Hybrid Inference with Invertible Neural Networks in Factor Graphs

Bart van Erp Bert de Vries
Department of Electrical Engineering, *Eindhoven University of Technology*, Eindhoven, The Netherlands

***Abstract*—This paper bridges the gap in the literature between neural networks and probabilistic graphical models. Invertible neural networks are incorporated in factor graphs and inference in this model is described by linearization of the network. Consequently, hybrid probabilistic inference in the model is realized through message passing with local constraints on the Bethe free energy. We provide the local Bethe free energy for the invertible neural network node, which allows for evaluation of the performance of the entire probabilistic model. Experimental results show effective hybrid inference in a neural network-based probabilistic model for a binary classification task, paving the way towards a novel class of machine learning models.**

***Index Terms*—Factor Graphs, Hybrid Inference, Invertible Neural Networks, Message Passing, Bethe Free Energy**

I. Introduction

Alongside the introduction of powerful neural networks in recent years, invertible neural networks (INNs) have gained significant attention in the literature. Their main usage in normalizing flows has fueled the interest of many [1]–[5]. INNs are enforced to be invertible, either by design [6] or by constraining its Lipschitz constant [7], and are often composed of multiple simpler mappings for improved expressivity. Despite their success in many signal processing tasks, INNs, and more generally black-box neural networks, have often been criticized for their obscure and difficult to evaluate operations.

In contrast to INNs, probabilistic models are known to be better explainable, because they retain uncertainty in their results and allow for incorporating domain-specific knowledge in their model structure. Furthermore, probabilistic graphical models (PGMs) are also well-known for their modularity in model-based machine learning tasks, especially when inference is performed through message passing-based algorithms. Forney-style Factor Graphs (FFGs), a particular kind of PGM, visualize the individual factors of a probabilistic model by nodes that are interconnected by edges that represent the variables in the model [8], [9]. Message passing-based inference in an FFG can be interpreted as a modular and distributed Bethe free energy (BFE) minimization procedure [10].

By combining INNs with PGMs, conventional model-based approaches can be naturally augmented with data-driven elements. Fusing domain-specific knowledge with these data-driven elements allows for more transparent model architectures, which are still capable of solving highly non-linear signal processing problems. This approach aids the development of explainable artificial intelligence.

Correspondence: b.v.erp@tue.nl

This paper incorporates INNs in FFGs using message passing-based probabilistic inference, allowing for joint model- and data-based approaches to signal processing. Under customizable local constraints [10] the neural network-based model easily submits to hybrid inference techniques. After reviewing factor graphs and hybrid message passing in Section II, we make the following contributions:

- We specify INNs as nodes in a factor graph in Section III-A.
- We describe how message passing-based probabilistic inference can be performed in factor graphs containing INNs in Section III-B through linearization of the INN.
- We evaluate the performance of the model through the variational free energy and we describe how it can be used to perform parameter estimation in the networks in Section III-C.
- We demonstrate our methodology by performing hybrid inference in a probabilistic model containing INNs for a binary classification task in Section IV.

II. Factor Graphs and Message Passing

In this section we introduce factor graphs and probabilistic inference by means of message passing. We choose this methodology because of its modularity, efficiency, automatability and scalability [9], [11].

A. Factor graphs

Factor graphs are a class of PGMs. Here we use FFGs as introduced in [12] with notational conventions adopted from [8]. FFGs are undirected graphs with nodes representing the factors of the probabilistic model. These nodes are interconnected by edges with a maximum degree of 2, denoting the variables in the model. For a more thorough review of factor graphs, we refer the interested reader to [8], [9].

B. Sum-product message passing

Probabilistic inference concerns calculating marginal distributions in the model. The calculation of a marginal distribution of some variable requires integration over all nuisance variables in the model. Because of the factorization in the model, this global integration can be performed by smaller localized computations, which summarize parts of the model. The results of these computations are called messages and are propagated over the edges of the graph. The sum-product message $\vec{\mu}(x_j)$ [13] flowing out of some node $f(x_1, x_2, \ldots, x_K)$ with incoming messages $\vec{\mu}(x_{\setminus j})$ is given by

$$\vec{\mu}(x_j) = \int f(x_1, x_2, \ldots, x_K) \prod_{k \neq j} \vec{\mu}(x_k)\ \mathrm{d}\boldsymbol{x}_{\setminus j}. \tag{1}$$

To distinguish between forward and backward messages propagating in or against the direction of some edge x_j, $\vec{\mu}(x_j)$ and $\overleftarrow{\mu}(x_j)$ respectively, the FFG has arbitrarily directed edges. The marginal distribution for variable x_j can be computed from the colliding messages as $p(x_j) \propto \vec{\mu}(x_j) \cdot \overleftarrow{\mu}(x_j)$.

C. Hybrid message passing

Quite often the integral in (1) is intractable and we need to resort to approximate inference methods. Consider the (unnormalized) probabilistic model $f(\boldsymbol{x})$ with factors $\{f_a \mid a \in \mathcal{F}\}$ and variables $\{x_i \mid i \in \mathcal{X}\}$, which factorizes as $f(\boldsymbol{x}) = \prod_{a \in \mathcal{F}} f_a(\boldsymbol{x}_a)$, where $\boldsymbol{x}_a$ denotes the argument variables of f_a. The true posterior distribution $p(\boldsymbol{x})$ is related through $p(\boldsymbol{x}) = f(\boldsymbol{x})/Z$, where $Z = \int f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ is the normalization constant. This constant is often unobtainable and can instead be bounded by the variational free energy (VFE) as

$$\mathrm{F}[q] = \mathbb{E}_{q(\boldsymbol{x})}\left[\ln \frac{q(\boldsymbol{x})}{f(\boldsymbol{x})}\right] = \mathrm{KL}\left[q(\boldsymbol{x}) \parallel p(\boldsymbol{x})\right] - \ln Z, \tag{2}$$

where $q(\boldsymbol{x})$ is an approximation to the true posterior $p(\boldsymbol{x})$.

A versatile approximation for $q(\boldsymbol{x})$ is the Bethe approximation [14], defined in an FFG by the factorization

$$q(\boldsymbol{x}) = \prod_{a \in \mathcal{F}} q_a(\boldsymbol{x}_a) \prod_{i \in \mathcal{X}} q_i(x_i)^{-1}, \tag{3}$$

which results into the Bethe free energy (BFE), an approximation to the VFE, that is exact for trees, as

$$\mathrm{F_B}[q] = -\sum_{a \in \mathcal{F}} \mathbb{E}_{q_a(\boldsymbol{x}_a)}\left[\ln f_a(\boldsymbol{x}_a)\right] - \sum_{a \in \mathcal{F}} \mathrm{H}[q_a] + \sum_{i \in \mathcal{X}} \mathrm{H}[q_i],$$

which decomposes the VFE into the sum of node-local free energy contributions and edge-specific entropies H. Importantly, by adding local constraints to the BFE functional, a variety of high performance inference algorithms can be recovered [10], including the sum-product (belief propagation) algorithm [13], [15], variational message passing [16], [17] and expectation propagation [18]. By combining different local constraints hybrid inference schemes can be obtained. We highly recommend the interested reader to [10] for an overview of these local constraints and their implications for the resulting inference algorithm.

III. Methods

A. Model specification

Consider a deterministic factor

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \delta(\boldsymbol{y} - g(\boldsymbol{x})) = \delta(g^{-1}(\boldsymbol{y}) - \boldsymbol{x}), \tag{4}$$

in an arbitrary probabilistic model, where $\delta(\cdot)$ represents the Dirac delta function. The invertible and differentiable mapping $g : \mathbb{R}^D \to \mathbb{R}^D$, with inverse g^{-1}, maps a (latent) input variable $\boldsymbol{x}$ to a (latent) output variable $\boldsymbol{y}$. The mapping g is allowed to be composed of several mappings (or layers) g_k as

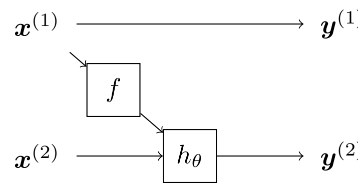


Fig. 1. Schematic overview of the coupling layer in (6) with coupling function h_θ and coupling flow f.

$$\boldsymbol{y} = g(\boldsymbol{x}) = g_K \circ \ldots \circ g_2 \circ g_1(\boldsymbol{x}), \tag{5}$$

where $k = 1, 2, \ldots, K$ denotes the index of the mapping. Latent intermediate outputs $\boldsymbol{x}_k$ of the composite mapping can be calculated recursively as $\boldsymbol{x}_k = g_k(\boldsymbol{x}_{k-1})$, further specifying $\boldsymbol{y} = \boldsymbol{x}_K$ and $\boldsymbol{x} = \boldsymbol{x}_0$. In contrast to [19, Ch.5], [20] we explicitly constrain g to be invertible and differentiable, and we encourage g to be composed of multiple mappings g_k.

Although g is constrained to be invertible, we are not guaranteed to have direct access to its inverse [4], whereas we require this for computationally efficient inference. In this case the coupling layers introduced in [6] can enforce this invertibility whilst retaining a highly expressive transformation. These coupling layers are defined for the forward mapping g (left) and backward mapping g^{-1} (right) as [4], [6]:

$$\begin{aligned} \boldsymbol{y}^{(1)} &= \boldsymbol{x}^{(1)} & \qquad \boldsymbol{x}^{(1)} &= \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} &= h_\theta(\boldsymbol{x}^{(2)}) & \qquad \boldsymbol{x}^{(2)} &= h_\theta^{-1}(\boldsymbol{y}^{(2)}) \end{aligned} \tag{6}$$

Here the input and output are partitioned into two disjoint subspaces $(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ and $(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$, respectively. Note that the superscript refers to the partition index, contrary to the subscript denoting the intermediate outputs. The bijection h_θ, also known as a *coupling function*, with known inverse h_θ^{-1}, is parameterized by $\theta = f(\boldsymbol{x}^{(1)}) = f(\boldsymbol{y}^{(1)})$. An example of such a coupling function is the additive coupling function $h_\theta(\boldsymbol{x}^{(2)}) = \boldsymbol{x}^{(2)} + f(\boldsymbol{x}^{(1)})$ with inverse $h_\theta^{-1}(\boldsymbol{y}^{(2)}) = \boldsymbol{y}^{(2)} - f(\boldsymbol{y}^{(1)})$. Here the *coupling flow* f is an arbitrarily complex function that is not required to be invertible and can therefore even be represented by a neural network. Fig. 1 shows a schematic overview of the coupling layer in (6).

For the modular usage of this factor node in a large probabilistic model, we specify interfacing with the rest of the model by incoming messages $\vec{\mu}(\boldsymbol{x})$ and $\overleftarrow{\mu}(\boldsymbol{y})$ and (to be determined) outgoing messages $\overleftarrow{\mu}(\boldsymbol{x})$ and $\vec{\mu}(\boldsymbol{y})$, see Table I.

B. Probabilistic inference

In contrast to conventional assumptions about INNs, we do not assume inputs or outputs to have fixed values. Inputs and outputs of the INN can originate from adjacent model sections, which can be arbitrarily complex and summarized by messages $\vec{\mu}(\boldsymbol{x})$ and $\overleftarrow{\mu}(\boldsymbol{y})$. As a result these messages can represent probability distributions that incorporate uncertainty about neighboring model sections. To support message passing-based inference with INNs, the incoming messages $\vec{\mu}(\boldsymbol{x})$ and

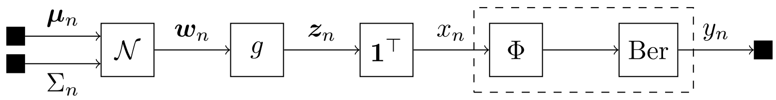
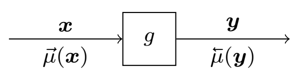


Fig. 2. The factor graph representation of the binary classifier in (9). The composite probit node of (10a) is enclosed by the dashed box.

TABLE I

TABLE CONTAINING THE FORNEY-STYLE FACTOR GRAPH REPRESENTATION OF THE INVERTIBLE NEURAL NETWORK FACTOR NODE, AN OVERVIEW OF THE CHOSEN INCOMING MESSAGES AND THE DERIVED MESSAGES FOR THE INVERTIBLE NEURAL NETWORK FACTOR NODE BY APPROXIMATING THE MAPPING g BY A FIRST-ORDER TAYLOR EXPANSION.

Invertible Neural Network Factor Node	
	
Incoming messages	**Functional form**
$\vec{\mu}(\boldsymbol{x})$	$\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{m}_x,\ \Sigma_x\right)$
$\overleftarrow{\mu}(\boldsymbol{y})$	$\mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{m}_y,\ \Sigma_y\right)$
Outgoing messages	**Functional form**
$\overleftarrow{\mu}(\boldsymbol{x})$	$\mathcal{N}\left(\boldsymbol{x} \mid g^{-1}(\boldsymbol{m}_y),\ J_g^{-1}(\boldsymbol{m}_y)\Sigma_y J_g^{-1}(\boldsymbol{m}_y)^\top\right)$
$\vec{\mu}(\boldsymbol{y})$	$\mathcal{N}\left(\boldsymbol{y} \mid g(\boldsymbol{m}_x),\ J_g(\boldsymbol{m}_x)\Sigma_x J_g(\boldsymbol{m}_x)^\top\right)$

$\overleftarrow{\mu}(\boldsymbol{y})$ need to be propagated through the INN node (4). The exact outgoing sum-product messages can be determined using (1). However, the non-linearity of g and its inverse make exact computations of the outgoing messages intractable. Next, we discuss a solution strategy for resolving these intractabilities.

A feasibly solution approach concerns the linearization of $g(\boldsymbol{x})$ and $g^{-1}(\boldsymbol{y})$ around the means of the incoming messages using a first-order vector Taylor expansion [19, Ch.5] as

$$g(\boldsymbol{x}) \approx g(\boldsymbol{m}_x) + J_g(\boldsymbol{m}_x)(\boldsymbol{x} - \boldsymbol{m}_x) \tag{7a}$$

$$g^{-1}(\boldsymbol{y}) \approx g^{-1}(\boldsymbol{m}_y) + J_{g^{-1}}(\boldsymbol{m}_y)(\boldsymbol{y} - \boldsymbol{m}_y), \tag{7b}$$

with means $\boldsymbol{m}_x = \mathbb{E}_{\vec{\mu}(\boldsymbol{x})}[\boldsymbol{x}]$ and $\boldsymbol{m}_y = \mathbb{E}_{\overleftarrow{\mu}(\boldsymbol{y})}[\boldsymbol{y}]$. J_g and $J_{g^{-1}}$ represent the Jacobian matrices of g and g^{-1}, respectively, which are related through $J_g^{-1} = J_{g^{-1}}$. The composition of g in (5) allows for the decomposition of the Jacobian J_g as

$$J_g(\boldsymbol{x}) = J_{g_K}(\boldsymbol{x}_{K-1}) \cdots J_{g_2}(\boldsymbol{x}_1) J_{g_1}(\boldsymbol{x}_0), \tag{8}$$

where the Jacobian matrices of the individual mappings are evaluated at their corresponding latent intermediate inputs $\boldsymbol{x}_k$.

When the incoming messages can be represented by Normal distributions, this linearization retains their Normal form. An overview of the model with the incoming and outgoing messages is shown in Table I. As an alternative to this linearization, we could also use the Unscented Transform [21], [22], numerical quadrature procedures [20] or general sampling methods (e.g. [23]).

C. Node-local free energy

For determining the performance of models containing INNs we need to determine the node-local free energy of the INN node. [10, Sec. 5.2] describes three approaches of determining the node-local free energy of a deterministic node. We follow their third approach and approximate the node-local free energy by the negative entropy of the input marginal distribution. This input marginal can be approximated as the product of the incoming and outgoing messages $\vec{\mu}(\boldsymbol{x})\overleftarrow{\mu}(\boldsymbol{x})$. With this node-local free energy it becomes possible to evaluate the variational free energy of the entire model. Parameter estimation then concerns the minimization of the variational free energy with respect to the parameters of the INN. This can be achieved by automatic differentiation.

IV. Experimental validation

In this section we perform hybrid message passing in a probabilistic model containing an INN. First, we give an overview of the experimental setup, followed by a formal specification of the probabilistic model and finally we will present the obtained results.[1]

A. Experimental set-up

All experiments are performed in the scientific programming language `Julia` [24]. The `ReactiveMP` package [25] allows us to perform efficient and automated probabilistic inference through tabulated message passing update rules. Besides a wide variety of (hybrid) inference techniques, `ReactiveMP` is compatible with automatic differentiation libraries. In this paper we use `ForwardDiff` [26] for this purpose with `Optim` [27] as optimization package.

B. Model specification

We specify a probabilistic model for binary classification,

$$p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}) = \prod_{n=1}^{N} p(y_n \mid x_n) p(x_n \mid \boldsymbol{z}_n) p(\boldsymbol{z}_n \mid \boldsymbol{w}_n) p(\boldsymbol{w}_n) \tag{9}$$

where the individual factors are given by

$$p(y_n \mid x_n) = \mathrm{Ber}(y_n \mid \Phi(x_n)) \tag{10a}$$

$$p(x_n \mid \boldsymbol{z}_n) = \delta(x_n - \mathbf{1}^\top \boldsymbol{z}_n) \tag{10b}$$

$$p(\boldsymbol{z}_n \mid \boldsymbol{w}_n) = \delta(\boldsymbol{z}_n - g(\boldsymbol{w}_n)) \tag{10c}$$

$$p(\boldsymbol{w}_n) = \mathcal{N}(\boldsymbol{w}_n \mid \boldsymbol{\mu}_n, \Sigma_n). \tag{10d}$$

The model can be interpreted as follows. A (latent) multivariate input $\boldsymbol{w}_n$ with mean vector $\boldsymbol{\mu}_n$ and covariance matrix Σ_n is fed into the INN node $p(\boldsymbol{z}_n \mid \boldsymbol{w}_n)$ with mapping g. Importantly, the invertible and differentiable mapping g is parameterized and is identical across all data points. The output of the INN $\boldsymbol{z}_n$ is then summed to $x_n \in \mathbb{R}$, which

[1] All experiments are available at https://github.com/biaslab/EUSIPCO2022-HybridInferenceInvertibleNeuralNetworks.

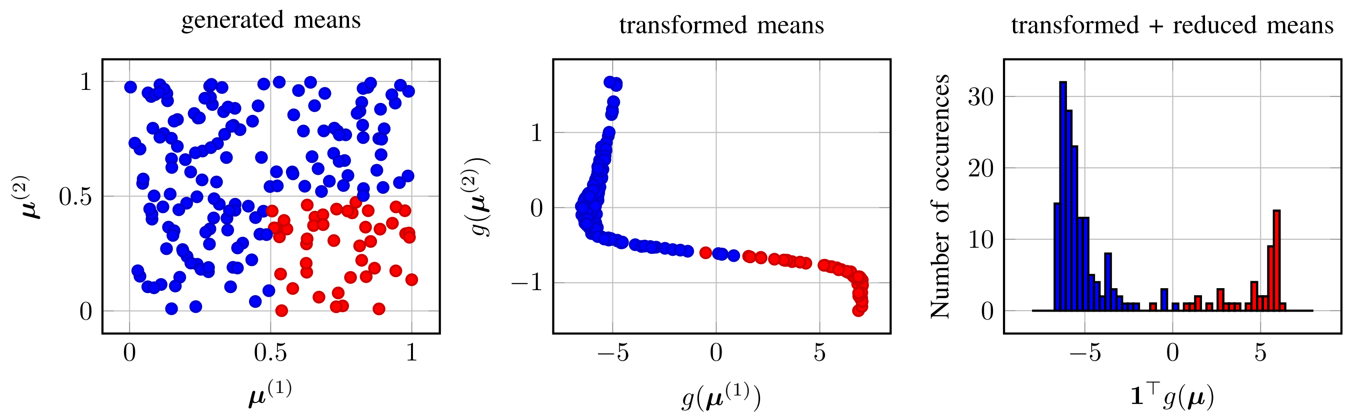


Fig. 3. An overview of the means $\boldsymbol{\mu}_n$ being propagated through the probabilistic model. (left) The generated means $\boldsymbol{\mu}_n$. (middle) Means $\boldsymbol{\mu}_n$ after transformation by the invertible neural network. (right) Histogram of the transformed and reduced means $\boldsymbol{\mu}_n$ at the input of the probit node.

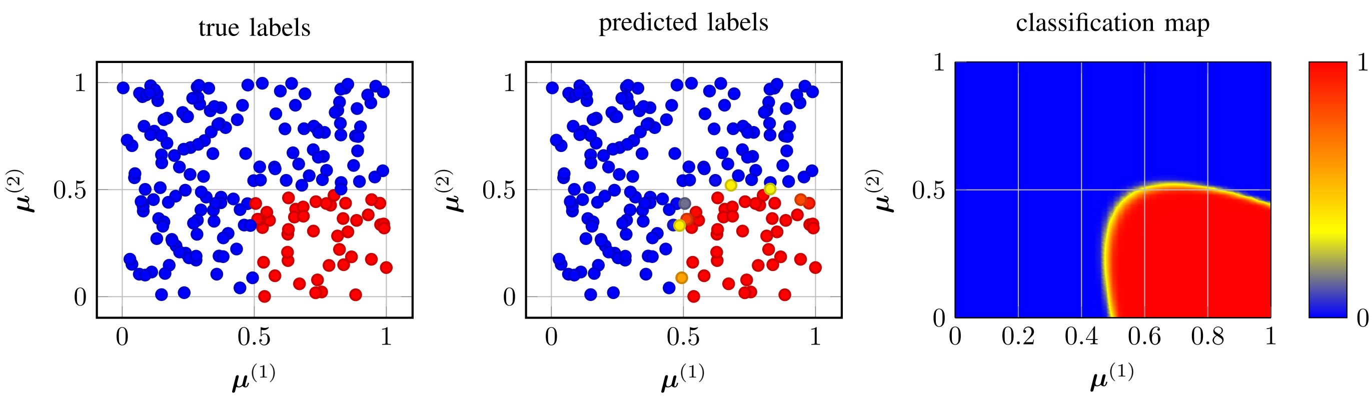


Fig. 4. An overview of the posterior classification of means $\boldsymbol{\mu}_n$. (left) True classification labels. (middle) Predicted posterior classification labels. (right) Predicted posterior classification map.

is mapped to an observed class label $y_n \in \{0, 1\}$ through the probit factor node $p(y_n \mid x_n)$, as specified in [28]. Here $\Phi(\cdot)$ denotes the cumulative distribution function of the standard Normal distribution. Fig. 2 shows the factor graph corresponding to the model in (9). For visualization purposes we restrict $\boldsymbol{w}_n$ (and related variables) to be 2-dimensional, although the specified model can be easily altered to deal with an arbitrary dimensionality.

The mapping g in the INN is modeled by 4 coupling layers. The coupling function is chosen as $h_\theta(x) = x + \theta$ with coupling flow $\theta = f(x) = x + \lambda_1 \tanh(\lambda_2 x + \lambda_3)$, with parameters $\lambda_i \in \mathbb{R}$, inspired by the planar flow of [3]. After each layer the partitions of $\boldsymbol{w}_n$ are permuted, because the coupling layers leave part of the input unchanged, whereas the layers need to modify both partitions [6].

The specified model is subject to hybrid inference because of the probit factor node. Depending on the local constraints [10] around the probit node, the node either sends out a sum-product message towards x_n or performs expectation propagation [18]. Although the functional form of the sum-product message is well-defined, it cannot be recognized as a common distribution. Propagating this message therefore leads to intractable inference. Here we will resort to expectation propagation by applying a moment matching constraint to the posterior distribution of x_n [10].

C. Results

We perform binary classification with the model in (9) for $N = 200$ data points. The covariance matrix Σ_n is set to $10^{-3}\mathrm{I}_2$, where I_d represents the identify matrix of shape $(d \times d)$. The parameters of the INN node are initialized by randomly sampling from a standard Normal distribution. Inference through the INN node is executed using message passing by linearizing the INN as described in Section III-B. The variational free energy is based on the resulting marginals and is minimized with respect to the INN parameters through automatic differentiation during training. Fig. 3 shows how the means $\boldsymbol{\mu}_n$ are transformed through the NF node and after the dot product. From this we can observe that the INN learns to separate the distinct labels. Fig. 4 gives an overview of the true labels of the data points, the predicted posterior labels of these points and posterior classification mapping for $\boldsymbol{\mu}_n \in [0, 1]^2$.

V. Discussion

Table I shows that the covariance matrix of the outgoing message requires the covariance matrix of the incoming message when linearizing the INN. Often it is instead preferred to propagate precision matrices as the calculation of the marginal distribution then does not require any inversions. The outgoing precision matrices can be obtained as a function of the incoming precision matrices by inverting the covariances matrices in Table I, which requires inverting the obtained Jacobian matrices. Using the identity $J_g^{-1} = J_{g^{-1}}$ and the fact that the INN node has access to its inverse, we can obtain inversion-free message and posterior marginal calculations. An interesting direction of future research would be to incorporate more efficient and robust inference operations as in [29]–[31].

VI. Conclusion

This paper has integrated invertible neural networks in probabilistic graphical models. The inference procedure in the network has been generalized for latent inputs and outputs by linearizing the network. Based on local constraints on the Bethe free energy, hybrid inference procedures can be employed on the neural etwork-based model, as illustrated by the experimental binary classifier. This research paves the way for a novel class of machine learning algorithms, where both the strengths of probabilistic graphical models and invertible neural networks are combined.

References

[1] E. G. Tabak and E. Vanden-Eijnden, "Density estimation by dual ascent of the log-likelihood," *Communications in Mathematical Sciences*, vol. 8, no. 1, pp. 217–233, Mar. 2010.

[2] E. G. Tabak and C. V. Turner, "A Family of Nonparametric Density Estimation Algorithms," *Communications on Pure and Applied Mathematics*, vol. 66, no. 2, pp. 145–164, Feb. 2013.

[3] D. J. Rezende and S. Mohamed, "Variational Inference with Normalizing Flows," *arXiv:1505.05770 [cs, stat]*, Jun. 2016.

[4] I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing Flows: An Introduction and Review of Current Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[5] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing Flows for Probabilistic Modeling and Inference," *arXiv:1912.02762 [cs, stat]*, Apr. 2021.

[6] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear Independent Components Estimation," *arXiv:1410.8516 [cs]*, Apr. 2015.

[7] J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J.-H. Jacobsen, "Invertible Residual Networks," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 573–582. [Online]. Available: https://proceedings.mlr.press/v97/behrmann19a.html

[8] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[9] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4282128/

[10] I. Şenöz, T. van de Laar, D. Bagaev, and B. de Vries, "Variational Message Passing and Local Constraint Manipulation in Factor Graphs," *Entropy*, vol. 23, no. 7, p. 807, Jul. 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/7/807

[11] M. Cox, T. van de Laar, and B. de Vries, "A Factor Graph Approach to Automated Design of Bayesian Signal Processing Algorithms," *International Journal of Approximate Reasoning*, vol. 104, pp. 185–204, Jan. 2019. [Online]. Available: http://arxiv.org/abs/1811.03407

[12] G. Forney, "Codes on graphs: normal realizations," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[13] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[14] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Bethe free energy, Kikuchi approximations, and belief propagation algorithms," *Advances in neural information processing systems*, vol. 13, 2001. [Online]. Available: https://www.merl.com/publications/docs/TR2001-16.pdf

[15] J. Pearl, "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach," in *Proceedings of the American Association for Artificial Intelligence National Conference on AI*, Pittsburgh, 1982, pp. 133–136.

[16] J. Winn and C. M. Bishop, "Variational Message Passing," *Journal of Machine Learning Research*, pp. 661–694, 2005.

[17] J. Dauwels, "On Variational Message Passing on Factor Graphs," in *2007 IEEE International Symposium on Information Theory*, Nice, France, Jun. 2007, pp. 2546–2550, iSSN: 2157-8117.

[18] T. P. Minka, "Expectation Propagation for Approximate Bayesian Inference," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 362–369. [Online]. Available: http://dl.acm.org/citation.cfm?id=2074022.2074067

[19] S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Institute of Mathematical Statistics Textbooks. Cambridge: Cambridge University Press, 2013. [Online]. Available: https://www.cambridge.org/core/books/bayesian-filtering-and-smoothing/C372FB31C5D9A100F8476C1B23721A67

[20] E. Petersen, C. Hoffmann, and P. Rostalski, "On Approximate Nonlinear Gaussian Message Passing On Factor Graphs," *2018 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 513–517, Jun. 2018.

[21] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068. International Society for Optics and Photonics, Jul. 1997, pp. 182–193.

[22] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. Lake Louise, Alta., Canada: IEEE, 2000, pp. 153–158. [Online]. Available: http://ieeexplore.ieee.org/document/882463/

[23] J. F. G. d. Freitas, M. Niranjan, A. H. Gee, and A. Doucet, "Sequential Monte Carlo Methods to Train Neural Network Models," *Neural Computation*, vol. 12, no. 4, pp. 955–993, Apr. 2000.

[24] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, Jan. 2017. [Online]. Available: https://epubs.siam.org/doi/10.1137/141000671

[25] D. Bagaev and B. de Vries, "Reactive Message Passing for Scalable Bayesian Inference," *arXiv:2112.13251 [cs]*, Dec. 2021. [Online]. Available: http://arxiv.org/abs/2112.13251

[26] J. Revels, M. Lubin, and T. Papamarkou, "Forward-Mode Automatic Differentiation in Julia," *arXiv:1607.07892 [cs]*, Jul. 2016. [Online]. Available: http://arxiv.org/abs/1607.07892

[27] P. K Mogensen and A. N Riseth, "Optim: A mathematical optimization package for Julia," *Journal of Open Source Software*, vol. 3, no. 24, p. 615, Apr. 2018. [Online]. Available: http://joss.theoj.org/papers/10.21105/joss.00615

[28] M. Cox and B. De Vries, "Robust Expectation Propagation in Factor Graphs Involving Both Continuous and Binary Variables," in *2018 26th European Signal Processing Conference (EUSIPCO)*. Rome: IEEE, Sep. 2018, pp. 2583–2587. [Online]. Available: https://ieeexplore.ieee.org/document/8553490/

[29] L. Bruderer, "Input Estimation and Dynamical System Identification: New Algorithms and Results," Ph.D. dissertation, ETH Zurich, Zurich, Switzerland, 2015. [Online]. Available: http://hdl.handle.net/20.500.11850/155268

[30] F. Wadehn, L. Bruderer, V. Sahdeva, and H. Loeliger, "New square-root and diagonalized Kalman smoothers," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016, pp. 1282–1290.

[31] F. Wadehn, "State Space Methods with Applications in Biomedical Signal Processing," Ph.D. dissertation, ETH Zurich, Jun. 2019, accepted: 2019-05-31T06:41:22Z ISBN: 9783866286405 Publisher: ETH Zurich. [Online]. Available: https://www.research-collection.ethz.ch/handle/20.500.11850/344762