Old Dominion University
ODU Digital Commons

Computer Science Theses & Dissertations | Computer Science

Summer 8-2023

# Inverse Mappers for QCD Global Analysis

Manal Almaeen
*Old Dominion University*, malma004@odu.edu

Follow this and additional works at: https://digitalcommons.odu.edu/computerscience_etds

 Part of the Artificial Intelligence and Robotics Commons, Elementary Particles and Fields and String Theory Commons, and the Nuclear Commons

## Recommended Citation

Almaeen, Manal. "Inverse Mappers for QCD Global Analysis" (2023). Doctor of Philosophy (PhD), Dissertation, Computer Science, Old Dominion University, DOI: 10.25777/vz9r-b385
https://digitalcommons.odu.edu/computerscience_etds/141

This Dissertation is brought to you for free and open access by the Computer Science at ODU Digital Commons. It has been accepted for inclusion in Computer Science Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

# INVERSE MAPPERS FOR QCD GLOBAL ANALYSIS

by

Manal Almaeen
B.S. July 2010, Jouf University, Saudi Arabia
M.S. April 2017, University of St. Thomas

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2023

Approved by:

Yaohang Li (Director)

Lusi Li (Member)

Fengjiao Wang (Member)

Nobuo Sato (Member)

Brandon Kriesten (Member)

## ABSTRACT

INVERSE MAPPERS FOR QCD GLOBAL ANALYSIS

Manal Almaeen
Old Dominion University, 2023
Director: Dr. Yaohang Li

Inverse problems – using measured observations to determine unknown parameters – are well motivated but challenging in many scientific problems. Mapping parameters to observables is a well-posed problem with unique solutions, and therefore can be solved with differential equations or linear algebra solvers. However, the inverse problem requires backward mapping from observable to parameter space, which is often nonunique. Consequently, solving inverse problems is ill-posed and a far more challenging computational problem.

Our motivated application in this dissertation is the inverse problems in nuclear physics that characterize the internal structure of the hadrons. We first present a machine learning framework called Variational Autoencoder Inverse Mapper (VAIM), as an autoencoder based neural network architecture to construct an effective "inverse function" that maps experimental data into QCFs. In addition to the well-known inverse problems challenges such as ill-posedness, an application specific issue is that the experimental data are observed on kinematics bins which are usually irregular and varying. To address this ill defined problem, we represent the observables together with their kinematics bins as an unstructured, high-dimensional point cloud. The point cloud representation is incorporated into the VAIM framework. Our new architecture point cloud-based VAIM (PC-VAIM) enables the underlying deep neural networks to learn how the observables are distributed across kinematics.

Next, we present our methods of extracting the leading twist Compton form factors (CFFs) from polarization observables. In this context, we extend VAIM framework to the Conditional -VAIM to extract the CFFs from the DVCS cross sections on several kinematics.

Connected to this effort is a study of the effectiveness of incorporating physics knowledge into machine learning. We start this task by incorporating physics constraints to the forward problem of mapping the kinematics to the cross sections. First, we develop Physics Constrained Neural Networks (PCNNs) for Deeply Virtual Exclusive Scattering (DVCS) cross sections by integrating some of the physics laws such as the symmetry constraints of the cross sections. This provides us with an inception of incorporating physics rules into our inverse mappers which will one of the directions of our future research.

Copyright, 2023, by Manal Almaeen, All Rights Reserved.

To my parents, my husband, and my children for their endless love and support.

# ACKNOWLEDGMENTS

This dissertation would not have been possible without the contribution of a great group of supporters. I first would like to thank my advisor Dr. Yaohang Li for his tremendous guidance, support, and encouragement during my Ph.D years. He spent countless hours, weekdays and weekends discussing my research and providing direction throughout the program. His expertise and guidance are precious to this dissertation. I also would like to thank the internal committee members: Dr. Lusi Li and Dr. Fengjiao Wang for their great comments, support and motivation. I also thank the CS graduate program director Dr. Michele Weigle and Himarsha Jayanetti for their help and guidance in the dissertation formats.

I would like to express my appreciation to Dr. Nobuo Sato from Jefferson Lab for introducing me to the field of Nuclear Femtography and inverse problems. He had a significant role in providing the physics data and insights into the success of this research. I appreciate all the time and effort he spent discussing and guiding my research.

I would like to thank Dr. Brandon Kriesten from the Center for Nuclear Femtography for providing me the knowledge about the GPDs. His significant efforts in providing the physics calculations and insights were critical to the success of my research. He continuously provided me with the direction and ideas that helped me to accomplish my dissertation.

I also would like to thank Dr. Simonetta Liuti from the University of Virginia for her significant role in leading and enhancing parts of my research. Her comments and suggestions were substantial to improve my work.

I also thank the Center for Nuclear Femtography committee members, Dr. Xiangdong Ji, Dr.

Latifa Elouadrhiri and Dr. David Richards for the valuable discussions during the fellowship meetings. Their comments, feedbacks and suggestions were important to the success of this work.

I also would like to thank our collaborators Dr. Wally Melnitchouk (Jefferson Lab), Dr. Yasir Alanazi (Jefferson Lab), Dr. Huey-Wen Lin (Michigan State University) and Dr. Joshua Hoskins (University of Virginia) for their guidance and insightful comments during our weekly meetings.

Finally, this research was generously supported by Jouf University (fully funded) and the Center for Nuclear Femtography at Jefferson Lab (partially funded). Thank you for providing me with this great opportunity.

# TABLE OF CONTENTS

Page

Page

# LIST OF TABLES

# LIST OF FIGURES

Figure Page

# CHAPTER 1

# INTRODUCTION

Recovering the hidden parameters from indirect observation is an inverse problem that is motivated in several science and engineering fields. Since the first half of the 20th century, the study of inverse problems has gained popularity with the first publications were related to nuclear physics (inverse problems of quantum scattering theory [1]), geophysics (inverse problems of electrical prospecting, seismology, and potential theory [2]), astronomy [3], and other fields of science. Like most forward problems, inverse problem is well posed when it satisfies Hadamard's principles [4] of well-posedness: a solution must be existed, a solution is unique and depends continuously on data. Hence reconstructing the solutions is subtle. On the other hand, inverse problems most often are not well posed in the sense of Hadamard and therefore are termed as ill-posed. Generally, ill-posed inverse problems do not have unique solutions, which dramatically affects the complexity and characteristics of the possible solutions.

Several decades of the high-energy scattering experiments have produced a large amount of data for the Quantum Chromodynamics (QCD) [5] analysis to explore the inner structure of protons and neutrons. In the near future, the 12 GeV nuclear physics program upgrade at Jefferson Lab and the planned Electron-Ion Collider (EIC) facility at Brookhaven National Laboratory is expected to deliver unprecedented quantities of high-precision data to reveal the nucleon's internal 3-dimensional image of quark and gluon structure in the theoretical framework of QCD. At present, all QCD analysis that explores nucleon structure or hadronization is based on Bayesian likelihood methods [6, 7, 8]. This is an inverse problem that connects the experimental data to quantum correlation functions (QCFs) that characterize nucleon structure and the emergence of hadrons in terms of quark and gluon degrees of freedom, including parton distribution functions (PDFs) [9], parton to hadron fragmentation functions (FFs) [10], transverse momentum dependent distributions (TMDs) [11], and generalized parton distributions (GPDs) [12, 13], which encode complex information on the internal structure of nucleons.

Similar to many science and engineering applications of recovering parameters from observables, the inverse problem in QCD analysis is ill-posed, where the solution is often non-unique and is sensitive to the change of the observable data. The recent advances of deep learning and data-driven methods have demonstrated that deep neural networks (DNNs) can be effectively used to address the ambiguity issue of the inverse problems. These DNN-based models have certain advantages over classical methods for solving inverse problems such as the potential of incorporating

a prior knowledge and regularization. For example, the work in [14] focuses on the usage of deep neural networks as regularization terms, while the other effort by [15] extracts prior knowledge using a convolutional network as a gradient-like approach to iteratively optimize the solutions.

In this dissertation, we develop a deep learning framework, the Variational Autoencoder Inverse Mapper (VAIM), as an end to-end machine learning architecture to solve inverse problems. VAIM adopts the architecture of an autoencoder [16], consisting of a forward mapper from parameter space to observable space and a backward mapper from observable space to parameter space. The latent space of VAIM captures the conditional posterior distribution of the inverse process. During training, the latent layer learns an estimator, which infers the posterior parameter distribution with respect to the given observables in an unsupervised manner, without explicit knowledge of the inverse problem. Given observables as input, sampling the variables in the latent layers by variational inference [17] then reveals the corresponding parameter distribution. We apply VAIM to several toy inverse problems with different solution characteristics. Since we are particularly interested in targeting an important nuclear physics application of reconstructing the internal structure of nucleons (protons and neutrons) within Quantum Chromodynamics (QCD), we apply VAIM to construct the inverse function mapping quantum correlation functions to experimental observables.

The VAIM model is developed to have the numerical values of the observables across regular, discretized kinematic bins as inputs to the inverse function. However, in reality the experimental data are much ill-defined, where the data in different experiments are observed on different kinematic bins. Thus, we extend the VAIM framework to address the ill-defined, varying observable space problem to allow observable input to be independent of the specific kinematic bin setups. Our solution is to represent the cross section observables together with their kinematic bins as a geometric point in space, where the cross section and the corresponding kinematic bins form their geometric coordinates. We represent all observed data points as a high dimensional point cloud, containing a set of geometric points in the cross section-kinematics space. The data points within the point cloud are unstructured and unordered while the number of data points can be different, too. The Point Cloud-based VAIM (PC-VAIM) accept unstructured observable representation and no longer limited by the discretized, predefined kinematic bins. We adopt the PointNet framework [18], which is designed to be invariant to input permutation, to handle the unordered and unstructured observable data in point cloud representation.

Connected to this work is the inverse problem of extracting the Compton form factors from Deeply Virtual Exclusive Scattering (DVCS) cross sections which is the easiest way to access the GPDs. In this context, we treat the experimental observable as an equation that is parametrized by CFFs, and we utilize the VAIM framework to study the effect of extracting CFFs on a single

kinematics values. Then, we extend our VAIM to the conditional VAIM (C-VAIM) to recover the CFFs on several sets of kinematics. C-VAIM architecture allows the flexibility of conditioning the latent distribution by the kinematics values. Thus, with the training process on the entire kinematics data, our backward mappers can produce CFFs that correspond to a certain kinematics region. We demonstrate our results of CFFs extraction from the unpolarized cross section, training on Jefferson Lab Halls A and B data, 6 GeV and 12 GeV experimental runs.

Finally, we focus on addressing some of the challenges in solving the inverse problems. In particular, purely data-driven models may perfectly fit observations, but predictions might be inconsistent, as a result of extrapolation or observational biases that may lead to poor generalization performance. Thus, there is an imperative need for combining fundamental physical rules and domain knowledge by "teaching" physical laws to machine learning models, which can, in turn, provide "informative priors". In this dissertation, We start this task by implementing a Physics Constrained Neural Networks (PCNNs) for a forward problem by incorporating physics laws to the NNs. In particular, we incorporate the angular symmetry constraints by using additional loss function to ensure that our predicted observables are symmetric. We apply PCNNs on the forward problem of constructing DVCS cross sections from the kinematics. We use the Bethe-Heitler process which is calculable to test PCNNs and, in particular, their ability to generalize information extracted from data. We then test on the DVCS cross section and compare the results to analytic model calculations. In our future work, we plan to extend this work and incorporate such a physics constraints to our inverse mappers for more robust framework.

## 1.1 MOTIVATION

This dissertation is mainly motivated by the following two pillars. First, the use of machine learning based approaches to solve the inverse problems. Section 1.1.1 presents the advantages of machine learning approaches over the traditional methods in solving the inverse problems. Second, the use of machine learning approaches to solve the inverse problems in nuclear physics. Section 1.1.2 describes the importance and advantages of employing machine learning methods to solve the inverse problems in QCD over the classical methods used in the physics community.

### 1.1.1 Machine Learning Based Approaches to Solve the Inverse Problems

Several analytical methods [19, 20, 21] have been utilized for solving the inverse problems that take advantage of domain knowledge to regularize and constrain the problem in order to achieve numerically stable solutions. These analytical approaches show success in solving well posed inverse problems. However, in ill posed inverse problems, traditional analytical methods such

as minimum distance, least squares and maximum likelihood are possibly not uniquely defined and the sensitivity of the solution to small perturbation in the data is often large. In fact, these classical strategies have many drawbacks in requiring explicitly identified prior knowledge, defined regularizers, some limitations in handling noise, computational complexity in inference because of the optimization-based mechanisms, and often limited applicability in the way that every inverse problem requires to be solved one-off. Thus, solving inverse problems is challenging and can be impractical using classical techniques [22].

In recent years, machine learning based approaches have manifested a great potential for solving a wide variety of inverse problems. For example, the effort in [23, 24, 25, 26] demonstrates that neural network-based models have a bright future for solving inverse problems in image processing. As well the reviewed works in [27] illustrate many emerging deep learning solutions to the geophysical inverse problems. With the recent advances of machine learning, deep neural networks (DNNs) have become a practical approach for solving inverse problems, with advantages over traditional methods. First, as shown in machine learning theory [28], neural networks (NNs) can approximate any continuous function, and potentially non-continuous functions, under mild assumptions. Second, as a data-driven approach, DNNs can lead to continuous performance improvement when large amounts of data are available. Third, regularization can be incorporated into NNs training to stabilize the inversion process. Finally, using a unified NNs architecture and an appropriate learning algorithm, it is possible to design an end-to-end model that performs well without reliance on sophisticated prior knowledge about the inverse problem, despite its inherent complexity.

Connected with the success and stability of solving the inverse problems using deep learning models is the implicit or explicit use of prior information about the unknown parameters to be reconstructed. Many of recent research studies the integration of prior knowledge into DNN solutions for inverse problems. Particularly, the use of regularization techniques and intelligent initialization of DNN weights form the main classes of existing work in this domain [29].

### 1.1.2 Machine Learning Based Approaches to Solve the Inverse Problems in Nuclear Physics

In nuclear physics, Bayesian inference-based approaches are classical methods for the inverse problem in QCD analysis [6, 7, 8]. Theory-based parameterization is a typical Bayesian inference method by fine-tuning the parameters to match the data through maximum likelihood estimators [30, 31]. Theory-based parameterization has been applied in QCD global analysis [32] based on the Hessian matrix optimization and has been widely used by several groups across the world [33, 34, 35, 36]. Recently, the emergence of Monte Carlo based strategies for Bayesian in-

ference [37, 30, 31] has been further applied to simultaneous extraction of different types of QCFs such as spin-dependent PDFs [38] and TMDs [39].

Neural Networks have also been explored for QCD global analysis. NNPDF [40, 41] and N3fit [42] groups adopt neural network parameterizations for the QCF modeling. In the context of QCD analysis, the modern machine learning approaches such as DNNs offer attractive advantages over traditional Bayesian inference approaches. First, DNNs are significantly less computationally costly than Bayesian inference modeling, due to the fact that the costly Monte Carlo sampling is no longer needed. Utilizing DNNs means that all of the time-consuming tasks of generating the samples and training DNNs need to be computed only once and then using the trained model, we can easily test on any example. However, for Monte Carlo sampling approach, we need to do the fitting for each test example which is computationally intensive and time consuming as it relays on the repeated sampling. Second, DNNs can directly reconstruct the inverse function from cross sections to the parameter space. The machine learning approaches also allow flexibility of data input to provide answers to questions such as where in the phase space need more experiments to collect data. Third, compared to the static LHAPDF tables [43] resulted from QCD global analysis, the DNN-based inverse mapper provides a dynamic representation for hadron structure data.

### 1.2 RESEARCH QUESTIONS

The main objective of this research is developing a machine learning framework to solve the inverse problems in QCFs which encode complex information on the internal structure of nucleons. We first aim to accurately connect the observables to the PDF space using our ML framework. Our second objective is to address the varying observable space problem in PDF. Third, we target the inverse problem in GPD by mapping the observables to the CFFs. Finally, we intend to enhance the performance of our frameworks by incorporating the domain knowledge to the ML. We list our research questions as follows:

- **RQ 1**: Can we develop an end-to-end machine learning framework to solve the inverse problems in PDF?
- **RQ 2**: Can we develop a kinematics independent inverse mapper to address the irregular observable space?
- **RQ 3**: Can we extract the GPD Compton form factors?

## 1.3 OUTLINES

The rest of this dissertation is organized as following. Chapter 2 provides background information on classical methods of solving the inverse problems as well the inverse problems in nuclear physics. In particular, the inverse problem of reconstructing the PDF and the GPD's CFFs. Chapter 3 presents the ML related works that have been done to solve the inverse problems. Chapter 4 discusses our method of developing VAIM to solve inverse problem in PDFs. This chapter addresses RQ 1 .Chapter 5 describes our work in extending VAIM to PC-VAIM to address some limitations related to the inverse problem in PDFs. This chapter addresses RQ 2. Chapter 6 describes our work in extracting the Compton form factors using C-VAIM to solve the inverse problem in GPD. This chapter answers RQ 3. Chapter 7 presents our work in integrating physics knowledge to ML models on a forward mapper as a technique to improve the ML prediction capability which can be further applied to our inverse mappers in the future. Chapter 8 summarizes our contributions, conclusions and presents the potential future work. The comprehensive workflow of this dissertation is described in fig. 1

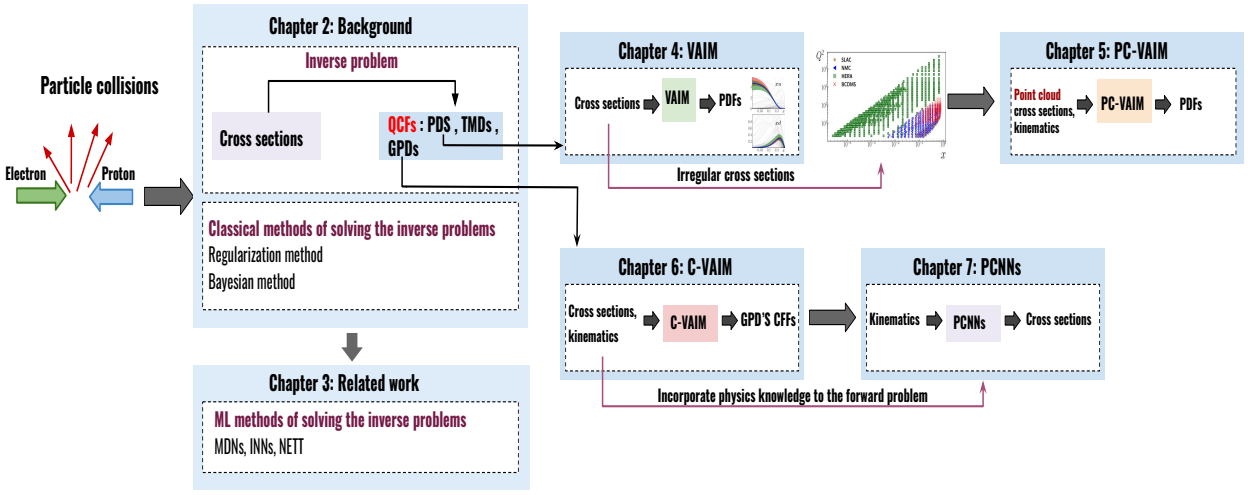



**Fig. 1.** Overall diagram illustrates the research road map. In particle collision experiments, probing the internal structure of the hadrons is an inverse problem, that connects the cross section to the QCFs. Chapter 2 provides a basic understanding of this inverse problem as well the classical methods that are used to solve the inverse problems in general. Since classical methods are not practical in solving ill posed inverse problems, chapter 3 provides a brief overview of the ML methods that are used in the literature. Chapter 4 presents our work in solving the inverse problem of mapping the cross sections to the PDFs parameters. Because the experimental cross section are observed on irregular kinematics bins, chapter 5 extends VAIM to point cloud based VAIM to address the varying observable space. In chapter 6, we extend VAIM to the conditional VAIM to extract the GPDs Compton form factors. Chapter 7 studies the effectiveness of integrating physics constraints to ML starting with the forward problem from the kinematics to the cross sections to be applied in the future to our inverse mappers that are developed in chapters 4, 5 and 6.

# CHAPTER 2

# BACKGROUND

This chapter provides a brief overview of the basic concepts related to this dissertation. First, we briefly discuss some of the inverse problem characteristics. Second, we introduce some of the classical methods of solving the inverse problems. Finally, we describe the inverse problems in nuclear physics, in particular, the inverse problem of PDF and the inverse problem of CFFs.

## 2.1 THE INVERSE PROBLEM CHARACTERISTICS

Solving the inverse problems is a difficult task because of the ill-posedness characteristics. A well-posed inverse problem according to Hadamard rules must satisfy three conditions. First, a solution always exists for all observables. Second, the relationship between inputs and outputs is continuous. In other words, the solution is stable with respect to small perturbations, meaning that the solution changes continuously with respect to the observable. Third, the solution is unique for each observable. Any inverse problem that does not satisfy these criteria is termed as ill-posed inverse problem and thus should be treated differently. Among the ill posed inverse problem characteristics, ambiguity forms a great challenge in our application– nuclear physics. Ambiguity refers to the nonuniqueness of the solution where several parameters can correspond to a unique single observable.

In many scientific applications, machine learning models are successfully utilized for the forward process, and easily can map the observables to the parameters. However, solving the ambiguous inverse problems is non-trivial because a standard neural network will approximate the average of the multiple solutions and lead to poor performance as the average of several solutions is not necessarily a valid solution [44].

## 2.2 THE CLASSICAL METHODS TO INVERSE PROBLEMS

Traditional research of the inverse problems has dedicated on incorporating conditions that insures solutions to such ill-posed problems are existed. In the next sections, we discuss the concept of two classical methods of solving the inverse problems: regularization and Bayesian methods.

### 2.2.1 Regularization Method

Regularization is one of the most common classical techniques in solving inverse problems.

Incorporating a regularization term allows to introduce a prior knowledge to the inverse problem which makes the approximation of the ill posed problems achievable. A well established regularization method which has been widely analyzed, is the so-called Tikhonov regularization method [45, 46, 47]. Tikhonov regularization has been one of the popular techniques for solving inverse problem. Assume that for a known matrix $A$ and vector $k$, we need to calculate a vector $u$ such that :

$$Au = k \tag{1}$$

The standard method to solve this problem is using ordinary least squares linear regression. But, if there is no $u$ satisfies the equation or more than one $u$ exists that satisfies the equation —that is the solution is not unique, the problem then is identified to be ill-posed. In such situations, applying ordinary least squares estimation leads to an overdetermined, or more likely an underdetermined system of equations. Moreover, instead of allowing for a model to be used as a prior for $u$, ordinary least squares implicitly revokes each element of the reconstructed version of $u$ that is in the null-space of $A$. Ordinary least squares aims to minimize the sum of squared residuals, that can be compactly written as:

$$\|Au - k\|_2^2, \tag{2}$$

where $\|.\|_2$ is the Euclidean norm or L2 norm.
In order to obtain a specific solution with desirable properties, a regularization term can be incorporated in this minimization:

$$\|Au - k\|_2^2 + \|\Gamma u\|_2^2 \tag{3}$$

Oftentimes, Tikhonov matrix $\Gamma$ is chosen as a multiple of the identity matrix $\Gamma = \alpha I$ giving the solutions with small norm high preference, this is also known as L2 regularization. The regularization parameter $\alpha$ is usually replaced by $\lambda$ with the property $\lambda \geq 0$.
Thereby, an explicit matrix solution $\hat{u}$, is given by:

$$\hat{u} = (A^\top A + \Gamma^\top \Gamma)^{-1} A^\top k \tag{4}$$

The value of the parameter $\lambda$ can be selected using some techniques such as cross validation [48], L-curve [49] and restricted maximum likelihood [50]. Based on different principles, more strategies for the regularization parameter selection have been discussed in the literature [51].

### 2.2.2 Bayesian Method

Traditional research on inverse problems heavily focuses on Bayesian methodology by incorporating prior knowledge which guarantee a more robust approximation of the solutions. The first publications in inverse problems adopt the Bayesian framework for the solution of inverse problems that arise in geophysical applications [52, 53]. Additional work by Tarantola and Valette in [54] also emphasizes on the use of Bayesian approach as a systematic method of solving inverse problems in scientific application. The Bayesian method is further enhanced and developed theoretically and empirically with scientific applications, especially in medical imaging [55].

Bayesian methodology tackles the inverse problems by modeling the unknown parameters as a random variable and describing it by it's distribution. Bayes theorem facilitates the incorporation of the information obtained during the measurement with the prior information. Introducing a suitable prior is significant for the solution of an inverse problem in the presence of few or noisy data, with the role of the prior is to provide a baseline description for the unknown parameters. By denoting a random variable as $X$ representing the unknown $x$, and $B$ to be a probabilistic model of the observed quantity. The forward model can be summarized as

$$B = f(X) + E, \tag{5}$$

where $E$ represents the Gaussian noise in the observation. Giving that $E$ and $X$ are independent, the probability density of $B$, conditional on $X$, is obtained by shifting the density $\pi_E$ around $f(x)$, leading to the likelihood density

$$\pi_{X|B}(x \mid b) = \frac{\pi_{B|X}(b \mid x)\pi_X(x)}{\pi_B(b)}, b = b_{observed} \tag{6}$$

Given that the denominator is the marginal density of the data given by

$$\pi_B(b) = \int \pi_{B|X}(b \mid x)\pi_X(x)dx$$

The complete solution of the inverse problem in the Bayesian framework is represented in the posterior density (6). However, it only provides an initial point for extracting information of the unknown parameter $x$. The selection of informative priors is a key element in the Bayesian framework. Several methodology of designing the prior are suggested in the literature such as smoothness priors [56, 57], structural priors [58], sparsity-promoting priors [59], and sample-based priors [60, 61].

At present, Bayesian methods have a great impact on the current approaches of solving the inverse problems. An example is that, the selection of the regularization penalty is often indicated

as "utilizing prior information". As well new creative methods and computationally efficient can be developed by incorporating prior information to the data driven methods.

## 2.3 THE INVERSE PROBLEMS IN NUCLEAR PHYSICS

The protons and neutrons, which are known as nucleons or hadrons, are the basic building blocks of all atomic nuclei. Nucleons are made of quarks and gluons (or partons), forming all the visible objects in the universe, including the oceans, the planets, the stars, and also the humans. The nucleons are not static and has complex internal structure. With the advanced technology, we are now entering a new era of probing the inner dynamics and structure of nucleons and nuclei at the sub-femtometer distance, which is a metric unit of length equal to one quadrillionth of a meter. This leads to a new emerging science field known as nuclear femtography [62].

Nuclear femtography is defined as a field of nuclear physics that studies the process of mapping out the QCFs which define the internal structure of nucleons and nuclei, as well as hadronization in high-energy reactions, in terms of the quark and gluon degrees of freedom of QCD. Unlike any existing system, the QCD partons cannot be detected experimentally because of the confinement, that avoids direct measurement of the QCFs. Thankfully, specific observables can be factorized in terms of convolutions between short-distance physics calculable in perturbative QCD and long-distance physics which is nonperturbative and converted in the formulation of QCFs [63]. However, to extract QCFs, we encounter an inverse problem that is caused by the mathematical relation between the experimental observables and the QCFs caused by the inability to have closed form solutions for the QCFs. Thus, several methods have been used to infer the QCFs solutions such as Bayesian inference-based methods [64, 30, 65], Monte Carlo based techniques for Bayesian inference [30, 65], and Machine learning approaches [40, 41, 42].

Solving the inverse problem of QCFs and exploring the inner structure of nucleons is significant to understand the fundamental matter of almost everything. This would provide crucial advances in the understanding of QCD. For example, understanding the binding forces that keep the nucleon together, will allow us to understand the stability of our existence and answer fundamental questions of our universe. This dissertation is mainly focused on the following two inverse problems in nuclear physics:

1. The inverse problem of Parton Distribution Function (PDF):
   Parton distributions function describes the binding of the quarks/ gluons within the proton. The parton density function $f_i(x, Q^2)$ is the probability of detecting in the proton a parton of flavour $i$ (quarks or gluon) carrying a fraction $x$ of the proton momentum with $Q$ representing the hard interaction energy scale. The cross sections is obtained through convoluting the parton level cross section with the PDFs. Understanding proton PDFs is essentially obtained from the Deep Inelastic Scattering (DIS) HERA, and TEVATRON data [66]. More details

about the inverse problems of the PDFs as well as our solutions are discussed in chapters 4 and 5.

2. The inverse problem of the GPD Compton form factors (CFFs):
   Generalized parton distribution has gained a significant attention in the theory and experimental physics communities in the recent years. GPD offer a rich concept of the internal structure of the nucleon. GPD can be measured through deeply virtual scattering processes, in particular, DVCS process has been identified as the cleanest probe of GPD. The extraction of the GPD from the DVCS is not direct but rather the integrals of GPD over the longitudinal momentum transfer of the quarks $x$, the Compton form factors. This indicates that the GPD information is hidden in the Compton form factors values because CFFs integrates the GPD over $x$. Thus, probing the GPD from the CFF represents an ill-posed inverse problem with possibly multiple solutions [67]. In this dissertation, we will carry out the inverse problem of extracting the leading twist CFFs from polarization observables, which is the easiest path to access the GPD. Chapter 6 illustrates our methods and results on several experimental DVCS scenarios.

# CHAPTER 3

# RELATED WORK

Recent advances of deep learning and data-driven methods have enabled new efforts in fitting DNN models to large data sets to solve inverse problems [68]. In many inverse problem applications, data-driven methods based on deep learning have demonstrated superior performance compared to traditional numerical solution methods, applied to various aspects of solving inverse problems. For example, Adler and Öktem [15] extracted prior knowledge using a convolutional network as a gradient-like approach to iteratively optimize the solutions.

This chapter reviews some of the machine learning efforts to solve the inverse problems. In particular, we discuss Mixture Density Network (MDN) [44] in section. 3.1, Invertible Neural Network (INN) [69] in section. 3.2 and network Tikhonov (NETT) [14] in section. 3.3.

## 3.1 MIXTURE DENSITY NETWORKS (MDN)

In machine learning literature, MDN is often used to solve inverse problems. The goal of the MDN is to construct a conditional probability distribution of the parameters, given the observable inputs, which addresses the one-to-many mapping issue in inverse problems. MDN combines a feed-forward NN with a mixture density model to approximate the conditional probability distribution, which often exhibits multi-modal properties. Assuming that a mixture of $m$ Gaussian components can sufficiently approximate the conditional probability distribution, the NN in MDN learns the means and variances of the $m$ Gaussian components as well as their mixing coefficients by optimizing the maximum likelihood. Once the MDN model is trained, the conditional probability function of the parameters with respect to the observables is then constructed.

The overall architecture of MDN is illustrated in fig. 2, consisting of two main components — a neural network and a mixture model. The neural network part can follow any standard feed forward structure that takes $x$ as input and converts it into a set of learned features. The mixture model has the flexibility to model a probability density function (pdf) $p(x)$ as a mixture of $m\ pdfs$ indexed by $i\ p_i(x)$, with weights $\prod = \{\pi, \pi_1, \pi_2, ....., \pi_{m-1}\}$ as following:

$$p(y \mid x) = \sum_{i=1}^{m} \alpha_i(x)\phi_i(y \mid x) \tag{7}$$

where $m$ is the number of the components in the mixture, $\alpha_i(x)$ represent the mixing coefficients that can be prior probabilities of the target vector $y$ having been generated from the $i^{th}$ mixture

components. The choice of the kernel function is restricted to the Gaussian kernel as any probability density function can be approximated, provided the mixing coefficients and the Gaussian parameters are appropriately selected. Then, the Gaussian kernel can be written in the form of:

$$\phi_i(y \mid x) = \frac{1}{(2\pi)^{m/2}\sigma_i x^m} \, exp\left\{ -\frac{\| \, y - \mu_i(x) \, \|^2}{2\sigma_i(x)^2} \right\} \tag{8}$$

The function $\phi_i(y \mid x)$ is the conditional density of the target $y$ for the $i^{th}$ kernel, where $\mu_i(x)$ represents the center of the $i^{th}$ kernel with component $m$.

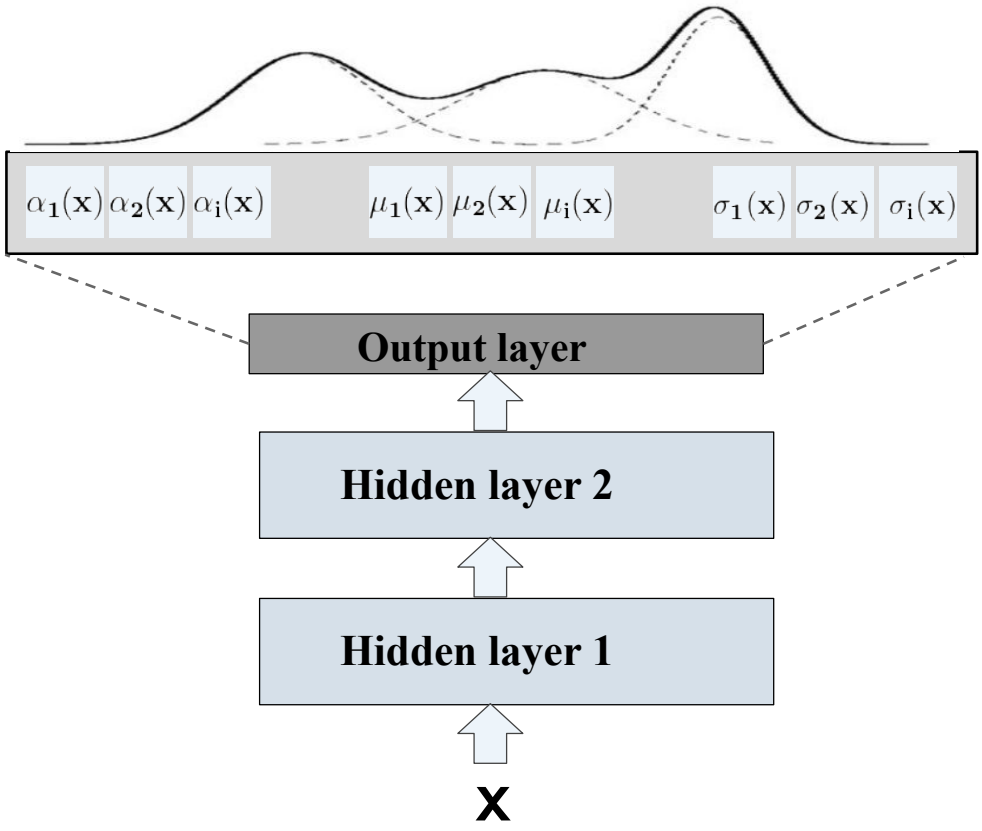



**Fig. 2.** Mixture Density Network architecture: the output of a neural network parametrizes a Gaussian mixture model.

Training the MDN model with sufficient kernel function and a neural network with appropriate

hidden units, MDN will be capable to estimate as closely as possible any conditional density $p(y \mid x)$. For MDN network training, the Negative Log Likelihood is used as a loss function, which is equivalent to the Maximum Likelihood Estimation.

To implement the MDN network, there are a few restrictions should be considered. First, the mixing coefficient $\alpha_i(x)$ should be less than one and sum to unity. This can be accomplished by passing the output of the mixing coefficient $\alpha_i(x)$ through a softmax function [70, 71]. Second, the variance $\sigma_i$ should be positive and, as suggested by Bishop [44], it is more convenient to be represented as the exponential function of the network output $\sigma_i = exp(h_i^{\sigma})$. Finally, the mean $\mu$ represents the location parameters and this should be represented directly by the raw network outputs of the mean neuron $\mu_{im} = h_{im}^{\mu}$.

### 3.2 INVERTIBLE NEURAL NETWORKS (INN)

INNs have been proposed to solve inverse problems by learning the forward mapping. Since the INN is invertible, the backward mapping is learned implicitly at the same time. A latent layer is also incorporated in INN to capture the lost information in forward mapping, which is sampled later to construct the posterior distribution. The main component of the INNs architecture is the reversible block containing of two affine coupling layers which is developed by Dinh and Sohl-Dicksteinl. [72] and known as Real NVP. The block's input vector $b$ is divided into two parts $b1$ and $b2$ that are transformed through an affine functions $s_i$ and $t_i$ $(i \in \{1,2\})$, using element-wise multiplication $(\odot)$ and addition:

$$v_1 = b_1 \odot exp(s_2(b_2)) + t_2(b2), \qquad v_2 = b_2 \odot exp(s_1(b_1)) + t_1(b_1) \tag{9}$$

Where the output $v$ is the concatenation of $[v_1, v_2]$. Then, we can recover $(b_1, b_2)$ using $(v_1, v_2)$ to calculate the inverse of the affine coupling layer.

$$b_2 = (v_2 - t_1(v_1)) \odot exp(-s_1(v_1)), \qquad b_1 = (v_1 - t_2(b_2)) \odot exp(-s_2(b_2)) \tag{10}$$

The functions $s_i$ and $t_i$ can be any arbitrary functions of $v_1$, and $b_2$ and do not have to be invertible. In INNs implementation, fully connected layers with leaky ReLU activations is used to represent $s_i$ and $t_i$.

INN resolves the ambiguity issue associated with the backward mapping $y \to x$ by incorporating latent variable $z$ that capture the lost information in the forward process and not contained in $y$. As illustrated in fig. 3, INN learns to map hidden parameter values $x$ with unique pairs $[y,z]$ of observable and latent variables. Forward training optimizes the mapping $[y,z] = f(x)$ and implicitly learns its inverse $x = f^{-1}(y,z) = g(y,z)$. Furthermore, it is crucial that latent variable $z$ follows

a well known distribution such as uniform or Gaussian distribution. Therefore, INN represents the desired posterior $p(x \mid y)$ by a deterministic function $x = g(y,z)$ that transforms the well determined distribution $p(z)$ to $x$, conditioned on $y$. Applying this architecture, we can now obtain the inverse $y \rightarrow x$ by running the backward mappers at inference time and repeatedly sampling $\mathbf{z}$.

INNs optimize the loss simultaneously on both the inputs and outputs. In each iteration, we perform forward and backward training in an alternating way, combining gradients from both directions before performing a parameter update. In the forward training, we penalize the output $y = f(x)$ with loss $\mathscr{L}_y(y_i, f_y(x_i))$ that can be selected from any supervised loss functions based on the problem such as mean squared error for regression or cross-entropy for classification. The unsupervised loss for the latent variable penalizes the mismatch between the predicted distribution of the invertable network outputs $q(y = f_y(x), z = f_z(x)) = p(x)/|\mathbb{J}_{yz}|$ and the product of latent $p(z)$ and the marginal distributions of the predicted outcomes $p(y = s(x)) = p(x)/|\mathbb{J}_s|$ as $\mathscr{L}_z(q(y,z), p(y)p(z))$. During the training, the unsupervised loss function $\mathscr{L}$ impose two conditions: the generated $z$ distribution follows the normal distribution $p\ (z)$. Second, $y$ and $z$ must be independent.

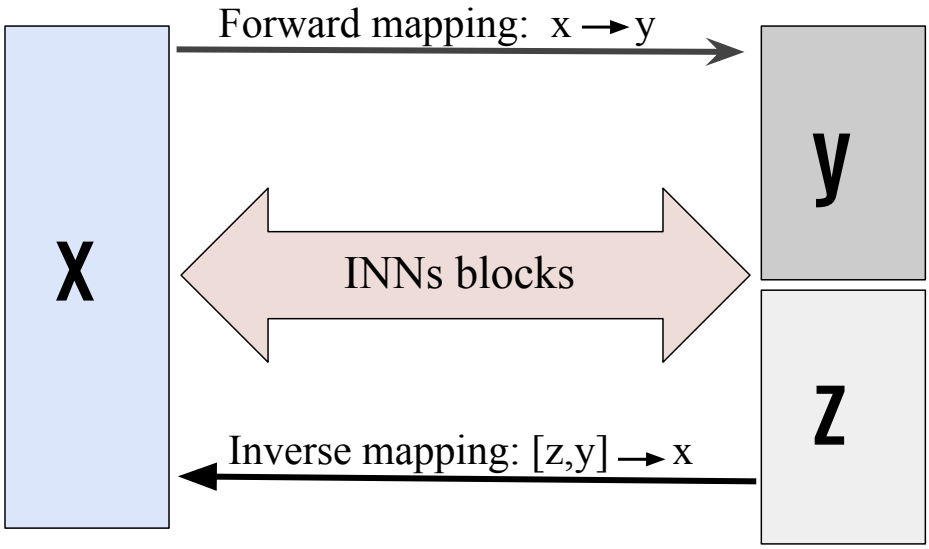



**Fig. 3.** Supervised loss only for the well-defined forward process $x \rightarrow y$. The generated $x$ follow the prior $p(x)$ by an unsupervised loss (USL). The latent variables $z$ are forced to follow a Gaussian distribution, also by an unsupervised loss.

The unsupervised loss $\mathscr{L}$ is implemented by Maximum Mean Discrepancy (MMD) which is a kernel-based approach that measures the distance between two probability distributions through sampling [73]. The MMD term is defined as

$$\begin{aligned} \mathrm{MMD}^2(\boldsymbol{p},\widetilde{\boldsymbol{p}}) = & \mathbb{E}_{\boldsymbol{p}_a,\boldsymbol{p}_{a'}\sim P_{\boldsymbol{p}}}[k(\boldsymbol{p}_a,\boldsymbol{p}_{a'})] \\ & + \mathbb{E}_{\boldsymbol{p}_b,\boldsymbol{p}_{b'}\sim P_{\widetilde{\boldsymbol{p}}}}[k(\boldsymbol{p}_b,\boldsymbol{p}_{b'})] \\ & - 2\,\mathbb{E}_{\boldsymbol{p}_a\sim P_{\boldsymbol{p}},\boldsymbol{p}_b\sim P_{\widetilde{\boldsymbol{p}}}}[k(\boldsymbol{p}_a,\boldsymbol{p}_b)], \end{aligned} \tag{11}$$

where $k(\boldsymbol{p}_a,\boldsymbol{p}_b)$ is a positive definite kernel function. The kernel can be a Gaussian kernel such that $k(\boldsymbol{p}_a,\boldsymbol{p}_b) = \exp\left[-(\boldsymbol{p}_a-\boldsymbol{p}_b)^2/2\sigma^2\right]$.

### 3.3 NETWORK TIKHONOV (NETT)

Several proposed methods to solve the inverse problems utilize a priori information about the unknowns parameters to be reconstructed. This information can be that the unknowns are derived from a specific distribution or has small rate of regularization. NETT framework adopts the use of regularizations to approach the inverse problems by incorporating deep neural networks with a Tikhonov regularization. For an inverse problem

$$y_\delta = F(x) + \xi_\delta \tag{12}$$

Where $x \in$ D from data, $F : D \subseteq X \to Y$ is a possibly non-linear operator. The variable $\xi_\delta \in Y$ represents the unknown data error that is assumed to follow the estimate $\| \xi_\delta \| \leq \delta$ for some noise level $\delta \geq 0$. NETT method defines the regularizer as following:

$$\forall x \in X : R(x) = R(\mathbb{V},x) := \psi(\phi(\mathbb{V},x)) \tag{13}$$

$\psi : \mathbb{X}_L \to [0,\infty]$ represents a scaler function, $\phi(\mathbb{V},.) : X \to \mathbb{X}_\mathbb{L}$ is a neural network of depth $L$ where $\mathbb{V} \in \nu$, for a vector space $V$ includes some free parameters that can be tuned based on the training samples.

Based on the previous equations, NETT framework is defined as following

$$\tau_{\alpha;y_\delta}(x) := D(F(x),y_\delta) + \alpha R(\mathbb{V},x) \longrightarrow min_{x\in D}$$

With $D : Y \times Y \to [0,\infty]$ represents a similarity measurement in the data space that can be Kullback-Leibler divergence or any other valid similarity metrics. The network regularizer $R(\mathbb{V},x)$ can be a user-specified, or a trained network, where the parameters are tuned on appropriate training data.

NETT framework is developed based on using a trained neural networks to learn regularizer given by a CNN that is trained using an encoder–decoder scheme. NETT is applied to a sparse data problem in photoacoustic tomography (PAT), demonstrates that NETT with the trained regularizer yields good results. Other extensions of NETT are proposed such as augmented NETT [74] and the synthesis NETT [75] which are also built utilizing the trained regularizer.

# CHAPTER 4

# VARIATIONAL AUTOENCODER INVERSE MAPPER (VAIM)

In this chapter we present our work developing VAIM framework that is published in [76] to solve the inverse problems. VAIM is built as an autoencoder style consisting of encoder and decoder neural networks approximating the forward and backward mapping, respectively. A latent layer is incorporated into VAIM to learn the posterior parameter distributions with respect to given observables. First, we describe VAIM architecture, modeling the inverse problems using our VAIM and the training procedures. Second, we show the capability of VAIM in solving multiple analytical inverse problems with different solution patterns as well as the inverse problem in QCD. Finally, we compare VAIM with the baseline models that are used to solve the inverse problems. This chapter address RQ1 discussed in Sec. 1.2

## 4.1 INTRODUCTION

We develop VAIM, as an end-to-end machine learning architecture to solve the inverse problems. We focus on addressing non-unique solutions while maintaining suitable behavior of other computational challenges, such as precision, stability, and solution robustness. Our approach here using VAIM adopts the architecture of an autoencoder [16], consisting of a forward mapper from parameter space to observable space and a backward mapper from observable space back to parameter space.

The latent space is a key element in VAIM architecture that captures the conditional posterior distribution of the inverse process. During training, the latent layer learns an estimator, which infers the posterior parameter distribution with respect to the given observables in an unsupervised manner, without explicit knowledge of the inverse problem. Given observables as input, sampling the variables in the latent layers by variational inference [17] then reveals the corresponding parameter distribution.

We apply VAIM to several toy inverse problems with different solution characteristics and analyze the patterns captured in the latent variables. Since we are particularly interested in targeting an important nuclear physics application of reconstructing the internal structure of nucleons (protons and neutrons) within QCD, we also apply VAIM to construct the inverse function mapping quantum correlation functions, in this case PDF to experimental observables.

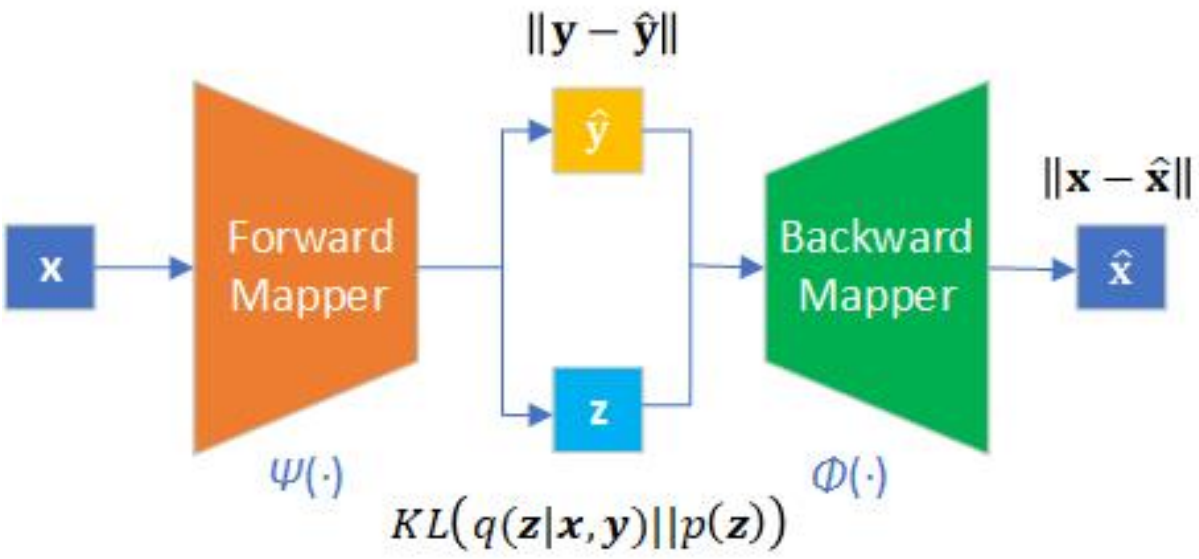



**Fig. 4.** Schematic representation of the VAIM architecture [76].

### 4.2 VAIM ARCHITECTURE

The architecture adopted for VAIM involves an autoencoder composed of two NNs, a forward mapper $\Psi(\cdot)$ and a backward mapper $\Phi(\cdot)$, as shown in Fig. 4. An inverse problem in VAIM is modeled as a statistical inverse problem, with the parameters considered to be random variables. More specifically, when an observable is given, instead of generating deterministic estimated parameter values, VAIM approximates a probability distribution of parameters. A latent layer is incorporated between the forward and backward mappers, whose purpose is to learn the patterns of the parameter distributions. The latent layer serves as part of the output from the forward mapper and part of the input for the backward mapper. Unlike the hidden layers of the forward or backward mappers, the latent layer is constraint to follow certain well-known distributions, such as Gaussian or uniform distributions. As solutions to the statistical inverse problem, for a given observable, sampling the latent distribution yields the corresponding parameter distributions.

The ill-posedness of inverse problems arises from the fact that certain information is lost during forward mapping, so that an observable may correspond to multiple parameters in backward mapping. Concatenating the latent layer to the output of the forward mapper is designed to capture such lost information during forward mapping, which is then used to reconstruct the parameters in the backward mapper. This step replaces the explicit inclusion of domain knowledge present in some of the inverse mapping methods discussed in Sec. 2.

Denoting the parameters as $\mathbf{x} \in \mathbb{R}^m$, the observables as $\mathbf{y} \in \mathbb{R}^n$, and the latent variables in the latent layer as $\mathbf{z} \in \mathbb{R}^k$, the forward mapper learns the one-to-many mapping from parameters to observables such that

$$\Psi(\mathbf{x}; w_f) = [\hat{\mathbf{y}}, \mathbf{z}],$$

where $w_f$ represents the weights of the forward mapper NN, and $\hat{\mathbf{y}}$ is an approximation to the observables $\mathbf{y}$. The backward mapper is trained to reconstruct $\mathbf{x}$ according to $\mathbf{y}$ and $\mathbf{z}$ such that

$$\Phi(\mathbf{y}, \mathbf{z}; w_g) = \hat{\mathbf{x}},$$

where $w_g$ are the weights of the backward mapper NN, and $\hat{\mathbf{x}}$ approximates parameters $\mathbf{x}$.

### 4.2.1 VAIM Model

From a statistical point of view, our forward mapper learns to approximate the posterior distribution $p(\mathbf{z}\,|\,\mathbf{x},\mathbf{y})$ with $q(\mathbf{z}\,|\,\mathbf{x},\mathbf{y})$, while the backward mapper learns to approximate the likelihood distribution $p(\mathbf{x},\mathbf{y}\,|\,\mathbf{z})$. Our goal is to approximate the true posterior distribution $p(\mathbf{z}\,|\,\mathbf{x},\mathbf{y})$ by learning a tractable distribution $q(\mathbf{z}\,|\,\mathbf{x},\mathbf{y})$ via variational inference. We therefore ensure that

$$q(\mathbf{z}\,|\,\mathbf{x},\mathbf{y}) \sim p(\mathbf{z}\,|\,\mathbf{x},\mathbf{y}),$$

by minimizing the Kullback-Leibler (KL) divergence between the two distributions. VAIM is then modeled as

$$\min \Big[ \mathrm{KL}\big(q(\mathbf{z}\,|\,\mathbf{x},\mathbf{y}) \;||\; p(\mathbf{z}\,|\,\mathbf{x},\mathbf{y})\big) \Big],$$

where $\mathrm{KL}\big(q(\mathbf{z}\,|\,\mathbf{x},\mathbf{y}) \;||\; p(\mathbf{z}\,|\,\mathbf{x},\mathbf{y})\big)$ denotes the KL-divergence between $q$ and $p$. The theoretical underpinning of a variational autoencoder (VAE) [77] indicates that the above optimization problem is equivalent to

$$\min_{w_f, w_g} \Big[ \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 + \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \mathrm{KL}\big(q(\mathbf{z}\,|\,\mathbf{x},\mathbf{y}) \;||\; p(\mathbf{z})\big) \Big],$$

where $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ is the forward mapping error, $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$ is the likelihood error of reconstructing the parameters, and $p(\mathbf{z})$ is the true prior distribution that we choose for convenience. Typically, $p(\mathbf{z})$ is selected to be a tractable, easy-to-generate distribution, such as a normal distribution or a uniform distribution. The KL-divergence term ensures that

$$q(\mathbf{z}\,|\,\mathbf{x},\mathbf{y}) \sim p(\mathbf{z}).$$

The latent variables $\mathbf{z}$ in the latent layer are the key to the VAIM model. The prior distribution $p(\mathbf{z})$ is tractable to approximate the true posterior distribution $p(\mathbf{z}\,|\,\mathbf{x},\mathbf{y})$. During training, the latent layer attempts to learn the lost information in forward mapping and converts the ill-posed

inverse problem into a well-posed regression problem in the backward mapper with augmented dimensionality from the latent space.

The dimensionality $k$ of $\mathbf{z}$ is an important hyperparameter in VAIM. Provided $k$ is not less than the actual dimensionality of the information lost in forward mapping, the latent layer can theoretically reconstruct the complete parameter distribution for the given observables. Nevertheless, as shown in the results presented in Sec. 4.3 below, a larger $k$ allows VAIM to represent the lost information more easily, and thus to better reconstruct the parameter distribution.

### 4.2.2 Training and Production

The training of VAIM is a hybrid of supervised learning and unsupervised learning. The training of the forward mapping function from parameters to observables and the backward mapping function from observables to parameters is supervised from the parameter-observable pairs in the training data set. The derivation of the posterior distribution $p(\mathbf{z}|\mathbf{x},\mathbf{y})$, on the other hand, is unsupervised. Therefore, VAIM can be used as a generalized end-to-end framework to be applied to different inverse problems and extract their unique patterns in forward and backward mappings, without relying on specific prior knowledge of the inverse problems.

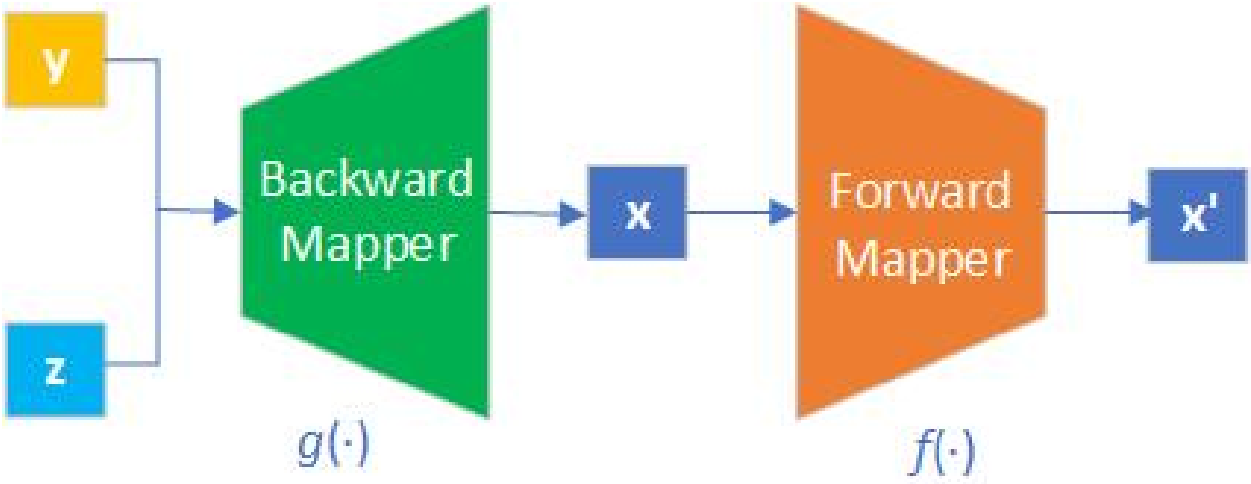



**Fig. 5.** Illustration of VAIM in operation. The backward mapper is used as a generative model to produce parameter samples by sampling $\mathbf{z}$ with respect to the given observables $\mathbf{y}$, while the forward mapper acts as a filter to eliminate noisy predictions [76].

Once VAIM is trained, the backward mapper is applied to inverse problems by predicting observables that are coupled with samples of the latent variables, as shown in Fig. 5. Here, the backward mapper is used as a generative model, where sampling the latent variables can lead to the likelihood distribution of the parameters. The forward mapper, handling the well-posed parameter to observable mapping, yields a high accuracy, and can be used as a filter on top of the backward mapper to remove noisy predictions.

### 4.3 RESULTS

In illustrating the applicability of VAIM for solving practical problems, we first test it on toy inverse problems with different solution patterns, before applying it to a simplified version of an essential application in fundamental nuclear physics, namely, QCD analysis of quantum correlation functions. We further investigate the patterns in the latent space that is learned during training. Here, both forward and backward mappers in VAIM employ a feedforward NN with 512 hidden nodes, which can be replaced by more sophisticated architectures such as DNNs or CNNs in more complex applications.

#### 4.3.1 Toy Inverse Problems

To demonstrate the effectiveness of VAIM we consider the following toy inverse problems with analytical solutions:

(1) $f(x) = x^2, x \in [-3,3]$,

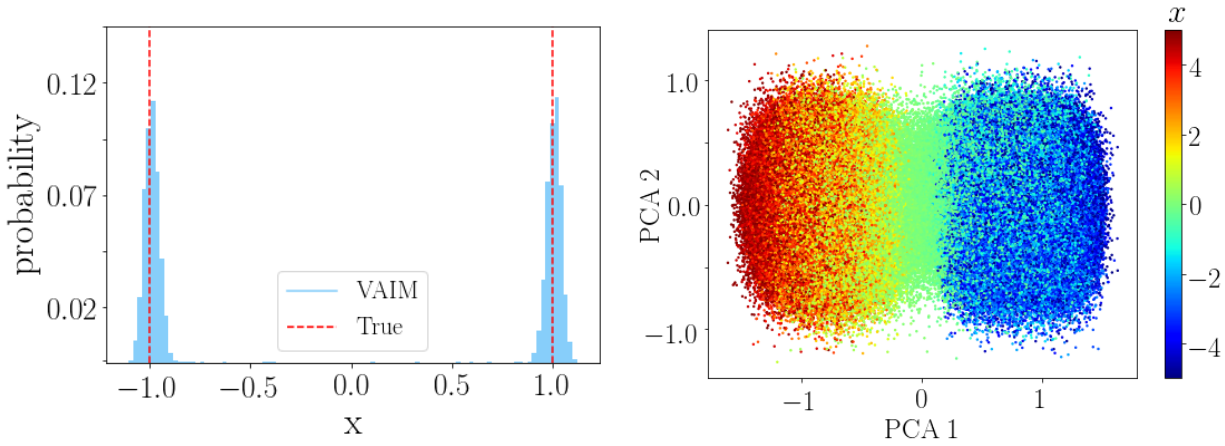
(2) $f(x) = \sin x, x \in [-2\pi, 2\pi]$,

(3) $f(\boldsymbol{x}) = \boldsymbol{x}^2 = x_0^2 + x_1^2$, with $\boldsymbol{x} = (x_0, x_1)$, $x_{0,1} \in [-2,2]$.

These toy problems yield very different solution patterns. For a given $f(x)$ value, the inverse problem of finding the corresponding $x$ has 2 distinct solutions in $f(x) = x^2$, except for $f(x) = 0$, for case (1); 2 to 5 distinct solutions in $f(x) = \sin(x)$ for case (2); and an infinite number of solutions in $f(\boldsymbol{x}) = x_0^2 + x_1^2$ for case (3).

The same VAIM architecture, except for the shapes of the input and output layers, which must match the dimensionality of each toy problem, is applied to these three problems. Both forward and backward mappers employ a simple fully-connected network with 512 hidden nodes. The size of hidden layer $k$ is set to 10 and the prior distributions of the latent variables are chosen to be uniformly distributed. Some $10^6$ training samples are generated to train the VAIM for each problem, and noise is added to the training data by allowing the data to vary within 5% of the true

values. The training terminates when the reconstruction error is less than $10^{-4}$. After training, we analyze the latent space and its association with $x$ by projecting the latent variables onto the first two principle components after principle component analysis (PCA).



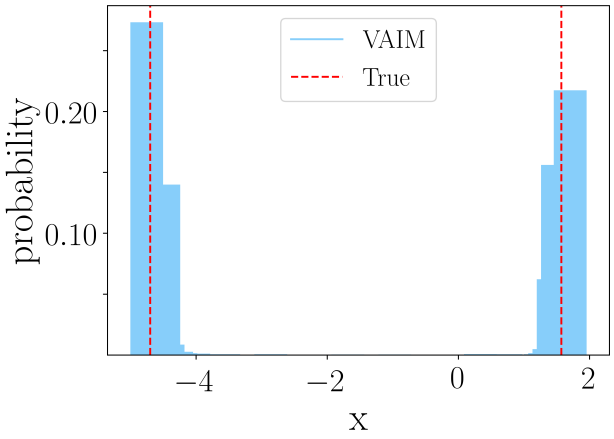

**A** . Predicted distribution of $x$ when $f(x) = 1$.

**B** . Distribution of $x$ on latent space projected on the first two principle components.

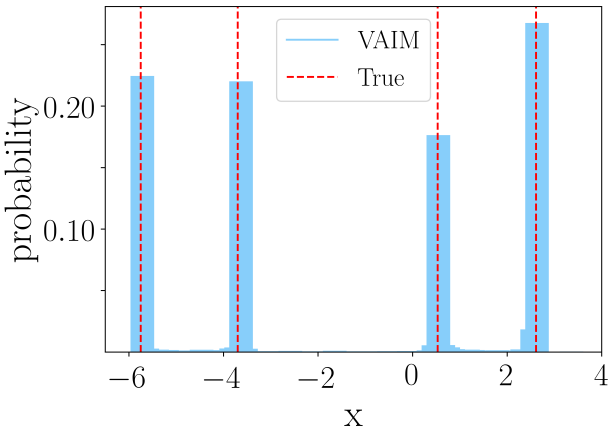
**Fig. 6.** VAIM predicted solution distributions and latent variable of $f(x) = x^2$ [76].

Figure 6A shows the VAIM predicted $x$ distribution $\big(f(x) = 1\big)$ in the inverse problem of $f(x) = x^2$, where the two peaks correspond to the two solutions of $x = 1$ and $x = -1$. The information lost in forward mapping in $f(x) = x^2$ is the sign of $x$. The distribution of $x$ samples on the latent space is plotted in fig. 6B for the first two PCA component vectors (PCA1 and PCA2). One can see that the lost sign information is well-captured in the latent variables, where the positive and negative samples form two separated clusters, connected by the samples with near 0 values.

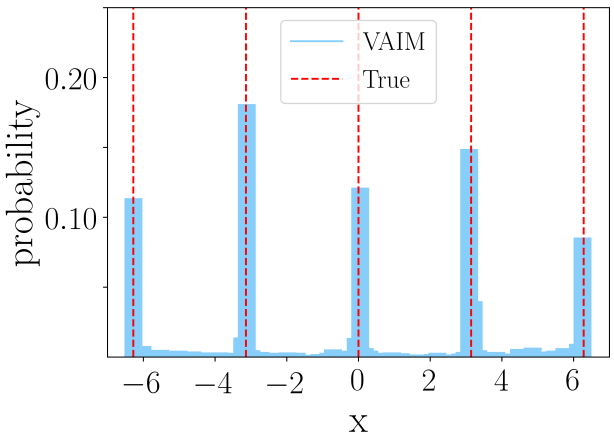
The VAIM predicted $x$ distributions with respect to $f(x) = 1$, $-\frac{1}{2}$, and 0 in the inverse toy problem $f(x) = \sin x$ are displayed in Fig. 7A, 7B, and 7C, respectively, exhibiting 2, 4, and 5 peaks which match the corresponding distinct analytic solutions. The distribution of $x$ on latent space projected on the first two principle components are shown in Fig. 7D. The period information lost in the forward mapping is well preserved in the latent space, where the $x$ samples in $[-2\pi, -\frac{3}{2}\pi]$, $[-\frac{3}{2}\pi, -\frac{1}{2}\pi]$, $[-\frac{1}{2}\pi, \frac{1}{2}\pi]$, $[\frac{1}{2}\pi, \frac{3}{2}\pi]$, and $[\frac{3}{2}\pi, 2\pi]$ are located in the stratified segments in the latent space. The preserved period information is later used to correctly construct the solution distribu-



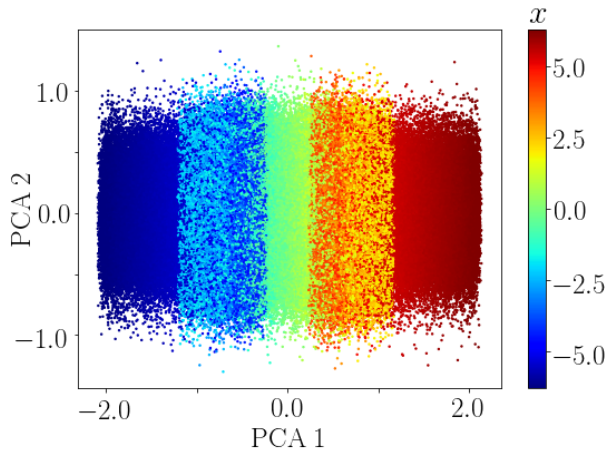

**A** . Predicted distributions of $x$ when $f(x) = 1$.




**B** . Predicted distributions of $x$ for $f(x) = -\frac{1}{2}$.




**C** . Predicted distributions of $x$ when $f(x) = 0$.



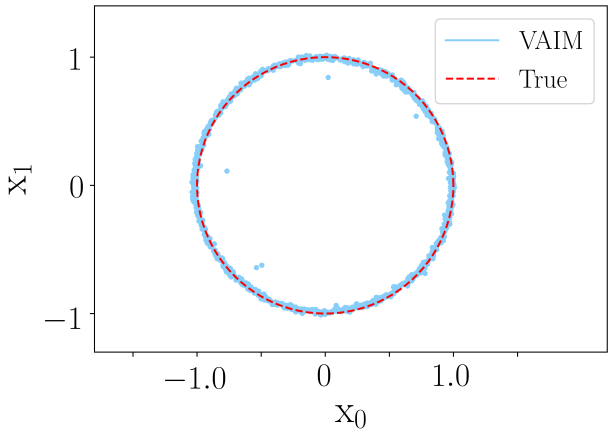

**D** . Distribution of $x$ on latent space projected on the first two principle components.

**Fig. 7.** VAIM predicted solution distributions (A)–(C) and latent variable projection (D) of $f(x) = \sin x$ [76].

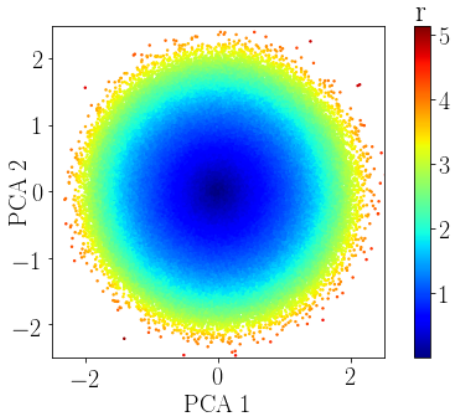
tions in Fig. 7A, 7B, and 7C.

Unlike the above two toy problems where the solution distribution is discrete with respect to a given $f(x)$ value, the solution distribution is continuous in the inverse problem of $f(\boldsymbol{x}) = x_0^2 + x_1^2$, so that there are infinite solutions available. As shown in Fig. 8A, the VAIM predicted solutions correctly lie on the unit circle when $f(\boldsymbol{x}) = 1$. The distributions of radius $r = \sqrt{x_0^2 + x_1^2}$ and polar angle $\theta = \arctan(x_1/x_0)$ on the latent variables projected on the first two principle components are shown in Fig. 8B and 8C, respectively. Interestingly, in the latent space, the same radii are located on the concentric circles, while the polar angles are nearly circularly uniformly distributed, indicating that the latent layer in VAIM has correctly captured the solution patterns in polar coordinates in this inverse problem.

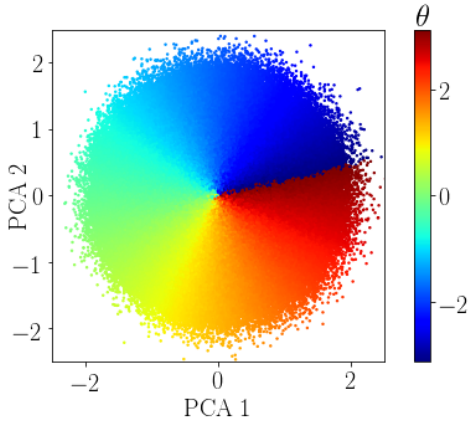
We stress that the same VAIM architecture is applied to the three toy problems with very different solution patterns. The training is supervised on matching $f(x)$ in forward mapping and reconstructing $x$ in backward mapping, as well as enforcing the latent variables to be uniformly distributed. No other information, such as the number of solutions, is fed to VAIM. Nevertheless, our analysis demonstrates that VAIM is able to capture the posterior parameter distribution, and thus successfully predicts the solution distributions.




**A** . Predicted distributions of $x$ when $f(x) = 1$.




**B** . Distribution of $r = \sqrt{x_0^2 + x_1^2}$ on latent space projected on the first two principle components.




**C** . Distribution of $\theta = \arctan(x_1/x_0)$ on latent space projected on the first two principle components.

**Fig. 8.** VAIM predicted solution distributions (A) and latent variable projections (B)–(C) of $f(\boldsymbol{x}) = x_0^2 + x_1^2$ [76].

### 4.3.2 VAIM for a Nuclear Physics Inverse Problem

#### Synthetic DIS problem

In this section we apply VAIM to a simplified version of inverse mappings in nuclear physics, which we use as a proxy for real QCD applications, such as extracting quantum correlation functions from experimental deep-inelastic lepton-nucleon scattering (DIS) data [78]. For this application, we construct the inverse function mapping the quantum correlation functions — in this case, parton distribution functions (PDFs) — to observables. In particular, we consider DIS from protons and neutrons, as two observables sensitive to different combinations of "up" and "down" quark PDFs, weighted by the squares of their charges, $+2/3$ and $-1/3$ respectively,

$$\sigma_p(x,Q) = \tfrac{4}{9}u(x,Q) + \tfrac{1}{9}d(x,Q),$$
$$\sigma_n(x,Q) = \tfrac{4}{9}d(x,Q) + \tfrac{1}{9}u(x,Q).$$

Here, $\sigma_p$ and $\sigma_n$ mimic the structure functions measured in DIS from protons and neutrons, respectively. For the functional form of the quark PDFs $q(x,Q)$, with $q = u$ or $d$, which are functions of the proton's momentum fraction $x$ carried by the quark and the momentum scale, $Q$, at which the PDFs are probed, we use the shape suggested in Ref. [79],

$$q(x,Q) = N_q(Q)\, x^{\alpha_q(Q)}(1-x)^{\beta_q(Q)} \times (1 + \gamma_q(Q)\sqrt{x} + \delta_q(Q)\, x),$$

where the shape parameters $p_q = \{N_q, \alpha_q, \beta_q, \gamma_q, \delta_q\}$ are parametrized as

$$p_q(Q) = p_q^{(0)} + p_q^{(1)} \log\left(\frac{\log(Q/\Lambda)}{\log(Q_0/\Lambda)}\right).$$

For the scale parameters we use $Q_0 = 2$ GeV and $\Lambda = 0.2$ GeV. Since in practice the $Q$ dependence is typically known, we set all the $p_q^{(1)}$ parameters that control the $Q$ dependence to the value 0.1, leaving 10 shape parameters $p_q^{(0)}$ to describe the $u$ and $d$ quark PDFs.

The VAIM model is trained using the data between the parameters and the cross section observables, and the prediction results are validated by a set of *control* samples which are not contained

in the training set. The parameter distributions generated in four control samples are displayed in Fig. 9. One finds that VAIM predicts a few solution clusters for each control sample, and the parameter vector of the control sample falling right into one of these clusters. This indicates that VAIM can precisely predict the parameter solution distributions.

The distributions of the "up" and "down" quarks on the latent space projected onto the first two principle components are shown in Fig. 10a and 10b, respectively, representing certain latent physics properties captured by VAIM. Note that similar VAIM architecture and training procedure are used here as the ones applied to the three toy problems in Sec. 4.3.1 above. The reconstructed PDFs are plotted in Fig. 11 and compared with a control sample. The results show that the control PDF is located within 1 standard deviation of the reconstructed PDFs from the predicted parameters.

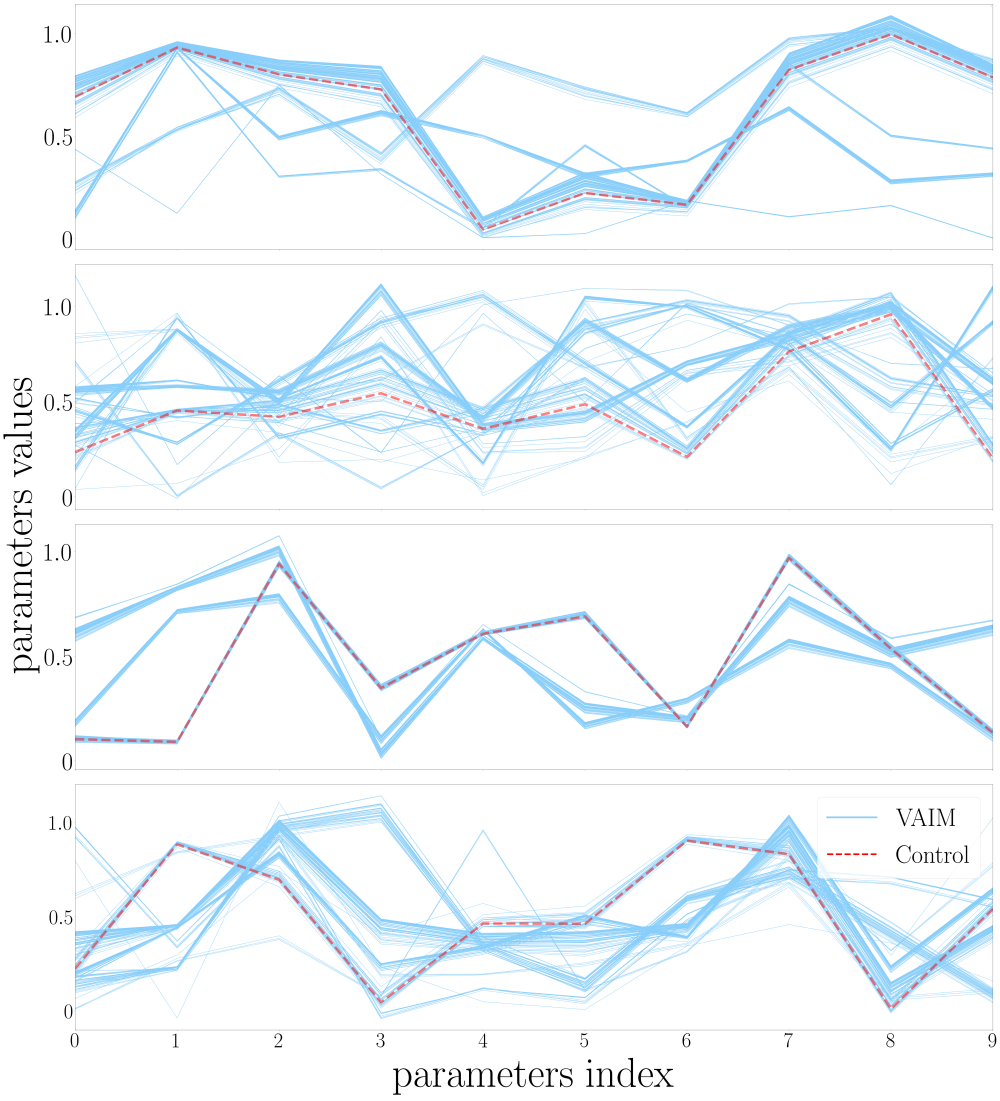



**Fig. 9.** PDF parameter distributions generated by VAIM in four control samples [76].

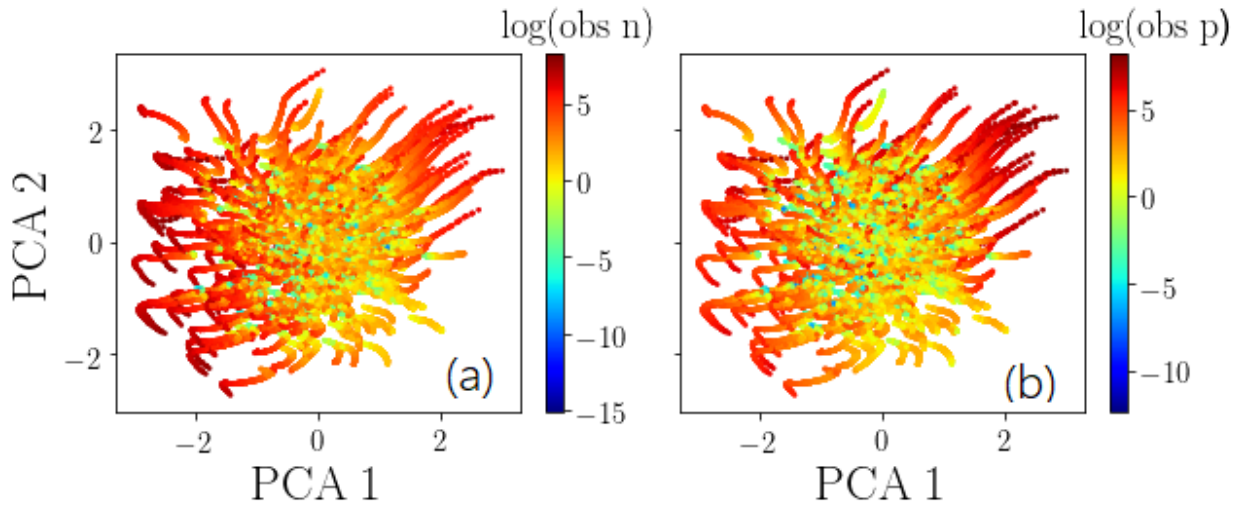



**Fig. 10.** Distributions of (a) "up" quarks and (b) "down" quarks on the latent space projected on the first two principle components, PCA1 and PCA2.

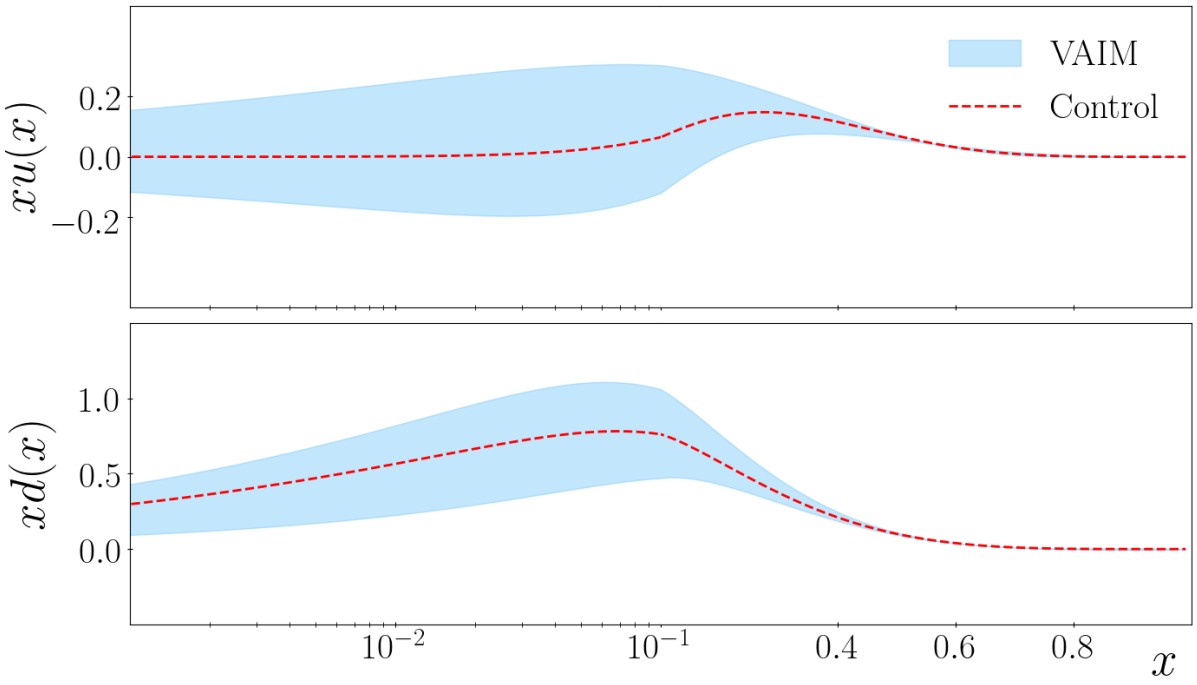



**Fig. 11.** PDF space for the "up" and "down" quarks using a control sample. The control PDFs are within 1 standard deviation of the reconstructed PDFs from the predicted parameters [76].

**Experimental DIS problem**

After validating the VAIM on the synthetic DIS data, we proceed next to carry out a similar analysis with real PDFs obtained from real global QCD analysis of experimental data on inclusive DIS cross sections. The cross section space is defined by the kinematics of the world's inclusive DIS datasets from Stanford Linear Accelerator Center (SLAC) [80], Bologna-CERN-Dubna-Munich-Saclay (BCDMS) [81], New Muon Collaboration (NMC) [82, 83] and Hadron-Electron Ring Accelerator (HERA) [84], with a total of 2,680 points. The kinematics of the data sets are illustrated in fig. 12, and the data have been restricted to be within the region $W^2 > 10$ GeV$^2$ and $Q^2 > m_c^2$, where $m_c = 1.3$ GeV is the charm quark mass. We generate training samples using the Jefferson Lab Angular Momentum (JAM) Collaboration codes [78] that map the PDFs into DIS observables at next-to-leading order accuracy in perturbative QCD.

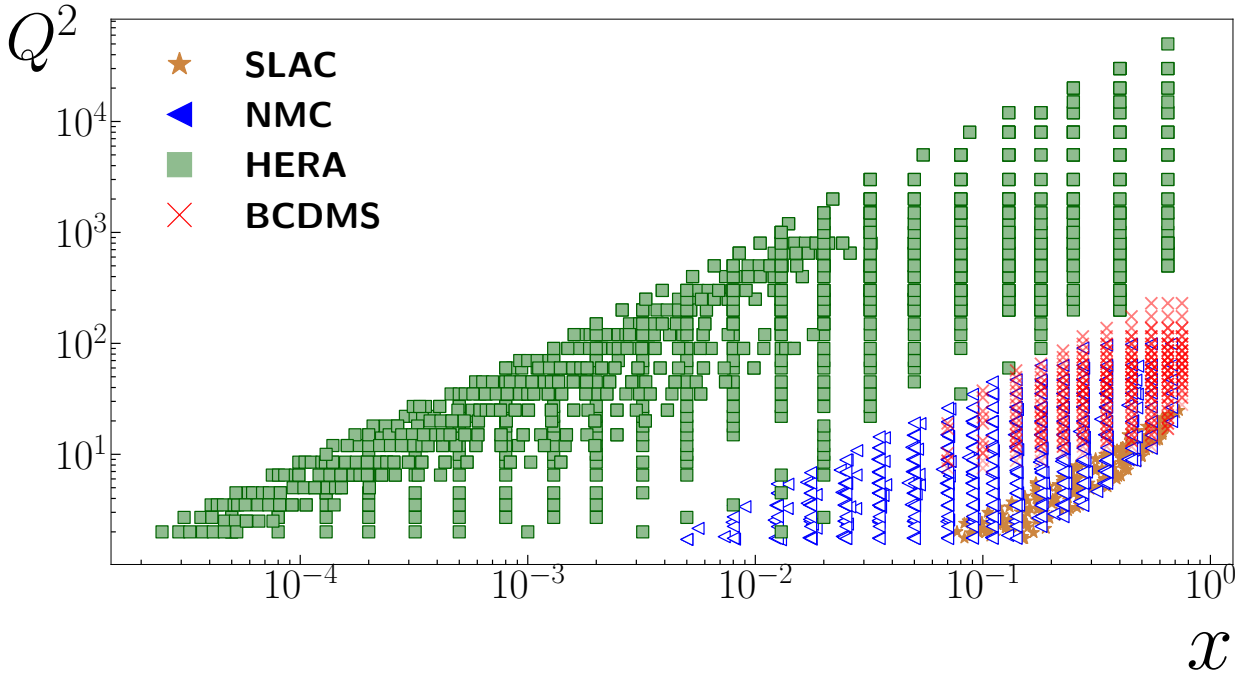



**Fig. 12.** Kinematics bins of SLAC, NMC, HERA, and BCDMS experiments

We train our VAIM architecture illustrated in fig. 4 with the forward network mapping the parameters space to the cross sections space and latent variable $z$ and the backward network mapping

the cross sections and $z$ to the parameters. We validate VAIM on real experimental example that is not included in the training process. The reconstructed PDFs are plotted in Fig. 13 and compared to the maximum likelihood sample which represents the control sample. The black curves demonstrates the training samples that we utilize to train VAIM. The hyperbox of the training samples is selected to be as wide as possible to make sure that the training data is not too informative. Our result shows that the reconstructed PDFs from the predicted parameters are located within one standard deviation of the control PDF.

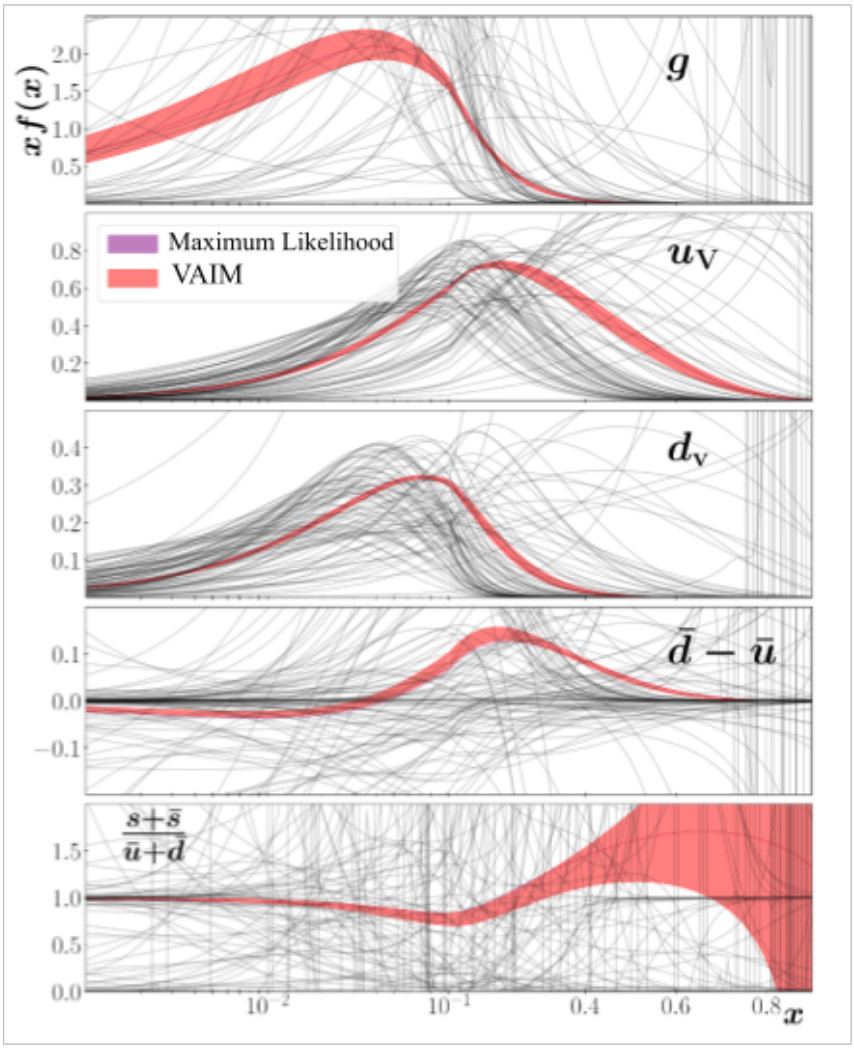



**Fig. 13.** VAIM predictions for PDFs (gluon $g$, valence $u_v$ and $d_v$, sea asymmetry $\bar{d}-\bar{u}$ and strange to nonstrange ratio $(s+\bar{s})/(\bar{u}+\bar{d})$) with black curves represent the training samples

To further test the quality of our predicted parameters, we reconstruct the corresponding cross sections using JAM calculations. We plot the reconstructed cross sections with the experimental cross section, demonstrating that if the VAIM was accurate we would have the predictions lying on the diagonal line as shown in fig 14. We also calculate the $\chi^2$ values of the reconstructed cross section of the predicted parameters and we obtain $\chi^2$= 1.32 which is comparable to the baseline model, JAM with $\chi^2$= 1.25. We further explore the $\chi^2$ using JAM code for each data set in SLAC, NMC, HERA, and BCDMS experiments as illustrated in table. 1. We found that VAIM achieves best accuracy for the parameters associated to SLAC dataset with $\chi^2$= 0.7.

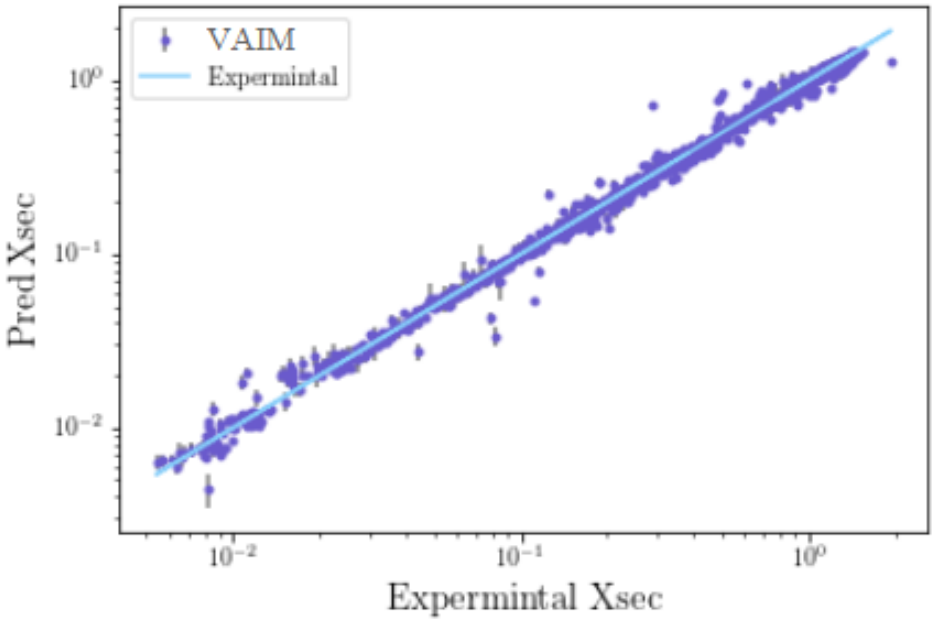



**Fig. 14.** The computed cross sections of the VAIM predicted parameters using theory model compare to the experimental cross section.

### 4.4 COMPARISON WITH INN

Invertible neural networks (INNs) have been proposed [69] to solve inverse problems by learning the forward mapping. Since the INN is invertible, the backward mapping is learned implicitly

**Table 1.** Chi-squared on the experimental example

| Experiment | number of points (npts) | $\chi^2/npts$ |
|---|---|---|
| SLAC | 218 | 1.5 |
| BCDMS | 348 | 1.2 |
| NMC1 | 273 | 1.7 |
| HERA II NC e+ (1) | 402 | 1.6 |
| HERA II NC e+ (2) | 75 | 1.1 |
| HERA II NC e+ (3) | 259 | 1.0 |
| HERA II NC e+ (4) | 209 | 1.1 |
| HERA II NC e- | 159 | 1.7 |
| HERA II CC e+ | 39 | 1.3 |
| HERA II CC e- | 42 | 1.3 |
| SLAC | 228 | 0.7 |
| BCDMS | 254 | 1.2 |
| NMC2 | 174 | 1.0 |

at the same time. A latent layer is also incorporated in INN to capture the lost information in forward mapping, which is sampled later to construct the posterior distribution. VAIM and INN utilize similar approaches based on the embedded latent layer to obtain the posterior distribution over parameter space, with key differences, as we discuss in the following.

In the loss function, VAIM uses KL-divergence, while INN adopts maximum mean discrepancy (MMD) [85] to measure the similarity of samples from the true prior distribution and those from the approximated posterior distribution. As pointed out by Ramdas *et al.* [86], the power of both KL-divergence and MMD degrade as the dimension increases. However, MMD degrades polynomially at best, while KL-divergence degrades at a constant rate.

To investigate this further, we revisit the toy inverse problem (3), $f(\boldsymbol{x}) = \sum_i x_i^2$, but expanded to higher dimensions. For a given $f(\boldsymbol{x})$, we use VAIM and INN to predict the parameter vector $\boldsymbol{x'}$ and then calculate $f(\boldsymbol{x'})$. The same number of hidden nodes and the same training stopping criteria are employed in VAIM and INN, and the same dataset is used to train the models. The distributions of $f(\boldsymbol{x'})$ obtained by VAIM and INN when $f(\boldsymbol{x}) = 1$ are compared in Fig. 15. For a 2-dimensional toy problem, Fig. 15 (top) shows that although the solution distribution obtained by INN yields

a bigger variance, both solutions distributions are correctly centered at 1 when $k = 10$. For a higher dimensional (10D) toy problem, the results in fig. 15 (bottom) show that the INN degrades drastically, with the solution distribution no longer centered at 1 even when the dimensionality of the latent layer $k$ is raised to 100. In contrast, VAIM is able to correctly construct the solution distribution centered at 1, although it is slightly wider than that in the 2D toy problem.

It is also important to note that the latent layer size $k$ plays an important role to the performance of VAIM. As $k$ increases, the latent layers in VAIM is able to better represent the information lost in forward mapping and thus leads to better construction of the solution distribution in backward mapping. These results are consistent with the analysis in [86], which suggests that VAIM is posed to be more suitable for handling high dimensional inverse problems, which are common in many scientific applications.

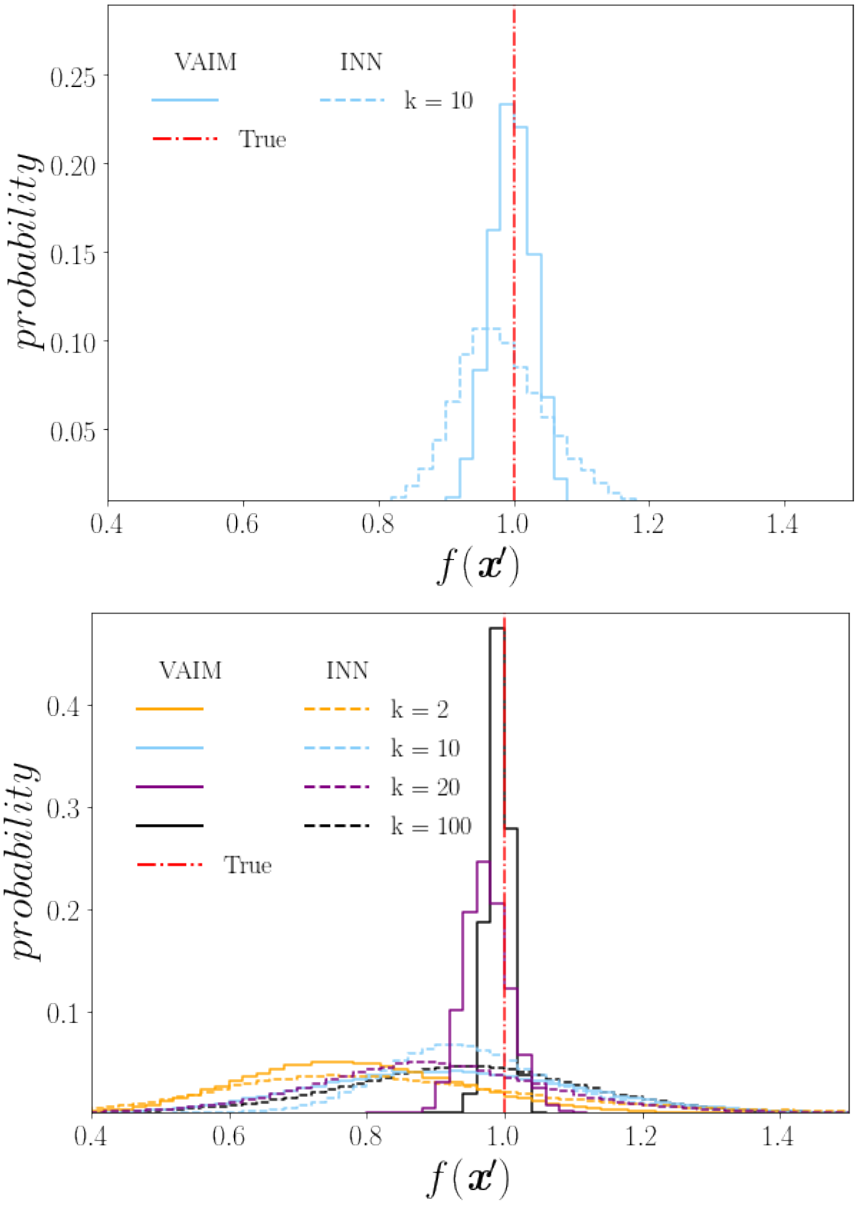



**Fig. 15.** Comparison of the solution distributions of $f(\boldsymbol{x}')$ obtained by VAIM and INN in the toy problem $f(\boldsymbol{x}) = \sum_i x_i^2$, when $f(\boldsymbol{x}) = 1$ is given, for the (Top) 2D and (Bottom) 10D cases.

## 4.5 COMPARISON WITH MDN

In machine learning literature, the mixture density network (MDN) [44] is often used to solve inverse problems. The goal of the MDN is to construct a conditional probability distribution of the parameters, given the observable inputs, which addresses the one-to-many mapping issue in inverse problems. MDN combines a feed-forward NN with a mixture density model to approximate the conditional probability distribution, which often exhibits multi-modal properties. Assuming that a mixture of $C$ Gaussian components can sufficiently approximate the conditional probability distribution, the NN in MDN learns the means and variances of the $C$ Gaussian components as well as their mixing coefficients by optimizing the maximum likelihood. Once the MDN model is trained, the conditional probability function of the parameters with respect to the observables is then constructed.

Compared to MDN, VAIM does not need to rely on the assumption of a Gaussian mixture model. For many inverse problems which are significantly non-Gaussian, the Gaussian mixture assumption can result in poor approximations in MDN. Although the mixing components in MDN do not have to be Gaussian, certain distribution, such as Bernoulli, must be assumed. More importantly, VAIM is not limited by the hyperparameter of $C$ mixing components required by MDN. To better illustrate this, we revisit the toy problem (3), $f(\boldsymbol{x}) = x_0^2 + x_1^2$, where, for a given $f(\boldsymbol{x})$, there is an infinite number of solutions $\boldsymbol{x}$ lying on a circle.

Figure 16 shows the results of MDN on $f(\boldsymbol{x}) = x_0^2 + x_1^2$ when $f(\boldsymbol{x}) = 1$ with respect to $C$ of 2, 4, 10, and 100. The same numbers of hidden nodes as in VAIM are used in the MDN NN, and the same dataset is used for the training. As more mixing components are incorporated, the MDN prediction improves. At the same time, the MDN architecture becomes more complicated, since the parameters of more mixing components need to be determined. In principle, MDN needs an infinite number of Gaussians to describe well this toy problem. In comparison, VAIM does not depend on the Gaussian assumption and does not need to estimate the number of mixing components. Instead, the latent layer in VAIM learns the posterior parameter distribution without an encoded form of the distribution. As a result, VAIM yields significantly better predictions in this toy inverse problem than does MDN, as shown in Fig. 16.

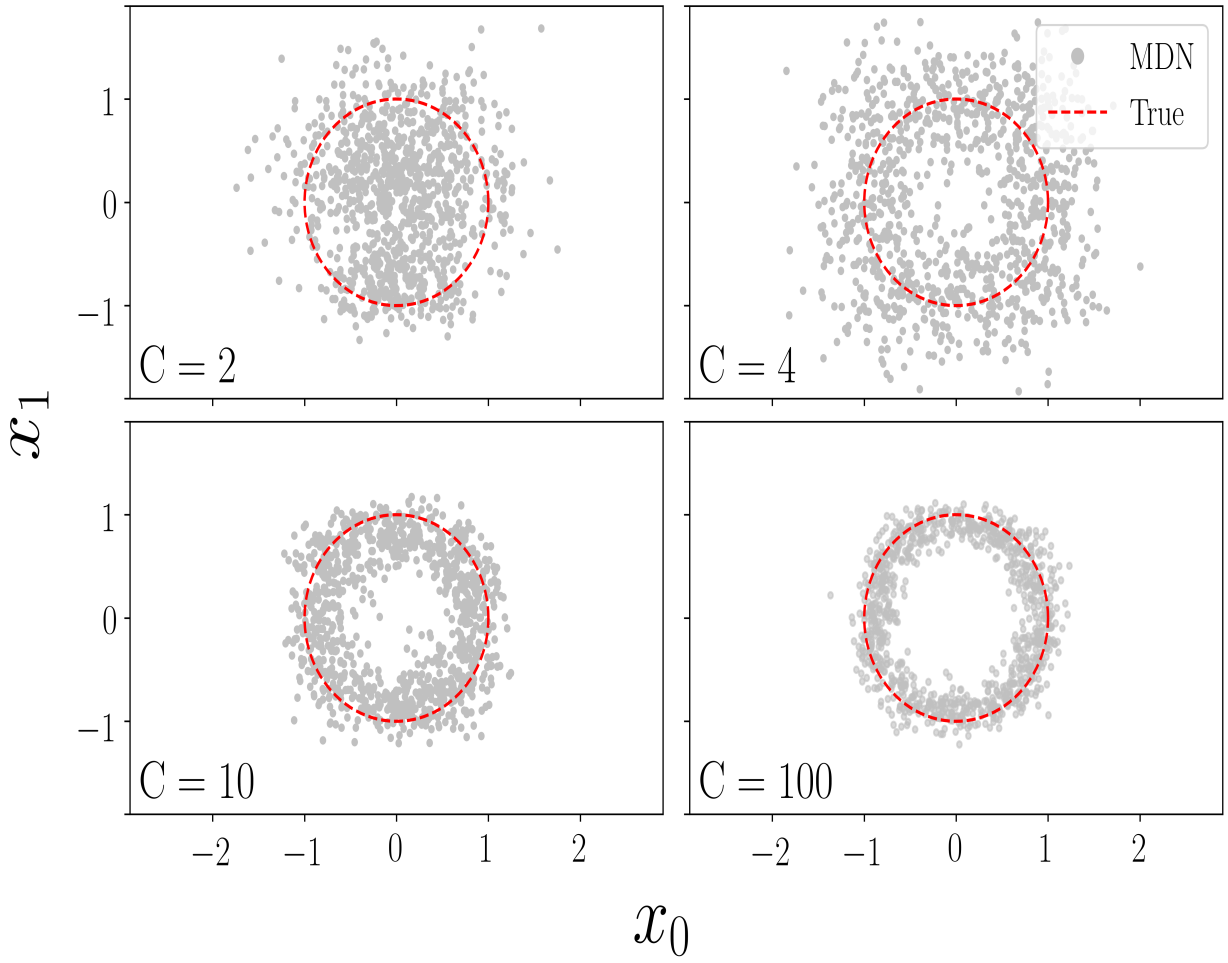



**Fig. 16.** Predicted solution samples with number of Gaussian Components $C = 2$, 4, 10, and 100 mixing components on $f(\boldsymbol{x}) = x_0^2 + x_1^2$ with respect to $f(\boldsymbol{x}) = 1$ using MDN. More Gaussian components result in a better solution distribution approximation in MDN, but MDN requires an infinite number of Gaussians to approximate well the solution distribution. In contrast, VAIM does not rely on the Gaussian assumption and gives a better match with the solution distribution along the unit circle, as shown in Fig. 8A.

## 4.6 CONCLUSION

We present VAIM, a deep learning framework, to solve inverse problems. The design of the VAIM model focuses on learning the solution patterns in an end-to-end manner, making little assumptions. VAIM integrates a forward mapper NN to learn mapping from parameter space to observable space and a backward mapper NN to reconstruct parameters from observables. A latent layer is incorporated between the forward and backward mappers, playing a key role in the approximation of the posterior parameter distribution with respect to observables via variational inference. The latent variables in the latent layer attempt to retain information otherwise lost in forward mapping. The learned latent space augments the observable space and converts the ill-posed inverse problem into a well-posed regression problem.

Application of the VAIM architecture to the three toy problems with different solution patterns shows that it is able to learn the patterns without relying on prior knowledge of the problems. VAIM is further applied to mapping quantum correlation functions, and in particular PDFs, to observables in nuclear physics, demonstrating its capability in a more complicated environment, where there is significant domain interest.

VAIM demonstrated improvement over other deep learning frameworks designed to construct the posterior parameter distribution. Compared to MDN, VAIM does not rely on the Gaussian mixture assumption, making it suitable for the many inverse problems whose solution distributions are non-Gaussian. Compared to INN, VAIM is better-positioned to handle high-dimensional inverse problems.

In general, VAIM is built on top of the theory of VAE and belongs to a broader category of generative models that include generative adversarial networks (GANs) [87] and normalized flow [88]. An inverse problem solver can therefore be implemented in a conditional GAN setup, and it will be interesting to compare the performance of VAIM and conditional GANs. Furthermore, VAIM is not limited to nuclear physics and has potential to be applied to a wide variety of applications, such as image processing [89] and many science and engineering problems.

# CHAPTER 5

# POINT-CLOUD BASED VARIATIONAL AUTOENCODER (PC-VAIM)

In this chapter we introduce our work implementing the point cloud-based VAIM (PC-VAIM) framework, published in [90], to address an application specific issue related to QCD inverse problem where the experimental data are observed on kinematics bins which are irregular. First, we demonstrate the QCD inverse problem and our method of representing the cross sections together with their corresponding kinematic location as a geometric point cloud. Second, we illustrate the PC-VAIM with forward and backward mapper architectures. Finally, demonstrate the application of PC-VAIM on a toy inverse problem with multiple solutions, as well on QCD inverse problem.

## 5.1 INTRODUCTION

Our work developing VAIM in Chapter. 4 has demonstrated preliminary success in solving the QCD inverse problem, where the parameters of the resulting inverse function for a given array of observables can be accurately predicted. The fundamental idea behind VAIM is to incorporate a variational latent layer to learn the posterior parameter distribution with respect to observables. In the situation when multiple solutions exist for the parameters, VAIM is able to reveal the parameter distribution by sampling the latent space. The VAIM model is designed to have the numerical values of the observables across regular, discretized kinematic bins as shown in Fig. 17 as inputs to the inverse function. The number of entries for the input must be equal to the number of available experimental data points. However, this limits VAIM to only accept experimental data points observed on specific kinematic bins. As a result, the DNNs in VAIM are only learning how to connect predefined ordering of the observables to QCF parameters. In reality, the experimental data are much ill-defined, where the data in different experiments are observed on different kinematic bins. Fig. 12 in chapter. 4 shows the kinematic bins in SLAC, NMC, HERA, and BCDMS. These kinematic bins are located at different positions in the kinematic space and the numbers of bins also vary in different experiments. Therefore, it is important for VAIM to have the flexibility to learn, more generally, how the observables are distributed across kinematics.

In this chapter, we extend the VAIM framework to address the ill-defined, varying observable space problem to allow observable input to be independent of the specific kinematic bin setups. Our solution is to represent the cross section observables together with their kinematic bins as a geometric point in space, where the cross section and the corresponding kinematic bins form their geometric coordinates. Accordingly, the set of all observed data points can be represented as a

high dimensional point cloud, containing a set of geometric points in the cross section-kinematics space. The data points within the point cloud are unstructured and unordered while the number of data points can be different, too. The point cloud representation enables the VAIM framework to accept unstructured observable representation, no longer limited by the discretized, predefined kinematic bins. We adopt the PointNet framework [18], which is designed to be invariant to input permutation, to handle the unordered and unstructured observable data in point cloud representation. The Point Cloud-based VAIM (PC-VAIM) enables the underlying DNN model to be flexible to observables at different kinematics bins. We first use a toy problem to demonstrate the capability of PC-VAIM on recovering the parameters by giving flexible observable inputs. Then, we show the results of applying PC-VAIM to construct the inverse function of mapping QCF to observables in a QCD analysis of nucleon structure.

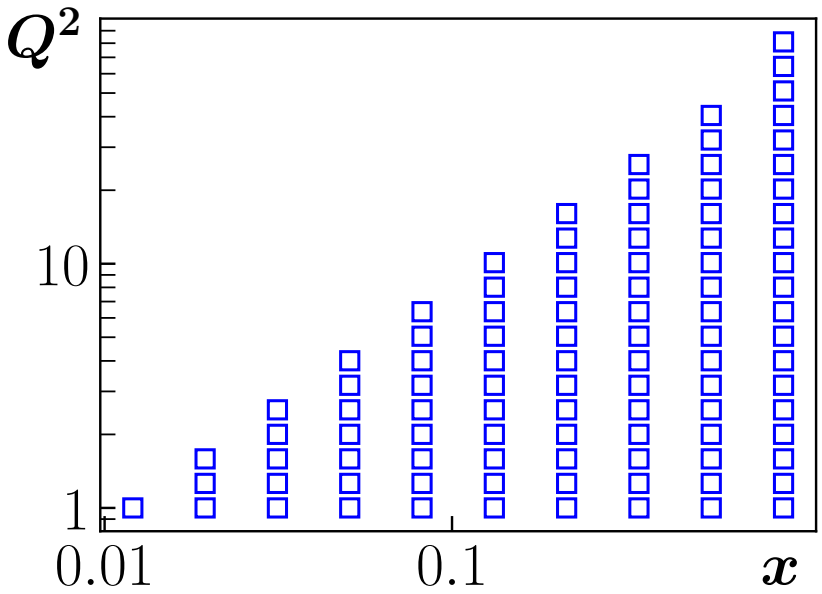



**Fig. 17.** Regular, discretized kinematic bins of $x$ and $Q^2$ where the cross sections are evaluated [90].

## 5.2 METHODS

In this section, we describe the inverse problem occurs in QCD analysis, and our method of representing the cross sections as point cloud structure.

### 5.2.1 The Inverse Problem in QCD Analysis

We perform QCD analysis of extracting quantum correlation functions from experimental deep-inelastic lepton-nucleon scattering (DIS) data. Here, we consider a proxy for real QCD application. Our observables are the cross sections, referring to the probability that two particles, such as protons and neutrons, will collide and react in a certain way. A cross section is calculated at specific kinematic bin, which describes momentum fractions $x$ and the momentum scales $Q^2$ of the particles. In particular, we consider cross sections $\sigma_p$ and $\sigma_n$ sensitive to different combinations of "up" and "down" quark Parton Distribution Functions (PDFs). The inverse problem in QCD analysis here is to construct the inverse function mapping the QCFs – in this case, PDFs – to the cross section observables. The cross sections $\sigma_p$ and $\sigma_n$ mimic the structure functions measured in DIS from protons and neutrons, respectively. They are weighted by the squares of their charges, $+2/3$ for "up" quark and $-1/3$ for "down" quark,

$$\sigma_p(x,Q^2) = \tfrac{4}{9}u(x,Q^2) + \tfrac{1}{9}d(x,Q^2),$$
$$\sigma_n(x,Q^2) = \tfrac{4}{9}d(x,Q^2) + \tfrac{1}{9}u(x,Q^2).$$

The functional forms of the quark PDFs $u(x,Q^2)$ and $d(x,Q^2)$ adopt the following shapes

$$u(x,Q^2) = N_u(Q^2)\,x^{\alpha_u(Q^2)}(1-x)^{\beta_u(Q^2)}$$
$$d(x,Q^2) = N_d(Q^2)\,x^{\alpha_d(Q^2)}(1-x)^{\beta_d(Q^2)},$$

where $\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$ are the shape parameters. Therefore, the goal of the inverse problem is to map the given cross section observables $\sigma_p$ and $\sigma_n$ at the kinematic bins $(x,Q^2)$ to the shape parameters $\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$.

### 5.2.2 Representing Observables as a Point Cloud

VAIM has demonstrated preliminary success in the above inverse problem in QCD analysis [76], having the numerical values of the observable cross sections across all kinematic bins $(x,Q^2)$, as shown in Fig. 17, as the input. The number of entries for the input has to exactly match the number of kinematic bins. Upon training, VAIM can predict accurately the parameters

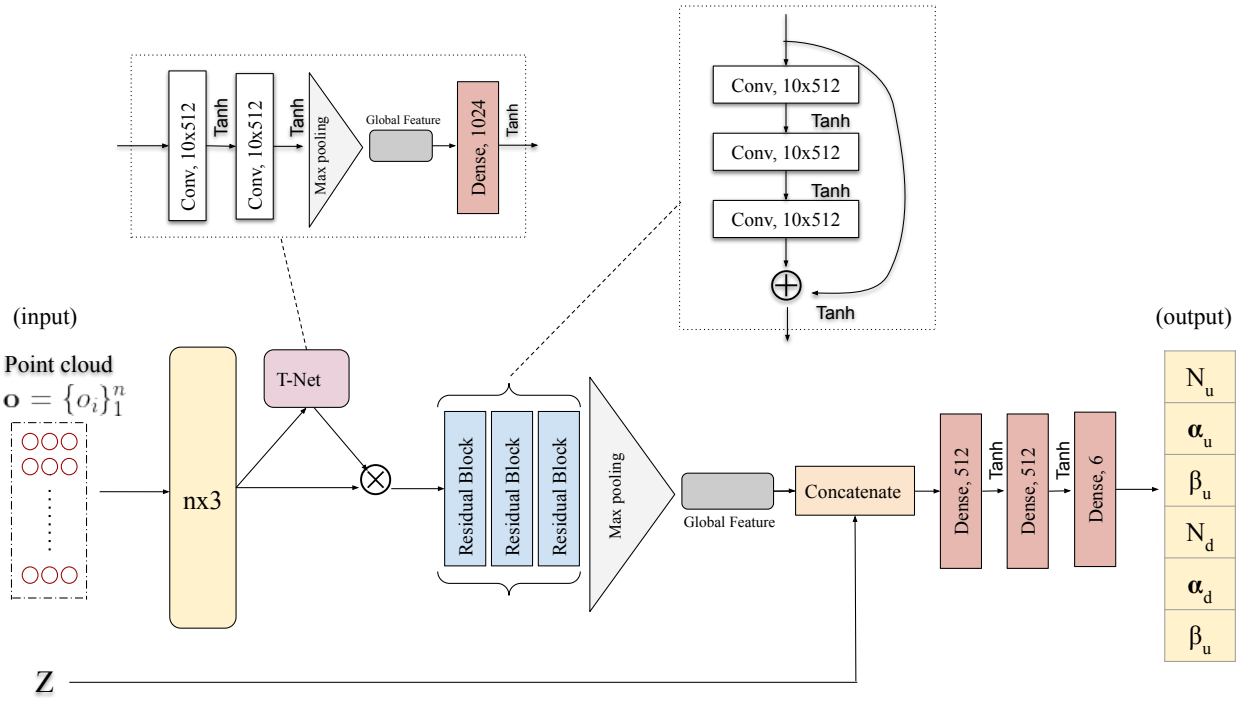



**Fig. 18.** A schematic view of the PC-VAIM backward mapper. A PointNet-based architecture is used to handle the observable input as a point cloud representation [90].

$\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$ for the given observable cross sections, $\sigma_p$ and $\sigma_n$, on the regular kinematic bins. However, this VAIM implementation has limitation on observable inputs from fixed kinematic bins and does not have the flexibility to handle the observables at different kinematics.

To address the ill-defined, varying observable space problem, we represent the cross section observables $(\sigma_p, \sigma_n)$ together with their kinematic location $(x, Q^2)$ as a geometric point in cross section-kinematics space, corresponding to their geometric coordinates $(\sigma_p, \sigma_n, x, Q^2)$. Then, all observables form a set of high dimensional geometric points, which is generally represented as a high dimensional point cloud. The data points within the point cloud are unordered and the number of data points are also flexible. As a result, the observable inputs are no longer limited by the discretized, predefined kinematic bins. We use a permutation invariant neural network based on PointNet to handle the unordered and unstructured observable data in point cloud representation. The fundamental idea behind a permutation invariant neural network is the use of symmetric functions, whose return value with respect to the multiple given arguments is the same regardless of the order of the arguments. Zero padding is used to allow flexible number of observable data points as input. Fig. 18 illustrates the point cloud representation of the cross section observables. The

overall architecture of PC-VAIM is showed in Fig. 19. PC-VAIM consists of two neural networks (NNs), a forward mapper $\Psi(\cdot)$ and a backward mapper $\Phi(\cdot)$. The forward mapper $\Psi(\cdot)$ maps the parameters to the observables represented as a point cloud while the backward mapper $\Phi(\cdot)$ maps the observables back to the parameters. Similar to VAIM, in PC-VAIM, an inverse problem is modeled as a statistical inverse problem, with the parameter space considered to be random variables. In particular, given the point cloud representation of the observables, PC-VAIM attempts to approximates the probability distribution of the shape parameters in the PDFs.

Due to information loss in forward mapping, the inverse problems are often ill-posed, resulting in that an observable is possibly correspond to multiple parameters in backward mapping. To capture the lost information, a latent layer $\mathbf{z}$ is incorporated between the forward and backward mappers, whose goal is to learn the patterns of the parameter distributions. $\mathbf{z}$ is restricted to certain well-known distributions, such as Gaussian or uniform distributions, for sampling convenience. Once PC-VAIM is trained, for a given observable, sampling the latent distribution as input to $\Phi(\cdot)$ yields the corresponding parameter distributions.

With the parameters are indicated as $\mathbf{p} \in \mathbb{R}^m$, the observables point cloud as $\mathbf{o} = \{o_i\}_1^n$, and the latent variables in the latent layer as $\mathbf{z} \in \mathbb{R}^k$, the forward mapper learns the one-to-many mapping from parameters to observables such that

$$\Psi(\mathbf{p}; w_f) = [\hat{\mathbf{o}}, \mathbf{z}],$$

where $w_f$ denotes the weights of the forward mapper NN and $\hat{\mathbf{o}}$ is an approximation to the observables $\mathbf{o}$. The backward mapper is trained to reconstruct $\mathbf{p}$ according to $\mathbf{o}$ and $\mathbf{z}$ such that

$$\Phi(\mathbf{o}, \mathbf{z}; w_b) = \hat{\mathbf{p}},$$

where $w_b$ is the weights of the backward mapper NN and $\hat{\mathbf{p}}$ approximates parameters $\mathbf{p}$.

The forward mapper $\Psi(\cdot)$ attempts to approximate the posterior distribution $p(\mathbf{z} \mid \mathbf{p}, \mathbf{o})$ with a learned distribution $q(\mathbf{z} \mid \mathbf{p}, \mathbf{o})$, while the backward mapper $\Phi(\cdot)$ learns to approximate the likelihood distribution $p(\mathbf{p}, \mathbf{o} \mid \mathbf{z})$. The fundamental idea in PC-VAIM is to approximate the true posterior distribution $p(\mathbf{z} \mid \mathbf{p}, \mathbf{o})$ by learning a tractable distribution $q(\mathbf{z} \mid \mathbf{p}, \mathbf{o})$ via variational inference [17]. Then, PC-VAIM model is designed to ensure that

$$q(\mathbf{z} \mid \mathbf{p}, \mathbf{o}) \sim p(\mathbf{z} \mid \mathbf{p}, \mathbf{o}),$$

by minimizing the Kullback-Leibler (KL) divergence between the two distributions. Consequently, PC-VAIM is modeled as

$$\min \Big[ \mathrm{KL}\big(q(\mathbf{z} \mid \mathbf{p}, \mathbf{o}) \,||\, p(\mathbf{z} \mid \mathbf{p}, \mathbf{o})\big) \Big],$$

where $\mathrm{KL}\big(q(\mathbf{z} \mid \mathbf{p},\mathbf{o}) \,||\, p(\mathbf{z} \mid \mathbf{p},\mathbf{o})\big)$ denotes the KL-divergence between $q(\cdot)$ and $p(\cdot)$. According to the theory of a Variational Autoencoder (VAE) [77], the above optimization problem is equivalent to the following

$$\min_{w_f,w_b} \Big[ L_2(\mathbf{o},\hat{\mathbf{o}}) + L_{Chamfer}(\mathbf{p},\hat{\mathbf{p}}) + \mathrm{KL}\big(q(\mathbf{z} \mid \mathbf{p},\mathbf{o}) \,||\, p(\mathbf{z})\big) \Big].$$

Here, $p(\mathbf{z})$ is the true prior distribution that we choose for convenience. Typically, $p(\mathbf{z})$ is selected to be a tractable, easy-to-generate distribution, such as a normal distribution or a uniform distribution. The KL-divergence term ensures that

$$q(\mathbf{z} \mid \mathbf{p},\mathbf{o}) \sim p(\mathbf{z}).$$

$L_2(\mathbf{p},\hat{\mathbf{p}}) = \|\mathbf{p}-\hat{\mathbf{p}}\|_2^2$ is the likelihood error of reconstructing the parameters. $L_{Chamfer}(\mathbf{o},\hat{\mathbf{o}})$ is the forward mapping error. Here we adopt the Chamfer distance [91] to measure the point cloud reconstruction error as the forward mapping error. More specifically, the Chamfer distance computes the sum of squared distance between each point in one point cloud to its nearest neighbor in another point cloud. Thus, the forward mapping error becomes

$$L_{Chamfer}(\mathbf{o},\hat{\mathbf{o}}) = \sum_{x\in\mathbf{o}} \min_{y\in\hat{\mathbf{o}}} \|x-y\|_2^2 + \sum_{y\in\hat{\mathbf{o}}} \min_{x\in\mathbf{o}} \|x-y\|_2^2.$$

Similar to that of VAIM, the latent variables $\mathbf{z}$ in the latent layer play an important role in the PC-VAIM model. The prior distribution $p(\mathbf{z})$ is tractable to approximate the true posterior distribution $p(\mathbf{z} \mid \mathbf{p},\mathbf{o})$. In PC-VAIM training, the latent layer attempts to learn the information lost during forward mapping and converts the ill-posed inverse problem into a well-posed regression problem in the backward mapper with augmented dimensionality from the latent space. Provided the dimensionality of $\mathbf{z}$ is greater or equal to the actual dimensionality of the information lost in forward mapping, the latent layer in PC-VAIM can theoretically reconstruct the complete parameter distribution for the given observables. On the other hand, the point cloud representation provides PC-VAIM the flexibility to handle observable on varying kinematic bins.

## 5.3 PC-VAIM ARCHITECTURE

Fig. 19 depicts the architecture of PC-VAIM composing of two main components: the forward mapper that is implemented as a fully connected DNN and the backward mapper that is implemented using a PointNet-based convolutional NN.

**Fully connected Forward Mapper $\Psi(\cdot)$:**

The forward mapper $\Psi(\cdot)$ maps the parameters to a observable point cloud and the latent variable. In our PC-VAIM implementation, the forward mapper NN is composed of an input layer

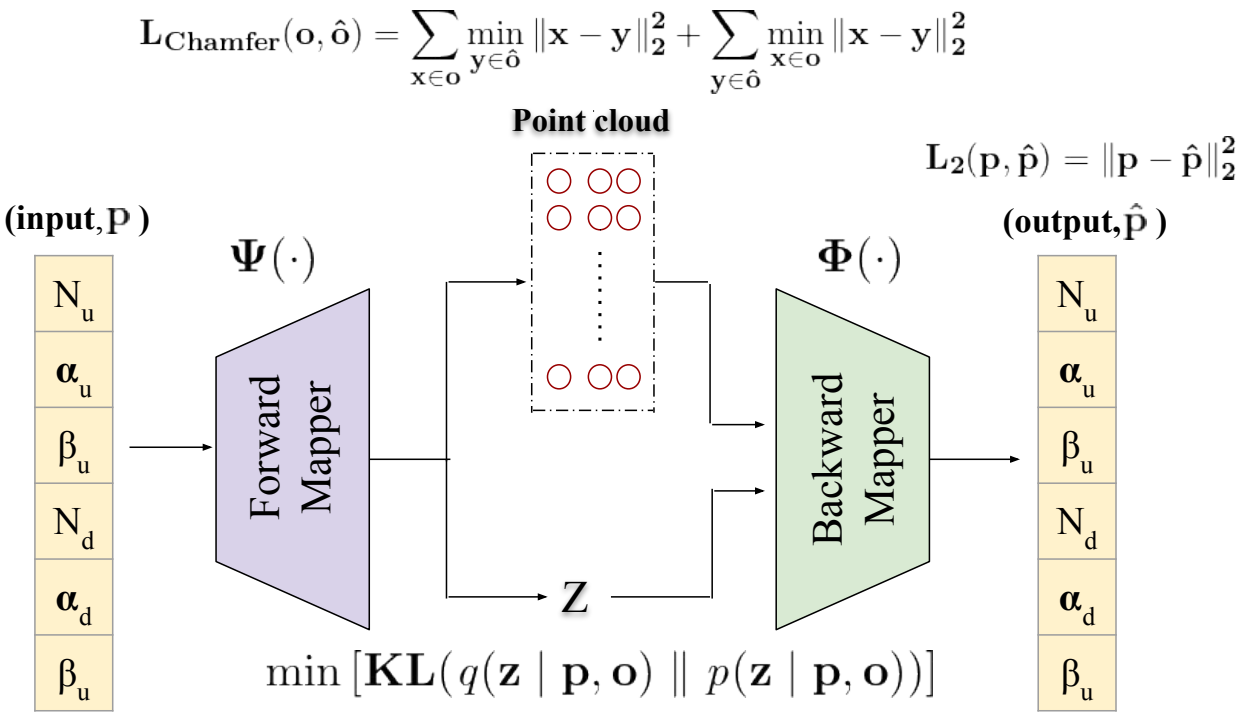



**Fig. 19.** Overview of PC-VAIM architecture: The forward mapper $\Psi(\cdot)$ takes parameters **p** as input and produces the observable point cloud **p** and the latent **z**. Then, the backward mapper $\Phi(\cdot)$ reconstructs the parameter $\hat{\mathbf{p}}$. The detailed architecture of the backward mapper $\Phi(\cdot)$ is illustrated in Fig. 18 [90].

whose nodes represent the parameter array, followed by three fully-connected layers forming a residual block, activated using hyperbolic tangent function (tanh). Each residual block contains 512 hidden nodes, regulated by a $L2$-norm regularizer and a dropout rate of 0.2. The output layer generates the observable point cloud and the latent variable $\mathbf{z}$, which is set to a 100-dimensional standard normally distributed random variable.

**PointNet-based Backward Mapper $\Phi(\cdot)$:**

The backward mapper $\Phi(\cdot)$ converts the observables and the latent variable sample back to the parameters. Since the observables, represented as a point cloud, is unordered, we employ a PointNet-based [18] architecture to convert them into high-dimensional features invariant of permutation. The PointNet-based architecture consists of two main components: a joint alignment networks (T-Net) and a max pooling layer. The T-Net aligns both input points and point features. The max pooling layer serves as a symmetric function, which is permutation invariant, to accumulate all of the points into a global feature vector. The global feature vector represents the accumulated features of the observable point cloud. Concatenated with the latent variable $\mathbf{z}$, the global feature vector is fed to two dense layers to reconstruct the parameter array.

Similar to VAIM, the training of PC-VAIM is a hybrid of supervised learning and unsupervised learning. The training of the forward mapper from parameters to observables and the backward mapper from observables to parameters is supervised from the parameter-observable point cloud pairs in the training data set. On the other hand, the derivation of the posterior distribution $p(\mathbf{z}\,|\,\mathbf{p},\mathbf{o})$ is unsupervised.

## 5.4 RESULTS

In this section, we discuss the applications of PC-VAIM. Starting with a simple toy inverse problem with multiple solutions, we investigate the capability of PC-VAIM in addressing the ambiguity issue in inverse problems. We then demonstrate the application of PC-VAIM on the inverse problem occurring in QCD global analysis.

### 5.4.1 A Toy Inverse Problem

We first demonstrate the capability of PC-VAIM in addressing the ambiguity issue in inverse problem using the following toy problem with known analytical solutions

$$f(x) = (ax)^2, a \in [-2,2], x \in [-1,1].$$

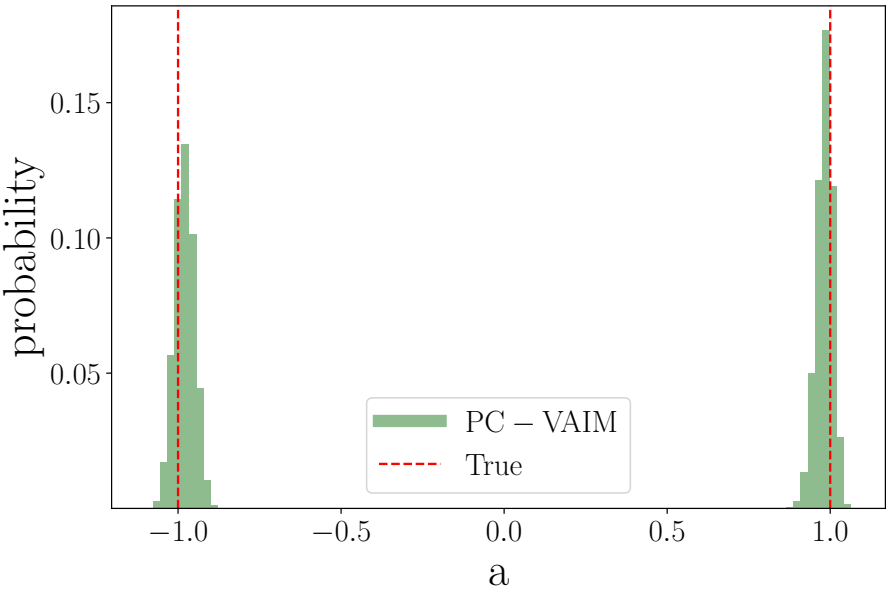
In this toy inverse problem, the observables are the $f(x)$ values at $x$ and the parameter to be de-

termined is $a$. Clearly, this toy inverse problem is ambiguous – for a set of observables $f(x)$ at $x$ corresponding to $a$, $-a$ is another parameter that can produce the same observables. The information lost in forward mapping in this toy problem is the sign of $x$, which leads to ambiguous $a$ and $-a$ in backward mapping.

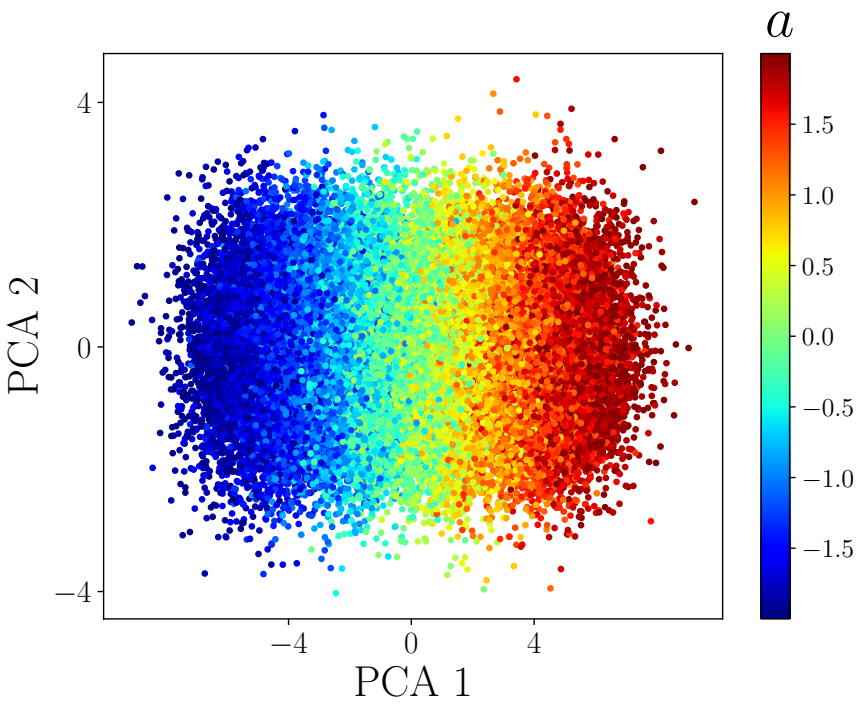
To generate training samples for this problem, we first generate a set of uniformly distributed parameters $a_i \in [-2,2]$. Then, for each parameter $a_i$, we randomly generate $n = 100$ points of $x_j \in [-1,1]$ and then calculate the $f(x_j)$. Hence, a 2D point cloud of $\mathbf{o_i} = \{x_j, f(x_j)\}_1^n$ is built for each $a_i$. The observable point cloud-parameter pair $\mathbf{o_i}, a_i$ becomes a training sample to train PC-VAIM. We generate $10^5$ point cloud-parameter pairs as the training set to train PC-VAIM. Noise is added to the training data by allowing the data to vary within 5% of the true values. The latent variables $\mathbf{z}$ is confined to a 10D uniform distribution in training. The training terminates when the reconstruction error is less than $10^{-4}$. Once PC-VAIM is trained, the backward mapper is applied to solve the toy inverse problem by predicting the parameters given the test observable point cloud coupled with samples of the latent variable $\mathbf{z}$. The backward mapper here behaves like a generative model, where sampling the latent variables can lead to the likelihood distribution of the parameter $a$.

Given a control point cloud generated by $a = 1$, Fig. 20(a) plots the reconstructed parameter distribution by sampling the latent variable $\mathbf{z}$. One can find that PC-VAIM inherits the capability of VAIM in addressing the ambiguous solutions in an inverse problem. The two peaks correspond to the two possible parameters of $a = 1$ and $a = -1$, indicating that PC-VAIM is able to correctly and completely identify the two solutions in this problem.

We further investigate the latent variables after training. Here, we perform principle component analysis (PCA) on latent variable $\mathbf{z}$ samples with respect to the input point clouds in the training set. These samples are plotted in Fig. 20(b) with respect to the first two PCA component vectors (PCA1 and PCA2). One can find that the lost sign information is well-captured in the latent variables, where the outputs corresponding to positive $a$s and those corresponding to negative $a$s are well separated. This indicates that the sign information is well preserved in the latent variable $\mathbf{z}$ after PC-VAIM training. When the latent variable is appropriately sampled, the sign information lost in forward mapping is retrieved, which helps the backward mapper reconstruct the two parameter solutions.




**A** . PC-VAIM predicts the parameter distributions at $a = 1$ and $a = -1$.




**B** . Distribution of $a$ on latent space projected on the first two principle components.

**Fig. 20.** The distribution of parameter $a$ predicted by PC-VAIM in toy inverse problem $f(x) = (ax)^2$ [90].

### 5.4.2 QCD Global Analysis

We apply PC-VAIM to an inverse problem for QCD global analysis described in Sec. 5.2.1. We build the training samples as follows. Firstly, we generate uniformly distributed data points for the PDF parameters within specific ranges for each parameter, namely, $0 < N_u < 2$ and $0 < N_d < 2$ for normalization, $0 < \alpha_u < 3$, $0 < \beta_u < 5$, $0 < \alpha_d < 3$, and $0 < \beta_d < 5$ for the polynomial coefficients. Then, for each parameter, we generate $\sigma_p$ and $\sigma_n$ values across the kinematics $x$ points according to Sec. 5.2.1. There are 10 random kinematic bins of $x$, which gives 10 data points for $\sigma_p$ and 10 for $\sigma_n$. Totally, there are 20 data points in combine. For each parameter vector, there is a corresponding list of cross section and kinematics $x$ values. Finally, using the above generated data, we construct our observable point clouds by stacking the kinematics $x$ and cross sections $\sigma_p$ and $\sigma_n$. Each observable point cloud corresponds to a 6D parameter array $\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$. After all, we generate $10^6$ point cloud-parameter pairs as the training set.

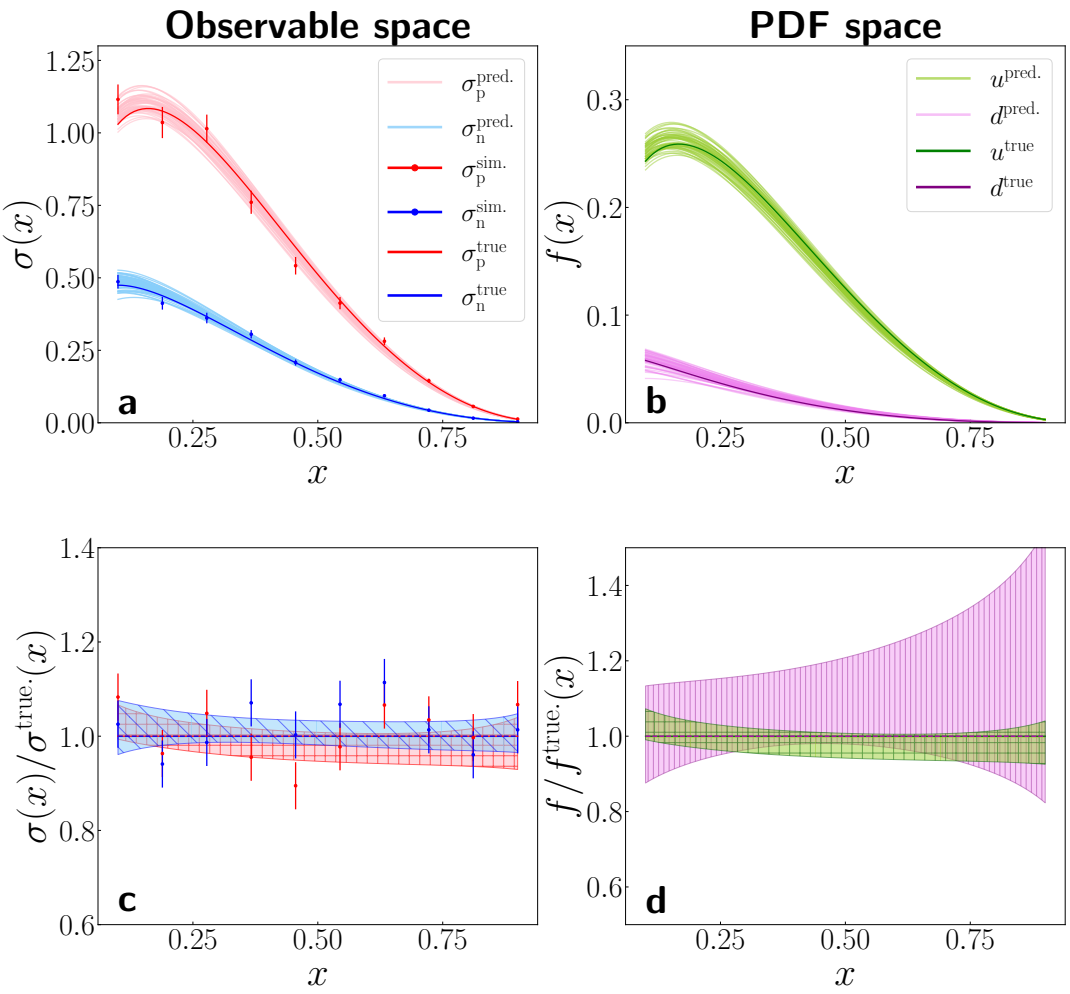



**Fig. 21.** PC-VAIM predictions on the 1,000 simulated DIS control samples given by $\sigma_p^{sim}$ and $\sigma_n^{sim}$. (a) Reconstructed observable $\sigma_p$ and $\sigma_n$ for the PC-VAIM predicted parameters using physics theory model. (b) PDFs for the "up" and "down" quarks produced by physics theory model corresponding to the predicted parameters. (c) Ratio of the reconstructed observables over the true observables. (d) Ratio of the reconstructed PDFs over the true. PC-VAIM correctly learns the mapping between the observable space and the PDF space [90].

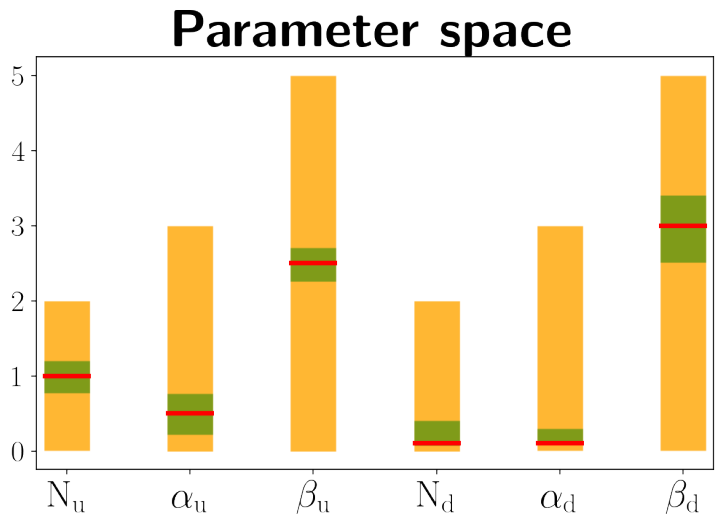



**Fig. 22.** Parameter regions from which the training samples are drawn (orange), the control data (red), and PC-VAIM predictions (green band) [90].

We train PC-VAIM, whose architecture is illustrated in Fig. 19, with the parameters $\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$ as inputs to the forward mapper and outputs point clouds ($x$, $\sigma_p$, $\sigma_n$) as well as the latent variable $\mathbf{z}$. The backward mapper takes point clouds ($x$, $\sigma_p$, $\sigma_n$) and $\mathbf{z}$ as inputs to predict the parameters $\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$. The training terminates when the reconstruction error is less than $10^{-4}$. Once our PC-VAIM is trained, we use the backward mapper to predict the parameters $\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$.

For validation of PC-VAIM predicted parameters, we use a physics theory model that can reconstruct the cross sections $\sigma_p$ and $\sigma_n$ back from the PC-VAIM predicted parameters $\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$. We test it on a simulated control sample $\{N_u = 1, \alpha_u = 0.5, \beta_u = 2.5, N_d = 0.1, \alpha_d = 0.1, \beta_d = 3.0\}$ that is not included in the original training data. We generate $1,000$ simulated control samples by adding 5% Gaussian noise to the true sample. The $\sigma_p$ and $\sigma_n$ functions are shown in Fig. 21(a) as red and blue curves with the $1,000$ simulated control samples of $\sigma_p$, $\sigma_n$ as the red and blue bars, respectively. Using the trained backward mapper on the control sample and sampling the latent variable $\mathbf{z}$, PC-VAIM predicts the distributions of parameters $\{N_u, \alpha_u, \beta_u, N_d, \alpha_d, \beta_d\}$. The predicted ranges of parameters using PC-VAIM are denoted as the green bars corresponding to the $1,000$ simulated control samples shown in Fig. 22, compared to the true parameters specified as the red lines with values and the distributions of the whole training parameters in orange. One can find that the parameters predicted by PC-VAIM are consistent with the true parameters. We

use the physics theory model to reconstruct $\sigma_p$ and $\sigma_n$ functions based on the predicted parameters. As shown in Fig. 21(a), the predicted $\sigma_p$ and $\sigma_n$ functions are in good agreement with the true ones, with $\sigma(x)/\sigma^{pred}(x)$ ratio near 1 (Fig. 21(c)). With the predicted parameters, the reconstructed PDFs, shown in Fig. 21(b), are also consistent with PDFs for the "up" and "down" quarks produced by the physics theory model, which are centered in the predicted PDFs (Fig. 21(d)). This indicates that the trained PC-VAIM has correctly learned the mapping relationship between the observable space and the PDF space.

One of the main advantages of PC-VAIM is that the point cloud representation allows flexible number of data points, which enable us to estimate the uncertainty associated with certain input. Here, we apply PC-VAIM to the exact control sample in the above scenario shown in Fig. 21, but performing data distortion by removing $\sigma_p$ and $\sigma_n$ data points corresponding to low kinematics $x < 0.5$. One can find that the lack of information in low kinematics region leads to large variance in the low $x$ region in the PDF space, as shown in Fig. 23. This analysis will guide the experiments in practice, demanding more experiments to collect data in low kinematics region.

## 5.5 CONCLUSION

Utilizing PC-VAIM to map the inverse functions in QCD global analysis can offer new possibilities beyond the well-adopted standard methodology based on Bayesian inference. In this work, we presented PC-VAIM, a deep learning framework based on flexible, unordered data points represented as a point cloud, to solve the inverse problem in QCD global analysis. The PC-VAIM architecture is composed of a forward mapper NN to learn mapping from parameter space to observable space and a backward mapper NN to recover parameters from observables. A latent layer, whose purpose is to capture information otherwise lost in forward mapping, is incorporated between the forward and backward mappers, forming a significant role in the approximation of the posterior parameter distribution with respect to observables through variational inference. The learned latent information augments the observable space and converts an ill-posed inverse problem into a well-posed regression problem. Incorporating the permutation invariant PointNet-based architecture to the backward mapper enable PC-VAIM to handle the ill-defined, varying observable inputs. This gives the PC-VAIM framework the ability to reconstruct the parameters in QCD global analysis independent of specific kinematics bin setups.

Our computational results of PC-VAIM on a toy inverse problem show its capability of addressing the ambiguous solution issue in inverse problems. PC-VAIM also demonstrates its preliminary success in mapping PDFs to cross sections in a proxy application for real QCD global analysis. PC-VAIM is an extension to the VAIM architecture, intending to address the input flexibility prob-

lem. Nevertheless, designing effective machine learning methods to solve the inverse problems in QCD global analysis is still in its beginning stage. There are many challenges ahead, including solution stability, robustness, and uncertainty quantification. These will be the focus of our future research.

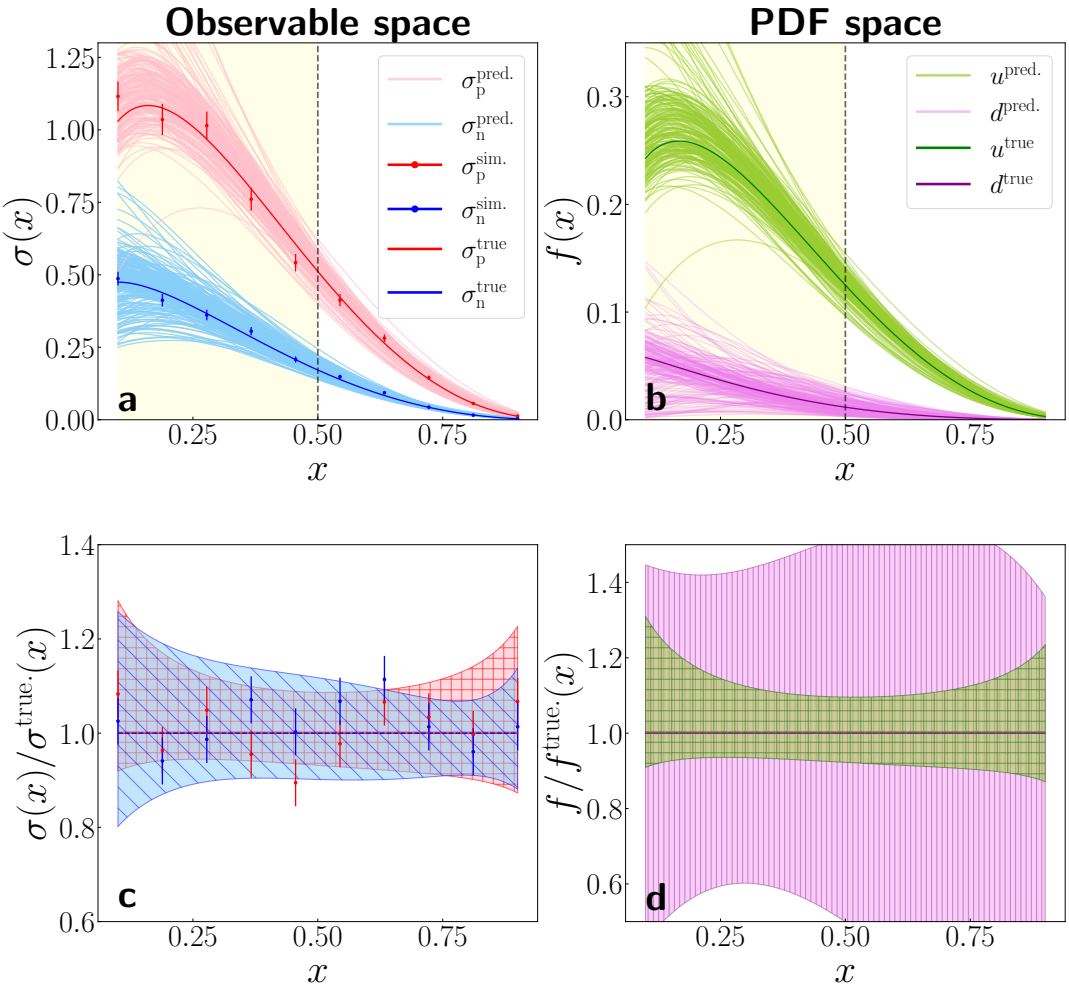



**Fig. 23.** PC-VAIM predictions on the simulated DIS control samples, with data points at low kinematics $x < 0.5$ removed. (a) Reconstructed $\sigma_p$ and $\sigma_n$ for the PC-VAIM predicted parameters using physics theory model. (b) PDFs for the "up" and "down" quarks produced by physics theory model corresponding to the predicted parameters. (c) Ratio of the reconstructed observables over the true. (d) Ratio of the reconstructed PDFs over the true PDFs. Large variance in low kinematics region in the PDF space indicates more data are needed in this region.

# CHAPTER 6

# COMPTON FORM FACTORS EXTRACTION

In this chapter we present our methods to the inverse problem of extracting the GPD Compton form factors (CFFs) from the DVCS cross sections. First, we utilize VAIM, introduced in chapter. 4 to extract Compton form factors on a fixed kinematics value. Second, we extend VAIM to the conditional VAIM (C-VAIM) for extracting CFFs on several kinematics values. Then, we present our results and analysis on the DVCS experimental examples.

## 6.1 INTRODUCTION

The high-energy scattering experiments in the past several decades have generated a large amount of data for the QCD [5] analysis to explore the inner structure of protons and neutrons. A fundamental tool for exploring the internal structure of the proton is deeply virtual scattering processes. In particular, Deeply Virtual Compton Scattering (DVCS) which encodes the quantum mechanical phase space distributions known as generalized parton distribution functions (GPDs). The GPDs are not accessed directly through DVCS but instead the integrals of GPDs over the longitudinal momentum transfer of the quarks $x$, and the Compton form factors. This means the GPDs information is hidden in the values of the CFFs because the latter one integrates the GPD over $x$. The challenge here is that the extraction of the GPDs from the CFFs represents an ill-posed inverse problem [67] with potentially multiple solutions. The inverse problem of extracting the CFFs from polarization observables presents a significant interest in QCD, because it is the easiest channel to access GPDs. With the success of extracting CFFs, the imaging of the nucleon's internal 3-dimensional structure is feasible.

Recent machine learning approaches have become state-of-the-art in solving various inverse problems. Applied to several inverse problems in the science and engineering fields, deep learning based methods have shown robust performance compared to traditional numerical solution methods [92]. In our previous work in chapter. 4 , we develop the VAIM to determine the multiple solutions to the parameterization of quark/anti-quark/gluon parton distribution functions from the observed cross sections. We conclude that through the use of the VAIM framework, one can reconstruct the lost information through the latent space variables. Given cross section as input, sampling the latent layer was able to reconstruct the lost information and as a consequence determine the corresponding parameter distribution of the parton distribution function.

In this chapter, we target a similar goal in the context of DVCS. Treating the experimental

observable as an equation that is parametrized by CFFs, we can reframe the VAIM to study CFFs extraction from the observable on fixed kinematics. We proceed then to extend our VAIM framework to the conditional VAIM (C-VAIM) to allow more flexibility to include several kinematics values. We demonstrate our results of CFFs extraction from a single experimental observable, the unpolarized cross section, training on Jefferson Lab Halls A and B data, 6 GeV and 12 GeV experimental runs.

## 6.2 BACKGROUND OF THE GPDS

The cross section for deeply virtual exclusive photoproduction, $ep \to e'p'\gamma$, on a proton reads[93, 94, 95], defined as

$$\frac{d^5\sigma}{dx_{Bj}dQ^2d|t|d\varphi d\varphi_S} = \Gamma\left|T\right|^2$$

At leading order in $1/Q^2$, $T$ is given by the superposition of the amplitudes for DVCS and a competing background process Bethe-Heitler (BH),

$$\left|T^2\right| = |T_{\rm BH} + T_{\rm DVCS}|^2 = |T_{\rm BH}|^2 + |T_{\rm DVCS}|^2 + \mathscr{I}$$

$$\mathscr{I} = T^*_{BH}T_{DVCS} + T^*_{DVCS}T_{BH}$$

where:

- $Q^2 = -(k-k')^2$ is the four momentum transfer squared between the initial ($e(k)$) and final ($e'(k')$) electrons.
- $x_{Bj} = Q^2/(2(pq))$ where $p$ and $q$ are the 4 vectors of the initial proton and virtual photon respectively.
- $t = \Delta^2 = (p'-p)^2$, the four momentum transfer squared between the initial and final state proton.
- $\varphi$ is the azimuthal angle between the lepton and hadron planes.
- $\varphi_S$, the azimuthal angle of the transverse proton spin vector, which we integrate over in our analysis since we are not considering scattering from a polarized target.

Encoded in the amplitude, $T$, are the Compton form factors, which are convolutions of GPDs with perturbatively calculable Wilson coefficient functions. The CFF $\mathscr{F}^q$ where $F^q$ represents the GPDs, $H^q, E^q$ for the quark flavor, $q = u, d$, is given as:

$$\mathscr{F}^q(\xi,t) = \mathscr{C}(x,\xi;Q^2) \otimes F^q(x,\xi,t) = e_q^2 \int_{-1}^{+1} dx \Big[\frac{1}{\xi - x - i\varepsilon} - \frac{1}{\xi + x - i\varepsilon}\Big] F^q(x,\xi,t)$$

and similarly for the polarized Compton form factors

$$\widetilde{\mathscr{F}}^q(\xi,t) = \mathscr{C}(x,\xi;Q^2) \otimes \widetilde{F}^q(x,\xi,t) = e_q^2 \int_{-1}^{+1} dx \Big[\frac{1}{\xi - x - i\varepsilon} + \frac{1}{\xi + x - i\varepsilon}\Big] \widetilde{F}^q(x,\xi,t)$$

One can separate out the real and imaginary parts of the CFFs and sum over parton flavors for the total proton contribution that enters the cross section:

$$\mathrm{Re}\, e\mathscr{F}(\xi,t) = e_q^2 P.V. \int_{-1}^{+1} dx \Big[\frac{1}{\xi - x} - \frac{1}{\xi + x}\Big] F^q(x,\xi,t)\, \mathrm{Im}\, m\mathscr{F}(\xi,t) =$$
$$\pi e_q^2 \Big(F^q(\xi,\xi,t) - F^q(-\xi,\xi,t)\Big)$$

where P.V. is the principal value integral. Similar expressions can be written for the polarized Compton form factors

$$\mathrm{Re}\, e\widetilde{\mathscr{F}}(\xi,t) = e_q^2 P.V. \int_{-1}^{+1} dx \Big[\frac{1}{\xi - x} + \frac{1}{\xi + x}\Big] \widetilde{F}^q(x,\xi,t)\, \mathrm{Im}\, m\widetilde{\mathscr{F}}(\xi,t) =$$
$$\pi e_q^2 \Big(\widetilde{F}^q(\xi,\xi,t) + \widetilde{F}^q(-\xi,\xi,t)\Big)$$

The $x$-dependence of the GPDs $H$ and $E$ directly describe physical properties of the nucleon such as partonic (for a quark, $q$, or a gluon, $g$) angular momentum in the proton through the sum rule first written down by Ji [96] as,

$$J^{q/g} = \frac{1}{2} \int_{-1}^{+1} dx x (H^{q/g}(x,0,0) + E^{q/g}(x,0,0)).$$

It is clear that the convolutions in the CFFs present a challenge for extracting GPDs from exclusive scattering experiments. The full $x$-dependence of the GPD is not directly accessible through any experimental observables; contrasted with inclusive scattering experiments from which the structure functions can be reconstructed through many measurements in values of $x_{Bj}$. The ultimate objective of nuclear femtography is to extract GPDs for nuclear imaging and understanding physical properties of the nucleon. We will break this task down into a multi-step framework to go from cross section data to the physical properties encoded:

1. Analyzing trends in exclusive scattering data, and understanding regions of high impact for experimental study.

2. Extraction of Compton form factors from data with exploration of various error analysis techniques.

3. Reconstructing the GPD using contributions from the lattice, and other theoretical constraints.

4. Calculating physical quantities from the reconstructed GPDs necessary for femtography, including images.

In this chapter, we target the Compton form factors extraction from the cross sections. We demonstrate the capabilities of ML algorithms, particularly VAIM in extracting Compton form factors from data including the propagation of experimental error on extracted quantities. We focus on unpolarized scattering data from Jefferson Lab 6 GeV and 12 GeV experiments. We will show the information that can be extracted from single polarization experimental observables [97].

## 6.3 VAIM FOR CFFS

As a first step in CFFs extraction, we utilize VAIM framework, discussed in chapter 4 to extract the CFFs with respect to the given unpolarized cross section on one set of kinematics value for $x_{bj}$, $t$ and $Q^2$. As illustrated in Fig 24, VAIM consists of two networks: a forward mapper $\Psi(\cdot)$ and a backward mapper $\Phi(\cdot)$. The forward mapper $\Psi(\cdot)$ maps the CFFs to the cross sections and latent variable $z$ and the backward mapper $\Phi(\cdot)$ maps the coupled cross sections and the latent variable $z$ to the CFFs.

Indicating the Compton form factors as $\mathbf{cff} \in \mathbb{R}^m$, the cross sections as $\mathbf{x} \in \mathbb{R}^n$, and the latent variables in the latent layer as $\mathbf{z} \in \mathbb{R}^k$, the forward mapper learns the one-to-many mapping from CFFs to the cross sections such that

$$\Psi(\mathbf{cff}; w_f) = [\hat{\mathbf{x}}, \mathbf{z}],$$

$w_f$ represents the weights of the forward mapper neural network, and $\hat{\mathbf{x}}$ is an approximation to the observed $\mathbf{x}$. The backward mapper is trained to reconstruct **cff** according to $\mathbf{x}$ and $\mathbf{z}$ such that

$$\Phi(\mathbf{x}, \mathbf{z}; w_g) = \hat{\mathbf{cff}},$$

Where $w_g$ are the weights of the backward mapper neural network, and $\hat{\mathbf{cff}}$ approximates the Compton form factors **cff**. Similar to modeling the inverse problem of PDF in sec.4.2.1 , we model VAIM to extract CFFs on a single kinematics bin as following:

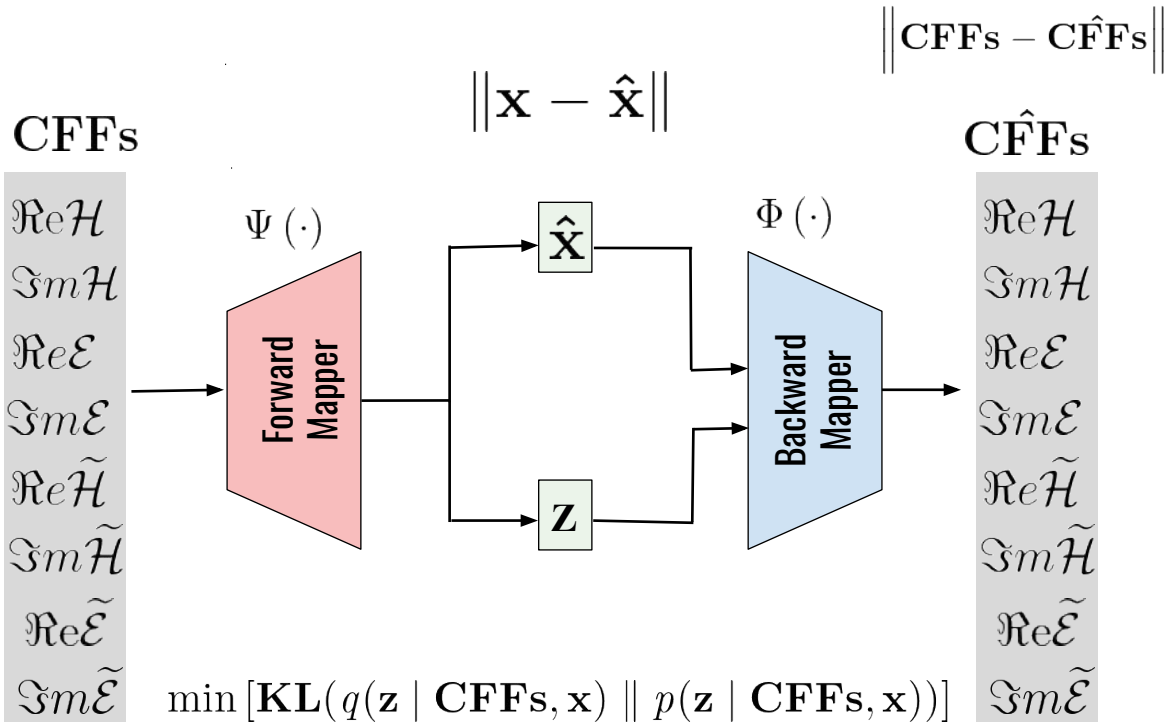



**Fig. 24.** A schematic view of VAIM architecture

$$\min_{w_f,w_g}\Big[\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2+\|\mathbf{cff}-\hat{\mathbf{cff}}\|_2^2+\mathrm{KL}\big(q(\mathbf{z}\mid\mathbf{cff},\mathbf{x})\mid\mid p(\mathbf{z})\big)\Big],$$

Where $\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2$ is the forward mapping error, $\|\mathbf{cff}-\hat{\mathbf{cff}}\|_2^2$ is the likelihood error of reconstructing the Compton form factors, and $p(\mathbf{z})$ is the true prior distribution. Throughout VAIM training, the latent layer seeks to learn the lost information in forward mapping and converts the ill-posed inverse problem into a well-posed regression problem in the backward mapper using the carried information in the latent space. Specifying $z$ dimensions as greater or equal to the actual dimensionality of the information lost in forward mapping, the latent layer can reconstruct the Compton form factors distribution for the given observables.

### 6.4 CONDITIONAL VAIM (C-VAIM)

In order to expand CFFs extraction to include several sets of kinematics, we extend VAIM to the Conditional VAIM (C-VAIM) that is based on the standard C-VAE [98]. The overall architecture of C-VAIM is shown in fig. 25 , consisting of a forward mapper $\Psi(\cdot)$ and a backward mapper $\Phi(\cdot)$. The forward mapper $\Psi(\cdot)$ maps the CFFs and the kinematics to the cross sections and latent variable $z$ and the backward mapper $\Phi(\cdot)$ maps the coupled cross sections, the kinematics and the latent variable $z$ to the CFFs.

C-VAIM provides the advantage of imposing conditions over the forward and backward mappers using the kinematics as an input. Thus, we can control the structure of the generated data during the backward mapping. In other words, with the training process on the entire kinematics data, we want our backward mappers to generate CFFs that correspond to a certain kinematics region. For this task, our VAIM assumption is slightly modified by conditioning the forward and the backward mapper to the kinematics $(x_{bj},Q^2,t)$ denoted as $v$. In C-VAIM, the forward mapper attempts to approximate $p(\mathbf{z}\,|\,\mathbf{cff},\mathbf{v},\mathbf{x})$ by another distribution $q(\mathbf{z}\,|\,\mathbf{cff},\mathbf{v},\mathbf{x})$, and the backward mapper learns to approximate the likelihood distribution $p(\mathbf{cff},\mathbf{v},\mathbf{x}\mid\mathbf{z})$.

Similar to VAIM, KL divergence is utilized in C-VAIM to minimize the divergence of the two distribution $q$ and $p$. Thus, C-VAIM is modeled as follows

$$\min\Big[\mathrm{KL}\big(q(\mathbf{z}\,|\,\mathbf{cff},\mathbf{v},\mathbf{x})\mid\mid p(\mathbf{z}\,|\,\mathbf{cff},\mathbf{v},\mathbf{x})\big)\Big],$$

The above optimization leads to the following revised objective function

$$\min_{w_f,w_g}\Big[\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2+\|\mathbf{cff}-\hat{\mathbf{cff}}\|_2^2+\mathrm{KL}\big(q(\mathbf{z}\mid\mathbf{cff},\mathbf{v},\mathbf{x})\mid\mid p(\mathbf{z}|\,\mathbf{v})\big)\Big],$$

$\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2$ represents the forward mapping error, $\|\mathbf{cff}-\hat{\mathbf{cff}}\|_2^2$ is the likelihood error of reconstructing the Compton form factors, and $p(\mathbf{z}|\,\mathbf{v})$ is the true prior distribution conditioned by the kinematics

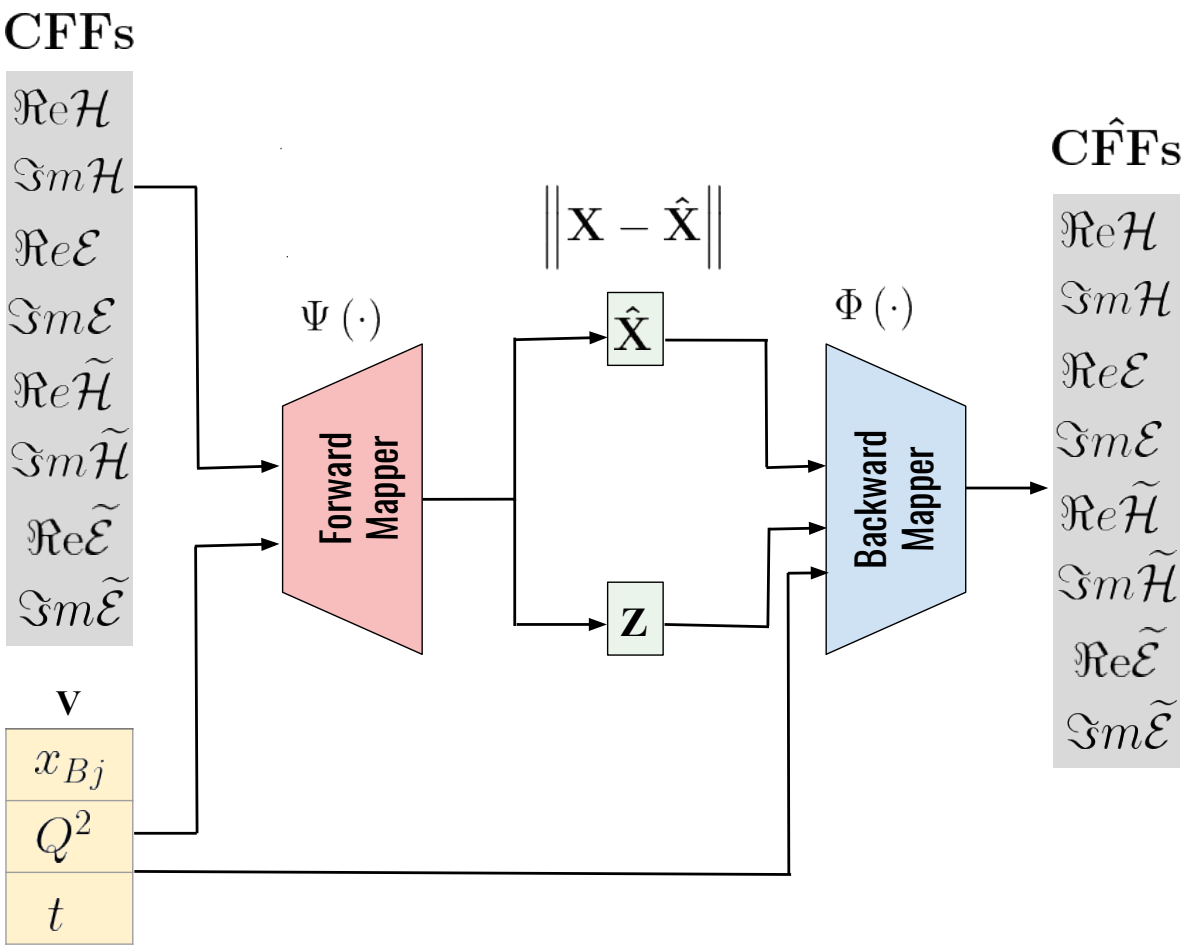



**Fig. 25.** C-VAIM architecture

$v$. As VAIM, we choose $p(\mathbf{z})$ to be a tractable, easy-to-generate distribution, such as a normal distribution or a uniform distribution. The KL-divergence approximation ensures that

$$q(\mathbf{z}\,|\,\mathbf{cff},\mathbf{v},\mathbf{x}) \sim p(\mathbf{z}|\,\mathbf{v}).$$

C-VAIM hyperparameters such as the number of neurons, number of layers, and the learning rate, is selected using Keras tuner [99], an open-source library to designate our architecture and perform hyperparameter tuning. The optimal hyperparameter is used as the final hyperparameter for our C-VAIM architecture. Our optimum forward mapper network architecture composed of an input layer whose nodes denote the concatenated arrays of the CFFs and kinematics followed by three fully-connected layers with 1024 neurons activated by a Leaky ReLU function. The output layer represents the cross section, kinematics and latent varibles $z$. The network is regularized by

an L2-norm penalty and a dropout rate of 0.1 to prevent overfitting. The backward mapper network architecture composed of an input layer that denotes the concatenated cross sections, kinematics with the latent layer $z$, followed by three fully-connected layers with 1024 neurons activated by a Leaky ReLU function.

## 6.5 CFFS EXTRACTION

C-VAIM training process is a hybrid of supervised learning and unsupervised learning. The training of the forward mapper from CFFs and kinematics $v$ to the cross section, and the backward mappers from cross section to CFFs is supervised training. Whereas the derivation of the posterior distribution $p(\mathbf{z}\,|\,\mathbf{cff},\mathbf{v},\mathbf{x})$ is unsupervised training. The latent variable $z$ is distributed under the conditional probability distribution $p(\mathbf{z}\,|\,\mathbf{v})$. This means for every value of $v$, we will have a $p(\mathbf{z})$ that we can use to generate the CFFs corresponds to a specific kinematics value. After C-VAIM is trained, the backward mapper is used as a generative model to extract CFFs by predicting the combined cross sections and kinematics that are coupled with samples of the latent variable that is drawn from a specific $v$, as shown in fig. 26.

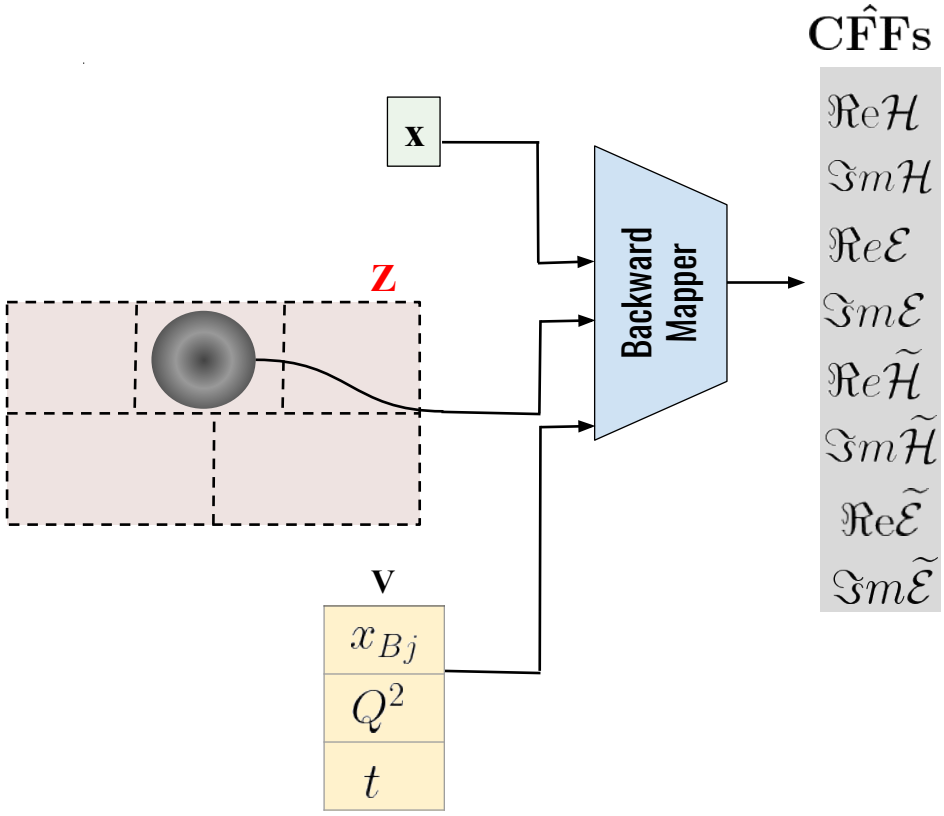



**Fig. 26.** C-VAIM backward mapper is used as a generative model to produce CFFs by sampling **z** with respect to the given cross section and kinematics *v*.

### 6.6 RESULTS

We first utilize the VAIM to extract the CFFs on a fixed kinematics in order to demonstrate the applicability of VAIM for CFFs extraction. Then, we proceed to train C-VAIM to extract CFFs on several kinematics values. We further analyze the results on the CFFs space as well the reconstructed cross sections space of the extracted CFFs using the physics model calculations.

#### 6.6.1 VAIM on One Kinematics Set

We study the feasibility of CFFs extraction using VAIM and training on one set of kinematics ($x_bj$= 0.343, $t$= -0.172, and $Q^2 = 1.82$ GeV$^2$). We test our trained VAIM on a test cross section sample from the separated test set which is not seen during the training process. The predicted CFFs are shown in fig. 27 (top) with high variance in some of the CFFs. In order to verify these

predicted CFFs, we use the theory model based on the physics calculations to reconstruct the corresponding cross section from our predicted CFFs and compare it with the true cross sections. Our results in fig. 27 (bottom) demonstrate a good agreement between the reconstructed cross section of the predicted CFFs and the true cross section.

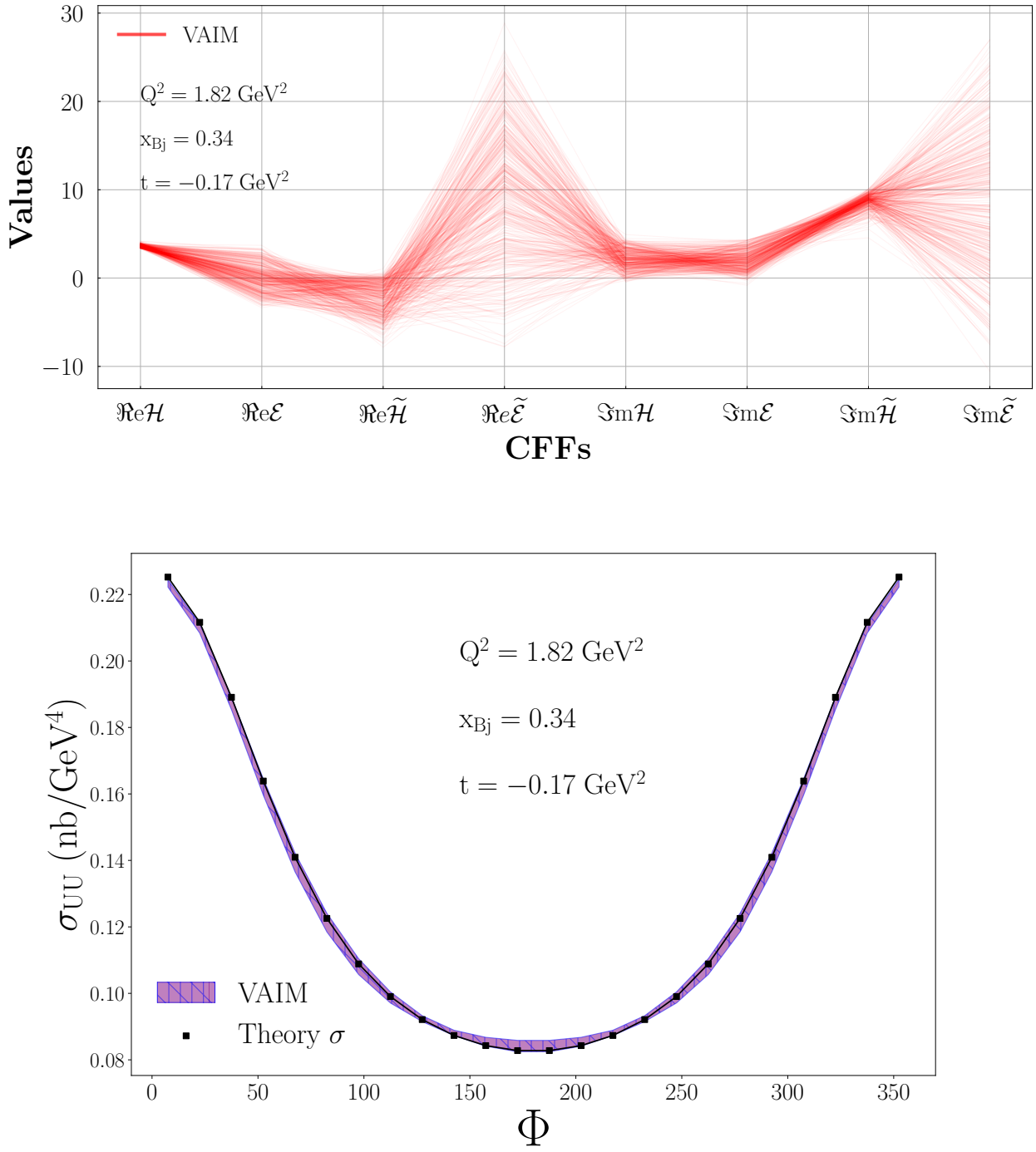



**Fig. 27.** VAIM predictions on a test example on the kinematics $x_{Bj} = 0.343$, $t = -0.17$ GeV$^2$, $Q^2 = 1.82$ GeV$^2$, and $E_b = 5.75$ GeV. (*Top*)VAIM predicted CFFs and (*Bottom*) reconstructed cross sections for the above CFFs using the theory model.

### 6.6.2 C-VAIM on Several Kinematics Sets

We extend our work in extracting CFFs to cover a larger kinematics region using C-VAIM framework. We train C-VAIM on several kinematics $x_{Bj}$, $t$, $Q^2$ values shown in table.2. C-VAIM is not limited to these values in table.2 and can be extended to include all available kinematics.

We test C-VAIM on several experimental DVCS examples and will point out here some of the results. We test on the DVCS cross section of the kinematics $x_b j$= -0.34, $t$= -0.17, and $Q^2$= 1.82. The results of the extracted CFFs is illustrated in fig. 28 (top), the predicted CFFs show a range of all possible solutions, with some CFFs being more well constrained than others by the widths of the prediction bands as compared to the widths of the training inputs. The CFFs are connected by lines to demonstrate correlations between predicted CFFs since all eight are predicted at the same time. The uncertainties of the experimental data are propagated into the CFF predictions through our method of random targets , where the cross sections are augmented using their statistical errors as explained in Sec. 7.4. We take these solutions as an ensemble of Gaussian distributed predictions which we can then statistically treat to find a central value and a standard deviation as the error. These central values and errors are then propagated through the DVCS cross section and show as a band of cross sections as shown in Fig. 28(bottom) which fit the unpolarized cross section.

The extraction of $\mathrm{Re}\,e\mathscr{H}$ seems to be very well determined as shown in Fig. 28(middle) under the influence from both the ML model error and the statistical error from the experimental data. In the same vein, the unpolarized cross section does not seem to be able to narrowly determine the CFFs $\mathrm{Re}\,e\widetilde{\mathscr{E}}$ nor $\mathrm{Im}\,m\widetilde{\mathscr{E}}$. We also see that the CFF from the longitudinally polarized GPD $\mathrm{Re}\,e$ and $\mathrm{Im}\,m\,\widetilde{\mathscr{H}}$ are also rather well determined through the C-VAIM architecture applied to the unpolarized cross section.

**Table 2.** Kinematics values that are used to train C-VAIM

| | $x_b j$ | $t$ | $Q^2$ |
|---|---|---|---|
| 1 | 0.343 | -0.172 | 1.820 |
| 2 | 0.368 | -0.232 | 1.933 |
| 3 | 0.375 | -0.278 | 1.964 |
| 4 | 0.379 | -0.323 | 1.986 |
| 5 | 0.381 | -0.371 | 1.999 |

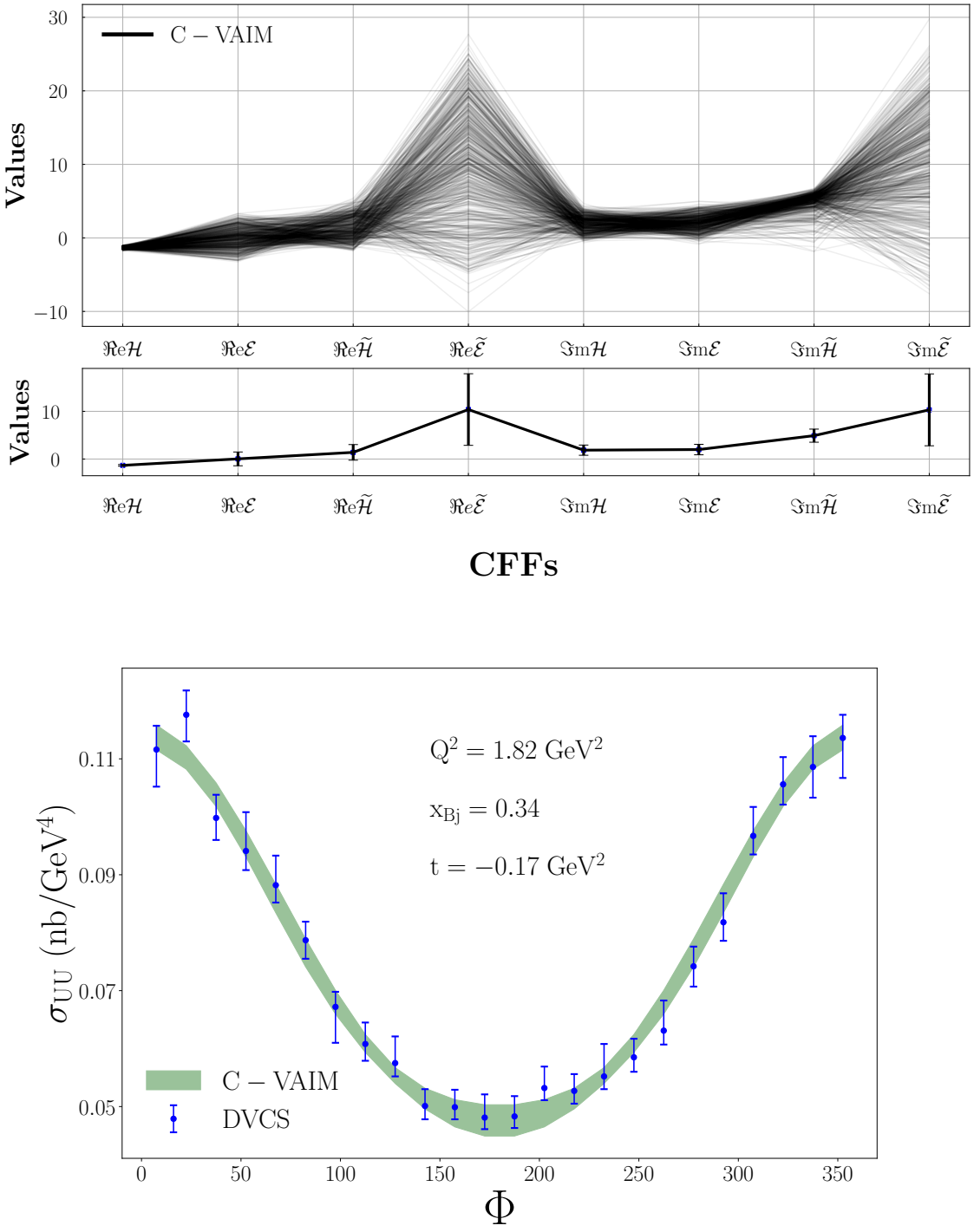



**Fig. 28.** (*Top*) C-VAIM predictions at a fixed kinematic experimental data $x_{Bj} = 0.343, t = 0.172$ GeV$^2$, and $Q^2 = 1.820$ GeV$^2$. (*Middle*) The CFFs central values (blue squares) and standard deviation (bars around the blue squares) of the above solutions. (*Bottom*) Ensembling the solutions as an uncertainty band, green, as compared to DVCS data, blue.

In Figure 29 we demonstrate predictions of CFFs on a range of $t$ values for a fixed kinematics $x_{Bj} = 0.35$ and $Q^2 = 1.9$ GeV$^2$ for Jefferson Lab data at 6 GeV. We use a physics informed deep learning model developed in [97] to interpolate the cross section in between experimental data points to feed into the C-VAIM architecture for predictions. Notice that the CFFs that were rather well determined in Fig. 28 demonstrate noticeable trends in $t$. This indicates that these three Compton form factors Re$e\widetilde{\mathscr{H}}$ , Re$e\widetilde{\mathscr{H}}$ and Im$m\widetilde{\mathscr{H}}$ are more dominating in respect to $t$ values.

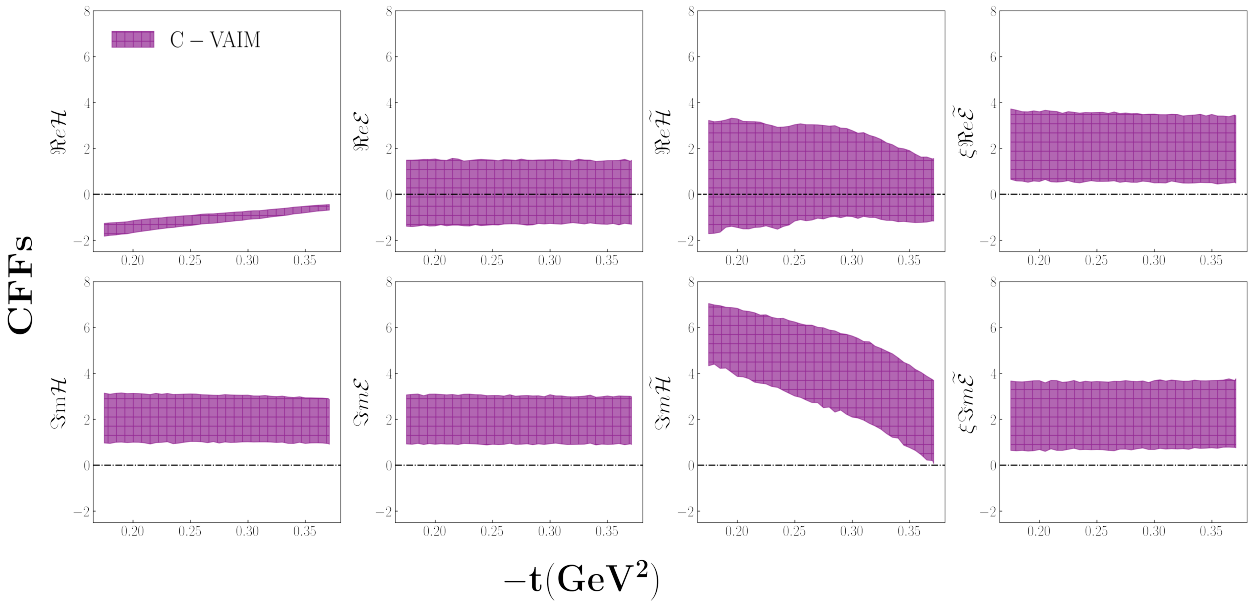



**Fig. 29.** Prediction of all eight CFFs as a function of the variable $t$ for the kinematics $x_{Bj} = 0.35$ and $Q^2 = 1.9$ GeV$^2$ for the JLab 6 GeV kinematics.

Furthermore, we test the C-VAIM architecture on the formulations of the cross section as they exist in the literature (for a review of the differences see [100]). The main cross sections as they are written are denoted as BKM [101], and UVA (labeled here as FemtoNet) [102, 100]. It is important with various formulations of the cross section to understand the effects on the extracted observables such as the CFFs. We generate training samples using the two cross sections formulations and train our C-VAIM independently. C-VAIM models are then applied on the DVCS cross sections for the kinematics $x_{Bj} = 0.343$, $t = -0.172$ GeV$^2$, and $Q^2 = 1.820$ GeV$^2$. The predicted CFFs using the two formulations are indeed overlapped as illustrated in Figure. 30, notice that there is not much difference between the predicted CFFs central values between BKM and FemtoNet; however, the

error analysis is much different. The histograms show this clearly where the CFFs predicted from BKM have a larger error due to the harmonics approach as compared to the FemtoNet cross section. We notice that the differences mainly appear in the interference terms which are $\mathrm{Re}\,e\mathcal{H}$, $\mathrm{Re}\,e\widetilde{\mathcal{H}}$ where they have a noticeable appearance. We compare the results of the central values of the C-VAIM predicted CFFs from both cross section formulations and compare them to the experimental data as shown in in Figure. 31. Since the CFFs only enter into the interference term and the DVCS squared contribution (and not the BH background) we subtract off the exactly calculable BH so that the differences can be seen. Notice that both formulations of the cross section are fitted to the data, so the differences can mainly be seen in the extracted observables as illustrated in Figure. 31.

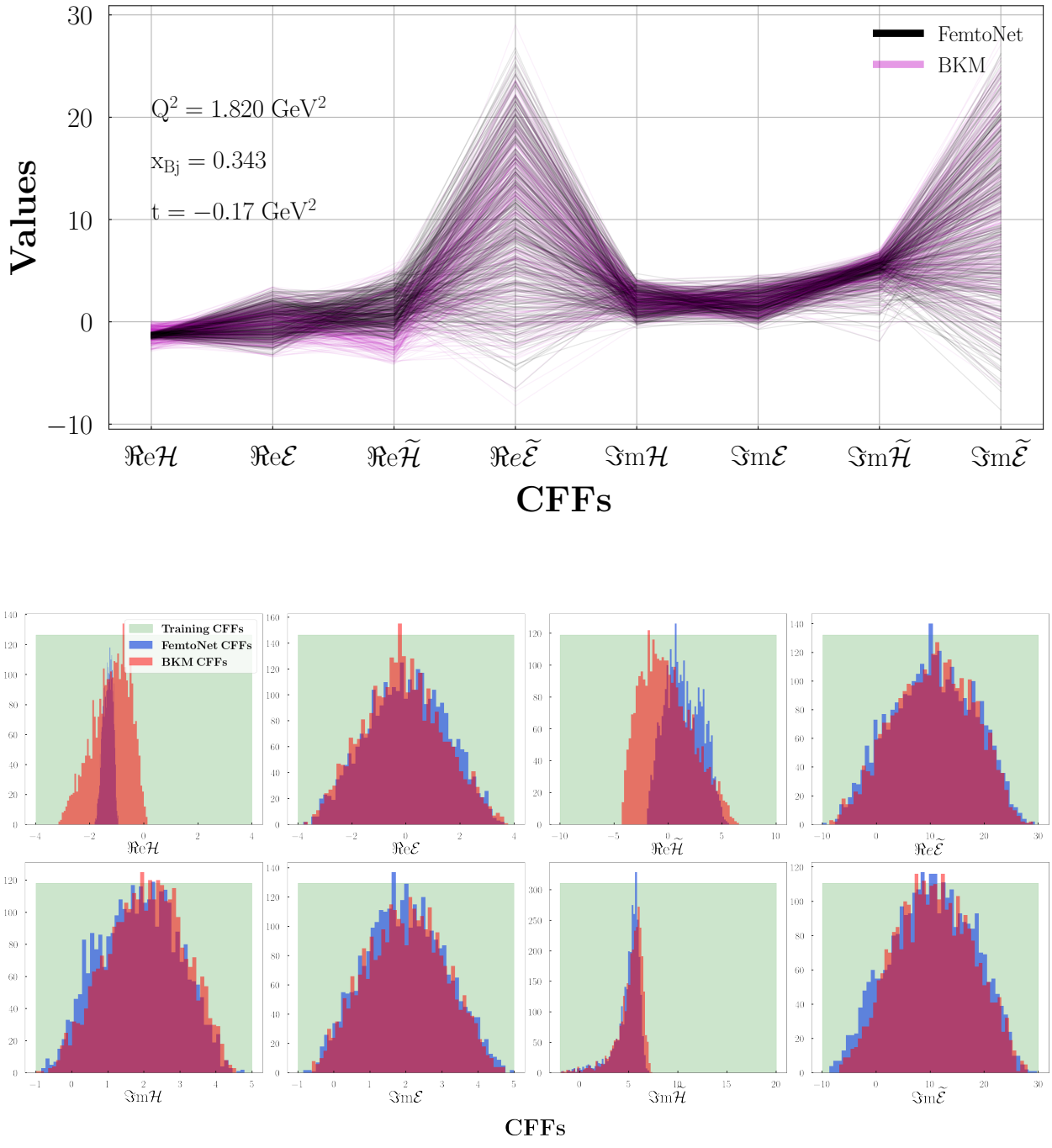



**Fig. 30.** Predictions of CFFs using C-VAIM framework comparing the cross section formulation of UVA labeled FemtoNet and the formulation of BKM. All results are shown for a specific kinematics for the unpolarized cross section at $x_{Bj} = 0.343, t = -0.172$ GeV$^2$, $Q^2 = 1.820$ GeV$^2$, and $E_b = 5.75$ GeV. (*Top*) Comparison of outcomes of C-VAIM trained on the two different cross sections. (*Bottom*) Binning the outputs of C-VAIM as a histogram for each CFF to see the differences between BKM and UVA predicted CFFs. Also shown is the uniform input of CFF solutions for training.

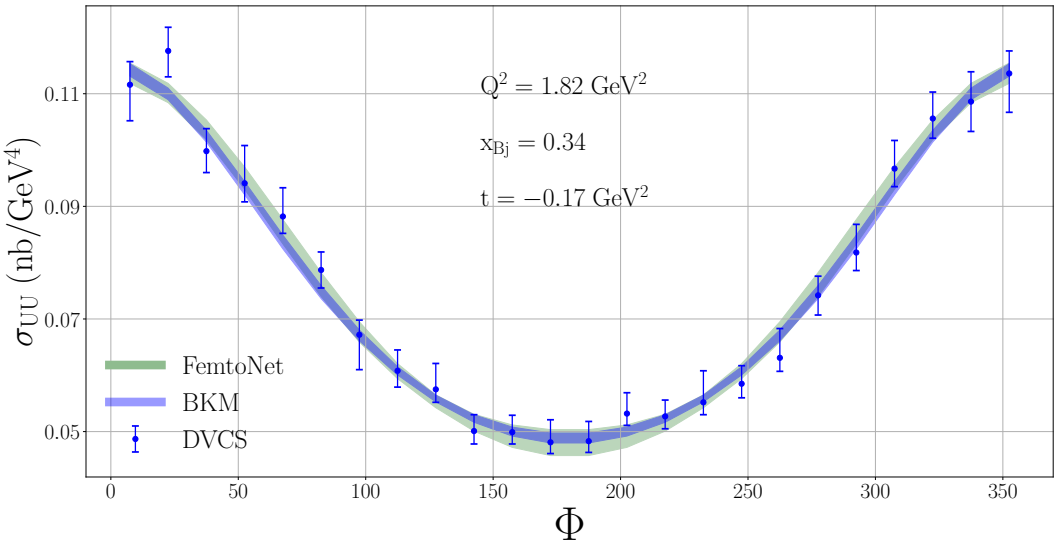
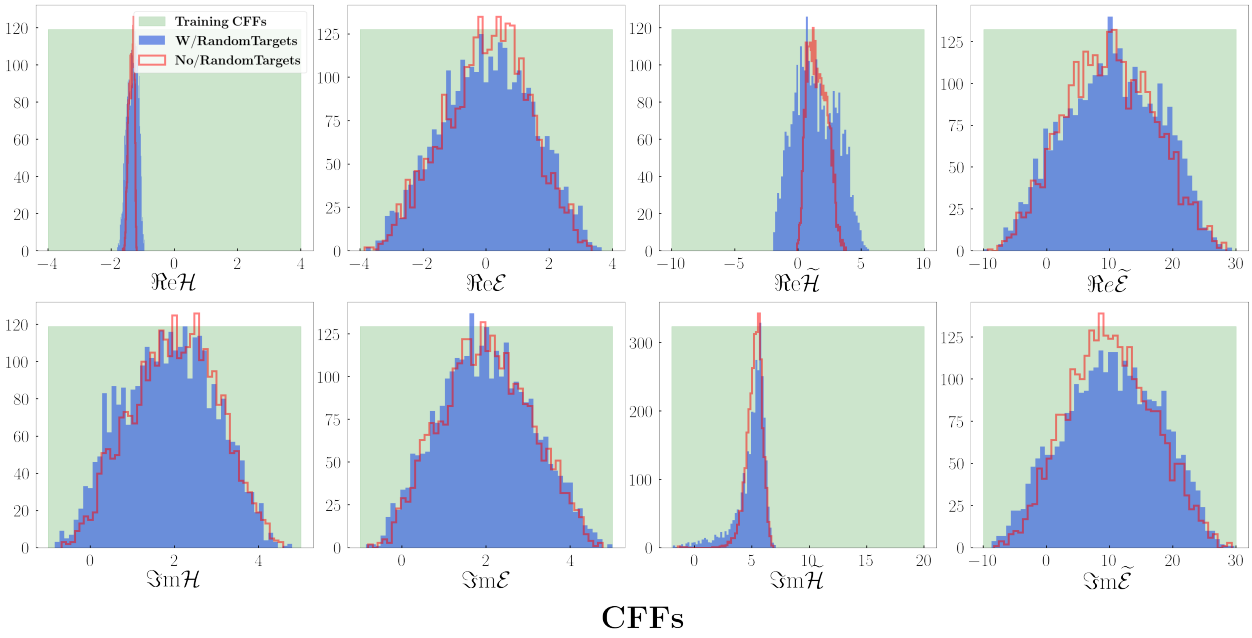



**Fig. 31.** Comparison of the ensemble of CFFs input into the reduced cross section which only has the contributions from DVCS.

In Figure 32 we compare the uncertainty associated to our results with and without the random targets method for the propagation of experimental uncertainties. This can mainly be seen in the presentation of the CFFs through histograms where we overlay the CFFs as predicted once with the random target and once with no random target. With no random target, the CFFs error bands are generated due to the variance of the solutions generated by C-VAIM through the latent variable (systematic errors from the C-VAIM architecture), and for the random targets method (combination of systematic from the C-VAIM and statistical errors from the experimental data). Notice that the random targets errors are consistently bigger, and making a more notable bandwidth on the three dominating CFFs which are $\mathrm{Re}\,e\mathscr{H}$ , $\mathrm{Re}\,e\widetilde{\mathscr{H}}$ and $\mathrm{Im}\,m\widetilde{\mathscr{H}}$ suggesting that these CFFs are more sensitive to the statistical error. This also indicates that we are properly propagating the data uncertainties; however, that the error on the extracted CFFs are almost all contributions from reducible systematic. The reducibility of the systematic errors is left for a future study.



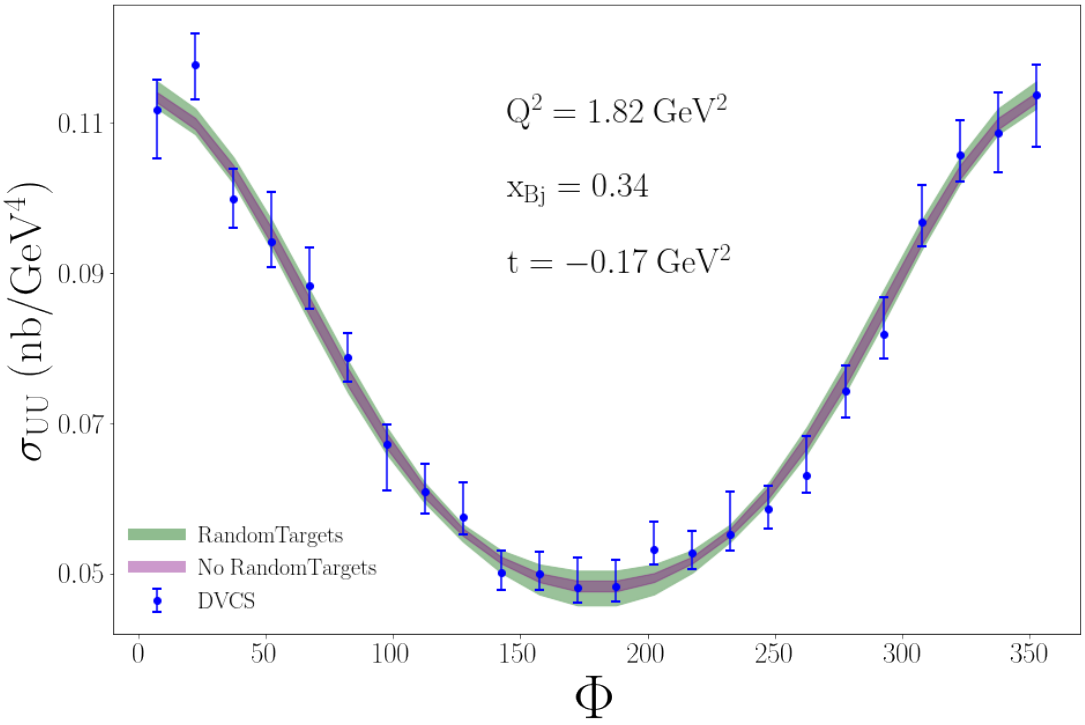




**Fig. 32.** Comparison of extracted CFFs at a particular kinematics $x_{Bj} = 0.343$, $t = -0.172$ GeV$^2$, $Q^2 = 1.820$ GeV$^2$ using C-VAIM architecture both with and without the random targets propagated error. (*Top*) Comparison of the CFFs with and without the random targets method to propagate uncertainty. (*Bottom*) The uncertainty propagated through the DVCS cross section for both the interference term and DVCS squared term.

Using the structure of the DVCS cross section as formulated in [103] and the information contained in the extraction of all eight CFFs we study the sensitivity of the analysis. Using the fact that the interference cross section is linearly dependent on three of the eight total CFFs, and is amplified with respect to the DVCS term by the interference with the BH background, we predict the three CFFs from the interference term and treat the total DVCS term as a parameter that C-VAIM can fit. This makes the extraction much simpler, in theory, as the C-VAIM no longer has to disentangle all eight CFFs from the DVCS term. Fig. 33 ( Top) shows the results of the reconstructed cross sections using the whole 8 CFFs, 6 CFFs and 3 CFFs. Fig. 33 (Bottom) shows the histogram of the extracted 8 CFFs, 6 CFFs and 3 CFFs where the CFFs are overlapping in all three scenarios. We further analyze $\chi^2$ of the reconstructed cross sections of our predicted CFFs on the 3 CFFs, 6 CFFs and 8 CFFs, as shown in table. 3, where the number of cross section points are 24. We can see that the $\chi^2$ values for 6 CFFs and 8 CFFs have approximately similar values with the least accurate prediction on $x_{bj}$ =0.381, $t$= -0.371, $Q^2$= 1.999.

**Table 3.** Chi-squared on the DVCS experimental data.

| $\chi^2/\sigma$ $npts$ | | | |
|---|---|---|---|
| Kinematics | 3 CFFs | 6 CFFs | 8 CFFs |
| $x_{bj}$ =0.343, $t$= -0.172, $Q^2$= 1.820 | 0.0011 | 0.0010 | 0.0011 |
| $x_{bj}$ =0.368, $t$= -0.232, $Q^2$= 1.933 | 0.0236 | 0.0066 | 0.0071 |
| $x_{bj}$ =0.375, $t$= -0.278, $Q^2$= 1.964 | 0.0356 | 0.0099 | 0.0113 |
| $x_{bj}$ =0.379, $t$= -0.323, $Q^2$= 1.986 | 0.0082 | 0.0029 | 0.0027 |
| $x_{bj}$ =0.381, $t$= -0.371, $Q^2$= 1.999 | 0.0459 | 0.0267 | 0.0273 |

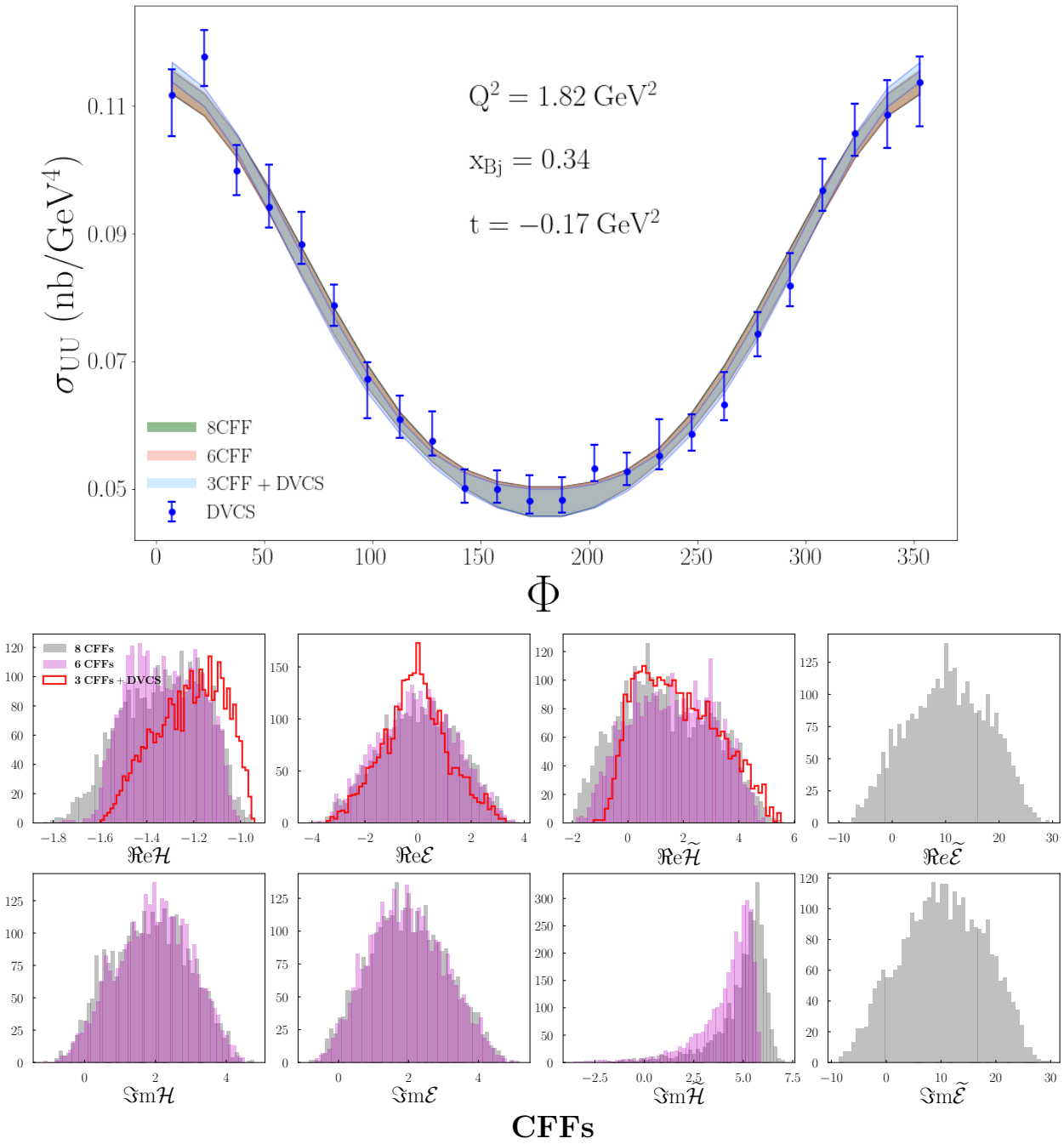



**Fig. 33.** (Top) comparison of reconstructed cross sections of 3 CFFs, 6 CFFs and 8 CFFs, (Bottom) C-VAIM predicted 3 CFFs, 6 CFFs and 8 CFFs.

## 6.7 LATENT LAYER ANALYSIS

The latent $z$ in the latent layer is the key component of the VAIM as it attempts to learn the lost information in forward mapping and converts the ill-posed inverse problem into a well-posed regression problem in the backward mapping. In this section, we analyze the latent space and its association with the CFFs using dimensionality reduction techniques. This allows us to explore and analyze the learned patterns which reflects the correlation between physics quantities, as well helps us to the understand the underlying physics.

We start with principle component analysis (PCA), a linear dimensionality reduction algorithm. After our VAIM is trained, we project the latent variables onto the first two principle components. The distribution of CFFs samples is visualized on the latent space in fig. 34 for the first two PCA components (PCA 1, PCA 2). We can see that there are clear patterns in respect to some of CFFs, in specific, $\mathrm{Re}\mathcal{E}$, $\mathrm{Re}\widetilde{\mathcal{H}}$, $\mathrm{Re}\widetilde{\mathcal{E}}$, $\mathrm{Im}\mathcal{E}$ and $\mathrm{Im}\mathcal{H}$ . This may indicate the latent variable is learning the hidden information on those CFFs with patterns.

We also analyze the latent space using t-Distributed Stochastic Neighbor Embedding (t-SNE) [104], a non linear algorithm. In the same way, we project the latent variables onto the two t-SNE components as demonstrated in fig . 35 with the distribution of the 8 CFFs is plotted on the latent space. One can find notable patterns in some of the CFFs like $\mathrm{Re}\mathcal{E}$, $\mathrm{Re}\widetilde{\mathcal{E}}$, $\mathrm{Im}\mathcal{H}$, $\mathrm{Im}\mathcal{E}$ and $\mathrm{Im}\widetilde{\mathcal{E}}$. Where the pattern seems apparent in some of the CFFs, the pattern of the two CFFs $\mathrm{Re}\mathcal{H}$ and $\mathrm{Im}\widetilde{\mathcal{H}}$ are noisy using both PCA and t-SNE.

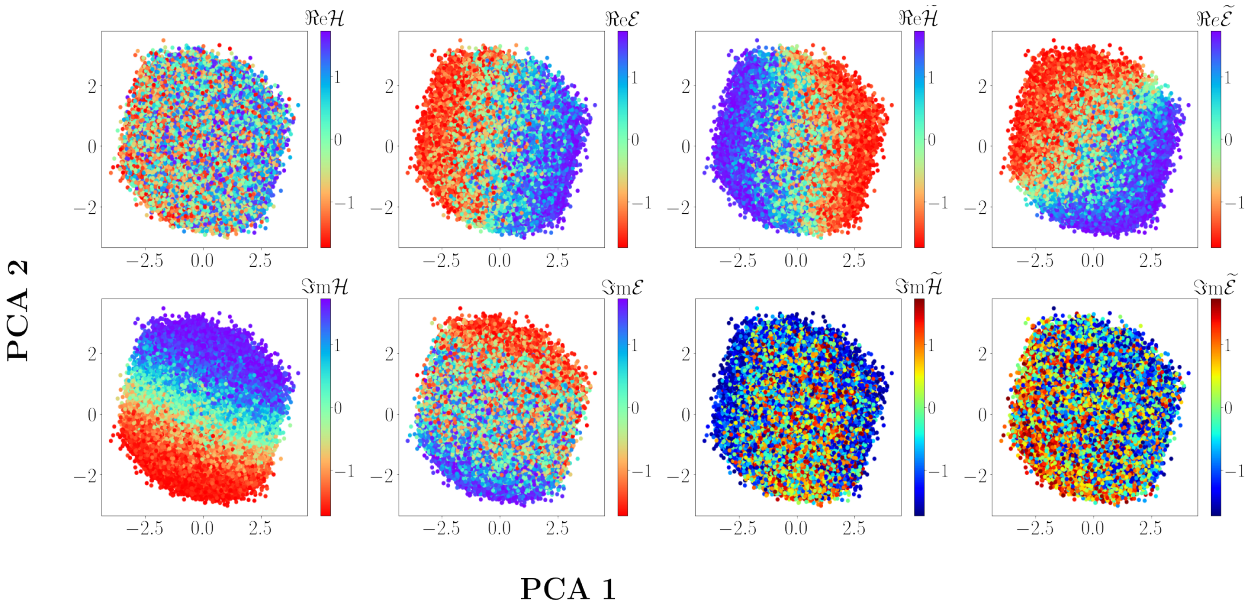



**Fig. 34.** The distribution of the 8 CFFs on latent space projected on the first two PCA.

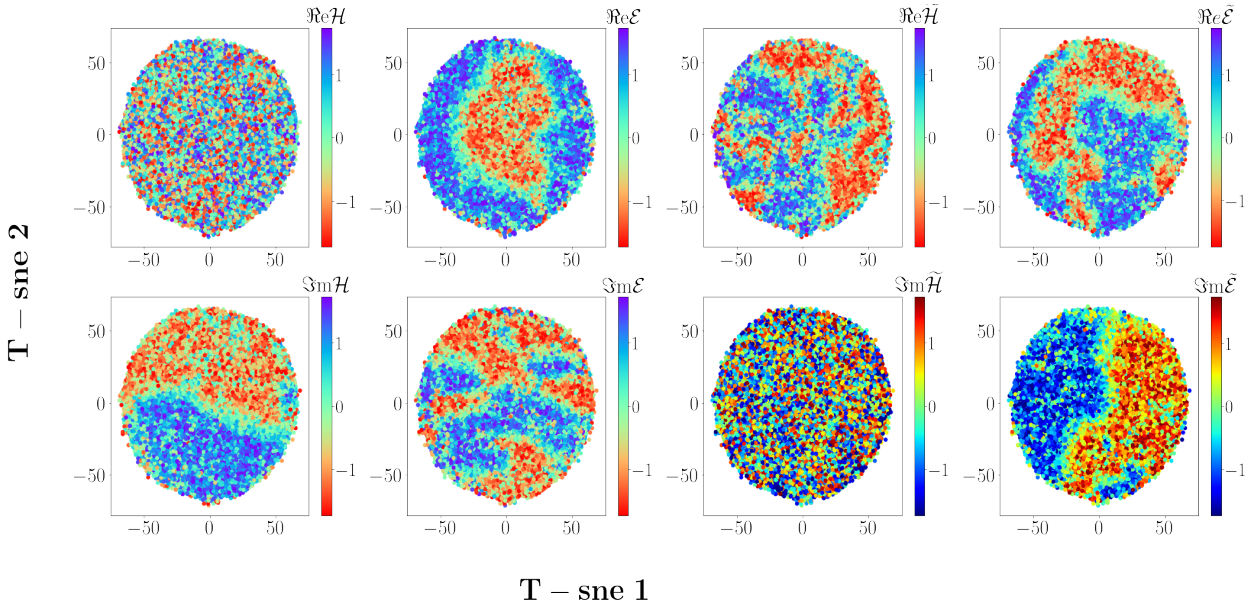



**Fig. 35.** The distribution of the 8 CFFs on latent space projected using two t-SNE components.

## 6.8 CONCLUSION

In this chapter, we present our work in solving the inverse problem of extracting the CFFs. We first adopt VAIM, discussed in chapter. 4 to extract CFFs on a specific kinematics values ($x_b$ $j$= 0.343, $t$= -0.172,$Q^2$ = 1.82 GeV$^2$, and $E_b$ = 5.75 GeV). We then extend VAIM to C-VAIM by conditioning our networks using the kinematics values. C-VAIM is composed of a forward mapper NN to learn mapping from CFFs and kinematics space to the cross section space and a backward mapper NN to reconstruct the CFFs from cross sections. A latent layer is incorporated between the forward and backward mappers to approximate the posterior CFFs distribution with respect to the given cross sections and kinematics. Once C-VAIM is trained, sampling the latent layer leads to reconstruct the lost information and determine the corresponding CFFs. Testing C-VAIM on several experimental DVCS examples, C-VAIM reconstructs the CFFs with potentially multiple solutions.

The focus of our future research is to incorporate all experimental DVCS data on all available kinematics points in the training of the C-VAIM framework. As well incorporating new simulated data from EIC. We will also incorporate physics knowledge to our inverse mapper such as positivity constraints [95] and dispersion relations [105, 106].

# CHAPTER 7

# PHYSICS CONSTRAINED NEURAL NETWORKS (PCNNS)

In this chapter, we first present our work developing Physics Constraints Neural Networks (PCNNs), published in [97], by incorporating physics laws to DNNs. PCNNs is applied on the forward problem of mapping the kinematics to the observed cross sections. We next describe the PCNNs architecture, hyperparameters selection, data augmentation and our methodology of incorporating physics knowledge. Finally, we show the capability of PCNNs testing on both DVCS cross sections and Bethe-Heitler (BH) process.

## 7.1 INTRODUCTION

DNNs models are adopted as a promising tool in many scientific applications including the process that are computationally infeasible or not completely understood. However, DNNs have usually limited success in scientific domains such as physics applications because training DNNs models typically require a large size of data that might not always be available or possible to obtain. As an alternative, DNNs models can be trained using additional knowledge gained by enforcing some physics laws for better accuracy and generalization [107]. In addition, purely data-driven models may perfectly fit observations, but predictions might be inconsistent, as a result of extrapolation or observational biases that may lead to poor generalization performance. Therefore, it is important to integrate fundamental physical rules and domain knowledge to machine learning models which provide informative priors. A recent example that follows this new learning technique is the family of "physics-informed neural networks" (PINNs) introduced in [108]. The major motivation for implementing these methods is that such prior knowledge or constraints can reduce the parameter space of the neural network to physics feasible space and lead to more interpretable machine learning methods that are robust in the presence of imperfect data such as missing values or outliers, and can provide precise and physically consistent predictions, even for generalization tasks.

In this chapter, we develop PCNNs integrating DNNs and the physics constraints, in specific, angular symmetries constraints. We use the BH process to test PCNNs and, in particular, their ability to generalize information extracted from data. We then test PCNNs on the full cross section and compare the results to analytic model calculations.

## 7.2 PCNNS ARCHITECTURE

Often when constructing a machine learning model, there will be presented several design choices to define the architecture as we do not know in advance what the optimal architecture should be for a given model. Basically, training machine learning models involves two types of parameters. First, trainable parameters that are learned by the algorithm through the learning process such as the weights and biases of a neural network. Second, the hyperparameters that the learning algorithm will utilize to learn the optimal parameters that accurately map the input features to the targets. The hyperparameters such as the number of layers, number of neurons, and the learning rate, need to be manually tuned before launching the training process. The hyperparameters tuning is an essential process of searching for an optimal set of hyperparameters that can leverage the highest performance of our model. Finding the optimal hyperparameters manually is a challenging task as there can be numerous hyperparameters. Therefore, we adopt Keras tuner [99], an open-source library to conveniently define our architecture and perform hyperparameter tuning. The keras tuner workflow as follows, first, a tuner is specified to determine the hyperparameter combinations that should be tested. Second, the library search algorithm such as Bayesian optimization executes the iteration loop to evaluate a particular number of hyperparameter combinations. Third, evaluation is performed by calculating the loss of the trained model on a held-out validation set. Finally, the optimal hyperparameter combination in terms of validation loss is retrieved to be tested on a held-out test set. The optimal hyperparameter is utilized as the final hyperparameter for our model architecture illustrated in Figure 36. Our optimum network architecture composed of an input layer whose nodes denote the array of the kinematic [$x_{bj}$, $t$, $Q^2$, $E_b$, $\Phi$], followed by three fully-connected layers with 1024 neurons activated by a Leaky ReLU function.The output layer represents our target cross section. The network is regularized by an L2-norm penalty and a dropout rate of 0.2 to prevent overfitting. Based on the hyperparameter search algorithm, a shallow and wider network can have better performance than a deeper network with less number of neurons.

Our model is optimized using a joint of three loss functions: The first and basic loss function is the mean squared error (MSE) to minimize the error between the predicted and observed cross sections. The second loss is to ensure that our predicted cross sections are symmetric. Lastly, a loss function to ensure the start and the end of the predicted data on a range of $\phi$ are matching.

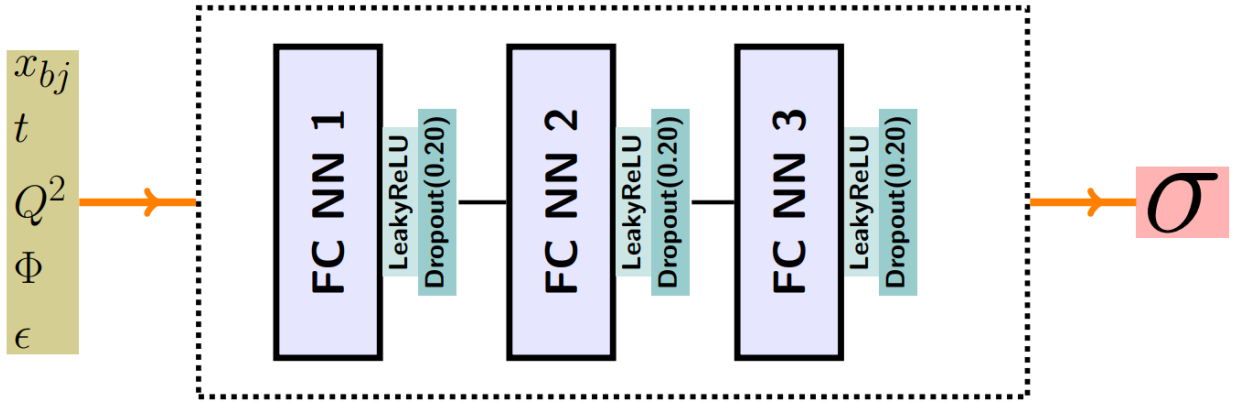



**Fig. 36.** Neural network architecture for DVCS data that takes in input variables $x_{Bj}, t, Q^2, \phi, \varepsilon$ and outputs the cross section. Each fully connected layer in this network architecture is followed by a nonlinear activation function (Leaky RELU) and a dropout mask with $p = 0.2$ [97].

**Physics Constraints –Parity Invariance**

The symmetry constraints on the cross section can be applied because there is no *"phase changing"* term that can enter the unpolarized cross section through physics constraints of parity. Any parity violating terms related to beyond the standard model physics we treat as part of the extracted error as compared to the size of the experimental statistical error. Studies for beyond the standard model physics that can be extracted from DVCS data we leave to a future study. We then deem the asymmetry in the $\phi$ dependence in the unpolarized cross section as due to the systematic uncertainties in the experimental data taking. With many points in $\phi$ the data is more sensitive to these type of systematic errors. We also note that taking many data points in $\phi$ is based on a harmonics analysis for extracting CFFs which we are not utilizing for our extraction method. It has been demonstrated in [109] that there are alternative approaches for extracting quantities related to angular momentum beyond the harmonics description.

To incorporate angular symmetries into the machine learning model, we use additional loss function terms

$$\|f(x_{Bj},t,Q^2,\phi,\varepsilon) - f(x_{Bj},t,Q^2,-\phi,\varepsilon)\|$$

and

$$\|f(x_{Bj},t,Q^2,\phi,\varepsilon) - f(x_{Bj},t,Q^2,\pi-\phi,\varepsilon)\|,$$

measuring the unpolarized angular symmetry and polarized angular symmetry, respectively.

The incorporation of the additional loss function term is an example of the physics constraints

that we propose to build into the architecture of the neural networks. It has been shown that such neural networks [110, 111] can improve the performance and generalizability of neural network predictions. Further physics inputs such as input from LQCD calculations, can additionally be built into neural network architectures at different steps of the analysis.

### 7.3 DVCS DATA

DVCS cross section (illustrated in [112] and [113]) is a function of four independent kinematic variables: the angle between lepton and hadron planes $\phi$, the scale $Q^2$, the skewness $\xi$ that is related to Bjorken $x$ and t; GPDs for all quark flavors appear in the cross section embedded in convolutions over x known as Compton form factors (CFFs), with complex kernels depending on ($x$, $\xi$, $Q^2$). Each record of the dataset has [$x_{bj}$, $t$, $Q^2$, $E_b$, $\phi$,$L$, $\sigma$,$\Delta\sigma$,$\delta\sigma$] where L is a label for polarization of beam and target - 1 is unpolarized beam/unpolarized target, 2 is polarized beam/unpolarized target; $\sigma$is the cross section value; $\Delta\sigma$ is the statistical error; and $\delta\sigma$ is the systematic error.

Each input to our model consists of [$x_{bj}$, $t$, $Q^2$, $E_b$, $\phi$] obtained from Jefferson Lab Halls A and B 6 GeV and 12 GeV experimental runs. For each kinematic bin of [$x_{bj}$, $t$, $Q^2$, $E_b$] there are 25 points in $\phi$ from 0-360. The total available data is 3,862 unpolarized and 3,884 polarized data points with kinematics range shown in Fig 37.

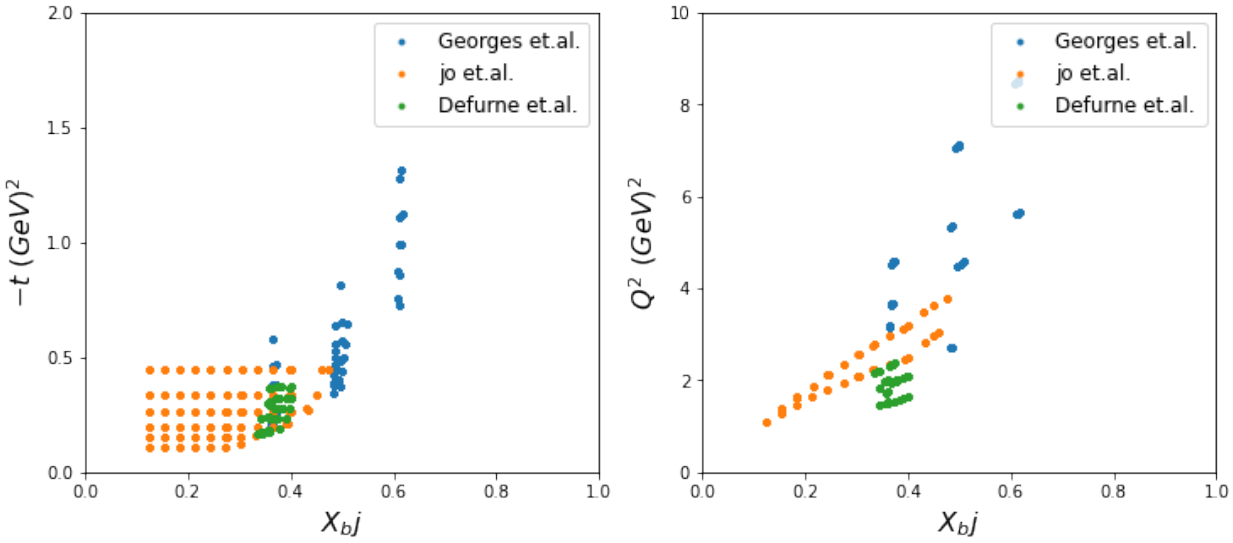



**Fig. 37.** Kinematic region in $x_{bj}$ , $t$, and $Q^2$ for the DVCS data from Jefferson Lab Halls A and B 6 GeV and 12 GeV runs.

## 7.4 DATA AUGMENTATION

Deep learning networks have made extraordinary progress in many scientific applications. However, these networks perform more effectively as there is a large amount of data available to train on. For example, text-based models have performed significantly better because of the release of a trillion-word corpus by Google [114]. The reason is that training on large datasets can avoid overfitting where a network learns a function with very high variance such as to perfectly fit the training data but do not generalize well on the unseen data. There are several proposed methods to reduce the problem of overfitting such as dropout [115], batch normalization [116], transfer learning [117, 118] and pretraining [119]. In addition to these methods is the data augmentation that approaches overfitting from the root of the problem, the training dataset. This is performed under the assumption that additional data and information can be obtained from the original dataset through augmentations. Data Augmentation involves a set of techniques [120] that increase the quality and size of training datasets so that more powerful deep learning models can be built.

We use data augmentation to increase the size of our DVCS data by adding slightly modified copies of our existing data. Since we have only 3,862 unpolarized and 3,884 polarized data points, data augmentation seems to be essential to apply in order to improve the model accuracy. Each DVCS data point includes a mean and a standard error indicating its uncertainty. The standard errors varies across data points. We increase the size of the data 10 times for each the unpolarized and polarized data by sampling the cross section using the statistical error $\Delta\sigma$. Data augmentation is applied on the unpolarized and polarized data independently as shown in Figure 39. There are two main advantages of data argumentation. Firstly, more data samples become available for training. Secondly and more importantly, the trained neural network is informed with the uncertainty of the training samples, where more model flexibility is allowed on the sample points with larger standard errors and vice versa. As a result, the neural network trained with augmented samples yield better accuracy and generalization.

Including the information contained in the experimental standard errors into the architecture of the network through data augmentation. We are treating the experimental errors as additional sources of physics information, errors are not a nuisance but rather a source of information. This is another example of building physics into the architecture of the network and utilizes standard neural network practices.

To show the impact of applying data augmentation to our data, we train two separate models with identical architecture once with applying data augmentation and once with no data augmentation. Figure 38 illustrates our comparison results using dropout method to determine the model uncertainty. With utilizing data augmentation, the prediction of the polarized and un polarized

cross sections becomes more accurate.

## 7.5 RESULTS

In this section we show the results of our study of incorporating physics into the architecture of a deep learning model trained to predict DVCS cross sections and how this improves both performance and generalization capabilities. We test our model on the exactly calculated Bethe-Heitler process in order to calibrate our analysis. Our results on the predicted cross section in an extrapolated region of kinematics outside of the range covered in experiments are shown in Figure 40.

A single cross section data point represents a point in a multi-dimensional kinematic phase space. In the context of local extractions, this point contains all of the information one can extract. However, it is necessary for a global analysis to study trends in this higher-dimensional space. In this sense, a group of $N$ data points represents more than $N$ disjoint pieces of information but rather a region of trends and patterns that can inform predictions in kinematic regions outside of the data. This includes correlations between data points that are not taken into account/cannot be taken into account in point-by-point extractions.

Neural networks learn complex representations of data while extracting trends that can be generalized to regions beyond the data. Generalization represents a unique opportunity to squeeze as much information from DVCS experiments as possible while quantifying learned uncertainties. These generalization capabilities will allow us to inform experiments of worthwhile, high impact measurements in the extraction of Compton form factors.

We test our trained model using pseudo-data calculated from the BH cross section. The BH cross section represents a significant portion of the full cross section at current experimental beam energies. It also has the added benefit in that it is exactly calculable in QED up to the parametrization of the elastic form factors, which for momentum transfers at current JLab DVCS experiments, are known to high precision. This means that the BH cross section is a good testing ground to benchmark our model's performance, especially outside of the range of current experimental data. We show the performance of the predicted BH as compared to the theory prediction both in the range of the data and generalizing outside of the range of the data in Figure 41. We plot the expected value versus the predicted, meaning that if the model was performing perfectly we would see a straight line. Any non-linear behavior demonstrates predicted values that are not well understood in our model.

In Figure 42 we investigate the generalization capabilities of the deep learning model as compared to the exactly calculated Bethe-Heitler process and a calculation of the DVCS process. In the

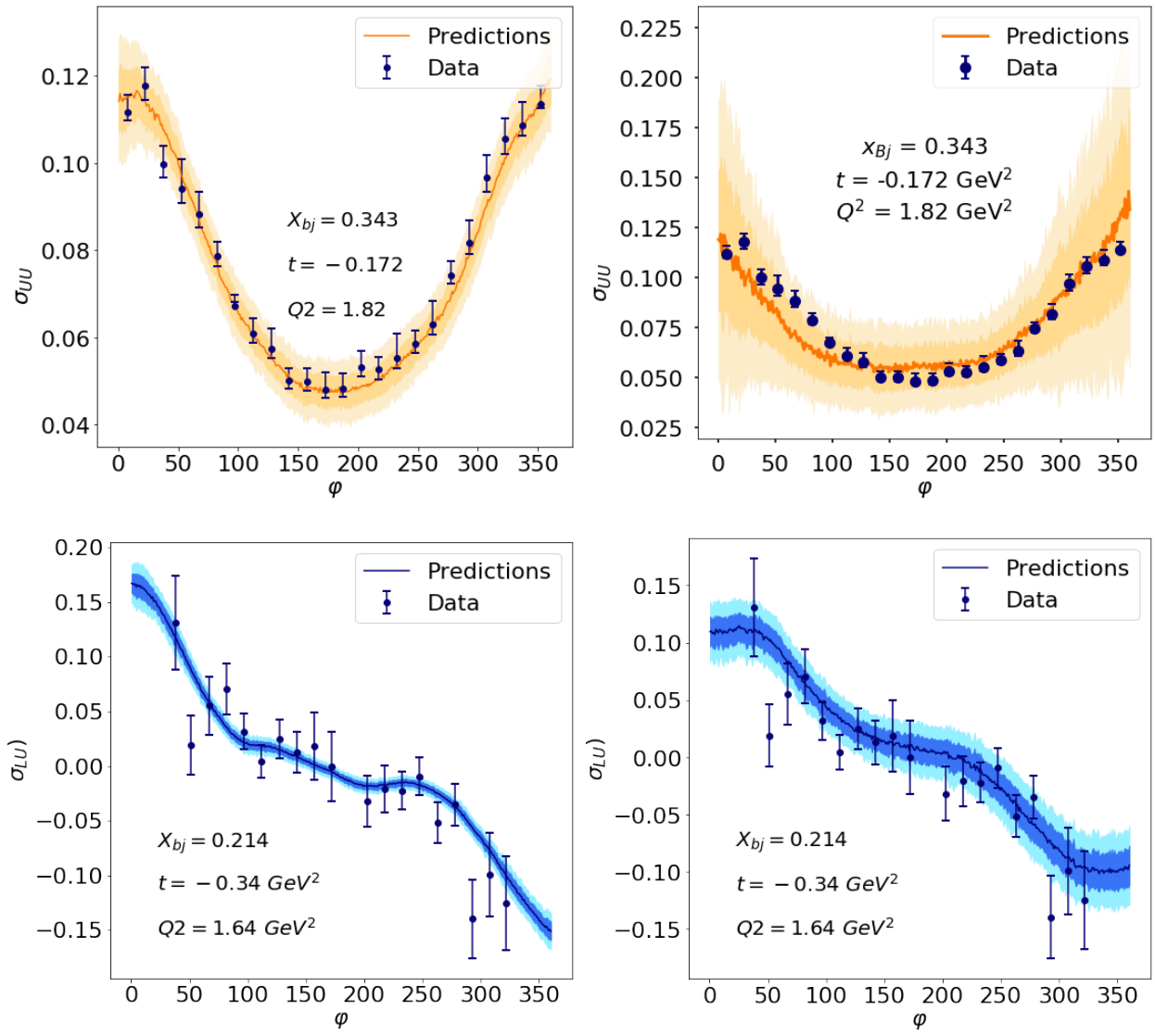



**Fig. 38.** The predictions of $\sigma_{LU}$(top row), and $\sigma_{UU}$ (bottom row) in unseen kinematic bins. Left column shows the predictions using data augmentation and right column shows the predictions with no data augmentation. The bandwidth indicates two standard deviations of 1000 predictions

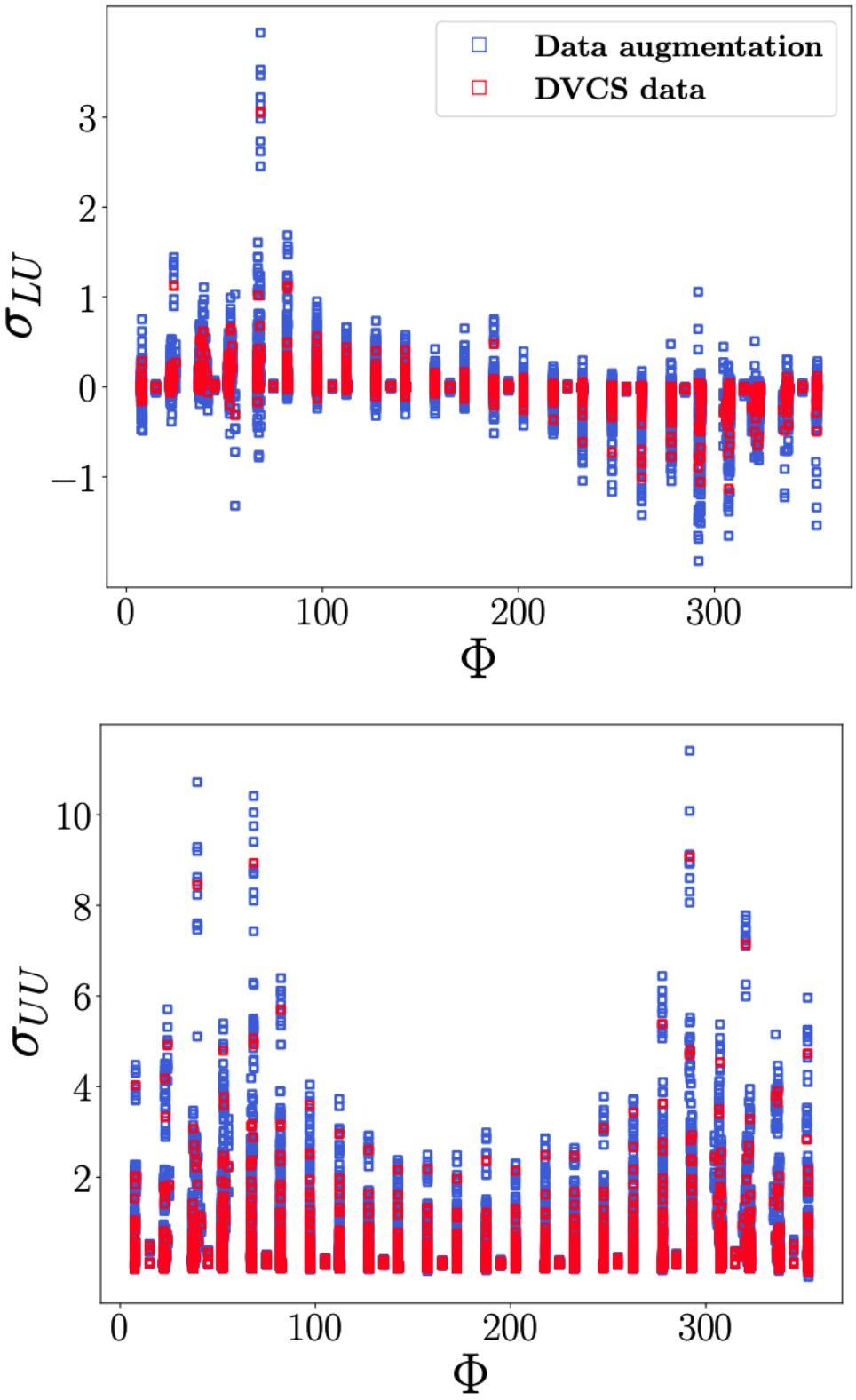



**Fig. 39.** Data augmentation on unpolarized (top) and polarized DVCS data(bottom) [97].

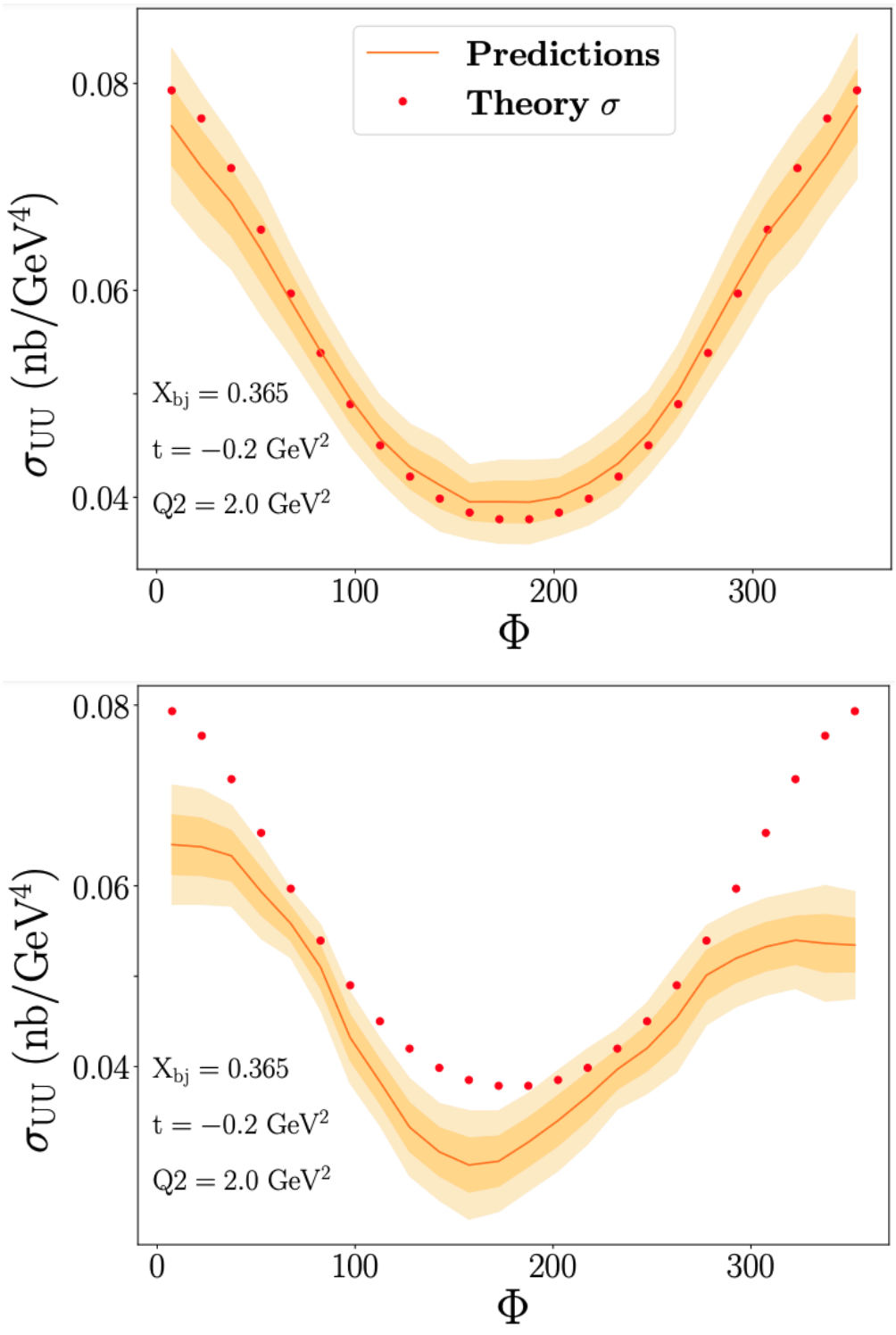



**Fig. 40.** BH extrapolation on kinematics outside of the range covered in experiment at the kinematic point $x_{Bj} = 0.365$, $t = -0.2$ GeV$^2$, $Q^2 = 2$ GeV$^2$, and $E_b = 5.75$ GeV. (*Top*)ML Model with Angle Symmetric Constraints (*Bottom*) ML Model without Angle Symmetric Constraints [97].

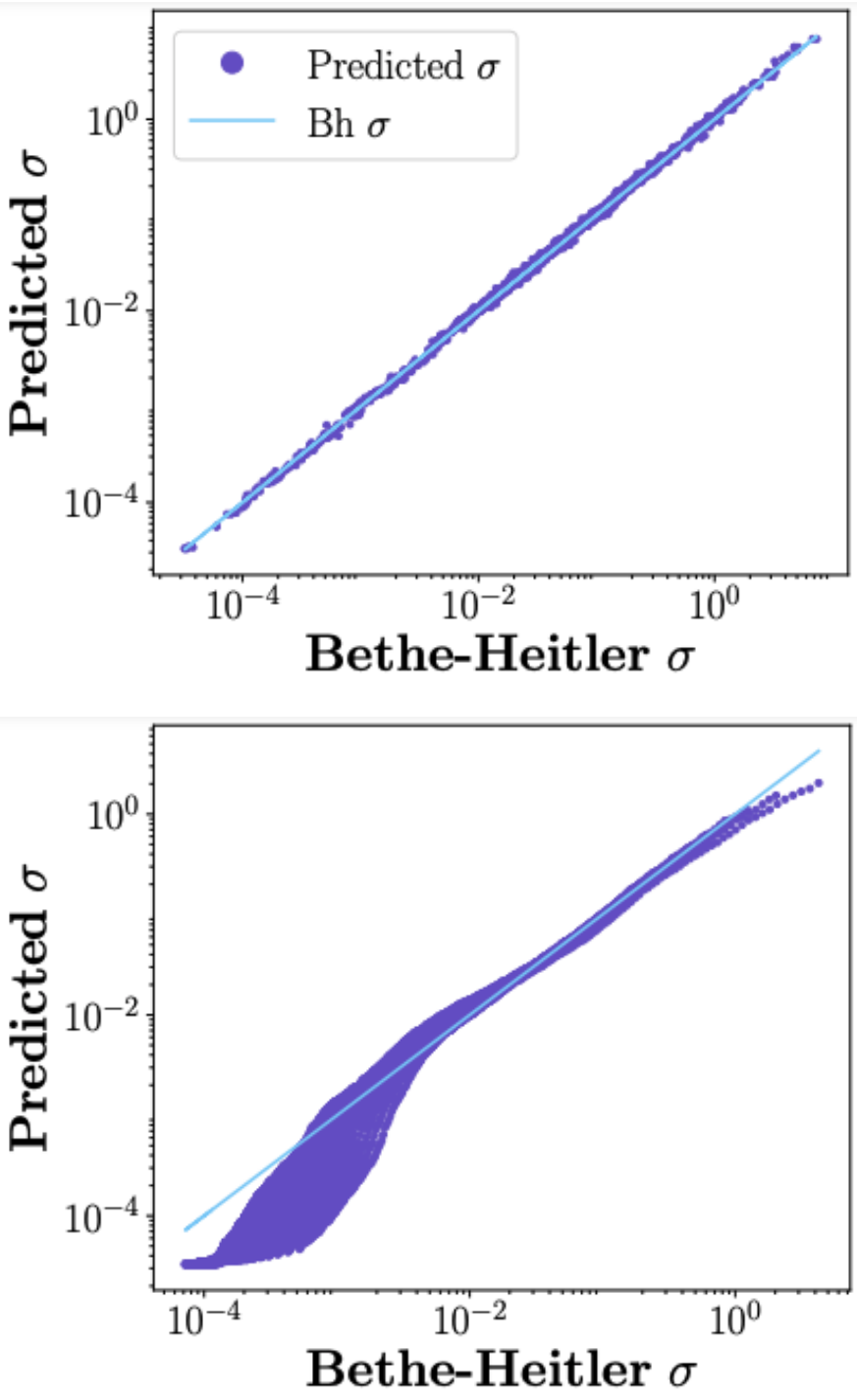



**Fig. 41.** Result of the DNN predicted BH cross section as compared to the theoretical calculation both within the test set (Top) and outside of the range data (bottom) [97].

slice of kinematic phase space shown in the figure ($Q^2 = 2$ GeV$^2$, $E_b = 6$ GeV), the experimental data lies in a small box defined between $\xi = 0.2 - 0.3$ and $-t = 0.17 - 0.4$ GeV$^2$ [121]. For the top figure, we take the kinematic points in $x_{Bj}, t, Q^2, E_b$, and $\phi$ and create pseudo-data for this region. We then train our neural network on this data and use it to predict in the full kinematic phase space that is shown in the figure. We compare the results of the neural network predictions against the calculated BH cross section at a given azimuthal angle $\phi = 90^\circ$ using the metric of absolute percentage error (APE) which is given as:

$$APE = \frac{|y - \sigma|}{\sigma}$$

In the bottom figure we use the exact same metric; however, we are comparing to a calculation of the DVCS cross section using a parametrization of the CFFs given in [122, 123, 124]. It is interesting to note that the BH cross section outperforms the DVCS cross section in terms of learned generalization across the kinematic phase space. Also that there are significant trends that the deep learning model can learn when given small amount of physics information.

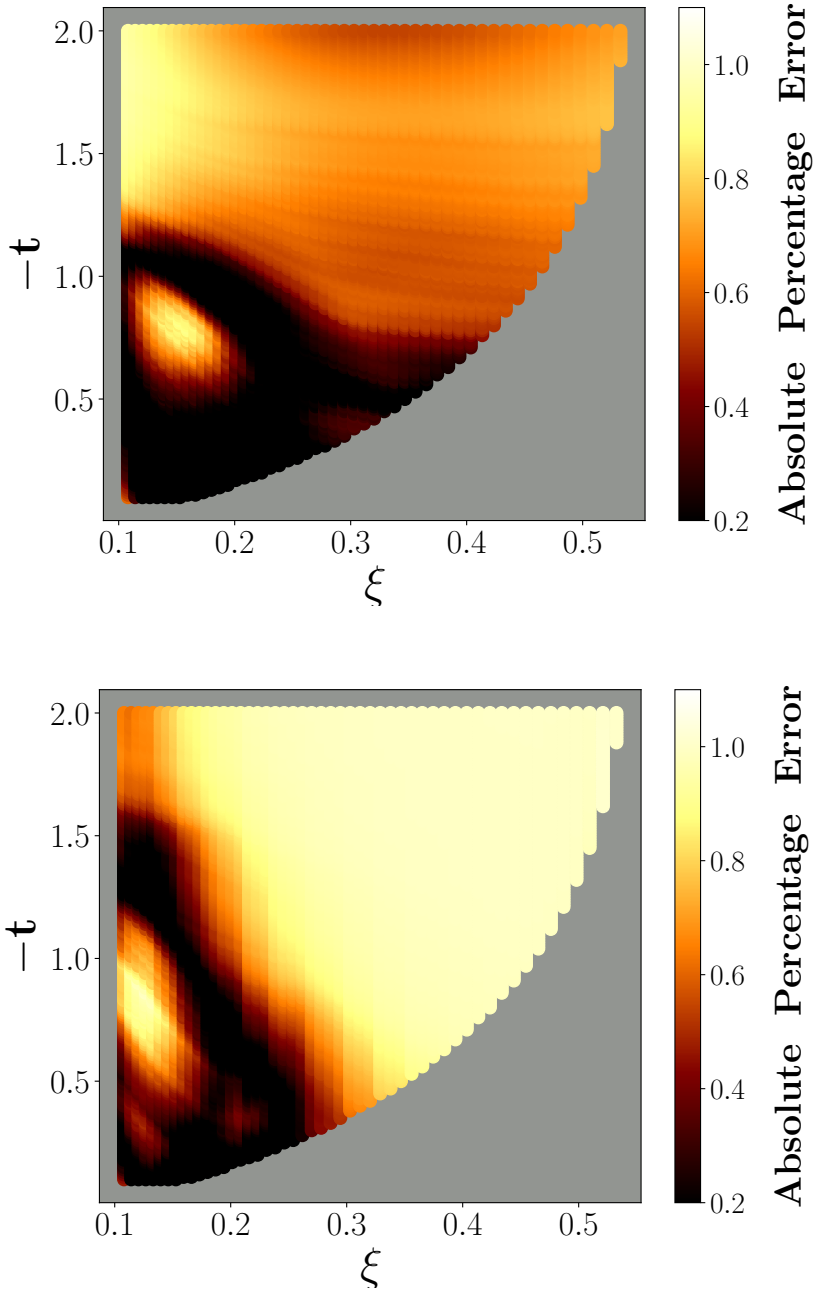



**Fig. 42.** (Top) The BH data, (Bottom) DVCS data errors at $\Phi = 90$ [97].

Since there is a statistical error $R_{\theta_t}$ associated with each data point $\theta_t$ in the cross section and these errors vary across data points, we incorporate this information to our MSE loss function, as

following:

$$\mathscr{L} = \frac{1}{n}\sum_{\theta_t} \frac{(y - f_{\theta_t}(x))^2}{R_{\theta_t}}$$

To balance the sensitivity with respect to the varying error in the experimental data. Where $n$ is the number of data samples. Similar to the results on fig. 42, we recalculate the APE in figure 43 after incorporating the statistical errors training on the BH and DVCS cross sections. While the overall error decreased in the BH data, one can find that in the DVCS data error is reduced in some regions on the small $x_{bj}$ of the cross sections compared to the DVCS results shown in fig. 42 (bottom).

## 7.6 CONCLUSION

In this chapter, we demonstrate the effectiveness of encoding physics knowledge to machine learning models. Our ultimate goal is to first analyze the impact of incorporating physics rules to a forward mapper and second to use this analysis for future work in incorporating physics knowledge to our inverse mappers. We develop PCNNs, a deep neural network architecture that satisfies some physics rules to predict the DVCS cross sections. We incorporate the angular symmetries into our PCNNs architecture by utilizing an additional loss function that ensures the symmetry constraints on the cross sections. Testing on DVCS cross sections and BH process, PCNNs show better performance compared to a typical neural networks architecture.

We utilize data augmentation to enlarge the DVCS training data by taking the advantage of the statistical error being available. We generate softly modified copies of the given unpolarized and polarized data by sampling the cross sections using the experimental error. Our results testing on the unpolarized and polarized test examples indicate that training with our data augmentation technique leads to better generalization and accuracy.

In our future work, we will extend this work to include further physics constraints from Lattice QCD. We also plan to incorporate such a physics knowledge to improve the capability of our inverse mappers in the presence of the physics challenges such as the lack of training data.

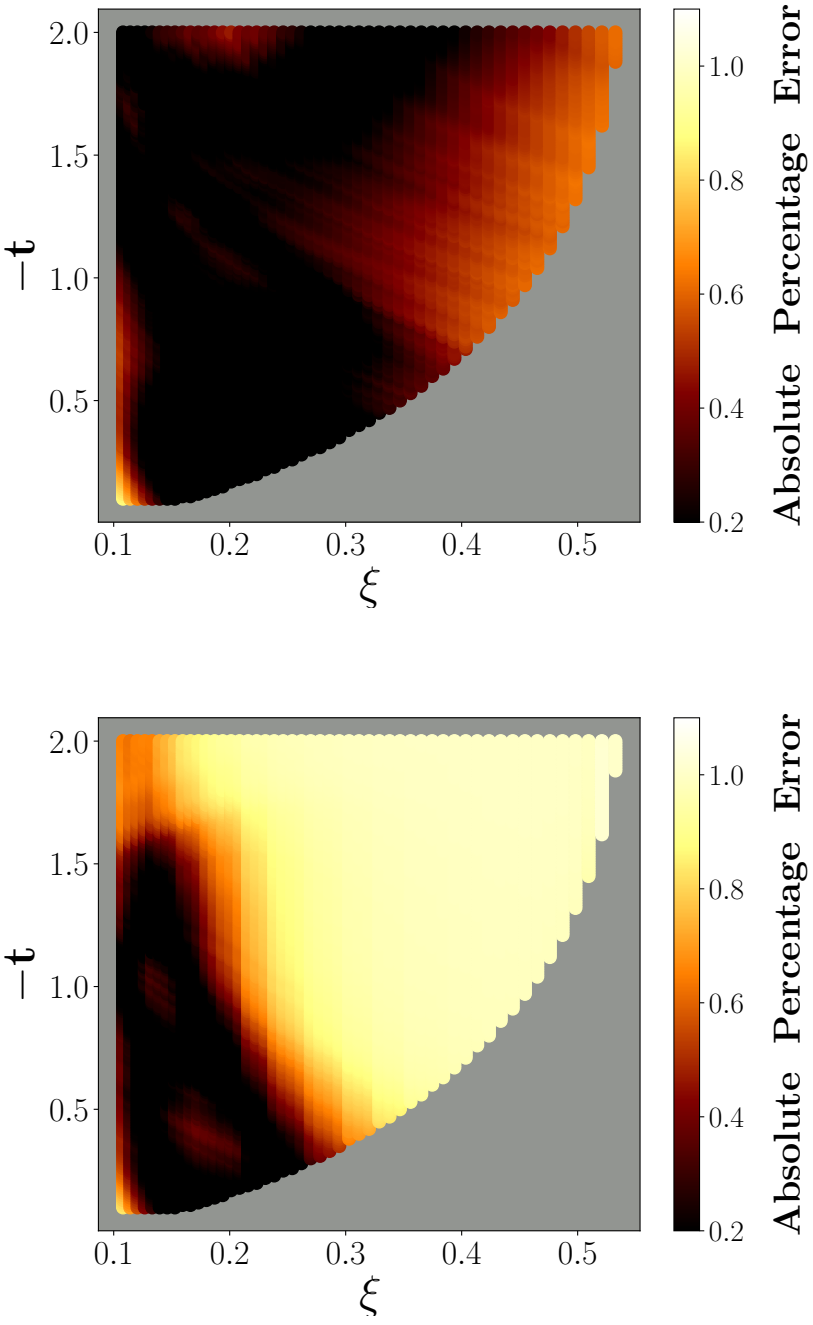



**Fig. 43.** (Top) The BH data , (Bottom) DVCS errors after incorporating the statistical error to the loss function at $\Phi = 90^\circ$ [97].

# CHAPTER 8

# CONTRIBUTION, CONCLUSIONS, AND FUTURE WORK

In this chapter, we summarize the major contributions, conclusions and future work of developing machine learning models for QCD global analysis.

## 8.1 CONTRIBUTIONS

Our contributions include many components described in chapters 4, 5, 6, and 7. The main contributions for each chapter are summarized in the following subsections.

### 8.1.1 VAIM

We develop the Variational Autoencoder Inverse Mappers (VAIM) focusing on learning the solution patterns in an end-to-end manner. VAIM does not depend on the Gaussian assumption, does not need a sophisticated prior and can successfully retrieve the multiple solutions of an inverse problem. The latent layer in VAIM plays a significant role in learning the posterior parameter distribution without an encoded form of the distribution. Thus, VAIM can be utilized for many inverse problems where solution distributions are non-Gaussian. Our work developing VAIM is published in [76] and tackles RQ 1: Can we develop an end-to-end machine learning framework to solve the inverse problems in PDF?.

### 8.1.2 PC-VAIM

We develop the Point-Cloud Based Inverse Mappers (PC-VAIM) based on flexible, unordered data points represented as a point cloud, to solve the inverse problem in QCD global analysis. We extend VAIM framework to PC-VAIM, in order to address the input flexibility problem. We incorporate the permutation invariant PointNet-based architecture to our backward mapper to allow PC-VAIM to handle the ill-defined, varying observable inputs. This design enables the PC-VAIM framework to reconstruct the parameters in QCD global analysis independent of specific kinematics bin setups. This work is published in [90] and answer the RQ 2: Can we develop a kinematics independent inverse mapper to address the irregular observable space?

### 8.1.3 C-VAIM

We extend VAIM to the conditional VAIM (C-VAIM) to model the inverse problem of ex-

tracting the CFFs from polarization observables and conditioned by the kinematics values. This provides the flexibility to the backward mappers to generate CFFs that correspond to a certain kinematics region without the need to retrain the model. This work is in preparation for publication and addresses the RQ 3: Can we extract the GPD Compton form factors?

### 8.1.4 PCNNs

We develop Physics Constrained Neural Networks (PCNNs) that satisfy the angular symmetries constraint to accurately predict the DVCS cross sections. We found that PCNNs yields better generalization than typical NNs that are implemented without physics knowledge. In this work, we perform data augmentation to enlarge the DVCS data using the DVCS statistical error. This work is submitted for publication and available in arXiv [125].

## 8.2 CONCLUSIONS

ML algorithms have emerged as a master tool for the inverse problems in different scientific applications. Our vision in this dissertation is to build the global QCD analysis tools using ML techniques to study the quantum correlation functions characterizing the internal structure of the nucleon. This can provide a simple and effective way to perform a global QCD analysis interactively, as all time-consuming work of generating the samples and training need to be computed only once.

In this work, we present VAIM to address the ambiguity issue associated with the inverse problems. Testing on several scenarios, VAIM can effectively reconstruct the multiple solutions for the inverse problems with different solution patterns. We also demonstrate the feasibility of applying VAIM in connecting unpolarized cross sections into the parameter space of the PDFs. We then extend VAIM to PC-VAIM to enable more flexible and varying observable space. We incorporated a permutation invariant neural network framework to handle the observables in unstructured and unordered point cloud representations. We show the effectiveness of the PC-VAIM on retrieving the PDFs parameters in a simulated DIS data. Furthermore, we develop C-VAIM to tackle the inverse problem of extracting the CFFs from polarization observables, which is the cleanest channel to access GPD from a theoretical point of view. We apply C-VAIM to study the effect of potential multiple solutions in DVCS data. To validate our extracted CFFs, we use the theory computations to reconstruct the corresponding cross section from our predicted CFFs and compared it with the true cross sections. Our computed cross sections demonstrates good agreement with the true cross sections. We develop PCNNs to the polarized and unpolarized DVCS cross sections by incorporating physics laws. We also use the Bethe-Heitler process to test PCNNs and in specific, their ability

to generalize information extracted from data. We found that incorporating physics constraints into the ML models is able to significantly improve the interpolation as well as generalization capabilities of the deep learning model.

Finally, this research provides valuable tools for QCD studies of hadron structure by making use of state-of-the-art machine learning technologies, and will pave the way towards the eventual extraction of more complicated QCFs, in specific the full GPDs distribution. Future work will be pursued using the developed inverse mappers to better understand the challenges associated with QCD global analysis. While the inverse problems in QCD are our main target in this dissertation, our inverse mappers can be easily utilized to other domains.

## 8.3 FUTURE WORK

### 8.3.1 Active Learning

In a wide variety of learning problems including our application of interest– nuclear physics, collecting sufficient data for modeling purposes forms a great challenge. In most situations, data is expensive to obtain, requires manpower and extensive resources. One of the attractive techniques to overcome this challenge is Active Learning (AL). AL is an essential machine learning subfield that is used to enable the learning algorithms to select the data that will be trained on. The key objective of AL is to reduce the number of training samples required to train the model, or substantially, to improve the precision of the prediction to obtain the same prediction quality with a smaller dataset. Several methods to AL are proposed in the literature [126, 127, 128, 129] that mainly focus on the classification problems [128, 129, 130], and with limited efforts to the regression problems [131, 132]. AL is basically categorized into two types: population based and pool-based. In population based AL, the distribution of test samples is known, and training samples at any desired locations can be obtained. While in pool-based AL, the test samples distribution is unknown but only samples from the test input distribution are given.

We first plan to utilize pool-based AL to gradually increase the size of the training data by sampling on a certain test example. The proposed AL method consists of an iterative process which is described as following. First, AL algorithm has access to a training set $T^1$ with input points $x$ and the corresponding set of the labels $y$ within specific range. Second, after training our model on this data set $T^1$, we test on a given test example $G^1$ and obtain the predictions $K^1$. Third, we sample new training data $T^2$ based on the range of our predictions of $K^1$. Then, we combine the training data $T^1$, $T^2$, and retrain our model. We test again on the test example $G^1$ and obtain our prediction $K^2$ for this second level of training. Then, we sample again more training data $T^3$

based on the data obtained in $K^2$, combine $T^3$ to the data from the previous levels of training ($T^1$, $T^2$) and retrain our model. This iterative process can be terminated once we have no improvement on our prediction on $G^1$.

We study the feasibility of this technique on the inverse function of mapping QCF parameters to observables in a QCD analysis of nucleon structure using the synthetic DIS problem described in sec. 5.2.1, with excluding the normalization terms $N_u$ and $N_d$. We employ VAIM to produce the prediction of our control parameters $\{\alpha_u = 0.5, \beta_u = 2.5, \alpha_d = 0.1, \beta_d = 3.0\}$, across multi-levels of training. We generate $1,000$ simulated samples of $\sigma_p$ and $\sigma_n$ for the control parameters by adding 5% Gaussian noise to the true sample. We start the training data with size of 10.000 samples. Then, at each training level, we add a batch of 1000 samples that are generated based on the predicted results from the previous level of training. All training levels have the exact VAIM architecture and a random initialization. Fig. 44 illustrates our preliminary results of applying active learning in three levels of training. One can find that the precision of the predicted parameters $\alpha_u, \beta_u$ and $\beta_d$ is clearly improved through the training levels with the third level predictions being more determined than the first level. The precision of the predicted parameter $\alpha_d$ is not much improved in the second and third levels of training, however; it becomes more centered around the control parameter, shown in red. We further validate these predicted parameters in the three levels of training by reconstructing the associated $\sigma_p$ and $\sigma_n$ using the theory model. Fig. 45 demonstrates the ratio to the true values of reconstructed $\sigma_p$ (top) and $\sigma_n$ (bottom). It is also notable that utilizing active learning significantly reduces the uncertainty associated with $\sigma_p$ and $\sigma_n$ as the ratio of the reconstructed $\sigma_p$ and $\sigma_n$ in the second and third levels of training have smaller bandwidth than the first level predictions.

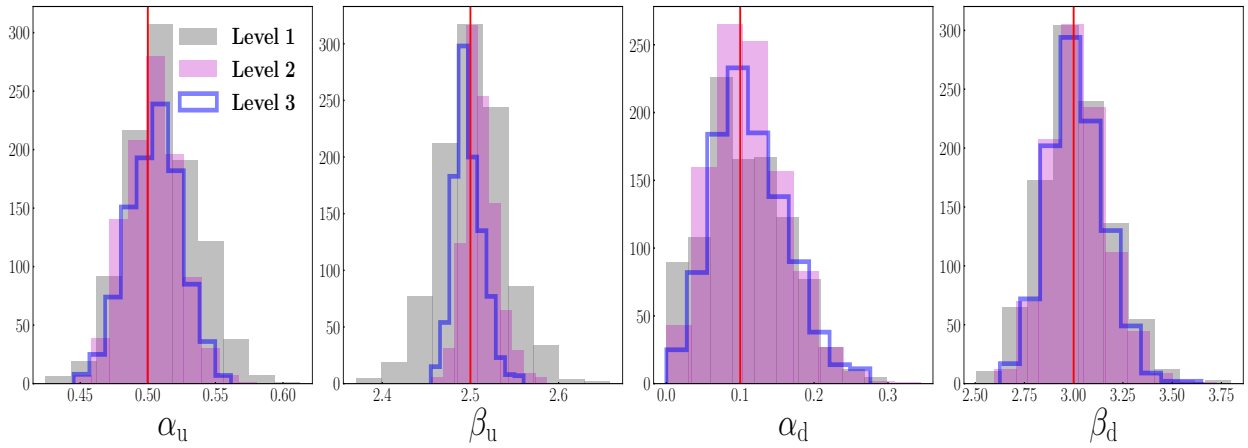



**Fig. 44.** The predicted parameters using active learning in three levels of training with the control parameter values indicated as red vertical lines.

Second, we plan to perform active learning approach based on importance sampling (AL-IS) scheme where the training data are incrementally increased on a multi-level training by sampling from the interesting region, in our case is the predictions with highest variance. In other words, at each level AL-IS strategy selects few test examples with the highest variance and obtain more data in the same distribution. We expect AL-IS will provide efficient strategies of collecting training data that is diverse, informative for reducing modeling errors, cover regions of the feature space where the predictor is most likely evaluated, and avoiding sampling the same portions frequently. Finally, several components of our methods such as the size of training data at the first level and the size of samples that is combined at each level, remain interesting questions for future research.

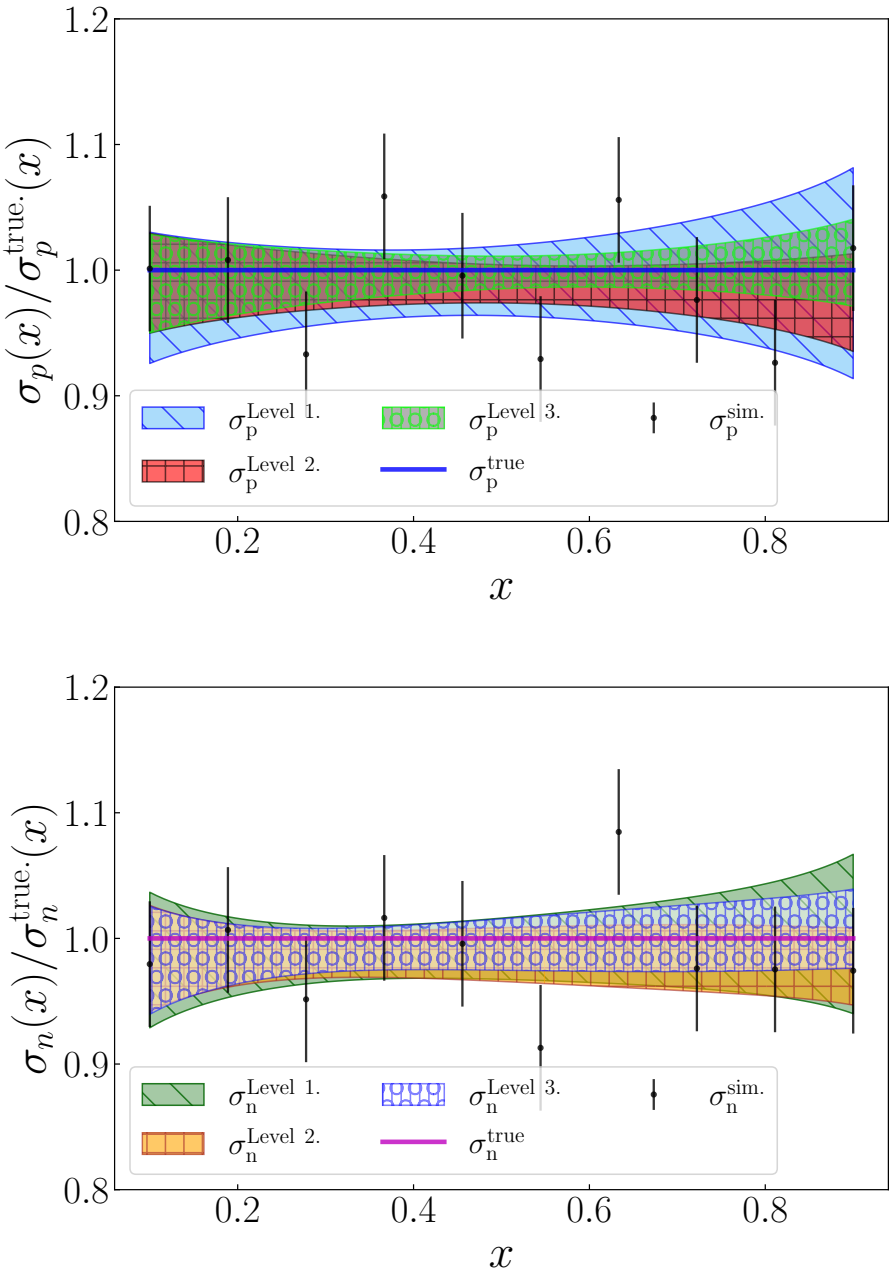



**Fig. 45.** Active learning predictions on the $1,000$ simulated control samples given by $\sigma_p^{sim}$ and $\sigma_n^{sim}$. (Top) the ratio of the reconstructed $\sigma_p$ to the true in the three levels of training. (Bottom) the ratio of the reconstructed $\sigma_n$ to the true in the three levels of training.

### 8.3.2 Extracting CFFs on all Available Kinematics

In chapter 6 we utilize C-VAIM to solve the inverse problem in CFF extraction from a single polarization observable. We train our C-VAIM on a set of five kinematics values that is listed in table 2. In our future work, we plan to extend our framework and incorporate all experimental DVCS data on all available kinematics points. This will include the kinematics values in fig. 46, with $1.1 \leqslant Q^2 \leq 8.5$ and $3.3 \leqslant E_b \leq 10.9$.

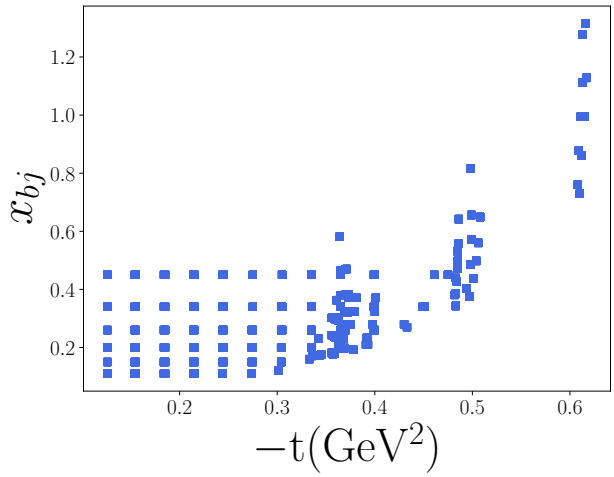



**Fig. 46.** The experimental DVCS kinematics values.

### 8.3.3 Inverse Mappers for Medical Imaging

Solving inverse problems is a significant research area in medical imaging applications. In particular, reconstructing Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) from observables (k-spaces for MRI and sinograms for CT), form critical inverse problems. These indicated inverse problems in medical field emerge because the target images are invisible to human eyes or unreachable. The objective behind addressing these problems is to provide a visual representation of internal human body organs, assisting in clinical analysis and medical interventions.

Recent deep learning methods have been utilized as a state-of-the-art technique for such inverse problems in imaging [133, 134]. The inverse mappers developed in this dissertation are not

exclusive to nuclear physics and can be adapted to solve inverse problems in different applications. In our future work, we plan to further investigate the capability of our inverse mappers on medical imaging problems.

# REFERENCES


[1] Chadan, K. & Sabatier, P. C. *Inverse Problems in Quantum Scattering Theory* (Springer Publishing Company, Incorporated, 2011), 2nd edn.

[2] Tarantola, A. & Valette, B. Inverse problems - quest for information. *Journal of Geophysics* **50**, 159–170 (1982).

[3] Craig, I. J. D. & Brown, J. C. Inverse problems in astronomy, a guide to inversion strategies for remotely sensed data (1986).

[4] Hadamard. Sur les problèmes aux dérivés partielles et leur signification physique. *Princeton University Bulletin* **13**, 49–52 (1902).

[5] Accardi, A. *et al.* Electron Ion Collider: The Next QCD Frontier. *European Physical Journal* **A52**, 268 (2016).

[6] Forte, S. & Watt, G. Progress in the determination of the partonic structure of the proton. *Annual Review of Nuclear and Particle Science* **63**, 291–328 (2013).

[7] Jimenez-Delgado, P., Melnitchouk, W. & Owens, J. F. Parton momentum and helicity distributions in the nucleon. *Journal of Physics G* **40**, 093102 (2013).

[8] Metz, A. & Vossen, A. Parton fragmentation functions. *Progress in Particle and Nuclear Physics* **91**, 136–202 (2016).

[9] Placakyte, R. Parton distribution functions. In *Proceedings of 31st International Symposium on Physics In Collision* (2011).

[10] Kretzer, S. Fragmentation functions from flavor inclusive and flavor tagged e+ e- annihilations. *Physical Review D* **62**, 054001 (2000).

[11] Aybat, S. M. & Rogers, T. C. TMD Parton Distribution and Fragmentation Functions with QCD Evolution. *Physical Review D* **83**, 114042 (2011).

[12] Diehl, M. Generalized parton distributions. *Physics Reports* **388**, 41–277 (2003).

[13] Belitsky, A. V. & Radyushkin, A. V. Unraveling hadron structure with generalized parton distributions. *Physics Reports* **418**, 1–387 (2005).

[14] Li, H., Schwab, J., Antholzer, S. & Haltmeier, M. NETT: Solving inverse problems with deep neural networks. *Inverse Problems* **36**, 065005 (2020).

[15] Adler, J. & Öktem, O. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems* **33**, 124007 (2017).

[16] Hinton, G. & Zemel, R. Autoencoders, minimum description length and Helmholtz free energy. In *Advances in neural information processing systems (NIPS'93)* (1994).

[17] Blei, D., Kucukelbir, A. & McAuliffe, J. Variational inference: A review for statisticians. *Journal of the American Statistical Association* **112**, 859–877 (2017).

[18] Qi, C., Su, H., Mo, K. & Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition* 77–85 (2017).

[19] Stuart, A. M. Inverse problems: A bayesian perspective. *Acta Numerica* **19**, 451–559 (2010).

[20] Daubechies, I., Defrise, M. & De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint (2003).

[21] Natterer, F. & Wübbeling, F. *Mathematical Methods in Image Reconstruction* (Society for Industrial and Applied Mathematics, 2001).

[22] Kamyab, S., Azimifar, Z., Sabzi, R. & Fieguth, P. W. Survey of deep learning methods for inverse problems. *CoRR* **abs/2111.04731** (2021). `2111.04731`.

[23] Zhang, S. & Salari, E. Image denoising using a neural network based non-linear filter in wavelet domain. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 2, 989–992 Vol. 2 (2005).

[24] Jain, V. & Seung, S. Natural image denoising with convolutional networks. In Koller, D., Schuurmans, D., Bengio, Y. & Bottou, L. (eds.) *Advances in Neural Information Processing Systems*, vol. 21 (Curran Associates, Inc., 2008).

[25] Agostinelli, F., Anderson, M. R. & Lee, H. Adaptive multi-column deep neural networks with application to robust image denoising. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z. & Weinberger, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 26 (Curran Associates, Inc., 2013).

[26] Schuler, C. J., Burger, H. C., Harmeling, S. & Schölkopf, B. A machine learning approach for non-blind image deconvolution. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 1067–1074 (2013).

[27] Adler, A., Araya-Polo, M. & Poggio, T. Deep learning for seismic inverse problems: Toward the acceleration of geophysical analysis workflows. *IEEE Signal Processing Magazine* **38**, 89–119 (2021).

[28] Csáji, B. C. *Approximation with Artificial Neural Networks*. M.s.thesis, Dept.Sciences, Etvs Lornd University, Budapest, Hungary (2001).

[29] Lucas, A., Iliadis, M., Molina, R. & Katsaggelos, A. Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Audio and Electroacoustics Newsletter* **35**, 20–36 (2018).

[30] Hastie, T., Tibshirani, R. & Friedman, J. *The elements of statistical learning: data mining, inference and prediction* (Springer), 2 edn.

[31] Murphy, K. P. *Machine learning : a probabilistic perspective* (MIT Press, 2013).

[32] Pumplin, J. *et al.* New generation of parton distributions with uncertainties from global QCD analysis. *Journal of High Energy Physics* **2002**, 012–012 (2002).

[33] Sato, N., Melnitchouk, W., Kuhn, S., Ethier, J. & Accardi, A. Iterative monte carlo analysis of spin-dependent parton distributions. *Physical Review D* **93** (2016).

[34] Harland-Lang, L. A., Martin, A. D., Motylinski, P. & Thorne, R. S. Parton distributions in the LHC era: MMHT 2014 PDFs. *European Physical Journal C* **75**, 204 (2015).

[35] Alekhin, S., Blümlein, J. & Moch, S. NLO PDFs from the ABMP16 fit. *Eur. Phys. J. C* **78**, 477 (2018). 1803.07537.

[36] Hou, T.-J. *et al.* New CTEQ global analysis of quantum chromodynamics with high-precision data from the LHC. *Physical Review D* **103**, 014013 (2021).

[37] Mehta, P. *et al.* A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports* **810**, 1–124 (2019).

[38] Ethier, J. J., Sato, N. & Melnitchouk, W. First simultaneous extraction of spin-dependent parton distributions and fragmentation functions from a global qcd analysis. *Physical Review Letters* **119**, 132001 (2017).

[39] Cammarota, J. *et al.* Origin of single transverse-spin asymmetries in high-energy collisions. *Physical Review D* **102**, 054002 (2020).

[40] Forte, S., Garrido, L., Latorre, J. I. & Piccione, A. Neural network parametrization of deep inelastic structure functions. *Journal of High Energy Physics* **05**, 062 (2002).

[41] Ball, R. D. *et al.* A first unbiased global nlo determination of parton distributions and their uncertainties. *Nuclear Physics* **838**, 136–206 (2010).

[42] Carrazza, S. & Cruz-Martinez, J. Towards a new generation of parton densities with deep learning models. *European Physical Journal C* **79** (2019).

[43] Whalley, M. R., Bourilkov, D. & Group, R. C. The Les Houches accord PDFs (LHAPDF) and LHAGLUE. In *HERA and the LHC: A Workshop on the Implications of HERA and LHC Physics* (2005).

[44] Bishop, C. M. Mixture density networks. Working Paper, Aston Univ. (1994).

[45] Tikhonov, A. N. On the stability of inverse problems. *Proceedings of the USSR Academy of Sciences* **39**, 195–198 (1943).

[46] Tikhonov, A. N. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.* **4**, 1035–1038 (1963).

[47] Tikhonov, A. N. On the stability of the functional optimization problem. *Ussr Computational Mathematics and Mathematical Physics* **6**, 28–33 (1966).

[48] Lukas, M. A. Robust generalized cross-validation for choosing the regularization parameter. *Inverse Problems* **22**, 1883 (2006).

[49] Hansen, P. The l-curve and its use in the numerical treatment of inverse problems. In *Computational Inverse Problems in Electrocardiology* (WIT Press, 2000).

[50] Bartlett, M. S. Properties of Sufficiency and Statistical Tests. *Proceedings of the Royal Society of London Series A* **160**, 268–282 (1937).

[51] Hansen, P. C. *Rank-Deficient and Discrete Ill-Posed Problems* (Society for Industrial and Applied Mathematics, 1998).

[52] Backus, G. Inference from inadequate and inaccurate data, iii. *Proceedings of the National Academy of Sciences of the United States of America* **67**, 282–289 (1970).

[53] Franklin, J. N. Well-posed stochastic extensions of ill-posed linear problems. *Journal of Mathematical Analysis and Applications* **31**, 682–716 (1970).

[54] Tarantola, A. & Valette, B. Inverse problems - quest for information. *Journal of Geophysics* 159–170 (1982).

[55] Kaipio, J. P. & Somersalo, E. Statistical and computational inverse problems (2004).

[56] Matérn, B. *Spatial variation*. 49:5 (Springer New York, 1960).

[57] Whittle, P. On stationary processes in the plane. *Biometrika* **41**, 434–449 (1954).

[58] Kaipio, J. P., Kolehmainen, V., Vauhkonen, M. & Somersalo, E. Inverse problems with structural prior information. *Inverse Problems* **15**, 713–729 (1999).

[59] Donoho, D. L., Johnstone, I. M., Hoch, J. C. & Stern, A. S. Maximum entropy and the nearly black object. *Journal of the Royal Statistical Society. Series B (Methodological)* **54**, 41–81 (1992).

[60] Calvetti, D. & Somersalo, E. Priorconditioners for linear systems. *INVERSE PROBLEMS* **21**, 1397–1418 (2005).

[61] Vauhkonen, M., Kaipio, J., Somersalo, E. & Karjalainen, P. Electrical impedance tomography with basis constraints. *Inverse Problems* **13**, 523–530 (1997).

[62] Qiu, J. Nuclear femtography a new frontier of science and technology (2020).

[63] Boehnlein, A. *et al.* Artificial Intelligence and Machine Learning in Nuclear Physics (2021). 2112.02309.

[64] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, Heidelberg, 2006).

[65] Murphy, K. P. *Machine learning : a probabilistic perspective* (MIT Press, Cambridge, Mass. [u.a.], 2013).

[66] Placakyte, R. Parton Distribution Functions. In *31st International Symposium on Physics In Collision* (2011). 1111.5452.

[67] Kriesten, B. T. *3D Femtography of the Nucleon Using Generalized Parton Distribution Functions*. Ph.D. thesis, Virginia U. (2021).

[68] Arridge, S., Maass, P., Öktem, O. & Schönlieb, C. Solving inverse problems using data-driven models. *Acta Numerica* **28**, 1–174 (2019).

[69] Ardizzone, L., Kruse, J., Rother, C. & Köthe, U. Analyzing inverse problems with invertible neural networks. In *Proceedings of 7th International Conference on Learning Representations* (2019).

[70] Bridle, J. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *NATO Neurocomputing* (1989).

[71] Jacobs, R. A., Jordan, M. I., Nowlan, S. J. & Hinton, G. E. Adaptive Mixtures of Local Experts. *Neural Computation* **3**, 79–87 (1991).

[72] Dinh, L., Sohl-Dickstein, J. & Bengio, S. Density estimation using real NVP. In *International Conference on Learning Representations* (2017).

[73] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. J. A kernel method for the two-sample problem. *CoRR* **0805.2368** (2008).

[74] Obmann, D., Nguyen, L., Schwab, J. & Haltmeier, M. Augmented nett regularization of inverse problems. *Journal of Physics Communications* **5**, 105002 (2021).

[75] Obmann, D., Schwab, J. & Haltmeier, M. Deep synthesis network for regularizing inverse problems. *Inverse Problems* **37**, 015005 (2020).

[76] Almaeen, M. *et al.* Variational autoencoder inverse mapper: An end-to-end deep learning framework for inverse problems. In *Proceedings of 2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8 (2021).

[77] Kingma, D. & Welling, M. Auto-Encoding Variational Bayes. In *Proceedings of 2nd International Conference on Learning Representations, (ICLR)* (2014).

[78] Sato, N., Andres, C., Ethier, J. J. & Melnitchouk, W. Strange quark suppression from a simultaneous Monte Carlo analysis of parton distributions and fragmentation functions (2019). `1905.03788`.

[79] Duke, D. W. & Owens, J. F. Q**2 Dependent Parametrizations of Parton Distribution Functions. *Phys. Rev.* **D30**, 49–54 (1984).

[80] Whitlow, L. W., Riordan, E. M., Dasu, S., Rock, S. & Bodek, A. Precise measurements of the proton and deuteron structure functions from a global analysis of the SLAC deep inelastic electron scattering cross-sections. *Physical Letters B* **282**, 475–482 (1992).

[81] Benvenuti, A., Bollini, D. *et al.* Test of qcd and a measurement of $\lambda$ from scaling violations in the proton structure function f2(x, q2) at high q2. *Physics Letters B* **223**, 490–496 (1989).

[82] Arneodo, M. *et al.* Measurement of the proton and deuteron structure functions, F2(p) and F2(d), and of the ratio sigma-L / sigma-T. *Nuclear Physics B* **483**, 3–43 (1997).

[83] Arneodo, M. *et al.* Accurate measurement of F2(d) / F2(p) and Rd - Rp. *Nuclear Physics B* **487**, 3–26 (1997).

[84] Abramowicz, H. *et al.* Combination of measurements of inclusive deep inelastic $e^{\pm}p$ scattering cross sections and QCD analysis of HERA data. *European Physical Journal C* **75**, 580 (2015).

[85] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B. & Smola, A. A kernel two-sample test. *Journal of Machine Learning Research* **13**, 723–773 (2012).

[86] Ramdas, A., Reddi, S. J., Póczos, B., Singh, A. & Wasserman, L. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *Proceedings of 29th AAAI Conference on Artificial Intelligence* (AAAI Press, 2015).

[87] Goodfellow, I. J. *et al.* Generative adversarial networks. *Proceedings of 27th International Conference on Neural Information Processing Systems* (2014).

[88] Kobyzev, I., Prince, S. & Brubaker, M. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1–1 (2020).

[89] Chaturvedi, I. & Xiang, J. Constrained manifold learning for videos. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)* (2020).

[90] Almaeen, M., Alanazi, Y., Sato, N., Melnitchouk, W. & Li, Y. Point cloud-based variational autoencoder inverse mappers (pc-vaim) - an application on quantum chromodynamics global analysis. In *Proceedings of 2021 International Conference on Machine Learning and Applications (ICMLA)* (2022).

[91] Akmal Butt, M. & Maragos, P. Optimum design of chamfer distance transforms. *IEEE Transactions on Image Processing* **7**, 1477–1484 (1998).

[92] Arridge, S. R., Maass, P., Öktem, O. & Schönlieb, C.-B. Solving inverse problems using data-driven models. *Acta Numerica* **28**, 1 – 174 (2019).

[93] Diehl, M. Generalized parton distributions. *Phys. Rept.* **388**, 41–277 (2003). hep-ph/0307382.

[94] Belitsky, A. & Radyushkin, A. Unraveling hadron structure with generalized parton distributions. *Phys.Rept.* **418**, 1–387 (2005). hep-ph/0504030.

[95] Kumericki, K., Liuti, S. & Moutarde, H. GPD phenomenology and DVCS fitting: Entering the high-precision era. *Eur. Phys. J. A* **52**, 157 (2016). 1602.02763.

[96] Ji, X.-D. Gauge-Invariant Decomposition of Nucleon Spin. *Phys. Rev. Lett.* **78**, 610–613 (1997). hep-ph/9603249.

[97] Almaeen, M. *et al.* Benchmarks for a Global Extraction of Information from Deeply Virtual Exclusive Scattering (2022). 2207.10766.

[98] Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M. & Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 28 (Curran Associates, Inc., 2015).

[99] O'Malley, T. *et al.* Kerastuner (2019).

[100] Kriesten, B. & Liuti, S. Theory of deeply virtual Compton scattering off the unpolarized proton. *Phys. Rev. D* **105**, 016015 (2022). 2004.08890.

[101] Belitsky, A. V. & Mueller, D. Exclusive electroproduction revisited: treating kinematical effects. *Phys. Rev.* **D82**, 074010 (2010). 1005.5209.

[102] Kriesten, B. *et al.* Extraction of generalized parton distribution observables from deeply virtual electron proton scattering experiments. *Phys. Rev. D* **101**, 054021 (2020). 1903.05742.

[103] Kriesten, B. *et al.* Extraction of Generalized Parton Distribution Observables from Deeply Virtual Electron Proton Scattering Experiments. *Phys. Rev. D* **101**, 054021 (2020). 1903.05742.

[104] van der Maaten, L. & Hinton, G. E. Visualizing data using t-sne. *Journal of Machine Learning Research* **9**, 2579–2605 (2008).

[105] Anikin, I. V. & Teryaev, O. V. Dispersion relations and subtractions in hard exclusive processes. *Phys. Rev. D* **76**, 056007 (2007). 0704.2185.

[106] Diehl, M. & Ivanov, D. Y. Dispersion representations for hard exclusive processes: beyond the Born approximation. *Eur. Phys. J. C* **52**, 919–932 (2007). 0707.0351.

[107] Karniadakis, G. *et al.* Physics-informed machine learning 1–19 (2021).

[108] Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **378**, 686–707 (2019).

[109] Kriesten, B., Liuti, S. & Meyer, A. Novel Rosenbluth extraction framework for Compton form factors from deeply virtual exclusive experiments. *Phys. Lett. B* **829**, 137051 (2022). 2011.04484.

[110] Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nat Rev Phys* **3**, 422–440 (2021).

[111] Raissi, M., Perdikaris, P. & Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **378**, 686–707 (2019).

[112] Defurne, M. *et al.* A glimpse of gluons through deeply virtual compton scattering on the proton. *Nature Commun.* **8**, 1408 (2017). 1703.09442.

[113] Defurne, M. *et al.* E00-110 experiment at Jefferson Lab Hall A: Deeply virtual Compton scattering off the proton at 6 GeV. *Phys. Rev. C* **92**, 055202 (2015).

[114] Halevy, A., Norvig, P. & Pereira, F. The unreasonable effectiveness of data. *IEEE Intelligent Systems* **24**, 8–12 (2009).

[115] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **15**, 1929–1958 (2014).

[116] Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. & Blei, D. (eds.) *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37 of *Proceedings of Machine Learning Research*, 448–456 (PMLR, Lille, France, 2015).

[117] Weiss, K. R., Khoshgoftaar, T. M. & Wang, D. A survey of transfer learning. *Journal of Big Data* **3**, 1–40 (2016).

[118] Shao, L., Zhu, F. & Li, X. Transfer learning for visual categorization: A survey. *IEEE Transactions on Neural Networks and Learning Systems* **26**, 1019–1034 (2015).

[119] Erhan, D. *et al.* Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* **11**, 625–660 (2010).

[120] Shorten, C. & Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data* **6**, 1–48 (2019).

[121] Defurne, M. *et al.* E00-110 experiment at Jefferson Lab Hall A: Deeply virtual Compton scattering off the proton at 6 GeV. *Phys. Rev.* **C92**, 055202 (2015). 1504.05453.

[122] Goldstein, G. R., Hernandez, J. O. & Liuti, S. Flexible Parametrization of Generalized Parton Distributions from Deeply Virtual Compton Scattering Observables. *Phys.Rev.* **D84**, 034007 (2011). 1012.3776.

[123] Gonzalez-Hernandez, J. O., Liuti, S., Goldstein, G. R. & Kathuria, K. Interpretation of the Flavor Dependence of Nucleon Form Factors in a Generalized Parton Distribution Model. *Phys. Rev.* **C88**, 065206 (2013). 1206.1876.

[124] Kriesten, B., Velie, P., Yeats, E., Lopez, F. Y. & Liuti, S. Parametrization of quark and gluon generalized parton distributions in a dynamical framework. *Phys. Rev. D* **105**, 056022 (2022). 2101.01826.

[125] Almaeen, M. *et al.* Benchmarks for a global extraction of information from deeply virtual exclusive scattering (2022).

[126] Fu, Y., Zhu, X. & Li, B. A survey on instance selection for active learning. *Knowledge and Information Systems* **35**, 249–283 (2013).

[127] Settles, B. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning (Morgan & Claypool Publishers, 2012).

[128] Aggarwal, C., Kong, X., Gu, Q., Han, J. & Yu, P. *Active learning: A survey*, 571–605 (CRC Press, 2014).

[129] Kumar, P. & Gupta, A. Active learning query strategies for classification, regression, and clustering: A survey. *J. Comput. Sci. Technol.* **35**, 913–945 (2020).

[130] Roy, N. & McCallum, A. Toward optimal active learning through sampling estimation of error reduction. In *ICML* (2001).

[131] MacKay, D. J. C. Information-Based Objective Functions for Active Data Selection. *Neural Computation* **4**, 590–604 (1992).

[132] Cohn, D. A., Ghahramani, Z. & Jordan, M. I. Active learning with statistical models. *J. Artif. Int. Res.* **4**, 129–145 (1996).

[133] Akcakaya, M., Moeller, S., Weingärtner, S. & Ugurbil, K. Scan-specific robust artificial-neural-networks for k-space interpolation (raki) reconstruction: Database-free deep learning for fast imaging. *Magnetic Resonance in Medicine* **81**, 439–453 (2019).

[134] Schlemper, J., Caballero, J., Hajnal, J., Price, A. & Rueckert, D. A deep cascade of convolutional neural networks for dynamic mr image reconstruction. *IEEE Transactions on Medical Imaging* **PP** (2017).

# VITA

Manal Almaeen
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

## EDUCATION

2010, B.S in Computer Science, Jouf University, Saudi Arabia.
2017, M.S in Software Engineering, University of St. Thomas, USA.
2023, PhD Candidate in Computer Science, Old Dominion University, USA.

## PUBLICATIONS

Google Scholar: `https://scholar.google.com/citations?user=HL6yNkYAAAAJ&hl=en`

## CONTACT

Email: mhmaeen@ju.edu.sa