enable competitive performance on robotic pose estimation tasks. These results motivate further experiments to understand how DNBP may be used in conjunction with uncertainty-aware planning systems for robotic manipulation tasks.

REFERENCES

[1] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2003, pp. 605–612.

[2] M. Isard, "PAMPAS: Real-valued graphical models for computer vision," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2003.

[3] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2004, pp. 421–428.

[4] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins, "Efficient non-parametric belief propagation for pose estimation and manipulation of articulated objects," *Science Robotics*, vol. 4, no. 30, 2019.

[5] J. Pavlasek, S. Lewis, K. Desingh, and O. C. Jenkins, "Parts-based articulated object localization in clutter using belief propagation," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[6] J. Ortiz, T. Evans, and A. J. Davison, "A visual introduction to gaussian belief propagation," *arXiv preprint arXiv:2107.02308*, 2021.

[7] R. Jonschkowski, D. Rastogi, and O. Brock, "Differentiable particle filters: End-to-end learning with algorithmic priors," in *Robotics: Science and Systems (RSS)*, 2018.

[8] J. Pearl, "Chapter 4 - belief updating by network propagation," in *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, Ed. San Francisco (CA): Morgan Kaufmann, 1988, pp. 143 – 237.

[9] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Conference on Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann, 1999, pp. 467–475.

[10] J. Sun, N. Zheng, and H. Shum, "Stereo matching using belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, 2003.

[11] K. Lee, D. Anguelov, B. Sumengen, and S. B. Göktürk, "Markov random field models for hair and face segmentation," in *International Conference on Automatic Face and Gesture Recognition (FG 2008)*. IEEE Computer Society, 2008, pp. 1–6.

[12] X. Lan, S. Roth, D. P. Huttenlocher, and M. J. Black, "Efficient belief propagation with learned higher-order markov random fields," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 3952. Springer, 2006, pp. 269–282.

[13] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, "Visual hand tracking using nonparametric belief propagation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, 2004, pp. 189–189.

[14] A. Ihler and D. McAllester, "Particle belief propagation," in *Artificial Intelligence and Statistics*, 2009, pp. 256–263.

[15] T. M. T. Do and T. Artières, "Neural conditional random fields," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, ser. JMLR Proceedings, Y. W. Teh and D. M. Titterington, Eds., vol. 9. JMLR.org, 2010, pp. 177–184.

[16] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Advances in Neural Information Processing Systems*, 2014, pp. 1799–1807.

[17] H. Xiong and N. Ruozzi, "General purpose MRF learning with neural network potentials," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, C. Bessiere, Ed. ijcai.org, 2020, pp. 2769–2776.

[18] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A Rao-Blackwellized particle filter for 6D object pose tracking," *Proceedings of Robotics: Science and Systems*, 2019.

[19] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, "Backprop KF: learning discriminative deterministic state estimators," in *Advances in Neural Information Processing Systems*, 2016, pp. 4376–4384.

[20] R. Jonschkowski and O. Brock, "End-to-end learnable histogram filters," in *Workshop on Deep Learning for Action and Interaction at NeurIPS*, December 2016.

[21] P. Karkus, D. Hsu, and W. S. Lee, "Particle filter networks with application to visual localization," in *Conference on Robot Learning (CoRL)*, vol. 87. PMLR, 2018, pp. 169–178.

[22] M. A. Lee, B. Yi, R. Martín-Martín, S. Savarese, and J. Bohg, "Multimodal sensor fusion with differentiable filters," *CoRR*, vol. abs/2010.13021, 2020.

[23] A. Kloss, G. Martius, and J. Bohg, "How to train your differentiable filter," *Autonomous Robots*, pp. 1–18, 2021.

[24] B. Yi, M. Lee, A. Kloss, R. Martín-Martín, and J. Bohg, "Differentiable factor graph optimization for learning smoothers," *arXiv preprint arXiv:2105.08257*, 2021.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2018.