Published by IOP Publishing for Sissa Medialab

Received: *March 22, 2024*
Revised: *June 23, 2024*
Accepted: *July 11, 2024*
Published: *August 28, 2024*

Artificial Intelligence for the Electron-Ion Collider
Washington, D.C., U.S.A.
28 November – 1 December 2023

Bayesian Neural Network Variational Autoencoder Inverse Mapper (BNN-VAIM) and its application in Compton Form Factors extraction

MD Fayaz Bin Hossen,[a] **Tareq Alghamdi,**[a,b] **Manal Almaeen**[c] **and Yaohang Li**[a,*]

[a] *Old Dominion University,*
Norfolk, VA 23529, U.S.A.
[b] *Al-Baha University,*
Al-Baha, Alaqiq 65779, Saudi Arabia
[c] *Jouf University,*
Sakaka 72341, Aljouf, Saudi Arabia

E-mail: yaohang@cs.odu.edu

Abstract: We extend the Variational Autoencoder Inverse Mapper (VAIM) framework for the inverse problem of extracting Compton Form Factors (CFFs) from deeply virtual exclusive reactions, such as the unpolarized Deeply virtual exclusive scattering (DVCS) cross section. VAIM is an end-to-end deep learning framework to address the solution ambiguity issue in ill-posed inverse problems, which comprises of a forward mapper and a backward mapper to simulate the forward and inverse processes, respectively. In particular, we incorporate Bayesian Neural Network (BNN) into the VAIM architecture (BNN-VAIM) for uncertainty quantification. By sampling the weights and biases distributions of the BNN in the backward mapper of the VAIM, BNN-VAIM is able to estimate prediction uncertainty associated with each individual solution obtained for an ill-posed inverse problem. We first demonstrate the uncertainty quantification capability of BNN-VAIM in a toy inverse problem. Then, we apply BNN-VAIM to the inverse problem of extracting 8 CFFs from the unpolarized DVCS cross section.

Keywords: Analysis and statistical methods; Data processing methods

*Corresponding author.

https://doi.org/10.1088/1748-0221/19/08/C08003

Contents

1 Introduction

Understanding the structure and dynamics of the nucleon is an important goal in modern nuclear physics. Deeply virtual exclusive scattering (DVCS) processes are considered as the golden channel for the extraction of information on partonic 3D dynamics in the nucleon [1, 2]. The Compton Form Factors (CFFs) [3], measured in DVCS, are two-dimensional slices of the Generalized Parton Distributions (GPDs) [2, 4, 5], which offer a universal way to characterize the nucleon structure in multi-dimension. Extracting CFFs has the potential to reveal new information on hadronic structure.

In our previous work, we attempt to extract CFFs from a single unpolarization observable [6]. We model the CFFs extraction as an inverse problem by inferring 8 CFFs from the unpolarized DVCS cross section. We employ an end-to-end deep learning architecture, so-called Variational Autoencoder Inference Mapper (VAIM), to handle the ambiguity issue in the ill-posed inverse problem of extracting CFFs. VAIM is an autoencoder-based deep neural network architecture [7]. The forward mapping and backward mapping are approximated by the encoder and decoder, respectively, and a variational latent layer is used to learn the posterior parameter distributions with respect to the given observable. The VAIM framework can be extended to accommodate unstructured and unordered experimental observables when point cloud representation is incorporated [8]. As a result, when VAIM is applied to CFFs extraction, by sampling the latent layer, VAIM is able to derive multiple potential CFFs solutions with respect to a given cross section measurement.

However, an important piece missing in the VAIM architecture is, when there are multiple solutions available, the uncertainty associated with each solution is unknown. To address this issue, in this work, we extend VAIM into so-called BNN-VAIM by replacing the traditional artificial neural network (ANN) in the decoder architecture with a Bayesian Neural Network (BNN). BNNs [9–11] are stochastic machine learning models typically used for uncertainty quantification in predictions.

2024 JINST 19 C08003

Associated with VAIM, BNN-VAIM is designed to provide an estimation of the uncertainty associated with each individual solution when multiple solutions are extracted in an ill-posed inverse problem. By applying BNN-VAIM to CFFs extraction, we are able to put a stochastic error bar on each CFF derived from a given cross section measurement.

2 Background

2.1 Inverse problems

In mathematical modeling, a forward problem is to find the observables y from the given parameters x using a mathematical model $f(\cdot)$. An inverse problem is an opposite procedure of a forward problem, which attempts to recover the parameters x from given observable y [12, 13]. A forward problem is usually well-posed, i.e., for a given x, there is a unique solution y. Whereas an inverse problem is often ill-posed, which means that its solution is not necessarily unique. An illustrative example of an inverse problem involves diagnosing a disease based on observed symptoms. The forward model maps a disease to its associated symptoms; for instance, influenza might indicate symptoms such as fever, cough, and fatigue. Conversely, the inverse problem involves deducing the disease from symptoms, meaning that given the symptoms, one can infer the potential diseases. Clearly, in this example of inverse problem, the possible solutions are not unique. In addition to influenza, other diseases, such as RSV, COVID-19, and Pheumonia, can exhibit the similar symptoms of fever, cough, and fatigue. Figure 1 shows the relationship between a forward problem and an inverse problem.

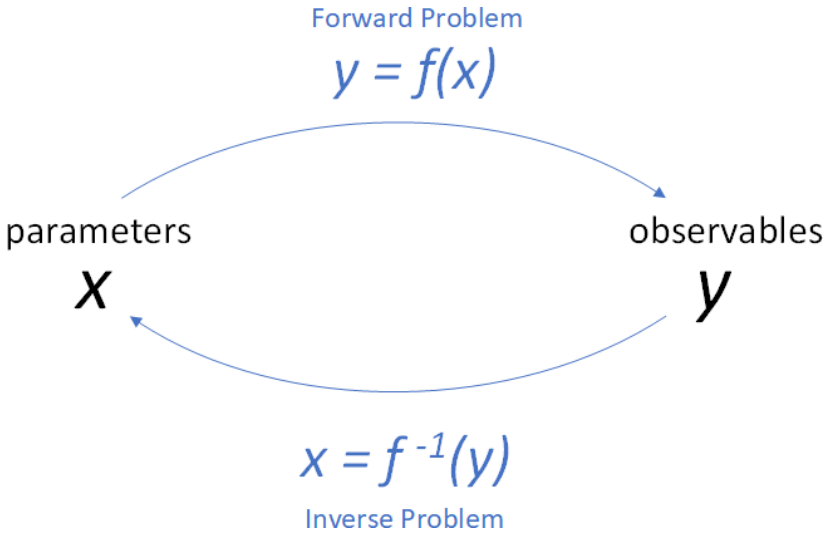

Figure 1. The forward problem and the inverse problem.

2.2 Variational Autoencoder Inverse Mapper (VAIM)

VAIM is an autoencoder-based deep learning approach for solving the inverse problems, particular when non-unique solutions are available [7]. The architecture of VAIM consists of two ANNs, a forward mapper as an encoder from parameters to observables and a backward mapper as a decoder from observables back to parameters. The key component of the VAIM architecture is a latent layer incorporated in between the forward mapper and the backward mapper, serving the purpose of

2024 JINST 19 C08003

capturing the information loss in forward mapping from parameters to observables, which is later used to reconstruct the parameter distribution from the given observables. In VAIM, an inverse problem is modeled as a statistical inverse problem by approximating a probability distribution of parameters when an instance of observables is given. Figure 2 shows the general architecture of VAIM.

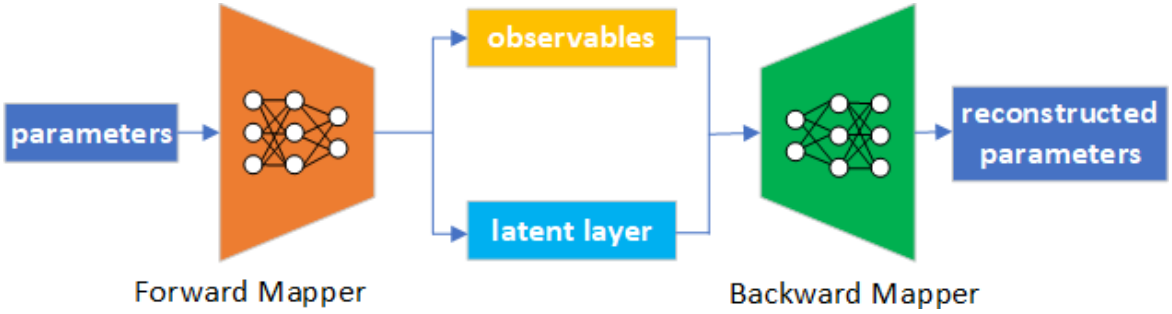

Figure 2. The general architecture of VAIM.

2.3 VAIM for CFFs extraction

The VAIM architecture is applied to the CFFs extraction problem, which is framed as an inverse problem of estimating the CFFs as the parameters from the measured unpolarized DVCS cross section as the observables [6]. A forward model based on Bethe-Heitler (BH) scattering is used to generate the cross sections [14, 15], which depend on 8 CFFs, $\Re e\mathcal{H}$, $\Im m\mathcal{H}$,$\Re e\mathcal{E}$,$\Im m\mathcal{E}$,$\Re e\widetilde{\mathcal{H}}$,$\Im m\widetilde{\mathcal{H}}$,$\Re e\widetilde{\mathcal{E}}$, and $\Im m\widetilde{\mathcal{E}}$. The forward mapping between the CFFs and the cross sections are used to generate the training data to train the VAIM. Figure 3 illustrates the VAIM architecture for CFFs extraction.

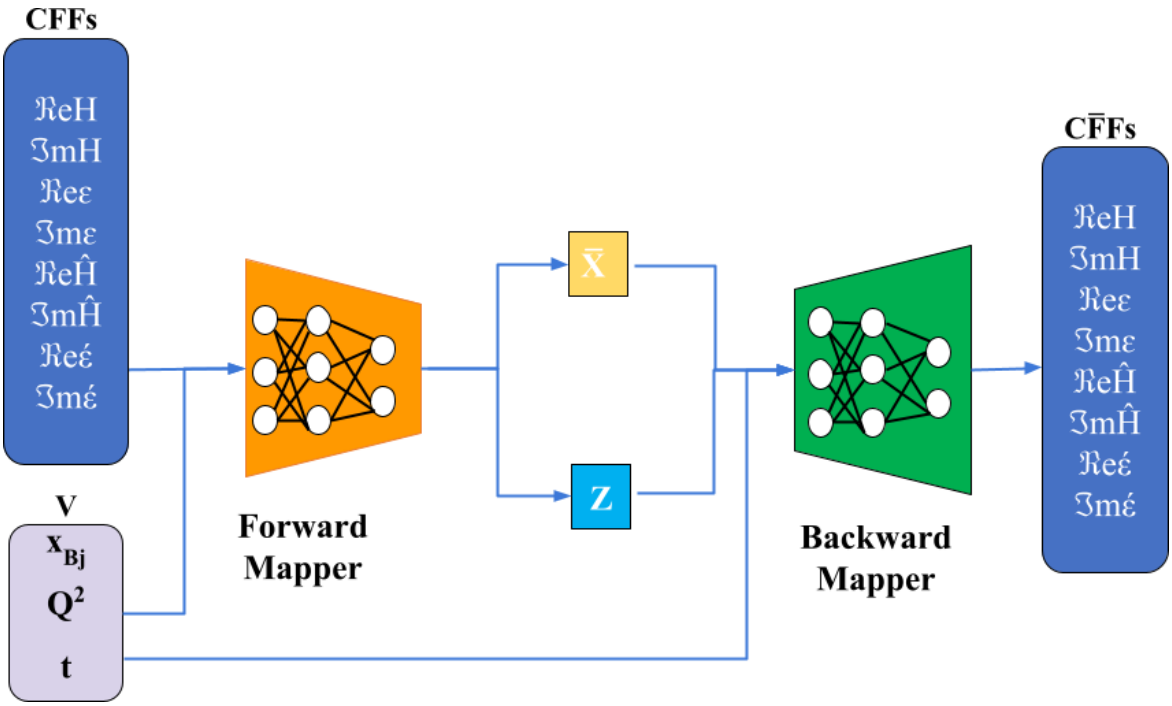

Figure 3. The VAIM architecture for CFFs extraction.

2024 JINST 19 C08003

2.4 Bayesian Neural Networks (BNNs)

BNNs are stochastic neural networks whose weights are probability distributions instead of fixed numerical values in ANNs. BNNs are trained using Bayesian inference, allowing one to estimate uncertainty in predictions by sampling the probability distribution of weights and biases. In a BNN where the weights and biases are represented as probability distributions, the output y of the BNN with respect to input x becomes a probability distribution such that

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D)d\theta,$$

where θ represents the BNN parameters, such as weights and biases, $p(y|x, \theta)$ is the likelihood function, and $p(\theta|D)$ is the Bayesian posterior distribution over θ given the observed data D. In particular, $p(\theta|D)$ can be calculated by Bayesian inference such that

$$p(\theta|D) = \frac{p(D|\theta) \cdot p(\theta)}{\int_{\theta'} P(D|\theta')P(\theta')},$$

where $p(\theta)$ is the prior and $p(D|\theta)$ is the likelihood. However, computing the Bayesian posterior distribution and sampling from it are usually intractable due to the difficulty of computing the integral. In practice, either Markov Chain Monte Carlo [16], approximating the integral by sampling, or variational inference [17], approximating the integral by optimization, are used to estimate the Bayesian posterior distribution $p(\theta|D)$. A detailed description and analysis of BNNs can be found in the tutorial by Jospin et al. [18]

3 Methodology

3.1 BNN-VAIM architecture

To quantify the uncertainty associated with the each individual solution found in VAIM, we incorporate BNN into the VAIM architecture (BNN-VAIM). More specifically, we replace the ANN in the backward mapper with a BNN. Similar to VAIM, BNN-VAIM comprises a forward mapper and a backward mapper. While the forward mapper is still an ANN that approximate the forward model from the parameters to the observables, the backward mapper is a BNN where the weights and biases are probability distributions instead of fixed numerical values in an ANN.

In our BNN-VAIM implementation, BNN adopts a Gaussian prior distribution for the weights and biases. These prior parameters represent our initial beliefs about the distributions of weights and biases before observing any data. During the forward pass, BNN uses forward computation to generate the parameters from the given observables. Instead of learning a single set of weights and biases, the BNN learns the posterior distribution over weights and biases, which represents the updated beliefs about the weights and biases after observing the data. During training, we use variational inference to approximate the true posterior distribution [19, 20], which is a Gaussian distribution in our implementation by maximizing the log likelihood of the data given the weights and biases as the BNN parameters. We minimize the Kullback-Leibler (KL)-divergence between the approximated posterior distributions and the prior, with the purpose of avoiding overfitting.

After the BNN-VAIM is properly trained, the backward mapper is used for solving the inverse problem, as shown in figure 4(b). First of all, similar to VAIM, feeding the given observables as

2024 JINST 19 C08003

input and sampling the latent layer generate multiple solutions of the parameters. Then, for each solution corresponding to the observables and the latent space input, sampling the distributions of the BNN weights and biases provides an estimation of its uncertainty.

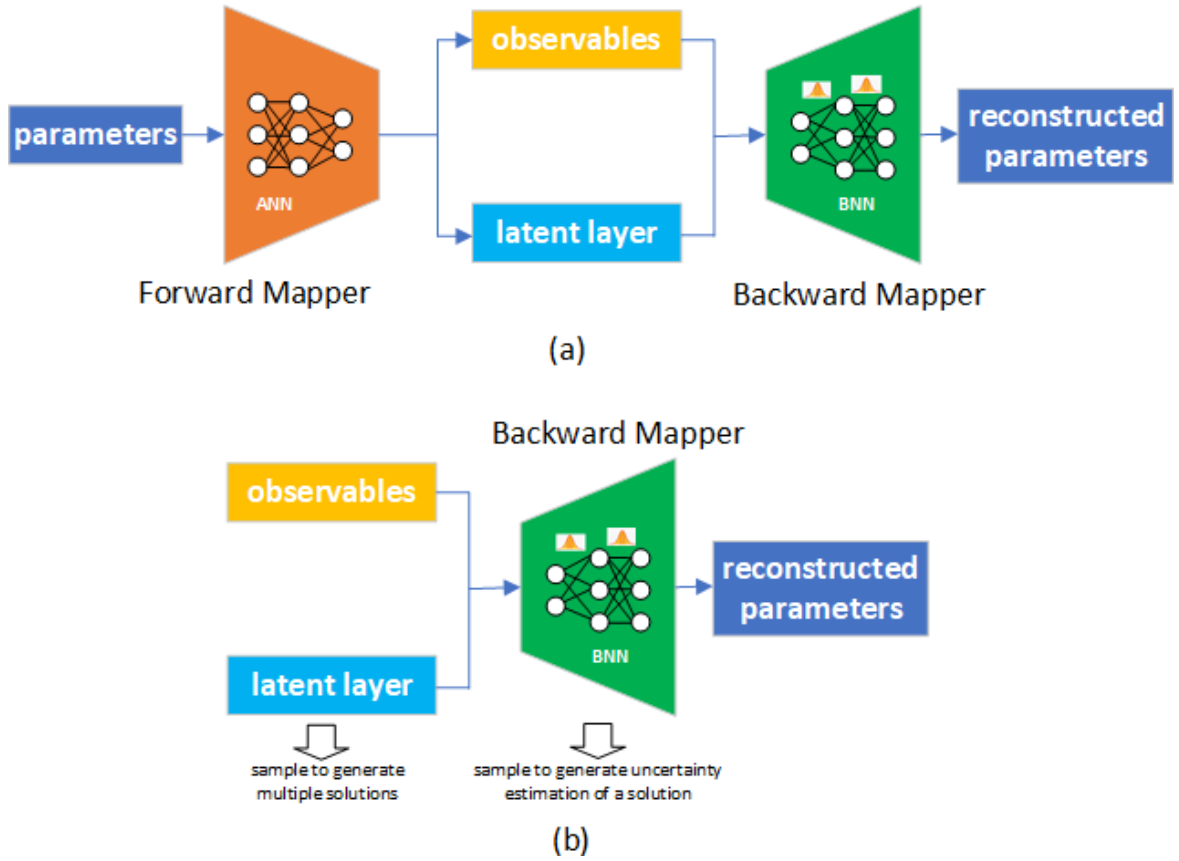

Figure 4. VAIM for CFFs extraction.

3.2 BNN-VAIM for CFFs extraction

Figure 5 shows the architecture of BNN-VAIM for CFFs extraction. Here the VAIM is implemented as a conditional Variational Auto Encoder (VAE), where x_{Bj}, Q^2, and t are treated as the input conditions for both the forward mapper and the backward mapper. The backward mapper adopts a BNN architecture. During training, the BNN-VAIM attempts to minimize the loss of reconstructing the cross-section x, reconstructing the 8 CFFs, $\Re e\mathcal{H}$, $\Im m\mathcal{H}$,$\Re e\mathcal{E}$,$\Im m\mathcal{E}$,$\Re e\widetilde{\mathcal{H}}$,$\Im m\widetilde{\mathcal{H}}$,$\Re e\widetilde{\mathcal{E}}$, and $\Im m\widetilde{\mathcal{E}}$, the KL-divergence of the distribution of the latent variable Z, and the KL-divergence of the posterior distribution of BNN to the Gaussian prior distribution. Once trained, by sampling the latent layer Z, the backward mapper generates the potential CFFs solutions with respect to the given cross-section and the condition of x_{Bj}, Q^2, and t. Sampling the distribution of weights and biases provides an uncertainty estimation for each CFF solution.

2024 JINST 19 C08003

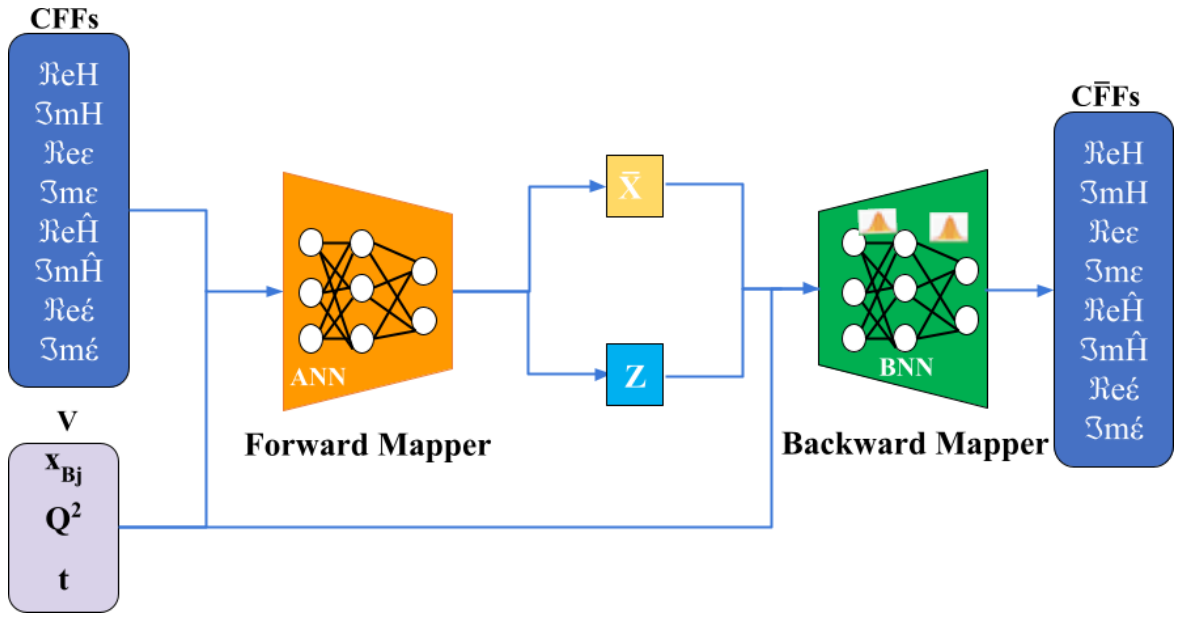

Figure 5. BNN-VAIM for CFFs extraction.

4 Results and discussion

In this section, we first demonstrate the effectiveness of BNN-VAIM on a toy inverse problem with ambiguous solutions. Then, we apply BNN-VAIM to extracting CFFs from the unpolarized DVCS cross section.

4.1 Toy inverse problem $f(x) = x^2$

We consider the following toy inverse problem of predicting x by given $f(x)$:

$$f(x) = x^2.$$

This toy inverse problem generally has two distinct solutions for every given $f(x)$, except for $f(x) = 0$.

We train the BNN-VAIM with 10^6 data samples uniformly distributed between $x = -2$ to $x = 2$. Smearing noise is introduced to the training data by allowing data to deviate within 5% of its original values. Both the forward mapper and the backward mapper employ fully connected networks with 3 layers, each layer having 512 hidden nodes. The dimension of the latent layer is 100. The dimension of the latent space is a hyperparameter of VAIM. Provided that the dimension of the latent space is not less than the dimension of the actual information lost during the forward mapping process, VAIM can in theory rebuild the complete parameter distribution from the observables. A larger latent space dimension provides more flexibility for VAIM to represent the lost information. However, a too large latent space dimension can slow down the training process, due to the fact of the resulting large neural networks for the forward and backward mappers. The prior of the latent variables is a uniform distribution. The prior of the BNN adopts a standard Gaussian distribution. We use Adam [21] as the optimizer with learning rate $1e^-5$. Training concludes once the reconstruction error falls below 10^{-3}.

The histogram in blue in figure 6 shows the distribution of the multiple x solutions with respect to $f(x) = 1$ by sampling the latent layer. Then, for a solution obtained, we sample the weights and

2024 JINST 19 C08003

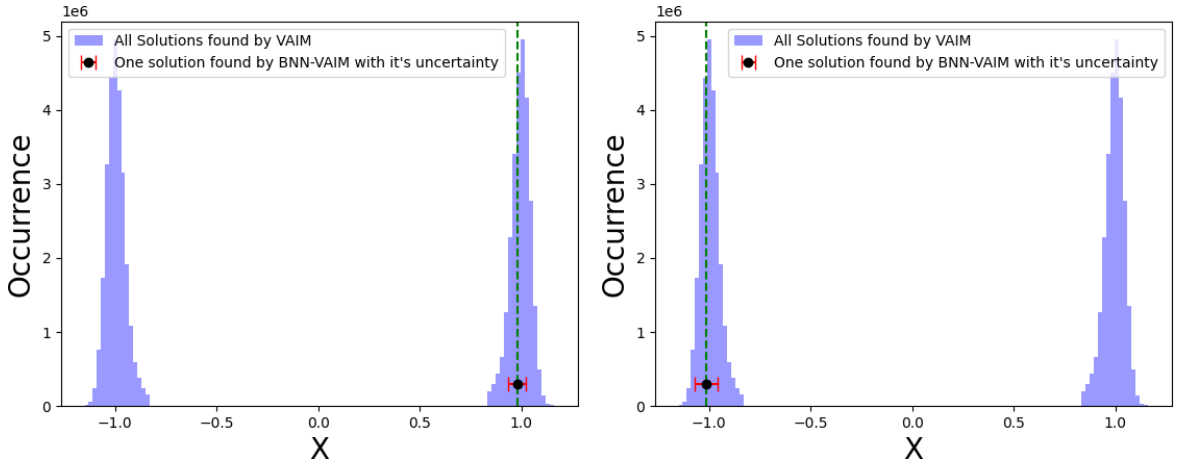

Figure 6. Predicted solution distributions of $f(x) = x^2$ and the estimated uncertainties of two solutions using BNN-VAIM.

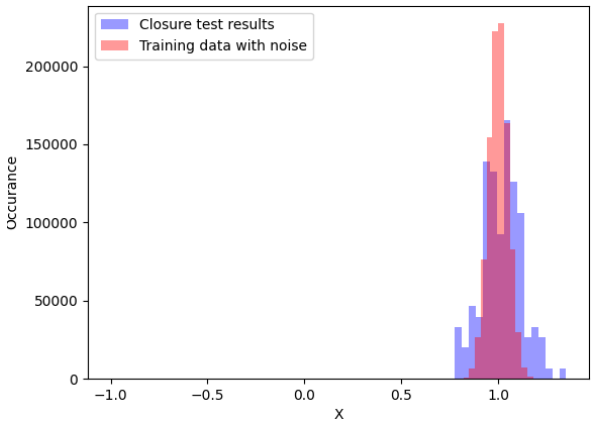

Figure 7. Closure test of $f(x) = x^2$ for $f(x) = 1$. The training data has a mean of 0.999, a standard deviation of 0.0500, a kurtosis of -0.0054, and a skewness of -0.0015. The closure test results show a mean of 1.019, a standard deviation of 0.102, a kurtosis of 0.212, and a skewness of 0.0856.

biases in BNN to generate the uncertainty estimation. Figures 6(a) and 6(b) show two individual solutions with error bars in 1 standard deviation, corresponding to $x = 1$ and $x = -1$, respectively. Figure 7 shows a closure test by sampling an individual solution within its uncertainty estimation and feeding the samples to the forward model to reconstruct $f(x)$. One can find that, although somehow over-estimating, the reconstructed $f(x)$ has good coverage of the training data within its noise range.

4.2 BNN-VAIM for CFFs extraction

We apply BNN-VAIM to extracting CFFs from the unpolarized DVCS cross-section. The forward mapper of BNN-VAIM includes an input layer representing the array of CFFs concatenated with the kinematics array, followed by three fully-connected layers comprising 2,048 neurons each, activated by a Parameterized ReLU function [22]. The output layer encompasses the cross section and latent

2024 JINST 19 C08003

variables (z). The backward mapper contains an input layer representing the concatenated cross section, kinematics, and latent variable layer (z), followed by three fully-connected Bayesian linear layers with 2,048 neurons each, activated by a Parameterized ReLU function. The output layer reconstructs the CFFs. We use 225,000 samples generated by BH scattering to train the BNN-VAIM model. Adam is used as the optimizer with learning rate $1e^{-}5$. We adopt an adaptive training, which terminates when the reconstruction error is less than 10^{-4}.

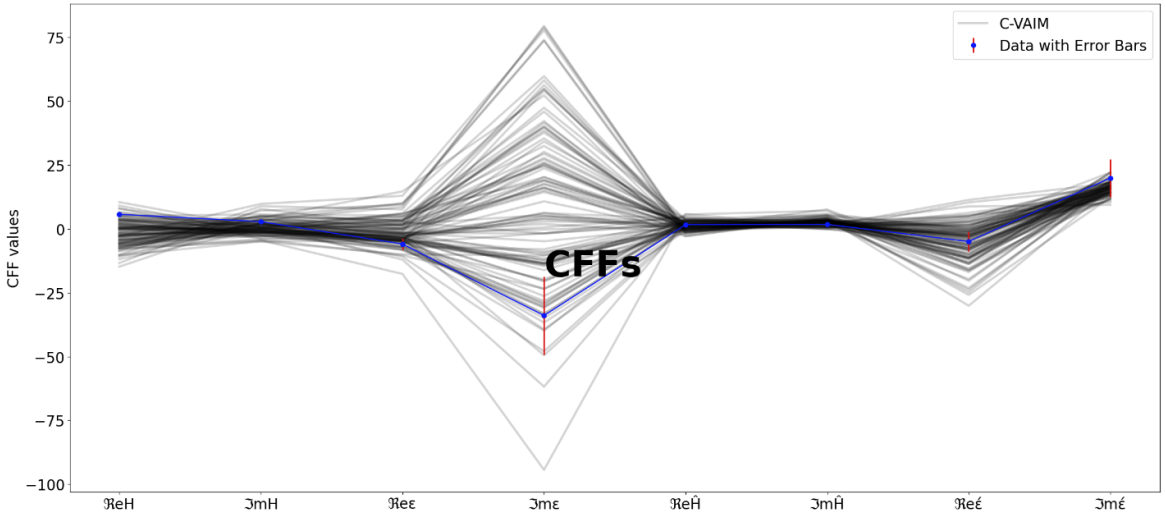

Figure 8. Predicted CFFs using BNN-VAIM at a fixed kinematic point $x_{Bj} = 0.343$, $t = 0.172\,\mathrm{GeV}^2$, and $Q^2 = 1.820\,\mathrm{GeV}^2$. 100 solutions are showed in grey lines. The uncertainty of one of these solutions, highlighted in blue line, are specified with error bar obtained by BNN.

Figure 8 shows the results using BNN-VAIM to predict the CFFs from a singular kinematic point $x_{\mathrm{Bj}} = 0.343$, $t = -0.172\,\mathrm{GeV}^2$, and $Q^2 = 1.82\,\mathrm{GeV}^2$ with respect to the unpolarized DVCS cross-section measurement at a beam energy of 5.75 GeV. While sampling the latent variables in VAIM generates an ensemble of 100 potential CFF solutions (grey lines in figure 8), sampling the BNN in the backward mappers provides an uncertainty estimation (red bar in figure 8) of one of these solutions. Compared to VAIM for CFF extraction, which provides the overall CFF solution distribution, BNN-VAIM further estimate the uncertainty associated with each solution. The size of the error bar indicates the sensitivity of each CFF solution. One can find that in the highlighted CFF solution in figure 8, $\Im m\mathcal{E}$ is most flexible while $\Re e\widetilde{\mathcal{H}}$ and $\Im m\widetilde{\mathcal{H}}$ are most constrained, which is quite consistent with the overall distribution of the CFF solutions.

5 Conclusion

In this paper, we incorporate BNN into our VAIM architecture for uncertainty quantification. By sampling the weights and biases distributions of the BNN in the backward mapper of the VAIM, BNN-VAIM enables the estimation of prediction uncertainty associated with each individual solution found in an ill-posed inverse problem. The uncertainty quantification capability of BNN-VAIM is demonstrated in a toy inverse problem of $f(x) = x^2$. We further apply BNN-VAIM to a CFFs extraction problem, whose goal is to extract 8 CFFs from an unpolarized DVCS cross-section. Despite

2024 JINST 19 C08003

the ill-posedness of the CFFs extraction problem, the uncertainty with respect to each CFF indicates its sensitivity to the cross-section measurement.

Acknowledgments

The work is partially supported by the DOE Research on Artificial Intelligence and Machine Learning For Autonomous Optimization And Control Of Accelerators And Detectors grant, under award number DE-SC0024603, by the Center for Nuclear Femtography (CNF), and by the 2023 ODU Data Science Seed Grant. We acknowledge the discussion of this work in the EXCLAIM collaboration with Simonetta Liuti, Huey-Wen Lin, Matthew Sievert, Gary Goldstein, Marie Boer, Gia-Wei Chern, Dennis Sivers, Michael Engelhardt, Douglas Adams, and Marija Čuić.

References

[1] J.C. Collins, L. Frankfurt and M. Strikman, *Factorization for hard exclusive electroproduction of mesons in QCD*, *Phys. Rev. D* **56** (1997) 2982 [hep-ph/9611433].

[2] D. Müller et al., *Wave functions, evolution equations and evolution kernels from light ray operators of QCD*, *Fortsch. Phys.* **42** (1994) 101 [hep-ph/9812448].

[3] X.-D. Ji, *Deeply virtual Compton scattering*, *Phys. Rev. D* **55** (1997) 7114 [hep-ph/9609381].

[4] X.-D. Ji, *Gauge-Invariant Decomposition of Nucleon Spin*, *Phys. Rev. Lett.* **78** (1997) 610 [hep-ph/9603249].

[5] A.V. Radyushkin, *Nonforward parton distributions*, *Phys. Rev. D* **56** (1997) 5524 [hep-ph/9704207].

[6] M. Almaeen et al., *VAIM-CFF: A variational autoencoder inverse mapper solution to Compton form factor extraction from deeply virtual exclusive reactions*, arXiv:2405.05826.

[7] M. Almaeen, Y. Alanazi, N. Sato, W. Melnitchouk, M.P. Kuchera and Y. Li, *Variational Autoencoder Inverse Mapper: An End-to-End Deep Learning Framework for Inverse Problems*, in the proceedings of the *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China (2021), p. 1–8 [DOI:10.1109/ijcnn52387.2021.9534012].

[8] M. Almaeen et al., *Point Cloud-based Variational Autoencoder Inverse Mappers (PC-VAIM) — An Application on Quantum Chromodynamics Global Analysis*, in the proceedings of the *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, Nassau, Bahamas (2022), p. 1151–1158 [DOI:10.1109/ICMLA55696.2022.00187].

[9] J. Lampinen and A. Vehtari, *Bayesian approach for neural networks — review and case studies*, *Neural Networks* **14** (2001) 257.

[10] D.M. Titterington, *Bayesian Methods for Neural Networks and Related Models*, *Statist. Sci.* **19** (2004) 128.

[11] E. Goan and C. Fookes, *Bayesian Neural Networks: An Introduction and Survey*, in *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, K.L. Mengersen, P. Pudlo and C.P. Robert eds., Springer International Publishing (2020), p. 45–87 [arXiv:2006.12024] [DOI:10.1007/978-3-030-42553-1_3].

[12] A.G. Ramm, *Uniqueness result for inverse problem of geophysics: I*, *Inverse Prob.* **6** (1990) 635.

[13] Y. Song, L. Shen, L. Xing and S. Ermon, *Solving Inverse Problems in Medical Imaging with Score-Based Generative Models*, arXiv:2111.08005.

[14] S.D. Drell and J.D. Walecka, *Electrodynamic Processes with Nuclear Targets*, *Annals Phys.* **28** (1964) 18.

2024 JINST 19 C08003

[15] T. Gehrmann and M. Stratmann, *The Bethe-Heitler process in polarized photon-nucleon interactions*, *Phys. Rev. D* **56** (1997) 5839 [hep-ph/9706414].

[16] W.K. Hastings, *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, *Biometrika* **57** (1970) 97.

[17] D.M. Blei, A. Kucukelbir and J.D. McAuliffe, *Variational Inference: A Review for Statisticians*, *J. Am. Statist. Assoc.* **112** (2017) 859 [arXiv:1601.00670].

[18] L.V. Jospin et al., *Hands-On Bayesian Neural Networks — A Tutorial for Deep Learning Users*, *IEEE Comput. Intell. Mag.* **17** (2022) 29.

[19] A. Graves, *Practical variational inference for neural networks*, in *Adv. Neural Inf. Process. Syst.*, vol. 24, J. Shawe-Taylor et al. eds., Curran Associates, Inc. (2011).

[20] G.E. Hinton and D. Van Camp, *Keeping the neural networks simple by minimizing the description length of the weights*, in *6COLT93: Proceedings of the sixth annual conference on Computational learning theory*, Santa Cruz, California, U.S.A., 26–28 July 1993, p. 5–13 [DOI:10.1145/168304.168306].

[21] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, in the proceedings of the *3rd International Conference for Learning Representations (ICLR)*, San Diego, CA, U.S.A., 7–9 May 2015 [arXiv:1412.6980].

[22] K. He, X. Zhang, S. Ren and J. Sun, *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*, in the proceedings of the *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile (2015), p. 1026–1034 [DOI:10.1109/ICCV.2015.123] [arXiv:1502.01852].

2024 JINST 19 C08003