Machine Learning Approximation of Bayesian Inference in Nuclear Fusion Research

vorgelegt von
M.Sc.
Andrea Pavone
ORCID: 0000-0003-2398-966X

von der Fakultät II – Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
— Dr. rer. nat. —

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr. Holger Stark
Gutachter: Prof. Dr. Dieter Breitschwerdt
Gutachter: Prof. Dr. Robert Wolf
Gutachter: Prof. Dr. Eric Sonnendrücker
Tag der wissenschaftlichen Aussprache: 05 Oktober 2020

Berlin, 2020

Verily, verily, I say unto you, except a corn of wheat fall into the ground and die, it abideth alone: but if it die, it bringeth forth much fruit.

John 12:24-26

Abstract

Machine learning algorithms, especially neural networks, are commonly used to automatically find and encode unknown relationships between sets of related data. They have been applied to problems in different fields, including the interpretation of data in scientific experiments. In this thesis, I consider the case where the relationship between a set of observed data and modeling parameters is known and formalized in a Bayesian model, and I show that neural networks can be trained to learn the relationship between the quantities defined by the model joint distribution. In this way, the network constitutes an approximation of the original Bayesian model. Two cases are considered: the case where the network is trained to approximate the mapping between the observable quantities and the free parameters, and between the joint space of both observables and free parameters, and the joint probability distribution value. Neural network approximate models can be particularly useful in the context of large scientific experiments, like the nuclear fusion experiments considered here. The trained network can be used to accelerate the Bayesian inference process carried out on experimental data by providing a fast approximate reconstruction of relevant quantities. Here, I consider the application of the network to the reconstruction of different plasma parameters from measured diagnostic data collected at two large fusion experiments, the Wendelstein 7-X stellarator and the Joint European Torus tokamak. When analysing experimental data, it is crucial to provide information about the reliability of the network reconstruction for further usages of the results. For this reason, I show here how uncertainties can be estimated with two different approaches, both based on a Bayesian interpretation of the training process. The approximation method developed here is general in the sense that its implementation and validity is not bound to a specific model or experiment. Especially, if the Bayesian models to be approximated are implemented within the Minerva Bayesian modeling framework [1], their joint distributions can be easily accessed and used to generate training samples in a manner that is common among different models. This opens the possibility to an entirely automatic procedure, where machine learning models

are automatically generated and trained to approximate any scientific model, constituting a step further towards the automation of general scientific inference.

Zusammenfassung

Algorithmen für Maschinelles Lernen, insbesondere neuronale Netze, werden üblicherweise verwendet, um unbekannte Beziehungen zwischen verwandten Datensätzen automatisch zu finden und zu beschreiben. Diese wurden bereits in verschiedenen Bereichen auf geeignete Probleme angewendet, einschließlich der Interpretation von Daten in wissenschaftlichen Experimenten. In dieser Arbeit betrachte ich eine Anwendung, in der die Beziehung zwischen einem Satz beobachteter Daten und Modellierungsparametern bekannt und in einem Bayesschen Modell formalisiert ist. Ich zeige, dass neuronale Netze trainiert werden können, um die Beziehung zwischen den durch das Modell definierten gemeinsamen Wahrscheinlichkeitsverteilungen zu lernen. Auf diese Weise stellt das Netzwerk eine Approximation an das ursprüngliche Bayessche Modell dar. Es werden zwei Fälle betrachtet: einerseits der Fall, in dem das Netzwerk trainiert wird, um die Abbildung zwischen den beobachtbaren Größen und den freien Parametern, sowie der Fall, die Abbildung zwischen dem gemeinsamen Raum von beobachtbaren und freien Parametern und dem Wert der gemeinsamen Wahrscheinlichkeitsverteilung zu approximieren. Approximationen durch neuronale Netze können besonders im Zusammenhang mit großen wissenschaftlichen Experimenten, wie den hier betrachteten Kernfusionsexperimenten, nützlich sein. Das trainierte Netzwerk kann verwendet werden, um den Bayesschen Deduktionsprozess zu beschleunigen, der auf experimentelle Daten angewendet wird, indem eine schnelle approximative Rekonstruktion relevanter Größen bereitgestellt wird. Hier betrachte ich die Anwendung des Netzwerks auf die Rekonstruktion verschiedener Plasmaparameter aus gemessenen Diagnostikdaten zweier großer Fusionsexperimente, dem Stellarator Wendelstein 7-X und dem Tokamak Joint European Torus. Bei der Analyse experimenteller Daten ist es wichtig, Informationen über die Zuverlässigkeit der Rekonstruktion durch das Netzwerk bereitzustellen, um die Ergebnisse anderweitig nutzen zu können. Aus diesem Grund zeige ich in dieser Arbeit, wie Unsicherheiten mit zwei verschiedenen Ansätzen abgeschätzt werden können, die beide auf einer Bayesschen Interpretation des Trainingsprozesses basieren. Die hier ent-

wickelte Approximationsmethode ist allgemeingültig in dem Sinne, dass ihre Implementierung und Gültigkeit nicht an ein bestimmtes Modell oder Experiment gebunden ist. Insbesondere wenn die zu approximierenden Bayesschen Modelle innerhalb des Bayesschen Modellierungsframeworks Minerva [1] implementiert sind, können ihre gemeinsamen Wahrscheinlichkeitsverteilungen leicht berechnet und verwendet werden, um Trainingsmuster auf eine Weise zu generieren, die für verschiedene Modelle identisch ist. Dies eröffnet die Möglichkeit eines vollautomatischen Verfahrens, bei dem auf maschinellem Lernen basierende Modelle automatisch generiert und trainiert werden, um beliebige wissenschaftliche Modelle zu approximieren. Dies ist ein weiterer Schritt in Richtung automatischer Bearbeitung wissenschaftlicher Fragestellungen.

Contents

Chapter 1.

Introduction

The work presented in this thesis aims at contributing to the quest of the automation of science. In recent years, the breakthroughs of modern AI systems have rightfully made the news, and unprecedented effort is being made in the academic community and the commercial sector towards the realization of general intelligent automation. Many applications have been developed which show how machines can autonomously learn and perform at close to human level in different tasks. For example, in the context of natural language processing, deep learning based models are able to predict the next word in a sentence or answer questions with unprecedented accuracy, as it is shown in [2, 3]; in the context of strategy-based video games, machines are able to defeat human players after learning rules and strategy without supervision, from scratch [4]; in the field of robotic, a human-like robot hand has learnt the movements and strategy necessary to solve th Rubik's Cube using only simulated data [5]. Less resonant in the public sphere, are the applications to scientific research. The work presented here is not, by itself, a solution to the attempt of realizing autonomous, intelligent, general scientific inference. If the problem will ever be solved, and I believe it will, it is hard to predict now what form the solution, or multiple solutions, will have, but I and many other scientists believe that it will be built upon a number of tools and ideas, some of them taken from the fields of computer science and probability theory, and being known by a long time. Specifically, concerning the work in this thesis, I refer to the subjects of machine learning with its most recent developments, and Bayesian probability theory, which was first introduced in the eighteenth century. Moreover, computer science has shown us that in order to effectively manipulate information and knowledge, it is fundamental to pay attention to how the knowledge is represented. The point of the *representation of knowledge* is seldom mentioned explicitly in

the publications that constitute the core of this thesis, partially because it is taken for granted and partially because the publications tend to have a rather technical focus. Nevertheless, it is highly relevant to this work and I wish to clarify why this is the case and make it evident here and in the later chapters. Besides being taken for granted in the publications presented here, I have come to realize that it is also often overlooked by the very users of such knowledge. The reason why this happens, I believe, is that once a kind of knowledge is well represented, then its use becomes immediate, smooth, as a second nature or as speaking one's native language. Consequently, the system making the representation possible is, by its nature, invisible, and acts unnoticed, although steadily, permeating the space where all subsequent actions take place.

When I started this work, such system was given to me in the form of a computational framework for the representation of complex models in the language of Bayesian graphical models, and all the work I have made has happened in the context of this framework, named Minerva [1]. Having had the chance to do my research under the supervision of its main author, I could get to know and understand the vision behind it. The quest that motivated its realization is the one mentioned at the opening of this introduction: clearly, automation of science is a very broad expression that requires specifications. I refer here, specifically, to scientific inference. Scientific inference is based on *forward models*: these are models of real world phenomena that are used to make predictions about what can be observed or measured, and whose outcomes are compared to what is observed in order to refine our knowledge of the world. Such models are made of a set of assumptions and beliefs - taking the form, for example, of physics relations - about how the world works, and when they are confronted to what is observed, they can be changed or updated accordingly. They represent our expectations and understanding of a system. For example, in the context of the work of this thesis, a forward model might be the description of a plasma in terms of parameters relevant to achieve nuclear fusion, like its temperature and density. Given the value of these parameters, the model may consist of a calculation, based on physics relations, which outputs the value of a quantity measured during an experiment. Once an experiment is performed, then, we have the possibility to compare the expected outcome with the observations, and find the value of the parameters that consistently describe the measurements. This allows us to refine our knowledge about the system or, more in general, the world, in a way that is consistent with what is observed. The refining of the model assumptions or parameters can be formalized and made

systematic in the framework of Bayesian inference. Given a set of assumptions and a forward model, Bayes formula provides a rule to update hypothesis according to the results of comparing observations with model predictions: the formalization of the process that is employed in rigorous scientific discovery. The idea of exploiting probability theory as a system to reason in science is famously described in [6] and [7]. I believe that part of the strength of Bayesian probability theory comes from the fact that hypothesis and assumptions are made explicit in the forward model *before* taking into account the observations: it is this modeling act, through the explicit writing down of hypothesis and beliefs, that makes the entire process repeatable and objective, as science is expected to be, and that uncovers our subjectivity when interpreting the outcomes of an experiment. We tend to agree that scientific knowledge should be objective and universally valid; if this is the case, the process of scientific discovery itself is meant to be independent of those who perform it, and therefore can be made automatic. I believe such automation can be achieved through the use of Bayesian inference and forward models in representing and confronting with reality. The framework of Bayesian inference is a very general one, as it does not require assumptions on the models involved: they can belong to physics, chemistry, social sciences, or any other field, as long as they can make predictions from clearly stated assumptions. Therefore, what if we could find a common language which allows to express *any* scientific model in a way that is in accordance with the requirements of Bayes theorem, and can be understood by computers, so that all processes of scientific inference and knowledge acquisition could be performed autonomously? This is the ambitious task that, before I started this work, the people I have worked with, have undertaken. The proposed solution took the form of the computational framework that I mentioned above, named as the Minerva framework. A more detailed explanation of how it works is given in the following chapters. Here it is enough to know that it allows to generalize the writing of scientific models: typically they are composed of different modules, or nodes, and they can be used to do Bayesian inference when observations are available. As a start, the framework has been used extensively at different nuclear fusion experiments (see for example [8–12]).

In this context, the usage of forward models is often computationally expensive because they require performing complex calculations and when they are used to do inference, they are often used in an iterative scheme involving up to millions of iterations. This is where machine learning, and specifically

neural network models, comes into play. Neural networks can process data at very short time scales, down to tens of microseconds. In this thesis, I show that they can be trained as an approximation, or *surrogate*, of a Bayesian model implemented within the Minerva framework. Then, they can provide a fast approximation of the Bayesian inference process, drastically reducing data analysis time. Besides this advantage, another crucial point of this work is the following: given a common framework where different models are expressed in a common way, the creation of approximate neural network models can be easily extended to any model implemented within the framework. This is relevant because it allows the approach to scale very easily, having the possibility to fully automate the procedure. The work presented in this thesis constitutes a step in this direction.

I will describe the application of this method to two different fusion experiments. In the context of the first one, the Wendelstein 7-X (W7-X) stellarator (see chapter 2), I have considered the system of an X-ray spectrometer [8, 13, 14]: it measures X-rays emitted in the plasma volume and collected along several lines of sight. The emission contains information about plasma parameters as the temperature of the ions and electrons, which are macroscopic quantities relevant for describing the state of a fusion plasma. The forward modeling of this system represents a complex problem: given the values of plasma parameters, the emission at each point along each line of sight is calculated from an atomic physics model, and then integrated along the line in order to predict the expected measurement. The evaluation of this model for the inference of possible values of plasma parameters with the conventional Bayesian inference approach takes up to tens of minutes for a single time measurement. The measuring device can record data with a frequency of few kHz, experiments currently can last up to tens of seconds and, in the future, up to tens of minutes, and during a campaign hundreds to thousands of experiments are run: this amounts to a volume of data that is hard to evaluate in its entirety with the conventional approach. Such system, therefore, clearly would benefit of an acceleration in the evaluation of the data: this can be provided by a neural network surrogate of the conventional Bayesian inference in an automatic way, by training the network only on data generated with the corresponding Bayesian model. No experimental data, or any other external data are required for the training. This is a crucial first achievement in view of automation, because it shows that the approach only relies on the Bayesian model and, therefore, can be seamlessly applied to any other model implemented within the Minerva framework. With

the use of neural networks, the data evaluation time can be drastically reduced to hundreds or tens of microseconds.

At the Joint European Torus (JET) experiment, instead, I have considered the case of a different spectroscopy system, the Li-beam diagnostic [15]: here, the emission stimulated by injection in the plasma of lithium atoms can be related to the density of plasma electrons, another useful parameter in the description of the plasma. Also in this case, the modeling within Minerva consists of calculating the expected radiation at each point along different lines of sight with a multi-state atomic physics model [10, 16]. The evaluation of this model in the conventional Bayesian inference scheme requires again tens of minutes for single time measurements. Thanks to the fact that also this forward model was implemented within the Minerva framework, the method used at W7-X could be straightforwardly implemented for this case, generating a neural network approximation of the traditional Bayesian inference. Without almost any further effort in the implementation, thanks to the common language of the Minerva models and the formalism of Bayesian inference, we could make neural network replicas of existing Bayesian inference in an automatic fashion for two different systems. One has to be clear: full automation was not entirely achieved at this stage, some aspects of the work still required human intervention, but this was a first step in the right direction. I believe, that, in the future, the procedure can be fully automated.

As it is now evident, different subjects are involved in this work. In the next chapters, I will introduce the three main ones: in chapter 2 I will introduce nuclear fusion, in chapter 3 I will introduce Bayesian inference and modeling in the Minerva framework, and in chapter 4 I will introduce the basic concepts of neural networks and expand on the core idea of neural network as approximate Bayesian inference. Chapter 5 contains a description of the applications of the neural network surrogates to nuclear fusion experiments, and chapter 6 is constituted of publications in peer reviewed journals containing the details of the aforementioned applications.

Chapter 2.

Nuclear fusion

The quest for safe and clean sources of energy is arguably one of the topics that simultaneously concern many parts of today's society, from policy makers, to entrepreneurs, environmental activists, politicians and the wide public in general. Nuclear fusion could provide a solution to the continuously increasing energy demand and the request for a form of energy production that does not produce green house gases, while at the same time having favorable safety properties. Similar to fission, nuclear fusion produces energy from the nuclear interaction of atoms, and it features a similarly large amount of specific energy, i.e. amount of energy released per mass of fuel burnt. Moreover, the fuel required is abundant on Earth.

Nuclear fusion is the process that powers the Sun. In the Sun, it involves two protons and produces, at its first step, a deuteron, a positron and a neutrino. Despite its very low cross-section, it occurs in the Sun thanks to the very large pressure. On Earth, most of the effort has been focused on the more likely deuterium - tritium reaction, having the highest reaction rate of all fusion reactions, which produces an alpha particle, a neutron and releases ≈ 17.6 MeV:

$$\mathrm{D} + \mathrm{T} \longrightarrow {}^{4}\mathrm{He} + \mathrm{n} + 17.6\,\mathrm{MeV} \tag{1}$$

For this reaction to occur, the Coulomb repulsive force between the two positive charged nuclei of deuterium and tritium has to be overcome. This is achieved by increasing the kinetic energy of the reactants by heating them up to the temperature of the order of $\approx 10^8$ °C, larger than temperature in the core of the Sun. At such temperatures, matter is found in the state of an ionized gas known as *plasma*. For example, for a plasma made of a 50:50 mix of deuterium and tritium, the optimal temperature for the fusion reaction to occur would be $k_b T \approx 20\,\mathrm{keV}$ (i.e., the maximum of the reactivity is found at such temperature),

with a particle density of $\approx 10^{20}\,\mathrm{m}^{-3}$ [17]. Because of the large thermal energy, the confinement of the particles is extremely difficult, and it constitutes, indeed, one of the major issues in today's nuclear fusion research. Plasma particles, as charged particles, have the advantage that their motion can be influenced by the electromagnetic force. This is why many experimental fusion devices rely on magnetic fields to confine the plasma in specific shapes within a vacuum chamber.

The two products of the reaction play an important role in a fusion reactor: the neutron, not being charged, can escape the magnetic confinement and its kinetic energy can be used to generate electric energy; the alpha particle, instead, is confined and through collisions can heat the plasma, providing enough self-heating to cover losses by conduction or radiation. By setting up a power balance between the energy produced by fusion reactions and lost due to particles escaping the systems, it is possible to derive an expression for what is the ideal minimum value that certain plasma parameters should have to sustain the production of energy. This is known as the *triple product*:

$$nT\tau_{\mathrm{E}} \geq 3 \cdot 10^{21}\,\mathrm{keVs/m^3} \tag{2.1}$$

where n is the plasma density, T is the plasma temperature, and τ_{E} is the energy confinement time, the characteristic time at which the system looses energy to the external environment - $\tau_{\mathrm{E}} = W/P_{\mathrm{heat}}$, where W is the thermal energy density of the plasma (units of energy per volume), and P_{heat} is the power needed to keep the plasma at the desired temperature and compensate losses. From the formulation of this criteria, we notice that two important plasma parameters are the plasma temperature and density. Since the plasma is made of a neutral gas ionized into free electrons and ions, it is useful to distinguish between the values that these quantities have for the two different species: the electron and ion temperature, T_{e} and T_{i} respectively, and the electron and ion density, n_{e} and n_{i}. A plasma is described as a quasi-neutral system where $n_{\mathrm{e}} = \sum_i Z_i n_{\mathrm{i}}$ with Z_i atomic number of ion species i, and n_i corresponding density. Therefore, for a pure hydrogen plasma (or its isotopes): $n_{\mathrm{e}} = n_{\mathrm{i}}$. The electron and ion temperature instead can be quite different, depending on the heating mechanisms and the collisional coupling of the two species. Although what is directly relevant for fusion is the temperature of the ions, in a burning D-T plasma, the fast α-particle population resulting from the fusion reaction predominantly heat the electrons, which, in turn, transfer their energy to the ion population. Moreover, there exist external heating mechanisms, such as the Electron Cyclotron Resonance

Heating (ECRH), that act in a similar way, preferably energizing electrons in the first step. Therefore, both parameters are widely used and of crucial relevance for any nuclear fusion experiment. For this reason, these three parameters are at the center of most of the publications attached to this thesis: article I in section 6.1 ([18]) shows how a neural network can be trained to speed-up the inference of ion and electron temperature from measurements collected with an X-ray imaging spectrometer at the W7-X stellarator, drastically reducing the analysis time and opening the possibility of real time applications; article II in section 6.2 ([19]) concerns the estimation of error bars in the neural network reconstruction of T_e and T_i; article IV in section 6.4 ([20]) concerns, instead, the case of fast, approximated neural network inference of the electron density from measurements performed with a lithium beam spectrometer at the JET tokamak.

When a plasma is operated within a reactor or an experimental device, it is never entirely free of impurities: the bulk of fusing ions is often contaminated by ion species released by the interaction of the plasma with surrounding materials. For example, at the W7-X stellarator, carbon ions are often found to contaminate the plasma because the divertor, a crucial component situated at the edge of the machine and used to collect the high heat loads in a sustainable and controlled manner, is made of graphite tiles. An important application of impurities is for the control of the radiative power to limit the heat flux to the divertor. The line emission caused by the presence of a controlled amount of impurities in certain regions of the plasma can help protecting the divertor and other plasma facing components. The presence of impurities can also be detrimental for the overall performance if above certain limits: high-Z impurities can cause energy to get lost by radiation, whereas low-Z impurities can dilute the fusion fuel. Therefore, during an experiment, it is often desirable to be able to measure and monitor the amount of impurity ions. A quantity related to their concentration is known as plasma *effective charge* Z_eff:

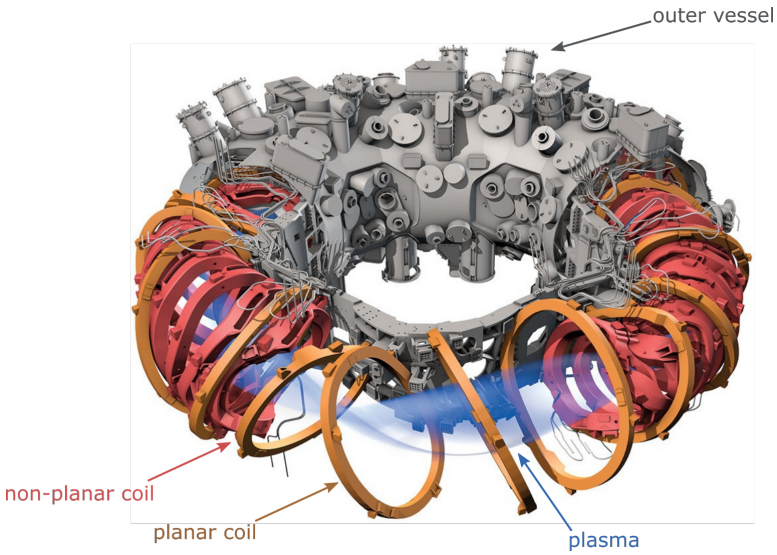
$$Z_\mathrm{eff} = \frac{\sum_i n_i Z_\mathrm{i}^2}{\sum_i n_\mathrm{i} Z_\mathrm{i}} \tag{2.2}$$

where i is an index labeling the i-th ion species in the plasma, n_i its corresponding density and Z_i the atomic number. Z_eff can also be derived from experimental measurement of bremsstrahlung emission, since this kind of emission is directly proportional to the effective charge. This kind of measurement also requires n_e

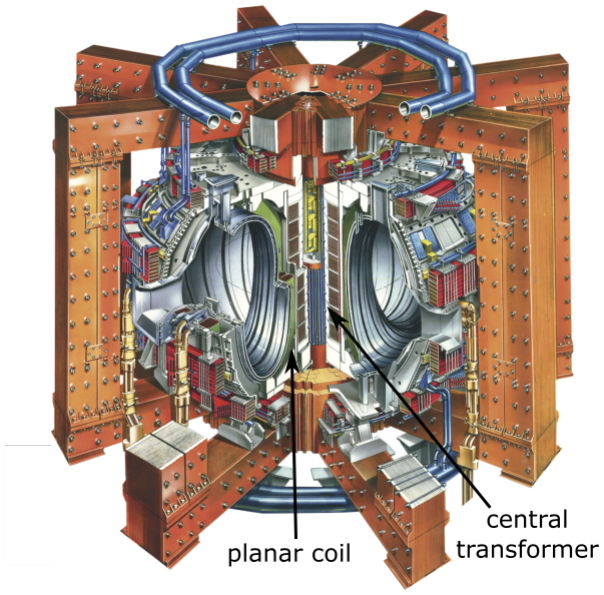
and T_e to be known. Article III in section 6.3 concerns precisely this case: Z_{eff} is inferred automatically with a Bayesian model within the Minerva framework, from data measured with a spectrometer collecting bremsstrahlung in the visible wavelength range at W7-X. According to the model, the bremsstrahlung emission can be calculated starting from n_e, T_e and Z_{eff}: if the first two are given, the third one can be inferred from the measurements. The Minerva inference runs in two stages: in the first one, the electron density and temperature is inferred from independent measurements with a Thomson scattering diagnostic [21]; in the second one, these parameters are used in a model which allows the inference of Z_{eff} from measured bremsstrahlung emission data.

The effort to achieve a good magnetic confinement of the plasma fuel resulted in the development of two main fusion device concepts. Both rely on a toroidal geometry, but differ from each other in the features of the magnetic field. They are known as tokamak and stellarator. Figure 2.1a and 2.1b show the W7-X stellarator [22] and the JET tokamak, respectively.

Each design has its own advantages and disadvantages. In order to confine the plasma in a toroidal geometry, the magnetic field has two components: one is oriented toroidally, following the large circular ring around the torus, encompassing the central axis; one poloidally, following the small circular ring around the surface. Figure 2.2 shows toroidal and poloidal directions in a torus. The resulting magnetic field is therefore twisted. In the tokamak, the toroidal component is generated by the external planar coils, whereas the poloidal component is generated by the plasma itself, carrying an electric current induced by a central transformer coil. The magnetic field in result is axi-symmetric. The advantage of this concept is that the planar geometry of the coils makes it relatively simple from an engineering point of view; on the other hand, the presence of a plasma current makes the plasma sensitive to a kind of instability which can lead to abrupt termination of the discharges, and the use of the central transformer poses a limitation to the duration of the pulses. In a stellarator, instead, the twisted magnetic field is generated entirely by the set of external coils, without recurring to a plasma current. In this way, stellarator discharges do not suffer from the same kind of instabilities of the tokamak, and, in principle, they can sustain long, steady-state plasmas. By generating an helical magnetic field with external coils, though, the toroidal symmetry that was present in tokamaks is broken. This has important consequences for plasma confinement: because of the complex, 3D geometry of the magnetic field, particle orbits are such that transport phenomena can lead to loss of particles. Nevertheless, for certain shapes and



(a) The Wendelstein 7-X stellarator. In red and orange are shown the superconductive non-planar and planar coils, respectively. They are operated in a low-pressure cryostat volume.



(b) The Joint European Torus tokamak. The set of planar coils is shown in green and the central transformer is visible in blue.

Figure 2.1.: The tokamak and the stellarator are the two main design concepts for a magnetic confinement fusion device.

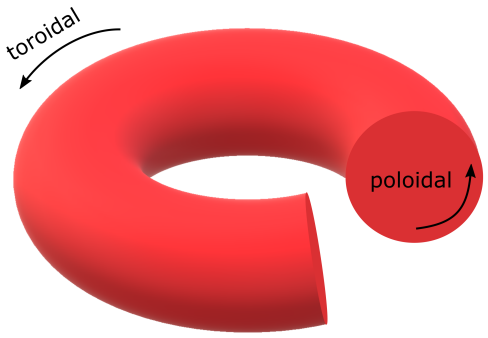


Figure 2.2.: Sketch of the toroidal and poloidal direction in a torus.

3D configurations of the magnetic field, this particle loss can be reduced [23–25]. When a stellarator is designed in such a way, it is said to be *optimized*. This is, indeed, the case of the W7-X stellarator. The optimization is carried out with computer codes and the coil shape resulting from it can be strongly irregular and, therefore, complex to realize from an engineering point of view. The Wendelstein 7-X stellarator was built with the mission to demonstrate the feasibility of steady-state pulses, where the deuterium or hydrogren plasma is confined for times up to $\approx$ 1800 s and certain plasma parameters are achieved and maintained for such long times. The experiment also aims at demonstrating the success of the optimization strategies. It started operation in 2015, and since then, two experimental campaigns were run, where high performance plasmas were sustained for up to $\approx$ 30 s. The main parameters of W7-X are summarized in table 2.1. In a hydrogen plasma discharge, a record for the triple product in a stellarator was achieved $nT\tau_E = 6.4 \cdot 10^{19}\,\mathrm{keVsm^{-3}}$ [26]. The JET tokamak was built with the objective of conducting experiments with plasmas in conditions approaching those expected in a thermo-nuclear fusion reactor. The main parameters of the machine are summarized in table 2.2 [27]. So far, it is one of the two machines in the world where experiments with a plasma made of a D/T mixture were conducted - the other one being the Tokamak Fusion Test Reactor (TFTR) in the United States [28]. The ex-

Quantity	Unit	Value
Plasma volume	m^2	30
Major radius	m	5.5
Minor radius	m	0.5
Magnetic field	T	2.5

Table 2.1.: The main parameters of W7-X.

perience and knowledge acquired through many years of experiments at JET are therefore crucial for the development of future experimental devices. The triple product measured in one of the D-T discharges (pulse no. 26148) was: $nT\tau_E = 3.8 \cdot 10^{20}\,\mathrm{keVsm^{-3}}$ [29].

We have seen that two important parameters in a fusion device are the density and temperature. Many diagnostic devices are devoted to the measurement of these parameters in W7-X and JET. Indeed, quantities like the density and temperature of the plasma electrons and ions are the key basic quantities required by more complex physics models and codes that are used to understand and predict the behavior of the plasma. It follows that it is crucial to be able to measure them accurately, and for this reason a lot of effort is put in the realization and operation of the diagnostic systems. These parameters, though, are not measured directly, but they are rather inferred from the direct measurements of other quantities, as, for example, spectroscopic emission. Therefore, being able to perform an accurate measurement is as important as being able to accurately infer the parameters from the measurements. Since no inference method is exact, it is necessary to be able to quantify the uncertainties of the inferred quantities. Bayesian inference provides the proper framework for handling uncertainties. It is particularly useful in this context because often signals measured by many different diagnostics carry information relevant to a common parameter, and Bayesian inference offers a method to account for different sources of uncertainties coming from different systems in a consistent, unified way. For this reason, most of the work presented in this thesis relies on Bayesian modeling for the inference of these parameters from diagnostic measurements.

Quantity	Unit	Value
Plasma volume	m^2	100
Major radius	m	2.96
Minor radius	m	1.25
Magnetic field	T	3.45
Plasma current	MA	3.2 - 4.8

Table 2.2.: The main parameters of JET.

Chapter 3.

Bayesian inference and modeling

Bayesian inference provides a formal method to make conclusions about a model hypothesis in light of real world observations. In this section, I will provide a short overview of Bayes theorem and how it is used in the Minerva modeling framework to handle inference in complex systems.

3.1. Bayes theorem

The process of Bayesian inference can be built from two constituents: the first one is the modeling of the problem and the definition of a priori hypothesis, together with the probability with which we believe the hypotheses and the model to be correct. The second one is the actual process of inference, the update of our knowledge about the hypotheses to incorporate the information brought in by new data as they are observed. The process is formally expressed by Bayes formula:

$$p(H|D) = \frac{p(D|H)p(H)}{p(D)} = \frac{p(D,H)}{p(D)} \tag{3.1}$$

where $p(H)$ denotes the *a priori* probability on the hypothesis H, $p(D|H)$ is the probability to observe the data D according to our model of the process, and is called the likelihood of H, and $p(H|D)$ is the updated *posterior* probability of the hypothesis H given the observation of the data D. The term $p(D,H)$ is the joint probability distribution of the data and the hypothesis; we will refer to it also as the joint distribution of the model. The denominator term $p(D)$ is called the *evidence*, and can be calculated as a normalization factor, integrating over

the free parameters in the numerator:

$$p(D) = \int p(D|H)p(H)dH \tag{3.2}$$

When modeling a physics system the probability $p(H)$ can be the probability that some physics quantities assume certain values, and they are known as the *free parameters* of the model, and the relation between D and H often can be expressed in a functional form, known as the *forward model*:

$$f = f(H) \tag{3.3}$$

The forward model allows us to generate predictions, and Bayes formula allows us to conduct a comparison to experimental measurement and update our prior beliefs about the free parameters accordingly. The likelihood function $p(D|H)$ is centered on the model predictions. For example, assuming a Gaussian distribution, it can be written as:

$$p(D|H) = \mathcal{N}(\mu = f(H), \sigma; D) \tag{3.4}$$

so that the Normal distribution is defined over the data space D, centered on the predictions made with the forward model $f(H)$, with uncertainties given by σ. The uncertainties, thus, are uncertainties in the prediction of the model and reflect the inability to model all aspects of a given experiment [30]. In Bayesian inference, measured data carry no 'error': they are given, and all sources of uncertainties are placed on the model. Under the traditional view, instead, there is the underlying assumption that data are generated in a random process, and we make use of probability theory to make estimates about the parameters underlying that process.

It is important to notice that a Bayesian model is defined not only by the choice of the free parameters H and forward function $f(H)$, but also by the choice of their corresponding distributions: the prior $p(H)$ and the likelihood $p(D|H)$. Therefore, two models differing only for the choice of the distributions should be regarded as two different models.

Considering now the case where a set of observables $D_1, ..., D_n$ are collected through different measurements in order to infer one common parameter, as it is often the case in physics experiments, we can see that Bayes formula offers a way to make use of the information consistently. The posterior for the parameter

H can be written as:

$$p(H|D_1, ..., D_n) = \frac{p(D_1, ..., D_n|H)p(H)}{p(D_1, ..., D_n)} \tag{3.5}$$

and if the measurement processes are independent $p(D_1, ..., D_n) = \prod_i p(D_i)$ and $p(D_1, ..., D_n|H) = \prod_i p(D_i|H)$. Each term D_i is associated to some forward model function $f_i(H)$, which allows to predict the corresponding observed data. Therefore, once models and assumptions are defined, estimating the uncertainties of a parameter from different sources of measurements occurs automatically by using Bayes formula. On the other hand, in conventional statistics approaches, it is not possible to assign a probability to an hypothesis and use the rules of probability theory to find the solution of an inference problem. Therefore, different kind of tests and methods, often known as the statistics and estimators, needs to be developed in order to relate observed data to unknown parameters and find corresponding uncertainties. This has led to different kind of schools, each one suggesting a different method or function to be used in the estimation of these quantities. This lack of clear underlying first principles is probably the main shortcoming of conventional statistics approaches, especially if it is to be applied in the context of complex heterogeneous systems. For this reason, Bayesian inference is particularly suitable when trying to solve the problem of scientific modeling and inference in complex systems, like experiments where different sources of measurements have to be taken into account simultaneously and in a consistent manner. A good example is provided by nuclear fusion experiments. In this field, Bayesian inference has been applied to the inference of plasma parameters from diagnostic data, as in the case of a Thomson scattering diagnostic at the W7-AS stellarator [31], a X-ray imaging diagnostic at the W7-X stellarator [8], a Lithium beam spectroscopy diagnostic at the JET tokamak [10], an electron cyclotron emission diagnostic at W7-X [9], the joint analysis of several profile diagnostics at ASDEX Upgrade [32] and W7-X [11, 12], and several tomographical problems from diagnostic measurements according to the maximum entropy formulation [33].

One concrete example of the advantage of performing inference with a forward model is given by the X-ray imaging diagnostic considered in article I (see chapter 6.1, reference [18]): this spectrometer observes impurity line emission along several lines of sight crossing the plasma volume. The measured spectra can then be used to infer the electron and ion temperature, impurity density and plasma rotation [8, 14]. In order to model the expected line of sight

integrated spectrum, the line emission is calculated at each point along the line of sight, taking into account its shape as a Voigt profile (the convolution of a Gaussian and Lorentzian shape), Doppler broadening and Doppler shift. When emission is integrated along the line, the resulting line shape can be different from a Gaussian due to the Doppler shift being different at different locations. In the conventional analysis [34], instead, the Doppler shift is found by fitting a Gaussian to the measured spectrum, which therefore relies on the inaccurate hypothesis that the observed line shape can be interpreted as a Gaussian. This kind of inaccuracy is easily avoided when the expected measurement is modeled through a forward model of the process.

3.2. Bayesian modeling in the Minerva framework

The Minerva framework has been developed with the idea of solving the problem of complexity arising from the handling of many sources of information and models in cases like large fusion experiments [1]. As we have seen, Bayesian inference offers the proper mathematical and probabilistic context to tackle the problem, and that is why Minerva is a framework for Bayesian modeling. It makes use of *Bayesian graphical models*, adopted from the computer science field [35]. A simplified sketch of the Minerva graph for a model of a system constituted, for example, of two generic different fusion diagnostics, with two free parameters m_1 and m_2, two forward functions f_1 and f_2 and observables D_1 and D_2 is shown in figure 3.1.

The data measured with both diagnostics carry information on parameter m_1 and can be used to infer it; moreover, data measured with diagnostic 2 can be used to make inference on parameter m_2. The colored nodes represent random variables, the white nodes represent deterministic, computational nodes. Such graphical representation is a way to visualize probabilistic relationships between random variables and is used to factorize the joint distribution of all graph variables according to the product rule of probability. The probability for each random variable can be written as a conditional probability conditioned only on the *parent* variables, i.e. the variables that directly or indirectly are linked to it through the arrows. In this case, the joint distribution of the graph

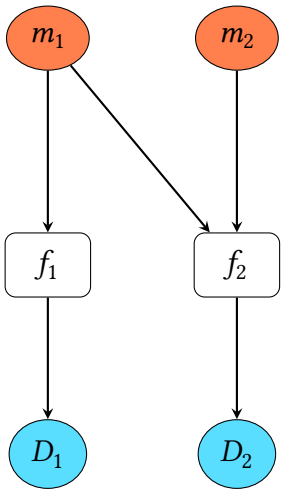


Figure 3.1.: Sketch example of a Minerva model graph for a systems constituted of two generic fusion diagnostics, modeled with free parameters m_1 and m_2, forward functions f_1 and f_2, and observables D_1 and D_2. Color nodes are probabilistic nodes representing random variables, white nodes represent deterministic, calculation nodes. Arrows express conditional probabilistic relationships between the variables: $P(D_2|m_1, m_2)$ and $P(D_1|m_1)$

.

can be written as:

$$p(m_1, m_2, D_1, D_2) = p(m_1)p(m_2)p(D_1|m_1)p(D_2|m_1, m_2) \tag{3.6}$$

Therefore, according to Bayes formula, the posterior distribution of the free parameters is:

$$p(m_1, m_2|D_1, D_2) \propto p(m_1)p(m_2)p(D_1|m_1)p(D_2|m_1, m_2) \tag{3.7}$$

As long as we are not interested in the exact values of the posterior distribution, the normalization factor at the denominator of Bayes formula can be neglected in most inference problems and this relation of proportionality is all that is needed. For example, the problem of finding the most likely values of m_1 and m_2 is an optimization problem that can be approached by finding the maximum of the right-hand side of equation 3.7 term with an optimization algorithm such as the Hooke and Jeeves pattern search [36]. The values found are known as the *Maximum A Posteriori* (MAP). The full answer in Bayesian inference

is anyway provided by the posterior distribution $p(m_1, m_2|D_1, D_2)$. Often, in practical applications, the functional form of the distribution is not needed to know the uncertainties of the free parameters. The uncertainties can be retrieved by looking at the distribution of samples drawn from the posterior, which can be obtained, for example, with a Markov Chain Monte Carlo (MCMC) based algorithm.

Although Bayesian inference offers a mathematically sound framework for performing scientific inference, its practical implementations often suffer from the burden of being computationally intensive. This is particularly evident in the case of the studies presented in article I (see chapter 6.1 and reference [18]) and IV (see chapter 6.4 and reference [20]). Article I concerns the inference of ion and electron temperature profiles from spectroscopic measurements of the X-rays emitted when the plasma interacts with impurity ions during a plasma shot at the W7-X stellarator. The forward modeling of this measurement procedure requires the calculation of the emission in each point along the collecting lines of sight crossing the plasma volume. This model is then used in iterative schemes in order to find the maximum of the posterior distribution of the desired parameters, or in order to sample from it with an MCMC algorithm. These schemes require iterating over such calculations hundreds or, in the case of MCMC, even million of times, making the inference process slow for complex models. Typically, the analysis time for a single measurement takes up to tens of minutes for a MAP estimate. In the context of large experiments, where tens of thousands of such measurements are collected during a single plasma shot, the conventional Bayesian inference can require long time scale to be carried out in a exhaustive manner. Something similar happens in the case of the study presented in Article IV. Here, spectroscopic measurements of light emitted in the interaction between plasma electrons and injected lithium atoms are used to infer the electron density at the edge of the JET tokamak. Once again, the forward model of this process involves complex calculations: the emission is calculated by solving the differential equation of a multi-state model at each point of the plasma volume that is observed through the collecting lines of sight. The model is then used in the inference process within iterative MAP or MCMC algorithms. Also in this case, finding the MAP solution requires tens of minutes for a single measurement. Analyzing the whole amount of data collected at several years of experiments carried out at JET would be very computational expensive. In both cases, we could accelerate the inference procedure by using machine learning algorithms, especially neural networks trained on the

information encoded in the Bayesian forward model. Neural networks allowed us to reduce the time to hundreds of microseconds for a single measurement, an improvement of several order of magnitude, which also opens the possibility of real time applications. Moreover, by developing this method in the context of the Minerva framework, we addressed another issue: namely, the fact that at each fusion experiment, several tens of different measurement devices collect data simultaneously during experiments. Each one, when modeled within the framework, can take advantage of such acceleration in an automatic manner. The method that allows us to train neural networks as approximations of Bayesian inference is in fact applicable to general Bayesian models. In the next chapter, we shall introduce and discuss the principles that we have originally developed in order to train neural networks a surrogate Bayesian models, and that we have called the *approximation framework*.

Chapter 4.

Approximate Bayesian inference with neural networks

Bayesian probability offers a theoretically clean and unified framework to perform inference. Real world implementations are less clean, suffering from the computational limitations of the chosen algorithms (Markov Chain Monte Carlo (MCMC), Maximum a Posteriori (MAP)). As already mentioned, one limitation is the computation time required to carry out the inference. The idea behind the work of this thesis is that this limitation can be overcome by using neural network models trained to approximate the full Bayesian inference, and to use them for fast, automatic computation. A network trained in this way can be called a *surrogate model*. In this chapter, I am going to outline the scheme that allows to train neural network surrogates to approximate an existing Bayesian model. For example, the network can be trained to approximate the inference of the MAP solution. In this case, the network function is trained to approximate the *inverse function* $f^{-1}(H)$, which maps the model predictions to the free parameter values maximizing the posterior. The network can then be evaluated on data measured at an experiment to infer the free parameter values in very short time scale: the data processing speed depends on the actual implementation of the network algorithm. Article I in chapter 6.1 shows an application where a convolutional neural network was run on a single GPU and could process a single measurement in $\approx$ 10 microseconds. The network was trained to predict relevant plasma parameters as ion and electron temperature from measured X-ray spectra. Another application is shown in article IV, where it was used to predict the electron density from spectra measured at the edge of the JET tokamak. It is important to notice that, although the network training can be very time consuming (several hours in most cases), it needs to be carried out

only once. The network can also be trained to approximate the forward function $f(H)$, in this way learning to reconstruct the model predictions given the free parameters. Afterwards, it can be used within one of the iterative schemes for the inference, as the MAP optimization, or the MCMC sampling. This option is not explored in this thesis and it is one of the aims of future works. Another possibility is to train the network to approximate the model joint distribution of the free parameters and observations $p(H, D)$. In this case, the network is given as input both the values of the free parameters and experimental measurements and learns to map them to the value of the joint distribution. Again, such network could be used to accelerate a MCMC sampling scheme. This possibility is investigated later in section 5.4, extending the work presented in article III (see chapter 6.3).

As it will be evident from the content of the next sections, the training data are generated from an existing Bayesian model with exactly the same procedure for all the training cases mentioned above. What differs in each case is the variables used during training as input and targets. All considered cases are instances of regression tasks for which a modeling function is already known.

I would like to emphasize that the training scheme was developed especially with one principle in mind: that it should be as general as possible, in the sense that it should require the least *ad hoc* manipulation for it to work in different contexts. From the theoretical point of view, this was possible because we developed the methods within the framework of Bayesian inference: as we have seen in chapter 3, it offers unified principles to perform scientific inference in general. This is achieved by first describing the scientific problem in terms of a Bayesian model - forward model and uncertainties quantified with probability distributions are its essential constituents; secondly, by applying the rules of probability theory in the form of Bayes formula in order to perform inference. All of this does not depend on the specific nature of the problem - e.g., physics, psychology, medicine experiments. From a practical point of view, generality was achieved by working within the Minerva Bayesian modeling framework, which offers a common language to write scientific models as computer codes and perform Bayesian inference with them. Independently of what they are models of, they are objects whose underlying probability distributions can be accessed in a single way. As we shall see, in order to train the network, we will only make use of the probability distributions defined by the models, so that the overall method is generally applicable to any scientific inference problem. The publications attached at the end of chapter 6 aim at demonstrating that this was

achieved, by showing that the method was successfully applied to two different physics systems of different fusion experiments. This aspect of the research is important because it allows the network surrogates to be used in a system for automatic scientific inference in complex systems of different nature.

A short remark on notation: in this chapter I am going to use bold symbols to denote vector variables, and plain text for scalar variables.

4.1. Neural networks

A neural network model represents a function f_{N} of the kind:

$$\mathbf{y} = f_{\mathrm{N}}(\mathbf{x}, \mathbf{w}) \tag{4.1}$$

where $\mathbf{x}$ denotes the network input vector and $\mathbf{w}$ denotes the network adaptable weights or parameters, and I have used $\mathbf{y}$ to denote the output vector. The function f_{N} is used to approximated an unknown function f through a composition of adaptive basis functions. The *architecture* of the network describes how the composition is performed. One of the simplest architecture is the *multi layer perceptron* (MLP) shown in figure 4.1. It was first introduced by Rosenblatt in the the late 50's in [37] as a model for the cognitive abilities of the brain, and still today it is at the foundations of more complex architectures.

Each circle is a *unit*. Each arrow represents a weight of the network. Each row of connecting arrows is called a *layer*. The units at the bottom represent elements of an input vector with four components. The units at the top represent the elements of a two dimensional output vector. The units in between are called hidden units. This kind of architecture represents a feedforward neural network, because the edges are directed from the input towards the output, without feedback loops. Each hidden unit stands for the application of a nonlinear *activation function* ϕ to a linear combination of the output and weights of the previous layer. The overall network function is found by recursively applying these functions layer by layer until the output is reached. For the MLP in the figure, the *i-th* component of the output vector can be written as:

$$y_l = \sum_k w_{kl}^{(3)} \phi_k^{(2)} \Big(\sum_j w_{jk}^{(2)} \phi_j^{(1)} \Big(\sum_i w_{ij}^{(1)} x_i \Big) \Big) \tag{4.2}$$

where $\phi_i^{(l)}$ denotes the activation function of unit i applied to the linear combination of the weights $w_{ij}^{(l)}$ (l) of layer (l). The activation function is usually

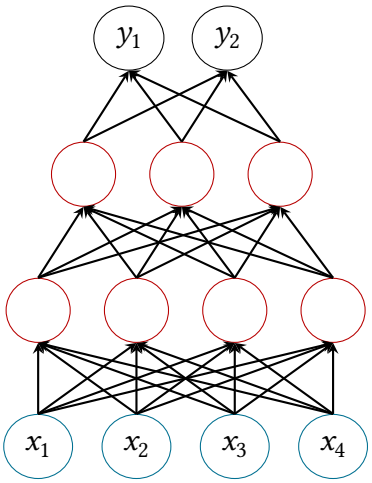


Figure 4.1.: The architecture of the multi layer perceptron (MLP), a feedforward neural network, with one hidden layer of weights.

the hyperbolic tangent, the logistic function, or the rectified linear unit, defined as $\mathrm{ReLU}(x) = \max(0, x)$. Hyperbolic tangent and logistic functions belong to a class of functions, known as sigmoid functions, which were originally chosen by inspiration from the pattern of activation of the neurons in the human brain. More recently, the ReLU function was found to be a better model from the biological point of view [38], and it became particularly popular in the deep learning community because it improved neural network training [39]. We have chosen to use the ReLU activation function in our application because of its recognized widespread success in many different problems. During training, the weights of the network are adapted to optimize a *loss function* $E(\mathbf{x}, \mathbf{w})$, that for regression problems is often chosen to be the mean square error between the network output and the training targets $\mathbf{t}$:

$$E(\mathbf{x}, \mathbf{w}) = \frac{1}{2N} \sum_n \sum_k (y_k(\mathbf{x}^n; \mathbf{w}) - t_k^n)^2 \tag{4.3}$$

where n is now an index labeling each of the N samples in the training data set, and k labels the k-*th* component of the output and target vectors. The choice of this error function can be motivated by the principle of maximum likelihood [40] where the distribution of the target data conditioned on the

input is given by a Gaussian distribution:

$$p(t_k|\mathbf{x}) = N(\mu = y_k(\mathbf{x};\mathbf{w});\sigma) \tag{4.4}$$

with the mean given by the network output and some standard deviation σ. The overall likelihood function for the training set is $\mathcal{L} = \prod_n p(\mathbf{t}^n|\mathbf{x}^n)p(\mathbf{x}^n)$. The error function is then found to be $E = -\ln(\mathcal{L})$ by neglecting all terms which do not depend on the weights $\mathbf{w}$, as it is the case for the term $p(\mathbf{x}^n)$. It can also be shown [40] that the error function in equation 4.3 can be written as:

$$\begin{aligned} E = &\frac{1}{2}\sum_k \int \{y_k(\mathbf{x};\mathbf{w}) - \langle t_k|\mathbf{x}\rangle\}^2 p(\mathbf{x})\, d\mathbf{x} \\ &+ \frac{1}{2}\sum_k \int \{\langle t_k^2|\mathbf{x}\rangle - \langle t_k|\mathbf{x}\rangle^2\} p(\mathbf{x})\, d\mathbf{x} \end{aligned} \tag{4.5}$$

and therefore its minimization occurs when the first term on the right-hand side is zero, since the second term does not depend on the network weights. We can then write:

$$y_k(\mathbf{x};\mathbf{w}^*) = \langle t_k|\mathbf{x}\rangle = \int t_k p(t_k|\mathbf{x})\, dt_k \tag{4.6}$$

where $\mathbf{w}^*$ denotes the weight vector found with the minimization. This equation says that the network function is given by the conditional average of the target data, where the conditioning is on the input vector. The second term can be written as:

$$\frac{1}{2}\sum_k \int \sigma_k^2(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \tag{4.7}$$

where σ_k^2 denotes the variance of the target data:

$$\begin{aligned} \sigma_k^2(\mathbf{x}) &= \langle t_k^2|\mathbf{x}\rangle - \langle t_k|\mathbf{x}\rangle^2 \\ &= \int \{t_k - \langle t_k|\mathbf{x}\rangle^2\} p(t_k|\mathbf{x})\, dt_k \end{aligned} \tag{4.8}$$

which says that the residual error of the training is given by the variance of the target data at the given input vector $\mathbf{x}$. The results in equation 4.6 and 4.8 will be useful in the next section, where I will show how the trained network can be seen as an approximation of a Bayesian model.

I believe it is worth to credit here the software libraries that allowed me to implement machine learning models as efficient computer codes. Nowadays, many framework are available. Since the start of my work in 2016, I have been using manly the Theano framework [41] (for the work in article I, section 6.1, and II, section 6.2), and the TensorFlow framework [42] (for the work in article IV, section 6.4) once the Theano development team announced suspending maintaining it.

4.2. The approximation framework

In this section, I am going to describe the novel method that we have developed to approximate Bayesian inference with neural networks. We have called the set of principles allowing to generate the approximate network model the *approximation framework*. I will start by describing the algorithm used to generate the training data. It is important to realize that the training set is generated entirely from the Bayesian model, without making use of external data, e.g. experimental data. This aspect is relevant from the point of view of the automation of the procedure - the Bayesian model is the only requirement, and in this way the procedure is applicable, in general, to any model. Also, we guarantee that the network is trained on the same assumptions that the original model is based on. The algorithm is quite simple: given a Bayesian model $\mathcal{G}$ with free parameters $\mathbf{m}$ and observables $\mathbf{d}$, then the joint distribution is $p(\mathbf{m}, \mathbf{d}) = p(\mathbf{d}|\mathbf{m})p(\mathbf{m})$, where $p(\mathbf{d}|\mathbf{m})$ is a chosen likelihood function, and $p(\mathbf{m})$ the prior distribution. Bayes formula for model $\mathcal{G}$ is then:

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d})} \tag{4.9}$$

The training data are generated by drawing N samples of the variables $(\mathbf{m}, \mathbf{d})$ from the joint distribution $p(\mathbf{m}, \mathbf{d})$. Note, that by sampling from $p(\mathbf{m}, \mathbf{d})$, we sample both from $p(\mathbf{m})$ and $p(\mathbf{d}|\mathbf{m})$, which implies that the observables $\mathbf{d}$ are noisy samples of the forward model prediction, where the features of the noise are specified by the likelihood function. The network then can be trained to learn the mapping from one set of variables to the other:

$$\begin{aligned} f &: \mathbf{m} \to \mathbf{d} \\ g &: \mathbf{d} \to \mathbf{m} \\ h &: (\mathbf{m}, \mathbf{d}) \to p(\mathbf{m}, \mathbf{d}) \end{aligned} \tag{4.10}$$

the mapping denoted by f corresponds to the forward model function of the model, g corresponds to its inverse, and h corresponds to the mapping from the joint space of the free parameters and observables $(\mathbf{m}, \mathbf{d})$ to the joint distribution of the model $p(\mathbf{m}, \mathbf{d})$. I will describe now in details only the case of the learning of mapping g, which is the one considered in articles I, II and IV in section 6.1, 6.2, 6.4, respectively. The case of mapping h will be considered in section 5.4. The case of mapping f is not considered in this thesis, and it will be object of future work.

In the case of mapping g, the network is trained to reconstruct the parameters of a physics model, given the observed quantities. Therefore, during training the input to the network are the samples of the observable quantities $\mathbf{d}$, and the targets are the free parameter samples $\mathbf{m}$; at evaluation time, the input are experimental measurements and the output the reconstructed parameters. For example, in a nuclear fusion experiment, this could be the case of inferring some key plasma profiles, such as plasma electron or ion temperature and density profiles, given some diagnostic measurements, for example spectroscopic emission (as it is the case in article I and IV). To see how the trained network relates to the Bayesian model, we can make use of the result shown in equation 4.6, which says that the network mapping is given by the conditional average of the target data, conditioned on the input. Since the training data have been obtained by sampling from the model, the target data t_k in equation 4.6 correspond to the sampled free parameters m_k, and the input data $\mathbf{x}$ correspond to the sampled observable quantities $\mathbf{d}$. Therefore we have:

$$y_k(\mathbf{d}; \mathbf{w}^*) = \langle m_k | \mathbf{d} \rangle = \int m_k \, p(m_k|\mathbf{d}) \, dm_k \tag{4.11}$$

where we notice that the conditional distribution $p(m_k|\mathbf{d})$ corresponds to the posterior of the Bayesian model $\mathcal{G}$ in equation 4.9. Under the conditions for which the relation in equation 4.6 is valid, we can say that the trained network learns a mapping that is given by the conditional average of the true posterior of the Bayesian model. These conditions are satisfied only in the ideal case of networks with number of weights and training data samples approaching infinity, and optimal minimization of the error function of equation 4.3; nevertheless, the relation in equation 4.11 provides an interpretation of the network function in relation to the original posterior distribution of the model $\mathcal{G}$. A similar argument holds for the residual error of the training, which was formulated in terms of the variance of the target data in equation 4.8, and now can be written

as:

$$\sigma_k^2(\mathbf{x}) = \int \{m_k - \langle m_k|\mathbf{d}\rangle^2\} p(m_k|\mathbf{d})\, dm_k \tag{4.12}$$

which says that the residual error is given by the variance of the true posterior distribution of the model $\mathcal{G}$.

Another insight is provided by considering the evidence term $p(\mathbf{d})$ in Bayes formula of equation 4.9. The training data are samples from the joint distribution $p(\mathbf{m};\mathbf{d})$ of the model $\mathcal{G}$. If $\mathbf{d}$ is fixed, by considering the distribution of the free parameters for the given data $\mathbf{d}$, we obtain the posterior distribution $p(\mathbf{m}|\mathbf{d})$ (except for the normalization factor $p(\mathbf{d})$). Similarly, by looking at the samples $\mathbf{d}$ letting $\mathbf{m}$ vary, we realize that they constitute samples from the evidence $p(\mathbf{d})$. In the limit of a large training data set, the distribution of the observable variables $\mathbf{d}$ is the evidence $p(\mathbf{d})$ of the model $\mathcal{G}$. Therefore, the evidence could in principle be estimated from the training samples. This can have different interesting implications, since the evidence is involved in higher order applications of Bayesian inference, for example for the problem of model selection. I will not investigate this possibility further, since it does not fall in the scope of the thesis. Instead, I will describe shortly another implication directly related to the neural network training, which has been investigated in article I in section 6.1. The distribution $p(\mathbf{d})$, being obtained by integration over the prior distribution, represents the probability that all data possibly described and predicted by model $\mathcal{G}$ assume certain values. Therefore, for a novel measured data point $\mathbf{d}^*$ having a low probability under the evidence, i.e. $p(\mathbf{d}^*) \ll 1$, we can say that the model $\mathcal{G}$ does not provide an adequate description. This can happen because of an inadequacy in the prior distributions, the forward model function or the likelihood function. The evidence, therefore, provides useful information about when a different model should be used to better describe the experimental measurements. This is particular relevant to the case of this study, since all the data used for training the network are generated 'blindly' through a Bayesian model. One common cause for a network performing well on training data and poorly on experimental data is, indeed, a poor model description of the measurements, which leads to the network being used to make predictions in a region of the input space not covered during training. Features of the model can then be changed accordingly by, for example, tuning the parameters controlling the prior distributions. It is important to notice that this sort of tuning, if carried out properly, does not generate an overly complex, specific model,

because the evidence term inherently penalizes for higher complexity through an Occam razor principle [43]. In this way, a model $\mathcal{G}^*$ can be found, such that $p_{\mathcal{G}^*}(\mathbf{d}^*) > p_{\mathcal{G}}(\mathbf{d}^*)$ and it can be used to generate new training data from its joint distribution $p_{\mathcal{G}^*}(\mathbf{m}, \mathbf{d})$ with the same scheme used for $\mathcal{G}$. In conclusion, given the training data, it is in principle possible to reconstruct the evidence, or an approximation of it, through any of the existing machine learning methods for probability density estimations, and use it to assess the quality of the training data in order to analyze, or prevent, network failures. In article I, a simpler method which does not rely on the reconstruction of the probability density, but makes use of a distance-based algorithm, is used to effectively improve the quality of the training data.

Chapter 5.

Application to nuclear fusion research

The approximation framework formulated in the previous chapter has been used in the context of nuclear fusion experiments. I would like to remark, though, that such formulation is not restricted to a specific system: indeed, no assumption was made on the internal features of the model or the data involved. The only requirement for a network to be trained in this way is the existence of a Bayesian model. This highlights one key strength of this approach, mentioned already in the introduction: it can be used to automatically generate neural network approximations of Bayesian models and inference. I have tried to demonstrate this by applying it to different measuring systems that are operated at different nuclear fusion experiments. In particular, I have taken advantage of the implementation of the models in the Minerva framework, which, by abstracting away their details, has simplified the replication of the training data generation procedure, ultimately opening up the possibility of full automation. Neural networks have been used in nuclear fusion research starting already in the 90's with applications involving the automatic analysis of JET charge exchange spectra [44], and the usage of synthetic data together with experimental data in the training procedure [45]. Real time applications were also considered, as shown in [46]. More recently, neural networks have been used at the Wendelstein 7-X stellarator with the purpose of protecting plasma facing components by providing a reconstruction of magnetic configuration properties from heat load patterns [47, 48], at the JET tokamak for the acceleration of tomographic reconstruction [49], and also as approximations of transport models [50–52].

In this chapter, I will introduce the work that constitutes the main result of this doctoral thesis, and which is fully elaborated in the published articles

available in chapter 6. Section 5.1 concerns publications I and II, section 5.2 concerns publication IV and section 5.3 concerns publication III. These sections are meant to provide a general overview of the work that is described with all details in the corresponding publications. Because of the fact that it features more details, section 5.4, instead, might look out of place to the reader. The reason for being like this is that it provides an extension of the work presented in publication 5.3, which has not yet been published in a scientific journal. In fact, the reader might notice that I have tried to structure the content of this section in a way that is similar to publications I and IV.

In general, the applications considered here concern one of the mappings appearing in equation 4.10. Section 5.1 and 5.2 concern mapping g, where the network is trained on the inference of a Bayesian model free parameters $\mathbf{m}$ from observable data $\mathbf{d}$. Section 5.3 and corresponding publication III themselves do not directly concern one of the mappings, rather they describe the Bayesian model and the experimental measurements which are used in section 5.4 to train a network on mapping h, corresponding to the inference of the model joint probability distribution $p(\mathbf{d}, \mathbf{m})$ from the joint space $(\mathbf{d}, \mathbf{m})$. As already mentioned elsewhere, the learning of mapping f, corresponding to the reconstruction of model prediction from free parameter values, is not considered in this thesis, and it is left for future investigations.

5.1. Inference of ion and electron temperature profiles at W7-X

The first application I have considered is the inference of ion and electron temperature profiles from the measurement of spectroscopic data from an X-Ray imaging crystal spectrometer (XICS) diagnostic [14] at W7-X (see article I, section 6.1). The X-rays are emitted in the atomic physics interaction between argon ions (Ar^{16+}), which are injected as neutral gas, and plasma electrons. The light is collected across several lines of sight, diffracted by a quartz crystal and imaged on a 2D detector with energy resolution in one direction and spatial resolution in the other. The lines of sight span covers a large part of the plasma cross section, so that the observed spectra carry information about the plasma parameters in the entire region. Figure 5.1a shows a sketch of the line of sight span on the bean shaped cross section, and figure 5.1b shows a measured raw

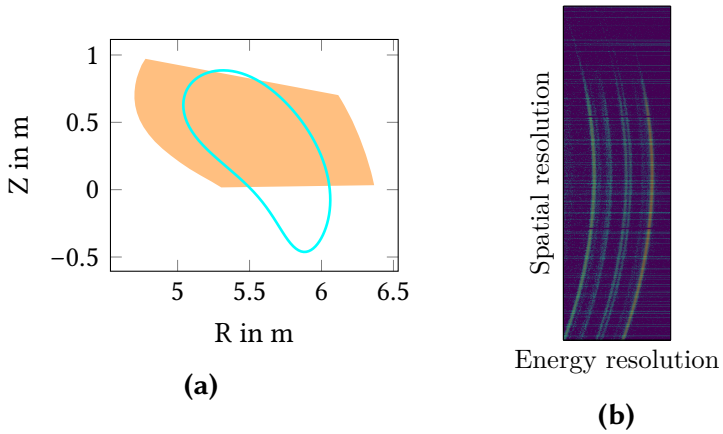


Figure 5.1.: (a) Sketch of the XICS system view on the bean shaped cross section at W7-X. The span of the lines of sight is represented by the colored area. (b) A raw image measured on the detector. The bright curved features denote the observed line emission.

image. The bright curved features in the image represent the observed line emission. From the broadening of the recorded line emission it is possible to infer ion temperature across the plasma, whereas from the ratio between different line intensities it is possible to infer electron temperature profiles. A neural network was trained with data generated by sampling from the joint distribution of a pre-existing Minerva Bayesian model of the diagnostic [13], and evaluated on measurements collected during the first experimental campaign at W7-X. The model allows to predict the measured spectrum along several lines of sight by calculating the line emission occurring in the atomic processes involving ionized argon impurities appositely released in the plasma, and the plasma species. The neural network prediction also includes uncertainties, calculated within a Bayesian framework for the network training, known as Bayesian neural networks [53]. It relies on the Laplace approximation of the network weight posterior, and it is extended here to account for the presence of noise in the input data and multiple minima in the training procedure. In order to achieve this, multiple networks were trained with different, random starting values for the weights, and the overall prediction was obtained with a Monte Carlo sampling scheme occurring both in input and weight space of different networks. This allowed to relax some of the simplifying assumptions required by the conventional Bayesian neural network framework. Article II in section

6.2 describes this procedure in full detail. The profiles inferred with the network from the measurements were compared to those inferred with the conventional Bayesian inference in a restricted number of cases. The comparison, although limited to the small amount of available data at the time of the start of the new W7-X experiment, showed in general good agreement between the two methods. In order to assess the quality of the training data as a description of the experimental measurements, a k-nearest neighbor algorithm was used to estimate the distance between a measured data point and the training data points. Under the assumption that the distance is inversely proportional to the probability, this calculation can be used to estimate how likely a given measurement is under the distribution of the training data. This method proved useful in identifying the reason behind the failed reconstruction of the profiles by the network, leading to the choice of a different Bayesian model in which the hyper-parameters of the prior distributions were varied so to generate a training set that better described the expected measurements. When evaluated on a GPU, the network could reconstruct a profile for a single measured data point in tens of microseconds, whereas the inference of the most likely profile with the Minerva model requires tens of minutes on a CPU.

5.2. Inference of edge electron density profiles at JET

In the second application, I have considered an entirely different problem: a different physics systems of a different diagnostic at the JET tokamak. The aim was to test the generality and validate the results of the approximation framework and training scheme. The network was trained to infer edge electron density profiles from measurements of a Lithium-beam spectroscopy diagnostic [15] at JET. Lithium atoms are injected vertically in the machine, and as they penetrate the plasma volume, different excited states are populated by the interaction with the plasma species. The lithium line emission collected by a spectrometer comes from the de-excitation of the first excited state into the ground state. The light is emitted at different vertical spatial locations, and its intensity depends on the electron density at those locations. Figure 5.2 shows a sketch of the system, where the orange line encloses the plasma volume and, on the right, the spectrometer is represented by a box. Article IV in section 6.4 describes the

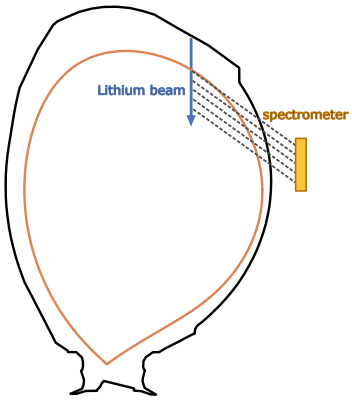


Figure 5.2.: Sketch of the lithium beam system at JET. The beam is injected vertically from the top of the machine and the emission is collected across multiple lines of sight by a spectrometer. The black contour represents the machine boundary, the orange one encloses the plasma volume.

work in full detail. The training of the network occurred in the usual way: the training data were generated sampling from the joint distribution of an existing Bayesian model of the diagnostic implemented within the Minerva framework. In this case, the spectral measurements were described by a multi-state atomic model, which allows to relate the intensities of the lines emitted by Lithium atoms through electronic excitation and spontaneous emission to the density of the plasma electrons [10]. The availability of a larger amount of measured data collected at the experiments through many years of JET operation allowed for a more extensive comparison with the conventional Bayesian inference. Approximately sixty experiments carried out in a wide range of plasma conditions were considered. The comparison was made between the experimental measurements and the observations reconstructed with the network and Minerva inferred profiles. In general, the Minerva inferred profiles could predict the measurements with a lower error, nevertheless the error in the network prediction was consistently below 20% across the entire data set. Also in this case, the network inference provided full uncertainties. In contrast to the W7-X application, the uncertainties were calculated within a Bayesian framework relying on variational inference and the state-of-the-art deep learning technique known as *dropout*, which allows to approximate the network weight posterior [54]. This method has the advantage of requiring little extra computations besides that of a standard forward pass through the network, and therefore is suitable for

real-time applications.

5.3. Inference of the effective ion charge and electron density and temperature profiles at W7-X

The third application involved the development of the automatic Bayesian inference of the plasma effective ion charge Z_{eff} from measurements of visual bremsstrahlung at W7-X. This application is described in article III in section 6.3 [55]. Z_{eff} is a quantity which gives an indication of the level of impurity contamination of the plasma. This can be harmful in two ways: a too large amount of low-Z impurities can lead to fuel dilution, whereas a too large amount of high-Z impurities can lead to high radiation losses. On the other hand, in a reactor, a controlled amount of impurities is desired in certain regions of the plasma, because the radiation loss that they cause can help distributing the heat coming from the plasma more homogeneously over the plasma-facing components. Therefore, being able to measure and control impurities is crucial. The measurements were collected with a single line-of-sight spectrometer during several experiments. The spectrometer collects light in a broad visible range, approximately from 300 nm to 1000 nm. In this wavelength region several lines are present together with the bremsstrahlung dominated background. The bremsstrahlung emission comes from the interaction between the plasma electrons and the ions present in the plasma. For example, one of the impurity frequently contaminating the plasma in W7-X is C^{6+}, as graphite is the material of some in-vessel components. The bremsstrahlung emission $V(\lambda)$ at a certain wavelength λ directly depends on Z_{eff} according to the following equation:

$$V(\lambda) = g_{\text{ff}}(Z_{\text{eff}}, T_{\text{e}}, \lambda) \frac{n_{\text{e}}^2 Z_{\text{eff}}}{\sqrt{k_b T_{\text{e}}}} \exp\left(-\frac{hc}{\lambda k_b T_{\text{e}}}\right) \frac{1}{\lambda^2} \tag{5.1}$$

where $g_{\text{ff}}(Z_{\text{eff}}, T_{\text{e}}, \lambda)$ is the free-free Gaunt factor modeled according to [56], and the remaining symbols are used in the conventional way referring to the respective physics constants in SI units. Therefore, given the observed spectra, it is possible to infer its value averaged along the line of sight. The inference is based on a model which allows to predict the expected bremsstrahlung emission in a given wavelength range from independent measurements of the electron

density n_e and temperature T_e profiles. In this case, these measurements were provided by a Thomson scattering diagnostic [21] and the profiles were inferred with a Bayesian model as well. This model makes use of Gaussian processes to model the profile function and inference was performed under the framework of Bayesian model selection in order to find the optimal trade-off between complexity of the function and a good fit to the data. This is accomplished by comparing different models through the evidence term in the denominator of Bayes formula. It can be shown, indeed, that it embodies an Occam razor principle [57]. The inference of the effective ion charge, and the required electron temperature and density profiles, was running routinely after each plasma discharge. The inferred Z_{eff} values were compared to a preliminary measurement of the carbon impurity concentration from a charge exchange recombination spectroscopy diagnostic [58, 59], showing encouraging consistency.

5.4. Learning the model joint distribution

In the two applications described in sections 5.1 and 5.2, the network was trained to learn the mapping from model observable data $\mathbf{d}$ to free parameters $\mathbf{m}$. This corresponds to the mapping g in equation 4.10. Here, I would like to consider a different case: the mapping from the model joint space $(\mathbf{d}, \mathbf{m})$ to the logarithm of model joint probability $\log p(\mathbf{d}, \mathbf{m})$, corresponding to mapping h in equation 4.10, taking as a starting point the work presented in the previous section and article III. The model considered here is the one developed for the inference of Z_{eff} from line integrated measurements of visible bremsstrahlung emission. As usual, the method we employ to train the network is general and valid for other Bayesian models. The reader interested in understanding how the measurements are performed, and further details about the physics models is recommended to first read article III. We will take for granted most of the information provided in the article and here extend the discussion to the learning of the model joint distribution. The goal is to train a neural network so that, given as input values of Z_{eff}, electron density n_e and electron temperature T_e independently measured with a Thomson scattering diagnostic [21], and bremsstrahlung measurement V_λ, the network is able to predict the logarithm of the model joint distribution values $\log\big(p(Z_{\text{eff}}, V_\lambda)\big)$. The logarithmic value is used because of the wide range of values typically covered by $p(Z_{\text{eff}}, V_\lambda)$. The training is carried out exclusively on data generated with the Minerva Bayesian

model, which I will describe in the next paragraphs. The trained network can be useful in the context of any optimization or sampling problem which requires the calculation of the joint probability value: as in the case of the optimization of the posterior distribution to find its maximum - known as maximum a posteriori (MAP), or the sampling from the posterior distribution with a Markov chain Monte Carlo (MCMC) algorithm. Commonly, a MAP optimization requires from hundreds to thousands of iterations before convergence is reached, and a MCMC can require millions of iterations before the posterior distribution is properly approximated. Therefore, the trained network can be used to significantly speed up these iterative schemes, which otherwise would require the repeated computationally expensive calculation of the full forward model.

5.4.1. The Bayesian model of bremsstrahlung within Minerva

Given values of Z_{eff}, electron temperature T_e and density n_e, the model allows to predict the expected bremsstrahlung emission observed with the spectrometer. The model's only free parameter is the effective ion charge Z_{eff}. T_e and n_e are inferred with a Minerva model from an independent Thomson scattering measurement. A sketch of the model graph implemented in the Minerva framework is shown in figure 5.3.

Colored orange and blue nodes represent probabilistic nodes, prior probability of Z_{eff} and likelihood function of the model parameter given the observed emission, respectively. White nodes represent either a parameter known at inference time, as for the n_e and T_e nodes, or a calculation node as the bremsstrahlung emission node. The latter one summarizes all the calculations needed to calculate the expected measurements, given values of Z_{eff}, n_e and T_e, which include, for example, the line integration of the calculated emission along the line of sight and the application of required calibration coefficients, as described in article III. We shall denote this forward function as $f_m(Z_{\text{eff}}, n_e, T_e)$. The observations are constituted by the bremsstrahlung emission V_λ detected at 15 spectrometer channels in the wavelength range of $\approx 627 - 641$ nm. The joint probability distribution for this model can be written as:

$$p(Z_{\text{eff}}, V_\lambda) = p(Z_{\text{eff}})p(V_\lambda|Z_{\text{eff}}; n_e, T_e) \tag{5.2}$$

where $p(Z_{\text{eff}})$ is the Z_{eff} prior probability chosen as a uniform distribution between 1 and 6 (assuming that carbon is the dominant impurity in a hydrogen

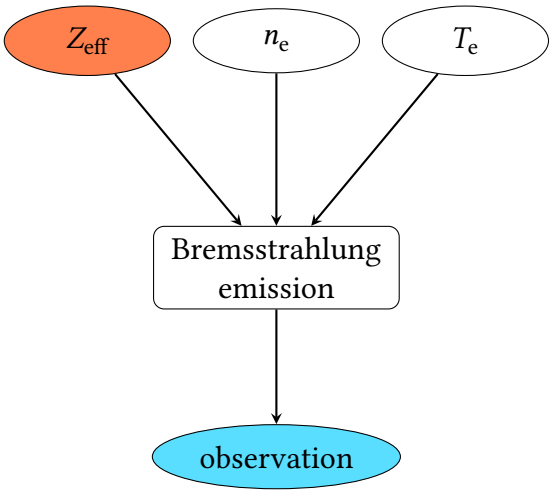


Figure 5.3.: Simplified visualization of the Bremsstrahlung emission model graph implemented within the Minerva framework. The orange node represents the prior of the model free parameter Z_{eff}, and the two white nodes n_e and T_e represent electron density and temperature. The Bremsstrahlung emission node represents a calculation node to predict the expected measured signal. The observed quantities are represented by the probabilistic blue node at the bottom, which stands for the likelihood function of the model.

plasma):

$$p(Z_{\text{eff}}) = \mathcal{U}(1, 6) \tag{5.3}$$

and the likelihood function $p(V_\lambda | Z_{\text{eff}}; n_e, T_e)$ is chosen as a Gaussian distribution centered at the model prediction f_m and with standard deviation $\sigma = 0.2 f_m$:

$$p(V_\lambda | Z_{\text{eff}}; n_e, T_e) = \mathcal{N}(\mu = f_m, \sigma = 0.2 f_m) \tag{5.4}$$

Figure 5.4 shows an example of the quantities used in the model. The top row shows a randomly generated n_e and T_e profile in the left and right plot, respectively. These are samples from the prior distributions: samples drawn from them might not look realistic, lacking the features commonly seen in observed plasma profiles. This is the case of the profiles shown in the figure, but the priors are broad enough to contain both 'realistic' and less 'realistic' instances of the plasma profiles (see also figure 5.5). The profiles are expressed as function of the so called effective radius coordinate ρ. This is a coordinate that extends from

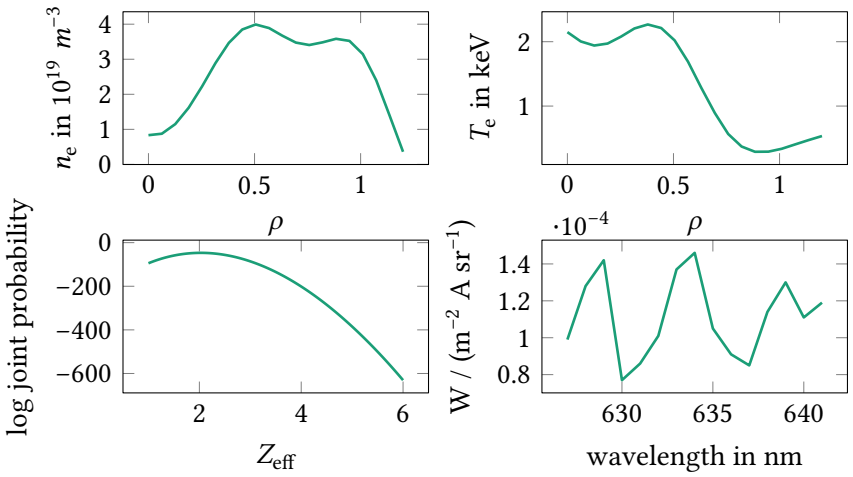


Figure 5.4.: An example case of the forward model calculation. In clockwise direction, from the top left plot the following quantities are shown: a random sample of electron density profile, a random sample of electron temperature profile, the logarithm of the joint probability of the model, and the observed simulated bremsstrahlung emission with added Gaussian noise from the error model. The profiles are sampled from their corresponding prior distributions: samples drawn from them might not look realistic, lacking the features commonly seen in observed plasma profiles, as it is the case for those shown here.

the plasma core at $\rho = 0$, towards the plasma edge at $\rho \approx 1.0$. The bottom row shows, in the left plot, the logarithm of the joint probability $\log\big(p(Z_{\text{eff}}, V_\lambda)\big)$ for $Z_{\text{eff}} \in [1, 6]$ and fixed value of V_λ, and, in the right plot, the bremsstrahlung emission V_λ calculated with the forward model, with added Gaussian noise from equation 5.4. Note that the maximum of the joint probability is found for $Z_{\text{eff}} \approx 2.0$, which is the value used to generate the data in the bottom-right plot.

In order to learn to approximate the model joint distribution, the network needs to take as input n_e, T_e, V_λ and Z_{eff}, and give as output $\log\big(p(Z_{\text{eff}}, V_\lambda)\big)$. We will generate the training data by sampling from the probability distributions assigned to each of the input quantities. The distributions for Z_{eff} and V_λ are given in equations 5.3 and 5.4. The electron density and temperature profile probability distributions are modeled with a 1D Gaussian process (GP) [60]. A GP is used to model the correlation between the values that a function assume

in its domain. A Gaussian distribution is chosen for the values of the function evaluated at a fixed number of domain locations (a grid), and its covariance is determined by the GP covariance function, which, through its parameters, controls the correlation between the function values on the grid points. One common choice for the covariance function is the squared exponential:

$$K(z_1, z_2) = \sigma_f^2 \exp\left(-\frac{(\rho_1 - \rho_2)^2}{2\sigma_x^2}\right) + \delta_{ij}\sigma_y^2 \tag{5.5}$$

where ρ_1 and ρ_2 are two locations along the effective radius, and the σ parameters influence the smoothness of the profile: σ_f regulates the overall variance of the profile, σ_x regulates the length scale of the profile variation in its domain. Small σ_x values imply quickly varying profiles, whereas larger values imply smoother, slower changing profiles. σ_y regulates the amount of noise expected in the profile. The following parameter values were used for the n_e GP covariance function:

$$n_e : \begin{cases} \sigma_f = 5.0 \cdot 10^{19}\,\mathrm{m}^{-3} \\ \sigma_x = 0.3 \\ \sigma_y = 0.005 \cdot 10^{19}\,\mathrm{m}^{-3} \end{cases} \tag{5.6}$$

and for T_e:

$$T_e : \begin{cases} \sigma_f = 2.0\,\mathrm{keV} \\ \sigma_x = 0.3 \\ \sigma_y = 5.0 \cdot 10^{-3}\,\mathrm{keV} \end{cases} \tag{5.7}$$

Figure 5.5 shows 20 samples drawn from the two distributions. The figure also shows a low value constrain being applied at the edge $\rho \approx 1.2$, where the n_e and T_e profiles are expected to assume values of $0.1 \pm 0.5 \cdot 10^{19}\,\mathrm{m}^{-3}$ and $0.1 \pm 0.5\,\mathrm{keV}$, respectively. The number of grid points along ρ is fixed to 20 for both profiles, equally spaced between 0 and 1.2.

5.4.2. Architecture and specifications of the network

A total number of 10^6 samples for each of the input quantities n_e, T_e, V_λ and Z_{eff} and the corresponding target joint probability distribution $\log\left(p(Z_{\mathrm{eff}}, V_\lambda)\right)$

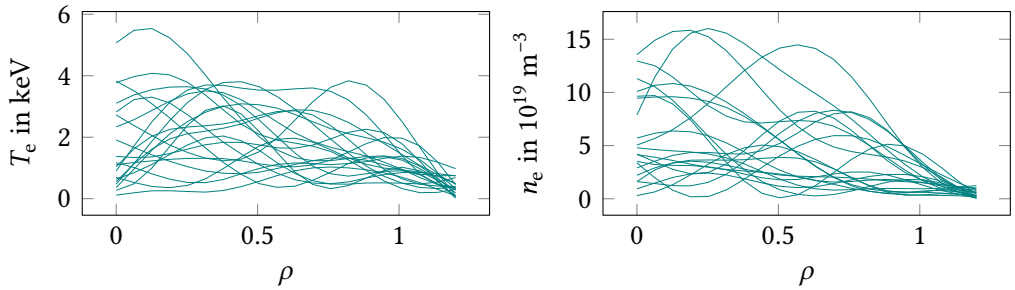


Figure 5.5.: Samples from the Gaussian process probability distributions of the T_e and n_e profiles. Note how the profile shape is in general smooth and not monotonic. A low value constraint is also visible at the edge $\rho \approx 1.2$.

were used to train the network. The number of network input units is therefore 56: 20 profile locations for each of the two profiles, 15 bremsstrahlung spectral channels and one Z_{eff} value; the number of output units is one: the $\log\left(p(Z_{\text{eff}}, V_\lambda)\right)$ value. The network architecture used was a multi-layer perceptron (MLP) (see figure 4.1) with three hidden layers of 50, 20 and 10 units, respectively. The network was implemented within the TensorFlow framework [42], using the AdamW optimizer [61] with learning rate $\alpha = 10^{-5}$, weight decay $\lambda = 10^{-5}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. A mean square error loss function was used (see equation 4.3) During training, the network was regularly tested on a test set made of 10000 samples randomly drawn from the aforementioned distributions and early stopping was used to prevent overfitting and improve generalization. According to this condition, the training is stopped when the network error on test data starts becoming larger than that on training data. Another stopping condition for the training was given by the maximum number of iterations through the whole training set to 2000.

5.4.3. Evaluation of the network on experimental measurements

The trained network was then evaluated on data and measurements collected during the latest experimental campaign (OP1.2 b). Six different days, arbitrarily chosen, for a total of 172 plasma discharges, and a number of measurements larger than 10 000 were considered without restrictions on the experiment features (e.g., amount of heating power, magnetic configuration, etc). For a full list

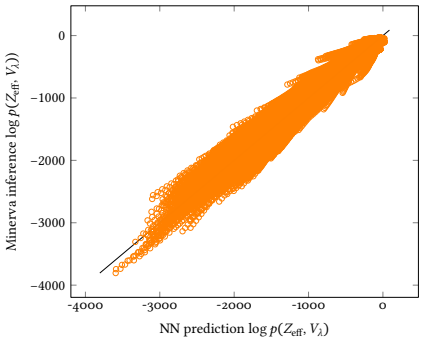


Figure 5.6.: Result of the neural network reconstruction of the model joint probability distribution values from more than 10 000 measurements collected during OP 1.2 W7-X experimental campaign. On the x-axis, the (logarithmic) probability values predicted by the network, on the y-axis, those predicted with the original Bayesian model, also used to generate the training data. The straight line shows the bisector.

of the experiment numbers please look at appendix A. Figure 5.6 shows one result from such evaluation. The network reconstructed values, on the x-axis, are compared to those inferred with the original Bayesian Minerva model used to generate the training data. Both network and Minerva model were evaluated on a grid of 100 linearly spaced values of $Z_{\text{eff}} \in [1, 6]$. Qualitatively speaking, the agreement between the two methods seem promising. A quantitative investigation of the error made by the network in the reconstruction is shown in figure 5.7. On the left, it is shown the mean relative error $E(Z_{\text{eff}})$ defined as:

$$E(Z_{\text{eff}}) = \frac{1}{N_m} \sum_i \left| \frac{\log_{\text{NN}}(p(Z_{\text{eff}}, V_{\lambda,i})) - \log_{\text{M}}(p(Z_{\text{eff}}, V_{\lambda,i}))}{\log_{\text{M}}(p(Z_{\text{eff}}, V_{\lambda,i}))} \right| \tag{5.8}$$

where N_m is the number of measured data points, i is an index labeling a single bremsstrahlung measurement $V_{\lambda,i}$, $\log_{\text{NN}}(p(Z_{\text{eff}}, V_{\lambda,i}))$ denotes the logarithmic probability value reconstructed by the network, and $\log_{\text{M}}(p(Z_{\text{eff}}, V_{\lambda,i}))$ denotes the value inferred with the Minerva model. The figure shows that the error is on average lower than 20% for most of the Z_{eff} values considered.

In the plot on the right, instead, the distribution across the measurement data set of the relative error $E(V_\lambda)$ averaged along Z_{eff} is shown. $E(V_\lambda)$ is defined as:

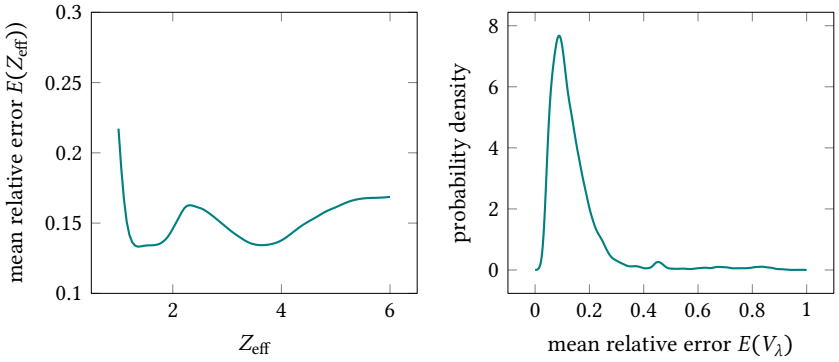


Figure 5.7.: Study of the discrepancy between the values of the joint probability reconstructed with the network and inferred with the original Bayesian model. On the left, the mean relative error between the probability values found with the two methods for each Z_{eff} value is averaged across all measurements considered. On the right, the distribution of the relative error averaged across the whole Z_{eff} range [1, 6] for each measurement considered is shown.

$$E(V_\lambda) = \frac{1}{N_Z} \sum_i \left| \frac{\log_{\text{NN}}(p(Z_{\text{eff},i}, V_\lambda)) - \log_{\text{M}}(p(Z_{\text{eff},i}, V_\lambda))}{\log_{\text{M}}(p(Z_{\text{eff},i}, V_\lambda))} \right| \tag{5.9}$$

where N_Z is the number of Z_{eff} values used in the evaluation, in this case fixed to 100 points in the range from 1 to 6, and i is an index labeling each of the values. The other quantities are defined analogously to equation 5.8. Approximately in 85% of the cases the network reconstruction differs by less than 20% from the original Bayesian model inference.

Figure 5.8 shows another comparison between network reconstruction and Minerva inference, this time for a single plasma discharge at W7-X (experiment 20180807.015). Hydrogen fueling was used for this discharge, a line integrated electron density of approximately 4 to 6 10^{19} m^{-2}, and an electron temperature between 2 and 4 keV were maintained throughout the pulse for approximately 8 seconds. The heating provided with the electron cyclotron resonance heating was of 4250 kW. In clockwise direction, starting from the plot on the top left corner, the figure shows the reconstruction of $\log(p(Z_{\text{eff}}, V_\lambda))$ for the bremsstrahlung emission observed at time 1.05 s, 3.40 s and 8.11 s as function $Z_{\text{eff}} \in [1, 6]$ for V_λ fixed at the measured value, for both the network and Minerva

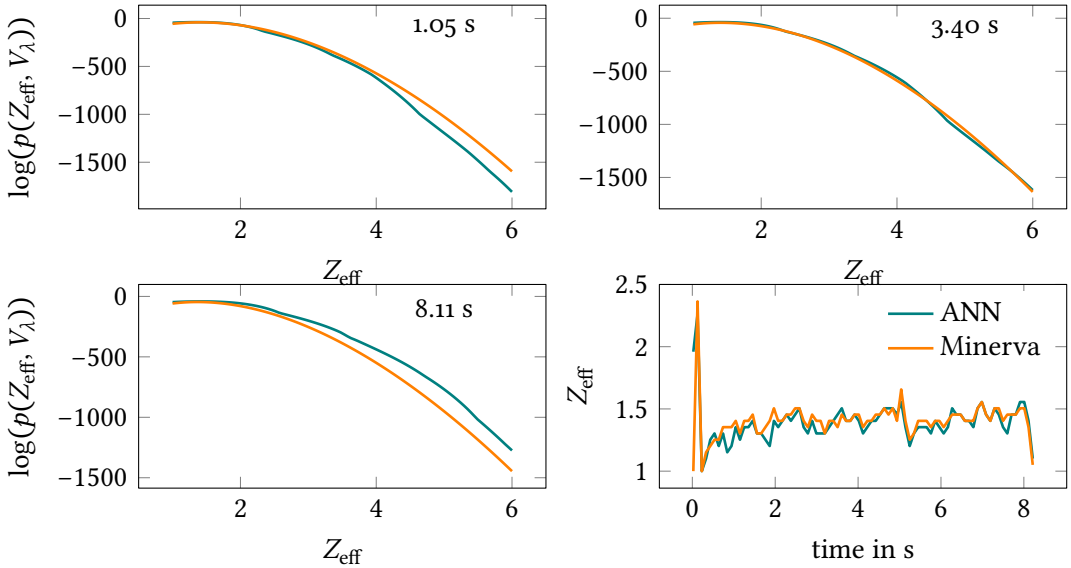


Figure 5.8.: Comparison of network and Minerva model reconstruction for W7-X experiment 20180807.015. In clockwise direction, starting from the top left, it is shown the value of $\log(p(Z_{\text{eff}}, V_\lambda))$ for $Z_{\text{eff}} \in [1, 6]$ and V_λ fixed at the measured value, as it is reconstructed by the network and with the Minerva model, at time 1.05 s, 3.40 s and 8.11 s, respectively. In the last plot, the maximum a posteriori value of Z_{eff} found by the network and with the Minerva model inference throughout the whole plasma discharge is shown.

model inference. This kind of reconstruction can be used to find the maximum a posteriori (MAP) value of Z_{eff}, i.e. the Z_{eff} value which maximizes the posterior distribution since $p(Z_{\text{eff}}|V_\lambda) \propto p(Z_{\text{eff}}, V_\lambda)$, throughout the entire discharge. This is shown in the plot on the right, where the Z_{eff} MAP values found with network and Minerva model are compared to each other. The analysis of the entire discharge with the Minerva model typically takes tens of minutes, whereas with the network it can be reduced to hundreds of microseconds.

5.4.4. A brief summary

With the example shown in this section, we have covered the last application that was mentioned in section 4.2, in the last formula of equation 4.10. Throughout the applications described in this chapter, the approximation framework

formulated in chapter 4 has been used to train neural networks in order to learn possible mappings between different set of variables and quantities defined by Bayesian models of nuclear fusion experiments. In the applications described in section 5.1 and 5.2, the network was trained to learn the mapping g from observable data $\mathbf{d}$ to free parameters $\mathbf{m}$, whereas in the application described in section 5.4, the network was trained to learn the mapping h between the joint space $(\mathbf{d}, \mathbf{m})$ and the (logarithmic) value of the joint probability distribution of the model $p(\mathbf{d}, \mathbf{m})$. The main advantage provided by the network is that of speeding up and automatize calculations which otherwise could be time consuming, especially when taking into consideration the large amount of data collected at these experiments. The speed-up is significant, in the range of several orders of magnitude. Another advantage of this specific implementation of the training method comes from the fact that we have exploited the common formulation of Bayesian models provided by the Minerva framework, which makes possible to standardize and automatize the sampling scheme, making it possible to apply it effortlessly to any model implemented within the framework. The application of this method is not, in principle, restricted to physics models of nuclear fusion experiments, and it is easily applicable to different systems, since the network can be successfully trained with data generated exclusively from existing Bayesian models. The theoretical framework provided by Bayesian formalism, and the common implementation language for scientific Bayesian models offered by the Minerva framework play an important role in making our method general, and, possibly, part of an autonomous system for scientific inference.

Chapter 6.

Publications

The publications listed in this chapter constitute the main outcome of my work during my Ph.D. studies. They mostly describe applications of the principles and methods that I, together with the other authors and group members, adopted and developed in order to improve scientific modeling and data analysis in nuclear fusion research. The publications feature technical details and describe solutions to issues that arise in practical applications. The reader that is not interested in such details, but rather in the concepts behind the methods and principles adopted, is recommended to look at the introductory chapters of this thesis.

6.1. Article I

A. Pavone et al.
»Neural network approximation of Bayesian models for the inference of ion and electron temperature profiles at W7-X«
In: *Plasma Physics and Controlled Fusion* **61** (2019)

Synopsis
The publication describes the application of neural network approximated Bayesian inference to the problem of the temperature profile reconstruction from X-ray imagining diagnostic measurements. In particular, we describe how the network can be trained to approximate the inference carried out with an existing Bayesian model by using it to generate the training data. The trained network can then be used to carry out fast inference on measurements, reducing the analysis time from tens of minutes to tens of microseconds. Also, we show how the quality of the training set can be assessed by looking at how well it describes the experimental measurements. We also demonstrate the performance of the network on a number of experimental cases, comparing its reconstruction with the profiles inferred with the Bayesian model.

OPEN ACCESS
IOP Publishing
Plasma Physics and Controlled Fusion
Plasma Phys. Control. Fusion **61** (2019) 075012 (14pp)
https://doi.org/10.1088/1361-6587/ab1d26

Neural network approximation of Bayesian models for the inference of ion and electron temperature profiles at W7-X

A Pavone[1], J Svensson[1], A Langenberg[1], U Höfel[1], S Kwak[1], N Pablant[2], R C Wolf[1] and the Wendelstein 7-X Team[1]

[1] Max-Planck-Institut für Plasmaphysik, Teilinstitut Greifswald, D-17491 Greifswald, DE, Germany
[2] Princeton Plasma Physics Laboratory, 08540 Princeton, NJ, United States of America

E-mail: andrea.pavone@ipp.mpg.de

Received 22 January 2019, revised 8 March 2019
Accepted for publication 26 April 2019
Published 30 May 2019

Abstract

In this paper, we describe a method for training a neural network (NN) to approximate the full model Bayesian inference of plasma profiles from x-ray imaging diagnostic measurements. The modeling is carried out within the Minerva Bayesian modeling framework where models are defined as a set of assumptions, prior beliefs on parameter values and physics knowledge. The goal is to use NNs for fast ion and electron temperature profile inversion from measured image data. The NN is trained solely on artificial data generated by sampling from the joint distribution of the free parameters and model predictions. The training is carried out in such a way that the mapping learned by the network constitutes an approximation of the full model Bayesian inference. The analysis is carried out on images constituted of 20×195 pixels corresponding to binned lines of sight and spectral channels, respectively. Through the full model inference, it is possible to infer electron and ion temperature profiles as well as impurity density profiles. When the network is used for the inference of the temperature profiles, the analysis time can be reduced down to a few tens of microseconds for a single time point, which is a drastic improvement if compared to the $\approx$4 h long Bayesian inference. The procedure developed for the generation of the training set does not rely on diagnostic-specific features, and therefore it is in principle applicable to any other model developed within the Minerva framework. The trained NN has been tested on data collected during the first operational campaign at W7-X, and compared to the full model Bayesian inference results.

Keywords: stellarator, x-ray imaging, neural network, Bayesian inference, modeling, Minerva framework, real time

(Some figures may appear in colour only in the online journal)

1. Introduction

Neural networks (NNs) are a powerful tool when it comes to speed and approximation of complex functions. universal approximation theorems have been shown to be valid for NNs under different assumptions, as in [1, 2] and [3]. The real time capabilities of NNs have also been shown in different fusion experiments, e.g. ion temperature profile inference and disruption prediction at JET as in [4–6] and at ASDEX Upgrade [7]. NNs have also been used for the reconstruction of plasma parameters from diagnostic data as in the case of charge exchange spectra automatic analysis at JET for reconstruction of ion temperature, rotation velocity and impurity density [8, 9], and in the case of electron temperature from a soft-x-ray system at NSTX [10]. In this paper, we focus on an

application of NN algorithms on x-ray imaging crystal spectrometer (XICS) measurements and we will describe an approach based on a different paradigm of NN training and reconstruction suitable when the physics model of the diagnostic is available. In particular, we will make use of the Bayesian implementation of the model within the Minerva modeling framework [11].

The Minerva framework provides a language to develop models of complex systems in a Bayesian way: models implemented within it share the same structure and are highly modular, so that different operations can be performed on them *independently* of the specific model under consideration. These operations can be the inference of the free parameter posterior distributions through different optimization and sampling algorithms, sampling schemes—as the one we make use of in this work, and storage/sharing. The elements/modules of a Minerva model are called *nodes*: these can be pieces of code that perform computation of physical quantities and they are such that they can be easily replaced. This makes building new models out of existing nodes particularly easy, as well as testing different models simply by switching their nodes. In this work, we have taken advantage of the modularity and shared structure of the models so that the method described here is general and can be easily applied to different problems modeled within the same framework.

NNs applied to diagnostic data are typically trained on real measurements and the corresponding quantities of interest in situations where a model of the problem is missing. Such an approach has the advantage of providing the NN with actually measured data, but it also has the limitation of depending on a fixed and restricted amount of training samples, the feature of which depends on the performed experiments, and on a limited parameter space. A different case is illustrated in [9]: here the authors have first reconstructed the distribution of a large set of measured physics parameters; then new parameters have been sampled from it, and used as input in a forward model in order to create simulated measurements. A NN was then trained on both measured and simulated data.

The way we try to overcome these limitations is by training the network solely on data synthesized through the Bayesian model specified within Minerva, the same that is used for the standard inference [12]. The parameters, corresponding to physics quantities, used to produce the data are sampled from the corresponding prior distributions and the synthesized observations are sampled taking into account the error model. This conferes control over the features we put in the training set and, consequently, the features the NN will be learning and will be sensitive to when evaluated on measured data. In addition, an advantage of this approach concerns its generality and the possibility of performing automatic data analysis based on physics models, which comes as a consequence of the sampling procedure described in the following chapters. This becomes of greater relevance as the scale of fusion experiments grows larger and the duration of plasma shots becomes longer. During the first operation campaign (OP 1.1, see [13]) of the W7-X stellarator several diagnostics [14, 15] were involved in the measurements, and the number increased during the second one (OP 1.2a). Together with the number of diagnostics, a large proportion of which is currently implemented in the Minerva framework, e.g. [12, 16, 17], also the duration of the plasma shots increased. All of this makes fast and automatic data analysis very desirable. The technical implementation of our approach only makes use of features shared between all models in Minerva, and thus it is easily transferable and applicable to other diagnostics modeled in the framework.

The paper is organized as follows: in section 2 we give a general outline of the core idea behind the training method, that is the sampling scheme from the Bayesian model; this constitutes the main piece of novelty of this work. The specific details necessary to understand each part of the entire scheme are then delivered in the following sections. In section 3, we give a brief overview of the hardware setup and the physics involved in the measurements performed with the x-rays imaging diagnostic. It is followed by an explanation of Bayesian modeling in section 5, which starts generally from Bayes theorem and goes into more specific details of the Minerva implementation of the diagnostic model. In section 6 we illustrate the sampling scheme used to generate the training data from the XICS Bayesian model, whereas details about the NN architecture and training algorithm are given in section 7. In section 8, we will show how training sets generated with different models can be compared to each other in order to be able to choose the one that better describes the measurements; this is particularly relevant in the context of this work since the training data are generated synthetically. In section 9 we compare the NN reconstructed profiles to those traditionally inferred with the Minerva Bayesian model in the case of measurements performed during the first experimental campaign at the W7-X experiment. In section 10 we draw some conclusive considerations.

2. The method

The core of the method that we want to illustrate in this section is the sampling scheme used to generate the NN training data from the Minerva Bayesian model of the XICS diagnostic measurements. In order to understand how it is practically carried out, it is necessary to have some basic notions about the XICS diagnostic and the Minerva modeling framework. Details about the former are given in section 3, while the latter is described in section 5.

An outline of the idea behind the method is the following: the XICS Bayesian model is a model whose key elements are the *free parameters*—the ion and electron temperature profiles—and the *observations*, the XICS measured images. Before any measurement is taken into account, a *prior distribution* is assigned to the free parameters, and a *likelihood function* is assigned to the observations. The former is usually a broad distribution which may include features we believe are found in measured plasma profiles, whereas the latter reflects the uncertainties of the model in predicting the data, which typically include measurement errors. The product of the two probability distributions defines the *joint probability*

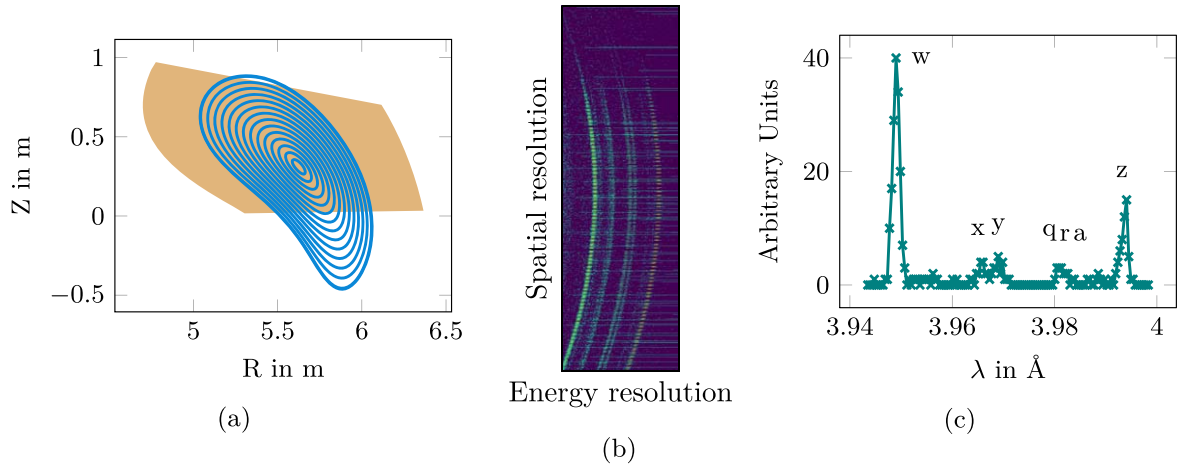


Figure 1. (a) Sketch of the XICS system view on the bean shaped cross section at W7-X. The viewing span of the lines of sight is represented by the ocher area. (b) A measured raw image detected on the CCD detector. The typical curved feature is due to the spherical bending of the crystal. (c) He-like Argon spectrum measured along one of the central lines of sight of the image in figure (b). The main emission lines are marked with their names.

Table 1. Atomic process description and corresponding plasma parameter dependency.

Atomic process	Plasma parameter dependency
(1) Excitation from ground state of He-like ions	n_e, n_{ArHe}, T_e, T_i
(2) Di-electronic recombination of He-like ions	n_e, n_{ArHe}, T_e, T_i
(3) Recombination of H-like ions	n_e, n_{ArH}, T_e, T_i
(4) Inner shell excitation of Li-like ions	n_e, n_{ArLi}, T_e, T_i

distribution of the model. The joint distribution completely describes the model, both from the point of view of its statistical properties—residing in the fact that it is a probabilistic distribution, and from the point of view of the physics implemented within it—found in the relation between free parameters and observations. Since we aim at training the NN to be an approximate *replica* of the full Bayesian model, we can create the training data set by sampling the free parameters and synthetic observations from the joint distribution. Afterwards, we train the NN on the inverse problem, which is the mapping of the observed XICS images to the ion and electron temperature profiles. We expect that, under the assumption that the training data properly resemble the measurements, the network can be used for fast reconstruction of temperature profiles from new experimental data. The goodness of this assumption for a given training data set can be tested quantitatively, so that it is possible to compare different models in their ability to produce data set that resemble the experimental measurements. In section 8 we describe a possible way to test this assumption.

3. XICS diagnostic

The XICS collects x-rays emitted in atomic processes involving ion impurities and plasma electrons occurring in the bean shaped cross section of the W7-X stellarator. The concept behind the diagnostic is described in [18]. A sketch of the view is shown in figure 1(a), where the plasma cross section and the line of sight (LOS) span are shown. For each LOS an integrated spectrum is collected. The set of all spectra forms one measured image. From an image it is possible to reconstruct plasma ion and electron temperature profiles, and impurity density profiles. In section 3.1 we will describe the setup of the diagnostic and in section 3.2 we will describe the physics involved in the measurements.

3.1. Setup

The system is equipped with a spherical bent crystal and the light is collected onto a CCD detector producing 2D images similar to the one shown in figure 1(b). The two dimensions in the image represent energy and spatial resolution respectively. The diagnostic is sensitive to the energy region of He-like Argon emission lines. The main emission lines constituting the spectrum are shown in figure 1(c) and they are the w, x, y and z for the $n = 2$ to $n = 1$ transitions in addition to numerous $n >= 2$ dielectronic satellites, e.g. the k lines for $n = 2$. A study of the spectrum and the atomic processes involved can be found in [19–21]. The detector covers the wavelength range from ≈3.94 to ≈4.00 Å along the central LOS.

3.2. Physics

The core component of the physics processes involved in the measurements is given by the atomic processes giving rise to the spectral emission. A detailed description of the calculation of the emission intensity for the different lines can be found in

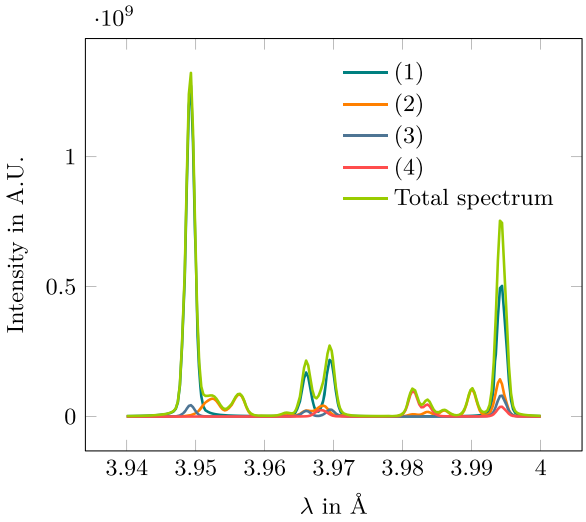


Figure 2. A He-like Argon spectrum calculated with the forward model for $T_i = 1$ keV, $T_e = 1.2$ keV, $n_e = 10^{13}$ cm^{-3}, $n_{ArHe} = 10^8$ cm^{-3}, $n_{ArLi} = n_{ArH} = n_{ArHe}/8$. The numbers in the legend refer to the atomic processes listed in table 1.

[19]. In table 1 we show the processes involved in the calculations relevant to this paper together with their dependence on plasma parameters.

The intensity of all lines depends on the electron temperature T_e through the corresponding effective rate coefficients, a calculation of which can be found in the appendix of [19]. The dependency on the ion temperature T_i comes as well into the calculation of all of the line shapes as Voigt profiles $V(\lambda_l, \lambda)$, which is the convolution of a Gaussian and Lorentzian shape, accounting for Doppler broadening and natural broadening, respectively. The photon emission of each process also depends on the electron density n_e, and on the density of Argon ions in one of the ionization stages: n_{ArHe} for Ar^{16+}, n_{ArLi} for Ar^{15+}, and n_{ArH} for Ar^{17+}. All of the quantities in table 1 are defined on a 3D Cartesian coordinate space, so that they are dependent on the position $\mathbf{x}$. The emission intensity $I(\lambda)$ at a given λ is then calculated performing an integration along the LOS paths L, as in the following equation:

$$I(\lambda) = \int_L n_e^2(\mathbf{x}) \sum_l V(\lambda_l, \lambda, \mathbf{x}) i_l(\mathbf{x}). \tag{3.1}$$

The quantity $i_l(\mathbf{x})$ is defined according to:

$$i_l(\mathbf{x}) = \sum_{j(l)} n_j(\mathbf{x}) k_{lj}(T_e(\mathbf{x}))\mathrm{d}\mathbf{x}. \tag{3.2}$$

It denotes the overall contribution from different ionization stages j to the emission line l. In the equations, λ_l denotes the wavelength for the given line, k_{lj} denotes the effective rate coefficient of the line l in the ionization stage j, and n_j denotes the density of ions in the ionization stage j.

The contributions to the overall He-like Argon spectrum arising from the different atomic processes and ionization stages are shown in figure 2. In the spectrum depicted, the lines q, r, and a, are also visible.

Since an absolute calibration of the diagnostic measurements is currently not available, the simulated images cannot reproduce the measurements in their absolute values. Therefore, both simulated and measured images are normalized dividing the intensity of each pixel by that of the brightest one.

4. Bayesian inference

When developing a Bayesian modeling and inference scheme, the first step is the definition of the model free parameters, w, and observed data, d. Probability distributions are assigned to both quantities, and they take the name of *prior* distribution, $P(w)$, and *likelihood* function, $P(d|w)$. The prior distribution represents the *a priori* knowledge that we have about the free parameters before taking the observations into account. The likelihood distribution represents instead the model uncertainties in the prediction of the data. The inference is the process of knowledge acquisition when new data are observed. According to Bayesian probability theory, it can be described as an update process of the *a priori* distribution. This process is formally expressed through Bayes formula:

$$P(w|d) = \frac{P(d|w)P(w)}{P(d)}. \tag{4.1}$$

The numerator in equation (4.1) corresponds to the joint distribution $P(d, w)$ of the observed data and free parameters, $P(d, w) = P(d|w)P(w)$. The term $P(w|d)$ is called *posterior* distribution of the free parameter w given the observed data d, and it represents the new state of knowledge on the model free parameters as new data are collected. The quantity $P(d)$ is called *evidence* or *prior predictive* and, as first interpretation, it plays the role of a normalization factor:

$$P(d) = \int P(d|w)P(w)\mathrm{d}w. \tag{4.2}$$

The hidden relevance of this term becomes evident when we switch from Bayesian inference for parameter estimation to Bayesian model selection, see for example [22]. Since the integral in equation (4.2) is a marginalization over the free parameters of a given model, we see that $P(d)$ is the distribution of all the data that a model can describe, quantifying how likely each data point is to be generated under the assumptions of the model. Once it is evaluated on a data point d^*, the evidence $P(d^*)$ defines how likely our model is as an explanation of such data point. The value of this probability depends on the prior distributions of the parameters and the uncertainties attributed to the model prediction in the likelihood term. The dependency is such that broader prior distributions, or prior distributions defined over higher dimensional parameter spaces, i.e. more complex models, will be penalized, and the overall probability will always constitute a trade-off between model complexity and good fit to the data. This is indeed an application of the *Occam*'s *razor* principle. An illuminating explanation of how Bayesian model selection naturally embodies Occam's razor is given in [23].

As data are measured, Bayesian inference can be carried out to extract information about the free parameters. The

target of Bayesian inference is the posterior distribution of the parameters, $P(w|d)$, the spread of which expresses the uncertainties on the inferred quantities.

5. The Minerva framework

In order to interpret the measured data, the physics describing the processes giving rise to the measurements is implemented in a forward model of the diagnostic within the Minerva framework [12, 24]. The Minerva modeling framework is a framework for the development of Bayesian models of complex systems. Within the framework it is possible to carry out both modeling and inference. The result of the modeling is an object that is called a *graph*. It contains all information about the physics and the probabilistic relations between the modeled quantities. It describes all the hypotheses used in the creation of the Bayesian model. A given model can be used to produce simulated observations and compare them to the measured observations. In this way, the model free parameters that better describe the data can be found. The advantage of doing Bayesian inference is that it formally defines a procedure to calculate the uncertainties of a solution, which simply involves the application of Bayesian probability rules.

The Minerva framework relies on graphical models [25] in order to express the conditional dependency between random variables in the model. For each model implemented in the framework, a *graph* object is created that describes the joint distribution of the free parameter and the measurements according to the forward model. A simplified version of the XICS model graph is shown in figure 3. In the graph the colored nodes are probabilistic nodes, where *orange* denotes the free parameters and *blue* denotes the observed quantities.

In the case of the XICS forward model, the free parameters can be the plasma parameters summarized in table 1, right column, and the observed data are the images constituted of spectra like the one shown in figure 2, calculated accounting for the atomic processes described in section 2. All the distributions in the graph are chosen to be normal distributions. Note that the likelihood function is then a normal distribution centered on the model prediction obtained with a given set of free parameter values:

$$P(d|w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(d - y(w))}{2\sigma^2}\right), \tag{5.1}$$

where we used $y(w)$ to denote the forward model function y dependent on the free parameter values w, and d to denote a measured data point. The white squared nodes in the graph of figure 3 represent deterministic calculation nodes, and the cloud node is used to denote a *data source*, i.e. a node that communicates with a database, here the W7-X ArchiveDB, where information about the diagnostic, e.g. geometry setup etc. are stored together with measured and analyzed data. The arrows represent dependencies between nodes rather than a computational flow. For example, all arrows from the free parameters reach, directly or indirectly, the observed node, i.e. the probability distributions of the observed quantities (d) should be conditioned on the value of the free parameters, $P(d|w)$. The joint probability distribution represented by the whole graph can be factorized and written as: $P(w, d) = P(w)P(d|w)$.

The T_e, T_i and n_k profiles, where n_k stands for Argon ion or electron density, are functions of the normalized effective radius $\rho_{\text{eff}} = \sqrt{\psi/\psi_{\text{LCFS}}}$. Thus, an equilibrium code, in this case VMEC [26], called from the corresponding node, is required to carry out the mapping to the 3D Cartesian coordinates. The impurity emission can then be calculated locally and integrated along the lines of sight. A background emission is also added to the spectrum. In order to calculate the detector pixel response, the instrumental function and the wavelength dispersion on the chip are also required. The data source node is used to provide the observed data for the inference and the l.o.s. geometry. The smoothness of the profiles is controlled by a zero mean Gaussian process (GP) prior [27] with a squared exponential covariance function, as defined in:

$$\text{cov}_{ij} = \sigma_f^2 \exp\left(-\frac{(\rho_i - \rho_j)^2}{2\sigma_x^2}\right) + \delta_{ij}\sigma_y^2. \tag{5.2}$$

The quantity cov_{ij} denotes the elements of the covariance matrix, ρ_i and ρ_j denote the location of any two profile points labeled with i and j, and the quantities σ_f, σ_x and σ_y denote the function variance, the length scale and the noise variance of the profiles, respectively. Note that the quantity cov_{ij} corresponds to the covariance between any two points in the profile, as function of the location ρ_{eff}. The length scale σ_x describes how smooth a function is. Small length scale values mean that function values can change quickly in their domain, whereas large values describe functions that change slowly. In our case, this refers to plasma profiles characterized by either quick or slow spatial changes. The function variance σ_f is a scaling factor and determines function value variations around the mean. Large values will allow again for bigger variations, smaller values will describe less varying functions. It also determines, together with σ_y, the value of the covariance matrix elements along the diagonal, where $i = j$. The noise variance σ_y is used to allow for noise present in the data and it specifies the amount of expected noise.

When measured data are available, inference can be performed on a Minerva model. In a practical implementation, when a single value solution is desirable, a Maximum A Posteriori optimizer can be used to find the maximum of the posterior distribution. The full Bayesian answer to the inference problem is nevertheless provided by the full posterior distribution, the spread of which expresses the uncertainties on the inferred quantities. In order to provide this information, a Markov Chain Monte Carlo (MCMC) sampler is usually adopted to generate the posterior samples. The samples can then be stored and used in further independent calculations providing full non-linear uncertainty propagation. Profiles inferred using XICS data can then be used, for example, in impurity transport studies, see [24, 28].

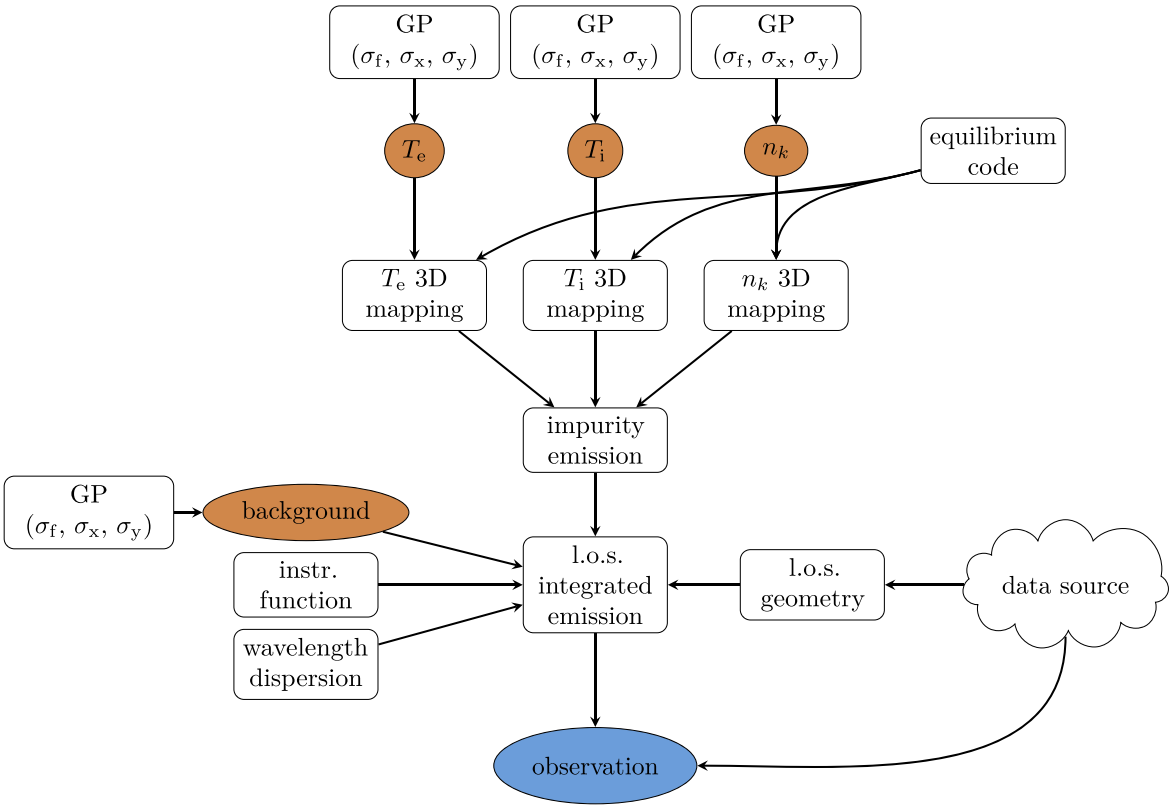


Figure 3. A simplified sketch of the XICS model graph. Colored nodes are probabilistic nodes, where *orange* denotes the free parameters and *blue* denotes the observed quantities. White nodes represent deterministic calculation nodes. The white GP nodes represent a Gaussian process prior, and the symbols σ_f, σ_x and σ_y denote the parameters in the expression of the squared exponential covariance function. The data source node is used to fetch diagnostic specific information and measurements from the W7-X Archive. The arrows represent direct or indirect dependencies in the probabilistic relations between the quantities in the probabilistic nodes.

The model depicted in figure 3 represents a sophisticated inference problem: first of all, the images fed to the inversion routines go through a preprocessing stage, occurring in the data source node, where they are (1) straightened, and (2) binned along the LOS direction in order to reduce the computational effort. At this point, the inversion problem consists of simultaneously fitting an image of 20×195 pixels along the LOS direction and the wavelength dispersion direction respectively, and doing a tomographic inversion of different plasma profiles. The full Bayesian inference takes from 1 to 4 h for each measured image, whereas a NN can process data at a time scale of tens of microseconds in good implementation conditions (e.g. on a GPU).

6. Creation of the training set

In order to generate the NN training set, we will only make use of the XICS Minerva model. All the features of the model are expressed in its joint distribution $P(d, w) = P(d|w)P(w)$: the distribution of the variables (d, w) depends on the functional form of the forward model, appearing in the likelihood term $P(d|w)$ as $y(w)$ (equation (5.1)) and which expresses the dependence of d on w, the uncertainties on the model prediction and the prior distribution $P(w)$. As our goal is to create a (approximated) copy of the original full Bayesian model, we must provide the NN with training data having the same properties described by the model: this is achieved by generating the training set data from samples of the full model joint distribution.

Practically, such training set can be obtained by iterating over the following three steps:

i. Draw and store a sample from the joint prior distribution of the free parameters: $P(w) = P(T_e, T_i, n_k, bg) = P(T_e)P(T_i)P(n_k)P(bg)$, where bg denotes the background l.o.s. integrated emission profile
ii. Run the forward model in order to calculate a synthetic observation with the given free parameters
iii. Store a number of samples drawn from the likelihood function of the synthetic observations, $P(d|w)$, which is fully specified by the given set of sampled free parameters and the model uncertainties.

The sampling procedure taking place at step (iii) will introduce noisy samples in the training set since the likelihood distribution expresses the uncertainties of the model prediction. This will help making the NN stable against small perturbations in the input data when evaluated on measured images. This is equivalent to the technique known as *data augmentation*, [29, 30]. The modifications we inject into the samples are based on the noise model that has been assumed for the problem, in this case a Gaussian noise model. The NN

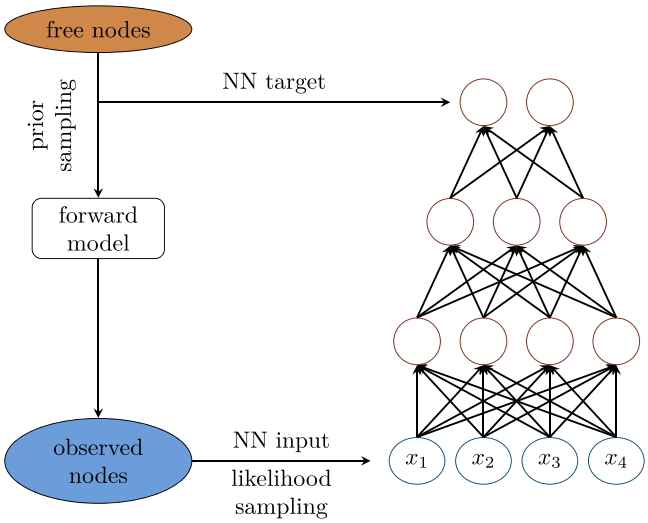


Figure 4. A sketch to illustrate the sampling procedure for the training set creation described in section 7. A sketch of the XICS Minerva model and the neural network is shown on the left and on the right, respectively. The NN takes as input images sampled from the likelihood function of the model, given a set of sampled free parameters. The blue nodes of the neural network denote the input pixel of the image and the two red nodes at the top denote the output points of the ion temperature profile.

is then trained on the mapping from the images to the ion and electron temperature profiles. Training on other profiles is also possible and straightforward, since it is just matter of choosing and storing another set of samples among those stored at step (i). As a consequence of such sampling procedure, the portion of the training set corresponding to the model prediction d is made of samples obtained by marginalizing the numerator of equation (4.1) over the model free parameters. In other words, these are samples from the evidence term $P(d)$, the denominator of equation (4.1).

A sketch of the procedure is illustrated in figure 4. The size of the input images is 20 pixels along the LOS dimension and 195 pixels along the wavelength dimension. The target ion and electron temperature profiles are defined with 15 points equally spaced along the effective radius. The training set is made of 500 000 samples. A test set made of 10 000 samples is used to check the generalization capabilities of the NN during training and it is generated in the same way as the training set.

Given the previously mentioned sampling procedure, an insightful interpretation can be given to the NN mapping. A well known result [31] in the NN field states that, under the assumption of a sum-of-squares error loss function as in equation (7.2), large training data set and successful optimization, the NN mapping f is given by the conditional average of the target data y_i, conditioned on the input vector $\mathbf{x}_i$:

$$f(\mathbf{x}_i; \mathbf{w}) = \langle y_i | \mathbf{x}_i \rangle. \tag{6.1}$$

In the specific contest of our study, the network's targets and input are the ion and electron temperature profiles and the synthetic XICS images. Given the fact that the training set is generated sampling from the joint distribution of the XICS Bayesian model, the distribution of the target data t given an input vector $\mathbf{x}$, $p(t|\mathbf{x})$, in the limit of large training data set, corresponds to the posterior distribution of the ion or electron temperature profile of the full Bayesian model. Therefore, we can state that the NN mapping, in ideal circumstances, is given by the mean of the full model posterior distribution. In real world circumstances, the data set size is finite and the optimization is never perfect, so that we can say that the inversion provided by a NN trained in such a way constitutes an *approximation* of the full model Bayesian inference.

7. NN input, output and architecture

It is worth to summarize here what the NN input and output are at the different stages. At *training time*:

- *Input*: synthetic images, generated with the XICS forward model and the sampling procedure described in section 6. These images supposedly closely resemble the actual XICS measurements (after few pre-processing operation, i.e. row binning and straightening, see next bullet point and figure 5). They are made of 20 $\times$ 195 pixels/values.
- *Target*: the ion or electron temperature profiles used to generate the corresponding images. These are made of 15 points along the effective radius $\rho_{\text{eff}} = \sqrt{\psi/\psi_{\text{LCFS}}}$, where ψ is the magnetic flux and ψ_{LCFS} is the flux at the LCFS. The points are obtained by sampling from the GP prior distributions of the temperature profiles.

At *evaluation time*:

- *Input*: the pre-processed, actual XICS measurements. The measured images, originally showing the curved feature shown in figure 1(b), are straightened according to the detector and crystal geometry. Afterwards, the original 1475 $\times$ 195 pixels of the image are binned along the largest dimension, which corresponds to the spatial resolution, where neighboring lines of sight overlap significantly. The binned images are made of 20 $\times$ 195 pixels. An example of a binned image is shown on the leftmost side of figure 5.
- *Output*: the estimated ion or electron temperature profiles. In case of successful training, it will match, within uncertainties, with the profile inferred through the full Bayesian model.

Essentially, two NNs, identical for all the features except for the target profiles, have been trained and tested independently: one for the inversion of ion temperature profiles and the other for the inversion of the electron temperature profiles.

Since the network's input are 2D images, the network architecture has been inspired by the LeNet-5 convolutional neural network (CNN) [32] and is shown in figure 5. This kind of architecture has been shown to be particularly effective when the input present a 2D structure. It has been successfully used in many image recognition problems, achieving state-of-the-art results [33, 34]. Here we expect to recurrently find across the image the features induced in the spectrum by the ion or electron temperature profiles, which

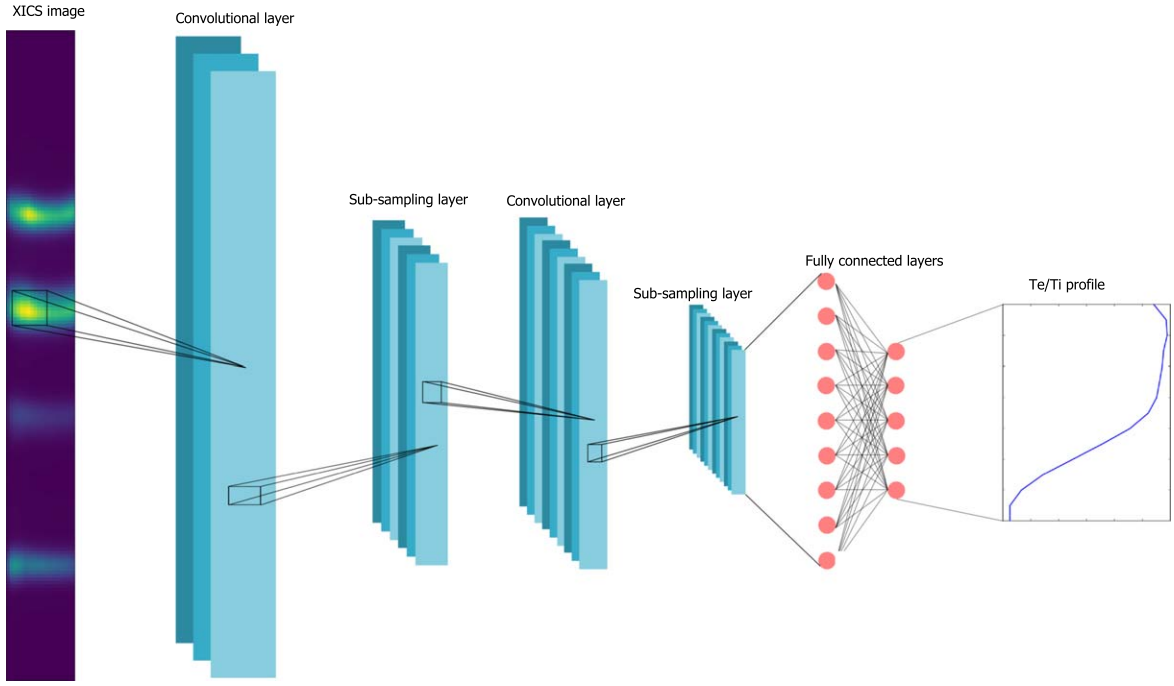


Figure 5. Architecture of the NN used. The input layer at the leftmost side is followed by a convolutional layer and a sub-sampling layer, which are followed again by a couple of convolutional and sub-sampling layers. Two fully connected feed forward layers follow up to the output layer. Each blue plane represents a feature map, where all the units share the same weights.

affect line width and intensities, respectively. Two convolutional layers C1 and C2, each one followed by one sub-sampling layer, are used in a hierarchical feature extraction structure. A convolutional layer applies a convolution filter or kernel to the input image, extracting information that are recurrent accross different locations in the image (for more details see [32]). The kernel dimension sizes used in the convolutional layers are respectively: (3, 16) and (2, 5), where the first and second dimensions refer to the LOS and wavelength dispersion dimension of the input images. The number of *feature maps*, i.e. sets of units whose weights are constrained to be identical [35, 36], is set to 30 for both convolutional layers. The sub-sampling layers use *max pooling* with a resolution of 2 by 2. Two fully connected layers M1 and M2 made of 20 and 18 units respectively, constitute the final layers, which will produce the desired 15 points ion or electron temperature profile output. The activation function in the convolutional layers is the rectified linear unit:

$$f(x) = \max(0, x), \tag{7.1}$$

whereas in the fully connected layers it is the hyperbolic tangent function.

CNNs are also especially suitable for parallelization on GPUs, which applied to our case made the training 30 times faster than on CPUs. The NN was implemented within Theano, a Python framework for fast symbolic computation [37].

The training was stopped either according to an early stopping criterion, i.e. when the network performance on the test starts degrading, or when the decay rate of the loss function is small enough. The loss function that the NN is trained to minimize is defined as:

$$S(\mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}(y^n - t^n)^2 + \sum_{k,i=1}^{L,N_k}\alpha_k w_{k,i}^2, \tag{7.2}$$

where $\mathbf{w}$ denotes the network weight vector, the first term on the right-hand side is the sum-of-square error between the network output y_n and the target t_n and n is an index that goes through the N samples in the training set. The second term on the right-hand side is a regularizing term, where $w_{k,i}$ denotes the network weight of unit i at layer k. The parameters β and α_k are scale parameters which control the relative importance of the two terms An insightful interpretation, based on a Bayesian view of the NN training [23, 38], can be provided to such expression and can be useful in the choice of the values of β and α_k. The values of $\beta = 10$ and $\alpha_k = (\alpha_{C1} = 68.00, \alpha_{C2} = 58.33, \alpha_{M1} = 576.67, \alpha_{M2} = 5.83)$ were used. More details about Bayesian NN training in the context of this study can be found in [39].

8. Training set comparison

When dealing with NN reconstructions, the following situation is likely to occur: the network might be able to reconstruct the target profiles from training data, but it might fail when applied to experimental data. One reason why this might happen is the following: the features of the training data do not resemble closely enough those found in the measurements. In this section, we illustrate how training data sets generated with different models can be compared to each other from the point of view of their adequacy in describing the experimental measurements. In particular, we will consider two cases: the case of a data set

generated by sampling unconstrained profiles that leads to a poorly performing NN, and the case of a data set where a realistic constraint is applied to the prior distributions, which in turn help improving the performance of the network. In general, one would try to introduce as little bias as possible to a given training set. As it is shown here, this does not always lead to a successful result. The reason of the failure can therefore be assessed with the method described here. It can also be used as a novelty detection system: it can be used to identify a novel incoming measured image as an outlier, i.e. a case that is relatively unknown to the network and on which it might perform poorly. In this way, one can be informed in advance about how reliable the NN output is going to be. In general, since several different prior distributions can be used to generate training data, this method provides a way to quantitatively compare them and choose the one that better describes the measurements before any training is performed. In section 8.1 we will describe the algorithm used for the comparison and in section 8.2 we will show its application to training sets generated with two different prior distributions.

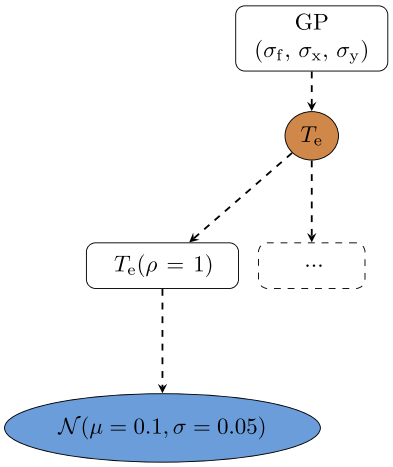


Figure 6. A sketch of the virtual observation constraint applied to the T_e profile. The blue ellipse represents the so called *virtual observation*, which states that the T_e profile has been 'observed' to have value 0.1 keV with standard deviation 0.05 keV, at the LCFS ($\rho = 1$).

8.1. The k-nearest neighbors algorithm

The features of the set of measurable images D_m are determined by the properties that the plasma profiles have during the experiments. An absolutely free, unconstrained sampling of the 15×7 points in the plasma profiles introduced in table 1, will produce a set of synthetic data D which likely will have little in common with D_m. Most of the samples in such a training set will have little use to our purposes, since they would not belong to the domain of the mapping that we want the NN to learn. In order the NN to be able to accurately predict temperature profiles from measured images, the D_m space has then to be covered densely enough by the training data set: we are not interested in generating all possible 15-dimensional output vectors, but only those which represent realistic ion or electron temperature plasma profiles. This is accomplished by refining the prior distributions in such a way that sampling from them will generate a data set of synthetic images which densely encloses D_m.

In principle, the full distribution of the data described by a given model is determined by the prior predictive distribution, equation (4.2). As we have seen in section 6, the training images constitute samples from such distribution, therefore they can be used to estimate how closely a training set resemble the actual measurements.

Several methods can be used to study the adequacy of the training set to the coverage of the D_m space of the measured images. Here we will describe an approach which relies on the *k-nearest neighbors* algorithm (*k*-NN). A *k*-NN algorithm is used to find a number k of data points in the training set that are the closest to a given observed data point, according to a metric measure. In this case the Euclidean distance has been used. We expect that if we compare the distance from the training samples to a measured image and to the test set samples, the former will be much larger of the latter in the case where the measurement is not properly described by the training data. This expectation is justified by the fact that we know that the network performs well on the test data and therefore they can be used as reference. Distance based methods are often used in the framework of outlier and novelty detection and similar methods are presented for example in [40, 41].

8.2. Refining the priors

An application to our study is shown in figure 7, where we have compared two training sets obtained with two different models. The difference in the models is in the prior distribution of the plasma profiles: in one case, labeled as W/O, the temperature profiles were left unconstrained in the region of the last closed flux surface (LCFS), being allowed to assume any value between 0 and 10 keV; in the other case, labeled as W., the profiles were constrained to assume low values in the LCFS region (0.1 keV + −0.5 keV) at $\rho_{\text{eff}} = 0.99$), a feature that is typically expected in such plasma profiles. Such a constraint enters the Minerva model as a so called *virtual observation*: at the level of the Minerva graph, this corresponds to a standard observed node connected to the profiles which states that the value of the profiles at the given position x_p has been 'virtually measured' to have value v_p with error ϵ_p, as shown in figure 6. The only difference with the other observed nodes in the graph is that it does not correspond to an actual measurement. Its role is to constrain the profile shape, and this is what observations in Bayesian models do: they constrain the solution found for the free parameters. In figure 6, the dashed arrow and box represent the connection to the rest of the model of figure 3. The computation node on the left of the dashed one represents the evaluation of the T_e profile at the $\rho_{\text{eff}} = 1.0$ position, which corresponds to the LCFS. This node is finally connected to the virtual observation, the blue circle, whose normal distribution is specified in term of its mean and standard deviation. If we exclude the graph represented in this figure

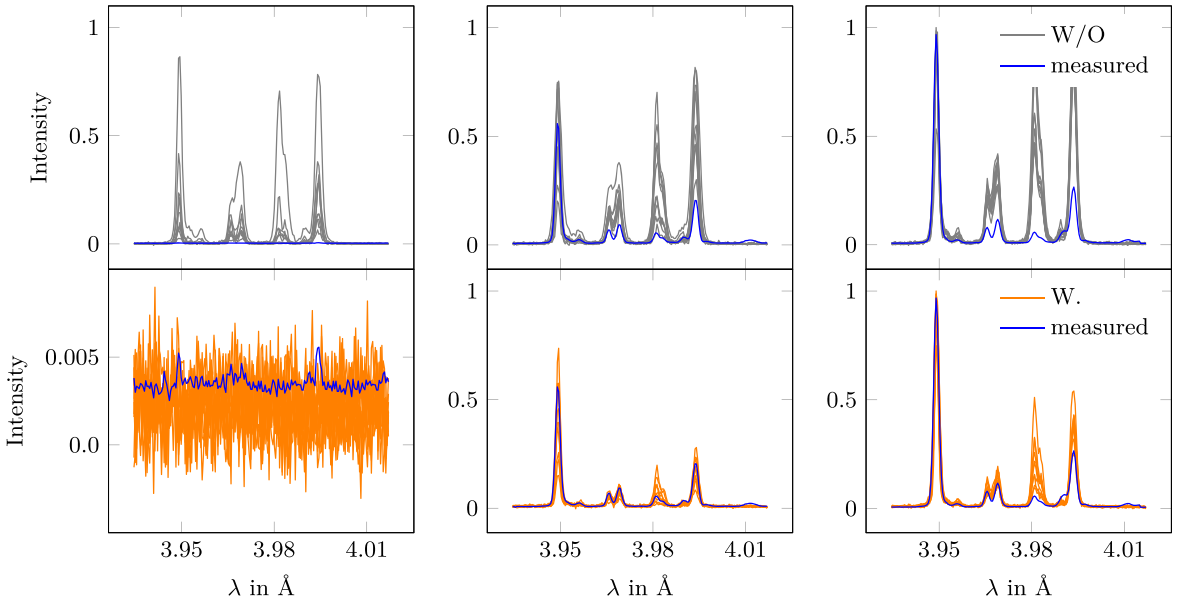


Figure 7. The *k*-NN algorithm is applied to find the 10 nearest neighbors of a measured image (blue line) among the training samples in two different training sets: (1) the T_e profiles are let vary without (W/O) constraint on the values they assume at the edge of the machine (three top plots), (2) the T_e profiles are sampled with (W.) constraint to low values at the edge, as described in the text (three bottom plots). The constraint is applied to the prior probability distributions. Each column shows spectra integrated along a different line of sight. From left to right: a line-of-sight through the edge plasma, one crossing the mid-plane half-way to the core, and one core line of sight.

from the rest of the model, and we then calculate the covariance matrix of the posterior distribution of the plasma profile, in this case T_e, given the virtual observation, we will find a covariance matrix which can be used later on to sample profiles which will feature the desired constraint. This has been done in order to obtain more realistic plasma profiles, and the effect of this on the training set is described in the next paragraph. Specifically, the virtual observation was implemented as a normal distribution with mean value of 0.1 keV and standard deviation of 0.05 keV at $\rho_{\text{eff}} = 0.99$ for both ion and electron temperature profiles.

The importance of sampling a realistic set of plasma profiles, which corresponds to a realistic set of synthetic measurements, is depicted in figure 7. The three plots on the top show a spectrum measured along three different lines of sight (blue line), and the 10 nearest neighbors (10-NN, gray lines) found among the samples in a training set where the electron temperature profiles were sampled without constraint (W/O) on the value assumed on any of the flux surfaces. The only constraint was a smoothness criterion induced by the GP prior. From left to right, the lines of sight of the plots are traversing the following regions of the machine: edge, half-way to the core and core. The intensity on the *y*-axis is normalized to the brightest pixel in the image, in this case a w spectral line along one of the central lines of sight. The three plots on the bottom, instead, show the same measured image and the 10-NN neighbors found among the samples in the training set with constraint (W.). The effect of such constraint on the sampled profiles is shown in figure 8, where 100 samples from each of the two training sets are drawn. It is evident that when the constraint is applied (bottom plot), the average electron temperature through the machine is lower. This brings the training samples closer to the measured data in

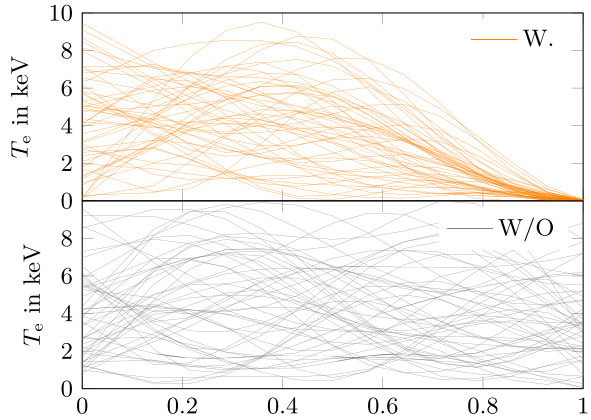


Figure 8. The T_e profile samples from the two training sets. Top, the profiles are sampled with (W.) constraints. Bottom, the profiles are sampled without constraints (W/O).

two ways: (1) the intensities measured along the edge lines of sight (see leftmost bottom plot in figure 7) show a smaller signal to noise ratio, (2) the ratio between different emission lines is smaller, see middle and rightmost plots in figure 7. The distance of a measured data point from the 100 000 nearest neighbors in the training set can be compared to the distance of 10 test set samples from 100 000 training set samples to get an estimation of the proximity of the measured data with respect to the training samples. These values are shown in table 2 in the two cases of training set created with and without constraint. The distance for the measured data point is substantially reduced when the constraint is applied to the electron temperature profile prior distribution, getting closer to the value of the test set samples. It is worth to note

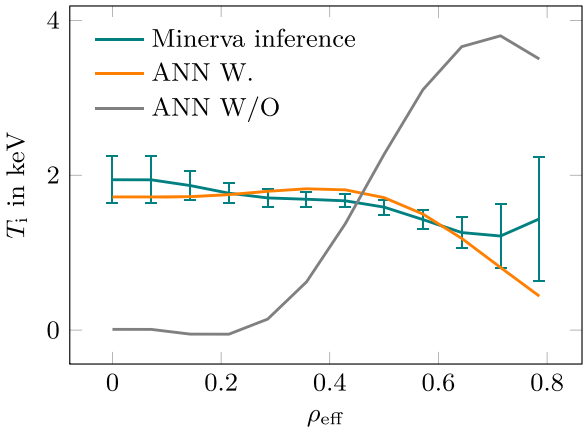


Figure 9. The standard Minerva Bayesian inference of a T_i profile is compared to the neural network inversion in two training cases. The orange and gray lines represent the output of the neural network trained on the training sets where the T_e profiles are sampled with (W) and without (W/O) constraints, respectively.

Table 2. The average distance from the measured data point to the 100 000 nearest neighbors in the training set is compared to the average distance of 10 test set samples from the corresponding 100 000 nearest neighbors in the training set, in the cases where the training set is created with (W) and without (W/O) constraint on the T_e profile prior distribution.

	Test set samples	Measured data
W/O constraint	47.8	1186.1
W constraint	63.5	198.4

that, even with the constraint, the sampled profiles are not necessarily monotonically decreasing, as shown in figure 8.

The improvement on the prediction capabilities of the NN when applied to the measured image is remarkable and it is shown in figure 9. The blue line in the plot denotes the mean value of 800 000 samples of ion temperature profiles drawn from the posterior distribution inferred within Minerva, and the corresponding standard deviation. The agreement between the NN prediction and the full Bayesian inference result is visibly better when the constraint is applied.

9. Results

A NN with architecture described in section 7 and illustrated in figure 5 has been evaluated on data from plasma shots of the first operational campaign at W7-X.

The prior distributions of the free parameters of the model used for the creation of the training set were all normal distribution functions with lower truncation at 0.0 keV. The T_e profile prior distribution had also upper truncation at 10 keV, whereas the other profiles had none. The values of the parameters of the GP squared exponential function defined in equation (5.2) were set to $\sigma_f = 2.0$, $\sigma_x = 0.3$, and $\sigma_y = 10^{-3}\,\sigma_f$ for the T_i profile and to $\sigma_f = 5.0$, $\sigma_x = 0.3$, and $\sigma_y = 10^{-3}\,\sigma_f$ for the T_e profile. The constraint discussed in section 8 was applied to the temperature profiles but not to the density profiles. The magnetic configuration was kept fixed during the sampling procedure and the NN was tested on data from shots having such configuration. A comparison between the standard Bayesian inference carried out within the Minerva framework and the NN inversion is shown in figure 10, for both ion and electron temperature profiles. The different plots in the figure refer to data from different shots and time points within a shot. In general, NN central prediction and full model central prediction, shown with the solid lines, are reasonably close to each other. In two cases the mismatch is especially pronounced: these are the cases of T_e for measurement 160223.007@1.0-1.2s and 160310.029@0.6-0.65s in the core region; when the uncertainties are taken into account, however, the agreement is still good. Providing uncertainties together with the NN central prediction is, indeed, an important point, especially when a comparison is made. We will now give an overview of how the uncertainties have been calculated; the readers that are interested in understanding the details can look at the work presented in [39]. The predicted profiles have been obtained from a committee of networks: a set of 10 NNs with same architecture but different weight initialization has been trained on the same training set. The error bars are calculated in a Bayesian fashion: the training procedure is seen as an inference problem on the network's weights, which gives as result an approximated Gaussian posterior distribution of the weights, centered on the network's weight vector found with the training process and whose standard deviation depends on the Hessian matrix of the loss function. The spread in the posterior produces then a spread in the network's prediction, and this is the source of the network's error bars. This procedure has been applied to the 10 networks in the committee. The committee prediction is then obtained by sampling a random member network, sampling a set of weights from the corresponding weight's posterior and then feeding the network with a sample of the input vector drawn from the XICS noise model. This corresponds to approximating the overall weight's posterior with a Gaussian mixture, where each Gaussian of the mixture is obtained with the single Gaussian approximation of each committee member, and it is centered on the weight vectors found with the different starting values [31]. This is necessary because, in principle, the single Gaussian approximation is valid only around the solution found with the training, and different starting values will lead to different solutions, i.e. different local minima of the optimization problem. In this way, it is then possible to take into account this fact and put together each committee member predictive distribution. The error bars shown in figure 10 also include uncertainties in the XICS measured data.

The full Bayesian model prediction is obtained as the average of 800 000 samples drawn from the posterior distribution of the corresponding free parameter, obtained with a MCMC sampler, and the error bars are obtained as the standard deviation of the samples.

It is important to note that the training set has been generated sampling with the LCFS constraint described in

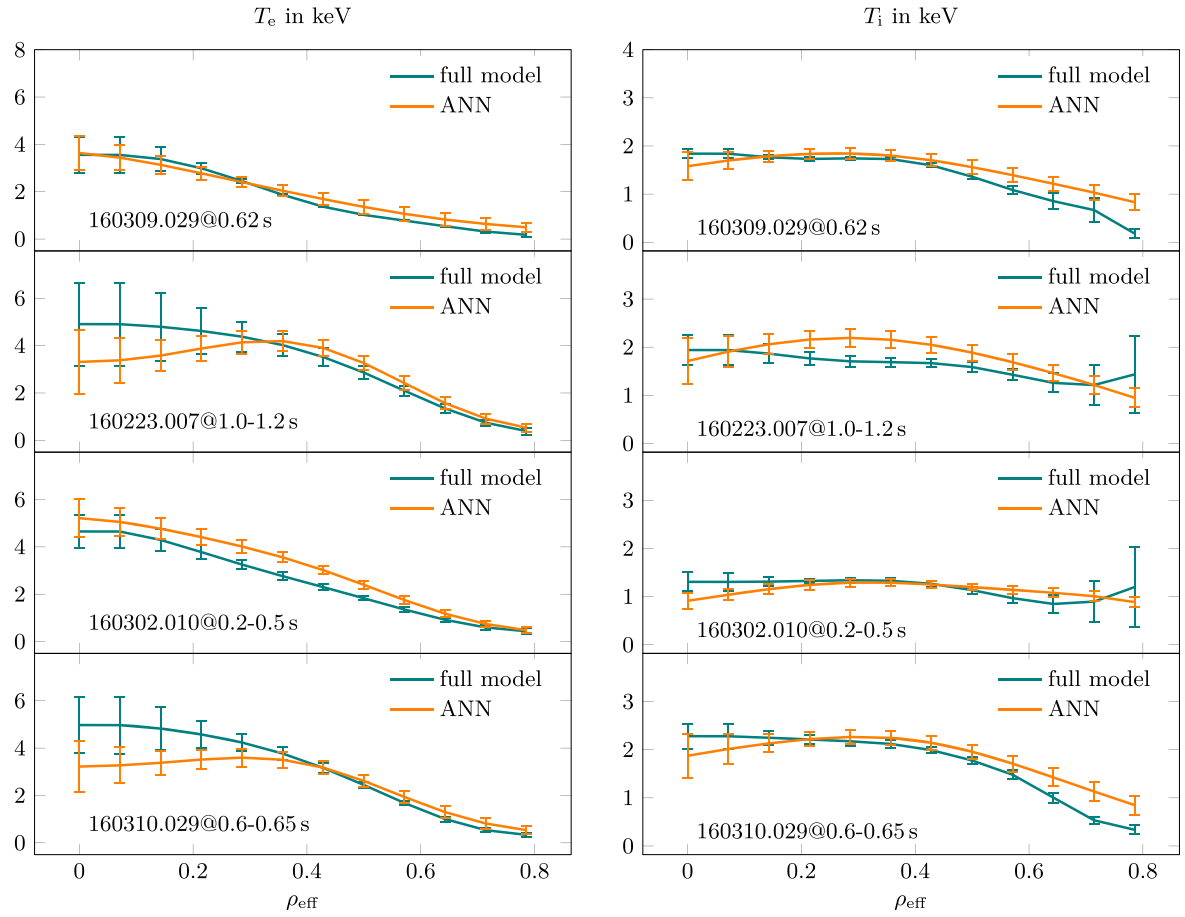


Figure 10. The results of the NN inversion compared to what obtained with standard Bayesian inference for different plasma shots. The left and right columns show T_e and T_i profiles respectively.

section 8: as a consequence, the profiles predicted by the networks show small variance towards $\rho = 1$. This is the reason why the uncertainties in the NN output of figure 10 are systematically lower for larger values of ρ_{eff}.

One of the main advantages in using NNs for data analysis is the speed-up that they provide. This is, indeed, substantial: the evaluation of one single data point takes $\sim 10\,\mu$s on a single CPU. The inference with MCMC sampling takes a few hours in similar conditions, thus the speed-up is of 10^9 orders of magnitude.

10. Conclusions

We have shown a Bayesian model oriented approach to NN training for the inference of ion and electron temperature profiles from data measured with an x-ray imaging diagnostic at W7-X. The model implemented within the Minerva framework is used to generate the training set, sampling from the prior distribution of the free parameters and from the likelihood function of the simulated data, i.e. from the joint distribution of the model. Since the joint distribution summarizes all the relevant properties of the Bayesian model, the trained network can be thought of as an approximation of the full Bayesian model. What makes this approach uncommon is the fact that the network is trained on a problem for which a model, and therefore a solution, already exists. The fact that we make use of a model to generate the training data gives us control over the features that we introduce in the training set, both from a point of view of the statistics and the physics. These features are completely determined by the joint distribution of the Bayesian model.

Since we can manipulate the prior probability distribution functions, we can also knowingly choose the model that better describes the measurements we expect to perform. This gives us the possibility to have control on the training of the network from the point of view of the physics parameters that, through the forward model, generate the expected observations. This is precisely what allows us to find a better training set for more accurate NN predictions.

The NN has been tested on measurements from different plasma shots from the first operational campaign at W7-X and compared with the results of the standard Bayesian inference. The first major advantage of this approach is the speed-up of the analysis, which can be carried out in tens of microseconds. The second advantage is that the sampling procedure for the creation of the training data only requires generic, not diagnostic-specific features of the Minerva model and thus is

in principle automatically applicable to any other diagnostic developed within the same framework.

Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014–2018 and 2019–2020 under grant agreement No. 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

ORCID iDs

A Pavone https://orcid.org/0000-0003-2398-966X
A Langenberg https://orcid.org/0000-0002-2107-5488
R C Wolf https://orcid.org/0000-0002-2606-5289

References

[1] Cybenko G 1989 Degree of approximation by superpositions of a sigmoidal function *Math. Control, Signals, Syst.* **2** 303–14
[2] Hornik K 1991 Approximation capabilities of multilayer feedforward networks *Neural Netw.* **4** 251–7
[3] Sonoda S and Murata N 2017 *Neural Network with Unboundedactivation Functions is Universal Approximator* https://arxiv.org/abs/1505.03654
[4] Svensson J *et al* 1998 Real-time ion temperature profiles in the JET nuclear fusion experiment *ICANN 98. Perspectives in Neural Computing* (London: Springer London) (https://doi.org/10.1007/978-1-4471-1599-1_30)
[5] Cannas B, Fanni A, Sonato P and Zedda M K 2007 A prediction tool for real-time application in the disruption protection system at JET *Nucl. Fusion* **47** 1559–69
[6] Wang S Y *et al* 2016 Prediction of density limit disruptions on the J-TEXT tokamak *Plasma Phys. Control. Fusion* **58** 055014
[7] Pautasso G and Tichmann C 2002 On-line prediction and mitigation of disruptions in ASDEX Upgrade *Nucl. Fusion* **42** 100–8
[8] Bishop C M, Roach C M and von Hellermann M G 1993 Automatic analysis of JET charge exchange spectra using neural networks *Plasma Phys. Control. Fusion* **35** 765–73
[9] Svensson J, von Hellermann M and König R W T 1999 Analysis of JET charge exchange spectra using neural networks *Plasma Phys. Control. Fusion* **41** 315–38
[10] Clayton D J *et al* 2013 Electron temperature profile reconstructions from multi-energy SXR measurements using neural networks *Plasma Phys. Control. Fusion* **55** 095015
[11] Svensson J and Werner A 2007 Large scale bayesian data analysis for nuclear fusion experiments *IEEE Int. Symp. on Intelligent Signal Processing* pp 1–6
[12] Langenberg A, Svensson J, Thomsen H, Marchuk O, Pablant N A, Burhenn R and Wolf R C 2016 Forward modeling of x-ray imaging crystal spectrometers within the Minerva Bayesian analysis framework *Fusion Sci. Technol.* **69** 560–7
[13] Wolf R C *et al* 2017 Major results from the first plasma campaign of the Wendelstein 7-X stellarator *Nucl. Fusion* **57** 102020
[14] König R *et al* 2015 The set of diagnostics for the first operation campaign of the Wendelstein 7-X stellarator *J. Instrum.* **10** P10002
[15] Krychowiak M *et al* 2016 Overview of diagnostic performance and results for the first operation phase in Wendelstein 7-X (invited) *Rev. Sci. Instrum.* **87** 11D304
[16] Bozhenkov S A *et al* 2017 The thomson scattering diagnostic at wendelstein 7-X and its performance in the first operation phase *J. Instrum.* **12** P10004
[17] Hoefel U *et al* 2019 Bayesian modelling of microwave radiometer calibration on the example of the wendelstein 7-X electron cyclotron emission diagnostic *Rev. Sci. Instrum.* **90** 043502
[18] Bitter M *et al* 2010 Objectives and layout of a high-resolution x-ray imaging crystal spectrometer for the large helical device *Rev. Sci. Instrum.* **81** 1–5
[19] Marchuk O, Wolf R and Kunze H J 2004 Modeling of He-like spectra measured at the tokamaks TEXTOR and TORE SUPRA *PhD Thesis* Ruhr-Universität Bochum
[20] Group T F R *et al* 1985 Dielectronic satellite spectrum of Helium-like Argon: a contribution to the physics of highly charged ions and plasma impurity transport *Phys. Rev.* A **32** 2374–83
[21] Vainshtein L A and Safronova U I 1978 Wavelengths and transition probabilities of satellites to resonance lines of H- and He-like ions *At. Data Nucl. Data Tables* **21** 49–68
[22] Sivia D and Skilling J 2006 *Data Analysis: A Bayesian Tutorial* (Oxford: Oxford University Press)
[23] MacKay D J C 1991 Bayesian Methods for Adaptive Models *PhD Thesis* California Institute of Technology
[24] Langenberg A *et al* 2018 Inference of temperature and density profiles via forward modeling of an x-ray imaging crystal spectrometer within the minerva bayesian analysis framework *Rev. Sci. Instrum.* Submitted
[25] Pearl J 1986 Fusion, propagation, and structuring in belief networks *Artif. Intell.* **29** 241–88
[26] Hirshman S P and Whitson J C 1983 Steepest-descent moment method for three-dimensional magnetohydrodynamic equilibria *Phys. Fluids* **26** 3553–68
[27] Rasmussen C E and Williams C K I 2006 *Gaussian Processes for Machine Learning* (Cambridge, MA: MIT Press)
[28] Langenberg A *et al* 2017 Argon Impurity Transport Studies at Wendelstein 7-X using x-ray Imaging Spectrometer Measurements *Nucl. Fusion* **57** 086013
[29] Kauderer-Abrams E 2017 *Quantifying Translation-Invariancein Convolutional Neural Networks* https://arxiv.org/abs/1801.01450
[30] Wong S C, Gatt A, Stamatescu V and McDonnell M D 2016 Understanding data augmentation for classification: when to warp? *CoRR* arXiv:abs/1609.08764
[31] Bishop C M 1995 *Neural Networks for Pattern Recognition* (Oxford: Oxford University Press)
[32] Lecun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proc. IEEE* **86** 2278–324
[33] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R and Fei-Fei L 2014 Large-scale video classification with convolutional neural networks *2014 IEEE Conf. on Computer Vision and Pattern Recognition* pp 1725–32
[34] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems 25* ed F Pereira *et al* (USA: Curran Associates, Inc.) pp 1097–105
[35] Lecun Y 1989 *Generalization and Network Design Strategies* (Amsterdam: Elsevier)

[36] LeCun Y, Boser B E, Denker J S, Henderson D, Howard R E, Hubbard W E and Jackel L D 1990 Handwritten digit recognition with a back-propagation network *Advances in Neural Information Processing Systems 2* ed D S Touretzky (San Mateo, CA: Morgan Kaufmann Publishers) pp 396–404

[37] Theano Development Team 2016 *Theano: A Python Frameworkfor Fast Computation of Mathematical Expressions* https://arxiv.org/abs/1605.02688

[38] Neal R M 1995 Bayesian learning for neural networks *PhD Thesis* University of Toronto

[39] Pavone A *et al* 2018 Bayesian uncertainty calculation in neural network inference of ion and electron temperature profiles at w7-x *Rev. Sci. Instrum.* **89** 10K102

[40] Knorr E M and Ng R T 1998 Algorithms for mining distance-based outliers in large datasets *Proc. 24rd Int. Conf. on Very Large Data Bases* pp 392–403 http://www.vldb.org/conf/1998/p392.pdf

[41] Dang T T *et al* 2015 Distance-based k-nearest neighbors outlier detection method in large-scale traffic data *2015 IEEE Int. Conf. on Digital Signal Processing (DSP)* pp 507–10

6.2. Article II

A. Pavone et al.
»Bayesian uncertainty calculation in neural network inference of ion and electron temperature profiles at W7-X«
In: *Review of Scientific Instruments* **89** (2018)

Synopsis
In this publication, [1] we describe how uncertainties of the neural network output can be calculated in a Bayesian framework. The framework is known as Bayesian neural network (BNN) and consists of an interpretation of the network model as a Bayesian model and the training problem as an inference problem. Under the so called Laplace approximation, it is possible to derive an analytical expression of the error bars dependent on the Hessian matrix of the training loss function. We apply this calculation to the case of the inference of ion and electron temperature profiles at W7-X, accounting also for noise in the input data and the presence of different local minima found by training, by using a Monte Carlo scheme of sampling both in input and weight space of multiple networks.

[1] Reproduced from A. Pavone et al. »Bayesian uncertainty calculation in neural network inference of ion and electron temperature profiles at W7-X«. In: *Review of Scientific Instruments* **89**, 10K102 (2018); https://doi.org/10.1063/1.5039286, with the permission of AIP Publishing.

REVIEW OF SCIENTIFIC INSTRUMENTS **89**, 10K102 (2018)

Bayesian uncertainty calculation in neural network inference of ion and electron temperature profiles at W7-X

A. Pavone,[1] J. Svensson,[1] A. Langenberg,[1] N. Pablant,[2] U. Hoefel,[1] S. Kwak,[1] R. C. Wolf,[1] and Wendelstein 7-X Team[1,a)]
[1]*Max-Planck-Institute for Plasma Physics, Greifswald 17491, Germany*
[2]*Princenton Plasma Physics Lab, Princeton, New Jersey 08543, USA*

(Presented 16 April 2018; received 7 May 2018; accepted 16 July 2018; published online 6 August 2018)

We make use of a Bayesian description of the neural network (NN) training for the calculation of the uncertainties in the NN prediction. Having uncertainties on the NN prediction allows having a quantitative measure for trusting the NN outcome and comparing it with other methods. Within the Bayesian framework, the uncertainties can be calculated under different approximations. The NN has been trained with the purpose of inferring ion and electron temperature profile from measurements of a X-ray imaging diagnostic at W7-X. The NN has been trained in such a way that it constitutes an approximation of a full Bayesian model of the diagnostic, implemented within the Minerva framework. The network has been evaluated using measured data and the uncertainties calculated under different approximations have been compared with each other, finding that neglecting the noise on the NN input can lead to an underestimation of the error bar magnitude in the range of 10%–30%. https://doi.org/10.1063/1.5039286

I. INTRODUCTION

In nuclear fusion research, neural networks (NNs) have been used for tasks such as prediction of disruption events from plasma parameters[1] and for diagnostic data analysis.[2] A special effort is often put in the development of real time systems.[3] In most of the applications, the output of the NN models is single "best guess" predictions, obtained with values of the adaptable parameters found minimizing a given cost function. We believe that, in order to have trust-worthy outcomes, uncertainty should be taken into account and delivered as part of the final predictions. In this paper, we will describe and make use of a Bayesian framework for the treatment of uncertainties, where the neural network model is seen as a Bayesian model and the training procedure is seen as an inference problem.[4,5] Applications of such a framework are scarcely encountered, although it posits a principled picture of neural network modeling. Its implementation relies on the calculation of the second derivative of the neural network's cost function with respect to the network weights, i.e., the Hessian matrix. This is an operation that scales with the square of the number of weights, i.e., as $O(W^2)$, where W is the number of weights. It is therefore a computational expensive calculation. However, the Hessian matrix needs to be calculated only once per training, as it is fixed at evaluation time, when the network is evaluated on the measurements. In Sec. II, we will illustrate the salient points of the Bayesian NN training from a theoretical point of view, describing three different procedures for the calculation of the uncertainties: the first one is derived neglecting noise in the NN input, the second one accounting for it, and the third one making use of a sampling scheme based on a non-linear multi-Gaussian approximation. In Sec. III, we will describe the specific application of the method to X-ray imaging crystal spectrometer (XICS) diagnostic data at W7-X, where the NN has been trained for the inference of electron and ion temperature profiles from XICS measurements. In Sec. IV, we will compare the two procedures where we either do take or do not take into account the noise in the NN input, and we will show a single illustrative example of uncertainty calculation with the multi-Gaussian sampling procedure.

II. BAYESIAN NEURAL NETWORKS

We shall describe now the salient points of the Bayesian perspective on NN training which will allow us to calculate uncertainties in the prediction. The notation used here is mostly taken from Ref. 5. The neural network is conceived here as a function f, which maps a generally multidimensional input vector $\mathbf{x}$ to a generally multidimensional output vector $\mathbf{y}$. The function f is also parametrized with a set of free parameters or weights $\mathbf{w}$, whose values are adapted or learned during the training procedure, so that it can be written that $\mathbf{y} = f(\mathbf{x}, \mathbf{w})$. In the specific case of this study, the input vector $\mathbf{x}$ would be an XICS measurement and $\mathbf{y}$ would be either an electron or ion temperature profile, T_e or T_i, respectively. In the analytical treatment that follows, we shall assume a one dimensional output y for the sake of clearer notation. The generalization to the multi-dimension output is straightforward. According to the traditional view, the NN training is the procedure employed to find a set of weight values $\mathbf{w}_{\mathrm{MP}}$ that minimizes a given *cost* or *loss* function $L(\mathbf{w})$. In regression problems such a function is often chosen to be the sum-of-square error between the NN's output y and the target training data t,

Note: Paper published as part of the Proceedings of the 22nd Topical Conference on High-Temperature Plasma Diagnostics, San Diego, California, April 2018.
a)See the authors list in T. S. Pedersen *et al.*, Nat. Commun. **7**, 13493 (2016).

$$L(\mathbf{w}) = \sum_{n=1}^{\mathrm{N}} (y^n - t^n)^2 + \nu(\mathbf{w}), \tag{1}$$

where N is the number of training samples, n is an index labelling the nth training sample, and $\nu(\mathbf{w})$ is a *regularizing term* which constrains the weight values to small values. The regularization allows finding a NN function which is smooth in w so that the generalization capabilities of the network are enhanced.[5] The set of weight values found is then used to make predictions at evaluation time. Using this approach, the outcome of the NN function is a single value estimate, given by $f(\mathbf{x}, \mathbf{w}_{\mathrm{MP}})$.

In the Bayesian framework of neural network training, the NN model is conceived as a Bayesian model, where the weights $\mathbf{w}$ are the free parameters and the target data of the training $\mathbf{t}_n$ are the observed data. According to Bayesian inference rules, a prior distribution $P(\mathbf{w})$ is assigned to the network weights before training, and a likelihood function $P(D|\mathbf{w})$ is assigned to the observed data, where $D \equiv (t_1, \ldots, t_{\mathrm{N}})$ denotes the target data from the training set. The training procedure is then an inference process on the network's weights. We can then write the Bayes formula to express the posterior distribution of the weights $P(\mathbf{w}|D)$ in terms of the prior and the likelihood function,

$$P(\mathbf{w}|D) = \frac{P(D|\mathbf{w})P(\mathbf{w})}{P(D)}, \tag{2}$$

where $P(D)$ is a normalization factor, independent of the weights, also known as the *evidence*. We have omitted the conditioning on the training input data $X \equiv (\mathbf{x}_1, \ldots, \mathbf{x_n})$ in all the terms, for the sake of simpler notation. The full outcome of the training, from the Bayesian point of view, is then not only a single set of values of the network's weights but also the entire posterior distribution $P(\mathbf{w}|D)$. At evaluation time, the spread of the distribution will then correspond to a distribution of the output, the *predictive distribution*. We shall see how, under certain assumption and approximations, we can get to an expression for the predictive error bars.

The first step in the application of such a method is the choice of the prior distribution $P(\mathbf{w})$ and the likelihood function $P(D|\mathbf{w})$. We shall assume for both of them Normal distributions. The reason behind this choice is that it allows making the analytical progress required to derive a mathematical expression for the error bars of the network's output. In this way, we will also recover results very well known and established under the traditional view of NN training. In the general case of a multi-layer neural network, we shall choose a prior of the form

$$P(\mathbf{w}) \propto \exp\left(-\frac{1}{2}\sum_k \alpha_k \|\mathbf{w}\|_k^2\right), \tag{3}$$

where $\alpha_k \equiv 1/\sigma_k^2$ with σ_k^2 denoting the variance of the distribution for the weights at the neural network's layer k. Concerning the likelihood function $P(D|\mathbf{w})$, we shall use an expression of the form

$$P(D|\mathbf{w}) \propto \exp\left(-\frac{\beta}{2}\sum_{n=1}^{N}(y^n - t^n)^2\right), \tag{4}$$

where $\beta \equiv 1/\sigma_D^2$ with σ_D^2 denoting the variance of the noise in the training target data, i.e., the spread of the distribution of the target variables, for a given, fixed input vector. We can now use the Bayes formula to find an expression for the posterior distribution of the weights. If we are interested in a single value solution, we can look for the weight values that maximize the posterior. This is equivalent to minimizing the negative logarithm of Eq. (2), $\ln(P(\mathbf{w}|D)) \equiv -S(\mathbf{w})$, which, substituting the expressions in Eqs. (3) and (4), can be written as

$$S(\mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}(y^n - t^n)^2 + \frac{1}{2}\sum_{k=1}^{L}\sum_{i=1}^{N_k}\alpha_k w_{k,i}^2, \tag{5}$$

where L is the number of layers in the network, i is an index labelling the weights at layer k, and N_k is the number of weights at layer k. Notice that we have omitted terms that do not depend on $\mathbf{w}$, specifically no contribution from the evidence term of the Bayes formula appears in this equation, since they would not have any effect in the minimization of $S(\mathbf{w})$ with respect to the weights. The expression in Eq. (5) resembles closely the one in Eq. (1). Indeed, this is how the Bayesian point of view and the traditional one come together. The first term on the right-hand side of Eq. (1) comes into Eq. (5) as the choice of the Gaussian noise model on the target training data, while the second one, the regularizing term, appears here as a consequence of the Gaussian prior on the network weights. In particular, we notice that the particular choice of the squared norm of the weight vector has led to a regularizing term well known in the neural network field as *L2 regularization* or *weight decay*: it has the effect of constraining the weights to small values with the consequence of improving the generalization of the network mapping, as described in Ref. 5.

An analytical expression for the full posterior $P(\mathbf{w}|D)$ can be found taking a Gaussian approximation of it around $\mathbf{w}_{\mathrm{MP}}$,[4] where $\mathbf{w}_{\mathrm{MP}}$ is a set of weight values found minimizing Eq. (5). This approximation is also known as the *Laplace approximation*, and it leads to

$$P(\mathbf{w}|D) \approx \exp\left(-S(\mathbf{w}_{\mathrm{MP}}) - \frac{1}{2}\Delta\mathbf{w}^{\mathrm{T}}\mathbf{A}\Delta\mathbf{w}\right), \tag{6}$$

where $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}_{MP}$ and $\mathbf{A} = \nabla\nabla\, S_{\mathrm{MP}} = \beta\nabla\nabla E^{\mathrm{MP}} + \sum_k \alpha_k \mathbf{I}$ is the Hessian matrix of the error function in Eq. (5), calculated with respect to the weights and evaluated at $\mathbf{w}_{\mathrm{MP}}$, with $E^{\mathrm{MP}} = \frac{1}{2}\sum_{n=1}^{N}(y^n - t^n)^2$ being the sum-of-square errors term evaluated at $\mathbf{w}_{\mathrm{MP}}$ and $\mathbf{I}$ being the identity matrix. We see therefore that $\mathbf{A}$ has two contributions, the first one coming from the choice of the likelihood function, controlled by the parameter β, and the second one coming from the choice of the prior distribution of the weights, controlled by the parameters α_k. This allows us now to calculate the distribution of the network outputs, when a new, unseen, input vector $\mathbf{x}$ is provided to the trained network, at evaluation time. It is obtained by marginalization over the network's weights,

$$P(t|\mathbf{x}, D) = \int P(t|\mathbf{x}, \mathbf{w})P(\mathbf{w}|D)\,\mathrm{d}\mathbf{w}, \tag{7}$$

where we have now explicitly included in the notation the dependence on the new input vector $\mathbf{x}$. The distribution $P(t|\mathbf{x}, \mathbf{w})$, which is evaluated at a fixed value of the weight

vector, is given by the noise model on the target data, as in Eq. (4). After some manipulation, we get to the final expression

$$P(t|\mathbf{x}, D) = \frac{1}{(2\pi\sigma_t'^2)^{1/2}} \exp\left(-\frac{(t - y_{\mathrm{MP}})^2}{2\sigma_t'^2}\right), \tag{8}$$

where

$$\sigma_t'^2 = \frac{1}{\beta} + \mathbf{g}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{g}, \tag{9}$$

where $\mathbf{g} \equiv \nabla_{\mathbf{w}} y|_{\mathbf{w}_{\mathrm{MP}}}$. The distribution of the network's output is then given by a Gaussian distribution, centered at the network prediction obtained with weights $\mathbf{w}_{\mathrm{MP}}$ and with standard deviation given by Eq. (9). The contribution to the predictive error has two components: one arising from the noise on the target data, controlled by β, and the other arising from the posterior width, controlled by $\mathbf{A}$. Equation (9) corresponds to the *first* procedure to calculate uncertainties.

So far, we have neglected uncertainties in the neural network input. This is of course not ideal when the input is a measured quantity with noise, as it is in our application. It can be shown[6] that an expression for the predictive error, which includes noise in the input, is given by

$$\sigma_t^2 = \sigma_t'^2 + \sigma_x^2 \mathbf{h}^{\mathrm{T}}\mathbf{h}, \tag{10}$$

where $\mathbf{h} \equiv \nabla_{\mathbf{x}} y|_{\mathbf{x}_{\mathrm{v}}}$, $\mathbf{x}_{\mathrm{v}}$ is the input vector, and σ_x^2 is the variance of the noise of the input vector, here assumed to be Gaussian. Equation (10) corresponds to the *second* procedure to calculate uncertainties. Three main assumptions that have been done to get to Eq. (10): the posterior distribution of the weights has been approximated with a Gaussian distribution around $\mathbf{w}_{\mathrm{MP}}$, the network function $y(\mathbf{x}; \mathbf{w})$ has been approximated by its linear expansion around $\mathbf{w}_{\mathrm{MP}}$ and x_{v} in the calculation of σ_t' and σ_t, respectively. Moreover, the Laplace approximation of the weight's posterior is only valid around $\mathbf{w}_{\mathrm{MP}}$. However, several minima of the cost function are likely to exist and they can be found training the network with different initial values of the weights. The single-Gaussian approximation so far described does not take them into account. In order to account for them, it is possible to approximate the posterior of the weights by a sum of Gaussians, each one centered on each of the minima.[5] This can be accomplished by training a *committee* of networks, where each member is trained with different initialization values, and carrying out the Laplace approximation of the posterior for each of them. The overall posterior is then given by

$$P(\mathbf{w}|D) = \sum_i P(\mathbf{w}|m_i, D)P(m_i|D), \tag{11}$$

where $P(m_i|D)$ is the *a priori* distribution of the minima m_i and $P(\mathbf{w}|m_i, D)$ is the posterior distribution of the weights corresponding to the local minima m_i, which can be approximated with the Laplace approximation. The predictive distribution can still be written as in Eq. (7), where now the second term on the right-hand side is obtained from Eq. (11). Assuming $P(m_i|D)$ to be uniform, we can obtain the uncertainties for a multi-Gaussian approximation of the posterior distribution in the following way: (i) we train a number of NNs with different weight initialization, corresponding to the NN functions f_i, (ii) for each of them, we calculate the posterior of the weights under the Laplace approximation, (iii) we obtain samples from the predictive distribution by randomly choosing one member of the committee, say member i, then, sampling a set of weight values, $\mathbf{w}_{\mathrm{MP}}^i$, and an input vector x^*, from the weight posterior and the input noise model, respectively, and calculating the corresponding NN output: $y_i = f_i(\mathbf{w}_{\mathrm{MP}}^i, \mathbf{x}^*)$. The whole procedure is repeated a number of times equal to the desired number of samples. The advantage of this sampling procedure to the estimation of the uncertainties is that it does not make use of the assumption of linearity of the NN function around $\mathbf{w}_{\mathrm{MP}}$ and the input vector $\mathbf{x}$. It is therefore more accurate. However, it requires large computational time, and it is therefore not suitable in those applications where the execution time is a concern. This is our *third* procedure for calculating uncertainties, and we will refer to it as the multi-Gaussian sampling scheme.

III. APPLICATION TO XICS DIAGNOSTIC DATA AT W7-X

A. The XICS diagnostic at W7-X

The XICS diagnostic at W7-X is equipped with a spherical bent crystal to image X-ray emission of Ar impurities. The emission is then collected on a CCD detector. The diagnostic layout and initial measurements during the first operational phase at W7-X have been described in Refs. 7–11. The collected images have spatial resolution along the vertical dimension, corresponding to different lines of sight, and wavelength resolution along the horizontal one. The wavelength range is 3.94–4.0 Å for He-like Ar spectra. From the measured data, it is then possible to reconstruct ion and electron temperature profiles. The ion temperature affects the Doppler broadening of the spectral lines, whereas the electron temperature affects the relative intensities. Given the electron density profile n_e, the impurity density profiles can be obtained.[8,12] A forward model of the diagnostic[7] has been developed within the Minerva Bayesian modeling framework,[13] and it is used for the inference of the plasma profiles of interest.

B. Neural networks as approximate Bayesian models

In the XICS Bayesian model, a prior distribution is assigned to the free parameters, which are temperature, electron and impurity density profiles, and a likelihood function is assigned to the observed quantities. A neural network has been trained with the goal to approximate the full model Bayesian inference. The training scheme is described in detail in Ref. 14. The training set is obtained sampling from the joint distribution of the model $P(T, I) = P(I|T)P(T)$: a set of free parameters is sampled from the prior distribution $P(T)$, and subsequently synthetic data are sampled from the likelihood function $P(I|T)$. The distribution $P(I|T)$ represents the noise model on the XICS measurements, which is given by a Gaussian distribution with mean and variance given by the forward model predicted photon counts. When sampling from the priors, all n_e, T_e, T_i, and impurity density profiles were free to vary, but only the T_i and T_e profiles were used as the target of the network's training. The set of sampled synthetic images constitutes the network's input during training. Note that such a training set is made exclusively of data synthesized with the

Bayesian model. The profiles are expressed with respect to the effective radius, defined as $\rho_{\text{eff}} = \sqrt{\psi/\psi_{\text{LCFS}}}$, where ψ is the magnetic flux and ψ_{LCFS} is the flux at the last closed flux surface. It is worth to emphasize that, because of training on a given fixed model, any systematic deviation introduced by the specific choice of the model would be reflected in the network's inversions.

IV. RESULTS

Two convolutional neural networks[14,15] (CNN), each one with two convolutional layers C1 and C2, followed by one hidden fully connected layer M1 and the output layer M2, have been trained on the inference of T_i and T_e profiles, respectively. The training has been carried out in the Bayesian scheme described in Sec. II. The values of $\beta = 10$ and $\alpha_k = (\alpha_{\text{C1}} = 68.00, \alpha_{\text{C2}} = 58.33, \alpha_{\text{M1}} = 576.67, \alpha_{\text{M2}} = 5.83)$ were used. The Hessian matrix $\mathbf{A}$ has been calculated in the diagonal approximation. The error bars calculated with Eqs. (9) and (10) have been compared with each other, and the results are shown in Figs. 1 and 2. In Fig. 1, the average relative uncertainty $\langle\sigma_{\text{rel}}\rangle$ calculated accounting and without accounting for the noise in the input, according to $\langle\sigma_{t,\text{rel}}\rangle \equiv \langle\sigma_t(\rho_{\text{eff}})/T(\rho_{\text{eff}})\rangle$ and $\langle\sigma'_{t,\text{rel}}\rangle \equiv \langle\sigma'_t(\rho_{\text{eff}})/T(\rho_{\text{eff}})\rangle$, respectively, is shown for each spatial location of both profiles. The average has been calculated from data collected across 15 plasma shots of the first operational campaign at W7-X. In the case of T_{e} profiles, it is found that $\langle\sigma_{t,\text{rel}}\rangle \approx 0.2$ for $\rho_{\text{eff}} < 0.4$, and $0.3 < \langle\sigma_{t,\text{rel}}\rangle < 0.4$ for $\rho_{\text{eff}} > 0.5$. In the case of T_{i} profiles, instead, the quantity $\langle\sigma_{t,\text{rel}}\rangle$ shows less variation across the different locations, assuming mostly values approximately equal to 0.1. The difference between $\langle\sigma_{t,\text{rel}}\rangle$ and $\langle\sigma'_{t,\text{rel}}\rangle$ reflects what also emerges from Fig. 2 and that we will comment in the next paragraph: the two quantities mostly diverge at the positions corresponding to the plasma core and toward the edge. In Fig. 2, the distribution of the contribution of the input noise term relative to the total error bar magnitude, calculated as $\Delta\sigma_{\text{rel}}(\rho_{\text{eff}}) \equiv (\sigma_t(\rho_{\text{eff}}) - \sigma'_t(\rho_{\text{eff}}))/\sigma_t(\rho_{\text{eff}})$, for each spatial location of the T_e and T_i profiles is shown. The distribution has been computed from the same data used for Fig. 1. The orange line connects the mean of each distribution. In the case of the T_e profile (left), it is found that for a significant proportion of the analyzed data the input noise contribution accounts for more than 10%, and mostly less than 20%, of the total error bar magnitude, especially in the positions corresponding to the core and toward the edge, $\rho_{\text{eff}} < 0.2$ and $\rho_{\text{eff}} > 0.7$. In the case of the T_i profiles, instead, it is found in a significant number of cases that the same noise source accounts for more than 20% of the total error, again mainly in the positions corresponding to the core and toward the edge of the plasma, $\rho_{\text{eff}} < 0.2$ and $\rho_{\text{eff}} > 0.6$. The input uncertainties are, therefore, in general not negligible. The purpose of Fig. 3 is to illustrate the result of the sampling procedure described at the end of Sec. II. The network's inversion has been applied on a single measured data point for illustrative purposes. The network's input has been obtained averaging over a 500 ms range. The bundle of gray lines is made of 1000 samples obtained sampling from the multi-Gaussian approximation of the network's weights, accounting also for the noise in the input. The orange line shows the network's prediction and corresponding error bar calculated with the single-Gaussian approximation, Eq. (10). The error bars show a $2\sigma_t$ deviation. The sampling scheme based on the multi-Gaussian approximation is a more accurate method to calculate uncertainties, which comes at the price of larger computational time: it is therefore suitable in NN applications where execution time is not a concern.

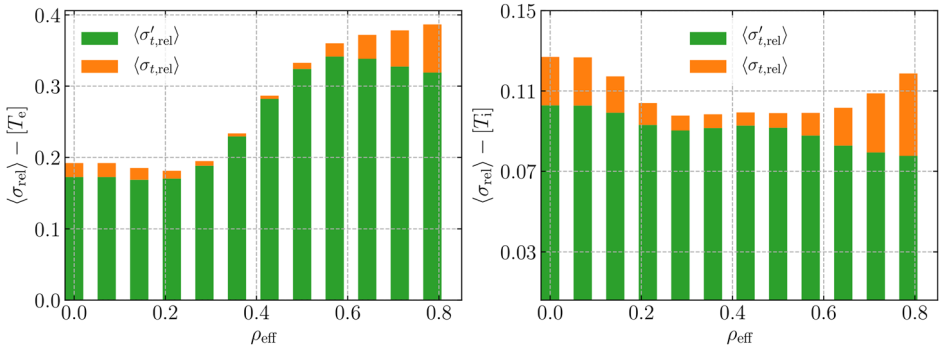


FIG. 1. The average value of the relative uncertainty calculated with (orange bars) and without (green bars) input noise contribution for both T_{e} (left) and T_{i} (right) profiles, as found from data collected across different experiments.

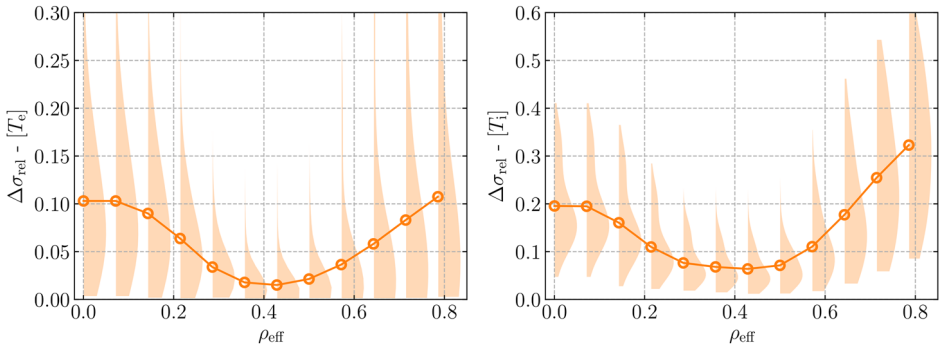


FIG. 2. The distribution of the contribution of the input noise term relative to the total error bar magnitude calculated across the data point from different plasma shots for each spatial location in the T_{e} (left) and T_{i} (right) profiles. The orange line connects the mean of the distribution at each position.

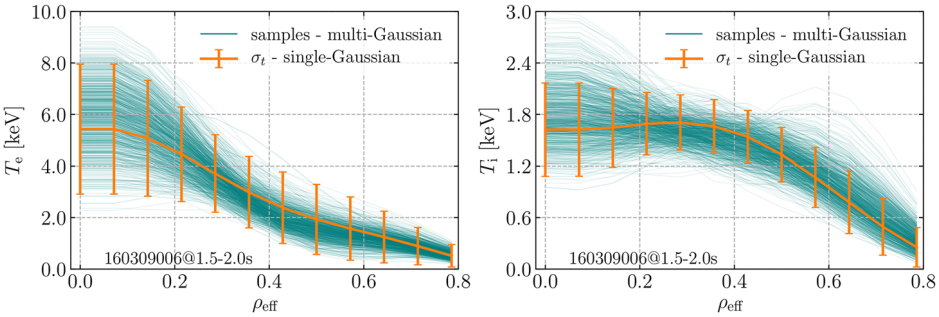


FIG. 3. The neural network prediction and uncertainties calculated in the case of the multi-Gaussian sampling procedure (gray lines) and the single-Gaussian approximation (orange line), for both T_e (left) and T_i (right) profiles.

ACKNOWLEDGMENTS

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under Grant Agreement No. 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

[1] B. Cannas *et al.*, Nucl. Fusion **47**, 1559 (2007).
[2] J. Svensson *et al.*, Plasma Phys. Controlled Fusion **41**, 315 (1999).
[3] J. Svensson *et al.*, in *Perspectives in Neural Computing* (ICANN, 98).
[4] D. J. C. Mackay, "Bayesian methods for adaptive models," Ph.D. thesis, California Institute of Technology, 1991.
[5] C. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, 1995).
[6] A. Wright, IEEE Transactions On Neural Networks **10**, 1261 (1999).
[7] A. Langenberg *et al.*, Fusion Sci. Technol. **69**, 560 (2016).
[8] A. Langenberg *et al.*, Nucl. Fusion **57**, 086013 (2017).
[9] R. König *et al.*, J. Instrum. **10**, P10002 (2015).
[10] M. Krychowiak *et al.*, Rev. Sci. Instrum. **87**, 11D304 (2016).
[11] N. A. Pablant *et al.*, in *41st EPS Conference on Plasma Physics* (EPS, 2014), Vol. 38F.
[12] A. Langenberg *et al.*, in *43rd European Physical Society Conference on Plasma Physics* (EPS, 2016), Vol. 40A.
[13] J. Svensson and A. Werner, in *IEEE International Symposium on Intelligent Signal Processing* (IEEE, 2007).
[14] A. Pavone, "Neural network approximation of Bayesian models for the inference of ion and electron temperature at W7-X" (unpublished).
[15] Y. LeCun *et al.*, Proc. IEEE **86**, 2278 (1998).

6.3. Article III

A. Pavone et al.
»Measurements of visible bremsstrahlung and automatic Bayesian inference of the effective plasma charge Z_{eff} at W7-X«
In: *Journal of Instrumentation* **14** (2019)

Synopsis
The publication [2] describes the first available measurements of visual bremsstrahlung and the inference of the plasma effective charge Z_{eff} at W7-X. Both the features of the measurement device and the Bayesian model used in the inference are described. Especially, Bayesian inference is run automatically after each plasma shot and it includes the inference of electron temperature and density profiles from independent measurements of the Thomson scattering diagnostic based on Gaussian processes.

[2] https://doi.org/10.1063/1.5039286

Published by IOP Publishing for Sissa Medialab

Received: *June 26, 2019*
Accepted: *September 10, 2019*
Published: *October 2, 2019*

3rd European Conference on Plasma Diagnostics (ECPD2019)
6–10 May 2019
Lisbon, Portugal

Measurements of visible bremsstrahlung and automatic Bayesian inference of the effective plasma charge Z_{eff} at W7-X

A. Pavone,[a,1] **U. Hergenhahn,**[a] **M. Krychowiak,**[a] **U. Hoefel,**[a] **S. Kwak,**[a] **J. Svensson,**[a] **P. Kornejew,**[a] **V. Winters,**[b] **R. Koenig,**[a] **M. Hirsch,**[a] **K.-J. Brunner,**[a] **E. Pasch,**[a] **J. Knauer,**[a] **G. Fuchert,**[a] **E.R. Scott,**[a] **M. Beurskens,**[a] **F. Effenberg,**[c] **D. Zhang,**[a] **O. Ford,**[a] **L. Vanó,**[a] **R.C. Wolf**[a] **and the W7-X team**

[a] *Max-Planck-Institut für Plasmaphysik, Teilinstitut Greifswald, Wendelsteinstrasse 1, D-17491 Greifswald, Germany*
[b] *University of Wisconsin-Madison, Madison, WI 53706-1609, U.S.A.*
[c] *Princeton Plasma Physics Laboratory, Princeton University, Princeton, NJ 08543, U.S.A.*

E-mail: andrea.pavone@ipp.mpg.de

[1] Corresponding author.

https://doi.org/10.1088/1748-0221/14/10/C10003

Abstract: The effective charge Z_{eff} indicates the overall impurity contamination of a plasma. Z_{eff} can be derived experimentally from the intensity of the plasma bremsstrahlung emission. We describe here the diagnostic set-ups and the Bayesian modeling allowing the inference of Z_{eff} at W7-X. First results from the operational campaigns in 2017 and 2018 are shown. Measurements of the visible plasma radiation along a single line-of-sight traversing the core plasma has been carried out using a compact USB-spectrometer with a time resolution of 100 ms. A spectral region (627–641 nm) that is free from line emission is selected for the analysis of the bremsstrahlung emission, which also depends on electron temperature and density profiles. Electron temperature profiles are derived from either the electron cyclotron emission or the Thomson scattering diagnostic. Electron density profiles, however, have their shape information derived from Thomson scattering measurements and absolute values from single line-of-sight interferometry measurements. The Minerva framework is used to infer the profiles with Gaussian processes and develop a Bayesian model of the bremsstrahlung emission to infer line averaged Z_{eff}. The sensitivity of the diagnostic enables Z_{eff} measurements down to the lowest core electron densities observed in the last campaign of 0.75×10^{19} m^{-3} with a statistical relative error of ≈50% (Z_{eff} = 3.2, 100 ms integration time). The analysis is automated to routinely compute Z_{eff} after every plasma discharges.

Keywords: Analysis and statistical methods; Plasma diagnostics - interferometry, spectroscopy and imaging

2019 JINST 14 C10003

Contents

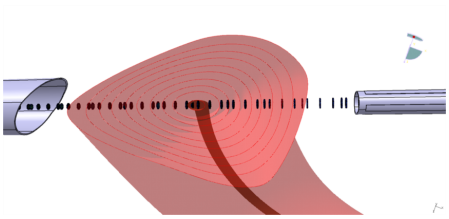
2019 JINST 14 C10003

1 Introduction

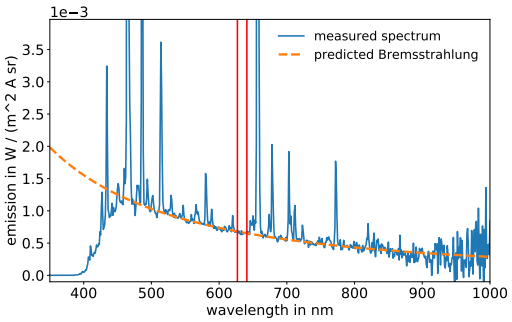
In magnetically confined fusion plasmas, the study of impurity behavior is important for the assessment of plasma performance and the investigation of impurity transport [1]. The effective charge $Z_{\mathrm{eff}} = \sum_i n_i Z_i^2 / \sum_i n_i Z_i$ is related to the concentration of impurities and indicates the overall contamination of the plasma with mainly low-Z impurities, e.g. Carbon. It is usually derived experimentally from the plasma ion-electron bremsstrahlung emission in the visible, IR or X-Ray spectral region [2–4], using an independent measurement of the electron density n_e and temperature T_e. In this work, we illustrate the diagnostic set-ups and the Bayesian modeling that allowed the inference of Z_{eff} at W7-X and we will show results from the OP1.2 experimental campaign, obtained from measurements performed with a compact USB-spectrometer. Also, we will describe the diagnostic set-up of other diagnostic systems which were routinely observing bremsstrahlung emission as well.

2 The single line-of-sight USB-spectrometer diagnostic

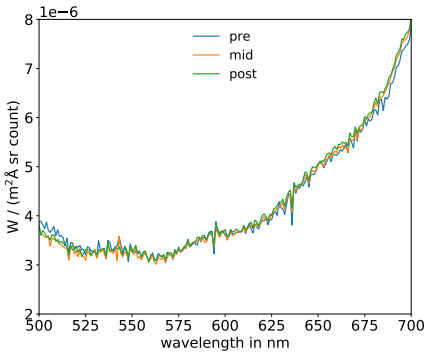
A compact USB-spectrometer (Red Tide USB650, Ocean Optics) collects light along a single line-of-sight that goes through the plasma core of W7-X, as shown in figure 1a. The system collects light in the visible and near infrared wavelength region, approximately from 350 to 1000 nm, as shown in figure 1b, with a time resolution of 100 ms. Due to the low light level of the calibration source only the spectral range above 450 nm can be used for the analysis. The figure also shows the bremsstrahlung emission predicted with $Z_{\mathrm{eff}} \approx 1.5$. Details about the predictive forward model are given in the following sections. In order to infer Z_{eff} from the measured spectrum, we have selected and used a fixed wavelength window that is free of line radiation, marked with two red vertical lines in the figure, in the range of $\approx 627 - 641$ nm. The system was absolutely calibrated by measuring the diagnostic response to an Ulbricht sphere of known emissivity. The calibration has been carried out prior (pre), during (mid) and after (post) both the experimental campaigns OP1.2a and OP1.2b. The sensitivity of the diagnostic system as a function of wavelength, in units of W / (m^2 Å sr count), is shown in figure 1c. Multiplication by this quantity converts the measured raw data to spectral



(a)



(b)



(c)

Figure 1. Figure (a) shows the single line of sight of the USB-spectrometer, the triangular W7-X plasma cross section and the magnetic axis in red. The line of sight ends in an opposite port (left hand side) thus eliminating the problem of plasma light reflections at the vessel walls. Figure (b) shows the measured spectrum of the photon flux of the plasma bremsstrahlung and line radiation (blue) as well as the predicted bremsstrahlung level with a $Z_{\text{eff}} \approx 1.5$ (dashed line). The two red vertical lines indicate the wavelength range selected and used in the analysis $\approx 627 - 641$ nm. Figure (c) shows the sensitivity spectrum of the diagnostic in the wavelength range of between 500 nm and 700 nm. Three different measurements were carried out, prior (pre), during (mid) and after (post) the experimental campaign OP1.2.

power density in absolute units (see also $1/C(\lambda)$ in equation (3.1)). According to the time interval in which the data were collected, the corresponding calibration curve is applied to the data. The relative variation between the different curves is always < 10% in the wavelength range shown in the figure, indicating that the calibration remained fairly constant during the course of the campaign.

3 Bayesian modeling and inference

A model to calculate the bremsstrahlung emission is implemented in the Minerva framework [5]. The Minerva framework allows to carry out Bayesian modeling and inference in complex systems. The expected measured signal $S(\lambda)$ can be calculated from the bremsstrahlung emission at a given wavelength $V(\lambda)$ collected along the line of sight, according to equation (3.1):

$$S(\lambda) = C(\lambda)V(\lambda) = C(\lambda) \int g_{\text{ff}}(Z_{\text{eff}}, T_{\text{e}}, \lambda) \frac{n_{\text{e}}^2 Z_{\text{eff}}}{\sqrt{k_b T_{\text{e}}}} \exp\left(\frac{hc}{\lambda k_b T_{\text{e}}}\right) \frac{1}{\lambda^2} dl \tag{3.1}$$

2019 JINST 14 C10003

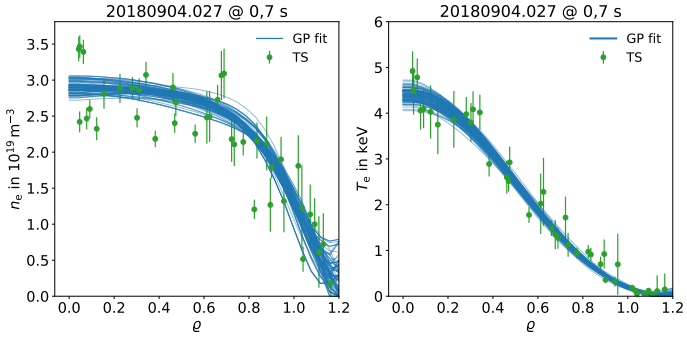


Figure 2. The electron temperature and density profiles measured by the Thomson scattering diagnostic (TS) are fitted with a Gaussian process (GP) model within the Minerva framework. The blue lines are samples from the posterior distribution found with Bayesian inference. The dots represent the measured data points together with their respective error bars.

where the integration is done along the line of sight path, $g_{\mathrm{ff}}(Z_{\mathrm{eff}}, T_{\mathrm{e}}, \lambda)$ is the free-free Gaunt factor modeled in Minerva according to [6], $C(\lambda)$ is an absolute calibration factor (figure 1c), and the remaining symbols are used in the conventional way referring to the respective physics constants in SI units. The single line of sight diagnostic does not allow to resolve the spatial profile of Z_{eff}, therefore, when Z_{eff} is used in the calculation of the emission along the line-of-sight, it is assumed to be constant.

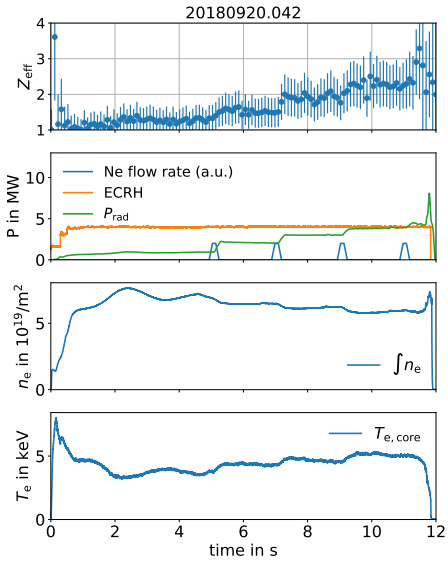
According to equation (3.1), n_{e} and T_{e} are quantities required to calculate the expected emission. They are provided by a spatially resolved Thomson scattering [7] measurement and a line-integrated n_{e} measurement by the dispersion interferometer diagnostic [8], which constrains the n_{e} absolute values. Both profiles are first inferred within the Minerva framework with a Gaussian processes (GP) Bayesian model [9], where the covariance of the normal prior distributions of the profiles is modeled with a covariance function, parametrised in terms of the profile length scale. The posterior solution found is "smooth" and it is affected by the number of observed data points and their respective uncertainties, which in this case do not include systematic errors. An example case of such procedure is shown in figure 2, where the samples from the posterior distribution are shown in blue and the measured data points are labeled as TS. The coordinate on the x-axis is the effective radius $\rho = \sqrt{\psi/\psi_{\mathrm{LCFS}}}$, where ψ is the magnetic flux and ψ_{LCFS} is the magnetic flux at the last closed flux surface.

Since the Bayesian analysis is meant to be carried automatically after every plasma discharge, a fallback solution is provided for those cases in which Thomson scattering measurements are not available. The n_{e} profile is assumed to be parabolic, and absolute values are scaled accordingly using interferometer measurements, as previously mentioned; the T_{e} profile, on the other hand, is obtained from measurements by the electron cyclotron emission (ECE) diagnostic using the cold resonance approximation [10, 11].

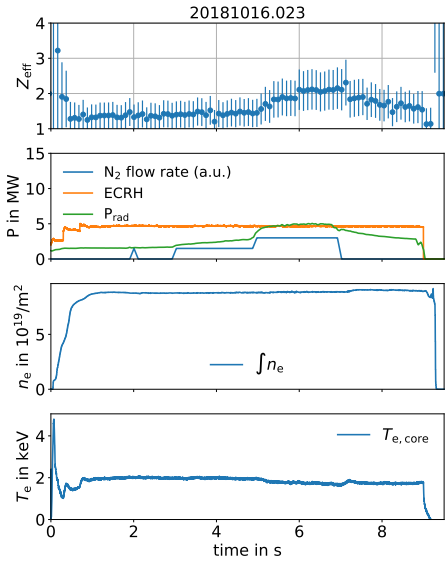
4 Results

Z_{eff} can be inferred by comparing predicted and measured bremsstrahlung emission signals. Two example cases from experiments 20180920.042 and 20181016.023 are depicted in figure 3a and 3b, showing discharges in which the plasma was seeded by Ne and N_2, respectively [12]. Ne was injected

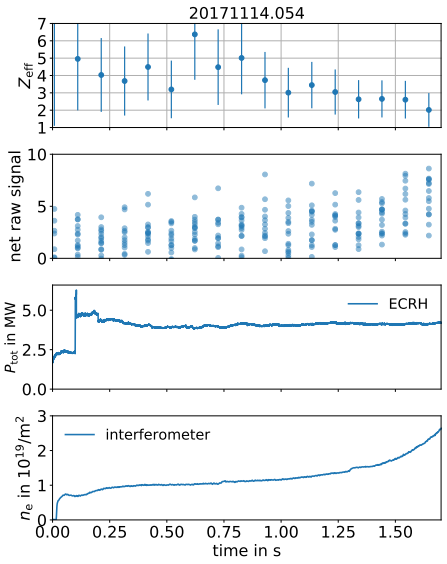
2019 JINST 14 C10003



(a) A case of Ne injection.



(b) A case of N_2 injection.



(c) A low density discharge.

Figure 3. The time evolution of Z_{eff} with respective error bars and other relevant parameters for three example cases. Figures (a) and (b) show the case of two seeding experiments, with Ne and N_2 respectively. A discrete increase in Z_{eff} and total radiated power P_{rad} is observed after each injection of seeding gas. The low density discharge in figure (c) allowed to assess the sensitivity of the diagnostic: a line-of-sight averaged density of 0.75×10^{19} m^{-3} at 0.5 s allowed to measure a $Z_{\mathrm{eff}} \approx 3.2$ with a statistical error of $\approx 50\%$.

2019 JINST 14 C10003

at 5, 7, 9 and 11 s for 200 ms, whereas N_2 was injected at 2 s for 50 ms and continuously at 3-7 s; a second valve was open at 5-7 s, increasing the gas flow rate by a factor of ≈ 2. Corresponding to the injection times, we observe an increase in the Z_{eff} values and in the total plasma radiation measured with a bolometer [13]. The Z_{eff} error bars are obtained taking into account signal statistics, the absolute calibration and the uncertainties in the n_e and T_e profiles. In the figures, the time evolution of other relevant parameter is also shown: the power from the electron cyclotron resonance heating (ECRH), the line integrated density n_e measured with the interferometer, and the value of the electron temperature in the core T_e as measured with the ECE diagnostic.

In figure 3c we show the case of a very low density discharge demonstrating the lower sensitivity limit of the diagnostic at 100 ms integration time. A line-of-sight averaged density n_e of 0.75 $\times 10^{19}$ m^{-3} as measured with the interferometer at ≈ 0.5 s allowed to measure a $Z_{\text{eff}} \approx 3.2$ with a statistical error of $\approx 50\%$. In the second plot from the top, the measured signal for each pixel in the considered wavelength range is shown at every time point; the large noise level is clearly visible.

The Z_{eff} values inferred with the USB-spectrometer were also compared to those found with the charge exchange recombination spectroscopy (CXRS) system [14, 15] for two discharges 20180927.042 and 046 in which He was injected during the experiments and the neutral beam injection (NBI) system was active. According to a preliminary analysis, the CXRS system allowed to measure a H/He ratio of 0.3/0.7 and 0.85/0.15 in the first and second discharge respectively, and a 2% concentration of C^{6+} in the core in both experiments. From these values, the lower limit of Z_{eff} was then estimated as ≈ 2.1 and 1.7, compared to 1.9 ± 0.4 and 1.5 ± 0.3 as found with the USB-spectrometer.

5 Conclusions and future works

A compact, single line-of-sight USB-spectrometer allows to measure bremsstrahlung emission and infer Z_{eff} at the Wendelstein 7-X stellarator. The diagnostic was operating during the OP1.2 routinely providing the line-of-sight averaged Z_{eff}. A Bayesian model was implemented in the Minerva framework, allowing to infer Z_{eff} and to provide Gaussian process fits of n_e and T_e profiles combining measurements of the Thomson scattering and dispersion interferometer measurements.

In the context of future works, we want to mention that more systems dedicated for the Z_{eff} measurement are also available and collected data during the experiments, but they are not yet modeled and inference was not carried out on such measurements. Specifically, two additional detectors share the same line of sight of the USB-spectrometer. One collects light emitted in the near infrared range of 750-950 nm, with spectral resolution of ≈ 1 nm and typical time resolution of 50 ms. A second one collects visible light at 523 nm and 630 nm using interference filters with a bandwith of 2 nm, and time resolution of 100 kHz. A third system is equipped with 27 lines of sight and operate in the range 750-950 nm, and can therefore provide information to infer spatially resolved Z_{eff}. In future works we aim at modeling all these systems and adding other diagnostics containing information on Z_{eff} (as CXRS, X-ray spectrometers) within the Minerva framework, so to exploit all available information for the inference of Z_{eff} profiles.

2019 JINST 14 C10003

Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work was supported in part by the U.S. Department of Energy (DoE) under grant DE-SC0014210 and funding by the Department of Engineering Physics and of the College of Nuclear Engineering at the University of Wisconsin — Madison, U.S.A.

References

[1] G. Verdoolaege, M.G. Von Hellermann, R. Jaspers, M.M. Ichir and G. Van Oost, *Integrated bayesian estimation of Z_{eff} in the TEXTOR tokamak from bremsstrahlung and CX impurity density measurements*, *AIP Conf. Proc.* **872** (2006) 541.

[2] M. Krychowiak, R. König, T. Klinger and R. Fischer, *Bayesian analysis of the effective charge from spectroscopic bremsstrahlung measurement in fusion plasmas*, *J. Appl. Phys.* **96** (2004) 4784.

[3] H.Y. Zhou, S. Morita, M. Goto and M.B. Chowdhuri, *Z_{eff} profile measurement system with an optimized Czerny-Turner visible spectrometer in large helical device*, *Rev. Sci. Instrum.* **79** (2008) 10F536.

[4] H. Meister et al., *An integrated system to measure the effective charge of fusion plasmas in the ASDEX upgrade tokamak*, *Rev. Sci. Instrum.* **74** (2003) 4625.

[5] J. Svensson and A. Werner, *Large scale bayesian data analysis for nuclear fusion experiments*, in *IEEE International Symposium on Intelligent Signal Processing*, IEEE, (2007), pg. 1.

[6] R.S. Sutherland, *Accurate free-free gaunt factors for astrophysical plasmas*, *Mon. Not. Roy. Astron. Soc.* **300** (1998) 321.

[7] S. Bozhenkov et al., *The Thomson scattering diagnostic at Wendelstein 7-X and its performance in the first operation phase*, 2017 *JINST* **12** P10004.

[8] J. Knauer et al., *A new dispersion interferometer for the stellarator Wendelstein 7-X*, in 43rd *European Physical Society Conference on Plasma Physics*, volume 4, Leuven, Belgium (2016).

[9] C.E. Rasmussen and C.K.I. Williams, *Gaussian processes for machine learning*, The MIT Press, U.S.A. (2006).

[10] M. Hirsch et al., *ECE diagnostic for the initial operation of Wendelstein 7-X*, *EPJ Web Conf.* **203** (2019) 03007.

[11] U. Hoefel et al., *Bayesian modeling of microwave radiometer calibration on the example of the Wendelstein 7-X electron cyclotron emission diagnostic*, *Rev. Sci. Instrum.* **90** (2019) 043502.

[12] F. Effenberg et al., *First demonstration of radiative power exhaust with impurity seeding in the island divertor at Wendelstein 7-X*, *Nucl. Fusion* **59** (2019) 106020.

[13] D. Zhang et al., *Design criteria of the bolometer diagnostic for steady-state operation of the W7-X stellarator*, *Rev. Sci. Instrum.* **81** (2010) 10E134.

[14] O. Ford et al., *Charge exchange recombination spectroscopy at Wendelstein 7-X*, *Rev. Sci. Instrum.* forthcoming (2019).

[15] L. Vanó et al., *Studies on carbon content and transport with charge exchange spectroscopy on W7-X*, in *EPS proceedings* forthcoming (2019).

2019 JINST 14 C10003

6.4. Article IV

A. Pavone et al.
»Neural network approximated Bayesian inference of edge electron density profiles at JET«
In: *Plasma Physics and Controlled Fusion* (2020)

Synopsis
The publication describes how neural networks can be trained to approximate Bayesian inference based on an existing Bayesian model for the reconstruction of the edge electron density profiles at the JET tokamak. We demonstrate here that the method previously developed and tested at the W7-X stellarator can be generalized to a completely new physics system, therefore hinting to the possibility, in future, to automate the procedure for approximating any physics model implemented within the same Bayesian modeling framework. The method is also extensively tested on a large number of different experimental cases, and compared to the conventional Bayesian inference results. We show also how the uncertainties of the network prediction can be calculated with an approach which relies on the deep learning technique known as dropout training and an interpretation of the training problem as a variational inference problem in the context of Bayesian neural networks.

OPEN ACCESS
IOP Publishing
Plasma Physics and Controlled Fusion
Plasma Phys. Control. Fusion **62** (2020) 045019 (13pp)
https://doi.org/10.1088/1361-6587/ab7732

Neural network approximated Bayesian inference of edge electron density profiles at JET

A Pavone[1], J Svensson[1], S Kwak[1], M Brix[2], R C Wolf[1] and JET Contributors[3]

[1] Max-Planck-Institut für Plasmaphysik, Teilinstitut Greifswald, D-17491 Greifswald, DE, Germany
[2] Culham Centre for Fusion Energy, Culham Science Centre, Abingdon OX14 3DB, United Kingdom

E-mail: andrea.pavone@ipp.mpg.de

Received 30 August 2019, revised 6 February 2020
Accepted for publication 17 February 2020
Published 5 March 2020

Abstract

A neural network (NN) has been trained on the inference of the edge electron density profiles from measurements of the JET lithium beam emission spectroscopy (Li-BES) diagnostic. The novelty of the approach resides in the fact that the network has been trained to be a fast surrogate model of an existing Bayesian model of the diagnostic implemented within the Minerva framework. Previous work showed the very first application of this method to an x-ray imaging diagnostic at the W7-X experiment, and it was argued that the method was general enough that it may be applied to different physics systems. Here, we try to show that the claim made there is valid. What makes the approach general and versatile is the common definition of different models within the same framework. The network is tested on data measured during several different pulses and the predictions compared to the results obtained with the full model Bayesian inference. The NN analysis only requires tens of microseconds on a GPU compared to the tens of minutes long full inference. Finally, in relation to what was presented in the previous work, we demonstrate an improvement in the method of calculation of the network uncertainties, achieved by using a state-of-the-art deep learning technique based on a variational inference interpretation of the network training. The advantage of this calculation resides in the fact that it relies on fewer assumptions, and no extra computation time is required besides the conventional network evaluation time. This allows estimating the uncertainties also in real time applications.

Keywords: JET, neural network, Bayesian inference, real time, dropout, Lithium beam diagnostic, edge electron density

(Some figures may appear in colour only in the online journal)

1. Introduction

The application of neural networks (NN) to fusion experiments is not new, dating back to the mid-1990s with examples at the JET experiment of reconstruction of ion temperature profiles in real-time [1] and analysis of charge exchange spectra [2, 3]. They have been used for the inference of plasma parameters from diagnostic data as well as the prediction of disruptive events from different parameters and measured quantities [4]. More recently, they have also been used at the Wendelstein 7-X experiments for the task of reconstructing magnetic configuration properties from heat load patterns on the plasma-facing components [5, 6]; at JET for tomographic reconstruction [7]; they have been used also as surrogates for transport models as shown in [8–10]. Different machine learning algorithms as Gaussian processes

[3] See the author list of Joffrin *et al* (https://doi.org/10.1088/1741-4326/ab2276) for the JET contributors.

have been used in surrogate-based optimization strategy for the accelerated validation of plasma transport codes as in [11]. NN are very desirable tools especially for two reasons: they are able to identify patterns for those phenomena where a physics model describing the process is missing, e.g. plasma disruption, and they can process data at very fast time scales, e.g. in the order of tens of microseconds. The latter feature is particularly relevant today as fusion experiments produce more data than we can hope to exhaustively analyze with traditional tools.

Here we train a NN as a fast approximation, i.e. a *surrogate* model, of a Bayesian model of the JET lithium beam spectroscopy (Li-BES) diagnostic for the inference of the edge electron density profiles from experimental measurements. The principles behind the functioning of the diagnostic are given in [12], whereas details about the experimental configuration and measurements at JET can be found in [13–15]. Edge electron density profiles are useful quantities in controlling and understanding plasma phenomena as edge localized modes (ELMs), L-H transitions and turbulence transport. A model for the diagnostic, described in detail in [16], is implemented within the Minerva Bayesian modeling framework [17]. The framework provides a common way to define models and perform Bayesian inference when measurements are available. The models are strongly modular so that different modules, or *nodes* in the jargon, can be easily used to build similar models for different systems, e.g. diagnostics at different fusion machines, or test different assumptions. Currently, the framework is extensively used at the fusion experiment JET, where its application is discussed in [18] and an application to the equilibrium reconstruction using microwave diagnostics is described in [19], and W7-X, where it has been used to model a microwave radiometer calibration for the electron cyclotron emission diagnostic [20], for the inference of electron, ion, and impurity density profiles from an x-ray imaging diagnostic [21], and for the inference of ion temperature from measurements of a coherent thomson scattering diagnostic [22].

In a previous work [23], it was shown that a NN can be trained as approximation of the Bayesian model of an x-ray imaging diagnostic at W7-X, and it was argued that the same method could be easily applied to a different system for which a Bayesian model implemented within Minerva was available. Extending such work, here we aim at validating this claim. Therefore we make use of the same method for training the network, i.e. we train the network on data generated exclusively with the Bayesian model sampling from its joint distribution, and we show that it can be successfully used to approximate the full model Bayesian inference of plasma parameters from data measured with a new physical system at a different fusion experiment, the edge electron density profiles from the Lithium beam emission spectroscopy diagnostic measurements at JET. In this way, we demonstrate that all that is required to obtain such network approximation is a Bayesian model. This is relevant because it shows that it is possible to replicate the method and achieve a fast reconstruction for any diagnostic modeled within the Minerva framework. Moreover, a major novel contribution is achieved by improving on the uncertainties calculation previously reported, which suffered from being slow and requiring limiting approximations. The calculation makes use of a novel state-of-the-art deep learning technique which can provide fast and at the same time accurate uncertainty estimates. This is of particular relevance if we think of using the network reconstructions in real time systems and control applications where we need to take decisions according to the network result and it is therefore crucial to know whether and to what extent the network output is accurate and can be trusted.

In section 2 we give an overview of the Lithium beam spectroscopy diagnostic to the extent that is relevant to this work, in section 3 we describe the Bayesian model of the diagnostic implemented within the Minerva framework, in section 4 we show how the network is trained making use of data generated with the Minerva Bayesian model in order to make predictions from experimental data, in section 5 we describe how the uncertainties of the network model can be calculated, and in section 6 we compare the network inference to the Bayesian inference carried out with the Minerva model on measurements collected at several JET pulses. We draw our conclusion in 7, where we also give an outlook on future developments.

2. The JET lithium beam spectroscopy diagnostic

The Li-BES system measures the spectral emission produced by the interaction of lithium atoms with the plasma species. The lithium atoms are injected with a beam vertically from the top of the machine, and as the beam penetrates the plasma it gradually gets excited and it is lost along the magnetic field lines when most of the atoms get ionized. A transmission grating spectrometer collects the radiation emitted along the penetration path, which is limited to the edge region of the plasma where it allows the reconstruction of the electron density. The spectrum is observed in a few nanometers wavelength range from 26 different spatial positions. A CCD camera is used to detect the photons with an integration time of typically 10 ms. A sketch of the system is shown in figure 1.

In order to understand the work presented here, details about the hardware are not as relevant as those about the model, which are given below. For the reader interested in knowing further details about the hardware set-up, detailed descriptions can be found in [13] and [14]. Here we will give an overview of the diagnostic principles and the model in order to provide the information required to understand the rest of the work. A full description of the Bayesian model and its usage to infer the electron density, including details about error treatment, modeling of the instrument function and calibration, is given in [16, 24].

The measured spectra contain different components: the Li I line radiation A, a bremsstrahlung dominated background

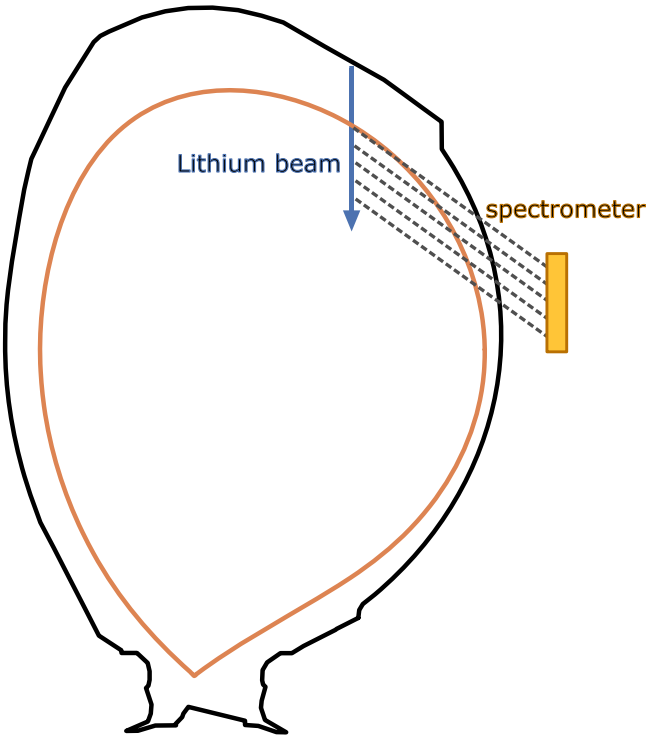


Figure 1. A schematic of the Li-BES system at JET. The lithium beam is injected vertically (blue arrow) and it penetrates the plasma volume, indicated by the orange ellipsoid, emitting light by interacting with the plasma species. The spatial positions of the measurements is indicated by the intersection of the lines of sight (dashed lines) with the lithium beam path. The light is collected by a spectrometer, in yellow in the figure.

B assumed to be constant in the wavelength range of interest, and an offset Z. The signal S can be found by taking into account an instrument function $C(\lambda)$ representing the shape of an infinitely narrow line on the detector and an interference filter function $F(\lambda)$ according to the following equation (the spectral width of the Li line is below the resolving capability of the instrument):

$$S(\lambda) = F(\lambda)[C(\lambda)A + B] + Z. \tag{2.1}$$

The quantity of interest for this study is the Li I line radiation A, which we will refer to as the measurement or observation of our system from now on. It is inferred from the measured signal S in a pre-processing stage, prior to any NN or Bayesian model evaluation, by first inferring the interference filter function F and the instrument function C from two independent and dedicated measurements without plasma, and then by inferring A, B and Z simultaneously from actual plasma experiments. We will not give further details here about how this is accomplished as it is not relevant for the rest of this work; the interested reader can find more information in [16].

The intensities of the Li I (2p-2s) line radiation come from the neutral lithium beam atoms injected into the vacuum vessel as they traverse and interact with the plasma. The atoms penetrating into the plasma undergo collisions with the plasma electrons, protons and other impurities and by mean of spontaneous emission processes they produce the line radiation that is collected by the diagnostic. The line radiation is emitted by the decay from the first excited state ($1s^2\ 2p^1$) to the ground state ($1s^2\ 2s^1$) of the beam atoms. The line intensity is then dependent on the population of the first excited stated. The change in the relative population of any excited state N_i as the beam atoms penetrate the plasma can be expressed in terms of the plasma electron density $n_e(z)$ and temperature $T_e(z)$ according to a multi-state collisional-radiative model firstly introduced in [25]:

$$\frac{dN_i(z)}{dz} = \frac{1}{v_{Li}} \sum_{j=1}^{M_{Li}} [n_e a_{ij}^e(T_e) + n_p a_{ij}^p(v_{Li}) + b_{ij}] N_j, \tag{2.2}$$

where z represents a coordinate along the penetration length of the beam. The coefficients a_{ij}^e and a_{ij}^p with $(i \neq j)$ and $a > 0$ are net population rate coefficients accounting for the contribution of plasma electrons and ions in populating the ith state from the jth state; whereas $a_{ii} < 0$ denotes a net de-population rate coefficient of the ith state accounting for excitation, de-excitation and ionization processes. The coefficients b_{ij} represent instead spontaneous emission rate coefficients or Einstein coefficients. v_{Li} is the lithium beam velocity corresponding to $\approx$50 keV beam energy, n_p is the density of plasma protons, and M_{Li} is the number of considered states of the neutral lithium atoms, which is 9 in this case. The dependency of the plasma profiles n_e, T_e and N_j on the z coordinate has been omitted for brevity. In order to be able to solve equation (2.2), an initial condition needs to be defined. It can be chosen to be:

$$N_i(z = 0) = \delta_{1i} \tag{2.3}$$

corresponding to the assumption that all lithium beam atoms are neutral in the ground state ($i = 1$) at $z = 0$, the position where they enter the vacuum vessel. In other words we assume $N_1(z = 0) = 1$. The population of the first excited state ($i = 2$) of the lithium atoms N_2 can then be calculated. This quantity is proportional to the observed lithium intensities $A(z)$ found from the signal measured with the CCD camera along the observation length. We therefore introduce a calibration factor α to express this relationship:

$$A(z) = \alpha N_2(z). \tag{2.4}$$

The factor is not known and it has to be inferred from the data. For the interested reader, a complete derivation and an explicit expression of α in terms of the CCD output counts can be found in [16].

Figure 2 shows an example calculation carried out with the forward model implementing the physics described so far. Given the plasma profiles in the two plots on the top, the relative population of the first excited state of the lithium atoms and then the Li I line intensity can be calculated. The plot on the bottom left representing the line intensity in arbitrary units also shows a 10% relative Gaussian noise added to the calculation (the scattered circles) in order to simulate the noise present in the measurements. As the beam

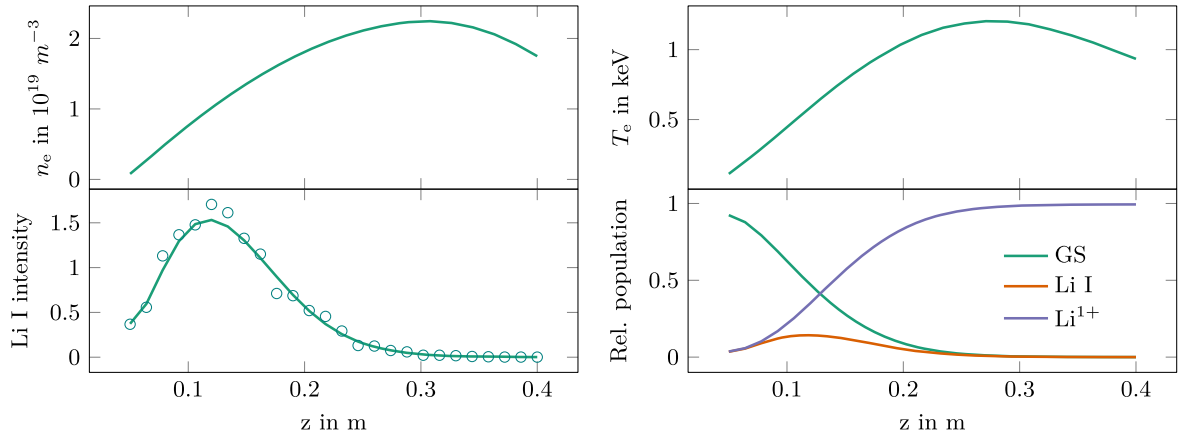


Figure 2. An example case of the Li-BES forward model calculation. In clockwise direction, from the top left plot the following quantities are shown: an electron density profile, an electron temperature profile, the Li I line intensity predicted with the forward model (solid line) together with the addition of 10% relative Gaussian noise (scattered dots), and the relative population of the ground state (GS), the first excited state Li I and the first ionized state Li^{1+} of the beam atoms. All quantities are expressed as function of the penetration distance inside the plasma, with $z = 0$ corresponding to the position where the beam enters the vacuum vessel.

atoms penetrate into the plasma they also get ionized and when this happens they follow the magnetic field lines as charged particles and do not contribute any longer to the collected emission. The ionized atom population is shown in violet in the plot on the bottom right.

The measured intensity can be used to infer the electron density profile at different edge locations along the penetration length, provided the electron temperature profile information. The latter is usually delivered at JET by the high resolution Thomson scattering diagnostic (HRTS) [26].

3. The Bayesian minerva model

The multi-state model described in section 2 is implemented within the Minerva Bayesian modeling framework. The Minerva modeling framework [17] is a framework that allows modeling complex systems and carrying out Bayesian inference with them. Models are expressed in a modular way, where the modules are called *nodes*. These modules can be easily switched and replaced so that different models can be easily built, and different assumptions can be easily tested. Nodes can represent physics quantities with associated probability distributions over the values they can assume, or they can represent deterministic calculations consumed by other nodes in the model. The models are used as forward models to predict observations from given free parameters. It makes use of graphical models [27] to represent models and the probabilistic relations between quantities in the model. An example of a Minerva graph for the lithium beam system is shown in figure 3, and it is described later in the section. Once a model has been defined within the framework, Bayesian inference can be performed with it. Thanks to the fact that model definition and Bayesian inference constitute two different and independent stages, such that the implementation details of one are abstracted away from the other, the framework offers a solution for performing scientific inference in complex systems which is general, and not strictly related to a single nuclear fusion

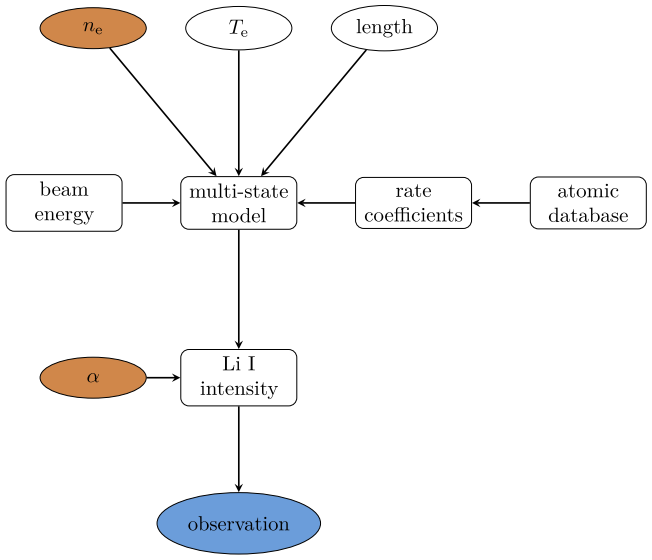


Figure 3. A simplified sketch of the Li-BES Minerva model graph. Colored nodes are probabilistic nodes, where *orange* denotes the free parameters and *blue* denotes the observed quantities. White nodes represents deterministic calculation nodes or other input parameters required by the model. The arrows represent direct or indirect dependencies in the probabilistic relations between the quantities in the probabilistic nodes.

experiment or even nuclear fusion research. As a Bayesian framework, it employs Bayesian probability theory to handle the uncertainties attributed to any modeled quantity. In Bayesian probability a prior distribution $p(T)$ is assigned to the model free parameters T and a likelihood function $p(D|T)$ is assigned to the model observations D. As measurements are available, they can be used to update the prior knowledge on the free parameters through Bayes formula:

$$p(T|D) = \frac{p(D|T)p(T)}{p(D)}. \tag{3.1}$$

The quantity $p(T|D)$ is called the posterior distribution and it reflects the new state of knowledge on the parameters T as the

observations D are taken into account. The numerator of the equation is also known as the joint distribution $p(D, T)$ of the observations and parameters. The denominator $p(D)$ is a normalization factor and it is referred to as the evidence.

3.1. Model parameters

The model free parameters are the electron density profile n_e and the absolute calibration factor α. The prior distribution for the n_e profile is modeled through a zero mean Gaussian process [28]. A Gaussian process is a stochastic process whose realizations are functions. In Bayesian inference they are used for models where the free parameters are functions, in this case 1D electron density profiles, and the observations are the values they assume in a number of domain locations. A realization of a random function drawn from the process is given by the values it assumes in a number of positions in its domain and its probability distribution is chosen to be Gaussian. Its covariance is known as *covariance function*. One common choice for it is the squared exponential, which regulates the smoothness of the function by modeling the correlation between points in the domain. For the density profiles it can be written as:

$$K(z_1, z_2) = \sigma_f^2 \exp\left(-\frac{(z_1 - z_2)^2}{2\sigma_x^2}\right) + \delta_{ij}\sigma_y^2, \tag{3.2}$$

where z_1 and z_2 are two positions along the z axis and the different σ parameters regulate the smoothness of the profile. σ_f regulates the overall variance of the profile and σ_x regulates the length scale of the changes in the profile. Small values mean that the profile can change quickly along z, whereas large values mean that it will change slowly. σ_y is used to allow for small amount of noise expected in the profile. A uniform distribution is used for the calibration factor α. The model observations are the Li I line intensities. The likelihood function is chosen to be a normal distribution centered on the forward model prediction.

3.2. Model graph

A sketch of the Minerva model graph for the Li-BES system is shown in figure 3. In the sketch, the nodes representing the free parameters n_e and α are in orange, and the node representing the observations is depicted in blue. The white nodes represent computation nodes, as the 'multi-state model' node which is used to calculate the predicted Li I line intensity, represented in the 'Li I intensity' node, or other quantities required by the model, as the energy of the lithium beam, represented by the 'beam energy' node, and the observation length, defined as the length along the beam path where the emission is observed, represented by the 'length' node. The observation length is a quantity that is known given the experimental setup and it can be different for different experiments. We make use of 20 and 26 equally spaced positions along the observation length for the profile and the observations locations, respectively. The calibration coefficient α is applied to the predicted Li I line intensities as an overall multiplicative factor. In the graph, we have also shown the dependency of the multi-state model from the rate coefficients that are taken from the Atomic Data and Analysis Structure (ADAS) database [29], a database containing data useful for modeling the radiating properties of ions and atoms in plasmas. The arrows represent direct or indirect dependencies in the probabilistic relations between the quantities in the probabilistic nodes, and should not be understood as a computational flow. All free parameters node reach, indirectly, the observation node and are not connected to each other. This expresses the fact that the joint distribution of the graph $p(D, T)$ can be factorized in terms of a conditional distribution of the observations conditioned on the free parameters $p(D|n_e, \alpha)$ and the product of two independent prior distributions over the electron density $p(n_e)$ and the calibration factor $p(\alpha)$:

$$p(D, T) = p(D|n_e, \alpha)p(n_e)p(\alpha). \tag{3.3}$$

4. NN training

Given the Bayesian model described in the previous section, we aim now at training a NN in such a way that it constitutes an approximation of the Bayesian inference that can be carried out with the full Minerva model. In order to achieve this, we use the Minerva model to generate the training data. In this section, we outline the procedure to the extent it concerns the specific case of the lithium beam system under investigation. For the interested reader, further conceptual and theoretical insights about how this method can provide a sound approximation are given in [23].

4.1. Generation of the training data

The electron temperature profile T_e and the observation length l are parameters that are known at inference time, when we perform inference with the Minerva model and the network: the former is provided by an independent measurement of the Thomson scattering diagnostic, the latter comes from the experimental setup. Both quantities constitute part of the network input, together with the measured lithium line intensities, and therefore need be generated with the Minerva model for training the network. As we aim at training the NN on the problem of inferring electron density profiles from measured Li I line intensities, the training input data are the Li I line intensities, the T_e profiles, and the length, while the training output data are the n_e profiles and the absolute calibration coefficient α. We generate the training data by sampling from the joint distribution of the model. This means that, as a first step, we draw a sample of n_e, T_e, l and α from the corresponding prior distributions and, given these values, we compute the predicted Li I line intensities and draw a sample from the likelihood function. We iterate over this process a number of times equal to the number of samples in the training set. As we need to generate data also for T_e and the observation length, we assign probability distributions also to them, so that the joint distribution of the model can be

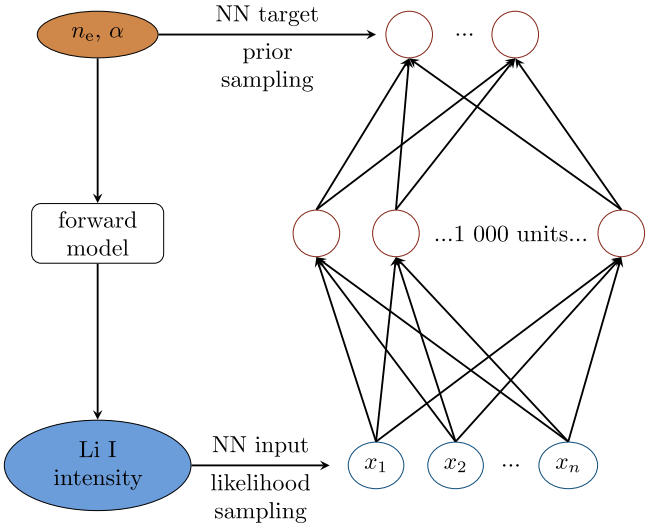


Figure 4. A sketch to illustrate the sampling procedure for the training set creation. A sketch of the Li-BES Minerva model and the neural network, having one hidden layer with 1000 units, is shown on the left and on the right, respectively. At training time, the NN takes as input the Li I line intensities generated with the Minerva model and sampled from the likelihood function together with the sampled T_e and observation length l. The sampled n_e and α used to generate the intensities are the target data of the network. The blue nodes of the neural network denote the input intensities and the two red nodes at the top denote the output points of the electron density profile.

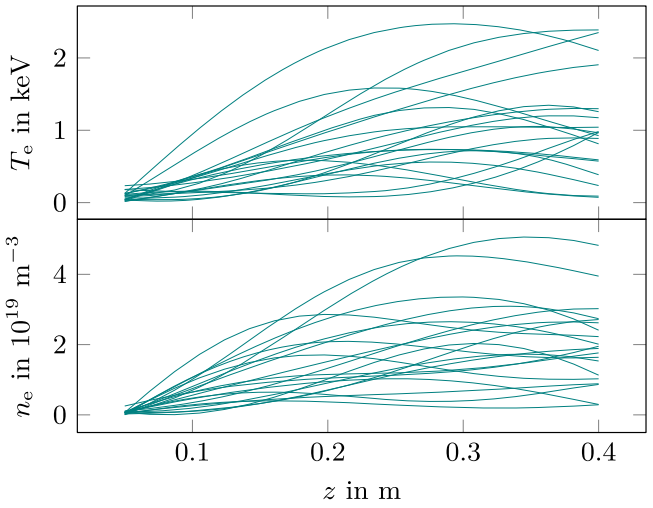


Figure 5. Samples from the Gaussian process priors for the T_e and n_e profiles, top and bottom figures, respectively. The x-axis position at 0.0 corresponds to the location where the beam atoms enter the plasma, which is at the edge of the machine. The low value constraint at such position is also visible in the shape of the sampled profiles.

written as:

$$p(D, T) = p(D, n_e, T_e, l, \alpha) \tag{4.1}$$

and

$$p(D, T) = p(D|n_e, T_e, l, \alpha)p(n_e)p(T_e)p(l)p(\alpha). \tag{4.2}$$

A sketch of the procedure is shown in figure 4.

We used for the n_e and T_e profiles a zero mean GP prior defined on a x domain of 20 linearly spaced positions between $x_0 = 0.0$ and $x_1 = 20.0$ with covariance function as in equation (3.2). The parameters of the GP are set to: $\sigma_x = 10.0$, $\sigma_y = 0.002 \times 10^{19}$ m^{-3}, $\sigma_f = 2.0 \times 10^{19}$ m^{-3} for n_e, and $\sigma_x = 10.0$, $\sigma_y = 0.002$ keV, $\sigma_f = 1.0$ keV for T_e. The profiles are constrained to be non-negative by rejecting the samples having negative values as they are drawn from the GP prior distributions until a positive valued sample is drawn and kept. Moreover, we constrain the profiles to assume low value at $x = 0$ corresponding to the position $z = 0$, the edge location where the beam enter the plasma. The constraint is implemented as a *virtual observation*, i.e. by implementing an observed node in the Minerva graph as a normal distribution with standard deviation 100 eV around a value of 100 eV for the T_e profile, and standard deviation 0.1×10^{19} m^{-3} around a value of 0.01×10^{19} m^{-3} for the n_e profiles. In this way, when the profiles are sampled, they are constrained by this virtual observation as if it was a real measurement, although no measurement of such kind actually occurred. Further details about how a virtual observation constraint is implemented are provided extensively in [23] and will not be treated further here, as they are not relevant to the understanding of the work that follows.

Samples from the n_e and T_e prior distributions are shown in figure 5. The distance from the location $z = 0$ at which the beam atoms enter the plasma is on the x axis. It is worth noticing that the profiles are not monotonic. For the calibration factor α we use a uniform distribution between 1.0 and 20.0. The choice for this prior was motivated by the information available from previous analysis, which showed values typically falling in this range. For the parameter l we use a uniform distribution between 0.2 and 0.4 cm. Finally, the conditional distribution of the simulated Li I intensity $P(D|T)$ is a normal distribution centered on the model prediction and with standard deviation equals to 10% relative error. In this way, we inject noise in the training input data, as we expect to have noise at evaluation time, when the input are the experimental measurements. Our training data set is made of 100 000 samples.

4.2. Network model

The NN architecture used for this problem is a multilayer perceptron (MLP) with one hidden layer with 1000 units. The activation function used in the hidden units is the so called scaled exponential linear function (SELU) [30] and the loss function used is the mean squared error:

$$L(\mathbf{w}) = \frac{1}{N}\sum_i (\mathbf{y}_i(\mathbf{w}) - \mathbf{t}_i)^2, \tag{4.3}$$

where N is the number of training samples, $\mathbf{w}$ is the vector of adaptable network weights, and $\mathbf{y}_i$ and $\mathbf{t}_i$ are the ith multi-dimensional output and target vector, respectively. The network was trained using the Adam optimizer with parameters: learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, see [31] for a description of the algorithm and parameters. The training data were divided in batches of 100 samples and the network weight training was terminated once 5000 passes

through the training set were reached (also called epochs). One single starting position was used for the initialization of the weights. The number of 1000 hidden units in one hidden layer was chosen by validating the network performance on a set of test data made of 1000 samples drawn from the joint distribution of the Bayesian model. The network was implemented within the TensorFlow framework [32]. The NN has been trained with dropout [33, 34]. Dropout is a technique originally introduced to prevent overfitting. Although this can be, by itself, a good reason to make use of it, there is at least another reason. Dropout training can also be used to estimate uncertainties in the network prediction; when used in this way it is referred to as *Monte Carlo (MC) dropout* [35]. In the next section we will give an overview of the theoretical framework that allows to interpret dropout training as a Bayesian inference technique. We will only touch the salient points of the derivation which are necessary to understand the current work, but for the reader interested in a deeper understanding of the theory behind it, details can be found in [35].

Before proceeding, we would like to summarize the relationship between the two key elements of this work:

- the *Minerva Bayesian model* is defined at the first step, and it is used to both carry out the full Bayesian inference of the electron density profiles from the measured experimental data, *and* to generate the training data for the NN from its joint distribution.
- the NN is first trained on data generated exclusively with the Minerva Bayesian model, afterwards it is applied to infer electron density profiles from the measured experimental data.

In this way, the full Bayesian inference and the NN inference are both based on the same Bayesian model, with the distinction that the latter approximates the former. The two inference methods will be compared in section 6.

5. NN uncertainties

Delivering uncertainties in the NN calculation is necessary in order to asses whether, and how far, the network prediction can be trusted. This is important when the network output is wanted for further calculations, and especially when a decision has to be taken according to its output, as in the case of real time control systems, e.g. feedback systems. Therefore, it is also important that the uncertainties can be calculated in a time scale comparable to the network processing speed itself. Here we give an overview of the theoretically sound and practically desirable method presented in [35].

5.1. Bayesian NNs

NN uncertainties can be calculated in a Bayesian framework known as Bayesian NNs [36]. In this context, the network training is seen as an inference problem, where the free parameters are the network weights $\mathbf{w}$ and the training target data are the observations $\mathbf{Y}$. It follows that we can write Bayes formula for the posterior of the network weights:

$$p(\mathbf{w}|\mathbf{Y}, \mathbf{X}) = \frac{p(\mathbf{Y}|\mathbf{w}, \mathbf{X})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}, \tag{5.1}$$

where $\mathbf{X}$ denotes the training input data. As we have now a distribution over the network weights, we will also have a distribution over the network's predictions $\mathbf{y}^*$ for a new input vector $\mathbf{x}^*$, given by:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{w}, \mathbf{x}^*)p(\mathbf{w}|\mathbf{Y}, \mathbf{X})\mathrm{d}\mathbf{w}. \tag{5.2}$$

This distribution is the one we are interested in and which prescribes the uncertainties in the network prediction.

5.2. Variational inference

For any interesting NN model, the posterior $p(\mathbf{w}|\mathbf{Y}, \mathbf{X})$ cannot be treated analytically because of the large number of weights and complex network function. We therefore make use of *variational inference* (VI) [37] in order to approximate it. In VI we choose an approximating variational distribution $q_\theta(\mathbf{w})$ parametrised by θ, which is easy to evaluate, in order to approximate the original posterior distribution. This is achieved by minimizing the Kullback–Leibler (KL) divergence with respect to θ, which can be thought as a measure of similarity between two distributions:

$$\mathrm{KL}(q_\theta(\mathbf{w})||\, p(\mathbf{w}|\mathbf{Y}, \mathbf{X})) = \int q_\theta(\mathbf{w})\log\frac{q_\theta(\mathbf{w})}{p(\mathbf{w}|\mathbf{Y}, \mathbf{X})}\mathrm{d}\mathbf{w}.$$

It can be shown that minimizing the KL divergence is equivalent to maximizing the so called *evidence lower bound* (ELBO) with respect to θ:

$$L_{\mathrm{VI}}(\theta) = \int_{\mathbf{w}} q_\theta(\mathbf{w})\log p(\mathbf{Y}|\mathbf{X}, \mathbf{w})d\mathbf{w} - \mathrm{KL}(q_\theta(\mathbf{w}) \,||\, p(\mathbf{w})),$$

where, noticeably, the KL divergence term now is between the approximating distribution $q_\theta(\mathbf{w})$ and the prior distribution $p(\mathbf{w})$, fact that explains the name of the expression. At this point we make use of the results derived in [35], where it is shown that the conventional dropout training of a NN is equivalent to the maximization of the ELBO function.

5.3. Dropout

When a network is trained with conventional dropout, at each iteration of the training, as a new training batch sample is provided to the network, some of its units are dropped. This makes the trained network more flexible, intuitively because the units need to learn to be useful also when some of the others are missing. To be more rigorous, dropout prevents overfitting by preventing co-adaption of the units. At evaluation time all units are retained, but their output is scaled down by the probability of dropping them, since now there is a larger number of units in the network.

5.4. MC dropout

In the conventional dropout picture of training, the stochastic process is applied in the unit (or feature) space. We can switch view and see the stochastic process as applied in the

weight space, since as units are dropped, also the corresponding weights that connect them are dropped. Under this view, it is finally possible to merge the dropout training with the VI approximation of the true weight posterior. This happens by re-parametrising the weights $\mathbf{w}$ in terms of a function g such that:

$$\mathbf{w} = g(\theta, \epsilon) = \{\mathrm{diag}(\epsilon)\mathbf{M}, \mathbf{b}\}, \tag{5.3}$$

where $\theta = \mathbf{M}$, $\mathbf{b}$ and $\epsilon \sim \mathrm{Bernoulli}(p_\mathrm{b})$ where p_b is the probability of dropping the units. ϵ is then a vector of zeros and ones, $\mathbf{M}$ is a Q by D deterministic matrix of connecting weights where Q is input vector size and D output vector size, $\mathbf{b}$ is the bias vector of dimension Q, and $\mathrm{diag}(\epsilon)$ is a diagonal matrix of same size as $\mathbf{M}$ having as diagonal elements the elements of the vector ϵ. The product $\mathrm{diag}(\epsilon)\mathbf{M}$ represents a matrix multiplication whose results end up 'selecting' what connecting weight is active at a given dropout step. At this point, after some more manipulations, we can rewrite the integral in the ELBO expression as an integral over $p_\mathrm{b}(\epsilon)$ and the derivatives required for the optimization as derivatives with respect to θ. In [35] it is then shown that, optimizing a NN dropout loss function is equivalent to optimizing the function $L_\mathrm{VI}(\theta)$. In conclusion, this means that, by using a well established method for training the network, we can at the same time approximate the posterior distribution of the corresponding Bayesian network via variational inference with Bernoulli approximating variational distribution.

The only difference with the standard dropout training is that at evaluation time, instead of retaining all units, we keep dropping them as several forward passes of the network are done, so to obtain a distribution of network predictions rather than a single best estimate. This corresponds to estimating the ELBO integral with a Monte Carlo integration. The major advantage of this approach is that it scales well with large networks: forward passes of the network are typically very fast and can also be run in parallel. Therefore, calculating uncertainties in this way does not require substantial extra computation time.

We used dropout probability $p_\mathrm{b} = 0.5$ for all units in the hidden layer, and $p_\mathrm{b} = 0.0$ for the input units, i.e. all input units were retained.

We have described how variational inference and dropout can be combined in a unified view of the network training, leading to a Bayesian NN interpretation. One must be aware, though, of some caveats that have been acknowledged regarding the theoretical framework supporting this technique: see for example [38], where it is claimed that in the case of simple linear networks, this method approximates the *risk* of a process rather than the *uncertainty* of the model because the variance found in this way do not vanish at the limit of very large amount of training data; see also [39], where it is shown that the variational inference framework described in [35], specifically with regards to the choice of some approximating distributions, can lead to undefined objective function of the network, and they propose an alternative to such objective; in general, some difficulties have been recognized in the application of standard variational inference approach, as indicated in [40], where pitfalls are found in the usage of the KL divergence, and a different distance is proposed.

In the next section we will show results obtained with MC dropout estimation of the uncertainties, as we tested the network on experimental data collected at the JET tokamak.

6. Results

We evaluated the NN on data collected at several JET pulses. In order to assess the quality of the network reconstruction we can compare the reconstructed electron density profiles to those inferred with the full Bayesian model. Also, we can use the reconstructed n_e profiles as input to the forward model and simulate Li I line intensities to compare with the measured ones. This is indeed a better way to assess the quality of the network reconstruction as we can see how well the NN prediction fits the data. In the same way, the full Bayesian inference reconstruction can be compared against the measurements and the quality of the fit compared to that obtained with the NN reconstruction. We want to point out that this kind of comparison is possible because we have a model for the measurement processes, and it is the same one used for generating the network training data and the full Bayesian inference. As we previously mentioned, we are comparing two inversion methods applied to the same Bayesian model: the network inversion being a fast approximation of the full Bayesian inference.

6.1. Uncertainties

One illustrative example of such comparison is shown in figures 6 and 7 for data collected at the JET pulse 89312 at time 48.295 s, just before NBI heating started, so the plasma was in L-mode and the line integrated density was $\approx 5 \times 10^{19}\,\mathrm{m}^{-2}$. In figure 6, the NN reconstructed density profiles are compared to those inferred with the full Bayesian inference (Minerva). In figure 7, the Li I line intensities generated with the Minerva and NN reconstructed profiles are compared to the measured ones. The multiple samples represent the uncertainties. In the NN case, these are 100 samples obtained with MC dropout; in the Minerva case, these are 100 samples drawn from the full model posterior distribution which has been explored with a Markov Chain Monte Carlo sampler. From figure 6 it is evident that the uncertainties of the density profiles inferred with the network and with Minerva can be quite different. This should not surprise. It is important to realize that the uncertainties stemming from the two methods arise from two different models, the network and Minerva model, and the corresponding Bayesian inference problems. In both cases, the uncertainties are calculated in a Bayesian framework, but the models and quantities that contribute to the uncertainties in the reconstructed profiles are different, as the inference task to be solved is different. This is made evident by looking at Bayes formula and the mathematical expression of the uncertainties for the two models. In the network case, the distribution of the predicted profiles is obtained by

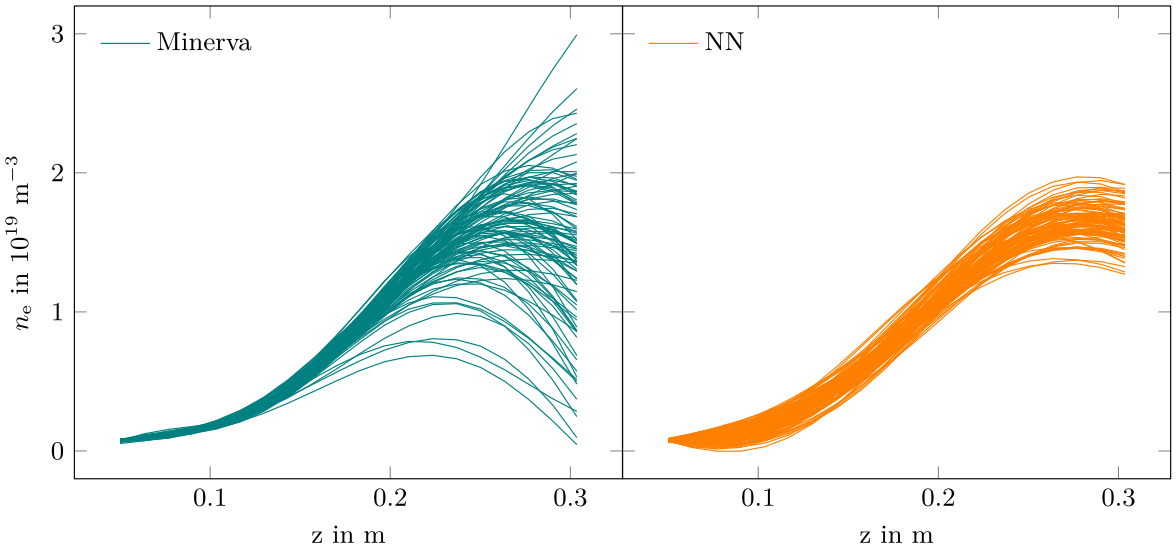


Figure 6. A comparison between the n_e profiles predicted with the NN and the full model Bayesian inference (Minerva). The samples represent the uncertainties from the MC dropout in the NN case, and the posterior distribution in the Minerva case. The data are taken from the JET pulse number 89312 at time 48.295 s.

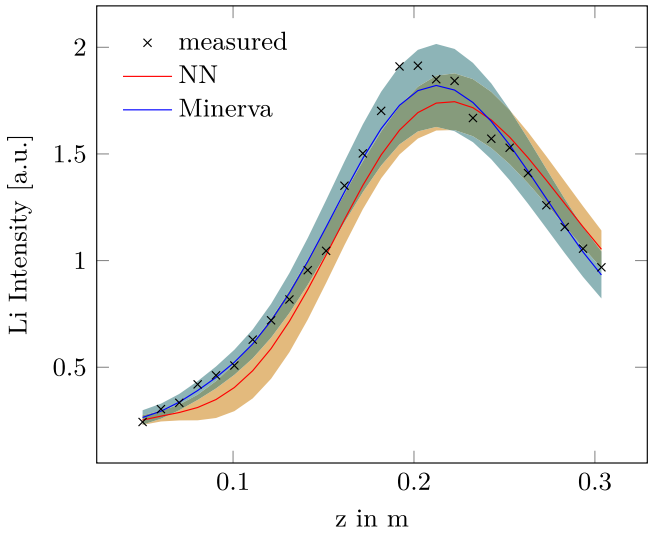


Figure 7. The Li I line intensities predicted with the NN and Minerva n_e profiles are compared to the measurements. The shadowed areas represent the uncertainties. The data are taken from the JET pulse number 89312 at time 48.295 s.

marginalization over the network weights $\mathbf{w}$ when a new input vector $\mathbf{x}^*$ is provided:

$$p(n_e|\mathbf{x}^*, \mathbf{Y}) = \int p(n_e|\mathbf{w}, \mathbf{x}^*)p(\mathbf{w}|\mathbf{Y})\mathrm{d}\mathbf{w} \tag{6.1}$$

which is the same expression of equation (5.2), in which we have omitted the dependence on the input variable $\mathbf{X}$ and substituted $\mathbf{y}^* = n_e$. When the network is evaluated on the measured line intensities, the input vector $\mathbf{x}^*$ is constituted of the electron temperature profile independently measured by a Thomson scattering diagnostic, the observation length used at that experiment, and the measured line intensities. The posterior of the network weights, instead, is given by $p(\mathbf{w}|\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{w})p(\mathbf{w})$ and it is found with variational inference with dropout training as described in section 5. We do not expect dropout training to reconstruct the posterior distribution of the Minerva model $p(n_e|D)$, but to approximate the true posterior distribution of the network weights $p(\mathbf{w}|\mathbf{Y})$; then, the spread of this posterior gives rise to a spread in the predicted profiles according to equation (6.1). Whereas, in the Minerva case the distribution of the inferred profiles is given by the posterior:

$$p(n_e|D) \propto p(D|n_e)p(n_e), \tag{6.2}$$

where D represents the measured Li I line intensities. The spread of the posterior, therefore, is influenced by the model uncertainties in predicting the measured Lithium 1 line intensity $p(D|n_e)$ (e.g. measurement errors) and the prior $p(n_e)$.

To highlight the difference between the two models, it is useful to notice what is the role of the different quantities in each of them: in the Minerva model, the free parameters are the electron density profiles, and the observations are the lithium line intensities. The inference task is then to find the electron density profiles which allow to predict the measured Li line intensities, given the measurements, the physics model, and the prior. These are the boundaries of the inference problem. The final posterior distribution expresses the uncertainties in the inference of the density profiles given the model and these boundary conditions. The uncertainties that arise in this case are related to the model uncertainties in the prediction of the Li intensities—typically estimated from the measurement errors, the sensitivity of the model to different values of the electron density, and the beam attenuation. For example, because the beam is attenuated as it penetrates the plasma and gets ionized, the model is less sensitive to changes in the electron densities in the locations closer to the core of the machine, and the uncertainties are therefore larger. Quantitative details about the estimation of the error from the measurements, and quantitative considerations on the beam

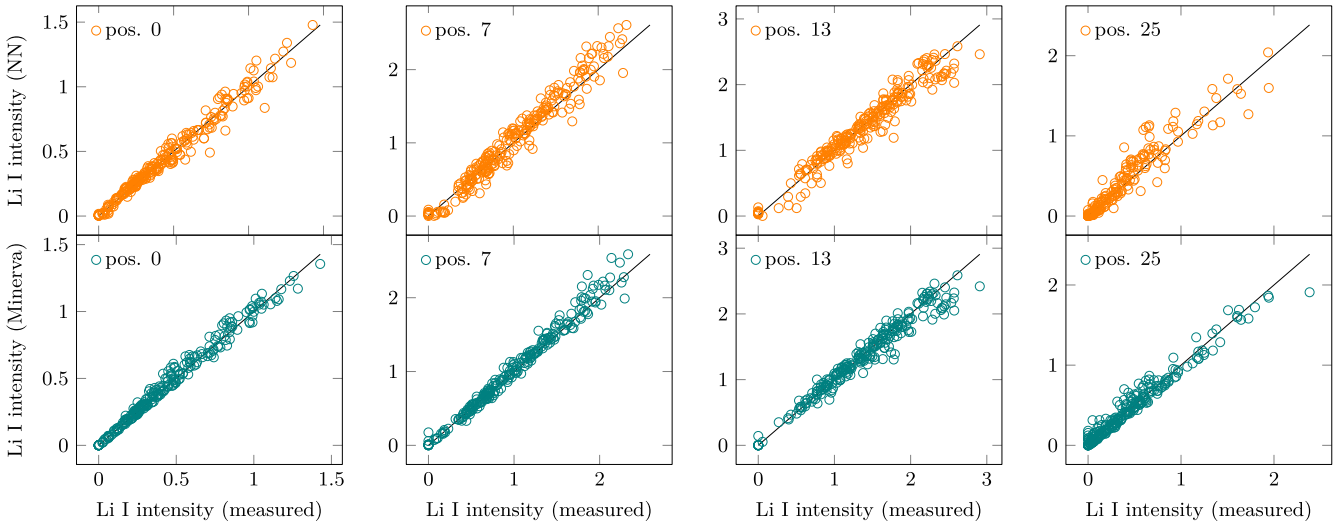


Figure 8. The Li I line intensity predicted with electron density profiles found by the network (top row) and the full model Bayesian inference (Minerva, bottom row), on the y-axis, are compared to the measurements, on the x-axis. Each column shows the comparison for a spatial position along the intensity profile. The real space coordinates of the positions can vary through the experiments, so here they are labeled by an index starting from the outermost position at index 0 to the innermost position at index 25. The solid line shows the $y = x$ line. More than 200 hundred measured data points collected across 65 pulses were used.

attenuation and sensitivity on the model are reported in previous works [16], and are not discussed here as they fall beyond the scope of this work. In the network model, the free parameters are the network weights, which lack any physics interpretation, and the observations are the set of target data in the training set, i.e. the sampled electron density profiles. The uncertainties of the network posterior depend on a combination of network structure, weight prior and approximating distribution, as it is indicated and further discussed in [35]. At training time, the network model inference task is to find the weights which allow to reconstruct the electron density profiles from the Li I line intensities, given a specific choice of network structure, weight prior, approximating distribution and training set, whose statistical properties are inherited from the Minerva model by sampling from its joint distribution. These are the boundary conditions of the inference problem for the network. The predictive distribution of equation (6.1), then, expresses the uncertainties of the model in making a prediction within these boundaries.

6.2. Li I line intensity reconstruction

The performance of the two methods is compared more extensively in figure 8, where the Li I line intensities predicted with electron density profiles found by the network (top row) and the full model Bayesian inference (Minerva, bottom row) are compared to the measurements in a scatter plot. The profiles used are the average of the MC dropout samples in the network case and the posterior distribution samples in the Minerva case. The solid line shows the $y = x$ line, where all points would lie if we had a perfect fit to the measurements. Each plot in a column shows a different spatial position along the intensity profile; since the corresponding real space coordinates may vary throughout the experiments, the positions are labeled according to an index ranging from 0 for the outermost location to 25 for the innermost one. More than 200 hundred measured data points collected across 65 pulses were considered in the analysis (see appendix for a list of the pulses). The pulses were arbitrarily chosen, without selecting for a specific set of features or plasma configurations. The pulses featured a broad range of parameters, including both L- and H-mode scenarios, low and high power and gas levels. Across all pulses, the NBI power ranged from ≈3.0 to ≈28 MW, the vacuum toroidal magnetic field from 1.6 to 3.3 T, the total ICRH power from ≈2.0 to ≈6.0 MW, the plasma current from ≈1.1 to ≈3.5 MA, and the line integrated density from $\approx 8.0 \times 10^{19}\,\mathrm{m}^{-2}$ to $\approx 2.6 \times 10^{20}\,\mathrm{m}^{-2}$. The agreement to the measurements is, in general, satisfactory for both methods. Although the network consists of a quick, approximated inversion of the full Bayesian inference, its reconstructions appear to be good enough to closely predict the data in most cases.

This is confirmed by figure 9, where we compare the mean relative error between the observations calculated with each of the two method inverted profiles and the measurements, for each position along the profile intensities:

$$E_{\mathrm{mre}} = \frac{1}{N}\sum_i \left| \frac{q_{1i} - q_{2i}}{q_{2i}} \right|, \tag{6.3}$$

where q_{1i} is the line intensity predicted by one of the methods, q_{2i} are the measured line intensities and N is the number of data points. The figure shows that the error for the network is consistently larger at every location, and it follows a trend similar to the full Bayesian inference case (Minerva). At most positions the error is below 20%, a reasonably good value, suggesting that the network inversion can provide a reliable approximated analysis.

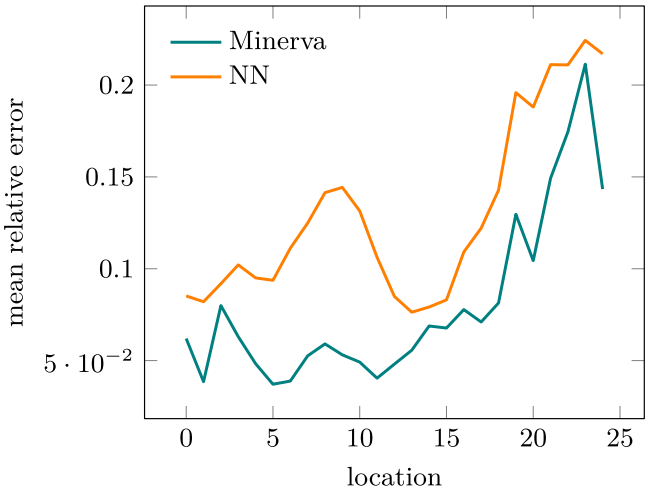


Figure 9. The mean relative error between measured Li I intensities and the intensities simulated with n_e profiles reconstructed by the network (NN) and the full Bayesian inference (Minerva) is shown at each position along the intensity profile. The calculation has been carried out for more than 200 measurements collected across 65 JET pulses.

6.3. Electron density profile inference

Finally, the electron density profiles inferred with Minerva can be compared to those found with the network as shown in figure 10. Each plot shows the n_e values at four different locations along the profile, indexed with an integer number from 0 to 19. The values are the average of the samples drawn from the posterior distribution inferred with Minerva (x-axis) and the samples found with the MC dropout network. The agreement is, in general, quite satisfactory. Indeed, an analysis of the mean relative error as defined in equation (6.3), with q_1 denoting the NN reconstructed profiles and q_2 the Minerva reconstructed profiles, shows that it is $<15\%$ at any spatial location. This can be seen in figure 11.

7. Conclusions

Extending from previous work [23], we have trained a NN as a fast, approximated Bayesian inference model for the inference of edge electron density profiles from measurements at the JET tokamak. Exploiting the NN well-known data processing speed, we can reduce the time required for the analysis from tens of minutes to tens of microseconds on a GPU, providing an approximated reconstruction. We have shown here, as it was suggested in [23], that all that is necessary in order to realize this kind of fast network approximation is the definition of a Bayesian model within the Minerva framework, since the network is trained exclusively on data generated with the model by sampling from its joint distribution. This is of particular interest because it opens the possibility to fully automate the process in order to be able to have a fast network approximation for any Bayesian model of any other diagnostic implemented within the framework.

Uncertainties can also be calculated for the network inversion. We made use of a state-of-the-art training method to approximate the network weight distribution with variational inference and calculate the uncertainties in the prediction. Compared to other existing methods, this method has the advantage of requiring essentially the same evaluation time of a standard network evaluation. It can be, therefore, particularly useful when the network is used in real time systems, which benefit of the uncertainty information when using the network prediction to make further actions or take decisions.

The network has been tested on data collected during several pulses at the JET tokamak, considering a wide range of plasma features and scenarios. A comparison of the network inferred profiles and those found with the conventional Bayesian inference shows a discrepancy in the two methods reconstructed uncertainties. This should not surprise, as they arise from two very different models with different free parameters, observed quantities, and different limitations, and therefore they are not expected to match. This discrepancy is a price that has to be paid to achieve the several orders of magnitude acceleration provided by the network. As we trained the network on a Bayesian model, we could use the same model to simulate the observations, given the network reconstructed profiles, and compare them against the measurements. We included in the comparison the full Bayesian inference reconstruction, which was carried out making use of the same model. The comparison was therefore fully consistent: the network inversion being a fast approximation of the full model one. The error in the prediction of the measurements is consistently larger when using the network predicted density profiles, as it might be expected from an approximated inversion. Still, the error is consistently below approximately 20% in all considered experimental cases, suggesting that the network inversion can be a reliable tool for fast analysis.

In future works, the NN could be used as a initial guess for the Bayesian inference carried out with the Minerva model, in this way speeding up the sampling of the posterior distribution with the MCMC by quickly providing a good starting location. The network could also be used independently, providing a fast edge profile reconstruction. For the reconstruction to be reliable, the network could be tested on a larger data set of measurements collected at previous experiments and the cases where the reconstruction fail should be investigated individually. Also, the implementation of a *novelty detection* system could be useful: this is a system which can preventively inform the user when a measurement represents an input which is unfamiliar for the network with respect to the data that had been used for training it. These cases often bring to unreliable network output and, in this way, they could be readily identified. A novelty detection method can rely on the reconstruction of the probability density of the input training data, which is then evaluated at the location of the incoming measurement input in order to asses its degree of novelty [41].

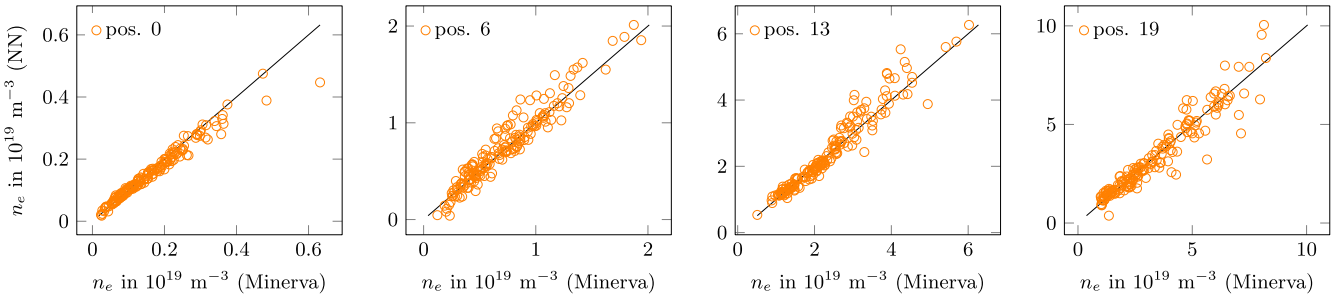


Figure 10. The electron density profile values inferred with model Bayesian inference (Minerva), on the x-axis, are compared to those inferred with the network (NN), on the y-axis. Each plot shows the comparison for a spatial position along the profile. The real space coordinates of the positions can vary through the experiments, so here they are labeled by an index starting from the outermost position at index 0 to the innermost position at index 19. The solid line shows the $y = x$ line. More than 200 hundred measured data points collected across 65 pulses were used.

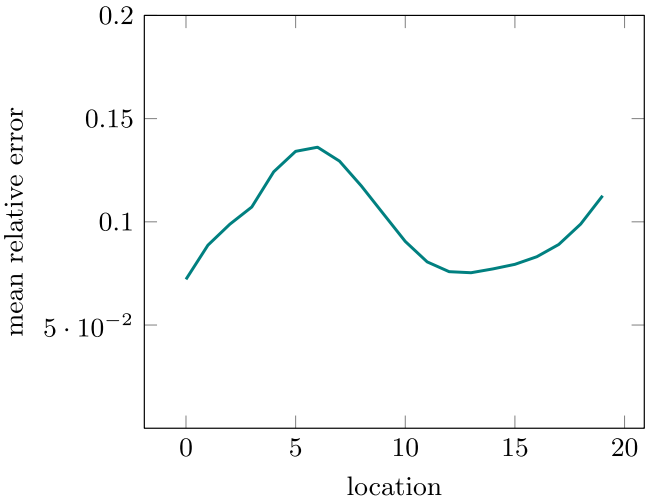


Figure 11. The mean relative error between electron density profile inferred with Minerva and with the network. The average values from the posterior samples in the Minerva case and the MC dropout sampels in the network case have been used. The calculation has been carried out for more than 200 measurements collected across 65 JET pulses.

Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014–2018 and 2019–2020 under grant agreement No. 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Appendix. List of JET pulses

What follows is a list of the JET pulses used in the analysis shown in figures 8 and 10, and discussed in section 6. The pulses were arbitrarily chosen, without selecting for a specific set of features or plasma configurations. The pulses featured a broad range of parameters, including both L- and H-mode scenarios, low and high power and gas levels. Across all pulses, the NBI power ranged from ≈3.0 to ≈28 MW, the vacuum toroidal magnetic field from 1.6 to 3.3 T, the total ICRH power from ≈2.0 to ≈6.0 MW, the plasma current from ≈1.1 to ≈3.5 MA, and the line integrated density from ≈8.0×10^{19} m^{-2} to ≈2.6×10^{20} m^{-2}.

86685, 86687, 86902, 86906, 86911, 86913, 86918, 86983, 87080, 87091, 87094, 87143, 87184, 87260, 87261, 87283, 87411, 87412, 87487, 87518, 87562, 87790, 87792, 87825, 87864, 87865, 87873, 89094, 89095, 89110, 89174, 89193, 89231, 89237, 89248, 89312, 89341, 89342, 89343, 89344, 89345, 89346, 89347, 89349, 89351, 89353, 89387, 89390, 89391, 89392, 89393, 89395, 89425, 89426, 89427, 89448, 89449, 89450, 89451, 89705, 89707, 89708, 89727, 89728.

ORCID iDs

A Pavone https://orcid.org/0000-0003-2398-966X
R C Wolf https://orcid.org/0000-0002-2606-5289

References

[1] Svensson J *et al* 1998 Real-time ion temperature profiles in the JET nuclear fusion experiment *ICANN 98. Perspectives in Neural Computing* (https://doi.org/10.1007/978-1-4471-1599-1_30)
[2] Svensson J, von Hellermann M and König R W T 1999 Analysis of JET charge exchange spectra using neural networks *Plasma Phys. Control. Fusion* **41** 315–38
[3] Bishop C M, Roach C M and von Hellermann M G 1993 Automatic analysis of JET charge exchange spectra using neural networks *Plasma Phys. Control. Fusion* **35** 765–73
[4] Cannas B, Fanni A, Sonato P and Zedda M K 2007 A prediction tool for real-time application in the disruption protection system at JET *Nucl. Fusion* **47** 1559–69
[5] Böckenhoff D *et al* 2018 Reconstruction of magnetic configurations in w7-X using artificial neural networks *Nucl. Fusion* **58** 056009
[6] Blatzheim M *et al* 2019 Neural network regression approaches to reconstruct properties of magnetic configuration from wendelstein 7-X modeled heat load patterns *Nucl. Fusion* **59** 126029
[7] Ferreira D R, Carvalho P J, Fernandes H and (JET Contributors) 2018 Full-pulse tomographic reconstruction with deep neural networks *Fusion Sci. Technol.* **74** 47–56

[8] Meneghini O *et al* 2017 Self-consistent core-pedestal transport simulations with neural network accelerated models *Nucl. Fusion* **57** 086034

[9] Meneghini O, Luna C J, Smith S P and Lao L L 2014 Modeling of transport phenomena in tokamak plasmas with neural networks *Phys. Plasmas* **21** 060702

[10] van de Plassche K L *et al* 2019 Fast modelling of turbulent transport in fusion plasmas using neural networks *Phys. Plasmas* **27** 022310

[11] Rodriguez-Fernandez P *et al* 2018 Vitals: a surrogate-based optimization framework for the accelerated validation of plasma transport codes *Fusion Sci. Technol.* **74** 65–76

[12] Pietrzyk Z A, Breger P and Summers D D R 1993 Deconvolution of electron density from lithium beam emission profiles in high edge density plasmas *Plasma Phys. Control. Fusion* **35** 1725–44

[13] Brix M *et al* 2010 Upgrade of the lithium beam diagnostic at jet *Rev. Sci. Instrum.* **81** 10D733

[14] Brix M *et al* 2012 Recent improvements of the jet lithium beam diagnostic *Rev. Sci. Instrum.* **83** 10D533

[15] Réfy D I *et al* 2018 Sub-millisecond electron density profile measurement at the jet tokamak with the fast lithium beam emission spectroscopy system *Rev. Sci. Instrum.* **89** 043509

[16] Kwak S, Svensson J, Brix J and Ghim Y-C 2017 Bayesian electron density inference from jet lithium beam emission spectra using gaussian processes *Nucl. Fusion* **57** 036017

[17] Svensson J and Werner A 2007 Large scale bayesian data analysis for nuclear fusion experiments *IEEE Int. Symp. on Intelligent Signal Processing* pp 1–6

[18] Svensson J *et al* 2011 Modelling of jet diagnostics using bayesian graphical models *Contrib. Plasma Phys.* **51** 03

[19] Schmuck S, Svensson J, Figini L and Micheletti D 2019 Towards a bayesian equilibrium reconstruction using JET's microwave diagnostics 07 *46th European Physical Society Conference on Plasma Physics (EPS 2019) (Milan, Italy)*

[20] Hoefel U *et al* 2019 Bayesian modelling of microwave radiometer calibration on the example of the wendelstein 7-x electron cyclotron emission diagnostic *Rev. Sci. Instrumen.* **90** 043502

[21] Langenberg A *et al* 2016 Forward modeling of x-ray imaging crystal spectrometers within the Minerva Bayesian analysis framework *Fusion Sci. Technol.* **69** 560–7

[22] Abramovic I *et al* 2019 Forward modeling of collective thomson scattering for wendelstein 7-x plasmas: electrostatic approximation *Rev. Sci. Instrum.* **90** 023501

[23] Pavone A *et al* 2019 Neural network approximation of bayesian models for the inference of ion and electron temperature profiles at w7-X *Plasma Phys. Control. Fusion* **61** 075012

[24] Kwak S, Svensson J, Brix M and Ghim Y-C 2016 Bayesian modelling of the emission spectrum of the joint european torus lithium beam emission spectroscopy system *Rev. Sci. Instrum.* **87** 023501

[25] Schweinzer J *et al* 1992 Reconstruction of plasma edge density profiles from li i (2s–2p) emission profiles *Plasma Phys. Control. Fusion* **34** 1173–83

[26] Pasqualotto R *et al* 2004 High resolution thomson scattering for joint european torus (jet) *Rev. Sci. Instrum.* **75** 3891–3

[27] Pearl J 1986 Fusion, propagation, and structuring in belief networks *Artif. Intell.* **29** 241–88

[28] Rasmussen C E and Williams C K I 2006 *Gaussian Processes for Machine Learning* (Cambridge, MA: MIT Press)

[29] Summers H P 2004 *The ADAS User Manual* version 2.6 (http://www.adas.ac.uk)

[30] Klambauer G, Unterthiner T, Mayr A and Hochreiter S 2017 Self-normalizing neural networks *Proceedings of the 31st International Conference on Neural Information Processing Systems (Long Beach, CA)* pp 971–81

[31] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization arXiv:1412.6980

[32] Abadi M *et al* 2016 TensorFlow: large-scale machine learning on heterogeneous systems arXiv:1603.04467

[33] Hinton G E *et al* 2012 Improving neural networks by preventing co-adaptation of feature detectors 07 arXiv:1207.0580

[34] Srivastava N *et al* 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58

[35] Gal Y 2016 Uncertainty in deep learning *PhD Thesis* University of Cambridge

[36] Mackay D J C 1991 Bayesian methods for adaptive models *PhD Thesis* California Institute of Technology

[37] Jordan M I *et al* 1999 An introduction to variational methods for graphical models *Mach. Learn.* **37** 183–233

[38] Osband I 2016 Risk versus uncertainty in deep learning:bayes, bootstrap and the dangers of dropout *NIPS*

[39] Hron J *et al* 2018 *Variational Bayesian dropout: pitfalls and fixes* arXiv:1807.01969

[40] Huggins H J *et al* 2018 Practical bounds on the error of bayesian posterior approximations: a nonasymptotic approach arXiv:1809.09505

[41] Bishop C M 1994 Novelty detection and neural network validation *IEE Proc., Vis. Image Signal Process.* **141** 217–22

Appendix A.

List of W7-X experiment numbers

The following experiments from the OP1.2 W7-X experimental campaign (2018) were considered in the study of section 5.4.

Day 1, 08/07/2018:
20180807.013 20180807.015 20180807.016 20180807.018 20180807.019 20180807.020 20180807.021 20180807.022 20180807.023 20180807.024 20180807.026 20180807.027 20180807.030 20180807.031

Day 2, 08/14/2018:
20180814.006 20180814.007 20180814.008 20180814.009 20180814.010 20180814.011 20180814.012 20180814.013 20180814.014 20180814.015 20180814.016 20180814.017 20180814.018 20180814.019 20180814.020 20180814.021 20180814.022 20180814.023 20180814.024 20180814.025 20180814.026 20180814.027 20180814.028 20180814.029 20180814.030 20180814.031 20180814.032 20180814.033 20180814.034 20180814.035 20180814.036 20180814.037 20180814.038 20180814.039 20180814.040 20180814.041 20180814.042 20180814.043 20180814.044 20180814.045 20180814.046 20180814.047 20180814.048 20180814.049 20180814.050

Day 3, 10/10/2018:
20181010.005 20181010.006 20181010.007 20181010.008 20181010.009 20181010.010 20181010.011 20181010.012 20181010.013 20181010.014 20181010.015 20181010.016 20181010.018 20181010.019 20181010.020 20181010.021 20181010.022 20181010.023 20181010.026 20181010.027 20181010.028 20181010.029 20181010.030 20181010.031 20181010.032 20181010.033 20181010.034 20181010.035 20181010.036 20181010.037 20181010.038 20181010.040

Day 4, 10/16/2018:
20181016.005 20181016.006 20181016.007 20181016.008 20181016.009 20181016.010 20181016.011 20181016.012 20181016.013 20181016.014 20181016.015 20181016.016 20181016.017 20181016.018 20181016.019 20181016.020 20181016.021 20181016.022 20181016.023 20181016.024 20181016.025 20181016.026 20181016.027 20181016.031 20181016.033 20181016.039 20181016.040

Day 5, 10/17/2018:
20181017.013 20181017.015 20181017.016 20181017.017 20181017.018 20181017.019 20181017.020 20181017.021 20181017.022 20181017.023 20181017.024 20181017.025 20181017.026 20181017.030 20181017.031 20181017.032 20181017.033 20181017.039 20181017.040 20181017.041

Day 6, 10/18/2018:
20181018.005 20181018.006 20181018.008 20181018.010 20181018.011 20181018.012 20181018.013 20181018.014 20181018.015 20181018.016 20181018.017 20181018.018 20181018.019 20181018.020 20181018.021 20181018.022 20181018.023 20181018.024 20181018.025 20181018.026 20181018.027 20181018.028 20181018.030 20181018.031 20181018.032 20181018.033 20181018.034 20181018.035 20181018.036 20181018.037 20181018.038 20181018.039 20181018.040 20181018.041

Acknowledgements

This work has been carried out within the framework of the EUROfusion consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

I would like to thank Prof. Dr. Robert C. Wolf for giving me the opportunity to carry out my Ph.D. work at the Max-Planck Institute for Plasma Physics in Greifswald. During the years I worked under his supervision, he has always demonstrated to be open minded and he has contributed in several occasions with insightful comments and discussions, especially in the time when I had been writing the thesis. I also wish to give him recognition for supervising my work in such a way that I have always felt comfortable with.

A special mention goes to Dr. Jakob Svensson, who, truly, has been more than a supervisor during these years. He has been an exceptional mentor, offering teachings beyond what was required by the work of the thesis. Especially, I am thankful for how he allowed communication and discussion of ideas to flow between us, always being open and respectful. Under his supervision, I have been able to carry out my work with a comfortable feeling of freedom and, at the same time, I have always felt inspired to do my best. He has never failed to show trust in my abilities, and this allowed me to grow as a scientist and as a person.

This work was possible thanks to the contribution of many people. Among these, I would like to mention Dr Andreas Langenberg and Dr Matthias Brix for always being available to provide help and suggestions when I applied my research to the measurements they are responsible for; Dr Udo Hoefel and Sehyun Kwak for being not just brilliant co-workers, but also friends and office mates whose company has been much needed in several occasions. Many others have contributed during these years: some relations have evolved into friendships, lasting until now, despite the distance that might separate us; many relations have stayed professionals, providing invaluable support in many occasions. Al-

though you all deserve to be mentioned, I will not attempt the impossible task of listing here all your names without forgetting someone. Nevertheless, I wish to thank you all.

Last but not least, I want to mention the immense gratitude that I feel towards the people that are part of my family. Despite the distance which separates us, they have never failed to provide me with the warmth of their closeness, in such a way that made possible for me to conduct my research - in professional and personal life - with a sense of infused courage and safety. The extent to which I feel free and encouraged to express my abilities in this world is deeply rooted in that sense of safety and I owe it to them. My affection and love go to all of them.

Statutory declaration

I hereby declare in accordance with the examination regulations that I myself have written this document, that no other sources as those indicated were used and all direct and indirect citations are properly designated, that the document handed in was neither fully nor partly subject to another examination procedure or published and that the content of the electronic exemplar is identical to the printing copy.

Greifswald, 8th October 2020

Andrea Pavone

List of Figures

List of Tables

Publications as first author

In this chapter, I have collected a list of the peer-reviewed article of which I am first author in the format of a bibliography. They have been published during the years of my Ph.D. studies and constitute the main outcome of my work during this time.

Peer-reviewed articles

[1] A. Pavone et al. »Neural network approximated Bayesian inference of edge electron density profiles at JET«. In: *Plasma Physics and Controlled Fusion*, Vol. 62.4 (Mar. 2020), page 045019. DOI: 10.1088/1361-6587/ab7732. The publication describes how neural networks can be trained to approximate Bayesian inference based on an existing Bayesian model for the reconstruction of the edge electron density profiles at the JET tokamak. We demonstrate here that the method previously developed and tested at the W7-X stellarator can be generalized to a completely new physics system, therefore hinting to the possibility, in future, to automatise the procedure for approximating any physics model implemented within the same Bayesian modeling framework. The method is also extensively tested on a large number of different experimental cases, and compared to the conventional Bayesian inference results. We show also how the uncertainties of the network prediction can be calculated with an approach which relies on the deep learning technique known as dropout training and an interpretation of the training problem as a variational inference problem in the context of Bayesian neural networks. I have written the entire text of the publication, developed the neural network approximation based on the Bayesian model and the dropout based uncertainty estimation, as well as performed all the calculations that allowed to obtain the results. M. Brix contributed as main responsible for the diagnostic device with his deep knowledge and understanding of the

physics involved in the measurements at the JET tokamak. J. Svensson is the author of the Minerva Bayesian modeling framework which is used to develop Bayesian models and carry out Bayesian inference, and contributed with his insights and wide knowledge of Bayesian inference and neural networks. S. Kwak is the main author of the physics model for the inference of the density profiles as implemented within the Minerva framework and contributed with invaluable comments and insightful discussions. All co-authors have contributed by extensively engaging in insightful scientific discussions and making possible the different measurements involved in the work.

[2] A. Pavone et al. »Measurements of visible bremsstrahlung and automatic Bayesian inference of the effective plasma charge Zeff at W7-X«. In: *Journal of Instrumentation*, Vol. 14.10 (Oct. 2019), pages C10003–C10003. doi: 10.1088/1748-0221/14/10/c10003. The publication describes the first measurements of visual bremsstrahlung and inference of the plasma effective charge Z_{eff} available at W7-X. Both the features of the measurement device and the Bayesian model and inference are described. Especially, the Bayesian inference is run automatically after each plasma shot and it includes the inference of the electron temperature and density profiles from independent measurements of the Thomson scattering diagnostic based on Gaussian processes. I have written the entire text of the publication and developed and implemented the Bayesian models for the inference of the plasma effective charge and the electron temperature and density profiles. These models were developed within the Minerva framework, the original author of which is J. Svensson, among the co-authors. U. Hergenhahn and M. Krychowiak have contributed as main responsibles for the actual realization of the diagnostic device and their deep understanding of the functioning of the measurement processes and the physics involved. The other coauthors contributed as diagnosticians involved in the development and operation of the Thomson scattering diagnostic, providing the corresponding data and by extensively engaging in insightful scientific discussions.

[3] A. Pavone et al. »Neural network approximation of Bayesian models for the inference of ion and electron temperature profiles at W7-X«. In: *Plasma Physics and Controlled Fusion*, Vol. 61.7 (May 2019), page 075012. doi: 10.1088/1361-6587/ab1d26. The publication describes the ap-

plication of neural network as approximated Bayesian inference to the problem of the temperature profile reconstruction from X-ray imagining diagnostic measurements. In particular, we describe how the network can be trained to appoximate the inference carried out with an existing Bayesian model by generating the training data with the given model, with the advantage of a significative acceleration of the data analysis. At the same time, we show how the quality of the training set can be assesed with respect to how well they describe the experimental measurements, a problem that can easily arise and has to be tackled when dealing with training data generated synthetically. We also demonstrate the performance of the network on a number of experimental cases, comparing its reconstruction with the profiles infered with the Bayesian model. I am the original and exclusive author of all the text and content in the publication, as well as the person who developed the approximation framework which the training method is based on, the neural network model and other mentioned algorithms, and made the final evaluation and comparison on experimental data. The data used in the comparison and inferred with the Bayesian model were provided by A. Langenberg, who is also the main author of the model and contributed with his deep knowledge of the diagnostic and the physics involved in the measurement processes. J. Svensson is the main author of the Minerva framework, where the Bayesian model is implemented and which is used to carry out Bayesian inference and generate the data to train the network. He has also contributed with invaluable and copious insights and discussions. All co-authors have contributed by extensively engaging in insightful scientific discussions and making possible the different measurements involved in the work.

[4] A. Pavone et al. »Bayesian uncertainty calculation in neural network inference of ion and electron temperature profiles at W7-X«. In: *Review of Scientific Instruments*, Vol. 89.10 (2018). DOI: 10.1063/1.5039286. In the publication we describe how uncertainties of the neural network output can be calculated in a Bayesian framework. The framework is known as Bayesian neural network (BNN) and consists of an interpretation of the network model as a Bayesian model and the training problem as an inference problem. Under the so called Laplace approximation, it is possible to derive an analytical expression of the error bars dependent on the

Hessian matrix of the training loss funciton. We apply this calculation to the case of the inference of ion and electron temperature profiles at W7-X, accounting also for noise in the input data and the presence of different local minima after training by a Monte Carlo scheme of sampling from the input space and a committee of networks. I have written the entire text of the publication, developed and implemented the BNN models and sampling scheme for the case under consideration, and all the calculations involved in the achievement of the results and applied the method to the experimental data. The measured data were provided by A. Langenberg, who is responsible for the diagnostic performing the measurement at W7-X. J. Svensson contributed vastly by providing invaluable insights, and spending his time in precious conversation with me. All coauthors have contributed by extensively engaging in insightful scientific discussions and making possible the different measurements involved in the work.

Publications as coauthor

The articles of which I am coauthor represent a collection of works which can be seen as a spin-off activity of the main research presented in this thesis. I have contributed to these articles in various ways, in general by developing Bayesian models and applying Bayesian inference to different physics systems. Because Bayesian inference is one of the main subjects involved in this thesis, I consider these works to be a relevant corollary of my main project. Here, I will provide, case by case, an explanation of how I have contributed to them.

Peer-reviewed articles

[1] I. Abramovic et al., amongst them **A. Pavone**. »Forward modeling of collective Thomson scattering for Wendelstein 7-X plasmas: Electrostatic approximation«. In: *Review of Scientific Instruments*, Vol. 90.2 (2019), page 023501. DOI: 10.1063/1.5048361. eprint: https://doi.org/10.1063/1.5048361. The collective Thomson scattering diagnostic allows to perform local measurements of the ion temperature in the core region of Wendelstein 7-X. A forward model of this diagnostic has been implemented within the Bayesian modeling framework Minerva. I have collaborated to the implementation of such model and the Bayesian inference procedure which allowed to obtain the ion temperature information given the measured emission. The inference was then performed on data collected during the OP 1.1 and OP 1.2 campaigns.

[2] T. Andreeva et al., amongst them **A. Pavone**. »Equilibrium evaluation for Wendelstein 7-X experiment programs in the first divertor phase«. In: *Fusion Engineering and Design*, (2019). DOI: https://doi.org/10.1016/j.fusengdes.2018.12.050. In this work, the first author makes use of the plasma effective charge Z_{eff} in order to assess the overall contamination of low-Z impurities in the plasma and estimate the pressure profile necessary for the equilibrium calculation. Z_{eff} can

be infered by Bremsstrahlung measurements. I have contributed to the development of a Bayesian model of the plasma Bremsstrahlung emission, which allowed to infer a line-of-sight averaged plasma effective charge Z_{eff} from measurements performed with a spectrometer. Electron density and temperature profiles are also required to calculate the Bremsstrahlung emission; I have used Bayesian inference and Gaussian processes to infer them from independent diagnostic measurements.

[3] F. Effenberg et al., amongst them **A. Pavone**. »First demonstration of radiative power exhaust with impurity seeding in the island divertor at Wendelstein 7-X«. In: *Nuclear Fusion*, Vol. 59.10 (Aug. 2019), page 106020. DOI: 10.1088/1741-4326/ab32c4. Impurities play a crucial role in the work presented in this paper; their behavior can be studied and understood by mean of the plasma effective charge Z_{eff}, which was estimated from the measurements of a single line-of-sight spectrometer. I have extensively worked on the Bayesian model and inference necessary to infer Z_{eff}, which also included the Bayesian inference of independently measured electron density and temperature profiles.

[4] D. Zhang et al., amongst them **A. Pavone**. »First observation of a stable highly-dissipative divertor plasma regime on the Wendelstein 7-X stellarator«. In: *Phys. Rev. Lett.*, Vol. 123 (2 July 2019), page 025002. DOI: 10.1103/PhysRevLett.123.025002. The work presented in this paper is based on the radiation behavior of impurities. A quantity which describes the overall low-Z plasma impurity concentration is Z_{eff}. This can be estimated by diagnostic measurements of the Bremmstrahlung emission in a given wavelength range. I have extensively worked on the Bayesian inference and modeling which allowed the estimation of Z_{eff} from spectral measurements, collected during the OP 1.2 experimental campaign. The Z_{eff} found in this way was then used to substantiate the radiation behavior of the impurities.

[5] A. Langenberg et al., amongst them **A. Pavone**. »Prospects of X-ray imaging spectrometers for impurity transport: Recent results from the stellarator Wendelstein 7-X (invited)«. In: *Review of Scientific Instruments*, Vol. 89.10 (2018), 10G101. DOI: 10.1063/1.5036536. eprint: https://doi.org/10.1063/1.5036536. The X-ray imaging spectrometer diagnostic collects the spectral emission resulting from the interacation between impurities and the plasma electrons along several line of sight.

Traditional Bayesian inference can be applied to the measurements in order to infer plasma profiles. This procedures, although accurate, is typically slow. For this reason, I have developed a fast neural network inversion based on the corresponding Bayesian model, which allows to quickly infer electron and ion temperature profiles. As a consequence, I have extensively studied and contributed to the Bayesian model and inference, of which the first author of this paper was the main developer.

[6] I. Abramovic et al., amongst them **A. Pavone**. »Collective Thomson scattering data analysis for Wendelstein 7-X«. In: *Journal of Instrumentation*, Vol. 12.08 (Aug. 2017), pages C08015–C08015. DOI: 10.1088/1748-0221/12/08/c08015. The collective Thomson scattering diagnostic allows to perform local measurements of the ion temperature in the core region of Wendelstein 7-X. A forward model of this diagnostic has been implemented within the Bayesian modeling framework Minerva. I have extensively taken part in the development of the Bayesian model and contributed significantly to the study of the collected measurements, carried out within the framework of Bayesian inference.

Bibliography

[1] J. Svensson and A. Werner. »Large Scale Bayesian Data Analysis for Nuclear Fusion Experiments«. In: *IEEE International Symposium on Intelligent Signal Processing*, (2007), pages 1–6. doi: 10.1109/wisp.2007.4447579.

[2] A. Vaswani et al. »Attention Is All You Need«. 2017. arXiv: 1706.03762 [cs.CL].

[3] Z. Lan et al. »ALBERT: A Lite BERT for Self-supervised Learning of Language Representations«. 2019. arXiv: 1909.11942 [cs.CL].

[4] O. Vinyals et al. »Grandmaster level in StarCraft II using multi-agent reinforcement learning«. In: *Nature*, Vol. 575.7782 (2019), pages 350–354. doi: 10.1038/s41586-019-1724-z.

[5] OpenAI et al. »Solving Rubik's Cube with a Robot Hand«. 2019. arXiv: 1910.07113 [cs.LG].

[6] H. Jeffreys. *Theory of Probability*. Third. Oxford, 1961.

[7] E. T. Jaynes. *Probability theory: The logic of science*. Cambridge: Cambridge University Press, 2003.

[8] A. Langenberg et al. »Inference of temperature and density profiles via forward modeling of an x-ray imaging crystal spectrometer within the Minerva Bayesian analysis framework«. In: *Review of Scientific Instruments*, Vol. 90.6 (2019), page 063505. doi: 10.1063/1.5086283. eprint: https://doi.org/10.1063/1.5086283.

[9] U. Hoefel et al. »Bayesian modeling of microwave radiometer calibration on the example of the Wendelstein 7-X electron cyclotron emission diagnostic«. In: *Review of Scientific Instruments*, Vol. 90.4 (2019), page 043502. doi: 10.1063/1.5082542. eprint: https://doi.org/10.1063/1.5082542.

[10] S. Kwak et al. »Bayesian electron density inference from JET lithium beam emission spectra using Gaussian processes«. In: *Nuclear Fusion*, Vol. 57.3 (2017), page 036017. DOI: 10.1088/1741-4326/aa5072.

[11] S. Kwak et al. »Bayesian modelling of Thomson scattering and multichannel interferometer diagnostics using Gaussian processes«. Nuclear Fusion. 2020. URL: http://iopscience.iop.org/10.1088/1741-4326/ab686e.

[12] S. Kwak et al. »Bayesian modelling of multiple diagnostics at Wendelstein 7-X«. 2020. to be submitted.

[13] A. Langenberg et al. »Forward modeling of X-ray imaging crystal spectrometers within the Minerva Bayesian analysis framework«. In: *Fusion Science and Technology*, (2016). DOI: 10.13182/FST15-181.

[14] A. Langenberg et al. »Prospects of X-ray imaging spectrometers for impurity transport: Recent results from the stellarator Wendelstein 7-X (invited)«. In: *Review of Scientific Instruments*, Vol. 89.10 (2018), 10G101. DOI: 10.1063/1.5036536. eprint: https://doi.org/10.1063/1.5036536.

[15] M. Brix et al. »Recent improvements of the JET lithium beam diagnostic«. In: *Review of Scientific Instruments*, Vol. 83.10 (2012), page 10D533. DOI: 10.1063/1.4739411. eprint: https://doi.org/10.1063/1.4739411.

[16] S. Kwak et al. »Bayesian modelling of the emission spectrum of the Joint European Torus Lithium Beam Emission Spectroscopy system«. In: *Review of Scientific Instruments*, Vol. 87.2 (2016), page 023501. DOI: 10.1063/1.4940925. eprint: https://doi.org/10.1063/1.4940925.

[17] G. H. Neilson, editor. *Magnetic Fusion Energy: From Experiments to Power Plants*. Woodhead Publishing, 2016.

[18] A. Pavone et al. »Neural network approximation of Bayesian models for the inference of ion and electron temperature profiles at W7-X«. In: *Plasma Physics and Controlled Fusion*, Vol. 61.7 (May 2019), page 075012. DOI: 10.1088/1361-6587/ab1d26.

[19] A. Pavone et al. »Bayesian uncertainty calculation in neural network inference of ion and electron temperature profiles at W7-X«. In: *Review of Scientific Instruments*, Vol. 89.10 (2018). doi: 10.1063/1.5039286.

[20] A. Pavone et al. »Neural network approximated Bayesian inference of edge electron density profiles at JET«. In: *Plasma Physics and Controlled Fusion*, Vol. 62.4 (Mar. 2020), page 045019. doi: 10.1088/1361-6587/ab7732.

[21] S. Bozhenkov et al. »The Thomson scattering diagnostic at Wendelstein 7-X and its performance in the first operation phase«. In: *Journal of Instrumentation*, Vol. 12.10 (Oct. 2017), P10004–P10004. doi: 10.1088/1748-0221/12/10/p10004.

[22] T. Klinger et al. »Overview of first Wendelstein 7-X high-performance operation«. In: *Nuclear Fusion*, Vol. 59.11 (June 2019), page 112004. doi: 10.1088/1741-4326/ab03a7.

[23] P. Helander et al. »Stellarator and tokamak plasmas: a comparison«. In: *Plasma Physics and Controlled Fusion*, Vol. 54.12 (Nov. 2012), page 124009. doi: 10.1088/0741-3335/54/12/124009.

[24] P. Helander. »Theory of plasma confinement in non-axisymmetric magnetic fields«. In: *Reports on Progress in Physics*, Vol. 77.8 (July 2014), page 087001. doi: 10.1088/0034-4885/77/8/087001.

[25] G. Grieger et al. »Physics optimization of stellarators«. In: *Physics of Fluids B: Plasma Physics*, Vol. 4.7 (1992), pages 2081–2091. doi: 10.1063/1.860481. eprint: https://doi.org/10.1063/1.860481.

[26] T. S. Pedersen et al. »First results from divertor operation in Wendelstein 7-X«. In: *Plasma Physics and Controlled Fusion*, Vol. 61.1 (Nov. 2018), page 014035. doi: 10.1088/1361-6587/aaec25.

[27] P. Rebut, R. Bickerton and B. Keen. »The Joint European Torus: installation, first results and prospects«. In: *Nuclear Fusion*, Vol. 25.9 (Sept. 1985), pages 1011–1022. doi: 10.1088/0029-5515/25/9/003.

[28] D. Meade. »Tokamak Fusion Test Reactor D-T results«. In: *Fusion Engineering and Design*, Vol. 30.1-2 (May 1995), pages 13–23. doi: 10.1016/0920-3796(94)00398-Q.

[29] J. Team. »Fusion energy production from a deuterium-tritium plasma in the JET tokamak«. In: *Nuclear Fusion*, Vol. 32.2 (Feb. 1992), pages 187–203. doi: 10.1088/0029-5515/32/2/i01.

[30] J. Svensson. »Connecting theory and experiments in complex systems«. Plasma Physics Summer School Talk. 2008-present.

[31] R. Fischer et al. »Thomson scattering analysis with the Bayesian probability theory«. In: *Plasma Physics and Controlled Fusion*, Vol. 44.8 (July 2002), pages 1501–1519. doi: 10.1088/0741-3335/44/8/306.

[32] R. Fischer et al. »Integrated Data Analysis of Profile Diagnostics at ASDEX Upgrade«. In: *Fusion Science and Technology*, Vol. 58.2 (2010), pages 675–684. doi: 10.13182/FST10-110. eprint: https://doi.org/10.13182/FST10-110.

[33] G. Cottrell. »Maximum entropy and plasma physics«. 1990. url: http://www.euro-fusionscipub.org/wp-content/uploads/2014/11/JETP90004.pdf.

[34] N. A. Pablant et al. »Investigation of ion and electron heat transport of high-TeECH heated discharges in the large helical device«. In: *Plasma Physics and Controlled Fusion*, Vol. 58.4 (Jan. 2016), page 045004. doi: 10.1088/0741-3335/58/4/045004.

[35] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988. doi: https://doi.org/10.1016/C2009-0-27609-4.

[36] R. Hooke and T. A. Jeeves. »" Direct Search" Solution of Numerical and Statistical Problems«. In: *J. ACM*, Vol. 8.2 (Apr. 1961), pages 212–229. doi: 10.1145/321062.321069.

[37] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1961.

[38] R. Hahnloser et al. »Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit«. In: *Nature*, Vol. 405 (2000). doi: 10.1038/35016072.

[39] X. Glorot, A. Bordes and Y. Bengio. »Deep Sparse Rectifier Neural Networks«. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Edited by G. Gordon, D. Dunson and M. Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Nov. 2011, pages 315–323. URL: http://proceedings.mlr.press/v15/glorot11a.html.

[40] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[41] Theano Development Team. »Theano: A Python framework for fast computation of mathematical expressions«. In: *arXiv e-prints*, Vol. abs/1605.02688 (2016). URL: http://arxiv.org/abs/1605.02688.

[42] Martín Abadi et al. »TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems«. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[43] D. J. MacKay. »Bayesian Interpolation«. In: *Neural Computation*, Vol. 4 (1991), pages 415–447.

[44] C. M. Bishop, C. M. Roach and M. G. von Hellermann. »Automatic analysis of JET charge exchange spectra using neural networks«. In: *Plasma Physics and Controlled Fusion*, Vol. 35 (1993), pages 765–773.

[45] J. Svensson, M. von Hellermann and R. W. T. König. »Analysis of JET charge exchange spectra using neural networks«. In: *Plasma Physics and Controlled Fusion*, Vol. 41 (1999), pages 315–338.

[46] J. Svensson et al. »Real-time ion temperature profiles in the JET nuclear fusion experiment«. In: *ICANN 98. Perspectives in Neural Computing*, (1998).

[47] D. Böckenhoff et al. »Reconstruction of magnetic configurations in W7-X using artificial neural networks«. In: *Nuclear Fusion*, Vol. 58.5 (Mar. 2018), page 056009. DOI: 10.1088/1741-4326/aab22d.

[48] M. Blatzheim and D. B. and. »Neural network regression approaches to reconstruct properties of magnetic configuration from Wendelstein 7-X modeled heat load patterns«. In: *Nuclear Fusion*, Vol. 59.12 (Oct. 2019), page 126029. DOI: 10.1088/1741-4326/ab4123.

[49] D. R. Ferreira et al. »Full-Pulse Tomographic Reconstruction with Deep Neural Networks«. In: *Fusion Science and Technology*, Vol. 74.1-2 (2018), pages 47–56. doi: 10.1080/15361055.2017.1390386. eprint: https://doi.org/10.1080/15361055.2017.1390386.

[50] O. Meneghini et al. »Modeling of transport phenomena in tokamak plasmas with neural networks«. In: *Physics of Plasmas*, Vol. 21.6 (2014), page 060702. doi: 10.1063/1.4885343. eprint: https://doi.org/10.1063/1.4885343.

[51] O. Meneghini et al. »Self-consistent core-pedestal transport simulations with neural network accelerated models«. In: *Nuclear Fusion*, Vol. 57.8 (July 2017), page 086034. doi: 10.1088/1741-4326/aa7776.

[52] K. L. van de Plassche et al. »Fast modelling of turbulent transport in fusion plasmas using neural networks«. 2019. arXiv: 1911.05617 [physics.plasm-ph].

[53] D. J. C. Mackay. *Bayesian Methods for Adaptive Models*. PhD thesis. California Institute of Technology, 1991.

[54] Y. Gal. *Uncertainty in Deep Learning*. PhD thesis. University of Cambridge, 2016.

[55] A. Pavone et al. »Measurements of visible bremsstrahlung and automatic Bayesian inference of the effective plasma charge Zeff at W7-X«. In: *Journal of Instrumentation*, Vol. 14.10 (Oct. 2019), pages C10003–C10003. doi: 10.1088/1748-0221/14/10/c10003.

[56] R. S. Sutherland. »Accurate free-free Gaunt factors for astrophysical plasmas«. In: *Monthly Notices of the Royal Astronomical Society*, Vol. 300 (1998), pages 321–330. doi: 10.1046/j.1365-8711.1998.01687.x.

[57] D. J. MacKay. »Bayesian model comparison and backprop nets«. In: *Advances in neural information processing systems*. 1992, pages 839–846.

[58] O. P. Ford et al. »Charge exchange recombination spectroscopy at Wendelstein 7-X«. In: *Review of Scientific Instruments*, Vol. 91.2 (2020), page 023507. doi: 10.1063/1.5132936. eprint: https://doi.org/10.1063/1.5132936.

[59] L. Vanó and other. »Studies on carbon content and transport with Charge Exchange Spectroscopy on W7-X«. In: *EPS Proceedings*, (2019).

[60] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[61] I. Loshchilov and F. Hutter. »Decoupled Weight Decay Regularization«. 2017. arXiv: 1711.05101 [cs.LG].