UNIVERSIDADE D
COIMBRA

Valéria Maria Dinis Carvalho

MAPPING NEUTRON STAR OBSERVATIONS TO THE EQUATION OF STATE:
A BAYESIAN NEURAL NETWORK APPROACH

Dissertation submitted for the Master's Degree in Nuclear and Particle Physics supervised by Dr. Márcio Rafael Baptista Ferreira and Professor and Dr. Maria Constança Mendes Pinheiro da Providência Santarém e Costa and presented to the Physics Department of the Faculty of Science and Technology of the University of Coimbra.

September of 2023

Mapping Neutron Star Observations to the Equation of State:

A Bayesian Neural Network Approach

Valéria Maria Dinis Carvalho

Supervisor:
Dr. Márcio Rafael Baptista Ferreira

Co-Supervisor:
Prof. Dr. Maria Constança Mendes Pinheiro da Providência Santarém e Costa

Dissertation submitted in partial fulfillment for the degree of Master of Science in Physics.

Coimbra, September 2023

Acknowledgments

This thesis would not have been possible without the indispensable support of my supervisors:

Dr. Márcio Ferreira, whose guidance and support from the very beginning has been invaluable. His willingness and knowledge to answer my questions and help me make important decisions have laid the foundation for the success of this work and my critical thinking. I must also acknowledge Prof. Dr. Constança Providência, whose knowledge and availability to help whenever challenges arose were valuable to this work. It has been a privilege and an enriching experience to work under their guidance.

I thank my friends who have supported me over the years.

Finally, I'm eternally grateful to my family for their affection, support and efforts on my behalf.

This thesis was supported with funds from FCT/MCTES under the project MagStarMatter, with reference 2022.06460.PTDC.

Resumo

As estrelas de neutrões (NS)s destacam-se como objetos astrofísicos excecionalmente enriquecedores para a compreensão e restrição da Equação de Estado (EoS) da matéria nuclear. Estes objetos, sujeitos a condições extremas, oferecem uma oportunidade única para explorar territórios desconhecidos no diagrama de fases da Cromodinâmica Quântica. No entanto, persiste o desafio de traduzir as observações das NSs em informação pertinente sobre a sua composição e a correspondente EoS. A natureza limitada e incerta das observações disponíveis piora ainda mais este mapeamento. Para resolver este problema crucial, é necessária uma ferramenta capaz de mapear diretamente as observações de NSs para a EoS, tendo em conta as incertezas inerentes aos dados. Durante este trabalho, pretendemos atingir este objetivo através da aplicação de redes neuronais bayesianas(BNN)s. Embora as metodologias de aprendizagem computacional, e mais ainda as redes neuronais, se tenham aventurado anteriormente a abordar a tarefa de mapear as observações para a EoS, tem faltado um aspeto crítico: a capacidade de captar a incerteza. É aqui que as BNNs surgem como uma solução inovadora, capaz não só de estabelecer ligações entre as observações e a EoS, mas também de encapsular as incertezas tanto no modelo como no conjunto de dados. Para atingir este objetivo, treinamos os nossos modelos das BNNs utilizando um conjunto de dados abrangente que engloba 25 000 EoS nucleares no âmbito do Campo Médio Relativístico (RMF). Este conjunto de dados é construído através de inferência Bayesiana, limitado por restrições mínimas a baixa densidade. Abrangendo observáveis de NSs como o raio, a massa e a deformabilidade das maré, este conjunto de dados contém a fração de protões e a velocidade do som nestes interiores enigmáticos.

Com o intuito de reproduzir observações reais, introduzimos modificações no conjunto de dados utilizado, incorporando ruído no vetor de entrada do nosso modelo. Isto foi feito de quatro formas diferentes para os conjuntos de dados de treino e de teste. Os nossos resultados demonstram que os modelos das rede neuronais bayesianas correlacionam com precisão as observações com as propriedades intrínsecas das NSs, ao mesmo tempo que fornecem uma medida quantificável da incerteza. Este resultado mantém-se consistente mesmo quando o modelo é testado com dados simulados do conjunto de dados DD2, uma classe que também pertence aos modelo do RMF mas com acoplamentos dependentes da densidade, utilizado para gerar as EoS que serviram para testar o modelo das BNNs.

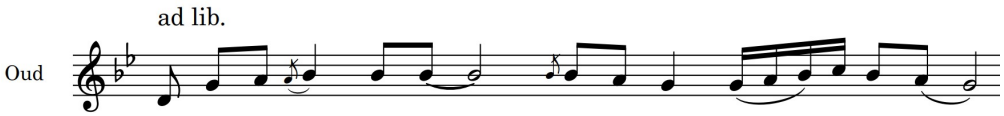
Palavras-chave: Estrelas de Neutrões, Redes Neuronais Bayesianas, Quantificação de Incerteza, Equação de Estado

Abstract

Neutron Star (NS)s stand out as uniquely compelling astrophysical objects for comprehending and constraining the Equation of State (EoS) of nuclear matter. These objects, subjected to extreme conditions, offer a unique opportunity to explore uncharted territories within the Quantum Chromodynamics phase diagram. However, the challenge persists in translating observations of NSs into meaningful insights about their composition and the corresponding EoS. The limited and uncertain nature of available observations further compounds this issue. Addressing this crucial problem necessitates a tool capable of directly mapping NS observations to the EoS while being aware of inherent data uncertainties. Herein, we aim to achieve this objective through the application of Bayesian Neural Network (BNN)s. While machine learning methodologies, even more so neural networks, have previously ventured into addressing the task of mapping observations to the EoS, a critical aspect has been missing: the ability to capture uncertainty. This is where BNNs emerge as a groundbreaking solution, capable of not only establishing connections between observations and the EoS but also encapsulating uncertainties within both the model and the dataset.

In pursuit of this goal, we train our BNN models using a comprehensive dataset encompassing 25,000 nuclear EoS within the Relativistic Mean Field (RMF) framework. This dataset is constructed through Bayesian inference, constrained by minimal low-density constraints. Spanning NS observables such as radius, mass, and tidal deformability, this dataset gives insights into the proton fraction and sound speed within these enigmatic interiors. To replicate real-world observations, we've introduced modifications to the dataset employed by incorporating noise into the input vector of our model. This was done in four different ways for both the training and testing datasets. Our results demonstrate the BNN models accurately correlate observations with the intrinsic properties of NSs, all while providing a quantifiable measure of the uncertainty. This achievement remains consistent even when the model is tested with simulated data from the DD2 dataset, a class that also belongs to the RMF models but with density-dependent couplings, used to generate the EoS used to test the BNN model.

Keywords: Neutron Stars, Bayesian Neural Networks, Uncertainty Quantification, Equation of state



*"The Astounding Eyes of Rita" by **Anouar Brahem***

Contents

List of Acronyms

NS	Neutron Star
EoS	Equation of State
QCD	Quantum Chromodynamics
QHD	Quantum Hadrodynamics
RMF	Relativistic Mean Field
TOV	Tolman-Oppenheimer-Volkoff
eXTP	enhanced X-ray Timing and Polarimetry mission
STROBE-X	Spectroscopic Time-Resolving Observatory for Broadband Energy X-rays
SKA	Square Kilometer Array
ML	Machine Learning
TD	Tidal Deformability
KL	Kullback-Leibler
ELBO	Evidence Lower BOund
BNN	Bayesian Neural Network
NN	Neural Network
VI	Variational Inference
MCMC	Markov Chain Monte Carlo
ReLU	Rectified Linear Unit

List of Figures

List of Tables

Notation

Physics theory

- $\hbar = c = G = 1$.
- $g_{\mu\nu} = (+---)$ – Signature convention .
- $M_\odot$ and $R_\odot$ – Mass and radius of the sun, $M_\odot = 1,989 \times 10^{30}$ kg, $R_\odot = 696340$ km .
- $x \equiv \left(t, x, y, z\right)$, $k \cdot x \equiv k_0 t - \boldsymbol{k}.\boldsymbol{r}$.
- $\phi(x^\mu) \equiv \phi(x)$.
- Pauli matrices:

$$\tau_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \tau_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \tau_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1}$$

- $\gamma^0 = \beta$ and $\gamma^0\gamma^i = \alpha^i$ – Gamma matrices .
- $\mathbb{1}$ – Identity matrix.

Table 1: Mass(m) in MeV, Spin (J), isospin projection (I_3) and charge (q) for the particles considered in our system namely baryons, mesons and leptons.

		m (MeV)	J	I_3	q
Baryons	p	939	$\frac{1}{2}$	$\frac{1}{2}$	1
	n	939	$\frac{1}{2}$	$-\frac{1}{2}$	0
Mesons	φ	500	0	0	0
	ω	782	1	0	0
	ϱ	770	1	1	$-1, 0, 1$
Leptons	e^-	0.511	$\frac{1}{2}$	$-\frac{1}{2}$	-1
	μ^-	106	$\frac{1}{2}$	$-\frac{1}{2}$	-1

Probability theory

- $\mathcal{U}[\min, \max]$ – Uniform distribution.
- $\mathcal{N}(\mu, \sigma^2)$ – Gaussian distribution.
- $\mathbb{E}_q[X]$ – Represents the expected value with respect to the values of x sampled from the distribution q.

1 Introduction

1.1 Motivation

Neutron Star (NS)s, which are the second most compact objects in the universe, possess a matter, at $T = 0$, with a composition very similar to the idealized matter found in the interior of atomic nuclei since they exhibit key characteristics such as baryon composition and densities, differing by merely one order of magnitude or less, in comparison to the matter typically encountered in atomic nuclei. However, the extreme matter present within NSs cannot be replicated in terrestrial laboratories, since heavy-ion collisions involve high temperatures when compressing nuclear matter to very high densities. Also in numerical methods, lattice Quantum Chromodynamics (QCD) calculations work well for zero density however in the finite-density region they don't, because of the sign problem. Consequently, the EoS governing dense and asymmetric nuclear matter within NSs remains an intriguing and elusive quantity. The EoS plays a crucial role in describing the thermodynamic relationship between energy density and pressure. In order to obtain the model independent EoS, one must solve for QCD, the fundamental theory of strong interactions, the phase diagram is represented in Fig. 1.1. From a theoretical standpoint, the description of nuclear matter and the EoS poses challenges at different density regimes. Chiral effective field theory, an *ab initio* method [1], for instance, is a powerful framework applicable at low densities, $1-2n_0$ ($n_0 = 0.16$ fm^{-3} for the saturation density). On the other hand, perturbative QCD, (for a review [2]) provides a reliable description at extremely high densities, $n \geq 50n_0$. However, in the intermediate density regime, the EoS still lacks trustable predictions. In order to bridge this gap, the EoS still relies on phenomenological approaches using many-body methods and effective interactions such as RMF theory.

However, significant progress has been made in constraining NS matter through observations. For instance, measurements of the Shapiro delay in pulsars[1] observations, such as those obtained from PSR J1614-2230 [4, 5, 6] with a mass of $1.908 \pm 0.016 M_\odot$, PSR J0348 - 0432 [7] with a mass of $2.01 \pm 0.04 M_\odot$, PSR J0740+6620 [8] with a mass of $2.08 \pm 0.07 M_\odot$, and J1810+1714 [9] with a mass of $2.13 \pm 0.04 M_\odot$, have provided valuable constraints on NS properties. Additionally, observations of quiescent low-mass X-ray binary systems and thermonuclear burst sources have also contributed to our knowledge of NS physics [10, 11]. Recent results from the NICER (Neutron star Interior Composition Explorer) mission have provided even more accurate X-ray spectral timing results, further constraining the masses of pulsars like PSR J0030+045 [12, 13] and the radius of PSR J0740+6620 [14, 15, 16].

[1] Rotation-powered NSs.

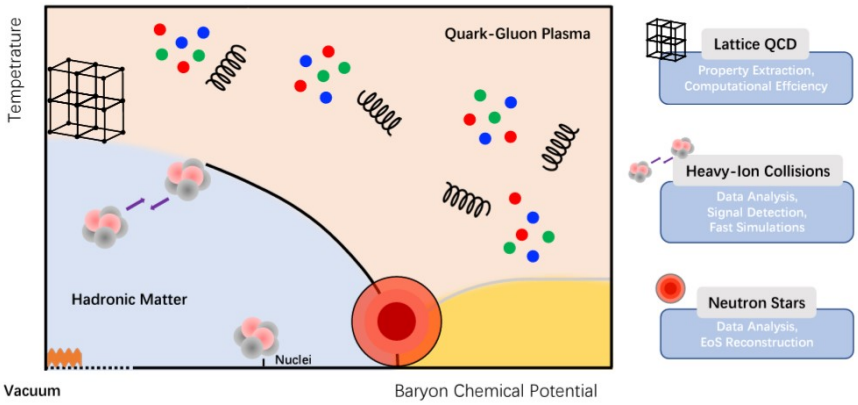


Figure 1.1: Schematic QCD phase diagram and the applications of ML in three domains, taken from [3].

The detection of gravitational waves from compact binary coalescence events, such as GW170817 [17] and GW190425 [18], by the LIGO[2]/Virgo collaboration has also played a significant role in constraining the EoS of NS matter. These events provide valuable information about the merger process and the resulting post-merger remnant, giving insights into the properties of dense matter. Future observations from upcoming experiments like the enhanced X-ray Timing and Polarimetry mission (eXTP) [19, 20], the Spectroscopic Time-Resolving Observatory for Broadband Energy X-rays (STROBE-X) [21], and the Square Kilometer Array (SKA) telescope [22] hold great promise for further improving our understanding of NSs. These missions will enable more precise measurements of NS radii and masses, with uncertainties reduced to a few percent. Typical NS observables are mass, radius, moment of inertia (I), quadrupole moment (Q), dimensionless Tidal Deformability (TD), compactness, also the last three observables can also be connected in I-Love-Q which relates I, k_2 and Q [23].

Having these observables, how can we construct the most adequate EoS?

Various numerical and statistical methods have been explored. Bayesian inference has been utilized [24, 25], allowing for probabilistic reasoning and estimation of model parameters. Non-parametric methods, such as Gaussian processes [26], have also been employed to capture complex relationships without relying on explicit functional forms. However, these methods possess certain drawbacks that should be considered. Some of the limitations include the computational complexity and the need for well-defined priors and assumptions in Bayesian inference, while Gaussian processes may face challenges in handling large datasets and scaling to higher dimensions.
In recent years, the application of Neural Network (NN)s in physics has gained significant attention as evidenced by the recent review [3], which highlights the use of NNs in various physics applications. NNs offer the advantage of a flexible function approximation and efficient training on large datasets. Building upon these advancements, our *first motivation* is to develop an artificial NN model that directly maps the observational properties of NS to the EoS. Specifically, our focus is on predicting the

[2] Laser Interferometer Gravitational-Wave Observatory

speed of sound and proton fraction by training and testing the model using a dataset generated from a RMF approach [25].

The use of artificial NNs in the realm of the dense matter EoS of NS has already been extensively explored in the literature. Several studies have investigated this topic, including the works by Fujimoto *et al.* [27, 28, 29], Soma *et al.* [30, 31], Chatterjee *et al.* [32], Morawski *et al.* [33], Ferreira *et al.* [34, 35], Krastev [36, 37] among others [38, 39]. Within these studies, our attention will be primarily directed towards articles that center on elucidating the mapping process from observed properties, like mass-radius or TD, to the underlying EoSs. Notably, we will delve into the works authored by Fujimoto *et al.* [28, 29, 27]. In this context, the final article serves as a culmination of the preceding two. Its core essence revolves around the intricate process of inverting the mapping, translating observables back into the EoS, facilitated through the utilization of a deep NN. The base dataset consists of an agnostic model made of polytropic EoS, where they simulate the observations by choosing 14 M-R pairs and applying a Gaussian noise [3], so they input to the model $\left[M_i, R_i, \sigma_{M,i}, \sigma_{R,i}\right]$ and the output is five values of the speed of sound squared. In order to obtain a better performance they implement data augmentation[4] and also compare their deep learning approach with polynomial regression, providing the robustness of their deep NN method. In order to quantify the output uncertainty they implement two methods, one based on the root-mean-square deviation for the validation dataset, and the other one based on an ensemble method. However, these aren't real uncertainties and only statistical estimations, which need to be better quantified. The same problem is found in the other articles aforementioned, as an example in [36, 37] that uses deep NNs to determine the properties of nuclear matter from either M-R or M-Λ, concludes once again, in both articles, that the uncertainty quantification is a big must do in future work, even more, to enhance the performance of the models. The ability to answer questions such as *How confident is a model about its predictions?* remains a crucial aspect to be addressed.

But how can we quantify the unpredictability of our model?

To tackle this challenge, we propose the implementation of a promising approach called BNN [40]. BNNs have gained attention in various fields of physics, such as [41, 42], and have shown promise in providing uncertainty quantification. Hence, our *second motivation* is to develop an inference framework that provides prediction uncertainties for any model prediction. By utilizing BNNs, we can capture and quantify the uncertainty associated with the predictions made by our model. This capability is crucial for decision-making processes that require a clear understanding of the confidence and reliability of the model's outputs. The study done on this work should be regarded as a *proof-of-concept study* realistic application regarding the true values of the observations are left for future work.

[3] This methodology is similar to the one we will use that is present in chapter 4.

[4] Technique used to artificially increase the dataset size.

1.2 Objective that leads to dissertation outline

In summary, the primary objective of this thesis is to establish a mapping from observational data of NS mass-radius and TD to the properties of the EoS, specifically focusing on predicting the speed of sound squared and proton fraction. Our approach involves employing BNNs as our modelling framework, allowing us to account for inherent uncertainties in the model as well as uncertainties in the observed data. By utilizing BNNs, we aim to gain a more comprehensive understanding of the predictive uncertainties involved in mapping observations to the EoS properties. Unlike many previous ML studies that use generic and agnostic parametrizations of the EoS, we adopt a specific family of nuclear models, the RMF model, to construct a set of possible EoSs. This microscopically motivated approach allows us to explore the density dependence of all degrees of freedom inside NSs, including the proton fraction.

To achieve the objectives set forth in this thesis, Chapter 2 will delve into the intricate properties of NSs, offering a comprehensive explanation of their key characteristics. A pivotal aspect of this chapter involves introducing the RMF model, which lays the foundation for the dataset we will employ. Moreover, we offer a succinct overview of the macroscopic attributes of NSs and elucidate the significance of the inverse mapping, which is the central topic of our research. Moving on to Chapter 3, we will introduce NNs, focusing specifically on the functioning of BNN. Chapter 4 will begin by presenting the base dataset created in a previous study [25]. Subsequently, we will discuss the adaptation of this dataset to suit our specific model, where is going to be used a similar approach as the one in [28, 29, 27]. In order to describe the model that we have employed in our study Chapter 5 will provide a comprehensive explanation of the architecture employed and the training process. Having established a solid foundation through theoretical insights and dataset description, we will delve into the heart of our findings in Chapter 6. Here, we will investigate the implications of introducing various forms of uncertainty into the input, while also evaluating the impact of the TD. Also, we will decompose the obtained uncertainty into the one acquired from the data and from the model. Furthermore, we will rigorously assess whether our model can successfully predict the DD2 EoS generated with a microscopic framework different from the one used to generate the EoS used to train the BNN model. Finally, in Chapter 7, we will provide a synthesis of the entire spectrum of results obtained throughout our study and discuss potential avenues for future research and the trajectory this field might undertake.

2 Bridging the Gap: Connecting Nuclear Models to Neutron Star Observations

In this chapter, we begin with an introductory exploration of NSs in Section 2.1. Moving forward, we delve into the computation of their composition within the framework of nonlinear relativistic nuclear field theory. We study the Lagrangian and the corresponding RMF approximation in Section 2.2, ultimately deriving the EoS. Subsequently, Section 2.3 delves into the macroscopic attributes that emerge from observing these stellar entities. We introduce the distinct macroscopic properties that arise when observing NSs, along with the central problem that this thesis seeks to address.

2.1 Neutron stars in a nutshell

Neutron Star (NS)s stand out as captivating cosmic laboratories for investigating fundamental physics. First detected in 1967 by Jocelyn Bell and Anthony Hewish these celestial objects possess a fascinating combination of characteristics, with a remarkably small radius of about 10 to 15 km (approximately $10^{-5} R_\odot$) and an immense mass ranging from $1 \sim 2 M_\odot$, making them incredibly dense. In fact, they rank as the second densest objects in the universe, surpassed only by black holes. True to their name, NSs primarily consist of neutrons, although they also contain a smaller proportion of protons, electrons, muons and possibly also other particles, such as hyperons, kaon or pion condensates, or deconfined quarks. The relative abundance of these particles is determined by the requirements of charge neutrality and beta stability, defined in Section 2.1.4. Some NSs possess an immensely powerful magnetic field, known as magnetars, with strengths reaching approximately $B \sim 10^{15}$ Gauss at the surface. Additionally, certain NSs exhibit rapid rotational periods in the millisecond range, earning them the name pulsars, the faster one to date spins at about 716Hz. They are considered stable objects due to the delicate equilibrium between gravity and the pressure generated by matter, stemming from the Pauli exclusion principle and the strong nuclear force, the fact that these stars are held by gravity is contrary to what happens in the nucleus.

2.1.1 How neutron stars are born?

If a star has a mass of approximately $\sim 8-30M_{\odot}$, normally called the red Supergiant, when it eventually dies, since they are more massive than most of the other stars, it has a very violent death as it explodes into a supernova of Type-II, Ib or Ic as is described in 2.1.

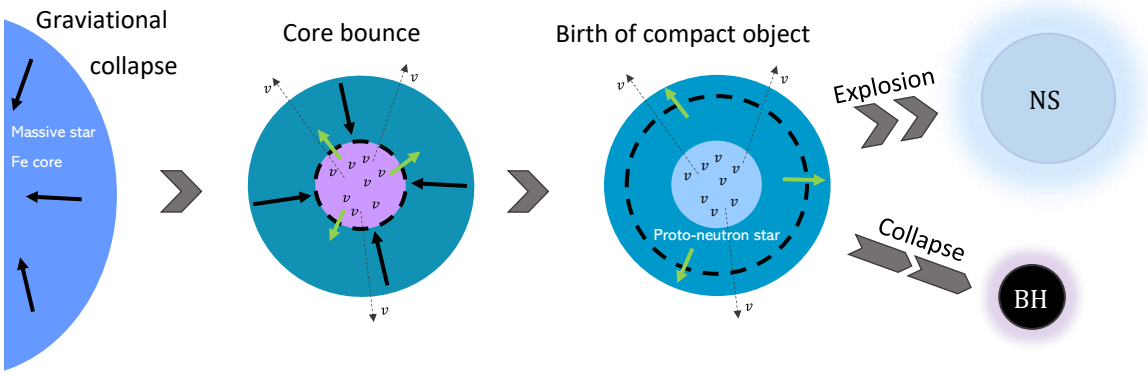


Figure 2.1: Visual representation showcasing the evolutionary progression of supernova cores, transforming from massive stars into compact objects, where NS stands for a Neutron Star remnant and BH for a Black Hole.

In essence, the process of a supernova commences when a star begins to fuse iron elements in its core. This leads to the gravitational collapse of the iron core, where the central region becomes highly compressed, effectively trapping neutrinos due to intense interactions within the extremely hot and dense matter. Subsequently, a shockwave is initiated as the central core rebounds off matter near the nuclear saturation density. If the shock wave propagates successfully by accreting the outer layers of the Fe-core, it leads to the supernova explosion and the birth of a NS; if it fails, it collapses into a black hole, a more detailed description can be found in [43].

Once the supernova explosion is over, the newly formed NS cools rapidly through the emission of neutrinos, where this mechanism is known as the direct Urca process, we will talk more about this process later. NSs can also be formed by the merger of two existing NSs. In such mergers, gravitational forces play a crucial role, and the collision can lead to the formation of either a black hole or a more massive NS.

2.1.2 The composition of highly compact stars

Since NSs are very dense objects, they can reach many times the nuclear saturation density, $n_0 = 0.16$ fm^{-3} , inside the star. They can therefore be divided into 5 main regions, considering the atmosphere, as shown in the scheme in Fig. 2.2. In each of these regions, they are described as:

- ➜ **Atmosphere** – It's a very thin region. Where the composition can be mainly hydrogen or heavier elements such as iron depending on when the star was formed.
- ➜ **Outer crust** – Solid region composed of heavy nuclei forming a coulomb lattice embedded in

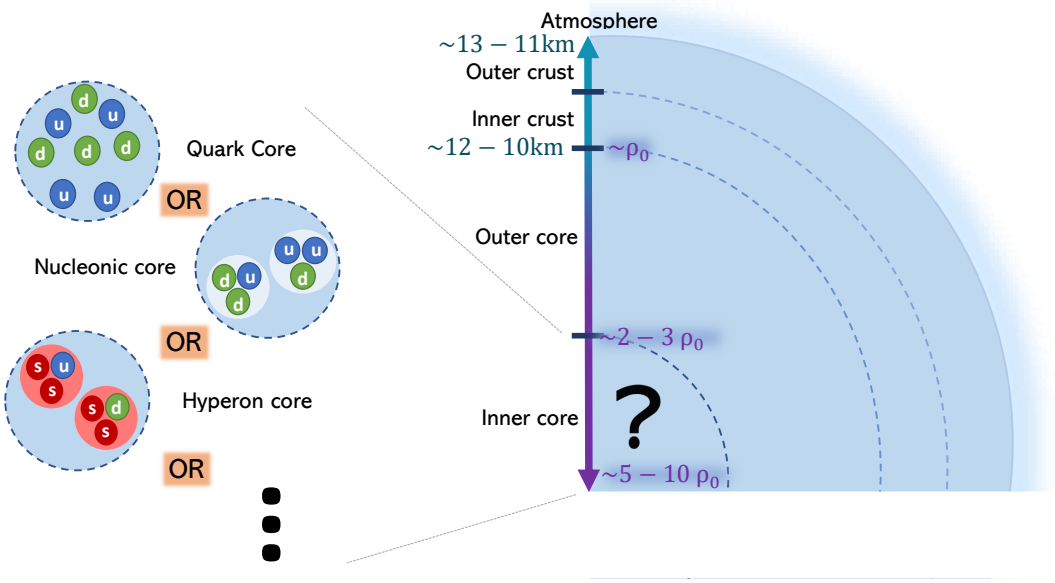


Figure 2.2: Illustration depicting the layered composition of NS, highlighting various potential configurations for the inner core matter.

a relativistic degenerate gas of electrons. The thickness of this layer is about several hundred meters. As the density increases, the energy of the electrons becomes high enough for inverse beta decay, leading to an increase in neutron-rich nuclei via electron capture. This continues until the neutron drip density is reached when neutrons begin to leave the nuclei.

➜ **Inner crust** – Highly neutron rich nuclei are immersed in a neutron gas in a superfluid state. The nuclear pasta phase is hypothesised to be in the end part of this region.

➜ **Outer core** – In this region of the star the composition of the star is mainly uniform nuclear matter mostly composed of neutrons and with a smaller portion of protons and electrons, to satisfy charge neutrality. When moving for the innermost layers at a certain density threshold muons μ star to appear. This matter defined frequently named $npe\mu$ matter, must also satisfy the condition of equilibrium with respect to weak interactions, namely beta equilibrium.

➜ **Inner core** – The innermost region of a NS remains one of the most mysterious and least understood regions. Scientists have proposed various possibilities for the composition of this region. Fig. 2.2 illustrates three potential scenarios: a quark core composed solely of quarks, a nucleonic core consisting of protons and neutrons, and a hyperon core with hadrons containing strange quarks. Additionally, also other propositions such as Kaon condensates, hybrid stars, dark matter and so on.

2.1.3 The connection to nuclear matter

Having understood the composition of NSs, significant parallels can be drawn between the matter within these compact objects and the idealized nuclear matter found in atomic nuclei. Extensive re-

search has been dedicated to comprehending the behavior of dense nuclear matter across a broad spectrum of densities and asymmetries. The EoS, which describes the relationship between pressure and energy density $P(\varepsilon)$, or pressure/energy density and density ($P(n)$ and $\varepsilon(n)$), plays a crucial role in characterizing such matter. Due to their nature as remarkably compact entities composed of baryons, NSs offer exceptional opportunities to constrain the EoS of nuclear matter across a vast range of densities. The extreme conditions within these celestial bodies provide valuable insights into the properties of nuclear matter under extreme densities that are otherwise unachievable in terrestrial laboratories.

What is nuclear matter ?

Nuclear matter is a theoretical concept used to represent a hypothetical form of strongly interacting matter. It is often characterized as an idealized system comprising an infinite and homogeneously symmetric distributed collection of nucleons, with infinite atomic mass number $A \to \infty$, where the effects of Coulomb interaction are disregarded. This simplification is beneficial as it reduces the influence of surface effects, which become negligible when considering an infinite radius approximation. The unique nature of NSs makes them particularly suitable for investigating nuclear matter. With typical radii of approximately 10 km and an approximate amount of number of nucleons $A = M_\odot / m_N \sim 10^{57}$, NSs offer a natural environment to study matter at extreme densities. In contrast, atomic nuclei have sizes on the order of 10^{-18} m, which is significantly smaller compared to the scale of NSs. Consequently, NSs provide a valuable opportunity to explore and study the properties of nuclear matter under conditions that are unattainable in terrestrial experiments.

What about the properties of nuclear matter?

The binding energy per nucleon, $\frac{B}{A}$, can be calculated through the energy density, $\varepsilon(n)$ as

$$\frac{B}{A} = \frac{\varepsilon}{n} - \bar{m}_N \,, \tag{2.1}$$

$\bar{m}_N = m_n\left(1-y_p\right) + m_p y_p \equiv m$, where $y_p = n_p/n$ is the proton fraction and m_n and m_p represent the mass of neutron and proton, respectively. In a crude approximation, this is considered to be the average of the nucleon mass. The minimum of these values is in the saturation density n_0.
The EoS can be decomposed to a good approximation into a symmetric and asymmetric part:

$$\frac{\varepsilon}{n}(n,\delta) - m \equiv \epsilon(n,\delta) \approx \epsilon_{SNM}(n) + E_{sym}(n)\delta^2 + \cdots \,, \tag{2.2}$$

ϵ is the binding energy per nucleon, $n = n_n + n_p$ is the baryon density that is the sum of neutron density n_n and proton density n_p, $\delta = \left(n_n - n_p\right)/n = \left(1-2y_p\right)$ is the isospin asymmetry parameter, $\epsilon_{SNM}(n) = \epsilon(n,\delta=0)$ represents the binding energy per nucleon of symmetric nuclear matter and $E_{sym}(n)$ is the symmetry energy, which basically represents the cost of how much energy is needed to convert symmetric nuclear matter into pure neutron matter

$$E_{sym}(n) = \left.\frac{\partial^2 \epsilon\,(n,\delta)}{2\partial\delta^2}\right|_{\delta=0} . \tag{2.3}$$

Expanding both terms $E_{sym}(n)$ and $\epsilon_{SNM}(n)$ for Eq. 2.2 in a Taylor series around the saturation density n_0 until fourth order and considering $x = \frac{n-n_0}{3n_0}$ it's obtained:

$$\epsilon(n,\delta) \approx \underbrace{\epsilon_0 + \frac{K_0}{2}x^2 + \frac{Q_0}{6}x^3 + \frac{Z_0}{4!}x^4}_{\epsilon_{SNM}(n)} + \underbrace{\left(J_{sym} + L_{sym}x + \frac{K_{sym}}{2}x^2 + \frac{Q_{sym}}{6}x^3 + \frac{Z_{sym}}{4!}x^4\right)}_{E_{sym}(n)}\delta^2 + \cdots . \tag{2.4}$$

The coefficient terms of the expansion are for the zero order $\epsilon_0 = \epsilon_{SNM}(n_0)$ the binding energy per particle at saturation density for symmetric nuclear matter, and $J_{sym} = E_{sym}(n_0)$, the symmetry energy at saturation density, the other terms for the symmetric nuclear matter come from:

$$X_0^{(k)} = (3n_0)^k \left.\frac{\partial^k \epsilon(n,0)}{\partial n^k}\right|_{n_0}, k = 2,3,4\,. \tag{2.5}$$

The parameters $X_0^2 = K_0$, $X_0^3 = Q_0$, and $X_0^4 = Z_0$ represent the incompressibility coefficient, skewness, and kurtosis, respectively, for symmetric nuclear matter. Regarding the remaining components within the symmetry energy expression,

$$X_{sym}^{(k)} = (3n_0)^k \left.\frac{\partial^k E_{sym}(n)}{\partial n^k}\right|_{n_0}, k = 1,2,3,4. \tag{2.6}$$

Where $X_{sym}^1 = L_{sym}$, $X_{sym}^2 = K_{sym}$, $X_{sym}^3 = Q_{sym}$, $X_{sym}^4 = Z_{sym}$, are the slope, curvature, skewness and kurtosis of symmetry energy at saturation density respectively. Exploring these quantities offers deeper insights into the behavior of nuclear matter. For a more detailed examination of these parameters, including asymmetric parameters in Eq. 2.2 with higher-order effects, see [44].

2.1.4 Neutron stars in β-equilibrium

Once we have grasped the properties of nuclear matter in the context of isospin asymmetric nuclei as observed in NSs, it is crucial to consider two essential conditions that the matter inside NSs must satisfy: charge neutrality and β-equilibrium. When considering the cooling of NSs, the dominant process is often the direct Urca process. This mechanism involves the rapid conversion of a neutron into a proton (β-decay) accompanied by the emission of an electron and an electron antineutrino and then the capture of electrons by protons (electron capture)

$$n \to p + e^- + \bar{\nu}_e\,, \quad p + e^- \to n + \nu_e\,.$$

However, due to momentum conservation, the direct Urca process, [45] can only occur when the proton fraction (the ratio of protons to baryons) exceeds a critical value. In cases where this condition is not met, a modified Urca process takes place instead. The modified Urca process involves the interaction between nucleons, resulting in a slower cooling mechanism.

Muons are only present when the Fermi energy of the electron exceeds the rest energy of the muons ($m_\mu c^2 \sim 105.7$MeV),

$$e^- \to \mu + \nu_e + \bar{\nu}_\mu\,.$$

Neutrinos and antineutrinos, which are the main cooling agents of the nascent proto-NS, generally escape the NS, and their chemical potentials can be effectively considered zero. The direct Urca process leads the nucleons into β-equilibrium. β-equilibrium arises from weak interactions within the NS. It dictates that the different species of particles, such as protons, neutrons, electrons, and muons, are in equilibrium with respect to weak decay and capture processes mediated by the weak force. In β-equilibrium, the rates of processes involving the conversion of neutrons into protons (β-decay) and the capture of electrons by protons (electron capture) are balanced, resulting in a stable equilibrium state, where the chemical potentials, defined as $\mu_j = \frac{\partial \varepsilon}{\partial n_j}$, thus satisfy for the β-decay:

$$\mu_n = \mu_p + \mu_e \,, \tag{2.7}$$

and for the unset of muons we have

$$\mu_\mu = \mu_e \,. \tag{2.8}$$

Furthermore, charge neutrality ensures that the overall charge of the NS matter, counting for protons, neutrons, electrons and muons [1] is balanced and neutral. This requires:

$$\sum_{b=p,n} q_b n_b + \sum_{l=e^-,\mu^-} q_l n_l = 0 \,, \tag{2.9}$$

$$n_p = n_e + n_\mu \,.$$

An intriguing study conducted by Ferreira *et al.*[35] employs deep NNs to establish a mapping between β-equilibrium NS matter and the corresponding properties of nuclear matter. This interplay between the properties of NS matter and nuclear matter is an avenue that promises valuable insights.

2.2 Microscopic neutron star properties

Having developed an understanding of the internal composition of NSs, the natural progression is to address the question: *how can we then obtain the EoS?* This involves utilizing a phenomenological approach. This could be non-relativistic, such as skyrme interactions, or it could take a relativistic approach, as seen in RMF models. Of these options, RMF models are often preferred due to their capability to account for the relativistic effects in dense matter scenarios. The forthcoming schematic illustration outlines the method employed to obtain the EoS:

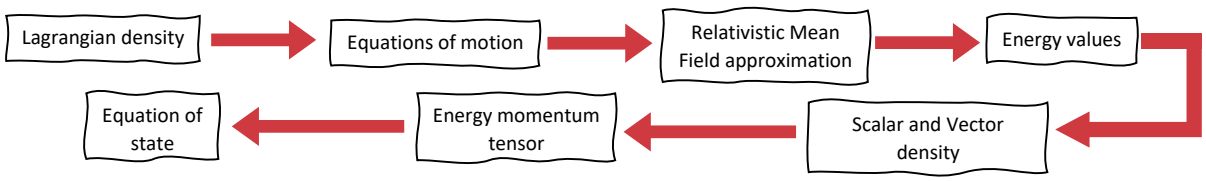


Figure 2.3: Illustration outlining the procedural steps involved in deriving the EoS.

[1] Specifications of the particles are in Table 1.

In this scheme, the first step is to define the Lagrangian density that governs the interactions of nucleons and mesons in nuclear matter. However, when one attempts to solve the equations of motion derived from the Lagrangian, they become highly complex and challenging to solve exactly. To make progress, the RMF approximation is applied. Under the RMF approximation, the energy spectrum of the system can be determined, allowing for the calculation of the vector and scalar densities. Once the values of the vector and scalar densities are obtained, the meson fields can be determined. With the values of the meson fields obtained, we can then calculate the EoS from the energy-momentum tensor, which relates to the RMF Lagrangian.

2.2.1 Lagrangian density

To describe the nuclear matter inside NSs, an advanced version of Walecka's original Quantum Hadrodynamics (QHD) model [46] is employed. This model postulates that nucleons interact with each other through the exchange of mesons. The Lagrangian of the model incorporates nucleonic fields Ψ, which interact via the exchange of three mesons: two isoscalar mesons, the scalar meson σ represented by a field φ, and the vector meson ω and one vector-isovector meson, the ρ meson represented by ϱ. The φ meson gives rise to the strong attractive force, while the ω mesons cause the strong repulsive force between the nucleons. The inclusion of the ϱ meson is also particularly important for accounting for the asymmetry of NSs, as it interacts differently with protons and neutrons due to their different isospin properties.

To account for β-equilibrium, free lepton fields Ψ_l, describing the electrons and the muons must also be added to the model, ensuring that NSs maintain charge neutrality. The particle specifications for the nucleons, mesons, and leptons involved in the model are summarized in Table 1. The Lagrangian density is then given by

$$\mathscr{L} = \mathscr{L}_N + \mathscr{L}_M + \mathscr{L}_{NL} + \mathscr{L}_{lep}, \tag{2.10}$$

where $\mathscr{L}_N$ describes the interaction of the nucleons through the mesons, $\mathscr{L}_M$ represents the self-interactions of the meson fields, $\mathscr{L}_{NL}$ represents the non-linear interactions and the cross interactions of the mesons and $\mathscr{L}_{lep}$ represents the leptons. Opening each of the defined Lagrangian densities gives

$$\begin{aligned}
\mathscr{L}_N =& \bar{\Psi}(x)\Big[\left(i\gamma_\mu\partial^\mu - g_\omega\gamma_\mu\omega^\mu(x) - g_\varrho\gamma_\mu \boldsymbol{t}\cdot\boldsymbol{\varrho}^\mu(x)\right) - \left(m - g_\varphi\varphi(x)\right)\Big]\Psi(x)\,,\\
\mathscr{L}_M =& \frac{1}{2}\left[\partial_\mu\varphi(x)\partial^\mu\varphi(x) - m_\varphi^2\varphi^2(x)\right] - \frac{1}{4}F^{(\omega)}_{\mu\nu}(x)F^{(\omega)\mu\nu}(x) + \frac{1}{2}m_\omega^2\omega_\mu(x)\omega^\mu(x) -\\
& \frac{1}{4}\boldsymbol{F}^{(\varrho)}_{\mu\nu}(x)\cdot\boldsymbol{F}^{(\varrho)\mu\nu}(x) + \frac{1}{2}m_\varrho^2\boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x)\,,\\
\mathscr{L}_{NL} =& -\frac{1}{3}b\,m\,g_\varphi^3\varphi^3(x) - \frac{1}{4}cg_\varphi^4\varphi^4(x) + \frac{\xi}{4!}g_\omega^4(\omega_\mu(x)\omega^\mu(x))^2 + \Lambda_\omega g_\varrho^2\boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x)g_\omega^2\omega_\mu(x)\omega^\mu(x)\,,\\
\mathscr{L}_{lep} =& \sum_l \bar{\Psi}_l(x)\left(i\gamma_\mu\partial^\mu - m_l\right)\Psi_l(x)\,.
\end{aligned}$$

The masses of the meson fields are defined as m_i where $i = \varphi,\omega,\varrho$, additionally, the vacuum masses for the proton and neutron have an average value m, and m_l represents the mass of each

lepton. The vector meson tensors are defined as $F^{(\omega)}_{\mu\nu}(x) = \partial_\mu \omega_\nu(x) - \partial_\nu \omega_\mu(x)$ and $\boldsymbol{F}^{(\varrho)}_{\mu\nu}(x) = \partial_\mu \boldsymbol{\varrho}_\nu(x) - \partial_\nu \boldsymbol{\varrho}_\mu(x)$[2]. The isovector meson has three isospin components $\boldsymbol{\varrho}^\mu = (\varrho^\mu_1, \varrho^\mu_2, \varrho^\mu_3)$ with respective charges, $\varrho^\mu_q(q = \pm 1, 0)$ represented as $\varrho^\mu_0 = \varrho^\mu_3$ and $\varrho^\mu_\pm = \frac{1}{\sqrt{2}}\left(\varrho^\mu_1 \pm i\varrho^\mu_2\right)$. The operator $\boldsymbol{t}$ represents the isospin operator. We are considering Ψ as a 8×1 spinor since it describes the nucleon doublet, while Ψ_l is a 4×1 spinor [3]. The g_ω, g_ϱ and g_φ are the coupling constants, which represent how strongly the nucleons interact with the mesons, they can also be considered density dependent as discussed in [47]. J. Boguta and A.R. Bodmer proposed an extension to the QHD Lagrangian density, which includes self-couplings of the scalar meson field σ. This is achieved by introducing terms such as φ^3 and φ^4 into the Lagrangian, with couplings b and c respectively (which came from [48]), allowing for additional interactions involving the scalar meson. Similarly, a self-interaction term for the vector meson ω, with coupling ξ term, introduced in [49], is included which is mainly responsible for how soft is the EoS at high densities. This addition provides a valuable tool to control the softness of the high-density EoS. For the Λ_ω coupling is characterizing the interaction between the vector meson fields ω and ϱ interfering with the density dependence of the symmetry energy.

2.2.2 The equations of motion

Once the Lagrangian density is defined, one can derive a set of field equations by using the Euler-Lagrange equation:

$$\partial_\mu \frac{\partial \mathscr{L}}{\partial(\partial_\mu \phi(x))} = \frac{\partial \mathscr{L}}{\partial \phi(x)}\,, \tag{2.11}$$

where ϕ stands for the different fields involved in the theory. All the details about the calculations can be found in Appendix A. The fermions are described by the Dirac equation of the form:

$$\left[i\gamma^\mu \partial_\mu - m_l\right]\Psi_l(x) = 0\,, \tag{2.12}$$

$$\left[\gamma^\mu\left(i\partial_\mu - g_\omega \omega_\mu(x) - g_\varrho \boldsymbol{t}\cdot\boldsymbol{\varrho}_\mu(x)\right) - \left(m - g_\varphi \varphi(x)\right)\right]\Psi(x) = 0\,. \tag{2.13}$$

For the meson filed with spin 0, it's described by the Klein-Gordon equation

$$\partial_\mu \partial^\mu \varphi(x) + m^2_\varphi \varphi(x) + b m g^3_\varphi \varphi^2(x) + c g^4_\varphi \varphi^3(x) = g_\varphi \bar{\Psi}(x)\Psi(x)\,. \tag{2.14}$$

For the meson fields with spin 1 they are described by the Proca equations:

$$\partial_a \boldsymbol{F}^{(\varrho)ab}(x) + 2\Lambda_\omega g^2_\varrho \boldsymbol{\varrho}^b(x) g^2_\omega \omega_\mu(x)\omega^\mu(x) + m^2_\varrho \boldsymbol{\varrho}^b(x) = \bar{\Psi}(x)\gamma^b g_\varrho \boldsymbol{t}\Psi(x)\,, \tag{2.15}$$

$$\partial_a F^{(\omega)ab}(x) + m^2_\omega \omega^b(x) + \frac{\xi}{3!} g^4_\omega(\omega_\mu(x)\omega^\mu(x))\omega^b(x) + 2\Lambda_\omega g^2_\varrho \boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x) g^2_\omega \omega^b(x) = g_\omega \bar{\Psi}(x)\gamma^b\Psi(x)\,. \tag{2.16}$$

[2] Here for the rho meson it should be $\boldsymbol{F}^{(\varrho)}_{\mu\nu}(x) = \partial_\mu \boldsymbol{\varrho}_\nu(x) - \partial_\nu \boldsymbol{\varrho}_\mu(x) - g_\varrho\left(\boldsymbol{\varrho}_\mu(x) \times \boldsymbol{\varrho}_\nu(x)\right)$ however since we are implementing the RMF approximation the cross product can be ignored.

[3] While γ^μ is a 8×8 matrix ($\gamma^\mu \otimes \mathbb{1}_{2\times 2}$) for Ψ, for Ψ_l is a 4×4 matrix but for the sake of simplicity we use the same notation.

These field equations are non-linear coupled equations making them challenging to solve exactly. However, to facilitate computations, these equations can be simplified by employing the RMF approximation.

2.2.3 Relativistic mean field approximation

The RMF approximation considers a system in its ground state, characterized by static and uniform matter. Within this approximation, the fields and sources lose their space-time dependence and become constant. This is justified by assuming translational invariance in infinite nuclear matter. For the vector mesons, the spacial components vanish due to rotational symmetry. Also for the isovector meson, ϱ, only the third component of isospin has a nonvanishing value because of charge conservation, more explanation can be found in [43].

$$\begin{aligned}
\varphi(x) \to \langle\varphi\rangle \to \langle\varphi\rangle \equiv \varphi \quad &\text{and} \quad \bar{\Psi}(x)\Psi(x) \to \langle\bar{\Psi}\Psi\rangle = \langle\bar{\Psi}\Psi\rangle\,, \\
\omega_\mu(x) \to \langle\omega_\mu\rangle \to \langle\omega_0\rangle \equiv \omega_0 \quad &\text{and} \quad \bar{\Psi}(x)\gamma^\mu\Psi(x) \to \langle\bar{\Psi}\gamma^0\Psi\rangle = \langle\Psi^\dagger\Psi\rangle\,, \\
\boldsymbol{\varrho}_\mu(x) \to \langle\boldsymbol{\varrho}_\mu\rangle \to \langle\varrho_{03}\rangle \equiv \varrho_{03} \quad &\text{and} \quad \bar{\Psi}(x)\gamma^\mu\boldsymbol{t}\Psi(x) \to \langle\bar{\Psi}\gamma^0 t_3\Psi\rangle = \langle\Psi^\dagger t_3\Psi\rangle\,.
\end{aligned}$$

The equations of motion in the RMF approximation for the mesons then become:

$$2\Lambda_\omega g_\varrho^2\varrho_{03}g_\omega^2\omega_0^2 + m_\varrho^2\varrho_{03} = g_\varrho\langle\Psi^\dagger t_3\Psi\rangle\,, \tag{2.17}$$

$$m_\omega^2\omega_0 + \frac{\xi}{3!}g_\omega^4(\omega_0)^3 + 2\Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0 = g_\omega\langle\Psi^\dagger\Psi\rangle\,, \tag{2.18}$$

$$m_\varphi^2\varphi + bmg_\varphi^3\varphi^2 + cg_\varphi^4\varphi^3 = g_\varphi\langle\bar{\Psi}\Psi\rangle\,. \tag{2.19}$$

The Lagrangian density reduces its form to:

$$\mathscr{L}_{RMF} = -\frac{1}{2}m_\varphi^2\varphi^2 - \frac{1}{3}b\,m\,g_\varphi^3\varphi^3 - \frac{1}{4}cg_\varphi^4\varphi^4 + \frac{\xi}{4!}g_\omega^4\omega_0^4 + \frac{1}{2}m_\omega^2\omega_0^2 + \frac{1}{2}m_\varrho^2\varrho_{03}^2 + \Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0^2\,. \tag{2.20}$$

2.2.4 The energy values

In order to compute the energy spectrum of the Dirac fields, we describe the fields as plane waves given by:

$$\Psi(x) = \Psi(k)e^{-ik\cdot x}\,. \tag{2.21}$$

Substituting this into Eq. 2.13 yields

$$\left[\gamma_\mu k^\mu - m + g_\varphi\varphi(x) + g_\sigma\gamma_\mu\omega^\mu(x) - g_\varrho\gamma_\mu\boldsymbol{t}\cdot\boldsymbol{\varrho}^\mu(x)\right]\Psi(k) = 0\,. \tag{2.22}$$

Applying the methodology used in [43], we introduce the following substitutions:

$$K^\mu = k^\mu - g_\sigma\omega^\mu - g_\varrho\boldsymbol{t}\cdot\boldsymbol{\varrho}^\mu\,, \tag{2.23}$$

$$m^* = m - g_\varphi\varphi(x)\,. \tag{2.24}$$

Here m^* is the effective mass. With these substitutions, Eq. 2.22 can be rewritten as:

$$\left[\gamma_\mu K^\mu - m^*\right]\Psi(K) = 0\,. \tag{2.25}$$

Rationalizing the Dirac operator by multiplying Eq. 2.25 with $(\gamma_\mu K^\mu + m^*)$ gives:

$$\begin{aligned}
\left(\gamma_\mu K^\mu + m^*\right)\left(\gamma_\mu K^\mu - m^*\right) &= \gamma_\mu K^\mu \gamma_\nu K^\nu - m^*\gamma_\nu K^\nu + m^*\gamma_\mu K^\mu - m^{*2}\,, \\
&= \gamma_\mu K^\mu \gamma_\nu K^\nu - m^{*2}\,, \\
&= \frac{\gamma_\mu K^\mu \gamma_\nu K^\nu + \gamma_\mu K^\mu \gamma_\nu K^\nu}{2} - m^{*2}\,, \\
&= K^\mu K^\nu \frac{\gamma_\mu\gamma_\nu + \gamma_\mu\gamma_\nu}{2} - m^{*2}\,, \\
&= K^\mu K^\nu \frac{2g_{\mu\nu}}{2} - m^{*2}\,, \\
&= K^\mu K_\mu - m^{*2}\,.
\end{aligned} \tag{2.26}$$

Where the properties of the gamma matrices $\{\gamma_\mu, \gamma_\nu\} = 2g_{\mu\nu}$ were utilized. This leads to the rewriting of Eq. 2.25 as:

$$\left[K^\mu K_\mu - m^{*2}\right]\Psi(K) = 0\,. \tag{2.27}$$

As a result, given that the operator now represents a number that, when multiplied by $\Psi(K)$, yields zero, this numerical value must be zero as well. Consequently,

$$\begin{aligned}
K^\mu K_\mu - m^{*2} &= 0\,, \\
K_0^2 - K_i^2 - m^{*2} &= 0\,, \\
K_0^2 &= K_i^2 + m^{*2}\,.
\end{aligned} \tag{2.28}$$

Since within the RMF approximation only the terms φ, ω_0, and ϱ_{03} remain, Eq.2.28 becomes:

$$\begin{aligned}
\left(k_0 - g_\omega\omega_0 - g_\varrho I_{3b}\varrho_{03}\right)^2 - k_i^2 - m^{*2} &= 0\,, \\
\left(k_0 - g_\omega\omega_0 - g_\varrho I_{3b}\varrho_{03}\right)^2 &= k_i^2 + m^{*2}\,, \\
k_0 &= g_\omega\omega_0 + g_\varrho I_{3b}\varrho_{03} \pm \sqrt{k_i^2 + m^{*2}}\,.
\end{aligned} \tag{2.29}$$

This enables us to express the energy values as

$$E_b(\boldsymbol{k}) = g_\omega\omega_0 + g_\varrho I_{3b}\varrho_{03} + \sqrt{\boldsymbol{k}^2 + m^{*2}} \quad , \quad b = n, p\,. \tag{2.30}$$

So the energy values for the particle and antiparticle are respectively:

$$E_b(\boldsymbol{k}) = g_\omega\omega_0 + g_\varrho I_{3b}\varrho_{03} + \sqrt{\boldsymbol{k}^2 + m^{*2}}\,, \tag{2.31}$$

$$\bar{E}_b(\boldsymbol{k}) = -g_\omega\omega_0 - g_\varrho \bar{I}_{3b}\varrho_{03} + \sqrt{\boldsymbol{k}^2 + m^{*2}} \quad , \quad b = n, p\,. \tag{2.32}$$

For the leptons, by using the same procedure we obtain

$$E_l(\boldsymbol{k}) = \sqrt{\boldsymbol{k}^2 + m_l^{*2}} \quad , \quad l = e^-, \mu^-\,. \tag{2.33}$$

2.2.5 Scalar and vector density

In order to calculate the expectation values of the sources for the baryon fields, Eqs. 2.17, 2.18 and 2.19, is going to be used the method introduced in [43]. This basically consists of when the expectation value of an operator Γ with respect to the ground state in the many-body system can be written as the continuum summation of the single-particle expectation value, $\left(\bar{\Psi}\Gamma\Psi\right)_{\boldsymbol{k},k}$, $\boldsymbol{k}$ is the momentum and k is the spin-isospin state of the single-particle

$$\langle\bar{\Psi}\Gamma\Psi\rangle = \sum_k \int \frac{d\boldsymbol{k}}{(2\pi)^3} \left(\bar{\Psi}\Gamma\Psi\right)_{\boldsymbol{k},k} \Theta\left(\mu_b - E(\boldsymbol{k})\right) . \tag{2.34}$$

The sum over k is the sum over the spin states of the occupied momentum states, μ_b is the chemical potential, and $\Theta\left(\mu_b - E(\boldsymbol{k})\right)$ is a step function defined as:

$$\Theta\left(\mu_b - E(\boldsymbol{k})\right) = \begin{cases} 1 & \text{if } |\boldsymbol{k}| \leq k_{F,b} \\ 0 & \text{if } |\boldsymbol{k}| > k_{F,b} \end{cases} .$$

The main goal is that all the information about Γ is found in the Baryon Dirac Hamilton, H_D, so in order to obtain it we need to find k_0 present in Eq. 2.13

$$\begin{aligned}
\left[\gamma_\mu(k^\mu - g_\omega\omega^\mu - g_\varrho \boldsymbol{t}\cdot\boldsymbol{\varrho}^\mu) - m^*\right]\Psi(\boldsymbol{k}) &= 0 , \\
\left[\gamma_0(k^0 - g_\omega\omega^0 - g_\varrho t_3\varrho_3^0) - \gamma_i k^i - m^*\right]\Psi(\boldsymbol{k}) &= 0 , \\
\left[\gamma_0(g_\omega\omega^0 + g_\varrho t_3\varrho_3^0) + \gamma_i k^i + m^*\right]\Psi(\boldsymbol{k}) &= \gamma_0 k^0\Psi(\boldsymbol{k}) , \\
\gamma_0\left[\gamma_0(g_\omega\omega^0 + g_\varrho t_3\varrho_3^0) + \gamma_i k^i + m^*\right]\Psi(\boldsymbol{k}) &= \gamma_0\gamma_0 k^0\Psi(\boldsymbol{k}) , \\
\left[g_\omega\omega^0 + g_\varrho t_3\varrho_3^0 + \gamma_0(\gamma_i k^i + m^*)\right]\Psi(\boldsymbol{k}) &= k^0\Psi(\boldsymbol{k}) , \\
H_D\Psi(\boldsymbol{k}) &= E(\boldsymbol{k})\Psi(\boldsymbol{k}) .
\end{aligned} \tag{2.35}$$

It's then possible to obtain the Hamiltonian

$$H_D = g_\omega\omega^0 + g_\varrho t_3\varrho_3^0 + \alpha_i k^i + \beta m^* , \tag{2.36}$$

where the terms α_i and β are defined in the Chapter Notation. Now, let's proceed to calculate the expected value of the Dirac Hamiltonian for a single-particle momentum state:

$$\left(\Psi^\dagger H_D\Psi\right)_{\boldsymbol{k},k} = \left(\Psi^\dagger E(\boldsymbol{k})\Psi\right)_{\boldsymbol{k},k} = E(\boldsymbol{k})\left(\Psi^\dagger\Psi\right)_{\boldsymbol{k},k} . \tag{2.37}$$

We first analyze the derivative of the expected value of the Dirac Hamiltonian with respect to a ran-

dom variable ζ:

$$\begin{aligned}
\frac{\partial}{\partial\zeta}\left(\Psi^\dagger H_D\Psi\right)_{\boldsymbol{k},k} &= \left(\frac{\partial\Psi^\dagger}{\partial\zeta}H_D\Psi\right)_{\boldsymbol{k},k} + \left(\Psi^\dagger\frac{\partial H_D}{\partial\zeta}\Psi\right)_{\boldsymbol{k},k} + \left(\Psi^\dagger H_D\frac{\partial\Psi}{\partial\zeta}\right)_{\boldsymbol{k},k}, \\
&= \left(\Psi^\dagger\frac{\partial H_D}{\partial\zeta}\Psi\right)_{\boldsymbol{k},k} + E(\boldsymbol{k})\left(\frac{\partial\Psi^\dagger}{\partial\zeta}\Psi\right)_{\boldsymbol{k},k} + E(\boldsymbol{k})\left(\Psi^\dagger\frac{\partial\Psi}{\partial\zeta}\right)_{\boldsymbol{k},k}, \\
&= \left(\Psi^\dagger\frac{\partial H_D}{\partial\zeta}\Psi\right)_{\boldsymbol{k},k} + E(\boldsymbol{k})\frac{\partial}{\partial\zeta}\left(\Psi^\dagger\Psi\right)_{\boldsymbol{k},k}, \\
&= \left(\Psi^\dagger\frac{\partial H_D}{\partial\zeta}\Psi\right)_{\boldsymbol{k},k} + \underbrace{E(\boldsymbol{k})\frac{\partial}{\partial\zeta}\left(\Psi^\dagger\Psi\right)_{\boldsymbol{k},k}}_{=0}, \\
&= \left(\Psi^\dagger\frac{\partial H_D}{\partial\zeta}\Psi\right)_{\boldsymbol{k},k}. \qquad (2.38)
\end{aligned}$$

In the last step, the second term went to zero since $\Psi(k)$ is an eigenfunction. Combining this result with Eq.2.37, we obtain:

$$\left(\Psi^\dagger\frac{\partial H_D}{\partial\zeta}\Psi\right)_{\boldsymbol{k},k} = \frac{\partial}{\partial\zeta}E(\boldsymbol{k})\,. \qquad (2.39)$$

Now in order to obtain the normalization condition of the wave function it's going to be used Eq.2.39 where the derivative is going to be taken with respect to ω_0 of Eq.2.36

$$\begin{aligned}
\left(\Psi^\dagger\frac{\partial H_D}{\partial\omega_0}\Psi\right)_{\boldsymbol{k},k} &= \frac{\partial}{\partial\omega_0}E(\boldsymbol{k})\,, \\
g_\omega\left(\gamma_0\right)^2\left(\Psi^\dagger\Psi\right)_{\boldsymbol{k},k} &= \frac{\partial}{\partial\omega_0}\left(g_\omega\omega_0+\sqrt{k^2+m^{*2}}\right), \\
g_\omega\left(\Psi^\dagger\Psi\right)_{\boldsymbol{k},k} &= g_\omega\,.
\end{aligned}$$

It was used the condition $\left(\gamma_0\right)^2 = I$. From this we are able to see that for the condition to be satisfied $\left(\Psi^\dagger\Psi\right)_{\boldsymbol{k},k} = 1$. Having obtained the normalization condition we are now able to calculate the vector baryon density, $\langle\Psi^\dagger\Psi\rangle$, using Eq.2.34 :

$$\begin{aligned}
\langle\Psi^\dagger\Psi\rangle &= \sum_k\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\Psi_p^\dagger\Psi_p\right)_{\boldsymbol{k},k}\Theta\left(\mu_p-E_p(\boldsymbol{k})\right) + \sum_k\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\Psi_n^\dagger\Psi_n\right)_{\boldsymbol{k},k}\Theta\left(\mu_n-E_n(\boldsymbol{k})\right), \\
&= (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\Psi_p^\dagger\Psi_p\right)_{\boldsymbol{k},k}\Theta\left(\mu_p-E_p(\boldsymbol{k})\right) + (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\Psi_n^\dagger\Psi_n\right)_{\boldsymbol{k},k}\Theta\left(\mu_n-E_n(\boldsymbol{k})\right), \\
&= (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\Theta\left(\mu_p-E_p(\boldsymbol{k})\right) + (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\Theta\left(\mu_n-E_n(\boldsymbol{k})\right), \\
&= (2J_B+1)\int_0^{k_{F,p}}\frac{4\pi k^2dk}{(2\pi)^3} + (2J_B+1)\int_0^{k_{F,n}}\frac{4\pi k^2dk}{(2\pi)^3}, \\
&= \frac{k_{F,p}^3}{3\pi^2}+\frac{k_{F,n}^3}{3\pi^2}, \\
&= n_p+n_n = n\,. \qquad (2.40)
\end{aligned}$$

The Fermi momenta for protons and neutrons denoted as $k_{F,p}$ and $k_{F,n}$ respectively, are indicative of the momentum at the Fermi surface for each nucleon type. These values are directly linked to the vector density of the respective particles. The factor $(2J_B+1)$ represents the particle's degeneracy,

where J_B signifies its spin quantum number. The final outcome deduced from Eq. 2.40 corresponds to the baryonic density, symbolized as n. To derive $\langle\bar{\Psi}\Psi\rangle$, it is essential to initially compute the expected value for a single particle, using Eq. 2.39:

$$\left(\bar{\Psi}_p\Psi_p\right)_{\boldsymbol{k},k} = \left(\Psi_p^\dagger\gamma_0\Psi_p\right)_{\boldsymbol{k},k} = \left(\Psi_p^\dagger\frac{\partial H_D}{\partial m^*}\Psi_p\right)_{\boldsymbol{k},k} = \frac{\partial}{\partial m^*}E_p(k) = \frac{m^*}{\sqrt{k^2+m^{*2}}}\,, \tag{2.41}$$

$$\left(\bar{\Psi}_n\Psi_n\right)_{\boldsymbol{k},k} = \left(\Psi_n^\dagger\gamma_0\Psi_n\right)_{\boldsymbol{k},k} = \left(\Psi_n^\dagger\frac{\partial H_D}{\partial m^*}\Psi_n\right)_{\boldsymbol{k},k} = \frac{\partial}{\partial m^*}E_n(k) = \frac{m^*}{\sqrt{k^2+m^{*2}}}\,. \tag{2.42}$$

Subsequently, employing Eqs. 2.41 and 2.34:

$$\begin{aligned}
\langle\bar{\Psi}\Psi\rangle &= \sum_k\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\bar{\Psi}_p\Psi_p\right)_{\boldsymbol{k},k}\Theta\left(\mu_p - E_p(\boldsymbol{k})\right) + \sum_k\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\bar{\Psi}_n\Psi_n\right)_{\boldsymbol{k},k}\Theta\left(\mu_n - E_n(\boldsymbol{k})\right)\,,\\
&= (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\bar{\Psi}_p\Psi_p\right)_{\boldsymbol{k},k}\Theta\left(\mu_p - E_p(\boldsymbol{k})\right) + (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\bar{\Psi}_n\Psi_n\right)_{\boldsymbol{k},k}\Theta\left(\mu_n - E_n(\boldsymbol{k})\right)\,,\\
&= \frac{1}{\pi^2}\int_0^{k_{F,p}} dk\frac{k^2m^*}{\sqrt{k^2+m^*}} + \frac{1}{\pi^2}\int_0^{k_{F,n}} dk\frac{k^2m^*}{\sqrt{k^2+m^*}}\,,\\
&= n_p^s + n_n^s = n^s\,.
\end{aligned} \tag{2.43}$$

Here n^s stands for scalar baryonic density. Then calculating for $\langle\Psi^\dagger t_3\Psi\rangle$:

$$\begin{aligned}
\langle\Psi^\dagger t_3\Psi\rangle &= \sum_k\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\Psi_p^\dagger t_3\Psi_p\right)_{\boldsymbol{k},k}\Theta\left(\mu_p - E_p(\boldsymbol{k})\right) + \sum_k\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\Psi_n^\dagger t_3\Psi_n\right)_{\boldsymbol{k},k}\Theta\left(\mu_n - E_n(\boldsymbol{k})\right)\,,\\
&= (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\Psi_p^\dagger t_3\Psi_p\right)_{\boldsymbol{k},k}\Theta\left(\mu_p - E_p(\boldsymbol{k})\right) + (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\left(\Psi_n^\dagger t_3\Psi_n\right)_{\boldsymbol{k},k}\Theta\left(\mu_n - E_n(\boldsymbol{k})\right)\,,\\
&= (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}I_{3,p}\left(\Psi_p^\dagger\Psi_p\right)_{\boldsymbol{k},k} + (2J_B+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}I_{3,n}\left(\Psi_n^\dagger\Psi_n\right)_{\boldsymbol{k},k}\,,\\
&= \frac{I_{3,p}k_{F,p}^3}{3\pi^2} + \frac{I_{3,n}k_{F,n}^3}{3\pi^2} = I_{3,p}n_p + I_{3,n}n_n\,.
\end{aligned} \tag{2.44}$$

In this context, $I_{3,(p,n)}$ denotes the isospin of the proton and neutron, which are $\frac{1}{2}$ and $-\frac{1}{2}$ respectively. Consequently, the equations of motion can now be expressed as:

$$m_\varphi^2\varphi + bmg_\varphi^3\varphi^2 + cg_\varphi^4\varphi^3 = g_\varphi\left(n_p^s + n_n^s\right)\,, \tag{2.45}$$

$$m_\omega^2\omega_0 + \frac{\xi}{3!}g_\omega^4(\omega_0)^3 + 2\Lambda_\omega g_\varrho^2\varrho_{03}^2g_\omega^2\omega_0 = g_\omega\left(n_p + n_n\right)\,, \tag{2.46}$$

$$2\Lambda_\omega g_\varrho^2\varrho_{03}g_\omega^2\omega_0^2 + m_\varrho^2\varrho_{03} = g_\varrho\left(I_{3,p}n_p + I_{3,n}n_n\right)\,. \tag{2.47}$$

2.2.6 Equation of state

In order to obtain the EoS, the energy-momentum tensor $T^{\mu\nu}$ comes into play, which is defined as :

$$T^{\mu\nu} = \left(\frac{\partial\mathscr{L}}{\partial\left(\partial_\mu\phi_i(x)\right)}\right)\partial^\nu\phi_i(x) - g^{\mu\nu}\mathscr{L}\,, \tag{2.48}$$

where $g^{\mu\nu}$ is the metric tensor and ϕ_i stands for the each field of the Lagrangian.

For an ideal fluid, characterized by the absence of shear forces and heat flux, the off-diagonal elements are null. Assuming isotropy in matter, the spatial diagonal elements in the rest frame are equal, implying equal pressure across the three directions. In this context, the energy-momentum tensor for an ideal fluid at rest takes the form

$$T^{\mu\nu} = \begin{bmatrix} \epsilon & 0 & 0 & 0 \\ 0 & P & 0 & 0 \\ 0 & 0 & P & 0 \\ 0 & 0 & 0 & P \end{bmatrix} . \tag{2.49}$$

Here, ϵ represents the energy density of the system and P is the pressure. So in order to obtain the energy density and pressure, we calculated the expected value of the energy tensor:

$$\begin{aligned} \varepsilon = \langle T^{00} \rangle &= \sum_{j=b,l} \left\langle \left(\frac{\partial \mathscr{L}}{\partial \left(\partial_0 \Psi_j \right)} \right) \partial^0 \Psi_j \right\rangle - \langle g^{00} \mathscr{L} \rangle \, , \\ &= \sum_{j=b,l} \langle \bar{\Psi}_j \gamma_0 i \partial^0 \Psi_j \rangle - \langle \mathscr{L} \rangle \, , \\ &= \sum_{j=b,l} \langle \bar{\Psi}_j \gamma_0 k^0 \Psi_j \rangle - \mathscr{L}_{RMF} \, . \end{aligned} \tag{2.50}$$

The term $\mathscr{L}_{RMF}$ has already been computed in Eq.2.20, yet $\langle \bar{\Psi} \gamma_0 k^0 \Psi \rangle$ is missing. Using the relation present in Eq.2.37, the following is obtained:

$$\begin{aligned} \sum_{j=b,l} \langle \bar{\Psi}_j \gamma_0 k^0 \Psi_j \rangle &= \sum_{j=b,l} \sum_{k_j} \int \frac{d\boldsymbol{k}}{(2\pi)^3} \left(\Psi_j^\dagger k_0 \Psi_j \right)_{\boldsymbol{k},k_j} \Theta\left(\mu_j - E_j(\boldsymbol{k}) \right), \quad b = p,n \, , \quad l = e^-, \mu^- \, , \\ &= \sum_{j=p,n} (2J_j+1) \int_0^{k_{F,j}} \frac{d\boldsymbol{k}}{(2\pi)^3} \left(g_\omega \omega_0 + g_\varrho I_{3j} \varrho_{03} + \sqrt{\boldsymbol{k}^2 + m_j^{*2}} \right) \\ &\quad + \sum_{j=e^-,\mu^-} (2J_j+1) \int_0^{k_{F,j}} \frac{d\boldsymbol{k}}{(2\pi)^3} \left(\sqrt{\boldsymbol{k}^2 + m_j^2} \right) , \\ &= \sum_{j=p,n} \left[g_\omega \omega_0 n_j + g_\varrho \varrho_{03} I_{3j} n_j + \frac{(2J_j+1)}{2\pi^2} \int_0^{k_{F,j}} \left(k^2 \sqrt{k^2 + m_j^{*2}} \right) dk \right] \\ &\quad + \sum_{j=e^-,\mu^-} \frac{(2J_j+1)}{2\pi^2} \int_0^{k_{F,j}} dk \left(k^2 \sqrt{k^2 + m_j^2} \right) , \\ &= \sum_{j=p,n} \left[\frac{(2J_j+1)}{2\pi^2} \int_0^{k_{F,j}} \left(k^2 \sqrt{k^2 + m_j^{*2}} \right) dk \right] + m_\omega^2 \omega_0^2 + \frac{\xi}{3!} g_\omega^4 \omega_0^4 + 2\Lambda_\omega g_\varrho^2 \varrho_{03}^2 g_\omega^2 \omega_0^2 \\ &\quad + m_\varrho^2 \varrho_{03}^2 + 2\Lambda_\omega g_\varrho^2 \varrho_{03}^2 g_\omega^2 \omega_0^2 + \sum_{j=e^-,\mu^-} \left[\frac{(2J_j+1)}{2\pi^2} \int_0^{k_{F,j}} dk \left(k^2 \sqrt{k^2 + m_j^2} \right) \right] . \end{aligned} \tag{2.51}$$

In the last step the equations of motion present in Eqs. 2.46 and 2.47 were used. The leptons' degeneracy is the same as for the protons and neutrons, and the isospin of both leptons is $-1/2$ as described in Table 1. Consequently, the energy density is given by:

$$
\begin{aligned}
\varepsilon = & \sum_{j=p,n}\left[\frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}}\left(k^2\sqrt{k^2+m_j^{*2}}\right)dk\right] + m_\omega^2\omega_0^2 + \frac{\xi}{3!}g_\omega^4\omega_0^4 + 2\Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0^2 \\
& + m_\varrho^2\varrho_{03}^2 + 2\Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0^2 + \sum_{j=e^-,\mu^-}\left[\frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}} dk\left(k^2\sqrt{k^2+m_j^2}\right)\right] \\
& - \left[-\frac{1}{2}m_\varphi^2\varphi^2 - \frac{1}{3}b\ m\ g_\varphi^3\varphi^3 - \frac{1}{4}c g_\varphi^4\varphi^4 + \frac{\xi}{4!}g_\omega^4\omega_0^4 + \frac{1}{2}m_\omega^2\omega_0^2 + \frac{1}{2}m_\varrho^2\varrho_{03}^2 + \Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0^2\right], \\
= & \sum_{j=p,n}\left[\frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}}\left(k^2\sqrt{k^2+m_j^{*2}}\right)dk\right] + \sum_{j=e^-,\mu^-}\left[\frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}} dk\left(k^2\sqrt{k^2+m_j^2}\right)\right] \\
& + \frac{1}{2}m_\omega^2\omega_0^2 + \frac{\xi}{8}g_\omega^4\omega_0^4 + 3\Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0^2 + \frac{1}{2}m_\varrho^2\varrho_{03}^2 + \frac{1}{2}m_\varphi^2\varphi^2 + \frac{1}{3}b\ m\ g_\varphi^3\varphi^3 + \frac{1}{4}c g_\varphi^4\varphi^4\,. \qquad (2.52)
\end{aligned}
$$

For the pressure, where ($g^{nn}=-3$), we have:

$$
\begin{aligned}
P = \langle T^{nn}\rangle &= \frac{1}{3}\sum_{j=b,l}\left\langle\left(\frac{\partial\mathscr{L}}{\partial\left(\partial_n\Psi_j\right)}\right)\partial^n\Psi_j\right\rangle - \frac{1}{3}\left\langle g^{nn}\mathscr{L}\right\rangle\,, \\
&= \frac{1}{3}\sum_{j=b,l}\left\langle\left(\frac{\partial\mathscr{L}}{\partial\left(\partial_n\Psi_j\right)}\right)\partial^n\Psi_j\right\rangle + \mathscr{L}_{RMF}\,, \\
&= \frac{1}{3}\sum_{j=b,l}\langle\bar{\Psi}_j\gamma_n k^n\Psi_j\rangle + \mathscr{L}_{RMF}\,. \qquad (2.53)
\end{aligned}
$$

Once again it is missing the value for $\langle\bar{\Psi}\gamma_n k^n\Psi\rangle$. Using the same procedure as before for the energy density:

$$
\begin{aligned}
\left(\bar{\Psi}\gamma_i k^i\Psi\right)_{\boldsymbol{k},k} = \left(\bar{\Psi}\,\boldsymbol{\gamma}\cdot\boldsymbol{k}\,\Psi\right)_{\boldsymbol{k},k} &= \left(\Psi^\dagger\,\boldsymbol{\alpha}\cdot\boldsymbol{k}\,\Psi\right)_{\boldsymbol{k},k} = \left(\Psi^\dagger\frac{\partial H_D}{\partial\boldsymbol{k}}\Psi\right)_{\boldsymbol{k},k}\cdot\boldsymbol{k}\,, \\
&= \frac{\partial E(\boldsymbol{k})}{\partial\boldsymbol{k}}\cdot\boldsymbol{k}\,. \qquad (2.54)
\end{aligned}
$$

It is then possible to obtain:

$$
\begin{aligned}
\sum_{j=b,l}\langle\bar{\Psi}_j\gamma_i k^i\Psi_j\rangle &= \sum_{j=p,n}(2J_j+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\frac{\boldsymbol{k}\cdot\boldsymbol{k}}{\sqrt{\boldsymbol{k}^2+\left(m_j-g_\varphi\varphi^2\right)^2}}\Theta\left(\mu_j-E_j(\boldsymbol{k})\right) \\
&\quad + \sum_{j=e^-,\mu^-}(2J_j+1)\int\frac{d\boldsymbol{k}}{(2\pi)^3}\frac{\boldsymbol{k}\cdot\boldsymbol{k}}{\sqrt{\boldsymbol{k}^2+m_j^2}}\Theta\left(\mu_j-E_j(\boldsymbol{k})\right)\,, \\
&= \sum_{j=p,n}\frac{(2J_j+1)}{2\pi^2}\int_0^{K_{F,j}}\frac{k^4}{\sqrt{k^2+\left(m_j-g_\varphi\varphi^2\right)^2}}dk \\
&\quad + \sum_{j=e^-,\mu^-}\frac{(2J_j+1)}{2\pi^2}\int_0^{K_{F,j}}\frac{k^4}{\sqrt{k^2+m_j^2}}dk\,. \qquad (2.55)
\end{aligned}
$$

The pressure can be determined by combining Eq.2.20 with Eq.2.55:

$$
\begin{aligned}
P = & \frac{1}{3}\left[\sum_{j=p,n}\frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}}\frac{k^4}{\sqrt{k^2+\left(m_j-g_\varphi\varphi^2\right)^2}}dk + \sum_{j=e^-,\mu^-}\frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}}\frac{k^4}{\sqrt{k^2+m_j^2}}dk\right] \\
& - \frac{1}{2}m_\varphi^2\varphi^2 - \frac{1}{3}b\ m\ g_\varphi^3\varphi^3 - \frac{1}{4}c g_\varphi^4\varphi^4 + \frac{\xi}{4!}g_\omega^4\omega_0^4 + \frac{1}{2}m_\omega^2\omega_0^2 + \frac{1}{2}m_\varrho^2\varrho_{03}^2 + \Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0^2\,. \qquad (2.56)
\end{aligned}
$$

To compute the symmetry energy present in Eq.2.3, a derivation is outlined in Appendix A.2, resulting in the following expression:

$$E_{sym}(n) = \frac{1}{8}\frac{\left(g_\varrho\right)^2 n}{2\Lambda_\omega g_\varrho^2 g_\omega^2 \omega_0^2 + m_\varrho^2} + \frac{1}{6}k_F^2\left[\frac{1}{\sqrt{k_F^2 + m^{*2}}}\right]. \tag{2.57}$$

2.3 Macroscopic neutron star properties

Once the EoS is obtained, one may wonder how to derive the macroscopic properties of the NS from the EoS, such as mass, radius and TD in the context of star mergers. The process of establishing the connection from the EoS to mass radius is when the Tolman-Oppenheimer-Volkoff (TOV) equations come into play. Gravitational wave signals from binary NS mergers allow us to probe an additional parameter predicted by theoretical models, namely the TD. TheTD quantifies the deformations induced in a NS by the tidal gravitational field of its companion in a binary system.

2.3.1 Tolman-Oppenheimer-Volkoff equations

The TOV equation, first derived by Tolman, Oppenheimer, and Volkoff in 1939 [50, 51], is a fundamental equation derived from Einstein's field equations. It characterizes the structure of spherically symmetric and static relativistic stars in hydrostatic equilibrium consisting of an ideal and isotropic fluid. The TOV equation arises from the interplay between the pressure exerted by matter within the star and the gravitational force acting on it. In hydrostatic equilibrium, the pressure acting on an area A counterbalances the inward gravitational force generated by the star's mass. This equilibrium condition ensures that the star remains stable and does not collapse under its own gravity

$$\frac{dP(r)}{dr} = \frac{\varepsilon(r)m(r)}{r^2}\left[1+\frac{P(r)}{\varepsilon(r)}\right]\left[1+\frac{4\pi r^3 P(r)}{m(r)}\right]\left[1-\frac{2m(r)}{r}\right]^{-1}. \tag{2.58}$$

For the mass equation in its differential form

$$\frac{dm(r)}{dr} = 4\pi r^2 \varepsilon(r)\,. \tag{2.59}$$

This first equation represents the net force acting outward on the surface of a thin shell of matter, caused by the pressure difference $dP(r)$ between the inner and outer edges of the shell. On the right side of the equation, the first factor corresponds to the attractive Newtonian gravitational force exerted by the mass enclosed within the shell. The remaining three factors account for the precise corrections introduced by general relativity. In addition to the TOV equation, the mass continuity equation is also crucial. It determines the mass contained within each shell of matter. By combining these equations, one can establish the balance at each radius (r) between the internal pressure and the gravitational attraction of the mass within that region. Collectively, these equations govern the hydrostatic equilibrium in the context of general relativity.

In order to solve the coupled equations, two essential ingredients are required: an EoS and appropriate boundary conditions:

➜ **Central condition of the star** – mass is $m(r=0)=0$ and pressure $P(r=0)=P_0$, where the central pressure is typically specified as the initial condition.

➜ **Surface conditions of the star** – $P(r=R)=0$ and $m(r=R)=M$, where M is the total mass of the star.

2.3.2 Tidal deformability

When discussing the formation of NSs in Section 2.1.1, we mentioned that one possible scenario is the merger of two NSs. In such binary systems, the dimensionless TD, Λ and its associated Love number k_2 play a crucial role. These quantities measure the degree of deformation experienced by a NS as a result of the tidal forces exerted by its companion in the binary system. This quantity is defined by

$$\Lambda = \frac{2}{3} k_2 (C)^{-5}, \tag{2.60}$$

where $C = M/R$ is the compactness of the star. As evident, Λ is related to the love number initially proposed by Augustus Love in 1909 [52] based on Newtonian theory. The relativistic theory of tidal effects was deduced in 2009 [53, 54], leading to the computation of Love numbers for NSs to become a field of intense investigation. The second order Love number is then defined as

$$\begin{aligned} k_2 = & \frac{8C^5}{5}(1-2C)^2\left[2+2C\left(y_R-1\right)-y_R\right]\{2C\left[6-3y_R+3C\left(5y_R-8\right)\right] \\ & +4C^3\left[13-11y_R+C\left(3y_R-2\right)+2C^2\left(1+y_R\right)\right] \\ & +3(1-2C)^2\left[2-y_R+2C\left(y_R-1\right)\right]\ln(1-2C)\}^{-1}, \end{aligned} \tag{2.61}$$

the boundary value of $y(r)$ at the surface, denoted as $y_R \equiv y(r=R)$ arises from the following first-order differential equation

$$\frac{dy(r)}{dr} = -\frac{y(r)^2}{r} - \frac{y(r)}{r} F(r) - rQ(r) \tag{2.62}$$

where $F(r)$ and $Q(r)$ being functionals of $\varepsilon(r)$, $P(r)$, and $m(r)$, are represented as

$$F(r) = \left(1-4\pi r^2[\varepsilon(r)-P(r)]\right)\left[1-\frac{2m(r)}{r}\right]^{-1}, \tag{2.63}$$

$$Q(r) = 4\pi\left[5\varepsilon(r)+9P(r)+\frac{\varepsilon(r)+P(r)}{v_s^2(r)}-\frac{6}{r^2}\right]\left[1-\frac{2m(r)}{r}\right]^{-1} - \frac{4m^2(r)}{r^4}\left[1+\frac{4\pi r^3 P(r)}{m(r)}\right]^2\left[1-\frac{2m(r)}{r}\right]^{-2}. \tag{2.64}$$

These equations can be solved numerically simultaneously with the TOV Eqs. 2.58 and 2.59, also taking into account $y(r=0)=2$, as shown in [36], along with the conditions for the center of the star. By doing so, the value of y_R can be determined and subsequently substituted into the Love number, Eq.2.61, to obtain the TD (Λ) using Eq.2.60.

2.3.3 The inverse mapping

The process of solving the TOV equation, coupled with the Love number calculation, can be regarded as a mapping from the functional space of the EoS onto the functional space of the mass-radius ($\boldsymbol{M}-\boldsymbol{R}$) and mass-TD ($\boldsymbol{M}-\boldsymbol{\Lambda}$) relations. This mapping can be defined as:

$$f_{curve}: \quad (EoS) \quad \rightarrow \quad (\boldsymbol{M},\boldsymbol{R},\boldsymbol{\Lambda})\,. \tag{2.65}$$

While this mapping is one-to-one and invertible when the complete mass-radius and mass-TD curves are available, the challenge arises from the fact that observations cannot provide the entire continuous curves. This is even more evident if one wants to obtain also the TD curve.
The inverse mapping from observational data is a critical aspect that needs to be addressed, and it can be expressed as:

$$f^{-1}_{data}: \quad (\boldsymbol{M}_R,\boldsymbol{R},\boldsymbol{M}_\Lambda,\boldsymbol{\Lambda})_{data} \quad \rightarrow \quad (EoS)\,. \tag{2.66}$$

This encompasses various types of observations, including electromagnetic data denoted by $\boldsymbol{M}_R-\boldsymbol{R}$ pairs and gravitational wave events identified by $\boldsymbol{M}_\Lambda$-$\boldsymbol{\Lambda}$. Fig.2.4 illustrates this inverse mapping.

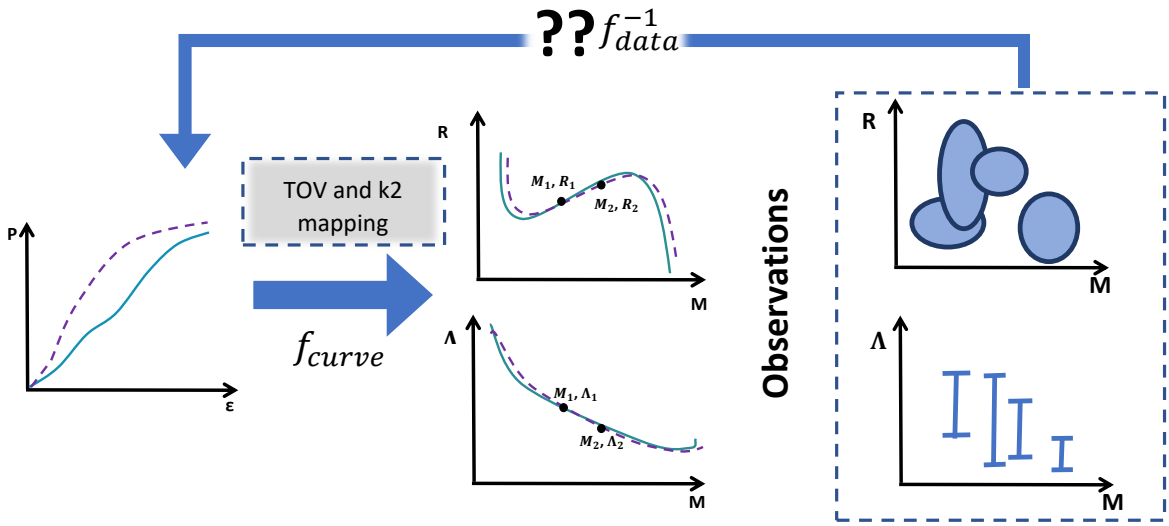


Figure 2.4: Schematic representation of the TOV and Love number mappings illustrated as f_{curve}, along with an illustration of the inverse mapping directly from observations denoted as f^{-1}_{data}.

Challenges in capturing this inverse mapping include:

➜ **Lack of observations** – The $\boldsymbol{M}-\boldsymbol{R}$ data obtained from observations does not form a continuous curve. Each NS observation contributes only a single point on the $\boldsymbol{M}-\boldsymbol{R}$ curve. Consequently, the available data is limited, providing only partial insights into the overall $\boldsymbol{M}-\boldsymbol{R}$ curve. Even with a significant number of observed NSs, crucial information about potentially unstable branches of the $\boldsymbol{M}-\boldsymbol{R}$ curve—regions where NSs cannot exist—remains absent. This problem becomes even more pronounced when considering TD, as observations for this parameter are scarcer, making its accurate mapping even more challenging.

➜ **Uncertainties in observations** – Observations are represented as distributions rather than single points that reflect the range of potential values. These distributions arise from the challenges of accurately collecting data from such remote objects. When considering properties like the mass-radius $\boldsymbol{M-R}$ relation, each data point is accompanied by a set of observational uncertainties, visualized through credibility distributions. However, it's important to recognise that the central peak position of these distributions does not necessarily correspond exactly to the true underlying $\boldsymbol{M-R}$ relationship. The same conclusion can be drawn for the TD in the realm of NS mergers.

The complexity and uncertainties inherent in this problem emphasize the need to not only accomplish the inverse mapping but also accurately account for the associated uncertainties. Our goal is to infer the most probable EoS from observations while accounting for the uncertainties introduced by the f_{data}^{-1} mapping. The central focus of this dissertation revolves around finding an approach capable of adeptly handling these observational uncertainties while conducting the mapping itself. This naturally leads to the question

But how can we achieve this?

3 Uncertainty Aware Neural Networks

As established in the preceding chapter, the challenge at hand involves finding a tool for the f_{data}^{-1} mapping, connecting observations to NSs EoS. To effectively tackle this challenge, we will harness the power of BNNs. BNNs introduce a distinct approach to NNs by representing weights not as point estimates but as distributions. To fully comprehend the significance of this approach, it is essential to provide contextualization. This can be achieved by first introducing Bayes rule in Section 3.1, next in Section 3.2 we will delve into the concept of NNs. Building upon this understanding, Section 3.3 will introduce the probabilistic view of NNs. In Section 3.4, we will arrive at the core topic of BNNs and finally in Section 3.5 we will illustrate the types of uncertainty with an example.

3.1 Bayesian statistics

In order to understand BNN it is necessary to first understand Bayes' Theorem, first described by Thomas Bayes in 1763. This theorem states that the posterior distribution is represented by Eq. 3.1, where D stands for some dataset, and $\boldsymbol{\theta}$ for the model parameters

$$\text{Posterior} \leftarrow P(\boldsymbol{\theta}|D) = \frac{\overbrace{P(\boldsymbol{\theta})}^{\text{Prior}}\overbrace{P(D|\boldsymbol{\theta})}^{\text{Likelihood}}}{\underbrace{P(D)}_{\text{Evidence}}} = \frac{P(\boldsymbol{\theta})P(D|\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} P(D|\boldsymbol{\theta}')P(\boldsymbol{\theta}')d\boldsymbol{\theta}'}. \tag{3.1}$$

Each component of the equation carries distinct significance:

- ➜ **Prior** – Prior knowledge about the model before seeing the data.
- ➜ **Likelihood** – Represents how likely it is the data given the parameters. It returns a single scalar representing the probability of drawing the dataset, given the model parameters.
- ➜ **Evidence** – Also known as Marginal likelihood, is interpreted as a normalizing constant ensuring that the likelihood times the prior sums to one over the domain of $\boldsymbol{\theta}$.

3.2 Neural networks in a nutshell

NNs have become a buzzword in physics recently. These come from a broader field named ML, which also came from Artificial intelligence. NNs were first proposed in 1944 by Warren McCullough and Walter Pitts, then in 1957 Rosenblatt came and gave birth to the perceptron, and in 1969 surged the

XOR problem (AI winter). In 1986, Dr.Geoffrey Hinton and his colleagues developed the backpropagation algorithm to train a multilayer NN. In 1987, was the first application of High energy physics with NNs [55].

Deep learning is often categorized into three distinct branches supervised, unsupervised and reinforcement learning. Supervised learning is task driven and characterized by architectures like Convolutional NNs and Recurrent NNs. Unsupervised learning is data-driven, encompassing models such as Variational Autoencoders and Generative Adversarial Networks and reinforcement learning involves interacting with the environment. Within these categories, we encounter discriminative and generative models, where the first one aims to reconstruct a target variable y given some observation x, and the last one aims to generate new data samples that resemble the training data.
Throughout this chapter, our focus remains exclusively on supervised regression tasks, as they provide the foundational framework for addressing our specific problem.

3.2.1 How it works

A NN, such as a feedforward network, consists of interconnected neurons organized into layers, including input $s = 0$, hidden $s = 1, ..., S-1$ and output layers, $s = S$, so $S+1$ is the total number of layers. A set of input-output vectors is given by $D = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{\mathscr{D}}$, where $\mathscr{D}$ is the number of samples of each pair. The input vector is denoted as $\boldsymbol{x}^{(i)} = \{x_q^{(i)}\}_{q=1}^{Q}$, and resides in $\boldsymbol{x}^{(i)} \in \mathbb{R}^Q$, while the output vector is represented by $\boldsymbol{y}^{(i)} = \{y_k^{(i)}\}_{k=1}^{K}$ and belongs to $\boldsymbol{y}^{(i)} \in \mathbb{R}^K$. Each layer is composed of neurons, and these neurons are connected to neurons in adjacent layers through weights, denoted as the matrix $\boldsymbol{W}$ and with W as elements. Additionally, each neuron has a bias term, denoted as $\boldsymbol{b}$, which serves as a threshold. The network output is defined as $\hat{\boldsymbol{y}} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, where we define the model parameters $\boldsymbol{\theta}$ defined as $\boldsymbol{\theta} = \{(\boldsymbol{W}^{[s]}, \boldsymbol{b}^{[s]})\}_{s=1}^{S}$. The computation within each neuron involves multiplying the weights with the corresponding neurons from the previous layer and summing these products for each hidden unit. The bias term is also added to the sum, giving a result of $\Sigma_n = \sum_{q=1}^{Q} W_{nq} x_q + b_n$, where n represents the nth forward neuron. Subsequently, a non-linear function named activation function, denoted as $\phi(\cdot)$, is applied to each neuron, both in the hidden layers and the output layer. This activation function essentially defines the neuron's behaviour or how active it becomes. The final result for the nth neuron is then defined as $a_n = \phi(\Sigma_n)$. A schematic representation of this process is provided in Fig. 3.1. The most frequently used activation functions are present in the following multicolumn of equations, with the respective graphical representation in Fig. 3.2

$$\text{Softplus} : \phi(x) = \log(1 + e^x), \tag{3.2}$$

$$\text{Hyperbolic Tangent (Tanh)} : \phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{3.3}$$

$$\text{Sigmoid} : \phi(x) = \frac{1}{1 + e^{-x}}, \tag{3.4}$$

$$\text{ReLU} : \phi(x) = max(0, x)\,. \tag{3.5}$$

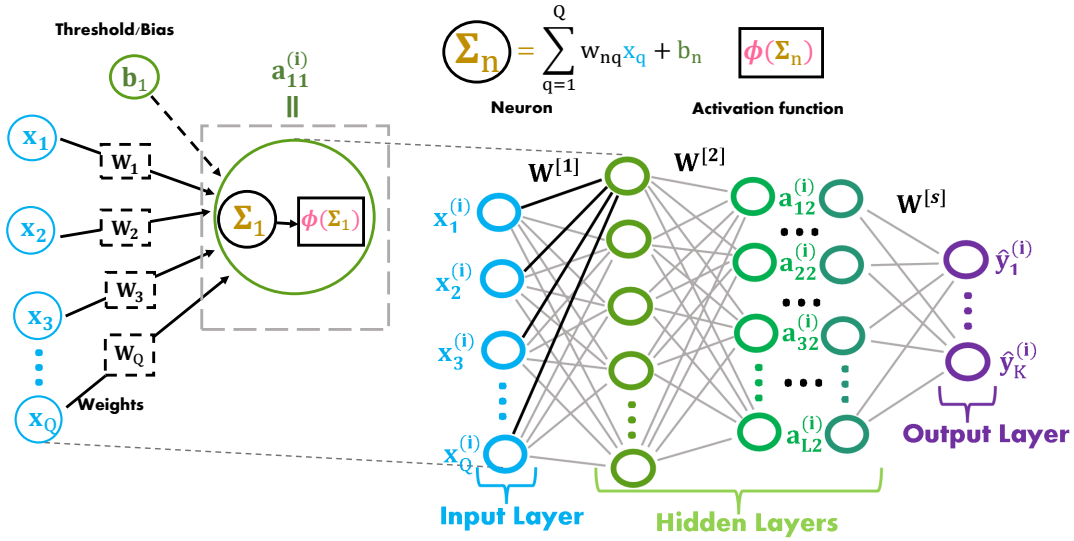


Figure 3.1: Sketch of a multilayer perceptron, depicting its diverse layers including input, hidden, and output layers. The diagram illustrates the arrangement of hidden units within the layers, displaying the fundamental components of a single perceptron.

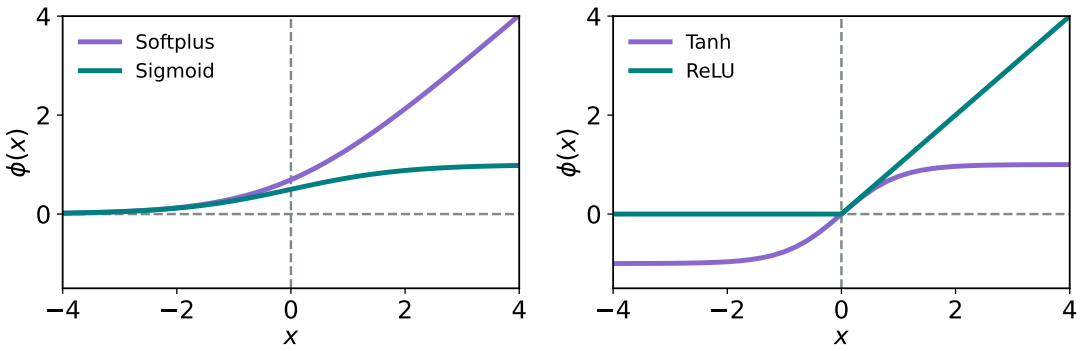


Figure 3.2: Active functions softplus, Eq. 3.2, and sigmoid, Eq. 3.4, are present in the left plot while hyperbolic tangent (Tanh), Eq. 3.3, and ReLU, Eq. 3.5 are in the right plot.

For the first hidden layer the hidden units would be represented as

$$\begin{bmatrix} a_{11}^{(i)} \\ \vdots \\ a_{n1}^{(i)} \end{bmatrix} = \phi^{[1]} \left(\begin{bmatrix} W_{11}^{[1]} & \cdots & W_{1q}^{[1]} \\ \vdots & \ddots & \vdots \\ W_{n1}^{[1]} & \cdots & W_{nq}^{[1]} \end{bmatrix} \begin{bmatrix} x_1^{(i)} \\ \vdots \\ x_q^{(i)} \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ \vdots \\ b_n^{[1]} \end{bmatrix} \right). \tag{3.6}$$

Passing through all layers, the final result is calculated as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \phi^{[S]} \left(\boldsymbol{W}^{[S]} \underbrace{\phi^{[S-1]} \left(\cdots \phi^{[1]} \left(\boldsymbol{W}^{[1]} \boldsymbol{x} + \boldsymbol{b}^{[1]} \right) \cdots \right)}_{\mathbf{a}_{S-1}} + \boldsymbol{b}^{[S]} \right), \tag{3.7}$$

in matrix form, this becomes

$$\begin{bmatrix} \hat{y}_1^{(i)} \\ \vdots \\ \hat{y}_k^{(i)} \end{bmatrix} = \phi^{[S]} \left(\begin{bmatrix} W_{11}^{[S]} & \cdots & W_{1h}^{[S]} \\ \vdots & \ddots & \vdots \\ W_{k1}^{[S]} & \cdots & W_{kh}^{[S]} \end{bmatrix} \begin{bmatrix} \mathrm{a}_{1S-1}^{(i)} \\ \vdots \\ \mathrm{a}_{hS-1}^{(i)} \end{bmatrix} + \begin{bmatrix} b_1^{[S]} \\ \vdots \\ b_k^{[S]} \end{bmatrix} \right). \tag{3.8}$$

Data spliting

When aiming to determine the optimal hyperparameters for both the model architecture and the training process, the ultimate goal is to create a model that not only fits the observed data but also demonstrates robust generalization. Achieving this objective involves addressing an optimization problem dedicated to finding the values of the most suitable hyperparameters. In NNs, this process involves a training set and a testing set, each serving a distinct purpose. The training set is utilized to fine-tune the model's parameters, while the testing set evaluates the model's performance using the optimized parameters obtained during training. In order to enable the model to quantify the discrepancy between the training data and the unseen data, thereby preventing overfitting, the train set is going to be divided into two sets, one for training and another for validation. The precise ratio of samples for training, validation and test subsets must be chosen based on the nature of the model and the available data samples.

3.2.2 How it trains

The primary objective during training is to minimize a designated loss function by optimizing the model's parameters, denoted as $\boldsymbol{\theta}$, in order to attain the lowest value of the loss

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(\theta). \tag{3.9}$$

To achieve this, the backpropagation method is employed, comprising two core phases: the forward pass and the backward pass as defined in [56]. In the forward pass, the loss function is computed, estimating the error between the known output and the predicted one

$$L(\boldsymbol{\theta}) = \frac{1}{\mathscr{D}} \sum_{i=1}^{\mathscr{D}} l\left(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\right), \tag{3.10}$$

the choice of the loss function varies depending on the task, for regression one possible choice can be the mean square error

$$l_{mse} = \left(\boldsymbol{y}^{(i)} - f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\right)^2, \tag{3.11}$$

or cross-entropy [1] for classification. The backward pass calculates derivatives of the loss function with respect to each parameter in the NN. These derivatives are then subtracted from the corresponding

[1] $l = -\left[\boldsymbol{y}^{(i)} \ln(f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})) + (1 - \boldsymbol{y}^{(i)}) \ln(1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))\right]$

parameter values to update them

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{\mathscr{D}} \sum_{i=1}^{\mathscr{D}} \frac{\partial l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))}{\partial \boldsymbol{\theta}}, \tag{3.12}$$

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \eta \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{3.13}$$

Here η representes the learning rate, a hyperparameter that governs the step size during parameter updates. The derivatives are computed using the chain rule:

$$\frac{\partial l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))}{\partial \boldsymbol{\theta}} = \frac{\partial l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))}{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})} \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})}{\partial \phi^{[S]}} \frac{\partial \phi^{[S]}}{\partial \boldsymbol{\theta}}. \tag{3.14}$$

This entire procedure is iteratively applied to update each parameter within the NN during the training process

$$\begin{bmatrix} \boldsymbol{W}'^{[1]} \\ \vdots \\ \boldsymbol{W}'^{[S]} \\ \boldsymbol{b}'^{[1]} \\ \vdots \\ \boldsymbol{b}'^{[S]} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}^{[1]} \\ \vdots \\ \boldsymbol{W}^{[S]} \\ \boldsymbol{b}^{[1]} \\ \vdots \\ \boldsymbol{b}^{[S]} \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))}{\partial \boldsymbol{W}^{[1]}} \\ \vdots \\ \frac{\partial l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))}{\partial \boldsymbol{W}^{[S]}} \\ \frac{\partial l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))}{\partial \boldsymbol{b}^{[1]}} \\ \vdots \\ \frac{\partial l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))}{\partial \boldsymbol{b}^{[S]}} \end{bmatrix}. \tag{3.15}$$

Weight updates in NNs are commonly accomplished through three distinct methods: stochastic gradient descent, minibatch gradient descent and batch gradient descent. Stochastic gradient descent, is where the weights are updated for each individual data point. While this can lead to rapid updates, it tends to be computationally expensive due to frequent weight adjustments. Minibatch gradient descent, on the other hand, is the most used method it involves segmenting the dataset into smaller mini-batches, with the batch size determining the number of data points used to update the weights in each iteration. This method offers a strike balance between accurate optimizations and computation efficiency. Batch gradient descent only updates the weights after processing the entire dataset, even though this approach can lead to more stable weight updates, it might require longer training times. An epoch is then when all the training dataset has been used.

When using minibatch gradient descent, the process can be described as follows:

$$\begin{bmatrix} \mathrm{a}_{11}^{(1)} & \cdots & \mathrm{a}_{11}^{(\frac{\mathscr{D}}{B})} \\ \vdots & \cdots & \vdots \\ \mathrm{a}_{n1}^{(1)} & \cdots & \mathrm{a}_{n1}^{(\frac{\mathscr{D}}{B})} \end{bmatrix} = \phi^{[1]} \left(\begin{bmatrix} W_{11}^{[1]} & \cdots & W_{1q}^{[1]} \\ \vdots & \ddots & \vdots \\ W_{n1}^{[1]} & \cdots & W_{nq}^{[1]} \end{bmatrix} \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(\frac{\mathscr{D}}{B})} \\ \vdots & \cdots & \vdots \\ x_q^{(1)} & \cdots & x_q^{(\frac{\mathscr{D}}{B})} \end{bmatrix} + \begin{bmatrix} b_1^{[1]} & \cdots & b_1^{[1]} \\ \vdots & \cdots & \vdots \\ b_n^{[1]} & \cdots & b_n^{[1]} \end{bmatrix} \right). \tag{3.16}$$

When minimising the loss function, the optimization process can encounter challenges like becoming stuck in local minimums or saddle points. Even when utilizing minibatch gradient descent, convergence to the global minimum can be prolonged. To address this issue, enhancements to the optimization methods are often employed. These improvements include the incorporation of momentum, which helps accelerate convergence, or the utilization of optimization algorithms such as the Adagrad [57] or Adam [58] methods. These strategies help to navigate the optimization landscape more effectively and facilitate convergence to the optimal solution.

3.2.3 How it improves

In the process of training a model, there's a risk of it learning the training data too well and becoming inflexible when faced with new unseen data. This is called overfitting. To strike a balance between making accurate predictions and avoiding overfitting, regularization techniques come into play.
One common regularizer is the L_2 regularizer, which aims to keep the weights of the model as small as possible. It's added to the optimization process and is represented as:

$$L(\boldsymbol{\theta}) = \frac{1}{\mathscr{D}} \sum_{i=1}^{\mathscr{D}} l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})) + \lambda ||\boldsymbol{W}||^2 . \tag{3.17}$$

Here, λ, a hyperparameter, controls the amount of regularization applied, and it encourages the model to have smaller weights. And $||\boldsymbol{W}||$ represents the norm of the weight vector.
Another approach is the L_1 regularizer, also known as Lasso, which aims to eliminate the weights of the least important features. It performs a kind of feature selection, where some features contribute less to the model's performance and are effectively removed. It's represented as

$$L(\boldsymbol{\theta}) = \frac{1}{\mathscr{D}} \sum_{i=1}^{\mathscr{D}} l(\boldsymbol{y}^{(i)}, f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})) + \lambda ||\boldsymbol{W}|| . \tag{3.18}$$

There are more methods two of them being: dropout and early stopping. Dropout involves randomly "dropping out" or deactivating certain neurons during training to prevent the model from relying too heavily on specific features. Early stopping on the other hand, involves monitoring the model's performance on a validation dataset during training. If the performance on the validation data starts getting worse, training is stopped early.

3.3 Exploring uncertainty quantification using neural networks

Once we have completed the review of traditional NNs, we will progress to introduce the probabilistic view of NNs. This viewpoint allows us to explore the incorporation of uncertainty into the weights and outputs of NNs. In the realm of uncertainty quantification, we encounter two distinct types: aleatoric and epistemic uncertainty (more information can be found [59]). Aleatoric uncertainty arises from the inherent uncertainty present in the data itself, while epistemic uncertainty relates to the uncertainty surrounding the model or network. It is important to note that while epistemic uncertainty can be mitigated by gathering more data or refining the model, aleatoric uncertainty remains unaffected by these measures. However, NNs can still effectively quantify aleatoric uncertainty, but epistemic uncertainty is obtained via BNNs, as will be explained in Section 3.4. Within aleatoric uncertainty, we can further differentiate between two types, discussed in [60]: homoscedastic, Section 3.3.1, and heteroscedastic, Section 3.3.2 uncertainty. Homoscedastic uncertainty refers to situations where the level of uncertainty is constant across all data points. Conversely, heteroscedastic uncertainty indicates that the level of uncertainty varies across different data points. By recognizing and understanding these distinct types of uncertainty, we gain valuable insights into the behaviour and limitations of NNs. This knowledge facilitates more informed decision-making and allows us to address different sources of uncertainty appropriately.

3.3.1 Homoscedastic aleatoric uncertainty

Homoscedastic regression is designed to account for constant noise in the target values, denoted as y, across all input points $x^{(i)}$. Consequently, the output of the NN results in a distribution characterized by a constant $\hat{\sigma}$. Assuming that the dataset adheres to the principles of being independent and identically distributed (i.i.d.), the likelihood function is defined as follows:

$$P(y|x,\boldsymbol{\theta},\hat{\sigma}^2) = \prod_{i=1}^{\mathscr{D}} \mathscr{N}(y(x^{(i)})|f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}),\hat{\sigma}^2)\,. \tag{3.19}$$

Here, $\hat{\sigma}^2$ acts as a hyperparameter. In contrast to the optimization process described in Section 3.2.2, our current objective is to maximize the likelihood of our data. This is equivalent to seeking the maximum log-likelihood of our data. We employ the logarithm because it is a monotonically increasing function and simplifies computational complexity when dealing with summation as opposed to multiplication. Equivalently, we aim to minimize the negative log-likelihood

$$\begin{aligned}
\boldsymbol{\theta}_{MLE} &= \arg\max_{\boldsymbol{\theta}} \log(P(y|x,\boldsymbol{\theta},\hat{\sigma}^2)) = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta})\,,\\
&= \arg\min_{\boldsymbol{\theta}} -\log(P(y|x,\boldsymbol{\theta},\hat{\sigma}^2))\,,\\
&= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{\mathscr{D}} -\log\left(\frac{1}{\sqrt{2\pi\hat{\sigma}^2}}\right) + \frac{[y(x^{(i)}) - \hat{\mu}^{(i)}]^2}{2\hat{\sigma}^2}\,,\\
&\propto \arg\min_{\boldsymbol{\theta}} \frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{\mathscr{D}} [y(x^{(i)}) - \hat{\mu}^{(i)}]^2, \quad \hat{\mu}^{(i)} = f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\,.
\end{aligned} \tag{3.20}$$

Where $L(\boldsymbol{\theta}) = l(\boldsymbol{\theta})/\mathscr{D}$. In this context, $\hat{\mu}^{(i)}$ denotes the predicted mean for the output $y\left(x^{(i)}\right)$. Now, the loss function being minimized simplifies to mean squared error since sigma is considered a constant. When you introduce a regularize to your NN model, it's like imposing a prior distribution on your parameters, which can be represented as $P(\boldsymbol{\theta}) = \mathscr{N}(\boldsymbol{\theta}|\mathbf{0},\alpha^{-1}\mathbb{1}) = \frac{\alpha}{2\pi}^{\frac{1}{2}} \exp\left(\frac{-\alpha\boldsymbol{\theta}^T\boldsymbol{\theta}}{2}\right)$. In the context of Bayes' rule, this corresponds to shifting the objective from merely maximizing the likelihood to maximizing the log posterior, often referred to as the Maximum A Posteriori (MAP) estimation

$$\begin{aligned}
\boldsymbol{\theta}_{MAP} &= \arg\max_{\boldsymbol{\theta}} \log(P(y|x,\boldsymbol{\theta},\hat{\sigma}^2)) + \log(P(\boldsymbol{\theta}))\,,\\
&\propto \arg\min_{\boldsymbol{\theta}} \frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{\mathscr{D}} [y(x^{(i)}) - \hat{\mu}^{(i)}]^2 + \frac{\alpha}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}\,,
\end{aligned} \tag{3.21}$$

which is equivalent to Eq. 3.17. Some intuition on this uncertainty is present in Fig. 3.3b.

3.3.2 Heteroscedastic aleatoric uncertainty

Heteroscedastic aleatoric uncertainty allows us to quantitatively assess the variability inherent in the data. This type of uncertainty allows one to capture the natural fluctuations in the output, taking into

account the influence of the input as well. Mathematically, the probability distribution of the output y given an input x and the model parameters $\boldsymbol{\theta}$ can be expressed as:

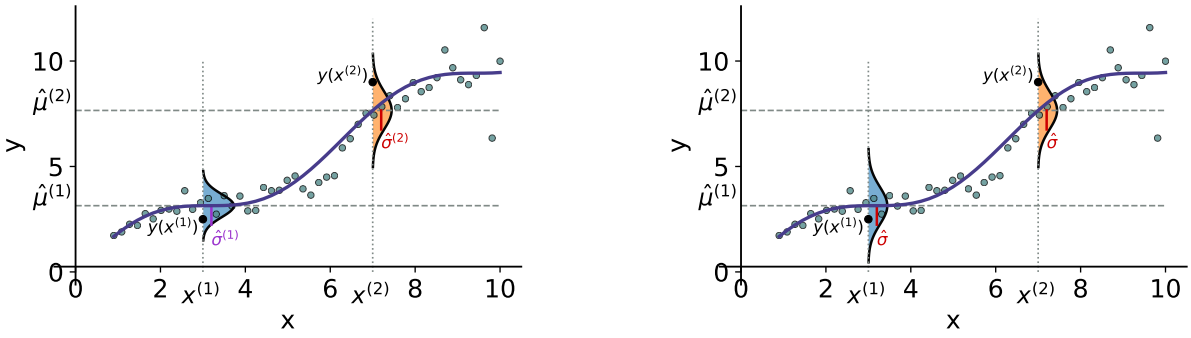
$$P(y|x,\boldsymbol{\theta}) = \mathcal{N}(y(x^{(i)})|\hat{\mu}^{(i)},\hat{\sigma}^{(i)2}) \,. \tag{3.22}$$

Here, $\hat{\mu}^{(i)}$ represents the predicted mean and $\hat{\sigma}^{(i)2}$ represents the predicted variance for the output y corresponding to the input $x^{(i)}$. The optimization changes slightly for heteroscedastic aleatoric uncertainty. The parameters $\boldsymbol{\theta}$ are optimized to minimize the negative log-likelihood, since we are considering the variance terms

$$\boldsymbol{\theta}_{MLE} = \arg\min_{\boldsymbol{\theta}} l_{NLL}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{\mathscr{D}} -\log\left(\frac{1}{\sqrt{2\pi\hat{\sigma}^{(i)2}}}\right) + \frac{[y(x^{(i)}) - \hat{\mu}^{(i)}]^2}{2\hat{\sigma}^{(i)2}} \,. \tag{3.23}$$

Where once again $L_{NLL}(\boldsymbol{\theta}) = l_{NLL}(\boldsymbol{\theta})/\mathscr{D}$. The intuition in this uncertainty is given in Fig. 3.3a. When doing backpropagation the derivatives with respect to the standard deviation and the mean will be different:

$$\nabla_\mu l_{NLL}(\boldsymbol{\theta}) = \sum_{i=1}^{\mathscr{D}} \frac{\left[y(x^{(i)}) - \hat{\mu}^{(i)}\right]}{\hat{\sigma}^{(i)2}} \quad \text{and} \quad \nabla_\sigma l_{NLL}(\boldsymbol{\theta}) = \sum_{i=1}^{\mathscr{D}} -\frac{1}{\hat{\sigma}^{(i)}} + \frac{\left[y(x^{(i)}) - \hat{\mu}^{(i)}\right]^2}{\hat{\sigma}^{(i)3}} \,. \tag{3.24}$$



(a) Heterocdastic aleatoric uncertainty.

(b) Homoscdastic aleatoric uncertainty.

Figure 3.3: Intuition on how aleatoric uncertainty is represented in the realm of homoscedastic and heteroscedastic uncertainty quantification. This is intended to emphasize that homoscedasticity assumes uniform uncertainty across all inputs, regardless of variations in noise levels. [2]

However, the parameters of the NN still represent point predictions. To capture epistemic uncertainty, which amounts from uncertainty in the model's parameters, we are going to introduce BNN.

3.4 Bayesian neural networks

BNNs where first introduced in 1992 by David McKay [40]. In [61], a broader review on BNNs is shown, here we will focus more on the Variational Inference (VI) method to train the network, more specifically with Bayes by backprop. BNNs simulate multiple possible NNs models by introducing stochastic weights. These networks operate by first choosing a functional model, i.e., a network architecture,

[2] The dataset used to illustrate the uncertainties was created using Eq. 3.44.

and then the stochastic model, i.e., the probability distributions for the weights. Bayesian inference is then required to train the network by defining the likelihood function of the observed data, $P(D|\boldsymbol{\theta})$, and the prior probability distribution over the model parameters, $P(\boldsymbol{\theta})$. It is then possible to employ Bayes theorem and obtain the posterior probability distribution, i.e. the probability of the model parameters given the data. Having a distribution on the weights, the prediction of these BNNs will become a Bayesian model average, the probability distribution of some unknown $\boldsymbol{y}^*$ given an input $\boldsymbol{x}^*$ is:

$$P(\boldsymbol{y}^*|\boldsymbol{x}^*, D) = \int_{\boldsymbol{\theta}} P(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}) P(\boldsymbol{\theta}|D) d\boldsymbol{\theta}. \tag{3.25}$$

$P(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})$ is considered to be the likelihood of our data, the distribution that comes out of the network and captures the noise present in our data, and $P(\boldsymbol{\theta}|D)$ is the posterior distribution of our weights, that brings up the uncertainty on the model. Another advantage of using these networks is that they capture two types of uncertainty, aleatoric uncertainty, uncertainty on the data, and epistemic uncertainty, uncertainty on the model estimation defined as $P(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})$ and $P(\boldsymbol{\theta}|D)$ respectively. However, solving Eq. 3.25 poses a significant challenge due to the complexity introduced by the posterior $P(\boldsymbol{\theta}|D)$ is a high dimensional and highly nonconvex probability distribution that also depends on the evidence $P(D) = \int_{\boldsymbol{\theta}'} P(D|\boldsymbol{\theta}') P(\boldsymbol{\theta}') d\boldsymbol{\theta}'$, which is a non-analytic expression that requires marginalizing over all model parameters.

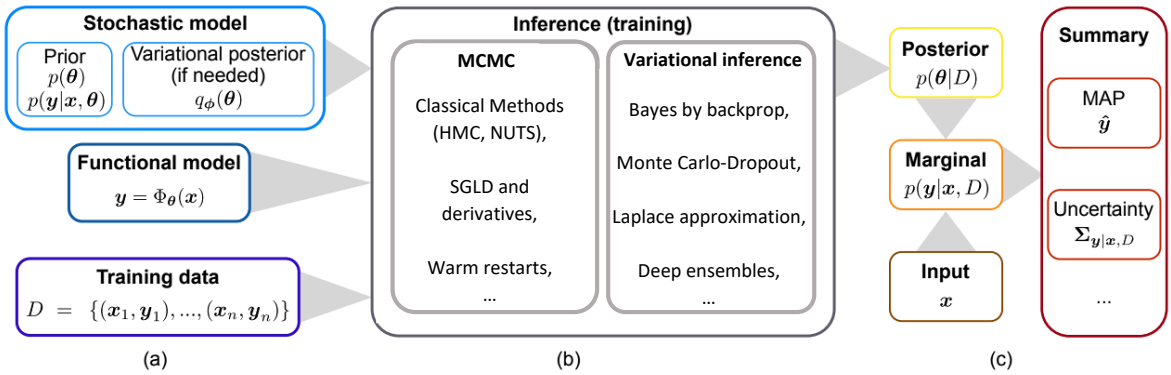


Figure 3.4: Framework adapted from [61], illustrating the step-by-step workflow for a) designing, b) training, and c) testing the BNN model, where is emphasised the different kinds of inference procedures.

In order to track the posterior distribution, there are two main approaches:Markov Chain Monte Carlo (MCMC) and VI. MCMC directly samples from the true posterior distribution by constructing a Markov chain that converges to the desired distribution. It is a powerful technique but can be computationally expensive and may require a large number of samples to obtain accurate estimates of the posterior. On the other hand, VI aims to approximate the true posterior distribution with a simpler distribution that is easier to work with. It involves finding the best approximation within a predefined family of distributions, such as Gaussian distributions. VI optimizes the parameters of this approximating distribution to minimize the divergence between the true posterior and the ap-

proximate distribution. While VI provides a computationally efficient approach, the quality of the approximation depends on the chosen family of distributions and the optimization algorithm used. Fig. 3.4 illustrates the workflow of a BNN, highlighting its three main stages: design, training, and prediction. In the design stage (a), the stochastic model is determined, which includes specifying the prior distribution for the model parameters, defining the likelihood function, and optionally selecting a variational posterior distribution. The training stage (b) involves approximating the posterior distribution using either MCMC or VI. This step aims to learn the model parameters based on the available data. Finally, in the prediction stage (c), with the posterior obtained during training and given some input, Bayesian model averaging can be performed to calculate predictions that take into account the uncertainty inherent in the model.

3.4.1 Variational inference

The VI method aims to approximate a variational posterior, $q_\phi(\boldsymbol{\theta})$, $\phi = (\boldsymbol{\mu}_\phi, \boldsymbol{\sigma}_\phi)$, to the real posterior, $P(\boldsymbol{\theta}|D)$, by minimizing the KL divergence between the two distributions. The KL divergence is a measure of dissimilarity between probability distributions. It quantifies the information lost when one distribution is used to approximate another. The KL divergence is asymmetric[3] and provides a weighted measure of the distance between the variational posterior and the true posterior. Specifically, it approaches zero when the variational posterior and the true posterior are identical, indicating a close match between the two distributions and is positive otherwise. Fundamentally, the objective is to find the variational posterior that minimizes the KL divergence with the true posterior. This minimization process seeks to find the best approximation of the true posterior within the chosen family of distributions parameterized by ϕ. By optimizing the variational parameters, the goal is to make the variational posterior as close as possible to the true posterior

$$q_{\phi^*} = \arg\min_{q_\phi} \mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)), \tag{3.26}$$

where the KL divergence is defined as

$$\begin{aligned}\mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) &= \mathbb{E}_{q_\phi(\boldsymbol{\theta})}\left[\log\left(\frac{q_\phi(\boldsymbol{\theta})}{P(\boldsymbol{\theta}|D)}\right)\right], \\ &= \int_{\boldsymbol{\theta}} q_\phi(\boldsymbol{\theta}) \log\left(\frac{q_\phi(\boldsymbol{\theta})}{P(\boldsymbol{\theta}|D)}\right) d\boldsymbol{\theta}\,. \end{aligned} \tag{3.27}$$

In order for the dependence on the true posterior to disappear, the last equation can be rewritten with the help of Bayes rule, Eq. 3.1, as

$$\begin{aligned}\mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) &= \int_{\boldsymbol{\theta}} q_\phi(\boldsymbol{\theta}) \log\left(\frac{q_\phi(\boldsymbol{\theta})P(D)}{P(D|\boldsymbol{\theta})P(\boldsymbol{\theta})}\right) d\boldsymbol{\theta}\,, \\ &= \mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta})) - \mathbb{E}_{q_\phi(\boldsymbol{\theta})}(\log P(D|\boldsymbol{\theta})) + \log P(D)\,, \\ &= F(D,\phi) + \underbrace{\log P(D)}_{=constant}\,, \end{aligned} \tag{3.28}$$

where $F(D,\phi) = \mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta})) - \mathbb{E}_{q_\phi(\boldsymbol{\theta})}(\log P(D|\boldsymbol{\theta}))$ is called the variational free energy. Ignoring the last term which is the logarithm of the evidence, since it is constant (the probability of the dataset

[3] $\mathrm{KL}(q||p) \neq \mathrm{KL}(p||q)$

doesn't change with the variational posterior) it does not affect the optimization process. So we can confidently say minimizing $\mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D))$ with respect to ϕ is the same as minimizing $F(D,\phi)$. Is going to be considered the Evidence Lower BOund (ELBO), which was originally derived in the literature through Jensen's inequality (more information can be found in Appendix B.4) and it is defined as ELBO $= -F(D,\phi)$. Eq. 3.28 can then be rewritten as

$$\mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) = -\mathrm{ELBO} + \log P(D)\,, \tag{3.29}$$

with the ELBO being

$$\mathrm{ELBO} = -\mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta})) + \mathbb{E}_{q_\phi(\boldsymbol{\theta})}(\log P(D|\boldsymbol{\theta}))\,. \tag{3.30}$$

The ELBO, as the name suggests, is the lower bound of the evidence, meaning since the KL divergence is always positive, $\mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) \geq 0$, and $\log P(D)$ is always negative, the ELBO will have to be always negative and smaller than the $\log P(D)$. Since the last term $\log P(D)$ is ignored Eq. 3.28 can then be rewritten as just equal to the $-$ELBO. To give some intuition on the ELBO interpretation, the $\mathbb{E}_{q_\phi(\boldsymbol{\theta})}(\log P(D|\boldsymbol{\theta}))$ is responsible for the data fit since it maximizes the expected log-likelihood of the data and the second, $\mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta}))$ serves as a complexity regularizer by punishing unnecessary divergence of the approximate posterior from the prior.
Having said that, our goal is now to maximize the ELBO:

$$\begin{aligned} q_{\phi^*} &= \arg\min_{q_\phi} \mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D))\,, \\ &= \arg\max_{q_\phi} \mathrm{ELBO} = \arg\min_{q_\phi} F(D,\phi)\,, \\ &= \arg\min_{q_\phi} \Big[\mathrm{KL}(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta})) - \mathbb{E}_{q_\phi(\boldsymbol{\theta})}(\log P(D|\boldsymbol{\theta}))\Big]\,. \end{aligned} \tag{3.31}$$

The final loss function we are trying to minimize uses Monte Carlo sampling to obtain the expected values, where $\boldsymbol{\theta}_n$ is being sampled from the variational posterior, $q_\phi(\boldsymbol{\theta})$,

$$F(D,\phi) \approx KL(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta})) - \frac{1}{N}\sum_{n=1}^{N} \log P(D|\boldsymbol{\theta}_n)\,, \tag{3.32}$$

where N represents the number of samples taken from the variational posterior for each value of the dataset. The KL divergence can either be considered exact if the two distributions are solvable, Appendix B.1 shows the example for two multivariate gaussian, or once again a Monte Carlo sampling approximation can be done. As was mentioned in Fig. 3.4, there are multiple methods of doing VI the method, that we are going to go into more detail about here is Bayes by backprop, but Monte Carlo dropout is also an interesting method as is like a simplification of VI, it was first introduced in [62], and one application in physics can be found in [63]. BNN are considered discriminate models, although they can also appear in generative models, those are not going to be mentioned here.

3.4.2 Bayes by backprop

When dealing with a Gaussian distribution without correlation for the variational posterior, a natural question arises: *How do you perform backpropagation when the weights are a distribution?* In

the article by Blundell *et al.* [64], a backpropagation algorithm is introduced, utilizing a generalized reparametrization trick designed for the weights. The algorithm is outlined as follows:

1. Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$;
2. Compute $\boldsymbol{\theta} = \boldsymbol{\mu}_\phi + \log(1+\exp(\boldsymbol{\rho})) \circ \boldsymbol{\epsilon}$, where $\phi = (\boldsymbol{\mu}_\phi, \boldsymbol{\sigma}_\phi = \log(1+\exp(\boldsymbol{\rho})))$ defines the variational parameters and $q_\phi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_\phi, {\boldsymbol{\sigma}_\phi}^2)$;
3. Calculate the gradients of the loss $f(D,\phi) = KL(q_\phi(\boldsymbol{\theta})||P(\boldsymbol{\theta})) - \log P(D|\boldsymbol{\theta})$:

$$\nabla_{\mu_\phi} = \frac{\partial f(D,\phi)}{\partial \boldsymbol{\theta}} + \frac{\partial f(D,\phi)}{\partial \boldsymbol{\mu}_\phi},$$
$$\nabla_\rho = \frac{\partial f(D,\phi)}{\partial \boldsymbol{\theta}} \frac{\boldsymbol{\epsilon}}{1+\exp(-\boldsymbol{\rho})} + \frac{\partial f(D,\phi)}{\partial \boldsymbol{\rho}};$$

4. Update the parameters:

$$\boldsymbol{\mu}'_\phi \leftarrow \boldsymbol{\mu}_\phi - \eta \nabla_{\mu_\phi},$$
$$\boldsymbol{\rho}' \leftarrow \boldsymbol{\rho} - \eta \nabla_\rho.$$

Here, $\circ$ represents element-wise multiplication. To ensure the positivity of σ, the softplus function, Eq. 3.2, present in Fig. 3.2 is employed for $\boldsymbol{\sigma}_\phi$, preventing it from becoming negative. The use of $\log(1+\exp(\rho))$ ensures numerical stability during training by avoiding potential issues with negative values. It's important to note that only a single sample of the loss function is used as mentioned in [61], leading to a potentially noisy loss function and gradients in comparison to classical backpropagation. However, when using the mini-batch method for the gradients that are already stochastic in nature, the Monte Carlo error only adds additional noise to an already noisy estimate. While the stochasticity of gradients might be perceived as a drawback in some cases, particularly in mini-batch optimization, it can actually assist in escaping local minima and achieving improved convergence. This reparametrization trick is derived from proposition 1 in the article by Blundell *et al.* [64].

3.4.3 Delving into the implications of correlated parameters

When considering correlation among the weights, the model's complexity increases substantially due to the significant expansion in the number of parameters. In such cases, the variational posterior $q_\phi(\theta)$ is represented by a vector of means μ_ϕ and a covariance matrix Σ_ϕ. To factorize the covariance matrix, the Cholesky decomposition is employed, as exemplified in Eq. 3.33 for a $n \times n$ matrix

$$A = \boldsymbol{L}\boldsymbol{L}^{\boldsymbol{T}} = \begin{bmatrix} L_{11} & & (0) \\ \vdots & \ddots & \\ L_{n1} & \cdots & L_{nn} \end{bmatrix} \begin{bmatrix} L_{11} & \cdots & L_{n1} \\ & \ddots & \vdots \\ (0) & & L_{nn} \end{bmatrix} = \begin{bmatrix} L_{11}^2 & & (symmetric) \\ \vdots & \ddots & \\ L_{n1}L_{11} & \cdots & \sum_{i=1}^{n} L_{ni}^2 \end{bmatrix}. \tag{3.33}$$

The softplus function, Eq. 3.2 (present in Fig. 3.2) is applied to the diagonal elements of the factorized matrix $\boldsymbol{L}$, resulting in $\boldsymbol{L}_S$, and then the covariance matrix is

$$\begin{aligned}\boldsymbol{\Sigma} = \boldsymbol{L}_S \boldsymbol{L}_S^T &= \begin{bmatrix} \log(1+\exp(L_{11}))^2 & & (symmetric) \\ \vdots & \ddots & \\ L_{n1}\log(1+\exp(L_{11})) & \cdots & \sum_{i=1}^{n-1} L_{ni}^2 + \log(1+\exp(L_{nn}))^2 \end{bmatrix}, \\ &\equiv \begin{bmatrix} \mathrm{Cov}(1,1) & & (symmetric) \\ \vdots & \ddots & \\ \mathrm{Cov}(n,1) & \cdots & \mathrm{Cov}(n,n) \end{bmatrix},\end{aligned} \tag{3.34}$$

where $\mathrm{Cov}(r,n) = \frac{\sum_i^N (r_i - \bar{r})(n_i - \bar{n})}{N}$ is the covariance for each term. Notably, it's crucial to highlight that we are describing the adopted methodology implemented by TensorFlow [65] when employing the *DenseVariational* Layers. Now, let's shift our focus to the parameters. We can distinguish between parameters related to the mean and covariance matrix. For the mean of the weights and bias, the number of parameters is $n \times q + n$, where is considered q the size of the input vector and n the number of hidden units per layer. These are represented as follows:

$$\begin{bmatrix} \mu_{W_{11}^{[1]}} & \cdots & \mu_{W_{1q}^{[1]}} \\ \vdots & \ddots & \vdots \\ \mu_{W_{n1}^{[1]}} & \cdots & \mu_{W_{nq}^{[1]}} \end{bmatrix} \text{ and } \begin{bmatrix} \mu_{b_1^{[1]}} \\ \vdots \\ \mu_{b_n^{[1]}} \end{bmatrix}. \tag{3.35}$$

In the case of the covariance matrix, the parameters being updated are contained within the $\boldsymbol{L}$ matrix. This matrix includes all the weights and bias parameters and thus has a total of $(n \times q + n) \times \frac{((n \times q + n)+1)}{2}$ parameters. The matrix is represented as:

$$\boldsymbol{L} = \begin{bmatrix} L_{W_{11}^{[1]} W_{11}^{[1]}} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ L_{W_{nq}^{[1]} W_{11}^{[1]}} & \cdots & L_{W_{nq}^{[1]} W_{nq}^{[1]}} & 0 & \cdots & 0 \\ L_{b_1^{[1]} W_{11}^{[1]}} & \cdots & L_{b_1^{[1]} W_{nq}^{[1]}} & L_{b_1^{[1]} b_1^{[1]}} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ L_{b_n^{[1]} W_{11}^{[1]}} & \cdots & L_{b_n^{[1]} W_{nq}^{[1]}} & L_{b_n^{[1]} b_1^{[1]}} & \cdots & L_{b_n^{[1]} b_n^{[1]}} \end{bmatrix}_{(n \times q + n)(n \times q + n)}. \tag{3.36}$$

Thus, the total number of parameters for the mean and the $\boldsymbol{L}$ matrix is $(n \times q + n)\left(1 + \frac{((n \times q + n)+1)}{2}\right)$. This substantial number of parameters naturally leads to significantly longer training times. The process of updating all these parameters demands more computational resources compared to a conventional NN, where the number of parameters per layer is typically $n \times q + n$

$$\begin{bmatrix} W_{11}^{[1]} & \cdots & W_{1q}^{[1]} \\ \vdots & \ddots & \vdots \\ W_{n1}^{[1]} & \cdots & W_{nq}^{[1]} \end{bmatrix} \text{ and } \begin{bmatrix} b_1^{[1]} \\ \vdots \\ b_n^{[1]} \end{bmatrix}. \tag{3.37}$$

An important note is that is only being consider the weights per Layer, so this would become even more challenging when considering an increase of the hidden layers, exacerbating the computational

complexity.

Upon incorporating the precise value of the KL divergence for a variational posterior with a covariance matrix, along with a prior characterized by a mean of zero and a covariance matrix equal to the identity (as outlined in Appendix B), and utilizing the mini-batch gradient descent method (also detailed in the same appendix), the obtained loss function takes the following form:

$$F(D,\phi) = \frac{1}{2\mathscr{D}}\left[\left(-\log\det(\Sigma_\phi)\right) - S_c + \operatorname{tr}\left(\Sigma_\phi\right) + (\mu_\phi)^T(\mu_\phi)\right] - \frac{1}{B}\sum_{i=1}^{B}\frac{1}{N}\sum_{n=1}^{N}\log P(y^{(i)}|x^{(i)},\theta_n)\,, \tag{3.38}$$

where B is the number of points in each mini-batch, N is the number of samples taken from the variational posterior, $\mathscr{D}$ is the number of rows of the training dataset and S_c is the dimension of the correlation matrix. The backpropagation algorithm continues to adhere to the previously explained reparametrization trick. However, the substitution of the vector σ_ϕ with the matrix $\boldsymbol{L}$ necessitates the incorporation of certain adjustments in the optimization procedure and results in an expansion of the number of parameters that require being updated.

3.4.4 Prediction

Once the network is trained and the best variational parameters are obtained for the distribution of the posterior, Eq. 3.25 becomes solvable and predictions are obtained using Monte Carlo estimations

$$\begin{aligned} P(\boldsymbol{y}^*|\boldsymbol{x}^*,D) &= \int_{\boldsymbol{\theta}} P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta})q_\phi(\boldsymbol{\theta})d\boldsymbol{\theta}\,, \\ &= \frac{1}{N}\sum_{n=1}^{N} P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}_n), \quad \boldsymbol{\theta}_n \sim q_\phi(\boldsymbol{\theta}). \end{aligned} \tag{3.39}$$

The mean $\hat{\boldsymbol{\mu}}$ and variance $\hat{\boldsymbol{\sigma}}^2$ vectors of the predicting distribution $P(\boldsymbol{y}^*|\boldsymbol{x}^*,D)$ can be calculated, for a fixed $\boldsymbol{x}^*$, by applying the law of total expectation

$$\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*,D] = \mathbb{E}_{q_\phi(\boldsymbol{\theta})}\left[\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]\right]\,, \tag{3.40}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N}\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}_n}\,, \tag{3.41}$$

and law of total variance (demonstrated in Appendix B.3),

$$\operatorname{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*,D] = \mathbb{E}_{q_\phi(\boldsymbol{\theta})}\left[\operatorname{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]\right] - \operatorname{Var}_{q_\phi(\boldsymbol{\theta})}\left[\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]\right]\,, \tag{3.42}$$

$$\hat{\boldsymbol{\sigma}}^2 = \underbrace{\frac{1}{N}\sum_{n=1}^{N}\hat{\boldsymbol{\sigma}}^2_{\boldsymbol{\theta}_n}}_{\text{Aleatoric uncertainty}} + \underbrace{\frac{1}{N}\sum_{n=1}^{N}(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}_n} - \hat{\boldsymbol{\mu}})\circ(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}_n} - \hat{\boldsymbol{\mu}})}_{\text{Epistemic uncertainty}}\,. \tag{3.43}$$

The predicted variance captures both epistemic and aleatoric uncertainties [66]. $\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]$ and $\operatorname{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]$ are respectively the mean and variance of $\boldsymbol{y}^*$ according to $P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta})$. The term $\mathbb{E}_{q_\phi(\boldsymbol{\theta})}\left[\operatorname{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]\right]$ represents the average value of $\operatorname{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]$ when $\boldsymbol{\theta}\sim q_\phi(\boldsymbol{\theta})$, and $\operatorname{Var}_{q_\phi(\boldsymbol{\theta})}\left[\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]\right]$ is the variance of $\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}]$ when $\boldsymbol{\theta}\sim q_\phi(\boldsymbol{\theta})$. This is being illustrated in Fig. 3.5.

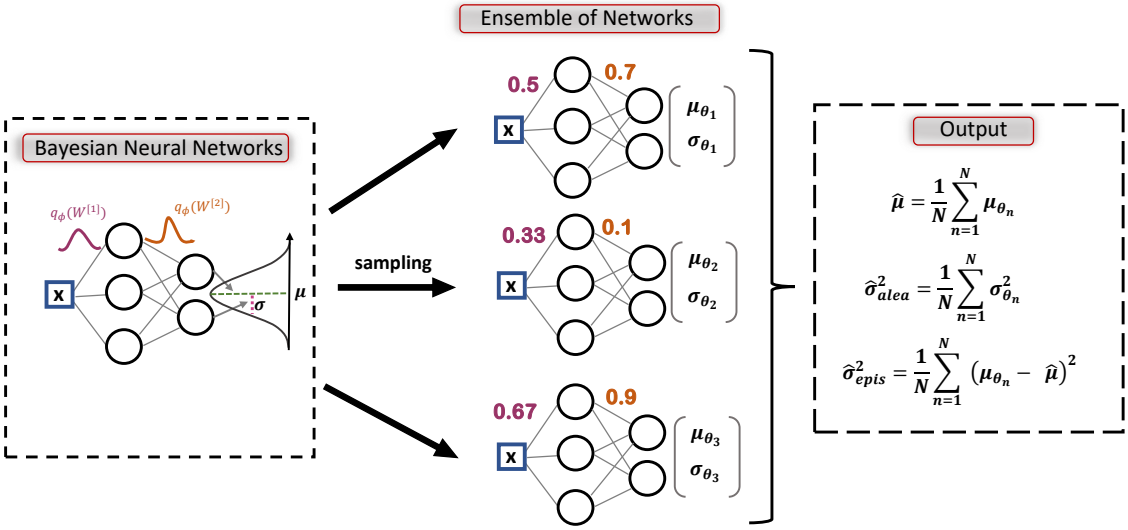


Figure 3.5: Intuitive illustration showcasing the predicted uncertainty quantification provided by the optimized BNN model.

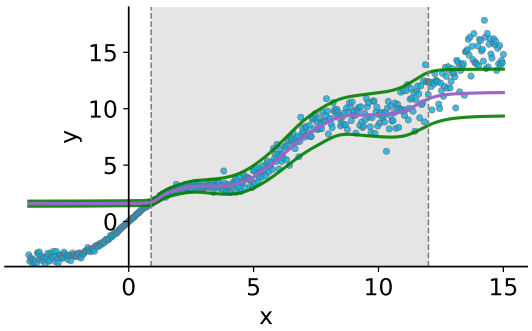
3.5 Illustration of the types of uncertainties

To gain a clearer insight into the mentioned types of uncertainties, we will illustrate their behavior using a simple regression model as an example. We selected a model with two hidden layers, employing sigmoid (Eq. 3.4) and ReLU (Eq. 3.5) activation functions respectively, and each hidden layer containing six hidden units. The training dataset was generated according to the equation:

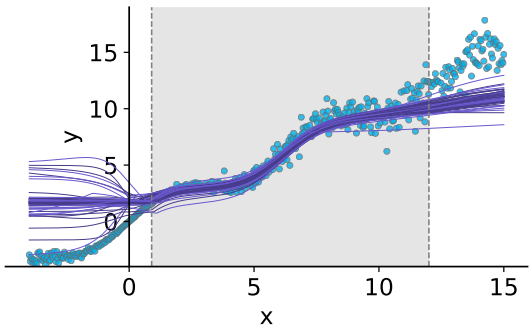
$$y = \sin(x) + x(1 + 0.1\epsilon(x)) , \quad \epsilon(x) \sim \mathcal{N}(0,1) . \tag{3.44}$$

Here, $\epsilon(x)$ represents the source of the noise. For the training process, we synthesized 1000 data points within the interval $x \in [0.9, 12]$ while during the prediction phase, we employed 500 data points within the range $x \in [-2, 16]$. Our exploration began by training a NN with a distribution as the output, effectively capturing the inherent uncertainty present in the data - known as aleatoric uncertainty. This is visually illustrated in Fig.3.6a. Subsequently, we introduced uncertainty solely to the weights, as shown in Fig. 3.6b - known as epistemic uncertainty. Notably, the uncertainty grows significantly in data-scarce regions.

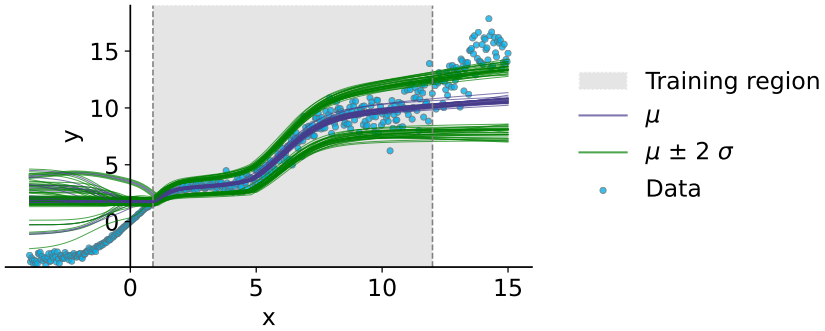
Finally, we combined both types of uncertainty, resulting in the model depicted in Fig. 3.6c. In this combined approach, the model exhibited heightened uncertainty in the low x region beyond the training data, while still effectively capturing the inherent uncertainty within the training region. By visually examining these prototype and simplistic models, we can develop a more intuitive grasp of aleatoric and epistemic uncertainties and their influence on predictions.



(a) Aleatoric uncertainty.



(b) Epistemic uncertainty.



(c) Epistemic and aleatoric uncertainty.

Figure 3.6: Illustration representing the discussed types of uncertainty. The grey band indicates the training dataset interval, while the points symbolize the test dataset. Beginning with aleatory uncertainty, the mean and standard deviation of a single prediction are depicted. Subsequently, epistemic uncertainty is illustrated through an ensemble of 30 predictions for only the mean. Lastly, a combination of epistemic and aleatoric uncertainty is shown for an ensemble of 30 predictions for the mean and the standard deviation of the dataset.

4 In-Depth Analysis of the Dataset

In this chapter, we explore the dataset construction and its behavior to align precisely with our requirements and objectives. The foundational raw data comes from the work of Malik *et al.* [25], a framework outlined in Section 4.1. Subsequently, we adapt this raw data to our unique problem, as detailed in Section 4.2. Within this section, we delve deep into the properties of the dataset's specifications that will significantly shape our forthcoming analyses and insights.

4.1 Unveiling the dataset source

In the article [25], a field theoretical approach is adopted to construct a dataset of nuclear EoSs using the RMF approximation. This approach incorporates self-interactions and mixed meson terms, which are crucial in determining the behavior of the EoS, as thoroughly detailed in chapter 2.2. The authors calculated the masses and radii of NSs using the TOV equations, as described in Section 2.3.1, and the TD using the method outlined in Section 2.3.2. The EoS is described by a set of coupling constants denoted by $\mathbf{\Theta} = \{g_\varphi, g_\omega, g_\varrho, \xi, b, c, \eta, \Lambda_\omega\}$, which govern the interactions of nucleons and mesons within the RMF framework. To obtain realistic values for these coupling constants, a Bayesian inference technique is employed. The Bayesian setup involves estimating the $\mathbf{\Theta}$ parameters in a way that satisfies various constraints imposed by a selection of nuclear saturation properties described in Section 2.1.3. These constrains include the saturation density n_0, the binding energy per nucleon of symmetric nuclear matter ϵ_0, the symmetry energy E_{sym} and the incompressibility K_0, all of which are evaluated at saturation density. Moreover, the NS maximum mass, the positive behaviour of the pure neutron matter pressure derivative with respect to the density and the low-density pure neutron matter EoS obtained from an accurate N^{3}LO calculation in chiral effective field theory are also taken into account as constraints.

Within this Bayesian framework, the $\mathbf{\Theta}$ parameters are systematically sampled to explore the parameter space effectively while adhering to the minimal constraints listed in Table 4.2.

But how does the mechanism of the Bayesian set up truly work?

Based on the constraints imposed by the fitted data, the authors define a prior belief on the parameters of the equations. This prior belief is then updated using Bayes theorem, introduced in Section 3.1, to obtain the posterior distribution. In order to establish a Bayesian parameter optimization

system, they specified four key components: the prior, the fit data, the likelihood function and the sampler, since they do not consider the evidence.

The Prior - The prior domain is determined using a Latin hypercube sampling method from a uniform prior distributions specified in Table 4.1. This way is then possible to cover a broad range of nuclear matter saturation properties for the underlying RMF model.

Table 4.1: Uniform prior, $\mathscr{U}[\min,\max]$, range for the coupling constantes represented as $\mathbf{\Theta}$, of the RMF models. Specifically, B and C are $b \times 10^3$ and $c \times 10^3$, respectively.

Nº	Parameters ($\mathbf{\Theta}$)	min	max
1	g_σ	6.5	15.5
2	g_ω	6.5	15.5
3	g_ϱ	6.5	16.5
4	B	0.5	9.0
5	C	-5.0	5.0
6	ξ	0.0	0.04
7	Λ_ω	0.0	0.12

The Fit Data - The fit data provided in Table 4.2 consists of important nuclear saturation properties, n_0, ϵ_0, K_0, and $J_{\text{sym},0}$, all evaluated at n_0 and considered to be gaussian distributions with the respective parameters. In addition to the saturation properties, the pressure of pure neutron matter at specific densities (0.08, 0.12, and 0.16 fm^{-3}) obtained from N^{3}LO calculations in chiral effective field theory [1], is incorporated. To account for the associated uncertainties in these calculations, they consider 2× N^{3}LO data uncertainty. Even more two important step function probabilities are considered, the maximum mass of a NS must exceed 2.0 $M_\odot$ and the derivative of the pressure with respect to density must always be greater than zero.

The Log-Likelihood - For the different parameters of the fit data present in Table 4.2 it's used the log-likelihood in Eq. 4.1 as a cost function for the nuclear saturation properties, taking into account the uncertainties σ_j associated with each data point j. For the pressure values in pure neutron matter from chiral effective field theory they used a box function probability present in Eq. 4.2, and for the maximum mass of NSs and the condition that the pure neutron matter pressure needs to be an increasing function of the density it's used a step function probability defined as Eq. 4.3.

$$Log(\mathrm{L}_{NSP}) = -0.5 \times \sum_j \left\{ \left(\frac{d_j - m_j(\mathbf{\Theta})}{\sigma_j} \right)^2 + Log(2\pi\sigma_j^2) \right\} , j = \rho_0, \epsilon_0, K_0, J_{sym,0} , \tag{4.1}$$

$$Log(\mathrm{L}_P) = Log \left\{ \prod_j \frac{1}{2\sigma_j} \frac{1}{\exp\left(\frac{|d_j - m_j(\mathbf{\Theta})| - \sigma_j}{0.015} \right) + 1} \right\} , j = P(0.08), P(0.12), P(0.16) , \tag{4.2}$$

$$\mathrm{L}_M = \begin{cases} 1 & \text{if} \quad M_{max} > 2 \\ 0 & \text{otherwise} \end{cases} , \qquad \mathrm{L}_{dP} = \begin{cases} 1 & \text{if} \quad \frac{dP}{d\rho} > 0 \\ 0 & \text{otherwise} \end{cases} . \tag{4.3}$$

So the final Likelihood would be

$$\mathrm{L}_{total} = \mathrm{L}_{dP} \times \mathrm{L}_{M} \times Log(\mathrm{L}_{NSP}) \times Log(\mathrm{L}_{P})\,. \tag{4.4}$$

The Sampler - To explore the seven-dimensional posterior, the nested sampling algorithm, initially proposed in [67], was employed. Specifically, the PyMultinest sampler [68, 69] was utilized due to its efficiency in handling low-dimensional problems. The sampling process began with four thousand "n-live" points, representing an initial ensemble of live points in the parameter space. The EoS dataset for further analyses was generated using the full posterior, resulting in a collection of 25287 EoS.

Table 4.2: The fit data used consists of the constraints on various quantities with the Bayesian inference method to generate the model sets. These constraints include the binding energy per nucleon ϵ_0, incompressibility K_0, and symmetry energy $J_{\mathrm{sym},0}$ at the nuclear saturation density n_0, each with a 1σ uncertainty. Additionally, the pressure of pure neutron matter is considered at specific densities, obtained from a chiral effective field theory calculation [1], and incorporated into the likelihood with a $2\times$ N^{3}LO uncertainty, which increases with density. Moreover, the maximum mass of NSs is constrained to be above $2M_\odot$.

Quantity		Value/Band	Ref
[fm^{-3}]	n_0	0.153 ± 0.005	[70]
NMP	ϵ_0	-16.1 ± 0.2	[71]
[MeV]	K_0	230 ± 40	[72, 73]
	$J_{\mathrm{sym},0}$	32.5 ± 1.8	[74]
PNM [MeV fm^{-3}]	$P(n)$	$2\times$ N^{3}LO	[1]
	dP/dn	>0	
NS mass [$M_\odot$]	$M_{\max}$	>2.0	[8]

This approach allows for a comprehensive study of nuclear matter and provides a rich dataset containing NS observables, including mass, radii and TD curves for each of the respective EoS. Moreover, it encompasses additional parameters such as the particle fraction in the system and the nuclear matter properties, obtained from rearranging the EoS as present in Eq. 2.4. For further insights and a visual representation of the posterior distribution of the parameters, we recommend the reader to examine Fig. 1 in the article by Malik *et al.* [25], for Set 0, which is the one we are employing from the article.

4.2 Navigating into the dataset reconstruction

Having understood how the underlying dataset for our study was created, let's proceed to explore our chosen output variable, as outlined in Section 4.2.1, and delve into the meticulous adaptation process to our unique objectives, detailed in Section 4.2.2. Subsequently, we detail the construction of "mock" observations for the input, Section 4.2.3. Finally, we delve into the correlations within the dataset in Section 4.2.4.

4.2.1 Key elements in neutron star understanding

Our central pursuit within this study revolves around the comprehensive examination of two key aspects: the speed of sound and the proton fraction. The determination of the speed of sound involves a direct calculation by evaluating the derivative of pressure with respect to energy density

$$\frac{v_s^2}{c^2} = \frac{\partial P}{\partial \varepsilon}\,. \tag{4.5}$$

However, it's essential to emphasise the significance of two critical considerations: thermodynamic stability and causality. The pressure must exhibit a monotonically non-decreasing relationship with the energy density. Additionally, the speed of sound, denoted as v_s, must satisfy causality[1]. Recalling the definition of partial derivatives and applying it to the pressure with respect to the baryonic density we express it as

$$\frac{\partial P}{\partial n} = \lim_{h\to 0} \frac{P(n+h)-P(n)}{h} = \lim_{(n_{i+1}-n_i)\to 0} \frac{P(n_{i+1})-P(n_i)}{n_{i+1}-n_i}\,, \tag{4.6}$$

where n_i stands for the ith segment of the baryonic density, and where we realized a change of variables of $(n+h)-n \to n_{i+1}-n_i$. Applying the same methodology to the energy density, we approximate Eq. 4.5 as

$$\frac{v_s^2(n_i)}{c^2} = \frac{\partial P}{\partial \varepsilon} = \frac{P(n_{i+1})-P(n_i)}{\varepsilon(n_{i+1})-\varepsilon(n_i)}\,. \tag{4.7}$$

The speed of sound is an important and open question that is being the target of a lot of studies lately. Having the speed of sound squared one can understand how does the pressure is varying with the energy density. Machine learning methods have also tried to obtain this quantity such as once again in the works by Fujimote *et al.* [28, 29, 27]. Also in a more recent work by Soma *et al.* [30], they reconstruct the speed of sound from the obtained EoS, and some conclusions are drawn such as that the speed of sound is extremely valuable in determining NS composition. With this quantity is also easier to capture oscillation in the EoS of the NS.

The proton fraction, denoted as $y_p = \frac{n_p}{n}$, is directly acquired from the coupled equations defined in Chapter 2 within the framework outlined in the article explained in Section 4.1. This quantity quantifies the asymmetry of nuclear matter within these compact objects and plays a crucial role in determining the possibility of the direct Urca process. Notably, it remains a relatively uncharted

[1] $v_s < c$

territory within the landscape of ML EoS construction methodologies. This distinctiveness arises from the fact that conventional ML methods often rely on agnostic models such as polytropes to formulate EoS datasets. In contrast, our approach stands apart by utilizing a specific nuclear model family to generate a comprehensive array of possible EoS.

The motivation behind exploring the proton fraction and the speed of sound is further enhanced by the use of the dataset established in [25] since the findings of earlier studies. Notably, previous attempts to reconstruct the EoS using meta-models or restricted models have revealed the inherent challenges in extracting nuclear matter properties solely from the β-equilibrium EoS without considering compositions (or symmetry energy at high densities) [44, 75].

Hence, the dataset we're relying on, originating from an RMF approach, holds significant importance in training our BNN model to accurately link NS observables with the EoS. Precisely, when predicting the squared speed of sound and the proton fraction. A few key aspects of this dataset should be highlighted. It's exclusively derived from a RMF approach, focusing on the core of NS and considering only hadronic matter. It takes into account constraints from observations and ensures that the matter is in beta equilibrium.

4.2.2 Unpacking input and output components

Our work aims to take simulated observations of NSs (referred to as the set $\boldsymbol{X}$) and input them into a BNN model, $P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})$. Here, $\boldsymbol{\theta}$ represents parameters sampled from a posterior distribution. The ultimate goal is then to create a probability distribution for the output space, which we refer to as the set $\boldsymbol{Y}$. This set provides insights into various properties of NS matter, including vital parameters like the speed of sound, $v_s^2(n)$, and the proton fraction, $y_p(n)$. Each of these sets, $\boldsymbol{Y}$ and $\boldsymbol{X}$, comprises $\mathscr{D}$ rows of vectors $\boldsymbol{y}$ and $\boldsymbol{x}$, respectively, $\mathscr{D}$ is the dataset size we are using . In simpler terms, $\boldsymbol{Y}$ is represented as $\boldsymbol{Y} = \{\boldsymbol{y}^{(i)}\}_{i=1}^{\mathscr{D}}$ and $\boldsymbol{X}$ is represented as $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{\mathscr{D}}$. To effectively capture the properties of NS, we've chosen to evaluate each element of $\boldsymbol{Y}$, at 15 specific baryonic densities denoted as n_k, where $K = 15$ is the size of the output vector, e.g., $\boldsymbol{y}^{(i)} \equiv \{y_p^{(i)}(n_k)\}_{k=1}^{15} = [y_p^{(i)}(n_1), y_p^{(i)}(n_2), \cdots, y_p^{(i)}(n_{15})]$ for the proton fraction and the same analogy for the speed of sound. These density points are evenly distributed between $n_1 = 0.15\ \text{fm}^{-3}$ and $n_{15} = 1.0\ \text{fm}^{-3}$, as outlined in Table 4.3. To better understand how we discretise the output space elements for $v_s^2(n)$ and $y_p(n)$, we refer to Fig. 4.1.

Table 4.3: Equally spaced values of the baryonic density used throughout the dissertation and respective designation.

Notation	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}	n_{13}	n_{14}	n_{15}
n_k [fm^{-3}]	0.15	0.21	0.27	0.33	0.39	0.45	0.51	0.57	0.64	0.70	0.76	0.82	0.88	0.94	1.00

Next, we construct two distinct types of datasets, each sharing the same structure for the output space $\boldsymbol{Y}$ but differing in the input space $\boldsymbol{X}$. These structures are defined as follows:

$$\boldsymbol{x} = [M_1, \cdots, M_5, R_1, \cdots, R_5]\,,$$
$$\boldsymbol{x'} = [M_1, \cdots, M_5, R_1, \cdots, R_5, M_1^{'}, \cdots, M_5^{'}, \Lambda_1, \cdots, \Lambda_5]\,.$$

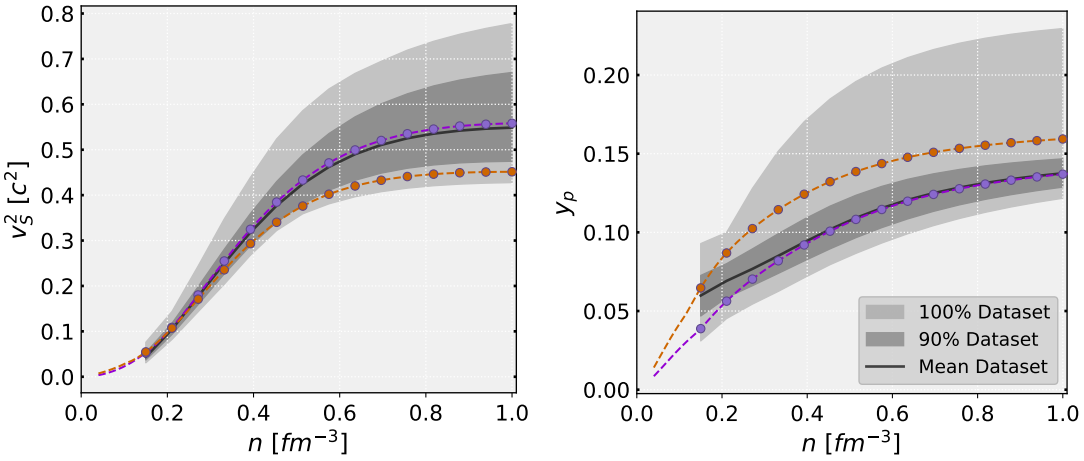


Figure 4.1: Visual representation of the output vector for $v_s^2(n)$ and $y_p(n)$ from left to right coming from the same two EoSs: each EoS is represented in our datasets by the 15 points of the baryonic density. The light grey area shows the range of minimum and maximum (100%) values in the training dataset, while the darker grey area represents 90% of the training dataset for both plots.

The first structure denoted as $\boldsymbol{x}$, consists of 10 values, $\boldsymbol{x}^{(i)} = \{x_q^{(i)}\}_{q=1}^{10}$, each representing five measurements of mass (M) and radius (R), and the second structure, denoted $\boldsymbol{x'}$, contains 20 values, $\boldsymbol{x'}^{(i)} = \{x_q^{(i)}\}_{q=1}^{20}$, covering five $M_j(R_j)$ and five $\Lambda_j(M_j^{'})$ observations. These different input structures enable us to evaluate the significance of including TD (Λ) measurements in our model predictions.

Regarding dataset size, we randomly split the total number of EoS into two sets: a training set, denoted as $\boldsymbol{X}$ and $\boldsymbol{Y}$, which includes 80% of the data ($\mathscr{D}$ =22 758 EoS), and a test set, denoted as $\boldsymbol{X}_T$ and $\boldsymbol{Y}_T$, comprising the remaining 20% ($\mathscr{D}_T$ =2 529 EoS). In a nutshell, our output elements are 15-dimensional vectors which we denote as $\boldsymbol{y}^{(i)} = \{y_k^{(i)}\}_{k=1}^{15} \equiv \{y_p^{(i)}(n_k)\}_{k=1}^{15}$ or $\boldsymbol{y}^{(i)} = \{y_k^{(i)}\}_{k=1}^{15} \equiv \{v_s^{2(i)}(n_k)\}_{k=1}^{15}$ for the speed of sound and have a total amount of training and test data denoted as $\boldsymbol{Y} = \{\boldsymbol{y}^{(i)}\}_{i=1}^{22758}$ and $\boldsymbol{Y}_T = \{\boldsymbol{y}^{(i)}\}_{i=1}^{2529}$ respectively. For the input space, we offer two variations. For the 10-dimensional option, $\boldsymbol{x}^{(i)} = \{x_q^{(i)}\}_{q=1}^{10} \equiv \{(M_j^{(i)}, R_j^{(i)})\}_{j=1}^{5}$, both the training and test data are denoted as $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{22758}$ and $\boldsymbol{X}_T = \{\boldsymbol{x}^{(i)}\}_{i=1}^{2529}$. For the 20-dimensional structure, $\boldsymbol{x'}^{(i)} = \{x_q^{(i)}\}_{q=1}^{20} \equiv \{(M_j^{(i)}, R_j^{(i)}), (M'^{(i)}_j, \Lambda_j^{(i)})\}_{j=1}^{5}$, the training and test data are denoted as $\boldsymbol{X'} = \{\boldsymbol{x'}^{(i)}\}_{i=1}^{22758}$ and $\boldsymbol{X}'_T = \{\boldsymbol{x'}^{(i)}\}_{i=1}^{2529}$. The selection between these options depends on the specific dataset we are working with.

Now, let's delve into how we created these synthetic observational datasets in the next section.

4.2.3 Simulating realistic mock data

The statistical procedure for generating the synthetic observational datasets is executed as follows:

1. For each EoS, we randomly select 5 NS mass values, denoted as $M_j^{\langle 0 \rangle}$, from a uniform distribution ranging between $1.0 M_\odot$ and $M_{\max}$.

2. Subsequently, the NS radius R_j is sampled from a Gaussian distribution centered at the TOV solution, indicated as $R(M_j^{\langle 0 \rangle})$, with a standard deviation of σ_R.

3. Finally, the ultimate NS mass is sampled from a Gaussian distribution centered at $M_j^{\langle 0\rangle}$ with a standard deviation of σ_M.

This process is summarized by the following equations:

$$M_j^{\langle 0\rangle} \sim \mathscr{U}[1, M_{\max}] \quad \text{(in units of } \mathrm{M}_\odot\text{)}\,, \tag{4.8}$$

$$R_j \sim \mathscr{N}\left(R\left(M_j^{\langle 0\rangle}\right), \sigma_R^2\right)\,, \tag{4.9}$$

$$M_j \sim \mathscr{N}\left(M_j^{\langle 0\rangle}, \sigma_M^2\right)\,, \quad j = 1, \cdots, 5\,. \tag{4.10}$$

From these equations it is then possible to construct the first input structure $\boldsymbol{x} = [M_1, \cdots, M_5, R_1, \cdots, R_5]$, where each pair (M_j, R_j) is a possible realisation (*observation*) that characterises the $M(R)$ diagram of the specific EoS. This procedure is similar to that used in [27], where a Gaussian noise was applied to 15 values of the $M(R)$ curve and then shifted from the original mass-radius curve: $M_j = M_j^{\langle 0\rangle} + \mathscr{N}\left(0, \sigma_M^2\right)$ and $R_j = R\left(M_j^{\langle 0\rangle}\right) + \mathscr{N}\left(0, \sigma_R^2\right)$, for $j = 1, \cdots, 15$.
To create the second structure, $\boldsymbol{x'} = [M_1, \cdots, M_5, R_1, \cdots, R_5, M_1^{'}, \cdots, M_5^{'}, \Lambda_1, \cdots, \Lambda_5]$, we repeat the 3 steps defined previously, but now add additional steps to implement the mass and TD pairs

$$M_j^{'} \sim \mathscr{U}[1, M_{\max}] \quad \text{(in units of } \mathrm{M}_\odot\text{)}\,, \tag{4.11}$$

$$\Lambda_j \sim \mathscr{N}\left(\Lambda(M_j^{'}), \sigma_\Lambda^2(M_j^{'})\right)\,, \quad j = 1, \cdots, 5, \tag{4.12}$$

where $\Lambda(M_j^{'})$ is given by the $\Lambda(M)$ relation of the specific EoS and $\sigma_\Lambda(M_j^{'})$ describes an overall dispersion around the mean value. The way we set up $\sigma_\Lambda(M_j^{'})$ should mirror our expectation of how the uncertainty in TD observations depends on the NS mass. In our study, we chose $\sigma_\Lambda(M_j^{'})$ to be represented as $\sigma_\Lambda(M_j^{'}) = \text{constant} \times \hat{\sigma}(M_j^{'})$. In this representation, the constant speaks for the magnitude of dispersion of the dataset and is multiplied by $\hat{\sigma}(M_j^{'})$, where $\hat{\sigma}(M_j^{'})$ represents how much the values of $\Lambda(M)$ vary in the EoSs dataset, as shown in Fig. 4.2.

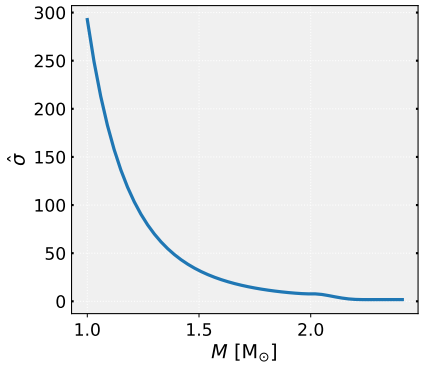


Figure 4.2: Standard deviation of $\Lambda(M)$ for the set of EoS.

In order to obtain multiple sets of observations for each EoS, we are going to repeat the aforementioned processes for the same EoS n_s times, where we define n_s as the number of times we resample the input vector, $\boldsymbol{x}$, for the same EoS. Consequently, the dataset is structured as $\mathbb{X} = \{\boldsymbol{X}_i\}_{i=1}^{n_s}$ and $\mathbb{Y} = \{\boldsymbol{Y}_i\}_{i=1}^{n_s}$. This approach expands our dataset to a size of $\mathbb{D} = n_s \times \mathscr{D}$, where $\mathscr{D}$ denotes the original dataset size for the number of EoSs. To illustrate, if we selected $n_s = 20$, this implies running the above procedures 20 times for each EoS (20 *observations*), and thus obtaining $\mathbb{X} = \{\boldsymbol{X}_i\}_{i=1}^{20} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, \cdots, \boldsymbol{X_{20}}\}$ and the output $\mathbb{Y} = \{\boldsymbol{Y}_i\}_{i=1}^{20} = \{\boldsymbol{Y}, \boldsymbol{Y}, \cdots, \boldsymbol{Y}\}$. This formalism is applied to both the training and testing sets, resulting in the generation of four datasets with distinct properties, as outlined in Table 4.4.

Regarding Table 4.4, Sets 1 and 2 exclusively contain information about NS radii (input vector $\boldsymbol{x}$ is 10-dimensional), while sets 3 and 4 also incorporate TD data (input vector $\boldsymbol{x'}$ is 20-dimensional).

Analyzing sets 1 and 2 helps us comprehend how reducing the spread of observations around the TOV solution impacts predictions and uncertainties. Similarly, sets 3 and 4 allow us to investigate the potential effects on model predictions resulting from an increased scattering of observations around the mean value of TD.

For our training sets, we employ 60 observations, denoted as $n_s = 60$ so $\mathbb{X} = \{\boldsymbol{X}_i\}_{i=1}^{60}$ and $\mathbb{Y} = \{\boldsymbol{Y}_i\}_{i=1}^{60}$, for each specific EoS. In contrast, for the test sets, we use $n_s = 1$ leading to $\mathbb{X}_T = \{\boldsymbol{X}_{T,i}\}_{i=1}^{10}$ and $\mathbb{Y}_T = \{\boldsymbol{Y}_{T,i}\}_{i=1}^{10}$. This distinction seeks to simulate a real-world scenario where we only have access to a single observation of the *true* EoS. Here, by single observation, we mean $n_s = 1$ that corresponds to five $M_j(R_j)$ observations (sets 1 and 2) or five $M_j(R_j)$ observations and five $\Lambda_j(M'_j)$ observations (sets 3 and 4).

To illustrate the process of dataset generation, Fig. 4.3 displays 60 observations for two distinct EoSs that belong to the generated datasets. Note that, for each EoS, there are a total of 300 points in the $M(R)$ diagram, resulting from 60 EoS observations, each with 5 NS observations. The difference in σ_R between dataset 1 and dataset 2 is evident, highlighting the distinctions between these two datasets. Additionally, Fig. 4.3 shows the TD values for datasets 3 and 4 on the left side.

Table 4.4: Generation parameters for each dataset. $\hat{\sigma}(M_j)$ denotes the standard deviation of $\Lambda(M)$ calculated on the train set.

Dataset	$\sigma_M\ [M_\odot]$	σ_R [km]	$\sigma_\Lambda(M_j)$
1	0.05	0.15	—
2	0.1	0.3	—
3	0.1	0.3	$0.5\hat{\sigma}(M_j)$
4	0.1	0.3	$2\hat{\sigma}(M_j)$

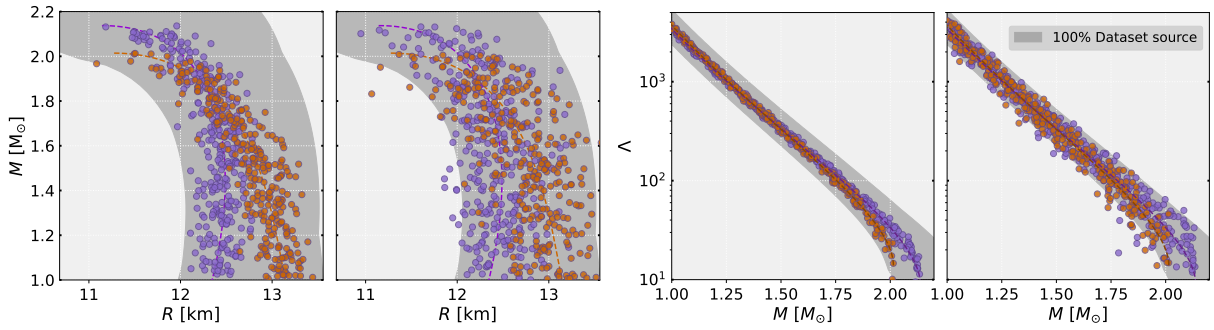


Figure 4.3: The $n_s = 60$ observations generated for two EoSs in the context of the $M(R)$ pairs contain dataset 1 (left) and dataset 2 (right), and in the case of the $\Lambda(M)$ pairs plots dataset 3 (left) and dataset 4 (right). The grey area represents the extremes (100%) of our EoS dataset (M-R and M-Λ curves without the noise). The two EoS coincide with the ones used in Fig. 4.1.

The description of the four datasets for the input is described in Appendix C, on the Tables C.1, and C.2. For the output the Tables are C.3.

One may ask but why did you choose the number of features to be 5 for each of the M, R, Λ elements and 15 for the output?

For the features, specifically the M-Λ pairs, we can say that is for future real data purposes, meaning, we know that we still don't have available five TD observations, still, we are hoping that in the near future, these will be discovered by the LIGO/Virgo/KAGRA[2] gravitational wave detectors, and from the next generation gravitational wave ground-based (e.g., Einstein Telescope and the Cosmic Explorer) and space-borne (e.g., LISA[3]) observatories.

When accounting for the M-R pairs, it is worth noting that in the case of BNNs, expanding the number of input pairs introduces a greater increase in the model parameters compared to traditional architectures, which is why we used a smaller amount of input pairs compared with previous articles using conventional NNs, as demonstrated in studies like [27] and related articles. Specifically, when increasing one pair of observations (2 neurons in the input layer), while a standard architecture, would only involve an increase of $2 \times n$ parameters, two weights multiplied by the number of forward neurons, n, the BNN architecture incurs an increase of $2n^2q + 4n^2 + 3n$ parameters, as is better explained in Section 3.4.3.

Now, let's turn our attention to the output. In assessing the optimal number of data points for our EoS, we employ a time-efficient yet robust method. We will focus solely on the speed of sound, although the same approach can be applied to the proton fraction. We begin by selecting a random EoS from our dataset and proceed to extract data points ranging from 3 to 22, denoted as $N_{out} = [3, \cdots, 22]$. These points are evenly spaced along the baryonic density axis, always falling within the range of $[0.15, 1.00]$ fm^{-3}. Consequently, our output vector takes the form $\boldsymbol{y} = \{v_s^2(n_k)\}_{k=1}^{N_{out}}$, with its size varying according to N_{out}. We perform linear interpolation to construct the $v_s^2(n)$ curve, and assess the sum of residuals - the differences between our dataset and the real EoS curve - at 1000 equally spaced baryonic density points, as illustrated in Fig. 4.4. The corresponding equation is defined as:

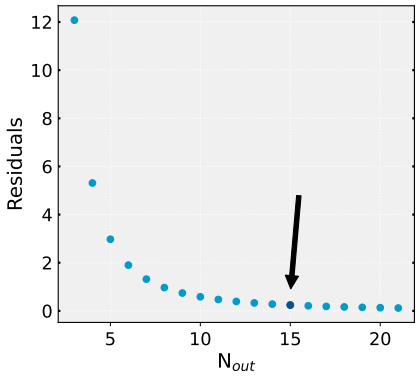


Figure 4.4: Residuals representation for Eq. 4.13.

$$Residuals = \sum_{t}^{1000} \frac{v_s^2(n_t) - \hat{v}_s^2(n_t)}{1000} \,. \tag{4.13}$$

Here, $\hat{v}_s^2(n_t)$ represents the actual curve, while $v_s^2(n_t)$ corresponds to the interpolated curve from our dataset. We decided that the best option that balances better the trade-off between accuracy and computational time consumption is for the 15 points of the output.

[2] Kamioka Gravitational Wave Detector
[3] Laser Interferometer Space Antenna

4.2.4 Correlation in neutron star data

To gain insight into the relationship between the input and output of our model, we present the Pearson correlation coefficient between $v_s^2(n)$ and $R(M)$ (left panel) as well as $\Lambda(M)$ (right panel), depicted in Fig. 4.5, for specific NS masses(indicated by different colors), along with the average value within the range $M/M_\odot \in [1, 2.2]$, represented in black squares. Similarly, we perform the same analysis for the proton fraction, as shown in Fig. 4.6. The Pearson correlation coefficient is calculated with

$$^{4}\mathrm{Corr}(a,b) = \frac{\sum_{i=1}^{\mathscr{D}}(a^{(i)}-\overline{a})(b^{(i)}-\overline{b})}{\sqrt{\sum_{i=1}^{\mathscr{D}}(a^{(i)}-\overline{a})^2}\sqrt{\sum_{i=1}^{\mathscr{D}}(b^{(i)}-\overline{b})^2}}\,, \tag{4.14}$$

where a comprises $v_s^2(n)$ and $y_p(n)$, and b represents R and Λ. It is important to note that this correlation measure is primarily sensitive to linear dependencies, potentially overlooking higher-order relationships.

Speed of sound

Examining the speed of sound, we can discern some notable patterns in Fig. 4.5. In the left panel, the speed of sound exhibits a more substantial correlation with radius, especially up to approximately four times the saturation density. Beyond this point, the correlation gradually diminishes, with the smallest correlation observed for lower masses. It's important to emphasize that our primary focus here isn't on how the correlation varies with mass within our dataset. This is due to the introduction of noise into the input vector, which shifts our attention more towards the mean correlation value indicated by the black squares. Turning our attention to the right panel, we encounter a similar pattern. Here, we again observe a stronger correlation, particularly up to approximately 0.4 fm^{-3}, followed by a declining trend. This behavior closely mirrors what we observed in the left panel.

For in-depth exploration of how the speed of sound correlates with radius and TD for different mass values, we recommend reading the work by Ferreira *et al.*[76]. They've covered the same results as us regarding how the speed of sound behaves with respect to the two NS observables we've considered, and its response to increasing baryonic density.

Proton fraction

Let's dive into the proton fraction analysis as depicted in Fig. 4.6. When we compare the left panel with the one obtained for the speed of sound, we observe that the correlation is weaker, and the highest correlation occurs earlier, specifically at the third density point. As we move to higher density values, this correlation progressively diminishes, eventually approaching zero.

In the right panel, we notice that the proton fraction exhibits an overall low correlation with the TD. Initially, this correlation is negative, but it transitions to a positive trend, reaching its approximate peak at around four times the saturation density. Subsequently, it gradually decreases towards zero values.

[4] We refer to this value as being big when between $0.8 \lesssim |\mathrm{Corr}(a,b)| \leq 1$, as being medium when $0.4 \lesssim |\mathrm{Corr}(a,b)| \lesssim 0.8$ and low for $0 < |\mathrm{Corr}(a,b)| \lesssim 0.4$.

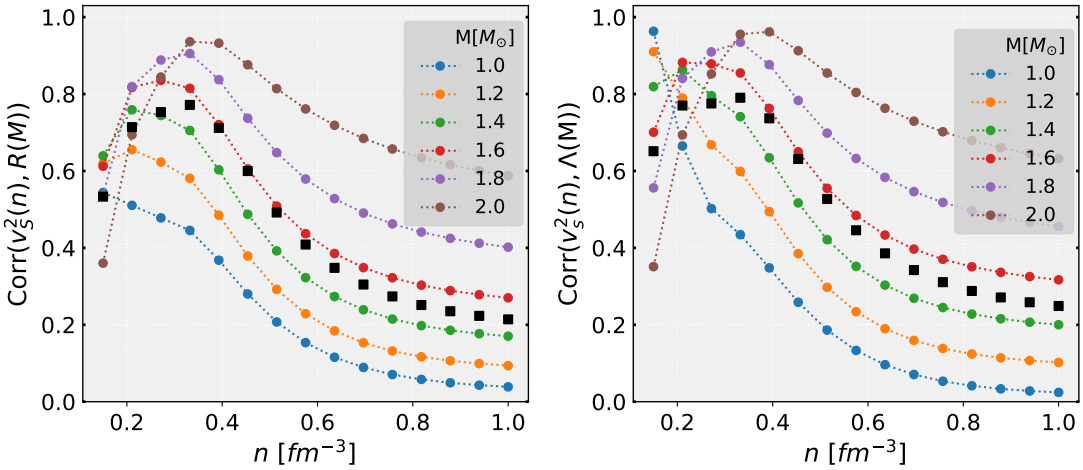


Figure 4.5: Correlation between $v_s^2(n)$ and $R(M)$ (left) or $\Lambda(M)$ (right) for fixed mass values. In black squares we show the mean correlation value for $M/M_\odot \in [1, 2.2]$.

Another study, described in [77], indirectly investigates the proton fraction and supports our findings. In this study, they look at how different nuclear matter parameters correlate with NS observables. Since the proton fraction correlates strongly with the symmetry energy in Eq. 2.3, the correlations between the symmetry energy, its slope, and curvature at saturation density with the radius and TD for varying mass values directly mirror the proton fraction behavior. In addition, how the proton fraction correlates with nuclear matter properties has been directly analysed in the study in [78].

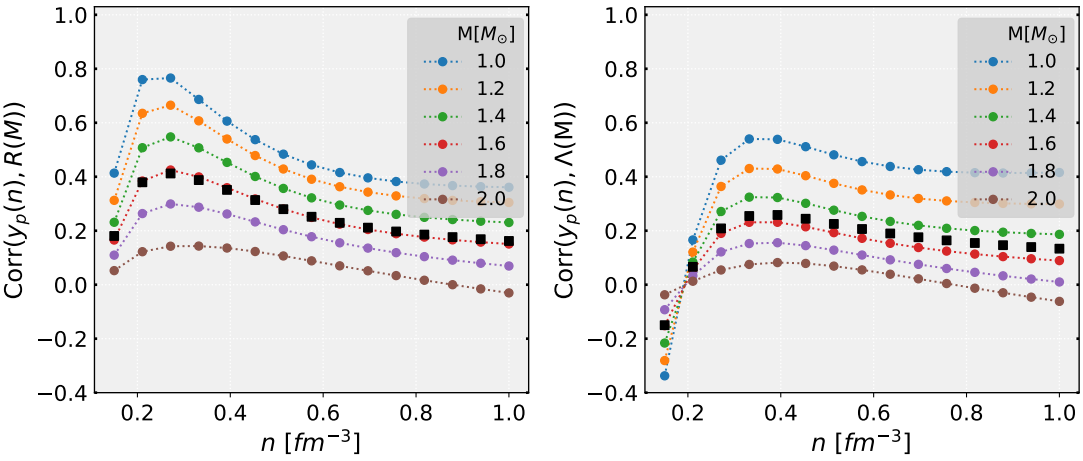


Figure 4.6: Correlation between $y_p(n)$ and $R(M)$ (left) or $\Lambda(M)$ (right) for fixed mass values. In black squares we show the mean correlation value for $M/M_\odot \in [1, 2.2]$.

5 Tuning Model Parameters

To assess how BNNs respond to varying levels of input noises and different output targets, we carried out a sequence of experiments involving the training of diverse functional and stochastic models. The details of these experiments are explained in Section 5.1. Furthermore, Section 5.2 discusses the determination of the optimal number of samples to be used during prediction.

5.1 Training procedure

To investigate the response of BNNs to varying input noises and output targets, we conducted a series of experiments involving the training of diverse functional and stochastic models, as elaborated in Section 3.4. The BNN models were trained using distinct datasets generated following the guidelines provided in Section 4.2. During the training stage, a portion of the training data was randomly chosen as a validation set. Specifically, we split the training data into an 80% portion for actual training and a 20% portion for validation. To ensure effective training, we normalized the input data for each of these subsets using the *standardscaler*[1] method available in the Python library Scikit-learn. This approach involves scaling the data to have a mean of zero and a standard deviation of one for each input feature. We chose this normalization technique because the range of values for each feature when using TD can differ significantly. By standardizing the data, the model can better comprehend the underlying data patterns, resulting in improved training efficiency. It's worth noting that the mean (μ) and standard deviation (σ) used for standardization in the training set were also applied to the test set, ensuring consistency in the normalization process [79]. This normalization strategy aligns with the practice of initializing weights close to zero, facilitating learning in the model as parameters will have a mean close to zero. The models were trained using the training subsets, and the validation set allowed us to independently assess whether the model was prone to overfitting.

5.1.1 The stochastic model

Regarding the stochastic model, we adopt a Gaussian prior with mean zero and standard deviation of one, which is not trainable, as mentioned in Section 3.4. While this prior choice lacks a specific theoretical justification, it serves as a reasonable default prior, as discussed in [61]. Future research could delve further into investigating the impact of prior parameters, similar to the approach taken

[1] $z = \frac{x-\mu}{\sigma}$, where μ stands for the mean of the training dataset and σ is the standard deviation of the training dataset.

in reference [41]. Additionally, we select a multivariate normal distribution as the variational posterior as explained in Section 3.4, initialized with mean 0 and a diagonal covariance matrix, where the standard deviation is equal to $\log(1+\exp 0) = 0.693$. Furthermore, we have a probabilistic output with a diagonal covariance matrix to capture aleatory heteroscedastic uncertainty, as this has shown better results in our experiments. All BNNs models were coded using TensorFlow library [65], more specifically we use Keras [80], an high-level API[2] of the TensorFlow.

5.1.2 The functional model

Defining the functional models involves adjusting the number of neurons, layers, and activation functions. For the hidden layers, we explore hyperbolic tangent, softplus, and sigmoid activation functions (Eqs. 3.3, 3.2 and 3.4 respectively), while utilizing a linear activation function for the output layer. As the input vector sizes differ (10 for sets 1 and 2, and 20 for sets 3 and 4), we employ more neurons per layer for larger input spaces. This is due to the increased complexity demanded by a greater number of parameters. Specifically, we use 15 and 10 neurons for sets 1 and 2, and 20 and 25 neurons for sets 3 and 4, respectively. The output layer consistently contains 30 neurons, with 15 representing the mean and 15 representing the standard deviation of the output probability distribution function. It is worth noting that we deliberately excluded the use of correlation in the output layer due to the inferior performance observed when attempting to incorporate it. As a result, the output layer is solely focused on capturing the mean and standard deviation information of the output distribution. The architecture employed in this study involved utilizing two to three hidden layers. The number of neurons within each hidden layer remained consistent, but it varied depending on the size of the input vector, as explained earlier. During the best model search process, we systematically explored four different architectures for each output. However, we narrowed our focus to datasets 1 and 3, as we specifically aimed to identify the most suitable architecture for the two different input sizes. The best outcomes are obtained by employing sigmoid as the activation function in the hidden layers, ensuring minimal loss and preventing divergence. Across all eight dataset configurations, i.e., two outputs ($v_s^2(n)$ and $y_p(n)$) and four datasets, see Table 4.4, it was found that the optimal number of hidden layers is two. For sets 1 and 2, the best performance was achieved with 15 neurons in each hidden layer, while for sets 3 and 4, 25 neurons were utilized in each hidden layer, as can be seen in the heatmaps for $v_s^2(n)$ and $y_p(n)$, Figs. 5.1 and 5.2 respectively, the validation loss is smaller for those architectures.

The final best architectures are then defined in Table 5.1 for the two output variables. During training, we employ a learning rate of 0.001 and utilize the ADAM optimizer [58] with the AMSgrad improvement [81]. The models are trained for 4000 epochs, with a mini-batch size of 768.

[2] Application Programming Interface

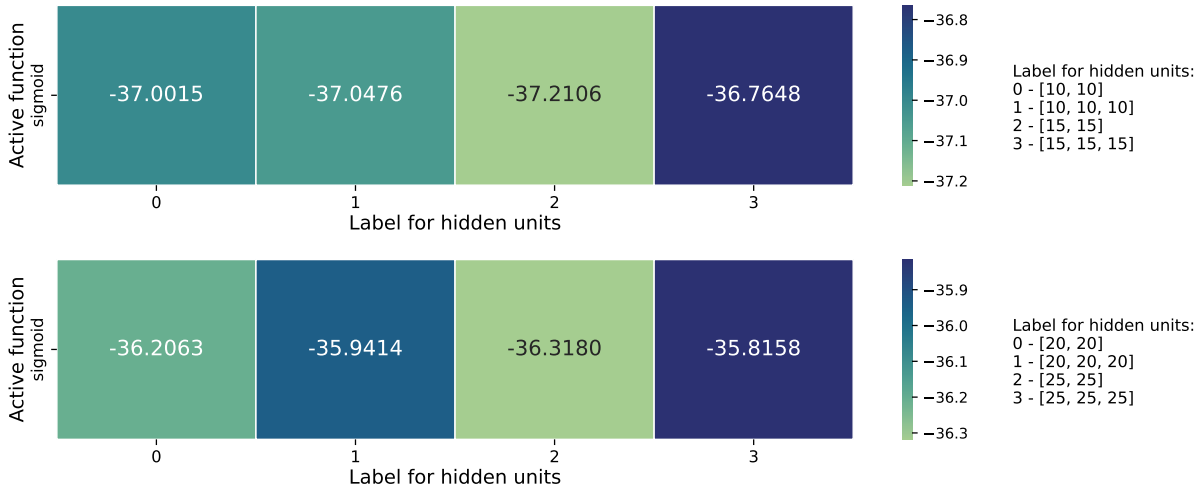


Figure 5.1: Tuning the hyperparameters of our model to find the most appropriate architecture from the four we defined for set 1 above and set 3 below for $v_s^2(n)$.

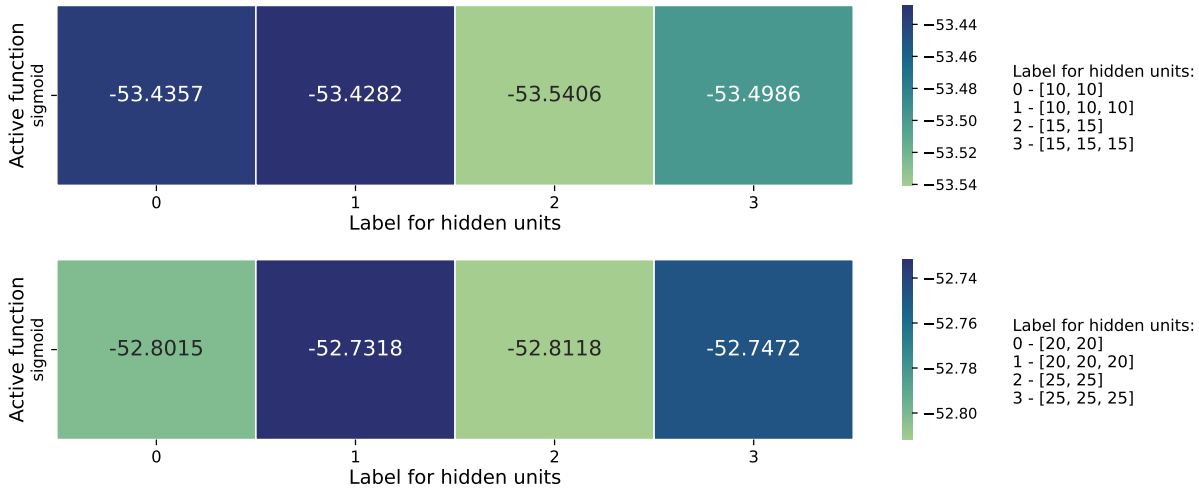


Figure 5.2: Tuning the hyperparameters of our model to find the most appropriate architecture from the four we defined for set 1 above and set 3 below for $y_p(n)$.

Table 5.1: Structures of the final BNN models. The $v_s^2(n)$ and $y_p(n)$ models have the same structure.

Layers	Activation	Neurons	
		Datasets 1 & 2	Datasets 3 & 4
Input	N/A	10	20
Hidden Layer 1	Sigmoid	15	25
Hidden Layer 2	Sigmoid	15	25
Output	Linear	30	30

A closer look into the loss function

To illustrate the usual pattern of our employed loss function, given by Eq. 3.38, over the course of 4000 epochs for our eight models, we provide a single example. This behavior remains consistent across the other models, with only variations in the value scale. For illustration, we have selected set 1 and shown the loss function curve for the proton fraction in Fig. 5.3.

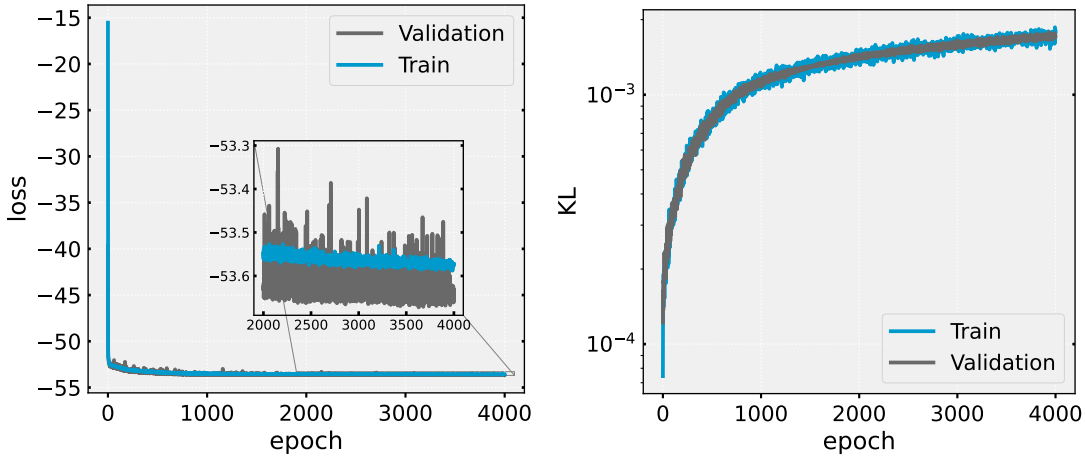


Figure 5.3: Evolution of a loss function of a BNN as a function of epochs during training for one of the sets of the model, the left plot shows the total loss and the right plot shows the KL divergence, where the y-axis is in logarithmic scale.

When analyzing the loss function, it's noticeable that the scale of the KL divergence part is significantly smaller than that of the total loss. This observation implies that the contribution of the KL term to the overall loss is minimal. Moreover, it's noticeable that the KL loss increases as the epochs progress, a behavior expected as it starts equal to the prior and then diverges from it. This trend eventually stabilizes and approaches a plateau, suggesting that the posterior distribution is reaching an optimal configuration. An important observation is the absence of overfitting, as evidenced by the close alignment of the validation and training curves. This indicates that the model is generalizing well and not merely memorizing the training data.

5.2 Bayesian neural network prediction trade-offs

When choosing the number of samples we would utilize in Eqs. 3.41 and 3.43, we wanted to strike a balance between time consumption and precision in future predictions. To understand how different sample sizes affect the convergence of our predictions, we conducted a brief experiment, for just one EoS of the test dataset, with up to 100 000 model, e.g., samples from the variational posterior $\boldsymbol{\theta}_n \sim q_\phi(\boldsymbol{\theta})$. It's important to note that this study focused on examining sample behavior for a specific set (set 2) and a particular output (v_s^2). Figure 5.4 displays the results, showing the relative error of the expected value computed for the 15 mean values within the output distribution, based on the number of samples we used. The computation of the discrepancy between the number of samples was executed using the subsequent equation

$$S(L_i) = \frac{\hat{\boldsymbol{\mu}}_{mean}(L_i) - \hat{\boldsymbol{\mu}}_{mean}(L_{i+1})}{\hat{\boldsymbol{\mu}}_{mean}(L_i)} \times 100\,, \tag{5.1}$$

where $\hat{\boldsymbol{\mu}}_{mean} = \frac{1}{15}\sum_{k=1}^{15} \hat{\boldsymbol{\mu}}(n_k)$ is representing the mean of the 15 mean values in the output vector and $L_i \equiv N$ corresponds to the number of samples we take from the variational posterior, more explicitly samples taken from Eq. 3.41.

L_i is the ith term in a set of elements described as $L = \{10^i + j \times 10^i | i \in \{2,3,4\}, j \in [0,9]\} \cup \{10^5\}$ [3] this notation means that L includes values starting from 100, increasing by 100 until it reaches 900, then jumping to 1000 and increasing by 1000 until it reaches 9000, and so on, until it reaches 100 000. Analysing the obtained results shown in Fig. 5.4, we marked a key point with a black star, that indicates the point at which it is present the range of L_i =10 000 to L_{i+1}=20 000 samples. Beyond this threshold, the predictions consistently converged towards zero values. Based on this analysis, we determined that employing 10 000 samples struck an optimal balance between computational efficiency and prediction accuracy for our experimental requirements.

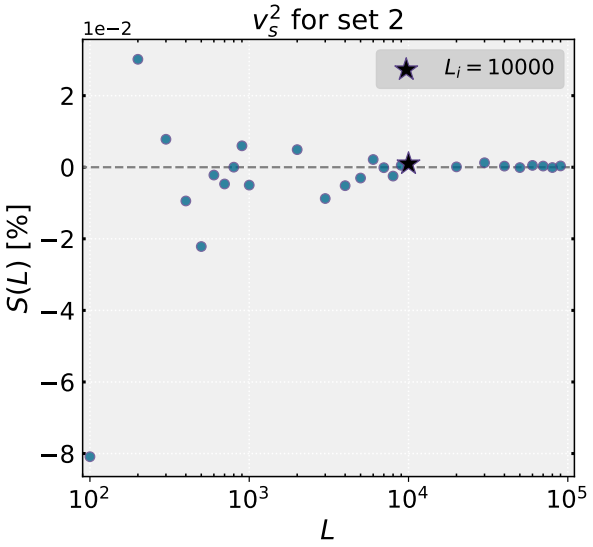


Figure 5.4: Visualization of the quantity $S(L_i)$ in Eq. 4.13, aimed at determining the optimal output size.

Our study predominantly emphasizes qualitative exploration rather than extensive quantitative analysis. Therefore, we did not delve deeply into quantitative aspects, as our primary objective was to extract qualitative insights rather than conduct an exhaustive quantitative investigation.

[3] $L = \{100, \cdots, 900, 1000, \cdots, 9000, 10000, \cdots, 90000, 100000\}$

6 Results and Discussion

With a solid foundation in the theories outlined in the previous Chapters 2 and 3, as well as an understanding of our dataset creation and model structure Chapters 4 and 5, we are now ready to unveil the outcomes of our study. This section will provide a comprehensive overview of the results obtained for both the speed of sound squared ($v_s^2(n)$) and the proton fraction ($y_p(n)$).

We will initiate our exploration by conducting a comprehensive analysis of the predicted distributions for an individual EoS within the test set, followed by an examination of the trends exhibited by the entire test set, Section 6.1. Subsequently, we will delve deeper into the uncertainty decomposition, Section 6.2. Lastly, we will investigate the performance of our model when faced with data generated from a distinct microscopic framework, Section 6.3.

Note: For the sake of simplicity, whenever we want to distinguish the different 8 BNNs models, we will, hereafter, refer to their training (data)sets, 1 to 4, and predicting quantity, $v_s^2(n)$ or $y_p(n)$. For instance, when we say the results of dataset 2 for $y_p(n)$, we mean the BNN model trained on dataset 2 that has the proton fraction as the target quantity.

6.1 Overall study of the predicted distributions

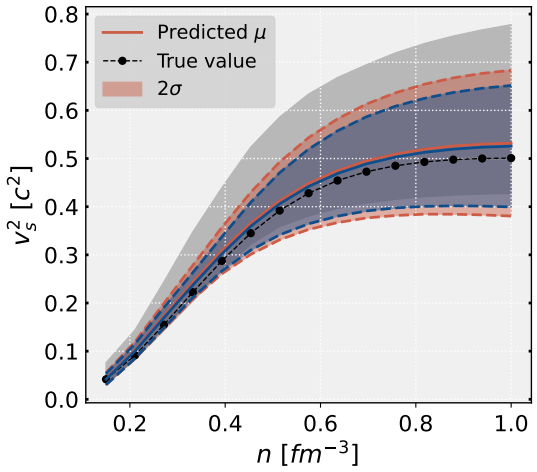
6.1.1 Speed of sound

Let's delve into the distinct predictions made by our BNNs model for the speed of sound squared using a randomly selected EoS from the test set. The results are presented in Fig. 6.1. In the left panel, you can observe predictions from sets 1 (blue) and 2 (pink) in Fig. 6.1a, while the right panel displays predictions for sets 3 (purple) and 4 (green) in Fig. 6.1b. These predictions are generated using Eq. 3.25, and we visualize the mean values (solid lines) along with the 2σ regions (colored areas).

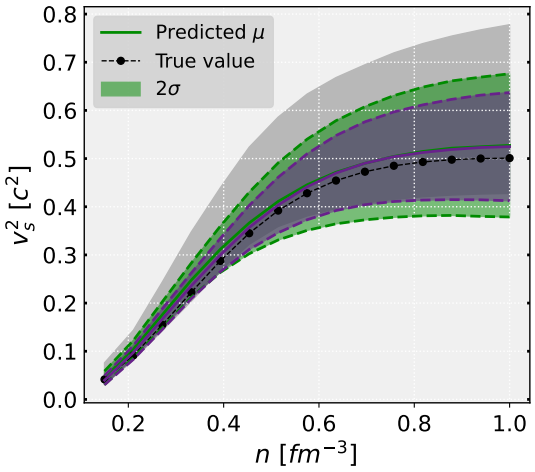
The left plot clearly illustrates that the prediction uncertainty, represented by the standard deviation of the distribution σ, is smaller in set 1 than in set 2. Moreover, the predicted mean values are closer to the actual values for set 1. The same pattern is observed in the right figure, where set 3 exhibits lower prediction variance than set 4.

Although a comprehensive understanding of the overall behavior requires analyzing the entire test set, Fig. 6.1 already suggests that the BNN models are capable of capturing the characteristics of different datasets (as detailed in Table 4.4). Specifically, the increased spread of NS observations around their true values results in more significant uncertainty when inferring the corresponding

properties of the EoS.



(a) Trained on datasets 1 (blue) and 2 (pink).



(b) Trained on datasets 3 (purple) and 4(green).

Figure 6.1: The BNNs predictions for $v_s^2(n)$ using one EoS of the test. The prediction mean values (solid lines) and 2σ confidence intervals are shown. The true values are shown in black dots and the range of $v_s^2(n)$ from the train set is indicated by the grey region.

In order to investigate the models predictions over the entire test set, we defined the normalized residuals' predictions as $\Gamma(n_k)$, the model residuals $\delta(n_k)$ and the dispersion by $\Sigma(n_k)$ at each of the prediction densities, i.e., $k = 1, ..., 15$,

$$\Gamma(n_k) = \left(v_s^2(n_k) - v_s^2(n_k)^{\text{true}}\right)/\sigma(n_k)\,, \tag{6.1}$$

$$\delta(n_k) = v_s^2(n_k) - v_s^2(n_k)^{\text{true}}\,, \tag{6.2}$$

$$\Sigma(n_k) = \sigma(n_k)\,. \tag{6.3}$$

A summary statistics of both quantities over all EoS of the test set is shown in Fig. 6.2.

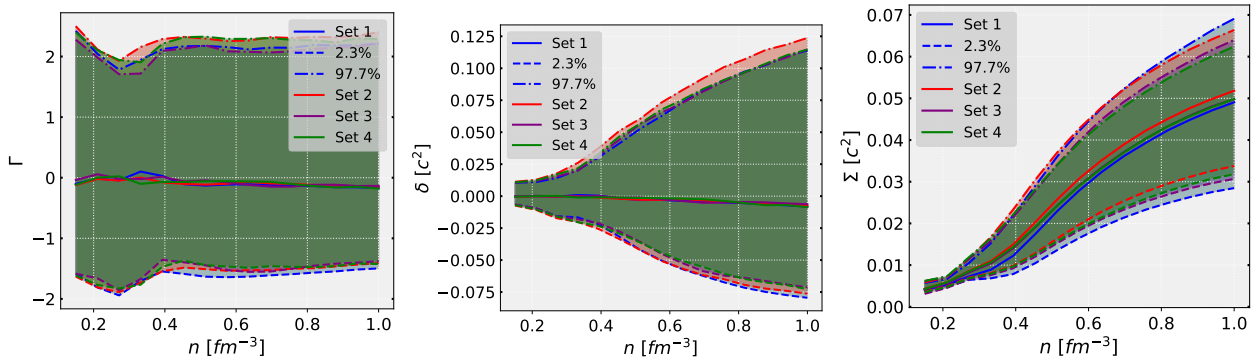


Figure 6.2: Median (solid line), 95.4% confidence interval (dashed and dotted lines), and extreme values (region) for $\Gamma(n)$ (left), $\delta(n)$ (center) and $\Sigma(n)$ (right) for each dataset.

The distribution of $\Gamma(n)$, as shown in the left panel across all four datasets, exhibits interesting properties. Approximately 50% of the values (indicated by the solid line) are clustered near zero,

suggesting that the median of the prediction values is unbiased. Additionally, when we examine the distribution at the 2.3% and 97.7% cumulative percentiles, we find that it falls within the range of -2σ and 2σ, respectively. This indicates that the prediction mean deviates from the true value by less than 2σ in 95.4% of the cases. What's particularly noteworthy is that the distribution characteristics of $\Gamma(n)$ remain consistent across all datasets and are independent of the density. This observation highlights that the BNN models accurately capture the dispersion of predictions while considering the corresponding mean residuals at each density value.

When examining the distribution of $\delta(n)$ (middle panel), it becomes evident that the median remains closely centered around zero as the density increases. However, the distribution becomes significantly wider for the 95.4% of the values. This phenomenon is directly linked to the behaviour observed in $\Sigma(n)$ (right panel) that also gets broader with the increase of density. This behavior is a direct reflection of the statistics present in the training dataset, since the EoS dataset was generated through Bayesian inference, where saturation properties were imposed, leading to a wider uncertainty at higher densities while low density regions are strongly constrained.

Upon examining the values of $\Sigma(n)$ (right panel), a noticeable shift to lower values is observed in the entire distribution of the BNN model trained on set 1(indicated by the blue line) specifically on 2.3% and 50% of the values, showing that there is a considerable decrease in uncertainty when the dispersion of NS observations is reduced by a factor of two, from ($\sigma_M = 0.1 M_\odot, \sigma_R = 0.3$ km) to ($\sigma_M = 0.05 M_\odot, \sigma_R = 0.15$ km) (see Table 4.4).

To assess the overall performance of the various BNN models, we've calculated their coverage probabilities, as illustrated in Fig. 6.3. Coverage probability serves as a metric to account for how effectively each model captures the data distribution. It does this by examining whether the percentage of values falling within 1σ of the output distribution – which is the number of values within a specific interval divided by the total number of values times one hundred percent – corresponds to 68.3% of the total number of values in the test set. This same evaluation is repeated for 2σ (95.4%) and 3σ (99.7%) intervals. We applied this procedure independently to each of the 15 output values, resulting in coverage probabilities relative to the output densities (as shown in Fig. 6.3 on the right). We then calculated the mean of these 15 coverage probabilities for each of the 4 sets, as depicted in Fig. 6.3 on the left. For the left plot the bars are closely aligned with the three specified percentages (68.3%, 95.4%, and 99.7%). However, a slight fluctuation is noticeable in the 68.3% region, where all four sets tend to overestimate the result uncertainties. This suggests that the models predict a percentage of data greater than 68.3% within the 1σ model interval. This overestimation is more pronounced in set 3. Examining the coverage probabilities for each density point, it becomes evident that uniformity is not consistently maintained across all densities. This non-uniform behavior is particularly pronounced at low densities, more specifically for the density points $\{n_1, n_3, n_4, n_5\}$, indicating that the model tends to slightly overestimate the uncertainty. This discrepancy, can possibly be attributed to the significant variation in standard deviation with increasing density, as evident in Table C.3, where the standard deviation ranges from approximately 0.005 c^2 to 0.06 c^2. The model, it seems, struggles to adequately reduce the standard deviation at lower densities. In summary, the results indicate that the model accurately estimates the data distribution above the 95.4% results.

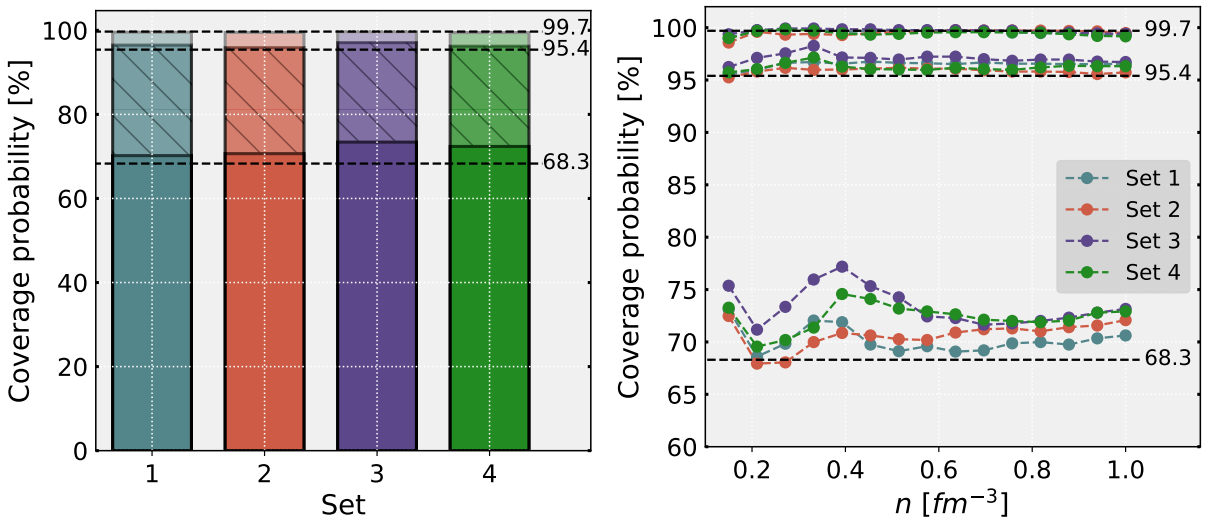


Figure 6.3: Coverage probability calculated on the test set of the $v_s^2(n)$ BNNs models for the individual densities (right) and for the mean of these densities (left).

To quantify how the increase in the observational scattering of (M, R, Λ) affects the model prediction uncertainties, let us define the following quantity

$$\eta\,[a,b]\,(n_k) = \frac{1}{T_E}\left(\sum_{i=1}^{T_E}\frac{\sigma_i^a(n_k)-\sigma_i^b(n_k)}{\sigma_i^b(n_k)}\right)\times 100, \tag{6.4}$$

where T_E is the total number of EoS in the test set, and $k = 1, ..., 15$. This quantity defines the percentage uncertainty deviation between models b and a at density n_k. Figure 6.4 shows the results, where we plotted 4 different comparisons.

Let's take a closer look at the results for $\eta(2,1)$ (in cyan). We observe that the prediction uncertainty increases when we consider the BNN model from dataset 1 compared to the one from dataset 2. Moreover, $\eta(2,1)$ reaches its maximum value of 20% at $n_4 = 0.33$ fm^{-3}. In simpler terms, when the synthetic observational data scattering doubles, going from ($\sigma_M = 0.05 M_\odot, \sigma_R = 0.15$ km) to ($\sigma_M = 0.1 M_\odot, \sigma_R = 0.3$ km), the uncertainty increases by approximately 4.9-20%. This increase is most prominent at densities where there is a strong correlation between the NS radius, $R(M)$, and the squared speed of sound, $v_s^2(n)$. More detailed information on this correlation is present in Fig. 4.5 and ref. [76].

The second major conclusion concerns the impact of including Λ information into the inference process. This becomes evident when we analyze the behavior and values of $\eta\,[3,2]$ (where the only difference between datasets 2 and 3 is the inclusion of tidal deformability information, as indicated in Table 4.4). The negative values signify that prediction uncertainty decreases when tidal information is added to the training process, as tidal deformability provides informative insights into $v_s^2(n)$ for NS matter. The maximum reduction in uncertainty is 7%, occurring at $n_6 = 0.45$ fm^{-3}. Considering $\eta\,[4,2]$ (blue), we once again observe negative values, indicating a reduction in uncertainty compared to dataset 2. However, beyond the seventh density, both values converge closely, suggesting that introducing more dispersion in the tidal deformability data does not provide significant additional in-

formation. Comparing $\eta[4,3]$ with $\eta[2,1]$, we notice that initially, they differ very little. However, as we move to higher densities, $\eta[2,1]$ consistently becomes more than 5 times larger. It's important to note a key factor here: the proportion of input values that have been altered. Dataset 3 and 4 involve changing the uncertainty of only 5 out of the 20 input values, essentially modifying a quarter of our input vector. In contrast, datasets 1 and 2 have modifications across all input values. This implies that the dispersion of the mass-radius pairs has a more significant impact on the model than the mass-tidal deformability pairs. Recognizing this difference in the proportion of altered input values provides valuable insights into how the model responds to changes in dispersion.

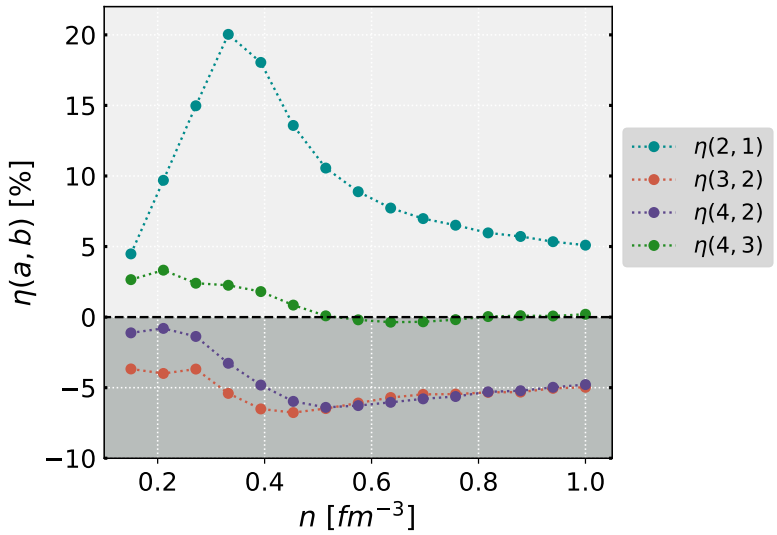


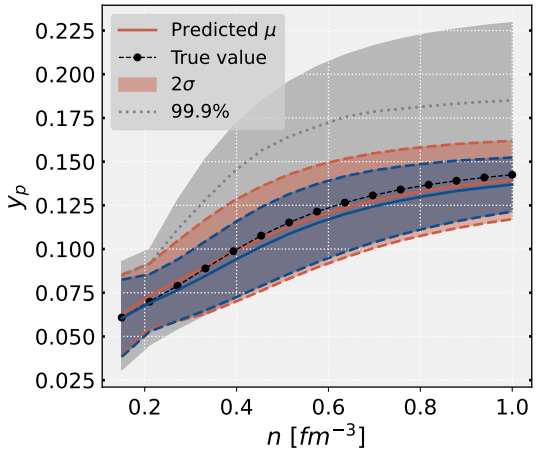
Figure 6.4: Prediction uncertainty deviation $\eta[a,b]$ between the $v_s^2(n)$ BNN models a and b (see text for details).

6.1.2 Proton fraction

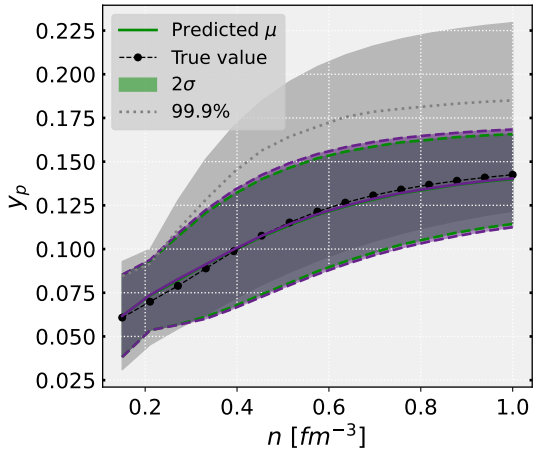
Now, let's delve into the model's predictions for the proton fraction, $y_p(n)$. To illustrate, we've selected a specific EoS from the test dataset. In Fig. 6.5, we present the model's predictions for each dataset: Fig. 6.5a displays sets 1 (blue) and 2 (pink), while the right panel showcases sets 3 (purple) and 4 (green). The grey region represents the range of $y_p(n)$ values from the training set, and the dashed grey line corresponds to the 99.9% probability line. It's essential to note that the upper region, between the 99.9% probability and the maximum boundary lines, is influenced by the presence of a single "extreme" EoS.

The observations from Fig. 6.5 mirror the findings from the $v_s^2(n)$ predictions (Fig. 6.1). Specifically, dataset 1 exhibits the narrowest prediction uncertainty, while dataset 2 shows a substantial increase in uncertainty. Predictions for datasets 3 and 4 yield similar uncertainty levels. However, to draw definitive conclusions, we need to perform a statistical analysis across the entire test set.

In Figure 6.6, we examine the same parameters as in the analysis of the speed of sound but focusing on the proton fraction ($y_p(n)$) across the four datasets. These parameters include the model residuals ($\delta(n_k)$) displayed on the left side, calculated using Eq. 6.2, the standard deviation ($\Sigma(n_k)$)



(a) Trained on datasets 1 (blue) and 2 (pink).



(b) Trained on datasets 3 (purple) and 4(green).

Figure 6.5: The BNNs predictions for $y_p(n)$ using one EoS of the test. The models trained on datasets 1 (blue) and 2 (orange) are in the left figure while datasets 3 (purple) and 4 (green) models are in the right figure. The predicted mean values (solid lines) and 2σ confidence intervals are shown. The true values are shown in black dots.

shown in the center, computed as per Eq. 6.3, and the normalized model residuals ($\Gamma(n_k)$) presented on the right side, following the calculation outlined in Eq. 6.1.

The results reveal that the model exhibits a wider spread of residuals, particularly in the region around 0.4-0.5 fm^{-3}. However, this increased spread is counteracted by larger standard deviation values in the same region. In essence, this indicates that the model adeptly captures and represents data statistics. Specifically, the BNN models effectively account for increased prediction uncertainties in regions where the proton fraction ($y_p(n)$) displays a higher degree of dispersion, as anticipated.

The quality of the models' predictions is exemplified by the consistent $\Gamma(n)$ statistics (right panel), where the models' residuals fall within 2σ approximately 95.4% of the time, irrespective of density. For further insights into these statistics, we refer to Chapter 4, specifically Fig. 4.1, which provides an interpretation of $\delta(n)$ and, notably, $\Sigma(n)$ behaviors. This analysis reveals that the standard deviation $\sigma(n)$ of the training set exhibits a non-monotonic pattern, peaking at approximately 0.5 fm^{-3} and decreasing for lower and higher densities.

Examining the coverage probability shown in Fig. 6.7, we can see that the results for each set, as presented in the left plot, accurately represent the output distribution. Further analysis of the individual densities, as shown in the right plot, reveals minimal oscillation, which is even less pronounced than what was observed for the speed of sound. This demonstrates the model's consistent ability to predict uncertainty as the density values increase, supporting our findings for the $\Gamma(n)$ quantity in Fig. 6.6.

We are now going to analyse the percentage uncertainty deviation between the models with $\eta\,[a,b]\,(n)$ (see Eq. 6.4) displayed in Fig. 6.8. Firstly, we observe that $\eta\,[2,1]$ follows a pattern quite similar to that of $v_s^2(n)$. However, it attains its maximum value earlier at n_3. This observation aligns precisely with the point where we've previously noted the highest correlation between $R(M)$ and $y_p(n)$, as dis-

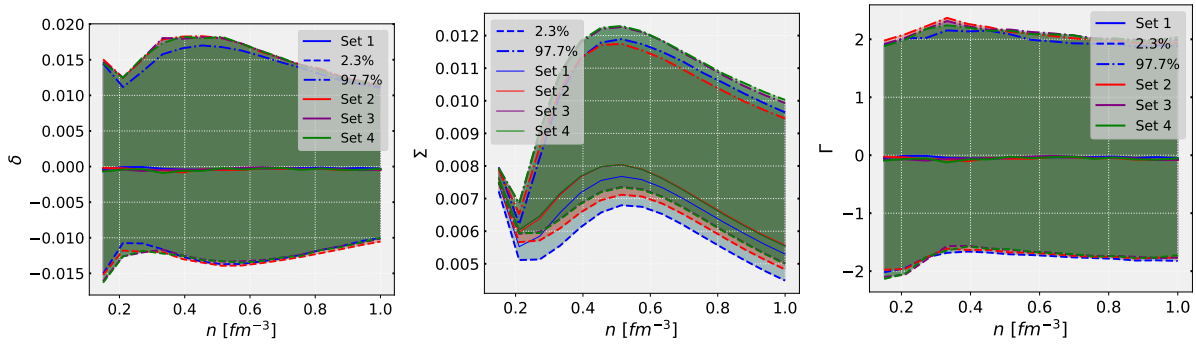


Figure 6.6: Median (solid line) and the 95.4% confidence interval (dashed and dotted lines) for $\delta(n)$ (left), $\Sigma(n)$ (center), and $\Gamma(n)$ (right).

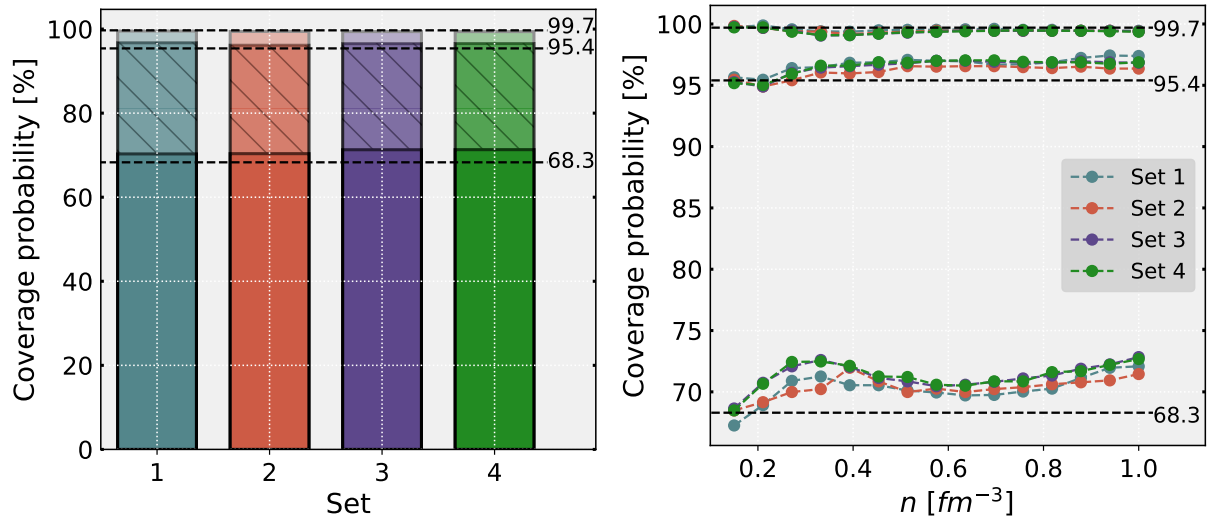


Figure 6.7: Coverage probability calculated on the test set of the $y_p(n)$ BNNs models for the individual densities (right) and for the mean of these densities (left).

cussed in Section 4.2.4. Now, when we compare $\eta[2,1]$ with $\eta[4,3]$, we find that the behavior mirrors what we've seen in the speed of sound dataset. However, this time, the relationship is even more pronounced. This pronounced behavior circles back to the correlation, which is nearly zero for $\Lambda(M)$ and $y_p(n)$.This observation raises the possibility that Λ may not contribute significantly to the model, especially since, this time, the ratio between $\eta[4,3]$ and $\eta[2,1]$ deviates even further from the expected one-quarter proportion due to the change in the input vector, compared to what we observed in the case of $v_s^2(n)$. Increasing the model's complexity without adding substantial information can lead to heightened confusion and greater uncertainty. This phenomenon is evident in $\eta[3,2]$ and $\eta[4,2]$, where both cases show positive η values. These positive values indicate higher uncertainty for sets 3 and 4 when compared to set 2.

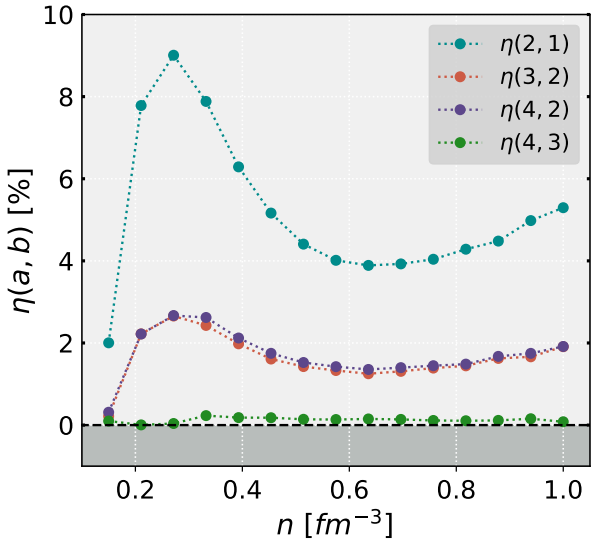


Figure 6.8: Prediction uncertainty deviation $\eta\,[a,b]$ between the $y_p(n)$ BNN models a and b (see text for details).

6.2 Understanding uncertainty components

In this section, we will delve deeper into understanding the inherent uncertainty within our predictions. Let's take a closer look at how the prediction uncertainty of BNNs is characterized, breaking down its distinct components [82]. As mentioned in Eq. 3.43 of Section 3.4, the prediction variance $\hat{\boldsymbol{\sigma}}^2$ is composed of two distinct terms:

$$\hat{\boldsymbol{\sigma}}^2 = \hat{\boldsymbol{\sigma}}^2_{\text{alea}} + \hat{\boldsymbol{\sigma}}^2_{\text{epist}},$$

Here, the aleatoric uncertainty $\hat{\boldsymbol{\sigma}}^2_{\text{alea}}$ signifies the average variance across the ensemble of models, while the epistemic uncertainty $\hat{\boldsymbol{\sigma}}^2_{\text{epist}}$ denotes the dispersion of ensemble models around the ensemble mean $\hat{\boldsymbol{\mu}}$. Epistemic uncertainty emerges from limited information or data and is encapsulated within the posterior probability $P(\boldsymbol{\theta}|D)$, reflecting the distribution of the model. On the other hand, aleatoric uncertainty emerges due to the inherent randomness of the dataset and is encoded within the data likelihood $P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta})$. While epistemic uncertainty diminishes with an increased amount of available data, aleatoric uncertainty remains constant as it stems from the intrinsic randomness of the data generation process itself.

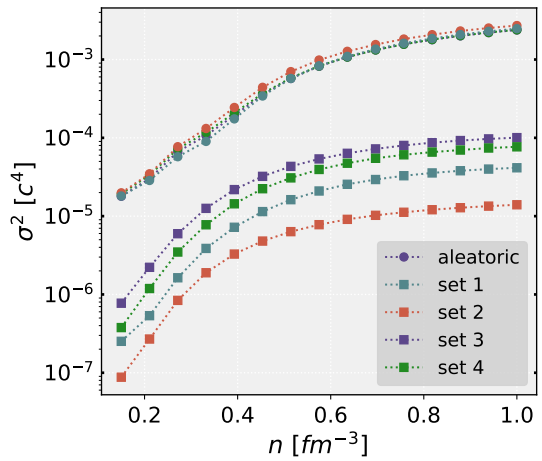
6.2.1 Speed of sound

Let's take a closer look at how our model interprets the two types of uncertainty in the context of speed of sound predictions, as depicted in Fig. 6.9. Starting with Fig. 6.9a, it's important to note that the epistemic uncertainty is consistently around one order of magnitude smaller than the aleatoric uncertainty. Both aleatoric and epistemic uncertainties follow a similar pattern across various density points. In the case of aleatoric uncertainty, it's behaviour mirrors the dataset's characteristics, which is expected since it's tied to the inherent randomness of the data itself.

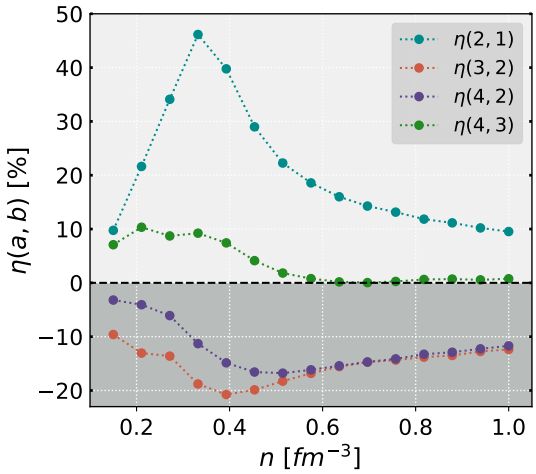
To gain a deeper insight into the behavior of epistemic uncertainty, we calculate the epistemic percentage. This metric helps us analyze the proportions of both types of uncertainty within the total prediction variance. The epistemic percentage, denoted as f_{epist}, is computed as $f_{\text{epist}} = (\hat{\boldsymbol{\sigma}}^2_{\text{epist}}/\hat{\boldsymbol{\sigma}}^2) \times 100\%$, as shown in Fig. 6.9c. What this figure reveals is that although epistemic uncertainty increases with higher baryonic density, it's essential to contextualize this increase within the scale of the overall uncertainty. Notably, the highest percentage of epistemic uncertainty occurs at density points where there's a stronger correlation between NS radius ($R(M)$) and the squared speed of sound ($v_s^2(n)$) as demonstrated in Fig. 4.5. This can be duo to the fact that when constructing the predicting ensemble, $P(\boldsymbol{y}^*|\boldsymbol{x}^*, D) = \frac{1}{N}\sum_{n=1}^{N} P(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}_n)$, by sampling from the variational posterior, $\boldsymbol{\theta}_n \sim q_\phi(\boldsymbol{\theta})$, the density points n_k with larger correlation with $v_s^2(n)$ are much more sensitive to model sampling than other density points where correlations are much smaller.

An intriguing observation in Fig. 6.9a is that epistemic uncertainty consistently tends to be greater for sets 3 and 4, in contrast to aleatoric uncertainty, which is more pronounced for sets 1 and 2. This suggests that the inclusion of TD measurements adds valuable information to the dataset, although it doesn't significantly impact the functional aspects of the model. Moreover, the trend in epistemic uncertainty is a reduction from set 1 to set 2 and another decrease from set 3 to set 4. This pattern implies that as aleatoric uncertainty captures the uncertainty inherent in the data and increases with it, epistemic uncertainty decreases. This phenomenon might be attributed to the model's ability to better capture the functional aspects when provided with more sparsely distributed input data points.

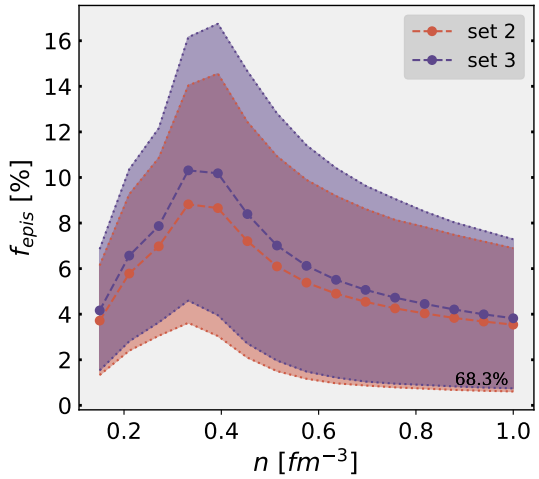
Now, turning our attention to Fig. 6.9b, we focus solely on the aleatoric component to provide a clearer view of how aleatoric uncertainty changes with each set. One essential aspect to note here is that the scale of η is considerably larger because we are dealing with variance. If not for this scaling, it would roughly share the same scale as the overall uncertainty depicted in Fig. 6.4. Upon closer examination, it becomes evident that aleatoric uncertainty in η essentially follows the same behavior observed in the overall uncertainty in Fig. 6.4.



(a) Epistemic (squares) and aleatoric (circles) uncertainty representation, where the y axis is in logarithmic scale, for the mean of the EoSs in the test set.



(b) Prediction for only the aleatoric uncertainty of $\eta[a,b]$, Eq. 6.4, but using σ^2_{alea}, so variance instead of standard deviation.



(c) The distribution of $f_{\text{epist}} = (\hat{\boldsymbol{\sigma}}^2_{\text{epist}}/\hat{\boldsymbol{\sigma}}^2) \times 100\%$ in sets 2 and 3 for mean values (dashed lines) and the 68.3% confidence interval regions (colored areas) of the EoS set.

Figure 6.9: Studying the aleatoric and epistemic uncertainty for $v_S^2(n)$, more details are given in the text.

6.2.2 Proton fraction

Analyzing the behavior of the proton fraction, as depicted in Fig. 6.10, we observe that the epistemic uncertainty is notably smaller than the aleatoric uncertainty, even more so for this particular quantity. The ranking of sets in terms of epistemic uncertainty follows the same order as that observed for the speed of sound. Therefore, similar conclusions can be drawn: uncertainty decreases with increasing input uncertainty. Additionally, as seen before, epistemic uncertainty increases with the incorporation of the TD, but so does aleatoric uncertainty. The behavior of aleatoric uncertainty mirrors the

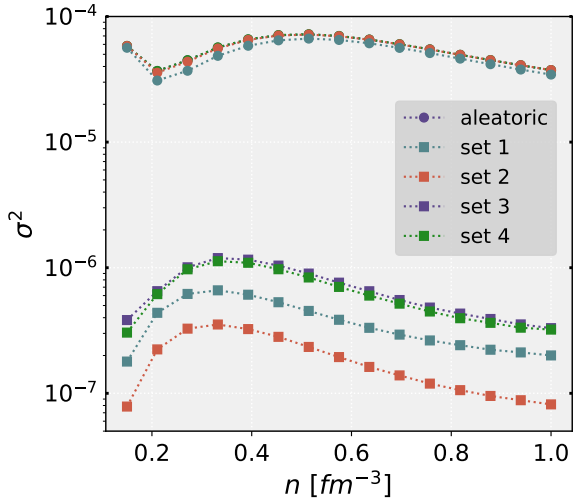


Figure 6.10: Representation of epistemic (squares) and aleatoric (circles) uncertainty, with a logarithmic scale on the y-axis, based on the mean values derived from the EoSs in the test set for $y_p(n)$.

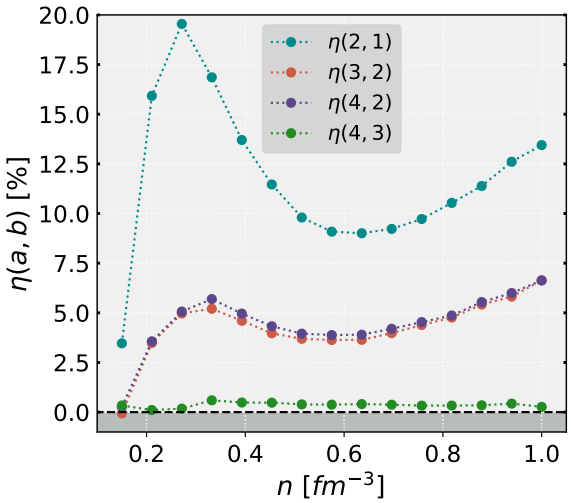


Figure 6.11: Prediction for the aleatoric uncertainty of $\eta[a,b]$, Eq.6.4, utilizing σ^2_{alea} (variance) instead of only σ (standard deviation) to maintain mathematical accuracy, for $y_p(n)$.

dataset statistics, once again as expected.

Turning our attention to Fig. 6.11, which focuses solely on aleatoric uncertainty (with variance as the scale), we observe a behavior closely aligned with that seen in the overall uncertainty in Fig. 6.8.

Utilizing the BNN models for $y_p(n)$, we present in the left panel of Fig. 6.12 both the mean values (dashed lines) and the 68.3% confidence interval regions (colored areas) for f_{epist} across the entire test sets (sets 2 and 3). This plot leads to a notable conclusion: the prediction variance $\hat{\boldsymbol{\sigma}}^2$ is primarily composed of aleatoric uncertainty (left panel), accounting for roughly 97% of the total uncertainty. This amount of aleatoric uncertainty is a consequence of the already substantial number of observations ($n_s = 60$), although it is comparatively smaller for the speed of sound. Furthermore, we observe that f_{epist} is lower for set 2 (depicted in orange) than for set 3 (in purple). This variation in epistemic uncertainty can be attributed to the increase in input dimensions, rising from 10 to 20, a change that has a noticeable impact on the variational posterior $q(\boldsymbol{\theta})$. Lastly once again as was seen for the speed of sound the percentage of epistemic uncertainty is larger at densities $0.2 - 0.4$ fm $^{-3}$ duo to the correlation as previously explained: density points n_k that exhibit stronger correlations with $y_p(n)$ are significantly more influenced by model sampling than other density points where correlations are substantially weaker. The right panel of Fig. 6.12 provides insight into f_{epist} using BNN models trained on set 1 while varying the number of mock observations: $n_s = 5$, $n_s = 20$, and the value employed in this study, $n_s = 60$. It's evident that as we decrease the number of observations, represented by n_s, and consequently, the overall number of training points, the epistemic uncertainty increases as anticipated. It's essential to recognize that the epistemic uncertainty tends to approach zero as the data points approach infinity, a concept discussed in more detail in [60].

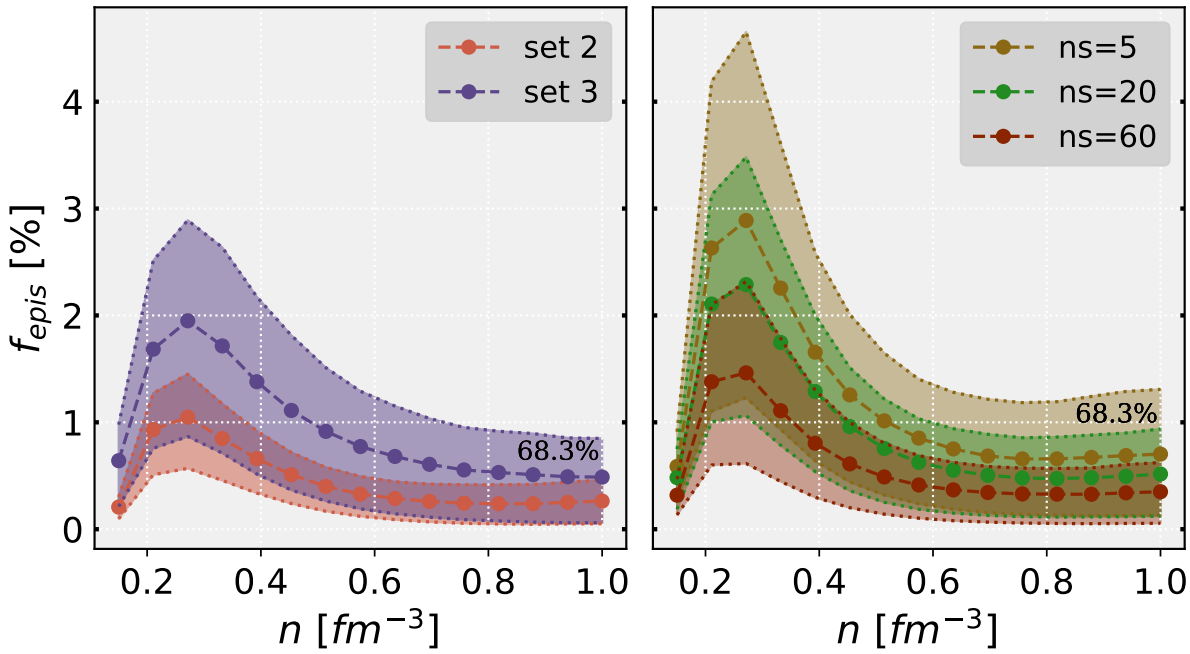


Figure 6.12: Distribution of $f_{\text{epist}} = (\hat{\boldsymbol{\sigma}}^2_{\text{epist}}/\hat{\boldsymbol{\sigma}}^2) \times 100\%$ for $y_p(n)$ BNN models in sets 2 and 3 (left) and for set 1 model but trained in different training datasets with different number of observations n_s (right).

6.3 Model performance under a different framework

For our final test, we want to see if our BNN model, previously trained on set 1 (refer to Table 4.4), is able to correctly estimate the results of a nuclear model with different properties from the ones used to train, in particular, obtained within a different microscopic description of nuclear matter. Specifically, we employed the DD2 model, a generalized RMF model featuring density-dependent couplings, and not containing the non-linear terms present in Eq. 2.10, [83].

Two significant distinctions set the DD2 model apart from the RMF family we employed to generate the set of EoS. Firstly, the DD2 model exhibits distinct high-density behavior in the symmetry energy. In the DD2 model, the coupling to the ρ-meson, which defines the isovector channel of the EoS, diminishes at sufficiently high densities. This behavior promotes the existence of highly asymmetric matter and has a notable consequence: the DD2 model does not predict nucleonic direct Urca processes inside NSs [84, 85]. Secondly, another notable difference lies in the behavior of the speed of sound with density between the DD2 class of models and the model class employed to train the BNN. In DD2-like models, the speed of sound consistently increases with density but remains well below the speed of light (c). In contrast, the class of models used for BNN training exhibits a different trend: the speed of sound flattens or even decreases beyond approximately three times nuclear saturation density (n_0). These two distinctions are expected to influence the performance of the BNN model.

Once the DD2 EoS was selected, we applied the statistical procedure outlined in Section 4 to generate a single observation ($n_s = 1$) using the properties of dataset 1. This dataset has lower σ_R and lacks information about Λ. In Fig. 6.13, it's possible to see the BNN model's predictions for the speed

of sound (left panel) and proton fraction (right panel). Although the DD2 EoS lies outside the training data range (grey region) for the speed of sound, the model's prediction uncertainty extends beyond the training maximum values and almost entirely encompasses the DD2 results. The prediction for $y_p(n)$ is quite accurate, with the mean value prediction closely matching the actual values. It is worth noting that this accuracy may be higher as the actual value falls within the trained region. While the results are indeed promising, it's important to highlight some key considerations. In the test phase mentioned above, we generated only one observation ($n_s = 1$) from the DD2 $M(R)$ curve, simulating a scenario with very limited access to NS observations. The process of generating a single observation ($n_s = 1$), which consists of five $M_i(R_i)$ values, is inherently random. As a result, different samples will yield different predictions. It's worth noting that $v_s^2(n)$ is more sensitive to this variability than $y_p(n)$, primarily because the DD2 target, for the last quantity, falls entirely within the training values region.

This variation in predictions is somewhat expected because we are attempting to characterize the entire $M(R)$ curve with only five randomly selected (M, R) values. While one sample may provide sufficient information about the general behavior of the $M(R)$ curve, others might be less informative, such as a sample where all five observations $M_i(R_i)$ cluster around the same M value. This challenge is a fundamental issue when inferring the EoS from a very limited number of NS observations, and it applies regardless of the inference model or framework used. The reliability of the BNN's performance assessment for the DD2 EoS would significantly improve with an increase in the number of points $M_i(R_i)$ comprising each observation (currently 5 in our work). With more data points, a random sample would become much more informative about the true $M(R)$ curve.

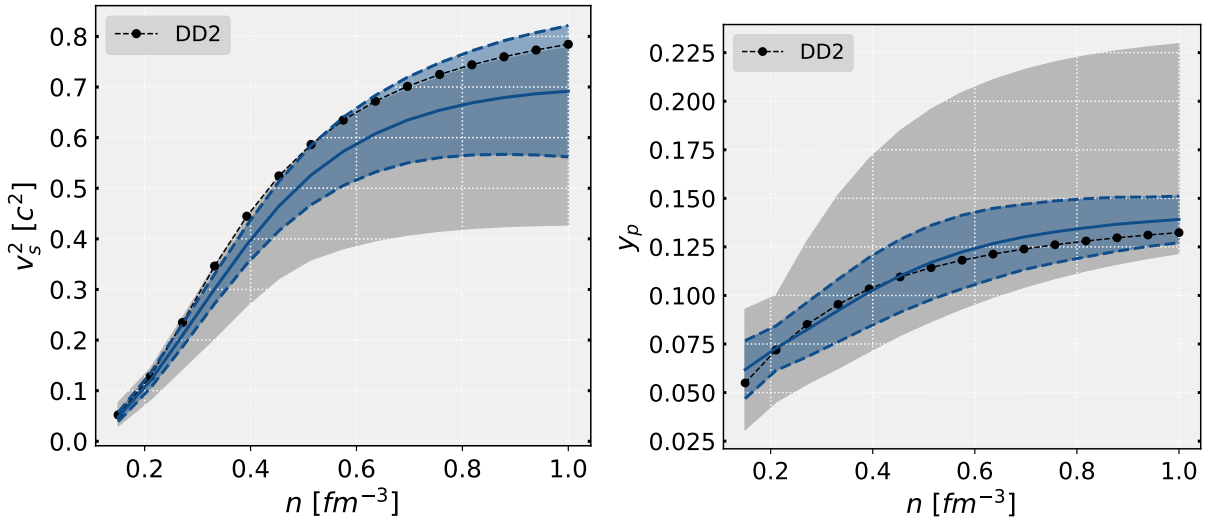


Figure 6.13: The BNN model predictions, $v_s^2(n)$ (left) and $y_p(n)$ (right), for one observation ($n_s = 1$) of the DD2 EoS, the blue area represents the 95.4% confidence interval, and the solid line the mean.

7 Conclusion and Future Work

7.1 Main conclusions

Throughout this study, we delved into the potentiality of BNNs, a probabilistic approach within the realm of ML, to predict essential parameters such as proton fraction and the squared of the speed of sound from a set of NS mock observations. Unlike conventional NNs, BNNs offer the unique advantage of providing uncertainty estimates alongside their predictions.

Our dataset for the EoS was constructed from a RMF approach using a Bayesian framework, incorporating constraints from both nuclear matter properties and NS observations. This approach resulted in a dataset of 25 287 different EoS. From this repository, we generated four different mock observational datasets, each simulating different scenarios of observational uncertainty. Two of these datasets included only simulated mass radius $M(R)$ observations, while the remaining two included additional information with TD $\Lambda(M)$ measurements. This resulted in the training of four different BNNs models to predict $v_s^2(n)$ for each of the four datasets, and the same for the $y_p(n)$.

At the core of our investigation lay a fundamental question: *How can we establish a reverse mapping from NS observations to the underlying EoS?* Our BNN-based approach successfully accomplished this task. We mapped from either five NS mass and radius measurements or, in an expanded scope, with five mass-TD measurements to the speed of sound squared or the proton fraction across fifteen fixed values of baryonic density. Importantly, this methodology accounted not only for dataset uncertainties but also for inherent model uncertainty, a unique achievement in this field.

We conducted experiments to explore the influence of TD on model predictions, as well as the impact of augmented input uncertainty on output prediction uncertainties for both speed of sound and proton fraction. We assessed the model's behavior for individual EoS within the test data and extended our analysis to cover all EoS in the test dataset. We introduced quantitative metrics, such as Γ, δ, Σ, and η, to evaluate model calibration and uncertainty. Our results highlighted that the uncertainty predicted is highly sensitive to the precision of observational data, especially when correlations with calculated properties exist, notably for the speed of sound at densities below three times saturation density, as extensively discussed in [76].

We discovered that augmenting our synthetic TD data with lower uncertainty significantly reduces prediction uncertainty for the speed of sound. However, this reduction in uncertainty doesn't extend to the proton fraction. The boost in predictive accuracy for the speed of sound can be attributed to the strong correlation between TD and the speed of sound, particularly evident near den-

sities approximately twice the saturation density. In contrast, the proton fraction demonstrates a higher sensitivity around the saturation density, particularly influenced by uncertainties in radius measurements, stemming from the correlation with the radii of low-mass stars.

Our analysis also included the decomposition of uncertainty into its epistemic and aleatoric components. We found that aleatoric uncertainty dominated in both speed of sound and proton fraction predictions. Furthermore, the decomposition of uncertainty revealed that the inclusion of tidal deformability had varying effects on the types of uncertainty, depending on the model's output. These findings align with our observations regarding total uncertainty. Additionally, we examined how an increase in dataset size influenced epistemic uncertainty, observing that it decreased as expected. To validate the robustness of our BNN model, we tested it with mock measurements from the DD2 EoS, generated using a different microscopic framework than our training data. This test confirmed the model's versatility and generalizability, reaffirming its predictive capabilities beyond the specific training data.

7.2 Limitations and future work

Throughout this study, we encountered several noteworthy challenges while working with BNNs. One of the most prominent challenges came from the considerable demands placed on computational resources. Despite having access to a computer with 64 cores, we encountered limitations due to the resource-intensive nature of the BNN model. For instance, we attempted to establish mappings between 'mock' observations and pressure as well as energy density. However, these efforts proved to be impractical due to the large amount of computing time and memory required. These parameters, while currently challenging, hold significant promise for future research directions. Another intriguing avenue for exploration lies in investigating the symmetry energy, as explored in Krastev's recent work [37]. Furthermore, the dataset utilized from this article [25] offers valuable insights, but the dataset including hyperons, from the same article, presents an equally enticing opportunity for future investigations.

The results obtained from the DD2 model highlight the potential benefits of expanding the input with more observable pairs, similar to the approach taken by Fujimoto *et al.* [27] and similar studies. Nonetheless, it's crucial to recognize that implementing such an approach, as elucidated in Section 4.2.3, necessitates substantial computational resources and time. While this presents a challenge, it's a limitation worth analyzing in future work.

In the realm of future research, an intriguing path involves subjecting the model to increased uncertainty beyond its original training scope. This exploration could offer valuable insights into the model's ability to effectively capture noise and variations in the input data. For the functional model, an exhaustive examination of the hyperparameter space presents a promising opportunity to enhance model performance. This may entail conducting a series of systematic tests, including the exploration of different optimizers, activation functions, and the utilization of grid search methodologies to optimize the number of layers and neurons, among other possibilities. Furthermore, in the case of the stochastic model, an exploration of the potential impact of prior parameters, as demon-

strated in Bollweg's work [41], could significantly enhance model performance and robustness. Additionally, the investigation of alternative distributions for the variational posterior is a path worth pursuing.
Moreover, within the context of the output distribution,it may be worthwhile to explore different distributions or embrace the intriguing approach known as quantile regression [86]. Quantile regression diverges from traditional distribution learning as it concentrates on learning the quantiles of the data rather than attempting to define an entire probability distribution for the output.

By this point, the reader may wonder: Why didn't we incorporate observational data?

For datasets 1 and 2, incorporating observational data could substantially enhance our analysis, drawing parallels with methodologies demonstrated in [29] and [30]. Nevertheless, the existing observations still require further refinement to more effectively constrain the EoS. In response to this challenge, we have taken a conservative approach when incorporating uncertainties associated with mock observations. This approach strikes a balance, preserving a degree of uncertainty while delivering sufficient constraint to encapsulate the behavior of the EoS. Looking forward, advancements in observatory capabilities, exemplified by projects like STROBE-X [21] and eXTP [19], may potentially yield radius measurements with uncertainties as low as 2%-5%. This anticipated progress, as demonstrated in our current study, holds the promise of significantly enhancing the predictive accuracy.

It's essential to recognize that both of the fields we are studying are continually evolving, so the possibilities for future work are boundless given adequate resources.

7.3 Final remarks

As a final remark, while the potential for further advancement in this field remains substantial, our contributions have aimed to propel its evolution. Specifically, we have successfully addressed the two primary questions posed in the introduction: constructing the EoS from observables and quantifying the unpredictability inherent in our model and our dataset. This dissertation's key findings have been successfully published in article [87].

Neutron Star (NS)s, as objects of study, hold immense appeal for researchers in the fields of Nuclear physics and ML. This appeal is attributable not only to the proliferation of observations coming ahead, resulting in rich and varied data sets, but also, and even more significantly, to the captivating questions that the study of these celestial objects promises to answer.

Bibliography

[1] K. Hebeler et al. “Equation of state and neutron star properties constrained by nuclear physics and observation”. In: *Astrophys. J.* 773 (2013), p. 11. DOI: 10.1088/0004-637X/773/1/11. arXiv: 1303.4662 [astro-ph.SR] (pages 2, 42, 43).

[2] Jacopo Ghiglieri et al. “Perturbative thermal QCD: formalism and applications”. In: *Physics Reports* 880 (2020), pp. 1–73 (page 2).

[3] Kai Zhou et al. “Exploring QCD matter in extreme conditions with Machine Learning”. In: (Mar. 2023). arXiv: 2303.15136 [hep-ph] (page 3).

[4] Paul Demorest et al. “Shapiro Delay Measurement of A Two Solar Mass Neutron Star”. In: *Nature* 467 (2010), pp. 1081–1083. DOI: 10.1038/nature09466 (page 2).

[5] Emmanuel Fonseca et al. “The NANOGrav Nine-year Data Set: Mass and Geometric Measurements of Binary Millisecond Pulsars”. In: *Astrophys. J.* 832.2 (2016), p. 167. DOI: 10.3847/0004-637X/832/2/167. arXiv: 1603.00545 [astro-ph.HE] (page 2).

[6] Zaven Arzoumanian et al. “The NANOGrav 11-year Data Set: High-precision timing of 45 Millisecond Pulsars”. In: *Astrophys. J. Suppl.* 235.2 (2018), p. 37. DOI: 10.3847/1538-4365/aab5b0. arXiv: 1801.01837 [astro-ph.HE] (page 2).

[7] J. Antoniadis et al. “A Massive Pulsar in a Compact Relativistic Binary”. In: *Science* 340 (Apr. 2013), p. 448. DOI: 10.1126/science.1233232 (page 2).

[8] E. Fonseca et al. “Refined Mass and Geometric Measurements of the High-mass PSR J0740+6620”. In: *Astrophys. J. Lett.* 915.1 (2021), p. L12. DOI: 10.3847/2041-8213/ac03b8. arXiv: 2104.00880 [astro-ph.HE] (pages 2, 43).

[9] Roger W. Romani et al. “PSR J1810+1744: Companion Darkening and a Precise High Neutron Star Mass”. In: *Astrophys. J. Lett.* 908.2 (2021), p. L46. DOI: 10.3847/2041-8213/abe2b4. arXiv: 2101.09822 [astro-ph.HE] (page 2).

[10] M Coleman Miller and Frederick K. Lamb. “Observational constraints on neutron star masses and radii”. In: *The European Physical Journal A* 52 (2016), pp. 1–20 (page 2).

[11] M Coleman Miller. “Astrophysical constraints on dense matter in neutron stars”. In: *Timing Neutron Stars: Pulsations, Oscillations and Explosions* (2021), pp. 1–51 (page 2).

[12] T. E. Riley et al. “A NICER View of PSR J0030+0451: Millisecond Pulsar Parameter Estimation”. In: *The Astrophysical Journal* 887.1 (Dec. 2019), p. L21. DOI: 10.3847/2041-8213/ab481c. URL: https://doi.org/10.3847%5C%2F2041-8213%5C%2Fab481c (page 2).

[13] M. C. Miller et al. “PSR J0030+0451 Mass and Radius from NICER Data and Implications for the Properties of Neutron Star Matter”. In: *The Astrophysical Journal* 887.1 (Dec. 2019), p. L24. DOI: 10.3847/2041-8213/ab50c5. URL: https://doi.org/10.3847%5C%2F2041-8213%5C%2Fab50c5 (page 2).

[14] Thomas E. Riley et al. “A NICER View of the Massive Pulsar PSR J0740+6620 Informed by Radio Timing and XMM-Newton Spectroscopy”. In: (May 2021). arXiv: 2105.06980 [astro-ph.HE] (page 2).

[15] M. C. Miller et al. “The Radius of PSR J0740+6620 from NICER and XMM-Newton Data”. In: (May 2021). arXiv: 2105.06979 [astro-ph.HE] (page 2).

[16] G. Raaijmakers et al. “Constraints on the dense matter equation of state and neutron star properties from NICER’s mass-radius estimate of PSR J0740+6620 and multimessenger observations”. In: (May 2021). arXiv: 2105.06981 [astro-ph.HE] (page 2).

[17] B. P. Abbott et al. “Properties of the binary neutron star merger GW170817”. In: *Phys. Rev.* X9.1 (2019), p. 011001. DOI: 10.1103/PhysRevX.9.011001. arXiv: 1805.11579 [gr-qc] (page 3).

[18] R. Abbott et al. “GW190814: Gravitational Waves from the Coalescence of a 23 Solar Mass Black Hole with a 2.6 Solar Mass Compact Object”. In: *Astrophys. J. Lett.* 896.2 (2020), p. L44. DOI: 10.3847/2041-8213/ab960f. arXiv: 2006.12611 [astro-ph.HE] (page 3).

[19] Anna L. Watts et al. “Dense matter with eXTP”. In: *Science China Physics, Mechanics, and Astronomy* 62.2, 29503 (Feb. 2019), p. 29503. DOI: 10.1007/s11433-017-9188-4. arXiv: 1812.04021 [astro-ph.HE] (pages 3, 72).

[20] Shuang-Nan Zhang et al. “The enhanced X-ray Timing and Polarimetry mission—eXTP”. In: *Sci. China Phys. Mech. Astron.* 62.2 (2019), p. 29502. DOI: 10.1007/s11433-018-9309-2. arXiv: 1812.04020 [astro-ph.IM] (page 3).

[21] Paul S. Ray et al. “STROBE-X: X-ray Timing and Spectroscopy on Dynamical Timescales from Microseconds to Years”. In: (Mar. 2019). arXiv: 1903.03035 [astro-ph.IM] (pages 3, 72).

[22] Anna Watts et al. “Probing the neutron star interior and the Equation of State of cold dense matter with the SKA”. In: *PoS* AASKA14 (2015), p. 043. DOI: 10.22323/1.215.0043. arXiv: 1501.00042 [astro-ph.SR] (page 3).

[23] Kent Yagi and Nicolas Yunes. “I-Love-Q”. In: *Science* 341 (2013), pp. 365–368. DOI: 10.1126/science.1236462. arXiv: 1302.4499 [gr-qc] (page 3).

[24] Tuhin Malik et al. “Relativistic Description of Dense Matter Equation of State and Compatibility with Neutron Star Observables: A Bayesian Approach”. In: *Astrophys. J.* 930.1 (2022), p. 17. DOI: 10.3847/1538-4357/ac5d3c. arXiv: 2201.12552 [nucl-th] (page 3).

[25] Tuhin Malik et al. “Spanning the full range of neutron star properties within a microscopic description”. In: *Physical Review D* 107.10 (2023), p. 103018 (pages 3–5, 41, 43, 45, 71).

[26] Reed Essick et al. “Direct Astrophysical Tests of Chiral Effective Field Theory at Supranuclear Densities”. In: *Phys. Rev. C* 102.5 (2020), p. 055803. DOI: 10.1103/PhysRevC.102.055803. arXiv: 2004.07744 [astro-ph.HE] (page 3).

[27] Yuki Fujimoto, Kenji Fukushima, and Koichi Murase. “Extensive studies of the neutron star equation of state from the deep learning inference with the observational data augmentation”. In: *Journal of High Energy Physics* 2021.3 (Mar. 2021). DOI: 10.1007/jhep03(2021)273. URL: https://doi.org/10.10077%5C%2Fjhep03%5C%282021%5C%29273 (pages 4, 5, 44, 47, 49, 71).

[28] Yuki Fujimoto, Kenji Fukushima, and Koichi Murase. “Methodology study of machine learning for the neutron star equation of state”. In: *Physical Review D* 98.2 (July 2018). DOI: 10.1103/physrevd.98.023019. URL: https://doi.org/10.1103%5C%2Fphysrevd.98.023019 (pages 4, 5, 44).

[29] Yuki Fujimoto, Kenji Fukushima, and Koichi Murase. “Mapping neutron star data to the equation of state using the deep neural network”. In: *Physical Review D* 101.5 (Mar. 2020). DOI: 10.1103/physrevd.101.054016. URL: https://doi.org/10.1103%5C%2Fphysrevd.101.054016 (pages 4, 5, 44, 72).

[30] Shriya Soma et al. *Reconstructing the neutron star equation of state from observational data via automatic differentiation*. 2022. arXiv: 2209.08883 [astro-ph.HE] (pages 4, 44, 72).

[31] Shriya Soma et al. “Neural network reconstruction of the dense matter equation of state from neutron star observables”. In: *Journal of Cosmology and Astroparticle Physics* 2022.08 (Aug. 2022), p. 071. DOI: 10.1088/1475-7516/2022/08/071. URL: https://doi.org/10.1088%5C%2F1475-7516%5C%2F2022%5C%2F08%5C%2F071 (page 4).

[32] Sagnik Chatterjee, Harsha Sudhakaran, and Ritam Mallick. *Analyzing the speed of sound in neutron star with machine learning*. 2023. arXiv: 2302.13648 [astro-ph.HE] (page 4).

[33] Filip Morawski and Michał Bejger. “Neural network reconstruction of the dense matter equation of state derived from the parameters of neutron stars”. In: *Astronomy & Astrophysics* 642 (2020), A78 (page 4).

[34] Márcio Ferreira and Constança Providência. “Unveiling the nuclear matter EoS from neutron star properties: a supervised machine learning approach”. In: *Journal of Cosmology and Astroparticle Physics* 2021.07 (2021), p. 011 (page 4).

[35] Márcio Ferreira, Valéria Carvalho, and Constança Providência. “Extracting nuclear matter properties from the neutron star matter equation of state using deep neural networks”. In: *Physical Review D* 106.10 (2022), p. 103023 (pages 4, 11).

[36] Plamen G. Krastev. “Translating Neutron Star Observations to Nuclear Symmetry Energy via Deep Neural Networks”. In: *Galaxies* 10.1 (Jan. 2022), p. 16. DOI: 10.3390/galaxies10010016. URL: https://doi.org/10.3390%5C%2Fgalaxies10010016 (pages 4, 22).

[37] Plamen G. Krastev. *A Deep Learning Approach to Extracting Nuclear Matter Properties from Neutron Star Observations*. 2023. arXiv: 2303.17146 [nucl-th] (pages 4, 71).

[38] Filip Morawski and Michał Bejger. “Detecting dense-matter phase transition signatures in neutron star mass-radius measurements as data anomalies using normalizing flows”. In: *Phys. Rev. C* 106 (6 Dec. 2022), p. 065802. DOI: 10.1103/PhysRevC.106.065802. URL: https://link.aps.org/doi/10.1103/PhysRevC.106.065802 (page 4).

[39] Ming-Zhe Han et al. “Bayesian nonparametric inference of the neutron star equation of state via a neural network”. In: *The Astrophysical Journal* 919.1 (2021), p. 11 (page 4).

[40] David JC MacKay. “A practical Bayesian framework for backpropagation networks”. In: *Neural computation* 4.3 (1992), pp. 448–472 (pages 4, 32).

[41] Sven Bollweg et al. “Deep-learning jets with uncertainties and more”. In: *SciPost Physics* 8.1 (2020), p. 006 (pages 4, 53, 72).

[42] Yu-Chiung Lin and Jiun-Huei Proty Wu. “Detection of gravitational waves using Bayesian neural networks”. In: *Physical Review D* 103.6 (2021), p. 063034 (page 4).

[43] N.K. Glendenning. *Compact Stars: Nuclear Physics, Particle Physics and General Relativity*. Astronomy and Astrophysics Library. Springer New York, 2012. ISBN: 9781468404913. URL: https://books.google.pt/books?id=cCDlBwAAQBAJ (pages 7, 14, 16).

[44] Pedro Barata de Tovar, Márcio Ferreira, and Constança Providência. “Determination of the symmetry energy from the neutron star equation of state”. In: *Phys. Rev. D* 104.12 (2021), p. 123036. DOI: 10.1103/PhysRevD.104.123036. arXiv: 2112.05551 [nucl-th] (pages 10, 45).

[45] J. M. Lattimer et al. “Direct URCA process in neutron stars”. In: *Phys. Rev. Lett.* 66 (1991), pp. 2701–2704. DOI: 10.1103/PhysRevLett.66.2701 (page 10).

[46] John Dirk Walecka. “A theory of highly condensed matter”. In: *Annals of Physics* 83.2 (1974), pp. 491–529 (page 12).

[47] Tuhin Malik and Constança Providência. “Bayesian inference of signatures of hyperons inside neutron stars”. In: *Physical Review D* 106.6 (2022), p. 063024 (page 13).

[48] J Boguta and AR Bodmer. “Relativistic calculation of nuclear matter and the nuclear surface”. In: *Nuclear Physics A* 292.3 (1977), pp. 413–428 (page 13).

[49] Horst Mueller and Brian D Serot. “Relativistic mean-field theory and the high-density nuclear equation of state”. In: *Nuclear Physics A* 606.3-4 (1996), pp. 508–537 (page 13).

[50] Richard C. Tolman. “Static Solutions of Einstein’s Field Equations for Spheres of Fluid”. In: *Phys. Rev.* 55 (4 Feb. 1939), pp. 364–373. DOI: 10.1103/PhysRev.55.364. URL: https://link.aps.org/doi/10.1103/PhysRev.55.364 (page 21).

[51] J. R. Oppenheimer and G. M. Volkoff. “On Massive Neutron Cores”. In: *Phys. Rev.* 55 (4 Feb. 1939), pp. 374–381. DOI: 10.1103/PhysRev.55.374. URL: https://link.aps.org/doi/10.1103/PhysRev.55.374 (page 21).

[52] Augustus Edward Hough Love. “The yielding of the Earth to disturbing forces”. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 82.551 (1909), pp. 73–88 (page 22).

[53] Taylor Binnington and Eric Poisson. “Relativistic theory of tidal Love numbers”. In: *Physical Review D* 80.8 (2009), p. 084018 (page 22).

[54] Thibault Damour and Alessandro Nagar. “Relativistic tidal properties of neutron stars”. In: *Physical Review D* 80.8 (2009), p. 084035 (page 22).

[55] Bruce H. Denby. “Neural Networks and Cellular Automata in Experimental High-energy Physics”. In: *Comput. Phys. Commun.* 49 (1988), pp. 429–448. DOI: 10.1016/0010-4655(88)90004-5 (page 26).

[56] Charu C Aggarwal et al. “Neural networks and deep learning”. In: *Springer* 10.978 (2018), p. 3 (page 28).

[57] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive subgradient methods for online learning and stochastic optimization.” In: *Journal of machine learning research* 12.7 (2011) (page 29).

[58] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014) (pages 29, 53).

[59] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”. In: *Machine Learning* 110 (2021), pp. 457–506 (page 30).

[60] Alex Kendall and Yarin Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in neural information processing systems* 30 (2017) (pages 30, 67).

[61] Laurent Valentin Jospin et al. “Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users”. In: *IEEE Computational Intelligence Magazine* 17.2 (May 2022), pp. 29–48. DOI: 10.1109/mci.2022.3155327. URL: https://doi.org/10.1109%5C%2Fmci.2022.3155327 (pages 32, 33, 36, 52).

[62] Yarin Gal and Zoubin Ghahramani. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2016. arXiv: 1506.02142 [stat.ML] (page 35).

[63] Laurence Perreault Levasseur, Yashar D. Hezaveh, and Risa H. Wechsler. “Uncertainties in Parameters Estimated with Neural Networks: Application to Strong Gravitational Lensing”. In: *The Astrophysical Journal* 850.1 (Nov. 2017), p. L7. DOI: 10.3847/2041-8213/aa9704. URL: https://doi.org/10.3847%5C%2F2041-8213%5C%2Faa9704 (page 35).

[64] Charles Blundell et al. *Weight Uncertainty in Neural Networks*. 2015. arXiv: 1505.05424 [stat.ML] (page 36).

[65] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/ (visited on 09/30/2010) (pages 37, 53).

[66] Audrey Olivier, Michael D Shields, and Lori Graham-Brady. "Bayesian neural networks for uncertainty quantification in data-driven materials modeling". In: *Computer methods in applied mechanics and engineering* 386 (2021), p. 114079 (page 38).

[67] John Skilling. "Nested Sampling". In: *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. Ed. by Rainer Fischer, Roland Preuss, and Udo Von Toussaint. Vol. 735. American Institute of Physics Conference Series. Nov. 2004, pp. 395–405. DOI: 10.1063/1.1835238 (page 43).

[68] J. Buchner et al. "X-ray spectral modelling of the AGN obscuring region in the CDFS: Bayesian model selection and catalogue". In: *Astron. Astrophys.* 564 (2014), A125. DOI: 10.1051/0004-6361/201322971. arXiv: 1402.0004 [astro-ph.HE] (page 43).

[69] Johannes Buchner. *Nested Sampling Methods*. 2021. arXiv: 2101.09675 [stat.CO] (page 43).

[70] S. Typel and H. H. Wolter. "Relativistic mean field calculations with density dependent meson nucleon coupling". In: *Nucl. Phys. A* 656 (1999), pp. 331–364. DOI: 10.1016/S0375-9474(99)00310-3 (page 43).

[71] M. Dutra et al. "Relativistic Mean-Field Hadronic Models under Nuclear Matter Constraints". In: *Phys. Rev. C* 90.5 (2014), p. 055203. DOI: 10.1103/PhysRevC.90.055203. arXiv: 1405.3633 [nucl-th] (page 43).

[72] Shlomo, S., Kolomietz, V. M., and Colò, G. "Deducing the nuclear-matter incompressibility coefficient from data on isoscalar compression modes". In: *Eur. Phys. J. A* 30.1 (2006), pp. 23–30. DOI: 10.1140/epja/i2006-10100-3. URL: https://doi.org/10.1140/epja/i2006-10100-3 (page 43).

[73] B. G. Todd-Rutel and J. Piekarewicz. "Neutron-Rich Nuclei and Neutron Stars: A New Accurately Calibrated Interaction for the Study of Neutron-Rich Matter". In: *Phys. Rev. Lett.* 95 (2005), p. 122501. DOI: 10.1103/PhysRevLett.95.122501. arXiv: nucl-th/0504034 (page 43).

[74] Reed Essick et al. "Detailed examination of astrophysical constraints on the symmetry energy and the neutron skin of Pb208 with minimal modeling assumptions". In: *Phys. Rev. C* 104.6 (2021), p. 065804. DOI: 10.1103/PhysRevC.104.065804. arXiv: 2107.05528 [nucl-th] (page 43).

[75] Chiranjib Mondal and Francesca Gulminelli. "Can we decipher the composition of the core of a neutron star?" In: (Nov. 2021). arXiv: 2111.04520 [nucl-th] (page 45).

[76] Márcio Ferreira et al. "Empirical constraints on the high-density equation of state from multimessenger observables". In: *Physical Review D* 101.4 (2020), p. 043021 (pages 50, 60, 70).

[77] Tuhin Malik et al. “GW170817: Constraining the nuclear matter equation of state from the neutron star tidal deformability”. In: *Physical Review C* 98.3 (2018), p. 035804 (page 51).

[78] Prasanta Char et al. “Generalised description of Neutron Star matter with nucleonic Relativistic Density Functional”. In: (July 2023). arXiv: 2307.12364 [nucl-th] (page 51).

[79] S. Raschka and V. Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow 2, 3rd Edition*. Expert insight. Packt Publishing, 2019. ISBN: 9781789955750. URL: https://books.google.pt/books?id=n1cjyAEACAAJ (page 52).

[80] François Chollet et al. *Keras*. https://keras.io. 2015 (page 53).

[81] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. “On the convergence of adam and beyond”. In: *arXiv preprint arXiv:1904.09237* (2019) (page 53).

[82] Armen Der Kiureghian and Ove Ditlevsen. “Aleatory or epistemic? Does it matter?” In: *Structural safety* 31.2 (2009), pp. 105–112 (page 64).

[83] S. Typel et al. “Composition and thermodynamics of nuclear matter with light clusters”. In: *Phys. Rev. C* 81 (2010), p. 015803. DOI: 10.1103/PhysRevC.81.015803. arXiv: 0908.2344 [nucl-th] (page 68).

[84] M. Fortin et al. “Neutron star radii and crusts: uncertainties and unified equations of state”. In: *Phys. Rev. C* 94.3 (2016), p. 035804. DOI: 10.1103/PhysRevC.94.035804. arXiv: 1604.01944 [astro-ph.SR] (page 68).

[85] Morgane Fortin et al. “Thermal evolution of relativistic hyperonic compact stars with calibrated equations of state”. In: *Phys. Rev. D* 103.8 (2021), p. 083004. DOI: 10.1103/PhysRevD.103.083004. arXiv: 2102.07565 [nucl-th] (page 68).

[86] Owen Convery et al. “Uncertainty quantification for virtual diagnostic of particle accelerators”. In: *Physical Review Accelerators and Beams* 24.7 (2021), p. 074602 (page 72).

[87] Valéria Carvalho et al. “Decoding neutron star observations: Revealing composition through Bayesian neural networks”. In: *Phys. Rev. D* 108 (4 Aug. 2023), p. 043031. DOI: 10.1103/PhysRevD.108.043031. URL: https://link.aps.org/doi/10.1103/PhysRevD.108.043031 (page 72).

[88] K. B. Petersen and M. S. Pedersen. “The Matrix Cookbook”. In: (2012). URL: http://www.math.uwaterloo.ca/~hwolkowi//matrixcookbook.pdf (page 87).

[89] Michael I Jordan et al. “An introduction to variational methods for graphical models”. In: *Machine learning* 37 (1999), pp. 183–233 (page 88).

Appendix A

Exploring the Equations of Motion and the Symmetry Energy in Nuclear Models

A.1 Equations of motion

The Lagrangian density we used is

$$\mathscr{L} = \mathscr{L}_N + \mathscr{L}_M + \mathscr{L}_{NL} + \mathscr{L}_{lep}, \tag{A.1}$$

where each term is represented as

$$\begin{aligned}
\mathscr{L}_N =& \bar{\Psi}(x)\Big[\left(i\gamma_\mu\partial^\mu - g_\omega\omega^\mu(x) - g_\varrho \boldsymbol{t}\cdot\boldsymbol{\varrho}^\mu(x)\right) - \left(m - g_\varphi\varphi(x)\right)\Big]\Psi(x)\,,\\
\mathscr{L}_M =& \frac{1}{2}\left[\partial_\mu\varphi(x)\partial^\mu\varphi(x) - m_\varphi^2\varphi^2(x)\right] - \frac{1}{4}F^{(\omega)}_{\mu\nu}(x)F^{(\omega)\mu\nu}(x) + \frac{1}{2}m_\omega^2\omega_\mu(x)\omega^\mu(x) -\\
& \frac{1}{4}\boldsymbol{F}^{(\varrho)}_{\mu\nu}(x)\cdot\boldsymbol{F}^{(\varrho)\mu\nu}(x) + \frac{1}{2}m_\varrho^2\boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x)\,,\\
\mathscr{L}_{NL} =& -\frac{1}{3}b\,m\,g_\varphi^3\varphi^3(x) - \frac{1}{4}cg_\varphi^4\varphi^4(x) + \frac{\xi}{4!}g_\omega^4(\omega_\mu(x)\omega^\mu(x))^2 + \Lambda_\omega g_\varrho^2\boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x)g_\omega^2\omega_\mu(x)\omega^\mu(x)\,,\\
\mathscr{L}_{lep} =& \sum_l \bar{\Psi}_l(x)\left(i\gamma_\mu\partial^\mu - m_l\right)\Psi_l(x)\,.
\end{aligned}$$

In order to obtain the equations of motion for a general field ϕ, one needs to solve the Euler-Lagrange equation:

$$\underbrace{\partial_\mu\frac{\partial\mathscr{L}}{\partial(\partial_\mu\phi(x))}}_{\mathbf{1.}} = \underbrace{\frac{\partial\mathscr{L}}{\partial\phi(x)}}_{\mathbf{2.}}\,. \tag{A.2}$$

We are then going to show how to solve this equation for the five fields being used $\varphi(x)$, $\omega^\mu(x)$, $\boldsymbol{\varrho}^\mu(x)$, $\Psi(x)$ and $\Psi_l(x)$, calculating the two parts **1.** and **2.** and then combining them.

A.1.1 Scalar-isoscalar meson field $\varphi(x)$

1.

$$\partial_a \frac{\partial \mathscr{L}}{\partial(\partial_a \varphi(x))} = \frac{1}{2}\partial_a \left[\frac{\partial\left(\partial_\mu \varphi(x)\right)}{\partial(\partial_a \varphi(x))}\partial^\mu \varphi(x) + \partial_\mu \varphi(x) \frac{\partial\left(\partial^\mu \varphi(x)\right)}{\partial(\partial_a \varphi(x))} \right] ,$$
$$= \frac{1}{2}\left[\partial_a \delta^\mu_a \partial^\mu \varphi(x) + \partial_a \partial_\mu \varphi(x) g^{\mu a}\right] = \frac{1}{2}\left(\partial_\mu \partial^\mu \varphi(x) + \partial^\mu \partial_\mu \varphi(x)\right) = \partial_\mu \partial^\mu \varphi(x) . \quad \text{(A.3)}$$

2.

$$\frac{\partial \mathscr{L}}{\partial \varphi(x)} = \frac{\partial}{\partial \varphi(x)}\left(g_\varphi \varphi(x)\bar{\Psi}(x)\Psi(x) - \frac{1}{2}m_\varphi^2 \varphi(x)^2 - \frac{1}{3}b g_\varphi^3 \varphi(x)^3 - \frac{1}{4}c g_\varphi^4 \varphi(x)^4 \right) ,$$
$$= g_\varphi \bar{\Psi}(x)\Psi(x) - m_\varphi^2 \varphi(x) - bm g_\varphi^3 \varphi(x)^2 - c g_\varphi^4 \varphi^3(x) . \quad \text{(A.4)}$$

The equation of motion is then

$$\partial_\mu \partial^\mu \varphi(x) + m_\varphi^2 \varphi(x) + bm g_\varphi^3 \varphi^2(x) + c g_\varphi^4 \varphi^3(x) = g_\varphi \bar{\Psi}(x)\Psi(x) . \quad \text{(A.5)}$$

A.1.2 Vector-isoscalar meson field $\omega^\mu(x)$

1.

$$\partial_a \frac{\partial \mathscr{L}}{\partial(\partial_a \omega_b(x))} = -\frac{1}{4}\partial_a \left[\frac{\partial F^{(\omega)}_{\mu\nu}(x)}{\partial(\partial_a \omega_b(x))} F^{(\omega)\mu\nu}(x) + F^{(\omega)}_{\mu\nu}(x) \frac{\partial F^{(\omega)\mu\nu}(x)}{\partial(\partial_a \omega_b(x))} \right] ,$$
$$= -\frac{1}{4}\partial_a \left[\frac{\partial\left(\partial_\mu \omega_\nu - \partial_\nu \omega_\mu\right)}{\partial(\partial_a \omega_b)} F^{(\omega)\mu\nu}(x) + F^{(\omega)}_{\mu\nu}(x) g^{\mu l} g^{\nu k} \frac{\partial\left(\partial_l \omega_k - \partial_k \omega_l\right)}{\partial(\partial_a \omega_b)} \right] ,$$
$$= -\frac{1}{4}\partial_a \left[\left(\delta^{\mu a}\delta^{\nu b} - \delta^{\nu a}\delta^{\mu b}\right) F^{(\omega)\mu\nu}(x) + F^{(\omega)lk}(x)\left(\delta^{la}\delta^{kb} - \delta^{ka}\delta^{lb}\right) \right] ,$$
$$= -\frac{1}{4}\partial_a \left[\left(F^{(\omega)ab}(x) - F^{(\omega)ba}(x)\right) + \left(F^{(\omega)ab}(x) - F^{(\omega)ba}(x)\right) \right] = -\partial_a F^{(\omega)ab}(x) . \quad \text{(A.6)}$$

2.

$$\frac{\partial \mathscr{L}}{\partial \omega_b(x)} = \frac{\partial}{\partial \omega_b(x)}\Big(-g_\omega \bar{\Psi}(x)\gamma^\mu \omega_\mu(x)\Psi(x) + \frac{1}{2}m_\omega^2 \omega_\mu(x)\omega^\mu(x) + \frac{\xi}{4!}g_\omega^4 (\omega_\mu(x)\omega^\mu(x))^2$$
$$+ \Lambda_\omega g_\varrho^2 \boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x) g_\omega^2 \omega_\mu(x)\omega^\mu(x) \Big) ,$$
$$= -g_\omega \bar{\Psi}(x)\gamma^b \Psi(x) + \frac{1}{2}m_\omega^2 \left(\delta^{\mu b}\omega^\mu(x) + \omega_\mu(x) g^{\mu b}\right) + \frac{\xi}{12}g_\omega^4 (\omega_\mu(x)\omega^\mu(x))\left(\delta^{\mu b}\omega^\mu(x) + \omega_\mu(x) g^{\mu b}\right)$$
$$+ \Lambda_\omega g_\varrho^2 \boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x) g_\omega^2 \left(\delta^{\mu b}\omega^\mu(x) + \omega_\mu(x) g^{\mu b}\right) ,$$
$$= -g_\omega \bar{\Psi}(x)\gamma^b \Psi(x) + m_\omega^2 \omega^b(x) + \frac{\xi}{3!}g_\omega^4 (\omega_\mu(x)\omega^\mu(x))\omega^b(x) + 2\Lambda_\omega g_\varrho^2 \boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x) g_\omega^2 \omega^b(x) . \quad \text{(A.7)}$$

The equation of motion becomes

$$\partial_a F^{(\omega)ab}(x) + m_\omega^2 \omega^b(x) + \frac{\xi}{3!}g_\omega^4 (\omega_\mu(x)\omega^\mu(x))\omega^b(x) + 2\Lambda_\omega g_\varrho^2 \boldsymbol{\varrho}_\mu(x)\cdot\boldsymbol{\varrho}^\mu(x) g_\omega^2 \omega^b(x) = g_\omega \bar{\Psi}(x)\gamma^b \Psi(x) . \quad \text{(A.8)}$$

A.1.3 Vector-isovector meson field $\boldsymbol{\varrho}^{\mu}(x)$

1.

$$\partial_a \frac{\partial \mathscr{L}}{\partial(\partial_a \varrho_{\lambda b}(x))} = -\partial_a \boldsymbol{F}^{(\varrho)ab}(x)\,. \tag{A.9}$$

2.

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \varrho_{\lambda b}(x)} &= \frac{\partial}{\partial \varrho_{\lambda b}(x)}\left(\bar{\Psi}(x)\gamma^{\mu} g_{\rho}\, t_{\lambda}\varrho_{\lambda\mu}(x)\Psi(x) + \Lambda_{\omega} g_{\varrho}^2 \varrho_{\lambda\mu}(x)\varrho_{\lambda}^{\mu}(x) g_{\omega}^2 \omega_{\mu}\omega^{\mu} + \frac{1}{2} m_{\rho}^2 \varrho_{\lambda\mu}(x)\varrho_{\lambda}^{\mu}\right)(x)\,,\\ &= \bar{\Psi}(x)\gamma^{b} g_{\rho}\, t_{\lambda}\Psi(x) + \Lambda_{\omega} g_{\varrho}^2 \left(\delta_b^{\mu}\varrho_{\lambda}^{\mu}(x) + \varrho_{\lambda\mu}(x) g^{b\mu}\right) g_{\omega}^2 \omega_{\mu}(x)\omega^{\mu}(x) + \frac{1}{2} m_{\rho}^2 \left(\delta_b^{\mu}\varrho_{\lambda}^{\mu}(x) + \varrho_{\lambda\mu}(x) g^{b\mu}\right),\\ &= \bar{\Psi}(x)\gamma^{b} g_{\rho}\, t_{\lambda}\Psi(x) + 2\Lambda_{\omega} g_{\varrho}^2 \varrho_{\lambda}^{b}(x) g_{\omega}^2 \omega_{\mu}(x)\omega^{\mu}(x) + \frac{1}{2} m_{\rho}^2 \varrho_{\lambda}^{b}(x)\,.\end{aligned} \tag{A.10}$$

The equation of motion is then expressed as follows

$$\partial_a \boldsymbol{F}^{(\varrho)ab}(x) + 2\Lambda_{\omega} g_{\varrho}^2 \boldsymbol{\varrho}^{b}(x) g_{\omega}^2 \omega_{\mu}(x)\omega^{\mu}(x) + m_{\rho}^2 \boldsymbol{\varrho}^{b}(x) = \bar{\Psi}(x)\gamma^{b} g_{\rho}\, \boldsymbol{t}\Psi(x)\,. \tag{A.11}$$

A.1.4 Nucleonic field $\Psi(x)$

1.

$$\partial_a \frac{\partial \mathscr{L}}{\partial(\partial_a \Psi^{\dagger}(x))} = 0\,. \tag{A.12}$$

2.

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \Psi^{\dagger}(x)} &= \frac{\partial}{\partial \Psi^{\dagger}(x)}\left[\Psi^{\dagger}(x)\gamma^0\left[\gamma^{\mu}\left(i\partial_{\mu} - g_{\omega}\omega_{\mu}(x) - g_{\varrho}\,\boldsymbol{t}\cdot\boldsymbol{\varrho}_{\mu}(x)\right) - \left(m - g_{\varphi}\varphi(x)\right)\right]\Psi(x)\right],\\ &= \gamma^0\left[\gamma^{\mu}\left(i\partial_{\mu} - g_{\omega}\omega_{\mu}(x) - g_{\varrho}\,\boldsymbol{t}\cdot\boldsymbol{\varrho}_{\mu}(x)\right) - \left(m - g_{\varphi}\varphi(x)\right)\right]\Psi(x)\,.\end{aligned} \tag{A.13}$$

The equation of motion can then be described by the following equation

$$\left[\gamma^{\mu}\left(i\partial_{\mu} - g_{\omega}\omega_{\mu}(x) - g_{\varrho}\,\boldsymbol{t}\cdot\boldsymbol{\varrho}_{\mu}(x)\right) - \left(m - g_{\varphi}\varphi(x)\right)\right]\Psi(x) = 0\,. \tag{A.14}$$

A.1.5 Leptonic field $\Psi_l(x)$

1.

$$\partial_a \frac{\partial \mathscr{L}}{\partial(\partial_a \Psi_l^{\dagger}(x))} = 0\,. \tag{A.15}$$

2.

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \Psi_l^{\dagger}(x)} &= \frac{\partial}{\partial \Psi_l^{\dagger}(x)}\left[\Psi_l^{\dagger}(x)\gamma^0\left[\gamma^{\mu} i\partial_{\mu} - m_l\right]\Psi_l(x)\right],\\ &= \gamma^0\left[i\gamma^{\mu}\partial_{\mu} - m_l\right]\Psi_l(x)\,.\end{aligned} \tag{A.16}$$

One can then describe the equation of motion as

$$\left[i\gamma^{\mu}\partial_{\mu} - m_l\right]\Psi_l(x) = 0\,. \tag{A.17}$$

A.2 Symmetry energy

In order to calculate the Symmetry energy we are going to start with the definition of the energy density previously calculated:

$$\begin{aligned}\varepsilon = &\sum_{j=p,n}\left[g_\omega\omega_0 n_j + g_\rho\varrho_{03}I_{3j}n_j + \frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}}\left(k^2\sqrt{k^2+m_j^{*2}}\right)dk\right]\\ &+\sum_{j=e^-,\mu^-}\frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}}dk\left(k^2\sqrt{k^2+m_j^2}\right)\\ &-\left[-\frac{1}{2}m_\varphi^2\varphi^2-\frac{1}{3}b\,m\,g_\varphi^3\varphi^3-\frac{1}{4}cg_\varphi^4\varphi^4+\frac{\xi}{4!}g_\omega^4\omega_0^4+\frac{1}{2}m_\omega^2\omega_0^2+\frac{1}{2}m_\varrho^2\varrho_{03}^2+\Lambda_\omega g_\varrho^2\varrho_{03}^2g_\omega^2\omega_0^2\right].\end{aligned} \tag{A.18}$$

Separating the nucleonic ε_b and the leptonic ε_l part

$$\varepsilon = \varepsilon_b + \varepsilon_l\,, \tag{A.19}$$

the nucleonic part stays

$$\begin{aligned}\varepsilon_b = &\sum_{j=p,n}\left[g_\omega\omega_0 n_j + g_\rho\varrho_{03}I_{3j}n_j + \frac{(2J_j+1)}{2\pi^2}\int_0^{k_{F,j}}\left(k^2\sqrt{k^2+m_j^{*2}}\right)dk\right]\\ &-\left[-\frac{1}{2}m_\varphi^2\varphi^2-\frac{1}{3}b\,m\,g_\varphi^3\varphi^3-\frac{1}{4}cg_\varphi^4\varphi^4+\frac{\xi}{4!}g_\omega^4\omega_0^4+\frac{1}{2}m_\omega^2\omega_0^2+\frac{1}{2}m_\varrho^2\varrho_{03}^2+\Lambda_\omega g_\varrho^2\varrho_{03}^2g_\omega^2\omega_0^2\right].\end{aligned} \tag{A.20}$$

Defining the fermi momentum as

$$k_F = \left(\frac{3\pi^2 n}{2}\right)^{\frac{1}{3}}\,. \tag{A.21}$$

The fermi momentum for the proton and neutron can be expressed in terms of the isospin asymmetry parameter $\delta = \frac{n_n - n_p}{n}$ as

$$k_{F,n} = k_F\,(1+\delta)^{\frac{1}{3}}\,, \tag{A.22}$$

$$k_{F,p} = k_F\,(1-\delta)^{\frac{1}{3}}\,. \tag{A.23}$$

Replacing the fermi momentums in the energy density it becomes

$$
\begin{aligned}
\varepsilon_b &= \sum_{j=p,n}\left[g_\omega\omega_0 n_j + g_\rho \varrho_{03} I_{3j} n_j\right] + \frac{1}{\pi^2}\int_0^{k_F(1-\delta)^{\frac{1}{3}}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk + \frac{1}{\pi^2}\int_0^{k_F(1+\delta)^{\frac{1}{3}}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk \\
&\quad - \left[-\frac{1}{2}m_\varphi^2\varphi^2 - \frac{1}{3}b\ m\ g_\varphi^3\varphi^3 - \frac{1}{4}c g_\varphi^4\varphi^4 + \frac{\xi}{4!}g_\omega^4\omega_0^4 + \frac{1}{2}m_\omega^2\omega_0^2 + \frac{1}{2}m_\varrho^2\varrho_{03}^2 + \Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0^2\right], \\
&= \sum_{j=p,n}\left[g_\rho\varrho_{03}I_{3j}n_j\right] - \frac{1}{2}m_\varrho^2\varrho_{03}^2 - \Lambda_\omega g_\varrho^2\varrho_{03}^2 g_\omega^2\omega_0^2 + \frac{1}{\pi^2}\int_0^{k_F(1-\delta)^{\frac{1}{3}}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk \\
&\quad + \frac{1}{\pi^2}\int_0^{k_F(1+\delta)^{\frac{1}{3}}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk + g_\omega\omega_0 n + \frac{1}{2}m_\varphi^2\varphi^2 + \frac{1}{3}b\ m\ g_\varphi^3\varphi^3 + \frac{1}{4}c g_\varphi^4\varphi^4 - \frac{\xi}{4!}g_\omega^4\omega_0^4 - \frac{1}{2}m_\omega^2\omega_0^2, \\
&= \frac{1}{2}\sum_{j=p,n}\left[g_\rho\varrho_{03}I_{3j}n_j\right] + \frac{1}{\pi^2}\int_0^{k_F(1-\delta)^{\frac{1}{3}}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk + \frac{1}{\pi^2}\int_0^{k_F(1+\delta)^{\frac{1}{3}}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk \\
&\quad + g_\omega\omega_0 n + \frac{1}{2}m_\varphi^2\varphi^2 + \frac{1}{3}b\ m\ g_\varphi^3\varphi^3 + \frac{1}{4}c g_\varphi^4\varphi^4 - \frac{\xi}{4!}g_\omega^4\omega_0^4 - \frac{1}{2}m_\omega^2\omega_0^2 .
\end{aligned}
\tag{A.24}
$$

Now we can expand the nucleonic component of the energy density per density in a symmetric and asymmetric part:

$$
\frac{\varepsilon_b}{n}(n,\delta) - m \equiv \epsilon(n,\delta) = \epsilon_{SNM}(n) + E_{sym}(n)\delta^2 \quad , \epsilon_{SNM}(n) \equiv \epsilon(n,\delta=0) . \tag{A.25}
$$

The symmetry energy is defined as:

$$
E_{sym}(n) = \left.\frac{\partial^2\epsilon(n,\delta)}{2\partial\delta^2}\right|_{\delta=0} . \tag{A.26}
$$

Calculating then the second derivative of the binding energy per nucleon with respect to the isospin asymmetry parameter one obtains

$$
\begin{aligned}
E_{sym}(n) &= \frac{1}{2}\left(\frac{\partial^2}{\partial\delta^2}\left(\frac{1}{8}\frac{\left(g_\rho\right)^2 n\delta^2}{2\Lambda_\omega g_\varrho^2 g_\omega^2\omega_0^2 + m_\rho^2} + \frac{1}{n\pi^2}\int_0^{k_F(1+\delta)^{1/3}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk + \frac{1}{n\pi^2}\int_0^{k_F(1-\delta)^{1/3}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk\right)\right), \\
&= \frac{1}{2}\left(\frac{1}{4}\frac{\left(g_\rho\right)^2 n}{2\Lambda_\omega g_\varrho^2 g_\omega^2\omega_0^2 + m_\rho^2} + \underbrace{\frac{\partial^2}{\partial\delta^2}\left[\frac{1}{n\pi^2}\left(\int_0^{k_F(1+\delta)^{1/3}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk + \int_0^{k_F(1-\delta)^{1/3}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk\right)\right]}_{D}\right),
\end{aligned}
\tag{A.27}
$$

where the derivative of the second term, denominated by D gives

$$
\begin{aligned}
D =& \frac{\partial}{\partial\delta}\left[\left(\frac{1}{n\pi^2}\right)\frac{\partial\left(k_F(1-\delta)^{1/3}\right)}{\partial\delta}\frac{\partial}{\partial\left(k_F(1-\delta)^{1/3}\right)}\int_0^{k_F(1-\delta)^{1/3}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk\right.\\
&\left.+\left(\frac{1}{n\pi^2}\right)\frac{\partial\left(k_F(1+\delta)^{1/3}\right)}{\partial\delta}\frac{\partial}{\partial\left(k_F(1+\delta)^{1/3}\right)}\int_0^{k_F(1+\delta)^{1/3}}\left(k^2\sqrt{k^2+m^{*2}}\right)dk\right],\\
=& \frac{\partial}{\partial\delta}\left[\left(\frac{1}{n\pi^2}\right)\left(-\frac{1}{3}k_F^3\right)\left(\sqrt{k_F^2(1-\delta)^{\frac{2}{3}}+m^{*2}}\right)+\left(\frac{1}{n\pi^2}\right)\left(\frac{1}{3}k_F^3\right)\left(\sqrt{k_F^2(1+\delta)^{\frac{2}{3}}+m^{*2}}\right)\right],\\
=& \left(\frac{1}{n\pi^2}\right)\left(\frac{1}{3}k_F^3\right)\left[\frac{-1}{2}\left(k_F^2(1-\delta)^{\frac{2}{3}}+m^{*2}\right)^{-1/2}\left(\frac{-2k_F^2}{3}\right)(1-\delta)^{-1/3}+\frac{1}{2}\left(k_F^2(1+\delta)^{\frac{2}{3}}+m^{*2}\right)^{-1/2}\left(\frac{2k_F^2}{3}\right)(1+\delta)^{-1/3}\right],\\
=& \left(\frac{1}{n\pi^2}\right)\left(\frac{1}{9}k_F^5\right)\left[\frac{(1-\delta)^{-1/3}}{\sqrt{k_F^2(1-\delta)^{\frac{2}{3}}+m^{*2}}}+\frac{(1+\delta)^{-1/3}}{\sqrt{k_F^2(1+\delta)^{\frac{2}{3}}+m^{*2}}}\right],\\
=& \left(\frac{3\cancel{\pi^2}}{2k_F^3\cancel{\pi^2}}\right)\left(\frac{1}{9}k_F^5\right)\left[\frac{(1-\delta)^{-1/3}}{\sqrt{k_F^2(1-\delta)^{\frac{2}{3}}+m^{*2}}}+\frac{(1+\delta)^{-1/3}}{\sqrt{k_F^2(1+\delta)^{\frac{2}{3}}+m^{*2}}}\right],\\
=& \frac{1}{6}k_F^2\left[\frac{(1-\delta)^{-1/3}}{\sqrt{k_F^2(1-\delta)^{\frac{2}{3}}+m^{*2}}}+\frac{(1+\delta)^{-1/3}}{\sqrt{k_F^2(1+\delta)^{\frac{2}{3}}+m^{*2}}}\right].
\end{aligned}
\tag{A.28}
$$

Substituing the result obtained for D in Eq. A.27 gives

$$
\begin{aligned}
E_{sym} =& \frac{1}{2}\left(\frac{1}{4}\frac{\left(g_\rho\right)^2 n}{2\Lambda_\omega g_\varrho^2 g_\omega^2\omega_0^2+m_\rho^2}+\frac{1}{6}k_F^2\left[\frac{(1-\delta)^{-1/3}}{\sqrt{k_F^2(1-\delta)^{\frac{2}{3}}+m^{*2}}}+\frac{(1+\delta)^{-1/3}}{\sqrt{k_F^2(1+\delta)^{\frac{2}{3}}+m^{*2}}}\right]\right)_{\delta=0},\\
=& \frac{1}{2}\left(\frac{1}{4}\frac{\left(g_\rho\right)^2 n}{2\Lambda_\omega g_\varrho^2 g_\omega^2\omega_0^2+m_\rho^2}+\frac{1}{6}k_F^2\left[\frac{1}{\sqrt{k_F^2+m^{*2}}}+\frac{1}{\sqrt{k_F^2+m^{*2}}}\right]\right)_{\delta=0},\\
=& \frac{1}{8}\frac{\left(g_\rho\right)^2 n}{2\Lambda_\omega g_\varrho^2 g_\omega^2\omega_0^2+m_\rho^2}+\frac{1}{6}k_F^2\left[\frac{1}{\sqrt{k_F^2+m^{*2}}}\right],\\
=& \frac{2k_F^3}{(3\pi^2)8}\frac{\left(g_\rho\right)^2}{2\Lambda_\omega g_\varrho^2 g_\omega^2\omega_0^2+m_\rho^2}+\frac{1}{6}\left[\frac{k_F^2}{\sqrt{k_F^2+m^{*2}}}\right]=\frac{k_F^3}{12\pi^2}\frac{\left(g_\rho\right)^2}{2\Lambda_\omega g_\varrho^2 g_\omega^2\omega_0^2+m_\rho^2}+\frac{1}{6}\left[\frac{k_F^2}{\sqrt{k_F^2+m^{*2}}}\right].
\end{aligned}
\tag{A.29}
$$

Then the symmetry energy becomes:

$$
E_{sym}(n)=\frac{1}{8}\frac{\left(g_\varrho\right)^2 n}{2\Lambda_\omega g_\varrho^2 g_\omega^2\omega_0^2+m_\rho^2}+\frac{1}{6}k_F^2\left[\frac{1}{\sqrt{k_F^2+m^{*2}}}\right].
\tag{A.30}
$$

And the binding energy per nucleon for symmetric nuclear matter

$$
\epsilon_{SNM}(n)=\frac{1}{n}\left[\frac{1}{2}m_\varphi^2\varphi^2+\frac{1}{3}b\,m\,g_\varphi^3\varphi^3+\frac{1}{4}cg_\varphi^4\varphi^4-\frac{\xi}{4!}g_\omega^4\omega_0^4-\frac{1}{2}m_\omega^2\omega_0^2+g_\omega\omega_0 n+\frac{2}{\pi^2}\int_0^{k_F}\left(k^2\sqrt{k^2+m^{*2}}\right)dk\right]-m\,.
\tag{A.31}
$$

Appendix B

Bayesian Neural Networks : Demonstrations and Theoretical Insights

B.1 Kullback-Leibler divergence

If the two distributions being calculated are two multivariate normal distributions with correlation such as $q(\theta) = \mathcal{N}(\mu_q, \Sigma_q)$ and $P(\theta) = \mathcal{N}(\mu_p, \Sigma_p)$ then :

$$\begin{aligned}
\mathrm{KL}(q(\theta)||P(\theta)) &= \int_\theta q(\theta) \log\left(\frac{q(\theta)}{P(\theta)}\right) d\theta \,, \\
&= \int_\theta q(\theta) \log\big(q(\theta)\big) - q(\theta) \log\big(P(\theta)\big)\, d\theta \,, \\
&= \mathbb{E}_q \frac{1}{2}\left[\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)} - (\mathbf{x}-\mu_q)^T\Sigma_q^{-1}(\mathbf{x}-\mu_q) + (\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p)\right] , \\
&= \frac{1}{2}\left[\mathbb{E}_q\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - \mathbb{E}_q(\mathbf{x}-\mu_q)^T\Sigma_q^{-1}(\mathbf{x}-\mu_q) + \mathbb{E}_q(\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p)\right] , \\
&= \frac{1}{2}\left[\underbrace{\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right)}_{1} - \underbrace{\mathbb{E}_q(\mathbf{x}-\mu_q)^T\Sigma_q^{-1}(\mathbf{x}-\mu_q)}_{2} + \underbrace{\mathbb{E}_q(\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p)}_{3}\right] . \qquad \text{(B.1)}
\end{aligned}$$

For the term 2 and 3, since it is a scalar $\in \mathbb{R}$ so it's possible to say that $\mathrm{tr}(X) = X$ leading to $\mathrm{tr}(ABC) =$

tr(BCA) as it is present in [88]:

$$
\begin{aligned}
\mathrm{KL}(q(\theta)||P(\theta)) &= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - \mathbb{E}_q\,\mathrm{tr}\left(\Sigma_q^{-1}(\mathbf{x}-\mu_q)(\mathbf{x}-\mu_q)^T\right) + \mathbb{E}_q\,\mathrm{tr}\left(\Sigma_p^{-1}(\mathbf{x}-\mu_p)(\mathbf{x}-\mu_p)^T\right)\right], \\
&= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - \mathrm{tr}\left(\Sigma_q^{-1}\mathbb{E}_q(\mathbf{x}-\mu_q)(\mathbf{x}-\mu_q)^T\right) + \mathrm{tr}\left(\Sigma_p^{-1}\mathbb{E}_q(\mathbf{x}\mathbf{x}^T - 2\mu_p\mathbf{x}^T + \mu_p\mu_p^T)\right)\right], \\
&= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - \mathrm{tr}\left(\Sigma_q^{-1}\Sigma_q\right) + \mathrm{tr}\left(\Sigma_p^{-1}\mathbb{E}_q(\mathbf{x}\mathbf{x}^T - 2\mu_p\mathbf{x}^T + \mu_p\mu_p^T)\right)\right], \\
&= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - \mathrm{tr}\left(\mathbb{1}_{S_c}\right) + \mathrm{tr}\left(\Sigma_p^{-1}\{\mathbb{E}_q(\mathbf{x}\mathbf{x}^T) - \mathbb{E}_q(2\mu_p\mathbf{x}^T) + \mathbb{E}_q(\mu_p\mu_p^T)\}\right)\right], \\
&= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - S_c + \mathrm{tr}\left(\Sigma_p^{-1}\{\Sigma_q + \mu_q\mu_q^T - 2\mu_p\mu_q^T + \mu_p\mu_p^T\}\right)\right], \\
&= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - S_c + \mathrm{tr}\left(\Sigma_p^{-1}\Sigma_q\right) + \mathrm{tr}\left(\Sigma_p^{-1}\{\mu_q\mu_q^T - 2\mu_p\mu_q^T + \mu_p\mu_p^T\}\right)\right], \\
&= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - S_c + \mathrm{tr}\left(\Sigma_p^{-1}\Sigma_q\right) + \mathrm{tr}\left(\Sigma_p^{-1}\{(\mu_q-\mu_p)(\mu_q-\mu_p)^T\}\right)\right], \\
&= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - S_c + \mathrm{tr}\left(\Sigma_p^{-1}\Sigma_q\right) + \mathrm{tr}\left((\mu_q-\mu_p)^T\Sigma_p^{-1}(\mu_q-\mu_p)\right)\right], \\
&= \frac{1}{2}\left[\left(\log\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) - S_c + \mathrm{tr}\left(\Sigma_p^{-1}\Sigma_q\right) + (\mu_q-\mu_p)^T\Sigma_p^{-1}(\mu_q-\mu_p)\right], \qquad \text{(B.2)}
\end{aligned}
$$

S_c is the dimension of the correlation matrix. For a prior equal to $P(\theta) = \mathcal{N}(0, \mathbb{1})$, the expression becomes:

$$
\mathrm{KL}(q(\theta)||P(\theta)) = \frac{1}{2}\left[\left(-\log\det(\Sigma_q)\right) - S_c + \mathrm{tr}\left(\Sigma_q\right) + (\mu_q)^T(\mu_q)\right]. \qquad \text{(B.3)}
$$

Note: During the dissertation, we use a different notation for the parameters of the variational posterior: $q(\theta) = \mathcal{N}(\mu_\phi, \Sigma_\phi)$.

B.2 Loss function for mini-batch

The loss function considering a mini-batch gets:

$$
\begin{aligned}
F(D,\phi) \approx & KL(q_\phi(\theta)||P(\theta)) - \frac{1}{N}\sum_{n=1}^{N}\log P(D|\theta_n), \\
= & KL(q_\phi(\theta)||P(\theta)) - \sum_{i=1}^{\mathscr{D}}\frac{1}{N}\sum_{n=1}^{N}\log P(y^{(i)}|x^{(i)},\theta_n), \\
= & KL(q_\phi(\theta)||P(\theta)) - \frac{\mathscr{D}}{B}\sum_{i=1}^{B}\frac{1}{N}\sum_{n=1}^{N}\log P(y^{(i)}|x^{(i)},\theta_n). \qquad \text{(B.4)}
\end{aligned}
$$

Here $\mathscr{D}$ is the training size, N is the number of Monte Carlo samples, and B is the batch size. The loss function is normally divided by the number of samples of the training dataset. This way the sum turns into an average :

$$
F(D,\phi) = \frac{1}{\mathscr{D}}KL(q_\phi(\theta)||P(\theta)) - \frac{1}{B}\sum_{i=1}^{B}\frac{1}{N}\sum_{n=1}^{N}\log P(y^{(i)}|x^{(i)},\theta_n). \qquad \text{(B.5)}
$$

Note: By default, when TensorFlow calculates the loss function, it averages over the mini-batch, hence, the factor $\frac{1}{B}$ is already being used, however, the first term is not, by default, divided by $\mathscr{D}$.

B.3 Law of total variance

The law of total expectation is formulated as:

$$\mathbb{E}[Y] = \mathbb{E}\left[\mathbb{E}[Y|X]\right] , \tag{B.6}$$

where Y and X are two random variables. Similarly, the law of total variance is expressed as

$$\mathrm{Var}[Y] = \mathbb{E}[\mathrm{Var}[Y|X]] + \mathrm{Var}[\mathbb{E}[Y|X]] . \tag{B.7}$$

The variance of a random variable Y is defined as $\mathrm{Var}[Y] = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2$. Here, we will illustrate how to deduce this formula using the law of total variance:

$$\begin{aligned}
\mathbb{E}[\mathrm{Var}[Y|X]] &= \mathbb{E}[\mathbb{E}[Y^2|X] - (\mathbb{E}[Y|X])^2] , && \text{(B.8)}\\
&= \mathbb{E}[\mathbb{E}[Y^2|X]] - \mathbb{E}([\mathbb{E}[Y|X])^2] , && \text{(B.9)}\\
&= \mathbb{E}[Y^2] - \mathbb{E}[(\mathbb{E}[Y|X])^2] . && \text{(B.10)}
\end{aligned}$$

In Eq.B.8, we use the definition of variance, and in Eq.B.10, we utilize the law of total expectation. Similarly, for the term $\mathrm{Var}[\mathbb{E}[Y|X]]$:

$$\begin{aligned}
\mathrm{Var}[\mathbb{E}[Y|X]] &= \mathbb{E}[(\mathbb{E}[Y|X])^2] - (\mathbb{E}[\mathbb{E}[Y|X]])^2 , && \text{(B.11)}\\
&= \mathbb{E}[(\mathbb{E}[Y|X])^2] - (\mathbb{E}[Y])^2 . && \text{(B.12)}
\end{aligned}$$

Again, in Eq.B.11, we apply the variance definition, and in the right term of Eq.B.12, we use the law of total expectation. Combining both terms:

$$\begin{aligned}
\mathrm{Var}[\mathbb{E}[Y|X]] + \mathbb{E}[\mathrm{Var}[Y|X]] &= \mathbb{E}[(\mathbb{E}[Y|X])^2] - (\mathbb{E}[Y])^2 + \mathbb{E}[Y^2] - \mathbb{E}[(\mathbb{E}[Y|X])^2] ,\\
&= \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2 = \mathrm{Var}[Y] ,
\end{aligned}$$

we arrive at the definition of the variance from the law of total variance.

B.4 Evidence lower bound

To gain a clearer understanding of the Evidence Lower Bound (ELBO), we can derive it using Jensen's inequality [89]. Jensen's inequality states that if $g()$ is concave, then $g(\mathbb{E}[X]) \geq \mathbb{E}[g(X)]$. Applied to the logarithm, the inequality takes the form: Jensen's inequality tells you that if $g()$ is concave then $g(\mathbb{E}[X]) \geq \mathbb{E}[g(X)]$, so for the logarithm:

$$\log \mathbb{E}[X] \geq \mathbb{E}[\log X] . \tag{B.13}$$

Now, let's explore the derivation of the ELBO using Jensen's inequality. Starting with the expression for the logarithm of the data likelihood:

$$\begin{aligned}
\log p(x) &= \log \int p(x,z) dz ,\\
&= \log \int p(x,z) \frac{q(z)}{q(z)} dz ,\\
&= log\, \mathbb{E}_q \left[\frac{p(x,z)}{q(z)} \right] . && \text{(B.14)}
\end{aligned}$$

By applying Jensen's inequality from Eq.B.13:

$$\log p(x) = log\,\mathbb{E}_q\left[\frac{p(x,z)}{q(z)}\right],$$
$$\log p(x) \geq \mathbb{E}_q\left[\log\frac{p(x,z)}{q(z)}\right]. \tag{B.15}$$

This is where the term "Evidence Lower Bound" originates. Moving forward, let's derive the expression for the ELBO Eq.3.30 :

$$\begin{aligned}
ELBO &= \mathbb{E}_q\left[\log\frac{p(x,z)}{q(z)}\right], \\
&= \mathbb{E}_q\left[\log p(x,z) - \log q(z)\right], \\
&= \mathbb{E}_q\left[\log p(z,x)\right] - \mathbb{E}_q\left[\log q(z)\right], \\
&= \mathbb{E}_q\left[\log p(x|z)\,p(z)\right] - \mathbb{E}_q\left[\log q(z)\right], \\
&= \mathbb{E}_q\left[\log p(x|z)\right] + \mathbb{E}_q\left[\log p(z) - \log q(z)\right], \\
&= \mathbb{E}_q\left[\log p(x|z)\right] + \mathbb{E}_q\left[\frac{\log p(z)}{\log q(z)}\right], \\
&= \mathbb{E}_q\left[\log p(x|z)\right] - KL(q(z)||p(z)).
\end{aligned} \tag{B.16}$$

Appendix C

Deep Dive Into the Dataset

Table C.1: Description for input of dataset 1 and 2 sampled with standard deviation present in Table 4.4.

	Dataset 1				Dataset 2				
Features	mean	σ	min	max	Features	mean	σ	min	max
M_1 ($M_\odot$)	1.553	0.320	1.000	2.372	M_1 ($M_\odot$)	1.552	0.317	1.000	2.381
M_2 ($M_\odot$)	1.553	0.320	1.000	2.363	M_2 ($M_\odot$)	1.553	0.317	1.000	2.371
M_3 ($M_\odot$)	1.553	0.320	1.000	2.347	M_3 ($M_\odot$)	1.553	0.317	1.000	2.370
M_4 ($M_\odot$)	1.553	0.320	1.000	2.378	M_4 ($M_\odot$)	1.553	0.317	1.000	2.386
M_5 ($M_\odot$)	1.553	0.320	1.000	2.371	M_5 ($M_\odot$)	1.553	0.317	1.000	2.378
R_1 (Km)	12.405	0.375	10.489	13.767	R_1 (Km)	12.406	0.456	10.053	14.301
R_2 (Km)	12.405	0.375	10.602	13.788	R_2 (Km)	12.405	0.456	10.247	14.458
R_3 (Km)	12.405	0.376	10.468	13.820	R_3 (Km)	12.406	0.457	9.847	14.386
R_4 (Km)	12.405	0.374	10.612	13.830	R_4 (Km)	12.404	0.456	9.777	14.221
R_5 (Km)	12.406	0.375	10.482	13.788	R_5 (Km)	12.405	0.456	10.133	14.335

Table C.2: Description for input of dataset 3 and 4 sampled with standard deviation present in Table 4.4.

	Dataset 3					Dataset 4			
Features	mean	σ	min	max	Features	mean	σ	min	max
M_1 ($M_\odot$)	1.553	0.317	1.000	2.379	M_1 ($M_\odot$)	1.553	0.317	1.000	2.379
M_2 ($M_\odot$)	1.553	0.317	1.000	2.379	M_2 ($M_\odot$)	1.553	0.317	1.000	2.379
M_3 ($M_\odot$)	1.553	0.317	1.000	2.373	M_3 ($M_\odot$)	1.553	0.317	1.000	2.373
M_4 ($M_\odot$)	1.552	0.317	1.000	2.374	M_4 ($M_\odot$)	1.552	0.317	1.000	2.374
M_5 ($M_\odot$)	1.553	0.317	1.000	2.359	M_5 ($M_\odot$)	1.553	0.317	1.000	2.359
R_1 (Km)	12.405	0.456	10.121	14.347	R_1 (Km)	12.405	0.455	10.128	14.347
R_2 (Km)	12.405	0.456	10.148	14.223	R_2 (Km)	12.406	0.456	10.148	14.270
R_3 (Km)	12.405	0.456	10.060	14.079	R_3 (Km)	12.406	0.456	10.060	14.129
R_4 (Km)	12.406	0.455	9.995	14.346	R_4 (Km)	12.406	0.455	9.995	14.346
R_5 (Km)	12.405	0.456	10.184	14.340	R_5 (Km)	12.405	0.456	10.184	14.340
$M_{1(0)}$ ($M_\odot$)	1.553	0.322	1.000	2.384	$M_{1(0)}$ ($M_\odot$)	1.553	0.322	1.000	2.384
$M_{2(0)}$ ($M_\odot$)	1.554	0.322	1.000	2.374	$M_{2(0)}$ ($M_\odot$)	1.554	0.322	1.000	2.374
$M_{3(0)}$ ($M_\odot$)	1.553	0.321	1.000	2.383	$M_{3(0)}$ ($M_\odot$)	1.553	0.321	1.000	2.383
$M_{4(0)}$ ($M_\odot$)	1.553	0.322	1.000	2.380	$M_{4(0)}$ ($M_\odot$)	1.553	0.322	1.000	2.380
$M_{5(0)}$ ($M_\odot$)	1.553	0.322	1.000	2.385	$M_{5(0)}$ ($M_\odot$)	1.553	0.322	1.000	2.385
Λ_1	574.87	728.76	2.034	5131.28	Λ_1	575.41	748.32	0	5303.05
Λ_2	574.57	728.27	0.008	5124.69	Λ_2	574.18	746.84	0	5701.24
Λ_3	574.86	728.84	1.234	4710.00	Λ_3	575.22	748.13	0.001	5511.38
Λ_4	576.31	729.39	1.269	4831.33	Λ_4	576.38	748.89	0.001	5223.20
Λ_5	576.16	729.50	1.345	4690.01	Λ_5	576.56	749.45	0.001	5723.07

Table C.3: Description of the two outputs we used the speed of sound squared and the proton fraction.

	v_s^2[c^2]					y_p			
Features	mean	σ	min	max	Features	mean	σ	min	max
$v_s^2(n_1)$	0.044	0.005	0.031	0.076	$y_p(n_1)$	0.060	0.0078	0.031	0.093
$v_s^2(n_2)$	0.105	0.008	0.082	0.144	$y_p(n_2)$	0.069	0.0065	0.045	0.100
$v_s^2(n_3)$	0.177	0.012	0.143	0.244	$y_p(n_3)$	0.077	0.0073	0.054	0.128
$v_s^2(n_4)$	0.250	0.017	0.204	0.348	$y_p(n_4)$	0.085	0.0081	0.062	0.152
$v_s^2(n_5)$	0.318	0.022	0.268	0.439	$y_p(n_5)$	0.094	0.0086	0.071	0.170
$v_s^2(n_6)$	0.377	0.028	0.322	0.524	$y_p(n_6)$	0.102	0.0088	0.080	0.185
$v_s^2(n_7)$	0.425	0.034	0.359	0.586	$y_p(n_7)$	0.109	0.0088	0.087	0.196
$v_s^2(n_8)$	0.462	0.039	0.381	0.634	$y_p(n_8)$	0.115	0.0086	0.094	0.204
$v_s^2(n_9)$	0.490	0.043	0.396	0.668	$y_p(n_9)$	0.121	0.0083	0.100	0.211
$v_s^2(n_{10})$	0.511	0.047	0.408	0.695	$y_p(n_{10})$	0.125	0.0079	0.105	0.216
$v_s^2(n_{11})$	0.525	0.050	0.415	0.719	$y_p(n_{11})$	0.128	0.0076	0.109	0.220
$v_s^2(n_{12})$	0.535	0.053	0.421	0.739	$y_p(n_{12})$	0.131	0.0072	0.113	0.223
$v_s^2(n_{13})$	0.542	0.056	0.424	0.754	$y_p(n_{13})$	0.134	0.0068	0.117	0.226
$v_s^2(n_{14})$	0.546	0.058	0.426	0.767	$y_p(n_{14})$	0.136	0.0065	0.120	0.228
$v_s^2(n_{15})$	0.549	0.060	0.428	0.778	$y_p(n_{15})$	0.138	0.0062	0.122	0.230