Dissertation

submitted to the

Combined Faculty of Mathematics, Engineering and Natural Sciences

of Heidelberg University, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

Da Eun Kang

born in: Seoul, Republic of Korea

Oral examination: May 24, 2023

Determining physical properties of star-forming regions using conditional invertible neural network

Referees:
Prof. Dr. Ralf Stephan Klessen
Prof. Dr. Eva Grebel

Abstract

Star formation is one of the most fundamental subjects in astronomy where astronomers have been seeking answers to key questions: how efficiently stars form and how newly born stars affect their surroundings. Our understanding of star formation relies mostly on the observations of star-forming regions. However, it is a non-trivial task to interpret the observations because diverse physical processes are non-linearly coupled so the observational data are highly degenerate. Additionally, the ever-expanding volume of observational data in recent days necessitates a new method that analyses large amounts of data more quickly and effectively.

In this thesis, we introduce deep learning-based tools we have developed to efficiently and effectively interpret massive data of observed star-forming regions. We adopt the conditional invertible neural network (cINN) architecture specialised in solving the inverse problem of degenerate systems. We introduce the cINNs developed for cloud-scale observations and cINNs for individual star-scale observations. Our networks are very useful tools that can consistently and quickly analyse large amounts of data. We evaluate the performance of the networks, demonstrating that our networks predict physical properties accurately and precisely.

Zusammenfassung

Die Sternentstehung ist eines der grundlegendsten Themen in der Astronomie, bei dem Astronomen nach Antworten auf Schlüsselfragen gesucht haben: wie effizient Sterne entstehen und wie neugeborene Sterne ihre Umgebung beeinflussen. Unser Verständnis der Sternentstehung stützt sich hauptsächlich auf die Beobachtungen von Sternentstehungsgebieten. Es ist jedoch eine nicht triviale Aufgabe, die Beobachtungen zu interpretieren, da verschiedene physikalische Prozesse nichtlinear gekoppelt sind, sodass die Beobachtungsdaten stark entartet sind. Darüber hinaus erfordert die ständig wachsende Menge an Beobachtungsdaten in den letzten Tagen eine neue Methode, die große Datenmengen schneller und effektiver analysiert.

In dieser Doktorarbeit stellen wir Deep-Learning-basierte Tools vor, die wir entwickelt haben, um massive Daten von beobachteten Sternentstehungsregionen effizient und effektiv zu interpretieren. Wir übernehmen die Architektur des bedingt invertierbaren neuronalen Netzwerks (cINN), die auf die Lösung des inversen Problems degenerierter Systeme spezialisiert ist. Wir stellen die cINNs vor, die für Beobachtungen im Wolkenmaßstab entwickelt wurden, und cINNs für einzelne Beobachtungen im Sternenmaßstab. Unsere Netzwerke sind sehr nützliche Werkzeuge, die große Datenmengen konsistent und schnell analysieren können. Wir bewerten die Leistung der Netzwerke und zeigen, dass unsere Netzwerke physikalische Eigenschaften genau und präzise vorhersagen.

Contents

List of Figures

List of Tables

1

Introduction

1.1 About this Thesis

The key questions in astronomy that astronomers have been trying to answer over a long period of time are in fact not very different in essence from the questions that the general public can think of when looking at the night sky or images of the universe observed via a telescope. *How did those things we're looking at form? Why are those in so many diverse shapes and colours? How many of those things are there in the universe? Are galaxies or stars constantly forming?* Even if it does not start out of the curiosity about the universe, people often ask questions similar to what astronomers do, starting from philosophical questions about their origin. *Where do we come from, where did the materials that make up our body come from, how did our planet form and what will happen in the future?*

For a long time, astronomers have been trying to solve many of these questions and astronomy has been divided into many fields in detail. The questions that astronomers focus on in modern astronomy have deepened enough for the public to feel distant from astronomy, but "star formation", the seemingly simple theme, still holds many questions that we have not fully answered yet and is one of the most fundamental subjects of astronomy. From studies of planets, stars and galaxies to cosmology, star formation has been treated as a key astronomical phenomenon because the formation, evolution and death of stars are the most basic set of processes that drive the overall evolution of the universe. The key questions about star formation in modern astronomy are: how efficiently stars are formed, how this efficiency is determined, how newly born stars affect the surrounding environment, etc.

The only information we can get from the universe is light that has reached Earth, emitted from celestial objects. As observation technology has developed remarkably in the past century, these days, we can collect light not only in the optical wavelength range but also in all wavelengths spanning from gamma rays to radios by using telescopes mounted either on Earth or in space.

However, there are inevitable and inherent difficulties in analysing and interpreting the observed light. When we receive 3D information about the object in the form of light, all of the line-of-sight direction information accumulates and is lost. In addition, phenomena in different scales occur simultaneously in astronomical objects but it is very difficult to disentangle them because we can only observe the final output of those phenomena. The different time scale between the universe and human beings is also one of the unsolvable difficulties. We rarely observe the immediate evolution of astronomical objects. Therefore, we understand the evolution of astronomical objects by analysing and comparing different objects in different evolutionary stages, which is not such a strict method because we cannot guarantee that different objects always undergo the same evolution.

To overcome the limitations of the scale of space-time, astronomers have begun theoretically modelling the universe. It has been possible to create virtual galaxies and universes and to investigate the evolution within the human time scale by including as many of the physical processes we know as possible in the theoretical model. Since the time scale that can be implemented through theoretical models depends on computational power, the field of theoretical simulation has also developed significantly with the development of computer technology. Because of the limited computation time and memory, it is not possible to include all the physics in the theoretical model, but theoretical studies play an important role in constraining the physical conditions required to explain the observed universe by examining how each physical process affects the evolution of the object or the universe. Furthermore, a model-fitting method, finding a model (a set of physical parameters) that best mimics the observation, is also widely used to analyse the observations.

Unlike in the past when it was difficult to obtain observational data of good quality, we can now rapidly obtain a large amount of high-quality observational data thanks to the advance in telescopes and instruments. Large observational surveys over the past decades such as Sloan Digital Sky Survey (SDSS) and Gaia mission have provided astronomers with vast amounts of data, which have led to many innovative discoveries. With the expanding volume of accumulated observations, finding the method of analysing observational data faster and in a consistent way has become a new important task in astronomy. This has made astronomers interested in machine learning, especially the deep learning technique.

Machine learning that can quickly solve tasks that humans cannot easily solve has been applied to many fields in astronomy and used as a new breakthrough. Astronomers are using machine learning for various tasks such as classifying numerous images quickly (Wu et al. 2019; Walmsley et al. 2021; Whitmore et al. 2021) or predicting physical properties from observational data (Fabbro et al. 2018; Ksoll et al. 2020; Olney et al. 2020a; Sharma et al. 2020a; Shen et al. 2022). In this thesis, we aim to develop machine-learning tools that help study star formation.

Deep learning, often referred to as an artificial neural network, is the subfield of machine

learning, inspired by the behaviour of the human brain. Among various types of network architectures, we apply one of the deep learning architectures called conditional invertible neural network (cINN) developed by Ardizzone et al. (2019a, 2021) in this thesis. The cINN is specialised in solving the ambiguous inverse problem. The translation of the physical system to the observation is often called the forward process while inferring the physical properties from the observations is referred to as an inverse process. The ambiguous inverse problem occurs in a degenerate system, which means the observations obtained from two different systems are almost similar or identical due to the information loss in the forward process. Given the mentioned inherent difficulties of analysing astronomical observations, many astronomical tasks are ambiguous inverse problems where cINN would be applicable.

We have developed cINN-based tools that can effectively and efficiently analyse large amounts of observed star-forming regions to understand star formation. Here in this thesis, we introduce and carefully evaluate our networks. We focus on spectral observation in the optical wavelength range which is easily observable on Earth and can be obtained from many instruments and telescopes. We developed two types of networks: one for the cloud-scale observations of star-forming regions and the other for observations of individual stars. The former case is designed to analyse clusters and star-forming clouds where it is not possible to resolve individual stars. The latter case aims to study stars with a mass similar to solar mass or smaller, targeting star-forming regions within the Milky Way where individual stars can be resolved.

1.2 Star formation

1.2.1 Birth of stars

Giant molecular clouds (GMCs), gravitationally bound massive and dense accumulations of molecules, are the birthplaces of stars. The most common component of the GMC is molecular hydrogen (H_2). As the molecule is the coldest phase of the interstellar medium (ISM), the temperature of GMC is also very low around 10–20 K (see Table 1 of Girichidis et al. 2020). GMCs range greatly in size and density. The characteristic density of GMC is $\sim$100 cm^{-3} though the density varies from 30 to 1000 cm^{-3}. Their sizes range from 20 to hundreds of pc and they cover a wide range of mass between 10^3 to 10^7 M$_\odot$ with a typical cloud mass of 10^4 M$_\odot$ (Goodwin 2013; Klessen & Glover 2016).

Hydrogen is the most abundant molecule in GMCs that contains other molecules as well (CO, NH_3, HCN, CS, etc.). However, hydrogen molecules are hardly visible when observing GMCs because H_2 lacks electric dipole-driven rotational- and vibrational transitions due to its symmetric homonuclear diatomic structure. The lowest transition of H_2 corresponds to the temperature of 510 K (Goldsmith et al. 2010) which is too hot to be emitted from GMCs with a typical temperature of $\sim$10 K. Instead, GMCs are bright in CO, the second most abundant molecule in GMCs. The temperature of $J = 1 \rightarrow 0$ transition of CO is 5.5 K (Draine 2011), therefore, CO is used as a primary tracer of GMCs. In most cases, the amount of molecular hydrogen in the GMC is obtained using a conversion factor between CO emissions and H_2 column density (N_{H_2}), so-called X-factor, assuming a constant conversion factor. However, measuring the accurate amount of hydrogen in GMCs remains a challenging task because the conversion factor can highly vary depending on the environment (Wolfire et al. 2010; Glover & Mac Low 2011).

The density inside the GMC is not uniform. Overdense regions within GMCs where $n_{H_2} \gtrsim 1000$ cm^{-3} are called clumps and even denser regions ($n_{H_2} \gtrsim 10^5$ cm^{-3}) are referred to as cores or prestellar cores. Clumps are associated with star cluster formation and have sizes of $\sim$1 pc and masses of a few hundred M$_\odot$, while individual stellar systems are formed in the cores that have sizes of $\sim$0.1 pc and masses of a few solar masses (Goodwin 2013).

The prestellar core is supported by forces opposing gravity such as the thermal pressure of the gas, turbulence, or magnetic field. The prestellar core collapses if its mass exceeds the critical mass M_J, the local Jeans mass (Jeans 1902)

$$\begin{aligned} M_J &= \frac{\pi}{6} \frac{c_s^3}{G^{3/2}\rho^{1/2}} \\ &\approx 2\ \mathrm{M}_\odot \left(\frac{c_s}{0.2\ \mathrm{km\ s^{-1}}}\right)^3 \left(\frac{n}{1000\ \mathrm{cm^{-3}}}\right)^{-1/2} \end{aligned} \tag{1.1}$$

where c_s is the sound speed and n is the number density of the gas. For a typical molecular cloud

with a temperature of ∼10 K, the sound speed is ∼0.2 $\mathrm{km\,s^{-1}}$. Thus, a prestellar core with a density of 1000 $\mathrm{cm^{-3}}$ collapses once it gathers a mass around a solar mass. Once the prestellar core collapses, a protostar with a size of ∼1 AU is formed in the contracting core. Due to the continuing mass accretion, the temperature of the protostar rises until it reaches the condition of hydrogen fusion at the centre, entering the main sequence.

Depending on the size of the protostellar core and the accretion environment of the protostar, newly formed stars have a wide range of masses. The distribution of the stellar masses that just entered the main sequence is called the initial mass function (IMF). The shape of the IMF plays a crucial role in the evolution and fate of the GMC and star cluster. The observed IMFs appear in similar shapes in diverse environments (Bastian et al. 2010; Klessen & Glover 2016). One of the widely accepted parameterizations of the observed IMF is the combination of three power-law segments, proposed by Kroupa (2001, 2002):

$$\frac{\mathrm{d}N}{\mathrm{d}\log(M_*)} = \xi(M) \propto \begin{cases} M^{0.7}, & 0.01\ \mathrm{M_\odot} \leq M < 0.08\ \mathrm{M_\odot} \\ M^{-0.3}, & 0.08\ \mathrm{M_\odot} \leq M < 0.5\ \mathrm{M_\odot} \\ M^{-1.3}, & M \geq 0.5\ \mathrm{M_\odot}. \end{cases} \tag{1.2}$$

The upper mass limit of the IMF is still debated observationally. There are two important implications from the form of the observed IMF. Firstly, the formation of low-mass stars is highly preferred. The 50 % of the total mass is in stars with $\mathrm{M} \lesssim 1\ \mathrm{M_\odot}$ (Kroupa 2001). Secondly, massive stars are not only short-lived but also are rarely born.

How much gas in the GMC is converted into stars is one of the key questions in star formation. To quantify this, we define two representative quantities: star formation efficiency (ε) meaning the ratio of newly born stellar mass over gas mass and the star formation rate (SFR), the mass of stars created per unit of time. Instantaneous star formation efficiency at any moment in time (t) can be defined as,

$$\varepsilon(t) = \frac{M_*(t)}{M_*(t) + M_{\mathrm{GMC}}(t)}, \tag{1.3}$$

but it is useful to use star formation efficiency integrated over a certain time scale. We usually adopt a free-fall time (t_{ff}) as a unit time, which is a time for the material to collapse by gravitational force. A free-fall time is determined only by the density of the gas,

$$t_{\mathrm{ff}} = \sqrt{\frac{3}{32G\rho}} \sim 3\ \mathrm{Myr} \left(\frac{n}{100\ \mathrm{cm^{-3}}}\right)^{-\frac{1}{2}}, \tag{1.4}$$

where n is the number density of the gas. We use star formation efficiency over a free-fall time ($\varepsilon_{\mathrm{ff}}$) and SFR per free-fall time ($\mathrm{SFR_{ff}}$), i.e., the average star formation during the free-fall time.

Assuming that the entire gas in the GMC is consumed to form stars, the lifetime of the cloud is determined by the depletion time (t_{dep}) i.e., the amount of the gas in the GMC divided by the

star formation rate. If stars are formed simply in a free-fall timescale, then the star formation efficiency per free-fall time ($\varepsilon_{\rm ff}$) is 100% ($t_{\rm dep} = t_{\rm ff}$). Considering the free-fall time and mass of the molecular gas in the Milky Way, the $\rm SFR_{ff}$ of the Milky Way should be $\sim$100 $\rm M_\odot/yr$. However, the observed $\rm SFR_{ff}$ of the Milky Way and nearby spiral galaxies is around a factor of 100 smaller, i.e., 1 $\rm M_\odot/yr$ (Leroy et al. 2012), meaning $\varepsilon_{\rm ff}$ of $\sim$ 1 %. From various studies (Lada et al. 2010; Murray 2011), the observed star formation efficiency per free-fall time is $\sim$1 % and the integrated star formation efficiency during the lifetime of massive stars is about 2$-$10 % (Chevance et al. 2020a). Due to the low star formation rate, the estimated depletion time for nearby spiral galaxies is around 2 Gyr (Leroy et al. 2008). This means that a simple gravitational collapse does not explain star formation and that there exist mechanisms that suppress star formation. The stellar feedback by massive stars is regarded as one of the most important mechanisms to explain the inefficiency of the observed star formation.

1.2.2 Young stellar objects

Young stellar objects (YSOs) refer to stars in their early evolutionary stage before entering the main sequence. YSOs are commonly classified into Classes 0, I, II, and III according to the characteristics of the observed spectral energy distributions (SEDs). The last stage, Class III, is also called the pre-main sequence phase.

Once the prestellar core meets the Jeans criteria, the molecular gases in the core gravitationally collapse, forming a protostar in the centre. The molecular gas of the core encloses the protostar (i.e., envelope) and the protostar continues to accrete matter from the infalling envelope. As the accretion luminosity is completely absorbed by the optically thick envelope, the re-emitted emissions from the envelope are the dominant source of the observed SED that falls primarily in the sub-mm wavelength range (Goodwin 2013; Klessen & Glover 2016). In the Class 0 phase, most of the mass is still in the envelope.

Due to the conservation of angular momentum, most of the infalling materials do not accrete directly to the protostar but form a rotationally supported accretion disk. The protostar continues to accrete matters from the accretion disk. This stage is called Class I. Due to the magnetised molecular core, the accretion disk can launch the magnetically driven outflow along the polar axis of the system (Pudritz et al. 2006; Klessen & Glover 2016). The outflow starts to disperse the remaining envelope surrounding the protostar. Starting from the polar direction, the area affected by the outflow gradually widens, and the inner disk region becomes to be visible. This changes the observed SED, shifting the peak of the SED from the sub-mm regime towards the infrared wavelength. The period of Class 0 + I, usually referred to as protostellar lifetime, is $\sim$0.5 Myr (Dunham et al. 2014).

Once most of the remaining envelope is dispersed, the central protostar and disk become directly visible, entering the Class II phase. The YSOs in the Class II phase are classified into

T-Tauri stars ($\lesssim 2$ M$_\odot$) and Herbig Ae/Be stars ($2 \lesssim M \lesssim 8$ M$_\odot$) according to their masses. The T-Tauri stars have a surface temperature of 3000 – 7000 K (Appenzeller & Mundt 1989), corresponding to the spectral type of M – F type, while the surface temperature of Herbig Ae/B stars ranges from 8000 to 20000 K which corresponds to the F – B0 type. Their observed SEDs peaking in the infrared regime indicate the presence of the accretion disk. However, most of the mass is already accreted to the central star and the remaining accretion disk contributes only a few per cent of the overall mass budget (Klessen & Glover 2016). Planets begin to form in this evolutionary phase.

The Class III phase (also known as the pre-main sequence phase) starts once the remaining envelope is completely removed and the central star becomes fully visible. The circumstellar disk is also exhausted leaving only the debris disk where planet formation continues. The central star loses energy due to radiative cooling but the energy is compensated by gravitational contraction. The central star continues to contract until its core becomes hot enough for nuclear fusion. This period is determined by the Kelvin Helmholtz time scale, where the time scale for solar-type stars is ∼20 Myr (Kippenhahn et al. 2012; Klessen & Glover 2016). Finally, the star enters the main sequence, where nuclear fusion energy supports gravity. The lifetime of the main sequence is determined by the nuclear burning time,

$$\tau_{\rm MS} \approx 10^{10} \text{ years} \left(\frac{M}{M_\odot} \right)^{-2.5}. \tag{1.5}$$

The lifetime of the main sequence for stars with a solar mass is around 10 billion years.

In the case of low-mass stars, when the envelope dissipates and accretion halts, the star in pre-main-sequence contracts to reach the temperature for hydrogen fusion. However, in massive cores where massive stars are formed, the accretion rate is very high that hydrogen fusion begins before the envelope completely disappears and accretion finishes. So, when the star becomes fully visible after removing the envelope, massive stars have already entered the main sequence stage and are evolving. For this reason, unlike low-mass stars, it is difficult to observe when massive stars have just entered the main sequence stage (Kippenhahn et al. 2012).

1.2.3 Stellar feedback

Massive stars stay on the main sequence for a very short period (∼ a few Myr). During the early evolutionary stages of the low-mass stars, massive stars have already died out due to their short lifetime. Therefore, massive stars are indicators of ongoing star formation. Although they are very short-lived and account for a small mass fraction of the cluster mass (see IMF in Section 1.2.1), massive stars have a great influence on the star-forming region during their short lifespan via stellar feedback.

Stellar feedback is the energetic interaction between young massive stars and their birth-

place (for a review about stellar feedback, see Krumholz et al. 2014; Dale 2015; Klessen & Glover 2016), playing an important role in the evolution of the interstellar medium. Massive stars inject lots of energy and momentum into the surrounding environment through various mechanisms, blowing away and dispersing the gas, which finally can destroy the star-forming region (Dale et al. 2012; Kim et al. 2021; Grudić et al. 2022). Stellar feedback complicates the star-forming region morphologically and dynamically and changes the surrounding environment being unfavourable to star formation (e.g., low gas or hot temperature), suppressing further star formation. Therefore, stellar feedback is regarded as one of the key phenomena that can explain the observed low star formation rate (Krumholz & Tan 2007; Murray 2011).

Mechanisms of stellar feedback

Massive stars affect their surroundings through various mechanisms on different scales. There are three major stellar feedback mechanisms: radiations, stellar winds, and supernovae.

Young massive stars have a high surface temperature that is enough to emit significant amounts of ionising photons ($E > 13.6$ eV). These ionise the surrounding ISM, producing a H II region filled with fully ionised hydrogen. Assuming a spherical H II region with uniform density, the radius of the H II region

$$\begin{aligned} R_s &= \left(\frac{3Q}{4\pi\alpha_{\rm B} n^2}\right)^{1/3} \\ &\sim 3.17 \text{ pc } \left(\frac{Q}{10^{49}\text{ s}^{-1}}\right)^{1/3} \left(\frac{n}{100\text{ cm}^{-3}}\right)^{-2/3} \left(\frac{T}{10^4\text{ K}}\right)^{0.28}, \\ &\text{for } \alpha_{\rm B} \approx 2.56\times10^{-13}\text{ cm}^3\text{ s}^{-1}\left(\frac{T}{10^4\text{ K}}\right)^{-0.83} \end{aligned} \tag{1.6}$$

is determined by the density (n), recombination coefficient $\alpha_{\rm B}$, and Q, the rate of emission of hydrogen-ionising photons. This radius is called Strömgren (1939) radius. Taking the typical values of density and Q, the radius of H II region is ∼3 pc (Draine 2011). Due to its hot temperature of ∼10000 K, the H II region is highly overpressurised compared to the ambient gas, leading to the expansion (Oort & Spitzer 1955). This produces a dense shell composed of swept-up materials, surrounding the inner low-density cavity. The strong radiation field generated by massive stars not only ionises matter but also directly contributes to the expansion of the shell by its radiation pressure (Mathews 1967). Non-ionising photons mostly interact with the dust grains in the shell, transferring momentum and exerting radiation pressure (Krumholz et al. 2014).

Additionally, the radiation of massive stars drives strong stellar winds, the ejection of material from the surface of the star. There are a few types of stellar wind, but the stellar wind of a massive star is driven by line scattering between photons and ionised metals in the stellar atmosphere. The stellar wind of massive stars is very fast with a velocity of ∼1000 to ∼3000 km s^{-1} (Crowther

et al. 2016) and the mass loss rate of stars by the stellar wind is $\sim 10^{-5} - 10^{-4}$ $M_\odot$/yr, meaning that massive stars lose about a solar mass per 1 Myr. The mass loss rate increases even more if a massive star becomes the Wolf–Rayet star or luminous blue variable, the evolutionary phase after the main sequence. Stellar wind ejects a large amount of material into the surrounding ISM. This shocks the ambient material and pushes them away, forming an expanding wind-driven bubble.

The first two feedback mechanisms (radiation and stellar wind) are early-type of feedback which are dominant when the stars are young. At the last moment of death, massive stars deliver enormous energy to surrounding ISM through supernovae. In the last evolutionary stage of a massive star, once the star cannot sustain gravity anymore, it explodes violently, leaving a neutron star or black hole behind. Supernovae of this type are called core-collapse supernovae and they produce $\sim 10^{51}$ erg of energy and eject several solar masses of materials through the explosion, shocking the ambient ISM, heating and blowing away them. Supernovae are a late type of feedback. Stars with $M > 100$ $M_\odot$ reach supernovae after ~3 Myr. After about 4.5 Myr, stars with $M > 40$ $M_\odot$ all die out, and it takes about 30 Myr for all stars above 9 $M_\odot$ to end up supernovae (Ekström et al. 2012).

Impact of stellar feedback

Massive stars eject enormous energy via various mechanisms, but gravity counteracts stellar feedback and can lower their power. The actual impact of the stellar feedback on the star-forming region depends on the balance between gravity and stellar feedback. If the power of feedback is strong enough against gravity, it will push away the surrounding ISM, destroying the molecular cloud and suppressing further star formation. On the other hand, if the feedback is not strong enough, the gas in the cloud may disperse and be heated by the feedback, but the molecular cloud will still be gravitationally bound. Since massive stars are short-lived (< 30 Myr), after the death of massive stars, the gas will gravitationally gather and collapse again and be able to form the star. This type of inefficient stellar feedback can slightly lower the star formation rate but will not fully suppress the star formation. If the molecular cloud is very dense and the initial star formation rate is low, stellar feedback may have an almost negligible effect on the star formation rate.

There are many observational and theoretical studies on the impact of stellar feedback. Recent theoretical studies (e.g., Dale et al. 2012, 2014; Kim et al. 2016, 2018; Rahner et al. 2017, 2019; Geen et al. 2020) have shown that stellar feedback can disperse the molecular cloud via various mechanisms and observational studies also supported the negative stellar feedback (e.g., Chevance et al. 2020b; Barnes et al. 2021; Kim et al. 2021; McLeod et al. 2021). For example, Chevance et al. (2020b) and Kim et al. (2021) measured the lifetime of GMCs in nearby galaxies and found that molecular clouds disperse within 1 – 7 Myr after the birth of massive stars. Rahner et al. (2019) theoretically showed that star formation efficiency of 1–6 % is sufficient to destroy

a molecular cloud. However, stellar feedback is not always efficient in all star-forming regions. For instance, Dale et al. (2012) theoretically demonstrated that the impact of feedback can vary depending on the density of the molecular cloud, and Watkins et al. (2019) presented an example of the star-forming region where the feedback is inefficient due to the high density of the molecular cloud. Some studies suggest the possibility of positive feedback. For example, Shetty & Ostriker (2008) and Hobbs et al. (2015) explain that stellar feedback reduces star formation efficiency overall, but can induce star formation locally by compressing the gas. However, many studies have shown that stellar feedback plays an important role in suppressing star formation.

1.3 Deep Learning

1.3.1 Motivation for using deep learning

Our understanding of star formation relies heavily on the interpretation of the observed star-forming region. This starts with determining several physical properties of the object from the observation: stellar parameters (temperature, gravity) of individual stars, mass and age of the star cluster, star formation rate, etc. The translation from the physical system to the observation is often called the forward process, while the opposite process, inferring the physical properties from observations, is referred to as the inverse process. Most of the tasks in astronomy including the field of star formation are to solve the inverse problem.

The forward process is complicated but is usually obvious and well-known. However, the inverse process is often vague because of the information loss during the forward process. For example, because of the inevitable projection effect, we only obtain 2-dimensional information and 3D information is lost. Moreover, observation of astronomical objects is limited to the present time and line of sight direction. The information we finally obtain through observation is limited due to various information loss. This causes different physical objects to be mapped onto similar or almost identical observations, which makes the inverse problem more difficult to solve. We refer to such a system as a degenerate system.

The development of the theoretical models has helped solve the inverse problem. Recent theoretical models not only well describe the overall evolution of astronomical objects but also succeed to produce synthetic observations that are very similar to what we really observe. It became possible to interpret the observations by finding the best-fit model that corresponds to the observation. For example, the age of the cluster can be estimated by finding the best-fit isochrone model on the observed colour-magnitude diagram (CMD) of the cluster, and by fitting the observed SED with the synthetic SEDs, it is possible to estimate the stellar mass of the cluster.

However, finding the best-fit model using the classical fitting methods is a non-trivial task because the overall forward process is very complicated. To find the best-fit model, it is essential to choose a reasonable mathematical function that describes the link between physical parameters and observations. However, the overall forward process is usually non-linear and complex so most of the forward process is described through numerical simulations. It is almost impossible to find the analytic model that describes this overall process. In addition, the forward process is still highly dimensional that entails many free parameters, although some parameters can be fixed by simplifying the models through physical assumptions. Even if it is possible to find the function that describes the forward process, finding the best-fit model (i.e., a set of parameters) would be a computationally very expensive task due to the high dimensionality.

Furthermore, in the case of the degenerate inverse problem, there could be multiple solutions that satisfy the observation. Therefore, it is necessary to find a full posterior distribution of physical systems conditioned on the observation, which is usually a computationally expensive task. For these reasons, fitting the theoretical model to the observations using general fitting methods may require oversimplification of the theoretical model or result in an incomplete solution by finding only one possible system.

The deep learning method, which is also called an artificial neural network, can help solve these difficulties. Artificial neural networks (NNs; Goodfellow et al. 2016) link physical parameters to observable quantities through a statistical model rather than depending on a preselected mathematical model. By deepening the network, NN is able to solve high-dimensional problems as well. If we choose the deep learning architecture that provides the posterior distribution, we can also find the full solution to the degenerate inverse problem.

As the volume of accumulated observations ever-expand in recent days, it is important to develop time-efficient methods to analyse large amounts of data in a faster and more consistent way. Although training the neural network usually takes time, once trained, NN is able to analyse numerous data very quickly and consistently. In this thesis, we aim to develop deep learning tools that help analyse the observed star-forming regions effectively and efficiently.

1.3.2 General Concept

Artificial intelligence (AI) is the concept of creating intelligent machines, especially intelligent computer programs. Machine learning is a subset of AI, where a system automatically learns and improves from experience. Deep learning (artificial neural network), a subset of machine learning, is based on a collection of connected units called artificial neurons, that mimic the behaviour of the human brain.

Structure of a neural network

A neuron (or a node), the basic unit of a neural network, is composed of input, weight, bias, activation function, and output. Given an N-dimensional input $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, the first computation that occurs in the neuron is the linear combination of input:

$$a = \sum_{i=1}^{N} w_i x_i + w_0 \tag{1.7}$$

where parameters w_i are called weights and parameter w_0 is called a bias (Bishop 2006). Then the first output (a), known as activation, is passed to the differentiable, non-linear activation function, Φ, to give the final output of the neuron, $z = \Phi(a)$. A neural network consisting of multiple layers composed of multiple neurons inside is often called a deep neural network.

Designing a network involves choosing many design features such as the number of layers,

the number of neurons in one layer (width of the network), the input and output dimensions, etc. A fully connected network is a commonly used network design where every input neuron is connected to every output neuron.

Learning of the network

In the field of machine learning including deep learning, there are three main algorithms for how a machine learns: supervised learning, unsupervised learning, and reinforcement learning. In this thesis, we only apply supervised learning.

Supervised learning is a method of learning from large amounts of data where both inputs and outputs are well labelled. The network learns the output corresponding to the input, understands the relationship between them, and can predict the output of new input data based on past experience. In this thesis, we aim to develop machines that predict the physical properties of new observational data. Therefore, we apply a supervised learning approach in our networks.

Unsupervised learning finds patterns from unlabelled data. Given a set of input data without any corresponding target values, the machine has to identify hidden features from the input data. Unsupervised learning is used in clustering tasks. Reinforcement learning aims to train a system to complete a task within an uncertain environment. The system learns from receiving new environments and corresponding rewards, the measurement of the successfulness of the system output.

Training a network means finding proper values for the network components: the weights and biases. Here we only explain the general training method used in the supervised learning approach. Given a set of training data, a set of input ($\mathbf{x}$) and output ($\mathbf{y}$), a network aims to describe $\mathbf{y}$ through a function $f(\mathbf{x};\theta) \rightarrow \mathbf{y}$, where θ indicates a set of all trainable parameters (weights and biases). To find the f where $f(\mathbf{x};\theta)$ best approximates $\mathbf{y}$, we define the loss $L(\mathbf{y}, f(\mathbf{x};\theta))$, the difference between $f(\mathbf{x};\theta)$ and $\mathbf{y}$, to be optimised during the training. Loss can be defined in various ways depending on the task. The commonly used losses are mean square loss (L2 loss), mean absolute loss (L1 loss), and cross-entropy loss.

1.3.3 Conditional Invertible Neural Network

Invertible neural network (INN) is a novel deep learning architecture developed by Ardizzone et al. (2019b). Due to the invertibility of INN architecture, it can automatically learn the inverse process by learning the forward process. In this thesis, we utilise a conditional invertible neural network (cINN; Ardizzone et al. 2021) which is the extension of the INN architecture. Both INN and cINN are able to solve inverse problems and provide a posterior distribution of desired physical parameters on a given observation.

To apply the cINN or INN in a given inverse problem, we assume that information loss occurs during the forward process that causes the degeneracy in the system. This means that different

sets of physical parameters ($\mathbf{x}$; physical properties of the object) are mapped onto identical observations ($\mathbf{y}$; observable quantities such as luminosity). In such a system, the degenerate $\mathbf{y}$ cannot uniquely explain the corresponding $\mathbf{x}$. INN solves the problem by introducing the latent variables ($\mathbf{z}$), which capture the information loss during the forward process, and by making a bijective mapping that could not be achieved with $\mathbf{x}$ and $\mathbf{y}$ alone. The original INN architecture links $\mathbf{x}$ and a unique pair of [$\mathbf{y}$, $\mathbf{z}$], making a bijective forward mapping $f(\mathbf{x}) = [\mathbf{y}, \mathbf{z}]$ and a inverse mapping $\mathbf{x} = f^{-1}(\mathbf{y}, \mathbf{z}) = g(\mathbf{y}, \mathbf{z})$.

However, there are a couple of limitations in the INN architecture: the forward process has to be deterministic and there are some requirements towards the intrinsic dimensionalities of $\mathbf{x}$ and $\mathbf{y}$. On the other hand, the cINN architecture avoids these problems by slightly shifting its structure. While it also introduces latent variables for the same purpose, it uses a different mapping system by inputting the observations $\mathbf{y}$ in both the forward and inverse process as a condition $\mathbf{c}$: $f(\mathbf{x}; \mathbf{c} = \mathbf{y}) = \mathbf{z}$, $\mathbf{x} = g(\mathbf{z}; \mathbf{c} = \mathbf{y})$ (Ardizzone et al. 2021). The cINN has the advantage of being free to choose the dimensions of $\mathbf{x}$ and $\mathbf{y}$. In the case of the INN which links $\mathbf{x}$ and a pair of [$\mathbf{y}$, $\mathbf{z}$], zero padding is necessary if the dimension of $\mathbf{x}$ is smaller than the dimension of [$\mathbf{y}$, $\mathbf{z}$] (Ardizzone et al. 2019b). However, in the cINN, $\mathbf{y}$ can have an arbitrarily large dimension regardless of the dimension of $\mathbf{x}$.

During training, we prescribe the latent variables to follow a standard normal probability distribution $p(\mathbf{z}) = N(\mathbf{z}, 0, \mathbf{I})$ with zero mean and unit width, where $\mathbf{I}$ is the identity matrix with a dimension of $\dim(\mathbf{z}) \times \dim(\mathbf{z})$. The posterior distribution of physical parameters, $p(\mathbf{x}|\mathbf{y})$, is estimated on the basis of the inverse mapping $f^{-1} = g$. Following the inverse process $\mathbf{x} = g(\mathbf{z}; \mathbf{c})$, the posterior distribution is a transformation of the known distribution $p(\mathbf{z})$ to x-space, conditioned on the observation. So, the posterior distribution for a given observation $p(\mathbf{x}|\mathbf{y})$ is determined by sampling the latent variable following the prior distribution and using the inverse process g.

1.4 Structure of the thesis

The main body of this thesis is structured with the following four main chapters. In Chapter 2, we introduce our first cINN-based tool for the analysis of cloud-scale star-forming region observations. In this study, we train the cINN using synthetic H II region models produced by using WARPFIELD-EMP pipeline (Pellegrini et al. 2020). The network introduced in this chapter provides a full posterior distribution of seven physical parameters of star-forming regions by using the luminosity of 12 optical emission lines which are commonly observable in star-forming regions. We evaluate the performance of the network using WARPFIELD-EMP models not used for the training.

In Chapter 3, we present an upgraded version of a cINN for the analysis of cloud-scale observations (Noise-Net), as a follow-up study of the first work (Chapter 2). Noise-Net uses the observation errors as an additional input of the network and learns the influence of errors on the parameter prediction, providing a posterior distribution of parameters reflecting the observational errors. We compare the performance of the new version with the first version and show that the Noise-Net can predict accurately even when the error is large.

In the next chapter (Chapter 4), we present the cINN that diagnoses the individual stellar spectrum of young low-mass stars and provides posterior distributions of three stellar parameters. In this work, we train the networks using Phoenix stellar atmosphere models. We evaluate the performance of the networks in various ways by applying them to the real stellar spectrum of Class III stars as well as the synthetic models. Additionally, we analyse the spectral part which the network relies mostly upon to estimate each parameter. We investigate the gap between the synthetic models and reality and how this gap affects the network performance.

In the last chapter, Chapter 5, we summarise our findings and discuss the strengths and limitations of the application of cINN to astrophysical problems. We conclude by introducing the future studies planned.

2

Emission-line diagnostics of H II regions using conditional invertible neural networks

This chapter is based on the paper Kang et al. (2022) published in Monthly Notices of the Royal Astronomical Society (MNRAS) in 2022. I am the first author and carried out network construction and training, all of the data analysis and writing of the paper. Eric W. Pellegrini, the second author, produced the database of WARPFIELD-EMP synthetic H II region models used in this work. To construct the network based on the cINN architecture, we used the FREIA (Framework for Easily Invertible Architectures; Ardizzone et al. 2019b) developed by Lynton Ardizzone, the third author of this paper, which is based on the 'PYTORCH' library (Paszke et al. 2019). The co-authors provided valuable comments and helped improve the manuscript.

Abstract

Young massive stars play an important role in the evolution of the interstellar medium (ISM) and the self-regulation of star formation in giant molecular clouds (GMCs) by injecting energy, momentum, and radiation (stellar feedback) into surrounding environments, disrupting the parental clouds, and regulating further star formation. Information of the stellar feedback inheres in the emission we observe, however inferring the physical properties from photometric and spectroscopic measurements is difficult, because stellar feedback is a highly complex and non-linear process so that the observational data are highly degenerate. On this account, we introduce a novel method that couples a conditional invertible neural network (cINN) with the WARPFIELD-emission predictor (WARPFIELD-EMP) to estimate the physical properties of star-forming regions from spectral observations. We present a cINN that predicts the posterior distribution of seven physical parameters (cloud mass, star formation efficiency, cloud density, cloud age which means the age of the first generation stars, the age of the youngest cluster, the number of clusters, and the evolutionary phase of the cloud) from the luminosity of 12 optical emission lines, and test our network with synthetic models that are not used during training.

Our network is a powerful and time-efficient tool that can accurately predict each parameter, although degeneracy sometimes remains in the posterior estimates of the number of clusters. We validate the posteriors estimated by the network and confirm that they are consistent with the input observations. We also evaluate the influence of observational uncertainties on network performance.

2.1 Motivation

Stellar feedback, the interaction between young massive stars and their birthplace, plays a crucial role in the evolution of the interstellar medium (ISM) in galaxies. Once massive stars are formed in giant molecular clouds (GMCs), they inject a large amount of energy and momentum into the surrounding environment, which leads to the destruction of their birthplace and suppression of further star formation in the molecular cloud (for a review about stellar feedback, see Krumholz et al. 2014; Klessen & Glover 2016).

There are diverse stellar feedback mechanisms that can act against gravity such as supernovae, stellar winds, radiation pressure, or photoionisation. Different feedback modes are coupled non-linearly and complicate feedback-influenced regions like H II regions and photodissociation regions (PDRs) morphologically and dynamically. Recent studies theoretically describe stellar feedback models including several different feedback mechanisms (e.g., Dale et al. 2014; Rahner et al. 2017, 2019; Kim et al. 2018; Ali & Harries 2019; Geen et al. 2020; Grudić et al. 2022) and apply their models to characterise observed star-forming regions (e.g., Pellegrini et al. 2011; Rahner et al. 2018; Rugel et al. 2019). However, finding a best-fitting model for observations using classical forward modelling methods is difficult and limited in parameter space due to the high dimensionality, nonlinearity, and degeneracy of the complex stellar feedback process.

Artificial neural networks (NNs; Goodfellow et al. 2016) link physical parameters to observational measurements through a statistical model rather than depending on a preselected physical model. NNs have been applied in various astronomical studies, e.g., to predict physical parameters (e.g., Fabbro et al. 2018; Ksoll et al. 2020; Olney et al. 2020a), for classification (e.g., Wu et al. 2019; Sharma et al. 2020a; Wei et al. 2020), and to identify structures in maps and images (Abraham et al. 2018). In this study, we apply NNs to estimate the physical properties of stellar feedback and star-forming regions from observations.

We adopt a supervised learning approach that trains the NNs based on a large database, which contains both the target physical parameters and corresponding observables, and is built from synthetic models or well-determined observations. The forward process that translates parameters into observations is well-defined but involves a loss of information, such that different parameter sets are mapped onto identical observations, rendering the inverse process ambiguous and degenerate. For inverse inference, we need a full posterior distribution conditioned on the

observed measurements to fully characterise the ambiguity.

The invertible Neural Network (INN; Ardizzone et al. 2019b) is an architecture introduced to solve ambiguous inverse problems. Unlike classical neural networks which solve the inverse problem directly, INNs learn the forward process, using additional latent output variables to capture the information otherwise lost. Leveraging their invertible architecture, INNs then derive a solution for the inverse process for free. Conditioned on the observations and the latent variable distribution, INNs can predict full posterior distributions, which is advantageous to study multi-modality or correlations between parameters. In this paper, we use conditional invertible neural networks (cINNs; Ardizzone et al. 2019a, 2021), an extended class of INN, which has been applied in Ksoll et al. (2020) to predict stellar parameters from photometric observations. cINN has also been applied in several studies outside of the astronomical field such as particle physics or medical science (e.g., Bellagente et al. 2020; Trofimova et al. 2020).

We present an application of a cINN to estimate the physical properties of star clusters and star-forming clouds from spectral observations of H II regions. For simplicity, we regularly use the term H II region as a synonym for the entire star-forming complex. This is justified because we focus in this study on optical emission lines that mostly trace the ionised gas in the feedback-generated hot bubble and the surrounding dense expanding shell. As mentioned above, supervised learning requires a large amount of well-interpreted data to train the network. In this paper, we build a training database by using the WARPFIELD emission predictor (WARPFIELD-EMP; Pellegrini et al. 2020) which allows us to collect both cloud properties and corresponding observable quantities (i.e., line luminosity). WARPFIELD-EMP describes the evolution of a cluster, expanding bubble, and the surrounding cloud using the 1D stellar feedback code WARPFIELD (Rahner et al. 2017, 2019) and calculates detailed emission predictions based on the output from WARPFIELD with the help of CLOUDY, a photoionisation code (see Ferland et al. 2017), and the radiative transfer code POLARIS (see Reissl et al. 2016). WARPFIELD takes into account several feedback mechanisms (i.e., stellar winds, radiation pressure, thermal gas pressure, supernovae, and gravity) self-consistently. Although 1D models cannot describe detailed structures as much as 3D simulations, they typically do provide very good approximations to the H II region properties of interest here, and due to the reduced dimensionality, they are well suited for building large numbers of models required for training the cINN.

In this study, we focus on examining how accurately the network understands the hidden rules in the given training data and how accurately it can estimate the parameters of test synthetic models, which share the same physical system as the training data. In this study, we do not yet apply our network to real observations, but in the latter part of the paper, we discuss how observational uncertainties affect the network performance using a statistical approach and show the change of individual posterior distributions with increasing observational errors.

This chapter is structured as follows. First, in Section 2.2, we introduce a new WARPFIELD-

EMP database used for network training. We also review the physics of our synthetic models and present details of the new database. In Section 2.3, we introduce the structure of the cINN, our network setup, and the performance evaluation methods for the trained network. We present the prediction performance of our network in Section 2.4 and validate the predicted posteriors in Section 2.5. Section 2.6 discusses the degeneracy in the posterior distributions. We explore the influence of observational uncertainties on the network performance in Section 2.7. In Section 2.8, we discuss the physical assumptions inherent in the training data, and training methods to improve the network prediction. Finally, we summarise our main results in Section 2.9.

2.2 Training data

We need a large data set containing both the physical parameters we want to predict and the observable quantities of H II regions that they give rise to in order to train the neural network. However, it is difficult to collect such a large amount of well-analysed H II regions from observations. Instead, we use a database of numerous synthetic H II region models created through a pipeline known as the WARPFIELD emission predictor (WARPFIELD-EMP; Pellegrini et al. 2020) to train the network. WARPFIELD-EMP follows the evolution of massive star-forming clouds using a 1D stellar feedback model (WARPFIELD; Rahner et al. 2017, 2019) and predicts the time-dependent continuum and line emission radiated from the evolving clouds by using CLOUDY and POLARIS. Pellegrini et al. (2020) presented the first WARPFIELD-EMP database composed of synthetic H II region models evolved from 180 initial clouds with different combinations of cloud mass ($M_{\rm cl}$), star formation efficiency (SFE), and density ($n_{\rm H,0}$). This first database is sufficient to reproduce the BPT diagram of observed H II regions in NGC628, but the sampling of initial parameters (i.e., $M_{\rm cl}$ SFE, and $n_{\rm H,0}$) is not dense enough to be used for our network training. Therefore, in this paper, we present a new, extended database of synthetic models evolved from 10,000 initial WARPFIELD clouds, which is suitable for our network training. We will first explain about the WARPFIELD-EMP pipeline and the inherent physical mechanisms of the synthetic model, and introduce the new database.

2.2.1 WARPFIELD and WARPFIELD-EMP

WARPFIELD-EMP (Pellegrini et al. 2020) is a pipeline that predicts continuum and line emissions of evolving star-forming clouds by coupling these three codes: WARPFIELD, CLOUDY, and POLARIS. First, WARPFIELD (Rahner et al. 2017, 2019) is a 1D spherical symmetric stellar feedback model which explains the evolution of the isolated massive star-forming cloud. The initial star-forming cloud, which we call WARPFIELD cloud in this paper, is determined by four initial parameters: $M_{\rm cl}$, star formation efficiency, $n_{\rm H,0}$, and metallicity (Z).

The evolution of a WARPFIELD cloud begins (at $t = 0$) with the formation of the first

star cluster at the centre with a mass of $M_* = \mathrm{SFE} M_{\mathrm{cl}}$, where SFE is the star formation efficiency. The cluster ionises a large area called the Strömgren sphere and the stellar winds freely expand outwards from the centre. This initial expansion phase is very short and essentially covered in the first time step of the WARPFIELD evolution. Quickly, the cloud can be separated into distinct regions around the cluster: an inner wind zone (i.e., diffuse central bubble) and the surrounding dense shell which consists of swept-up material affected by stellar feedback. WARPFIELD describes the evolution of this system by solving the equation of motion of the dense shell, considering the effects of the stellar wind, radiation pressure, thermal gas pressure, supernovae, and gravity. We assume that the ionised and neutral/molecular phases of the shell are in quasi-hydrostatic equilibrium and that the evolution of the cloud can be characterised by four distinct evolutionary phases depending on the dynamics of the shell.

In the earliest stage, Phase 1, the shocked wind material reaches a very hot temperature, resulting in a fast adiabatic expansion. The influence of gravity and radiation pressure on the shell is negligible in this phase. It ends when the bubble loses the hot gas either because the gas cools by radiative cooling (t_{cool}) or because the bubble bursts and the hot gas escapes. In the latter case, we assume that the bubble bursts only when the shell sweeps up the whole material in the cloud (t_{sweep}, i.e., when $R_{\mathrm{shell}} = R_{\mathrm{initial\ cloud}}$), because we do not consider the three-dimensional structures such as inhomogeneity or asymmetries. Therefore, Phase 1 ends after $t = \min(t_{\mathrm{cool}}, t_{\mathrm{sweep}})$.

If the hot gas has cooled before the shell has swept away all of the cloud material ($t_{\mathrm{cool}} < t_{\mathrm{sweep}}$), the cloud evolution enters Phase 2. After Phase 1 or Phase 2, when the shell eventually sweeps up all of the cloud material ($t > t_{\mathrm{sweep}}$), the shell enters Phase 3, expanding into a low-density warm neutral medium outside the cloud. The shell expansion in Phase 2 or Phase 3 is dominated by the ram pressure exerted on the shell by stellar winds and supernovae, and by radiation pressure. The counteracting effect of gravity is now not negligible anymore. To take this into account, WARPFIELD includes at each time step both the gravity from the central cluster and the self-gravity of the shell in the calculation of shell dynamics.

The fate of the evolving cloud is divided into two cases depending on the effect of stellar feedback against gravity. When gravity becomes dominant, the shell stops expanding and begins to recollapse. Please note that, in this paper, we designate this recollapsing evolutionary phase as Phase 0 to conveniently feed it into the neural network. In Rahner et al. (2017), the evolution terminated when the shell recollapsed to a radius of 1 pc, but in Pellegrini et al. (2020) and this study, we assume that recollapse triggers the birth of a new star cluster followed by a second expansion, and that the star formation efficiency of the new burst of star formation is the same as for the original burst (i.e., $M_{*,\mathrm{second}} = \mathrm{SFE}(M_{\mathrm{cl}} - M_{*,\mathrm{first}})$). If gravity never becomes dominant, the shell continues to expand into the low-density ambient ISM without a recollapse. The density of the shell gradually decreases and it eventually becomes indistinguishable from the ambient

ISM. We terminate the calculation if the maximum density of the shell is smaller than 1 cm^{-3} for a period of 1 Myr or more.

WARPFIELD accounts for the time-dependent stellar feedback effect exerted on the shell during the whole evolution period. We assume that stellar mass within the star cluster follows a Kroupa initial mass function (Kroupa 2001). The evolution of the star cluster as a function of time is calculated with STARBURST99 (Leitherer et al. 1999, 2014) by using Geneva stellar evolution tracks for rotating stars (Ekström et al. 2012; Georgy et al. 2012, 2013). The spectral energy distribution (SED) of the cluster and surrounding cloud in the WARPFIELD model are time-dependent due to the evolving physical conditions and complicated because we may be looking at the combination of multiple clusters of different ages if recollapse occurred.

The information on the time-dependent physical conditions is then passed on to CLOUDY, a spectral synthesis code developed by Ferland et al. (2017). WARPFIELD-EMP uses the most recent version of CLOUDY, C17. It calculates both continuum emission and a large set of line emissions as well as the corresponding opacities as a function of position within the shell and surrounding natal cloud. If the shell swept up the entire natal cloud, we run CLOUDY only once to determine the emission from the shell, whereas if the natal cloud remains, we need to run CLOUDY twice; first for the shell and second for the static natal cloud. The output of the first CLOUDY run for the shell is used as the incident flux on the static cloud in the second CLOUDY run.

In the last step, we use POLARIS (Reissl et al. 2016, 2019) to solve the radiative transfer equation for rays passing through a 3D grid of the shell and the natal cloud. Based on the CLOUDY output, POLARIS calculates the absorption and emission from the dust so that we obtain the luminosity information about the continuum and the lines, taking into account the overall attenuation inside the shell and the cloud. As WARPFIELD is a 1D spherical symmetric model, we select a 3D spherical grid for POLARIS calculations. The output containing the three-dimensional attenuated luminosity information is then projected onto a 2D space. This process provides a 2D map of the velocity-integrated luminosity of any given line and the continuum, which is similar to spatially resolved observations. Finally, we spatially integrate the 2D map and obtain the 1D integrated luminosity of a series of emission lines that correspond to a given WARPFIELD model at a certain age.

WARPFIELD-EMP provides us with a vast amount of information on a H II region from the fundamental physical parameters such as mass or age to the final observable quantities like emission-line luminosity. Although our synthetic model does not account for the 3D structures and small-scale instabilities, because WARPFIELD is a 1D spherical symmetric model, it has been confirmed that WARPFIELD well describes the observations of feedback-affected regions around the star clusters (Rahner et al. 2018; Rugel et al. 2019). Moreover, Pellegrini et al. (2020) demonstrates that a WARPFIELD-EMP model is close to mimic the real observations, finding a

good agreement between the BPT diagram of WARPFIELD-EMP models and that of observed H II regions in NGC628.

2.2.2 Database

In this section, we introduce the new database of WARPFIELD-EMP synthetic H II region models, which provides a considerable extension of the first database presented in Pellegrini et al. (2020). As mentioned in Section 2.2.1, the main initial parameters of WARPFIELD models are cloud mass $M_{\rm cl}$, star formation efficiency SFE, initial cloud density $n_{\rm H,0}$, and metallicity Z. However, we fix the metallicity at solar metallicity so that the initial WARPFIELD cloud in our database is determined only by a combination of the other three parameters. There are several parameters that control the physical condition of the cloud but we constrain all the other parameters to be fixed in the database. For example, we can add the effect of magnetic field or turbulent pressure in WARPFIELD, but in this database, we turn off both of them. As the cloud begins its evolution, age t becomes the fourth independent parameter. So, each synthetic model, describing a certain evolution state of the cloud, is determined by the four parameters: $M_{\rm cl}$, SFE, $n_{\rm H,0}$, and t.

To create an evenly and densely populated database, we follow a total of 10,000 initial WARPFIELD clouds and randomly sample $M_{\rm cl}$, star formation efficiency, and $n_{\rm H,0}$ within the range of $10^{5-7}{\rm M}_\odot$, 2-10%, and 100-500 cm^{-3}, respectively. The ranges of the three parameters are similar or slightly narrower than those of the previous database. Then we evolve these uniformly distributed 10,000 initial clouds until an age of 30 Myr. Depending on the evolutionary condition, WARPFIELD may terminate the calculation earlier if the cloud already dissolved into the ambient ISM. We constrain the maximum age of the cloud to be 30 Myr to prevent infinite expansion, and also because we expect some of the base assumptions of the model (that the cloud is isolated, or that it is unaffected by external feedback and large-scale galactic dynamics) to break down for very old clouds.

WARPFIELD saves the evolution of the cloud in prescribed time intervals. For the new database, we adopt an interval of 0.1 Myr. However, we do not run CLOUDY and POLARIS for all of the saved models because of the limitations of data volume and computational time. Instead, as mentioned in Pellegrini et al. (2020), we run CLOUDY and POLARIS whenever the physical properties of the cloud or star cluster have changed sufficiently to result in a considerable change in the emission. For example, we calculate the model whenever the shell density at the boundary of the inner shell, shell radius, shell mass, or the ionising photon flux changes by 10%. We also calculate the model if the evolutionary phase of WARPFIELD changes. Therefore, the cloud age and the youngest cluster age are not sampled in constant intervals. Especially, the time interval of the old cloud is sometimes wider because the physical properties mentioned above do not change as rapidly as during the early phases of evolution. This wide time interval is exhibited

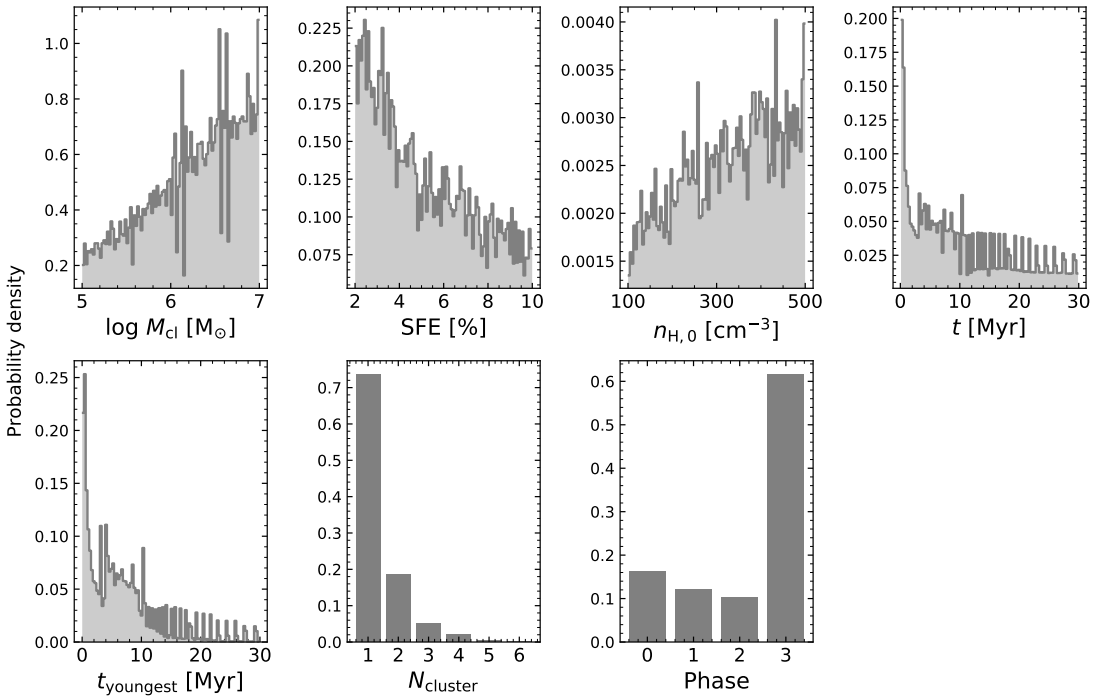

Figure 2.1: Distribution of seven physical parameters in the database used for the network training and evaluation. The database consists of 505,748 synthetic H II region models simulated by WARPFIELD-EMP (Pellegrini et al. 2020), which are evolved from 10,000 initial clouds. The first three parameters (cloud mass, star formation efficiency, and density) are initial conditions provided to WARPFIELD. The fourth, fifth and sixth panels correspond to the cloud age (as measured from the onset of star formation), the age of the youngest cluster in the cloud, and the number of clusters in the cloud, respectively. Phase is classified by the evolutionary state of the WARPFIELD cloud depending on its dynamics: Phase 1 is the energy-dominated phase, Phase 2 is momentum-dominated, Phase 3 is free-expansion, and Phase 0 denotes a recollapsing cloud (see the more detailed explanation in Section 2.2.1 or in Rahner et al. 2017; Pellegrini et al. 2020).

Table 2.1: List of seven physical parameters of H II region that our network predicts from the observation.

Parameter	Symbol
initial mass of star-forming cloud	$M_{\rm cl}$ [$M_\odot$]
initial star formation efficiency	SFE [%]
initial cloud density	$n_{\rm H,0}$ [$\rm cm^{-3}$]
age of the cloud	t [Myr]
age of the youngest cluster	$t_{\rm youngest}$ [Myr]
number of star clusters	$N_{\rm cluster}$
evolutionary phase of the cloud	Phase

as periodic patterns at $t > 10$ Myr in Figure 2.1. For the new database, we sample the time adaptively but, if needed, we can sample intermediate times by interpolating the evolution track of each WARPFIELD cloud. Our final sample consists of 505,748 synthetic H II region models in total with different values of $M_{\rm cl}$, SFE, $n_{\rm H,0}$, and t.

Although the initial WARPFIELD clouds were uniformly distributed in our parameter space, the final database has a non-uniform parameter distribution because each cloud evolves differently. Figure 2.1 shows the non-uniform distributions of the seven physical parameters listed in Table 2.1 that we aim to predict with the cINN. As seen in Figure 2.1, our database includes more H II region models corresponding to massive clouds or to clouds with smaller star formation efficiencies. This reflects the fact that in these models the power of stellar feedback against gravity is often not large enough to immediately disrupt the cloud, leading to one or more episodes of recollapse and hence a greater number of total output snapshots. On the other hand, more than 70% of our H II region models have only one stellar cluster, since in the majority of cases, the initial burst of feedback is enough to destroy the cloud and recollapse does not occur. In the case of the phase distribution, about 60% of the outputs correspond to clouds in Phase 3. The reason for this is that clouds typically remain in Phase 3 for a long period of time until the H II region dissolves into the ambient ISM, whereas clouds in the other evolutionary phases usually evolve rapidly and change into different phases.

In terms of training the network, we need to construct the training data to be as evenly distributed as possible, because over- or less-populated regions may introduce a bias to the trained network. In this case, the network might provide poor predictions for clouds with less popular characteristics in Figure 2.1. To remedy this problem, one could potentially post-process the database and augment it where needed to even out the parameter distributions. However, in this study, there are some difficulties in applying such an approach. First, the evolution of each cloud is a complex result as a function of four independent parameters. So, if we oversample less populated characteristics to achieve an even distribution for one parameter, this can lead in turn to a more biased distribution in the other parameters. For the same reason, it is also

Table 2.2: List of 12 emission lines whose luminosities are used as a condition for our network to predict the physical parameters listed in Table 2.1.

Line	Wavelength
[O II]	3726Å
[O II] (blend)	3727Å
[O II]	3729Å
Hβ	4861Å
[O III]	5007Å
[O I]	6300Å
Hα	6563Å
[N II]	6583Å
[S II]	6716Å
[S II] (blend)	6720Å
[S II]	6731Å
[S III]	9531Å

difficult to plan additional sampling measures, because it is not easy to predict the evolution of the cloud from a given initial condition. Moreover, WARPFIELD-EMP is not able to interpolate between different $M_{\rm cl}$, star formation efficiency, or $n_{\rm H,0}$, so it is not easy to fill less-populated regions by interpolation from the given database. Although the parameter distributions in the current database are not entirely optimal for training, we decide to train the network without any further processing. We discuss the influence of the bias in the training data on the network performance in Section 2.4.3 and 2.5.

We divide the database into two parts: a training set and a test set. 80% of the database is used to train the network (training set) and the remaining 20% is a set of held-out models (test set), which is used to evaluate the network training and the performance of the trained network. Training and test set have the same distribution because they are randomly selected. To sum up, our training set includes 404,599 H II region models. Each synthetic model has information about 7 physical parameters (i.e., $M_{\rm cl}$, SFE, $n_{\rm H,0}$, age, age of the youngest cluster, $N_{\rm cluster}$, and phase) as well as the luminosity of 10 optical emission lines within the wavelength range 3700Å – 9600Å. These lines are [O II] 3726Å, [O II] 3729Å, Hβ 4861Å, [O III] 5007Å, [O I] 6300Å, Hα 6563Å, [N II] 6583Å, [S II] 6716Å, [S II] 6731Å, and [S III] 9531Å). We also store the total strength of the [O II] and [S II] doublets, referred to hereafter as [O II] 3727Å (blend) and [S II] 6720Å (blend), since at low spectral resolution these lines will not always be separately resolved. Our choice of emission lines is motivated by their strength and the fact that they will be targeted in the forthcoming SDSS-V LVM survey of ionized gas in the Milky Way and other Local Group galaxies (Kollmeier et al. 2017). Most of these lines are also included in the recent

PHANGS-MUSE survey[1] of H II regions in nearby spiral galaxies (Emsellem et al. 2022) or the SIGNALS survey using the SITELLE spectrograph at the CFHT (Rousseau-Nepton et al. 2019). Please note that WARPFIELD-EMP provides information on many additional emission lines at frequencies ranging from the optical to the radio (see Table D1 and D2 in Pellegrini et al. 2020), but we restrict ourselves here to the optical emission lines most relevant for the above-mentioned surveys.

2.3 Neural Network

2.3.1 cINN

In this study, we apply a conditional invertible neural network (cINN; Ardizzone et al. 2021) to predict physical parameters of H II regions from their spectral observations. The cINN is an extension of the INN architecture described in Ardizzone et al. (2019b). Both INN and cINN are able to solve inverse problems and provide posterior distribution of desired physical parameters on a given observation. Ksoll et al. (2020) used the same cINN architecture and demonstrated that the cINN is able to successfully estimate stellar parameters such as stellar age and mass from HST photometry observations for individual stars. The advantage of these invertible architectures is that the network can automatically learn the inverse process when it is trained to approximate a known forward process.

To apply the cINN or INN in a given inverse problem we assume that information loss occurs during the forward process such that different sets of physical parameters ($\mathbf{x}$) are mapped onto identical observations ($\mathbf{y}$). Consequently, the degenerate $\mathbf{y}$ cannot uniquely explain the corresponding $\mathbf{x}$. By introducing the latent variables ($\mathbf{z}$), which capture the information loss during the forward process, we can make a bijective mapping that could not be achieved with $\mathbf{x}$ and $\mathbf{y}$ alone. The original INN architecture links $\mathbf{x}$ and a unique pair of [$\mathbf{y}$, $\mathbf{z}$], making a bijective forward mapping $f(\mathbf{x}) = [\mathbf{y}, \mathbf{z}]$ and a inverse mapping $\mathbf{x} = f^{-1}(\mathbf{y}, \mathbf{z}) = g(\mathbf{y}, \mathbf{z})$. A schematic overview of INN architecture is described in Ardizzone et al. (2019b) and Ksoll et al. (2020). However, this method does not apply to all types of inverse problems: the forward process has to be deterministic and there are some requirements towards the intrinsic dimensionalities of $\mathbf{x}$ and $\mathbf{y}$. On the other hand, the cINN, used in this paper, avoids these problems. While it also introduces latent variables for the same purpose, it uses a different mapping system by inputting the observations $\mathbf{y}$ in both the forward and inverse process as a condition $\mathbf{c}$: $f(\mathbf{x}; \mathbf{c} = \mathbf{y}) = \mathbf{z}$, $\mathbf{x} = g(\mathbf{z}; \mathbf{c} = \mathbf{y})$ (Ardizzone et al. 2021). This has the advantage that there are no assumptions or restrictions about the intrinsic dimensionalities of $\mathbf{x}$ and $\mathbf{y}$, meaning that effects such as stochastic modelling noise or measurement noise can also be accounted for.

[1] The exception is the [O II] doublet, which lies outside of the frequency range of ESO's MUSE integral field unit.

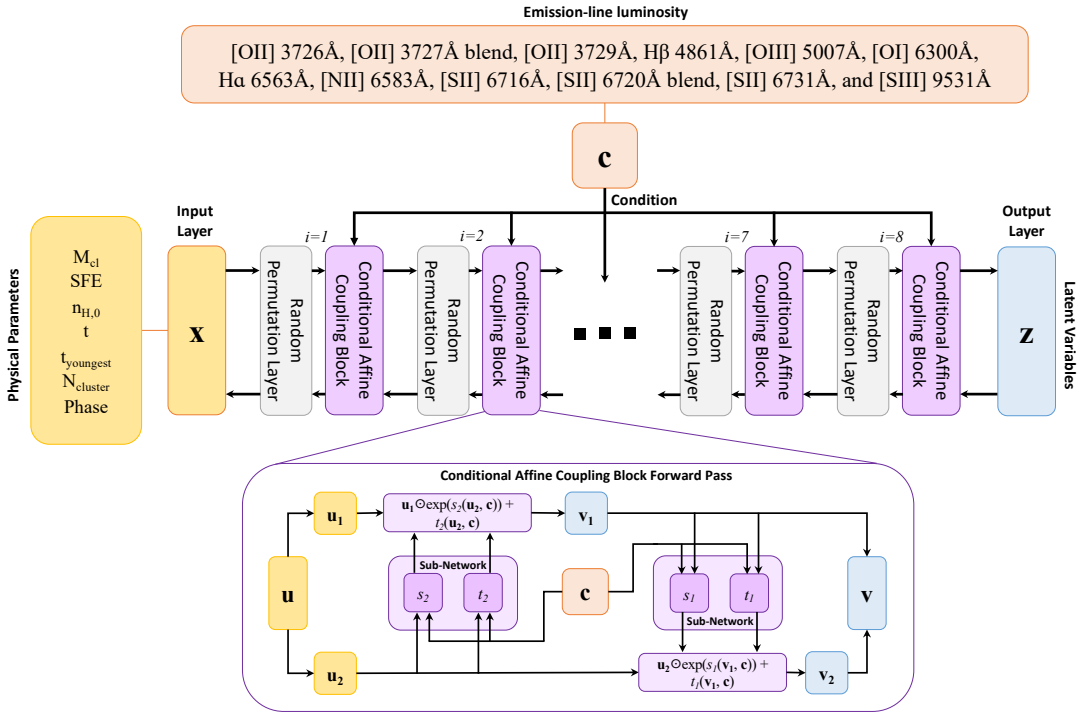

Figure 2.2: Schematic overview of our cINN architecture with the information of input parameters and observations for a condition. Our network consists of 8 affine coupling blocks interchanged with permutation layers. The zoom-in panel of the conditional affine coupling block shows how the information is passed through the block in the forward direction. For each affine coupling block, we use the GLOW configuration where the internal transformations $s_i()$ and $t_i()$ are described by a single sub-network.

The posterior distribution of physical parameters, $p(\mathbf{x}|\mathbf{y})$, is estimated on the basis of the inverse mapping $f^{-1} = g$. During training, we prescribe the latent variables to have a standard normal probability distribution $p(\mathbf{z}) = N(\mathbf{z}, 0, \mathbf{I})$ with zero mean and unit width, where $\mathbf{I}$ is the identity matrix with a dimension of $\dim(\mathbf{z})\times\dim(\mathbf{z})$. Following the inverse process $\mathbf{x} = g(\mathbf{z}; \mathbf{c})$, the posterior distribution is a transformation of the known distribution $p(\mathbf{z})$ to $\mathbf{x}$-space, conditioned on the observation. So, the posterior distribution for a given observation $p(\mathbf{x}|\mathbf{y})$ is determined by sampling the latent variable following the prior distribution and using the inverse process g.

The INN and cINN are very similar overall, but unlike the INN, the cINN has the advantage of being free to choose the dimensions of $\mathbf{x}$ and $\mathbf{y}$. In the case of the INN which links $\mathbf{x}$ and a pair of [$\mathbf{y}$, $\mathbf{z}$], zero padding is necessary if the dimension of $\mathbf{x}$ is smaller than the dimension of [$\mathbf{y}$, $\mathbf{z}$] (Ardizzone et al. 2019b). However, in the cINN, $\mathbf{y}$ can have an arbitrarily large dimension regardless of the dimension of $\mathbf{x}$ because the cINN uses observation $\mathbf{y}$ as a condition and connects $\mathbf{x}$ and $\mathbf{z}$, matching the dimension of each other.

According to Ardizzone et al. (2019b, 2021), cINNs and INNs consist of a series of affine coupling blocks following an architecture proposed by Dinh et al. (2016a). The schematic structure of our conditional affine coupling block is described in Figure 2.2. Each coupling block splits the input $\mathbf{u}$ into two parts $[\mathbf{u_1}, \mathbf{u_2}]$ and passes them through two invertible affine transfor-

mations. The output of the coupling block $\mathbf{v}$ is the concatenation of outputs from each affine transformation $\mathbf{v_1}$ and $\mathbf{v_2}$. The invertibility of the cINN and INN architecture is based on each reversible affine coupling block. The difference between cINN and INN is that the cINN uses the observation as an additional input in each affine transformation as follows:

$$\begin{aligned} \mathbf{v_1} &= \mathbf{u_1} \odot \exp(s_2(\mathbf{u_2}, \mathbf{c})) + t_2(\mathbf{u_2}, \mathbf{c}), \\ \mathbf{v_2} &= \mathbf{u_2} \odot \exp(s_1(\mathbf{v_1}, \mathbf{c})) + t_1(\mathbf{v_1}, \mathbf{c}). \end{aligned} \tag{2.1}$$

The internal transformations s_i and t_i are only evaluated in the forward direction even when the network implements the inverse process:

$$\begin{aligned} \mathbf{u_2} &= (\mathbf{v_2} - t_1(\mathbf{v_1}, \mathbf{c})) \odot \exp(-s_1(\mathbf{v_1}, \mathbf{c})), \\ \mathbf{u_1} &= (\mathbf{v_1} - t_2(\mathbf{u_2}, \mathbf{c})) \odot \exp(-s_2(\mathbf{u_2}, \mathbf{c})). \end{aligned} \tag{2.2}$$

Therefore, an arbitrary neural network that does not need to be invertible itself can represent the internal transformations s_i and t_i. In this study, we use a single sub-network for the internal transformations in each coupling block, adapting the GLOW (Generative Flow; Kingma & Dhariwal 2018b) configuration.

2.3.2 Network setup

Architecture

To construct our network with a cINN architecture we use the 'Framework for Easily Invertible Architectures' (FrEIA) for Python (Ardizzone et al. 2019b, 2021) which is based on the 'pytorch' library (Paszke et al. 2019), just as Ksoll et al. (2020) did in their study. As described in Figure 2.2, seven physical parameters of the H II region are the input $\mathbf{x}$ for our network, while the condition $\mathbf{y}$ is given by the twelve emission line luminosities that we can obtain from observations. Following the structure of the cINN architecture, the dimension of $\mathbf{z}$ matches that of $\mathbf{x}$ and we have seven latent variables in our network.

We build our network with 8 conditional affine coupling blocks, and for the sub-networks in the affine coupling blocks, we adopt a simple three-layer, fully-connected architecture with a width of 256, using rectified linear units (ReLU) as the activation functions. Additionally, we apply soft clamping on the sub-network output $s_i()$, which is introduced in Ardizzone et al. (2021) to prevent instability from the exponential component in Eq. 2.1.

To mix the information stream $\mathbf{u_1}$ and $\mathbf{u_2}$, we add permutation layers with a random orthogonal matrix after each coupling layer. The permutation layer is invertible and fixed during the training. Therefore, the final network is made up of 8 invertible blocks where each block is the combination of an affine coupling block and a permutation layer.

We train the network to minimize the maximum likelihood loss, as described in Ardizzone

et al. (2021) and Ksoll et al. (2020). The cINN model is trained until both the train-loss curve calculated by the training set and the test-loss curve calculated by the test set converge and deviation between the two curves is small enough. The training time varies depending on the batch size and the number of training epochs, even in the same network setting. Training our network for 300 epochs took about 2 hours with a batch size of 1024 and about 6 hours with a batch size of 256 when we used an NVIDIA GeForce RTX 2080 Ti graphic card. The GPU memory used for each case was 892 MB and 976 MB respectively.

Data Pre-processing

When training a network or using the network in practice, we apply pre-processed physical parameters and observations based on the procedure described by Ksoll et al. (2020). First, we transform the variables that have a relatively broad range of values in linear space to logarithmic space. In our case, we transform the emission line luminosities (y_i) into log-scale. We retain most of the physical parameters (x_i) in linear space because when generating the WARPFIELD-EMP database, we sampled each physical parameter within reasonable ranges. The exception is the cloud mass ($M_{\rm cl}$), which we sampled in logarithmic space in the beginning.

Next, we add artificial noise to relatively discretized parameters such as the $N_{\rm cluster}$ and phase, which have a sampling interval of 1 (see Figure 2.1). According to Ardizzone et al. (2021), smoothing out the distribution using a small amount of Gaussian noise helps the network converge in training. We smooth out the distribution of $N_{\rm cluster}$ and phase by adding Gaussian noise with a standard deviation of 0.05. The noise augmentation also improves the prediction performance of our network. A more detailed explanation of the effect of this smoothing process on the network performance is given in Section 2.8.2, where we compare the performance of our network with and without noise augmentation.

Last, we re-scale the distribution of physical parameters and observations by using linear transformations. We transform the physical parameter, x_i, following

$$\hat{x}_i = (x_i - \mu_{x_i}) \cdot \frac{1}{\sigma_{x_i}}, \tag{2.3}$$

where μ_{x_i} and σ_{x_i} are the mean and standard deviation of the physical parameter x_i in the database so that the re-scaled distribution of each parameter has zero mean and unit standard deviation. In the case of y_i, we first centre the observable ($\tilde{y}_i = y_i - \mu_{y_i}$) and whiten the observable matrix ($\hat{\mathbf{Y}} = \mathbf{W}_{\tilde{\mathbf{Y}}}\tilde{\mathbf{Y}}$), following Equation 35 in Hyvärinen & Oja (2000), so that the variance of each emission line is unity and the covariance of the matrix $\hat{\mathbf{Y}}$ is an identity matrix. The values of μ, σ, and $\mathbf{W}$ used in the linear transformations are calculated based on the whole database including both the training set and test set. We use the same values when training the network and utilising the trained network. When we predict physical parameters through the

inverse process of the cINN model, we transform y_i to $\hat{y}_i$ for the condition of the network and transform the output $\hat{x}_i$ to x_i.

2.3.3 Network evaluation methods

In this section, we describe how we evaluate the trained network using the held-out 101,149 models of the test set. As mentioned in Section 2.2.2, we split the original database and only use 80% of it for training and retain the rest for network evaluation.

We evaluate the network in the following five ways. First, we need to verify the network by using the latent variables of the test set, $\mathbf{Z_{test}}$. If the network is converged to a good solution, latent variables should follow the prescribed Gaussian normal distribution. This can be confirmed by checking whether the covariance matrix of $\mathbf{Z_{test}}$ is close to an identity matrix and whether the distribution of each latent variable follows the standard normal distribution.

The next three measures allow us to evaluate how accurately and precisely the network predicts each physical parameter with respect to the true value. To quantify the accuracy and precision, we compute the median calibration error ($e_{\rm cal}^{\rm med}$), median uncertainty at 68% confidence interval ($u_{68}^{\rm med}$), and the root mean square error (RMSE). The calibration error, our second criterion, evaluates the shape of the posterior distribution. At a given confidence interval q it is defined as

$$e_{\rm cal} = q_{\rm inliers} - q, \tag{2.4}$$

where $q_{\rm inliers}$ is the fraction of test models where the true value falls within the given confidence interval of the posterior distribution. The calibration error is an important evaluation index of the network because it represents the correctness of the shape of the posterior distribution (Ardizzone et al. 2019b). A negative calibration error indicates an overconfident network, which means that the predicted posterior distribution is too narrow, whereas a positive value means the opposite (under-confident network). We calculate $e_{\rm cal}^{\rm med}$, the median of the absolute value of calibration error over the confidence range from 0.01 to 0.99 in 0.01 confidence level interval for each physical parameter.

The third quantity is a median uncertainty interval at a 68% confidence level. The uncertainty interval is the width of the posterior distribution corresponding to the given confidence interval. We chose a confidence level of 68%, close to the width of $\pm 1\sigma$, and take a median value over the whole test set.

Fourth, we determine the root mean square error (RMSE) of the maximum a posteriori (MAP) point estimates, with respect to the ground true value (x^*). The RMSE of each parameter,

$$\mathrm{RMSE} = \sqrt{\frac{\Sigma_{i=1}^{N}(x_i^{\rm MAP} - x_i^*)^2}{N}}, \tag{2.5}$$

indicates how accurately our network can predict the true value.

We use a similar method as Ksoll et al. (2020) did to determine the MAP value from each posterior distribution. We perform a Gaussian kernel density estimation on a posterior distribution and find the point where the probability density becomes maximum. To find the suitable bandwidth of the kernel, we generally apply Silverman's rule of thumb (Silverman 1986) but, we also apply the Improved Sheather-Jones (ISJ) algorithm (Botev et al. 2010) which works better for a multi-modal distribution in some cases. For simplicity, we did not check the multi-modality of the distribution for all seven parameters, and only investigate the multi-modality in the $N_{\rm cluster}$ posterior distribution. The reason is that the multi-modality of the posteriors in our network usually results from the degeneracy in the $N_{\rm cluster}$ prediction, as we explain in Section 2.6. When the posterior distribution of $N_{\rm cluster}$ shows multi-modality, we apply a factor of five narrower width than that of Silverman's rule of thumb except for the posterior distribution of age. We apply the ISJ algorithm to the age posterior distribution which needs more careful kernel density estimation because it usually has both very narrow mode and wide mode in one distribution.

The final method we use to validate the network is to examine whether or not our cINN model constrains the physical parameters correctly. This is different from the previous three approaches of evaluating how accurately or precisely the network predicts the true values. Different H II regions can have similar emission line strengths, so our cINN model is designed to find all possible physical models $\mathbf{x}'$ conditioned on a given observation $\mathbf{y}$ ($= \mathbf{c}$). Therefore, even if the predicted value of parameters is different from the true value, it does not demonstrate the incapacity of our network. What we need to examine is whether the predicted model has the same emission line luminosity as the conditioned observation. Observational properties $\mathbf{y}$ cannot be obtained through the forward process of the cINN because $\mathbf{y}$ is used as a condition in both the forward and backward process. For that reason, we resimulate the emission line strengths of predicted models using WARPFIELD-EMP in the same way as we created the database.

As explained in Section 2.2.2, one synthetic H II region is determined by four independent physical parameters ($M_{\rm cl}$, star formation efficiency, $n_{\rm H,0}$, and the age of the cloud). We first evolve the cloud using WARPFIELD, taking as initial conditions the values of $M_{\rm cl}$, the star formation efficiency, and $n_{\rm H,0}$ predicted by the network. We stop the evolution of the cloud at the age predicted by the network and then use CLOUDY and POLARIS to compute the line emission produced at this time. In practice, we generally cannot collect the cloud information at the very age that we want because WARPFIELD records the evolution with a finite time interval. In this re-simulation process for network validation, we adopt a time interval of 0.05 Myr so that the age difference between the re-simulated model and the predicted model is certain to be less than 0.025 Myr.

2.4 Training Results

Once trained, the network is able to sample posterior distributions very efficiently. It takes less than 10 minutes to generate posterior distributions for the whole 101,149 observations in our test set, sampling 4096 times for each observation with an NVIDIA GeForce RTX 2080 Ti graphic card (the same graphic card used to measure training time in Section 2.3.2). On average, we can obtain posterior distributions of 170 observations per second through our network.

2.4.1 Training evaluation

In this section, we evaluate our trained network by using four of the five methods presented in Section 2.3.3: $\mathbf{Z_{test}}$, $e_{\rm cal}^{\rm med}$, $u_{68}^{\rm med}$, and RMSE of MAP estimates. First, we confirm whether the latent variables follow the prescribed Gaussian normal distribution on the test set ($\mathbf{Z_{test}}$) or not. In Figure A.1, we present the covariance matrix of $\mathbf{Z_{test}}$ and the probability distribution of each latent variable. The covariance matrix is close to the unit matrix and distributions of latent variables are almost following the Gaussian normal distribution with a residual of less than 0.03. These results confirm that our network is trained well.

The other three evaluation indices ($e_{\rm cal}^{\rm med}$, $u_{68}^{\rm med}$, and RMSE) are presented in Table 2.3. The median calibration errors of our cINN model are very low, around 1% or less in most cases. This means that the expected accuracy of our model is well calibrated to the confidence. The largest value is 2.34% for $N_{\rm cluster}$. This means that the network is the least calibrated for the $N_{\rm cluster}$ prediction among the seven parameters, but the error value of 2.34% is still very low and highly acceptable (Guo et al. 2017).

The second and third rows in Table 2.3 show the median uncertainty at 68% confidence interval (i.e., $u_{68}^{\rm med}$) for each parameter in the re-scaled parameter space ($\hat{x}$-space) and original parameter space (x-space), respectively. For the original parameter space, $u_{68}^{\rm med}$ has the same units as the physical parameter, whereas for the re-scaled parameter space $u_{68}^{\rm med}$ is dimensionless since it is the width of the posterior distribution with respect to the range of the parameter in the database. According to the $u_{68}^{\rm med}$ for the $\hat{x}$-space, $n_{\rm H,0}$ and $N_{\rm cluster}$ have on average wider predicted posterior distributions compared to the other parameters. Nevertheless, when it is transformed to real physical space, the uncertainty interval of $n_{\rm H,0}$ is around 20 cm^{-3} and that of $N_{\rm cluster}$ is 0.11 which is small compared to its sampling size of 1.

The RMSE of each parameter is also calculated for both the re-scaled parameter space and the physical parameter space (the fourth and fifth rows in Table 2.3). Age and $N_{\rm cluster}$ have relatively large RMSEs in the re-scaled space, implying again that the age of the cloud and $N_{\rm cluster}$ are difficult to predict. On the other hand, $M_{\rm cl}$ has a low RMSE in both x-space and $\hat{x}$-space; in x-space, the RMSE for the cloud mass is around 0.04 dex, corresponding to an uncertainty of less than 10%.

Table 2.3: Overview of our network performance using all of the 101,149 H II region models in the test set. For each parameter, we present median calibration error, median width of posterior distributions (uncertainty at 68% confidence interval), and mean accuracy of the MAP estimates (RMSE). We divide median width and mean accuracy into two types depending on whether we use re-scaled parameters ($\hat{x}_i$) or original parameters (x_i). The one using the re-scaled parameters is dimensionless whereas the one using the original parameters has the same unit as the parameters.

Performance measure	$\log M_{\rm cl}$ $\log [M_\odot]$	SFE [%]	$n_{\rm H,0}$ [$\rm cm^{-3}$]	t [Myr]	$t_{\rm youngest}$ [Myr]	$N_{\rm cluster}$	Phase
Median calibration error [%]	0.44	0.26	0.87	1.27	1.05	2.34	0.12
Median uncertainty at 68% confidence ($\hat{x}$)	0.0284	0.0945	0.1694	0.0176	0.0097	0.1512	0.0855
Median uncertainty at 68% confidence (x)	0.0154	0.2173	19.0633	0.1491	0.0637	0.1108	0.0995
RMSE ($\hat{x}$)	0.0823	0.2506	0.2846	0.5589	0.1210	0.6385	0.1724
RMSE (x)	0.0448	0.5760	32.0280	4.7285	0.7918	0.4681	0.2006

Considering the overall results in Table 2.3, our cINN model is well-calibrated and is able to predict each parameter accurately and precisely in general. However, some parameters such as $N_{\rm cluster}$ and the age of the cloud are relatively difficult to predict. On the other hand, $M_{\rm cl}$ and phase have small values in all three indices, meaning that our network predicts $M_{\rm cl}$ and phase stably.

2.4.2 Posterior probability distribution

In this section, we show representative posterior distributions conditioned on individual observations. We select three H II region models from the test set that exhibit typical shapes of the posterior distribution. Figure 2.3 shows the one-dimensional posterior distributions of each parameter for these examples.

The first model (left column) represents extremely well-predicted cases. As shown in the figure, the posteriors of this model have a clear unimodal distribution with narrow width, and the MAP estimate is very close to the true value marked by red vertical lines or red-edged bars for $N_{\rm cluster}$ and phase. In the case of age, for which our cINN model showed a less good performance in the previous section, the difference between the MAP estimate and the true value is less than 0.01 Myr, and the width of the posterior distribution represented by u_{68} is about 0.05 Myr. Though this model is one of the best examples, the accurate and precise unimodal posterior distribution is the most common characteristic found in our test set.

The other two models are examples of degenerate cases, for which posterior distributions usually have two or more peaks as shown in Figure 2.3. The second case in the middle column shows two different solutions, one with a higher probability and the other with a lower probability, whereas the last example has two solutions with similar probabilities. In the second example, $M_{\rm cl}$, star formation efficiency, and phase have a unimodal distribution (grey histograms) and the other four parameters have a bimodal distribution where true values always fall within the first mode with a higher probability density. The posterior distribution of the youngest cluster

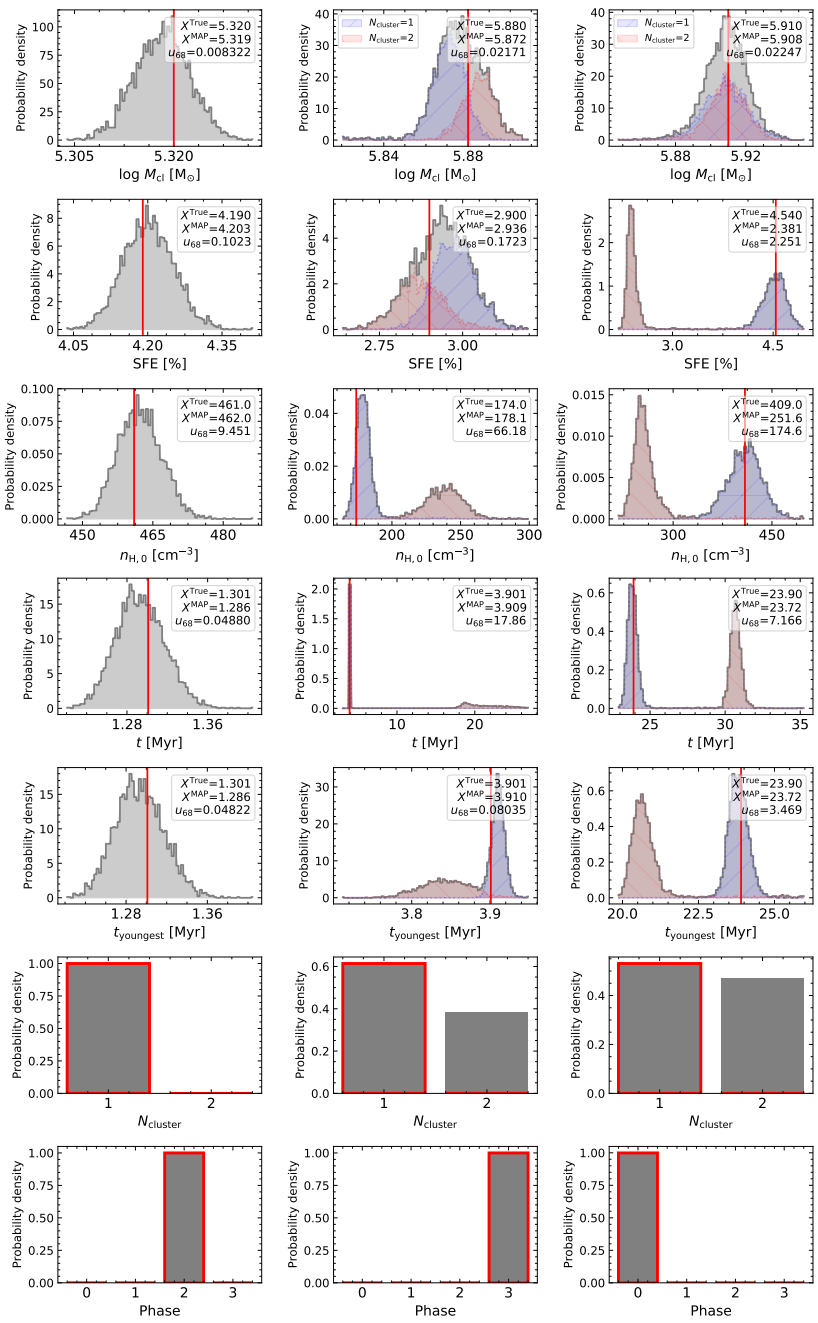

Figure 2.3: Posterior probability distributions (grey) of seven physical parameters. Each column corresponds to a different model selected from the test set. Red vertical lines (or red edges for bar-shaped histograms) denote the true parameter values of the model. The true value, the MAP estimate, and uncertainty at 68% confidence interval of the posterior distribution are presented in the upper right corner of each panel except for $N_{\rm cluster}$ and phase. The left column shows an example of an extremely well-predicted case. The other columns represent degenerate cases where the posterior of some parameters have multi-modal distribution. The middle column has a bi-modal distribution where one mode has a higher probability than the other, whereas the model in the right column represents a degenerate case where different solutions have similar probabilities. For the two degenerate cases, we divide the posteriors into two groups depending on the $N_{\rm cluster}$ posterior: blue distributions indicate posteriors that predict $N_{\rm cluster}$ as 1, and red distributions indicate posteriors that predict $N_{\rm cluster}$ as 2.

age has an overlapping area between the first mode and the secondary mode, suggesting that the unimodal distribution such as in $M_{\rm cl}$ could be decomposed into two different modes that are considerably overlapped with each other. The third case shows two distinct peaks with similar amplitudes without overlap in all parameters excluding $M_{\rm cl}$ and phase. The true value lies on the first mode in the case of cloud age and the youngest cluster age but in the case of star formation efficiency and $n_{\rm H,0}$, the true value falls in the secondary mode. Nevertheless, it is clear that the true value is within the range of the posterior distribution.

Degeneracy

For these degenerate examples, we divide the posteriors into two groups based on the $N_{\rm cluster}$ prediction: the first group predicting $N_{\rm cluster}$ as 1 and the second group predicting $N_{\rm cluster}$ as 2. Then we decompose the posterior distribution of the remaining parameters except for the phase into each group and present them in Figure 2.3 using different colours: blue for the first group ($N_{\rm cluster}$=1) and red for the second group ($N_{\rm cluster}$=2). The red and blue histograms in Figure 2.3 clearly show that the two groups classified by the $N_{\rm cluster}$ prediction correspond to the two different modes exhibited in the posterior distributions. Moreover, in the second example, the unimodal posterior distribution for $M_{\rm cl}$ and star formation efficiency is divided into two modes albeit with a wide range of overlap. As shown in these cases, most of the degeneracy revealed in the posterior distribution reflects the degeneracy in the $N_{\rm cluster}$ prediction. In other words, this kind of degeneracy suggests how other parameters should change to produce the same amount of emission line luminosity with more or fewer star clusters.

Both degenerate examples actually have only one cluster, or more precisely one stellar generation in the central cluster, so the red mode in Figure 2.3 suggests another solution with one more star cluster (stellar generation) which generates the same emission. In the case of the second example, the initial clouds of the red mode are on average more massive and denser but have smaller star formation efficiency compared to those of the blue mode (i.e., true model). The red mode shows that to emit the same amount of emissions as the true model, the cloud of the red mode should live longer till the second-star cluster is a bit younger than that of the blue mode. In the third example, the second solution requires a similar mass of cloud but with smaller star formation efficiency, smaller density, older cloud age, and younger age for the youngest cluster.

The degenerate solution demonstrates that our network understands the correlation between parameters that affect the emission line luminosity. Based on the posterior of cloud mass, star formation efficiency, and the number of clusters, we additionally calculate the total stellar mass of clusters in the cloud because stellar mass is a more direct index of the ionising power. As the youngest cluster dominates the ionising luminosity, we also calculate the mass of the youngest cluster. In Figure 2.4, we present the posterior distribution of these two parameters for the second example and the third example and divide the distribution into two groups based on the

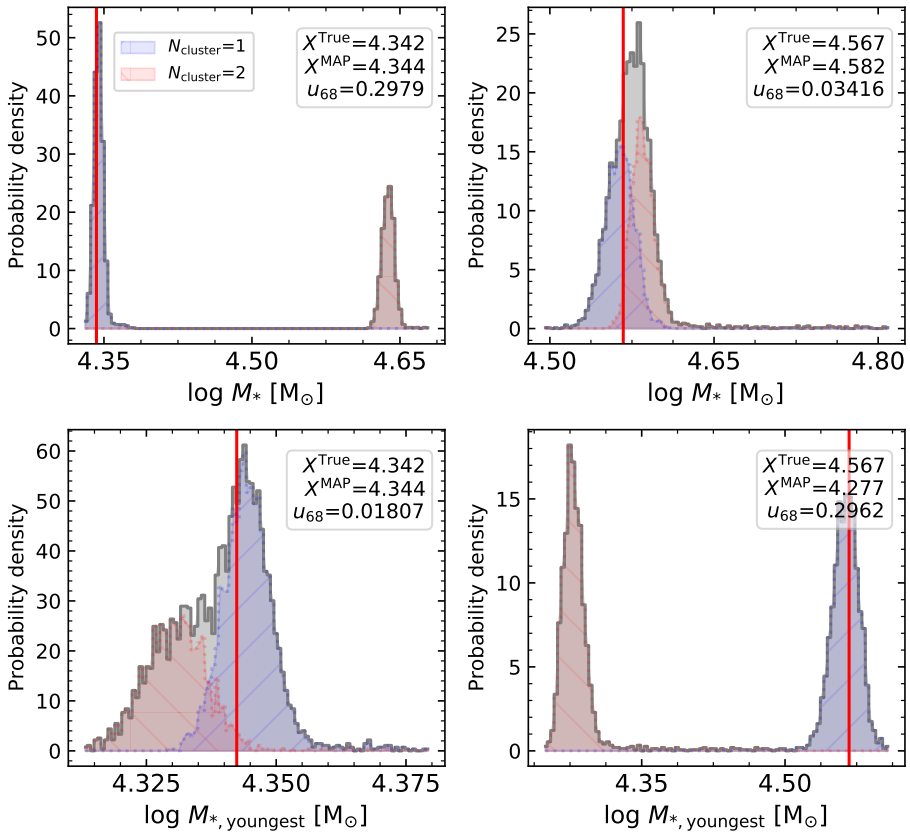

Figure 2.4: Posterior distributions (grey) of total cluster mass and the youngest cluster mass. The first and second columns correspond to the second example and the third example in Figure 2.3 respectively. Lines and legends are the same as in Figure 2.3. Posteriors are divided into two groups depending on the $N_{\rm cluster}$ posterior as shown in Figure 2.3: the blue mode whose $N_{\rm cluster}$ prediction is 1 and the red mode whose $N_{\rm cluster}$ prediction is 2.

$N_{\rm cluster}$ prediction in the same way as in Figure 2.3. In the case of the second example, the total stellar mass of the red mode with two clusters is a factor of two larger than that of the blue mode, whereas the youngest cluster mass is almost identical with only a 0.015 dex difference. The second cluster of the red mode and the cluster of the blue mode, which are the main luminosity sources of each mode, may have similar luminosity because they have similar mass and similar age around 3.8∼4 Myr. The small difference in the cluster age offsets the small difference in the cluster mass. The first cluster of the red mode is less luminous than the others in spite of its similar stellar mass because it is older than the others (∼20 Myr) so that all of its bright O-type stars and many of its brightest B-type stars are already dead. However, the first cluster slightly increases the total luminosity of the red mode. This additional luminosity is cancelled out by the larger amount of dust attenuation due to the higher cloud density of the red mode.

In Figure 2.3, the red mode of the third example shows a similar trend to the red mode of the second example, except for the density. However, Figure 2.4 shows that the situation is different from the second example. The total mass of the clusters in the red and blue modes

are almost the same, but the mass of each of the two clusters in the red mode is half of that in the blue mode. The age of the clusters is also different from the second example. All three clusters are older than 20 Myr, so they have already lost their brightest stars. In this case, the age difference between the individual clusters is not significant. The total luminosity of the blue and red mode is similar because it is determined by their total stellar mass. However, the stellar feedback of the red mode is weaker than that of the blue mode because the mass of the individual cluster is about a factor of two smaller. In order for the red mode to have two clusters and to evolve to a phase similar to that of the blue mode, the gravitational potential must be smaller to balance the weak stellar feedback. As the two modes have similar cloud mass, the red mode has a smaller density to reduce the gravitational potential. The results of these two degenerate examples reveal that our cINN model successfully learned the hidden rules in the training data and provides physically reasonable alternatives as well as the posteriors close to the true solution.

Re-simulating posterior models

We now investigate the emission line luminosity of the predicted posteriors to validate our network performance. As explained in Section 2.3.3, we reproduce the synthetic H II region model of the posterior and calculate the luminosity of 12 emission lines through WARPFIELD-EMP based on the four parameters that determine a unique star-forming cloud: $M_{\rm cl}$ SFE, $n_{\rm H,0}$, and age. The posterior distribution of each example model in Figure 2.3 consists of 4096 posteriors, but we only simulate 1024 posteriors of them.

In the middle panel of Figure 2.5, we present a 2D comparison of the [O III]/Hβ and [N II]/Hα line ratios – an example of a so-called BPT diagram (Baldwin et al. 1981) – for the entire 101,149 test set models. Here the colour indicates the number of different models at each point in this 2D histogram. The true locations of our three example models are represented by yellow stars. In each zoom-in panel of Figure 2.5, we compare the [N II]/Hα and [O III]/Hβ line ratios produced by the models sampled from the predicted posteriors with the true line ratio values (red lines) of each example model. The green circles show the area in which 68% of the posteriors are included from the centre of the distribution. The left panel corresponds to the first example model. We find that the line ratio distribution of the posterior samples is very narrow and lies close to the true values. The peak of the distribution is just shifted by about 0.002 dex from the expected point and the overall width of the distribution is around 0.01 dex. The green circle shows that 1σ of the distribution is within the 0.003 dex radius. Even though we only present two line ratios in the figure, we have confirmed that the re-simulated luminosity of all 12 emission lines also shows a good agreement with the true values with an average error of 0.013 dex in the logarithmic scale.

The second example (upper right panel in Figure 2.5) also indicates a good result. The reproduced line ratios are very narrowly distributed within a box of size 0.01 dex. The centre

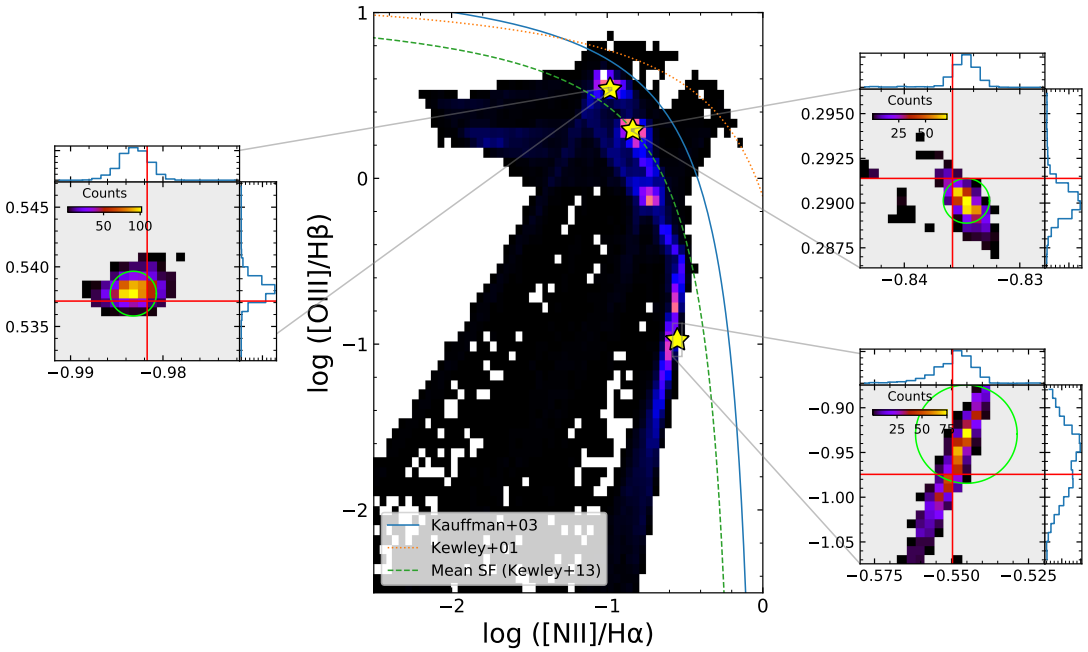

Figure 2.5: Middle panel: 2D histogram showing the [N II]/Hα and [O III]/Hβ line ratios for all of the models in the test set, where brighter colour indicates a higher number of models. Overlaid as yellow stars are the corresponding values of the models presented in Figure 2.3. Also shown are demarcation curves distinguishing star-forming galaxies and AGN, taken from Kauffmann et al. (2003) (blue solid line) and Kewley et al. (2001) (orange dotted line) as well as the mean line of star-forming galaxies from Kewley et al. (2013) (green dashed line). Star-forming galaxies are expected to sit below and to the left of the Kauffmann et al. and Kewley et al. lines. Zoom-in panels show the distribution of the line ratios that we recover if we sample the posterior distribution for each example model 1024 times and use the resulting values as input for new WARPFIELD-EMP calculations, as described in Section 2.3.3. The true line ratio values for each example model are represented by red lines in these zoom-in panels. The left panel corresponds to the first model, whereas the upper right and lower right panels correspond to the second and third models, respectively. Green circles in each zoom-in panel indicate the area in which 68% of the models are included from the centre of the distribution.

of the distribution is shifted by 0.0025 dex from the true values and 68% of the entire models are located within a radius of 0.002 dex. On the other hand, the last example (lower right panel in Figure 2.5) exhibits a wider distribution and a larger offset compared to the other two. The line ratio distribution is elongated along the y-axis over a range of 0.2 dex and its peak is shifted by about 0.04 dex. Although the overall results are worse than for the other two examples, still 68% of the posterior samples are within a 0.05 dex radius of the true values. We have verified that the luminosities of the 12 emission lines are in good agreement with the true values with an average error of 0.12 dex in log units, or around 30%.

Overall, the results of the reproduced emission lines of the posterior samples shown in Figure 2.5 demonstrate that our network is understanding the hidden rules in the training data and is able to provide reliable posteriors, especially in that the secondary solutions in two degenerate examples also match the observation successfully.

2.4.3 Overall performance

Now we extend the target to all 101,149 test models to probe the overall predictive performance of our network. We sample sets of latent variables 4096 times for each test model and measure the MAP estimates as the representative of each posterior distribution. We compare the true values of the seven parameters with all the obtained posterior estimates in Figure 2.6 or only with the MAP estimates in Figure 2.7. Please note that colours representing the number of models in the 2D histogram in Figures 2.6 and 2.7 are in logarithmic scale. For $N_{\rm cluster}$ and phase, we additionally present the confusion matrix of the 2D histogram, showing the per cent of the number of models in each point to the number of all models in the same column, to visualise the prediction performance depending on the true values. Most of the predictions are in excellent agreement with the true values albeit with large scatters around the one-to-one correspondence, and the scatter is much smaller when we compare MAP estimates with the true values.

Both figures indicate that it is difficult to predict the cloud age and the number of clusters, especially when the number of clusters is larger than three. In the age prediction in Figure 2.6, we find two characteristic features. The first one is that the older the true age, the wider the scatter around the one-to-one correspondence. We can find a similar feature in the age MAP estimates in Figure 2.7 and the result of the youngest cluster age in both Figures as well. Analysing the posterior distribution of various models, we notice that young H II regions, less than a few million years old, usually have very narrow age posterior distributions compared to old H II regions. The u_{68} values of the first model and the third model in Figure 2.3 clearly show this feature. A larger scatter around the true value for old H II regions reflects this general feature exhibited in most posterior distributions. The second feature in the age prediction is two parallel lines shifted by about 6 Myr above and below the one-to-one correspondence line. These parallel lines reflect solutions that are younger or older than the true age values due to the degenerate prediction of the number of clusters as shown in Figure 2.3.

In Figure 2.7, the MAP estimates of the cloud age are usually distributed around the true values or at younger ages, which is different to the case in Figure 2.6 using all posteriors estimates where outliers are scattered in both older and younger age range. The parallel lines shown in Figure 2.6 using all posterior estimates remain in Figure 2.7 using MAP estimates, but only the bottom line is clear in Figure 2.7. This is related to the first characteristic mentioned above that the age posterior distributions of young H II regions are narrower than those of old H II regions. When the posterior distribution has multi-modality, the width of the younger mode is by and large narrower than that of the older mode. This is because the probability density of the younger mode is higher even if the number of posteriors corresponding to the younger mode is similar to that of the older mode. Sometimes, even if most of the posterior samples belong to the older mode, the peak of the younger mode is determined as a MAP estimate because of

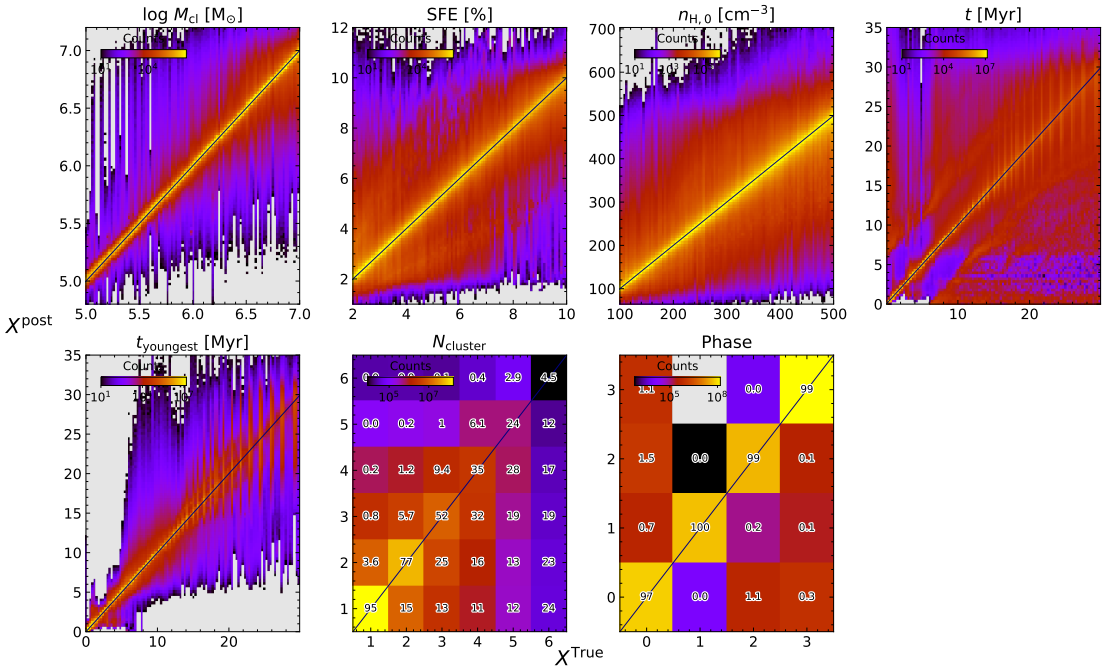

Figure 2.6: 2D-histogram comparing all posterior estimates predicted by our network with the true values of each parameter for all 101,149 models in the test set. We sampled 4096 posteriors for each test model. Colours indicate the number of models at each point in the two-dimensional histograms. Please note that we only plot areas with more than 10 models, leaving the otherwise part in grey. For the two discretized parameters ($N_{\rm cluster}$ and phase), we additionally present the confusion matrices on the 2D histograms that show the number of models in each point divided by the number of all models over the column in per cent.

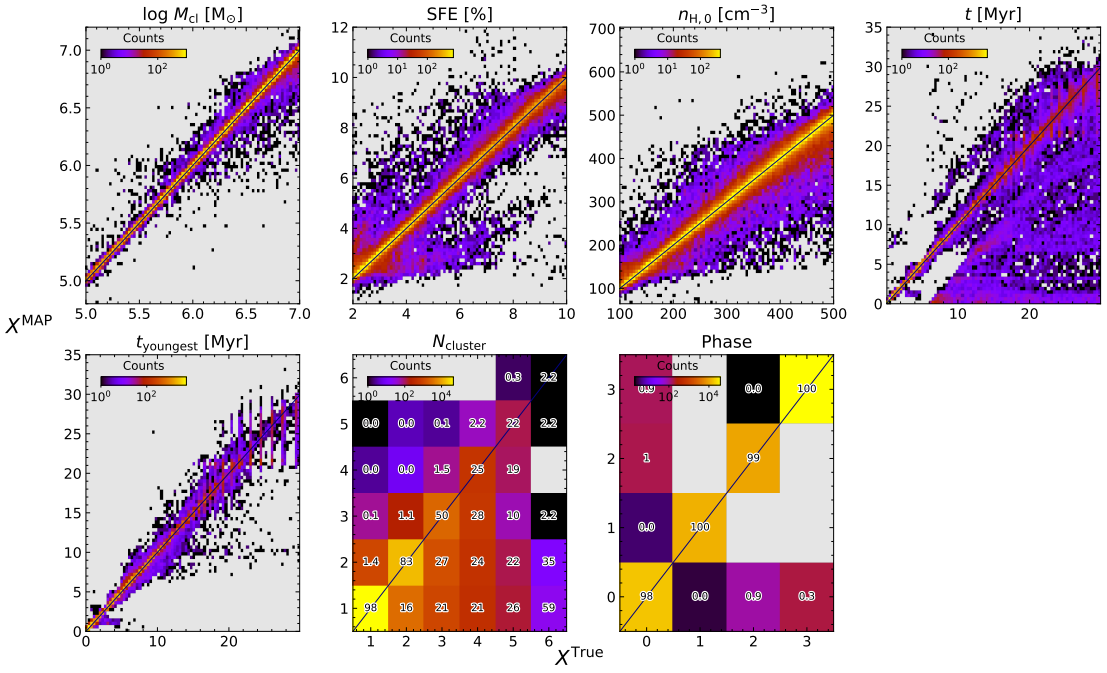

Figure 2.7: 2D histogram comparing the MAP estimates with the true values of each parameter for all 101,149 test models. The colour code is the same as in Figure 2.6 but the grey colour indicates the area without any posterior model.

its extremely narrow width compared to that of the older mode. Numerous outliers below the one-to-one line in Figure 2.7 originate from this feature.

The reason why our network predicts the cloud age more precisely for younger H II regions is related to the training data. For both cloud age and the youngest cluster age, the fraction of young models in the database is larger as shown in Figure 2.1 so that the network may have learned better about younger models. Moreover, as explained in Section 2.2.1, cloud age and the youngest cluster age are not sampled in a constant interval, unlike the other five parameters. Age intervals are sometimes wider in the old period of the cloud because the cloud does not evolve as dramatically as in the younger period.

In the case of $N_{\rm cluster}$, the more star clusters (more precisely stellar generations) contribute to the observed luminosity, the harder it is for the cINN to predict the correct number of clusters (generations). When this number is one or two, the network predicts the posterior accurately close to the true value but it shows poor performance when the number is larger than three. The poor performance for models with many clusters is also attributed to the distribution of $N_{\rm cluster}$ in the training data. Our network is trained mostly on the single cluster H II regions which occupy more than 70% of the training set as shown in Figure 2.1. Apart from the bias of the training data, it is difficult to accurately predict the number of clusters because of the intrinsic physical degeneracy of the H II region models with respect to the twelve optical emission lines that we use to predict the parameters. The strengths of these lines are dominated mostly by the ionising luminosity of the youngest stellar generation of the H II region model. So adding one or more old clusters with low ionising power does not have much impact on the overall strength of most of the lines, making it hard for our network to identify the accurate number of clusters.

In the star formation efficiency case, we discover an arrow-shaped structure pointing towards the lower left side in both Figures 2.6 and 2.7. These are similar to a one-to-two line and two-to-one line respectively, which implies the degeneracy revealed in the posterior distribution. As most of the degeneracy occurring in our network is because of the degenerate $N_{\rm cluster}$ prediction, this feature is more distinct in the range of low star formation efficiency. As shown in the $N_{\rm cluster}$ predictions, the degeneracy occurs more frequently for H II regions with many clusters. These H II region models usually have small star formation efficiencies, which makes star-forming clouds easily collapse again as stellar feedback is too weak to destroy the cloud.

Prediction of the cloud initial density and the cloud mass is well constrained to the true values. The scatter in the cloud mass distribution is larger for higher cloud masses. We consider that this feature is also related to the degeneracy of $N_{\rm cluster}$ because the massive clouds are more likely to recollapse. Phase prediction is accurate and less degenerate compared to the other parameters. We notice that even when the $N_{\rm cluster}$ prediction is degenerate and the other parameters have multi-modal posterior distributions, the posteriors of the phase are usually accurate and have unimodal distributions, as shown in Figure 2.3.

2.5 Validation of the network

In Section 2.4.2, we validated our network prediction for three example models based on the re-simulation method described in Section 2.3.3. Now we assess the general performance of our tool with more test models by comparing the luminosity of all twelve emission lines used in the network with the true values. To complete the validation of our tool, we have to check two things. First, we need to confirm whether the network correctly learned the physical rules hidden in the training data. Second, we need to examine how well the database used for training reproduces real nature. In this paper, we focus on the first validation step, evaluating the machine learning aspects of the problem, since the underlying physical input models from WARPFIELD-EMP have already been probed by previous studies (e.g., Rahner et al. 2018; Rugel et al. 2019; Pellegrini et al. 2020). Nevertheless, we will address some of the limitations of the WARPFIELD-EMP synthetic model and our training data in Section 2.8.1.

Altogether, we select 100 models from the test set for the network validation and draw 100 samples for the posterior of each model. We randomly select these 100 models, subject to the following criteria. First, we only use models with [N II]/H$\alpha \geq$ -3 and [O III]/H$\beta \geq$ -2 because the line ratios of most of the observed star-forming galaxies (Kauffmann et al. 2003; Kewley et al. 2006) or H II regions (Sánchez et al. 2015) are larger than these minimum values. Second, four of the 100 models are randomly selected among extreme cases, which are located in the BPT diagram beyond the revised demarcation curve between starburst galaxies and active galactic nuclei (AGNs) of Kauffmann et al. (2003). One of our four extreme cases is located even beyond the demarcation curve of Kewley et al. (2001). Third, for the other 96 models, we set the fractions of models with relatively uncommon characteristics in order to prevent too many models from being selected close to the mean of the distribution. Considering the parameter distribution of our database (Figure 2.1), we limited the fraction of single cluster models to 60% and the fraction of Phase 3 models to 40%. The former is to ensure a sufficient number of multicluster models in the validation-test sample and the latter is to prevent too many Phase 3 models from being selected.

We present the location of the 100 selected models in the BPT diagram in Figure 2.8. As shown in the figure, test models are selected from both high-density and low-density regions with respect to the BPT locations of all held-out models. In Figure 2.9, grey histograms represent the distribution of parameters for the selected test sample, whereas the red dashed lines show the distributions of our entire database, which are the same as in Figure 2.1. As we set selection criteria, some parameters have slightly different distributions to those of the database.

As indicated above, for our 100 test models, we draw 100 posterior samples per model, giving us a total of 10,000 sets of parameters ($M_{\rm cl}$, SFE, $n_{\rm H,0}$, and age). We then re-run WARPFIELD-EMP for each set of values, giving us 100 sets of emission line luminosities for each of our test

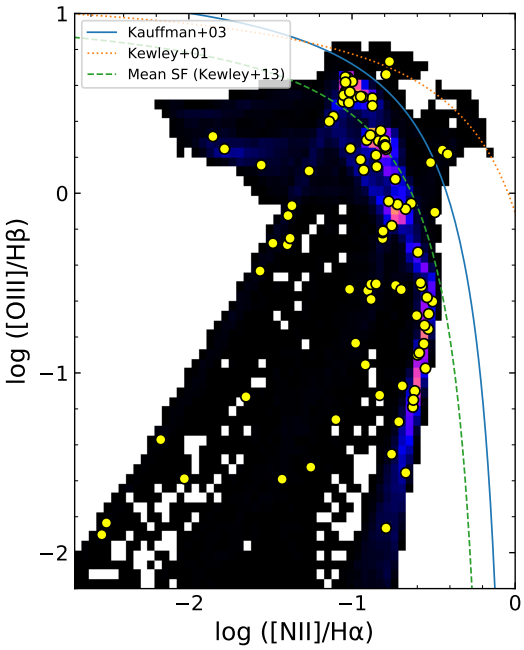

Figure 2.8: BPT diagram showing the locations of the 100 test models (yellow circles) used for the network validation in Section 2.5 and all of the models in the test set. The three demarcation lines and the colour code of the 2D histogram are the same as in Figure 2.5.

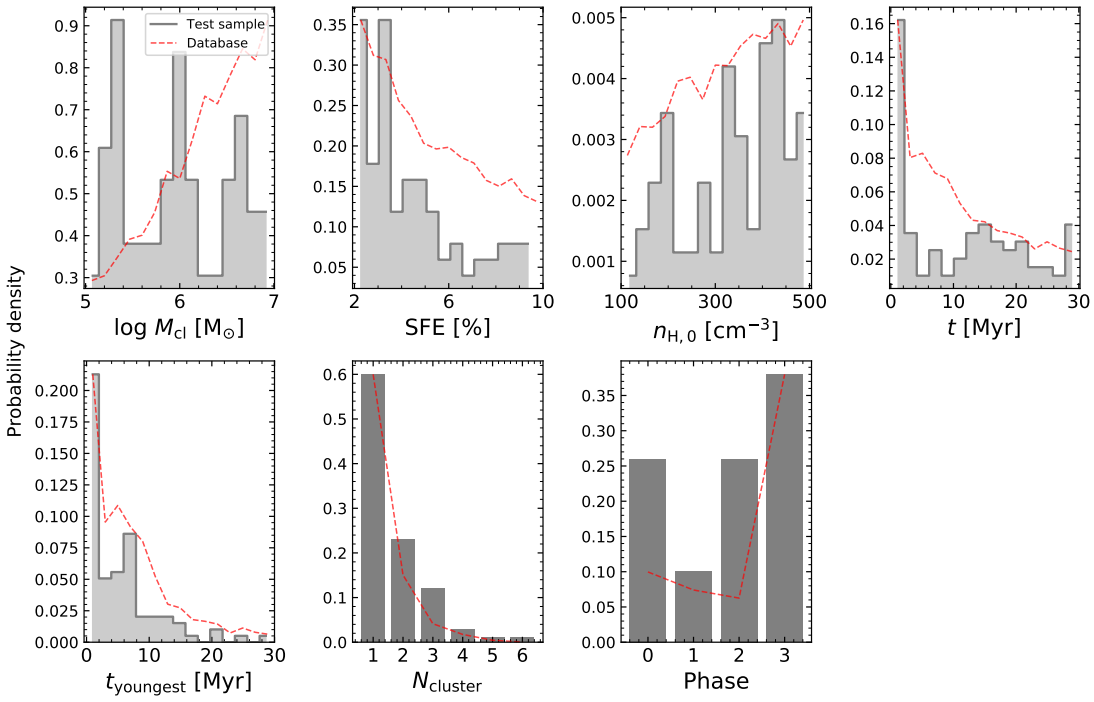

Figure 2.9: Distribution of the seven physical parameters of the 100 randomly selected test models for the network validation in Section 2.5. The grey histogram shows the distribution of the 100 models and the dashed red line shows the distribution of the whole database shown in Figure 2.1, where the amplitudes of the distributions are re-scaled.

models. For each test model, we then investigate whether the luminosities derived from the posterior samples are equal to the input-condition luminosity (i.e., the true luminosity of each model). In Figure 2.10, we present a density histogram of the logarithmic difference between the luminosities from the posterior samples and the true luminosity values. All twelve emission lines used in our network reveal similar distributions in Figure 2.10. We quantify the accuracy and precision of the luminosities derived from the posterior samples with respect to the true luminosity values by calculating the first and the second moment of each distribution. In each panel in Figure 2.10, the vertical dashed orange line and the horizontal green bar represent the first moment and twice the second moment respectively, the legend indicating their corresponding values. All 12 emission lines have very small first-moment values. The smallest absolute value is $\sim 2.65 \times 10^{-5}$ dex for the [S II] 6716Å distribution and the largest one is around 3×10^{-3} dex in [O III]. The width of the distribution is denoted by the green bar, twice the second moments, and all twelve distributions have a similar width within the range of 0.2 dex $\sim$ 0.22 dex. As shown in Figure 2.10, the posterior samples predicted by the network have emission line luminosities very close to the true luminosities of the test models, although there is a $\pm$0.11 dex error, which is equivalent to a factor of 1.3. This demonstrates that the posteriors predicted by our network are reliable and correctly conditioned on the given observation apart from the result of parameter prediction.

The performance of the network varies depending on the characteristic of the observed target that we want to analyse. For example, there is a large scatter or degeneracy in parameter predictions especially when the target H II region is very old or has many star clusters in the cloud. Since we confirmed in Figure 2.10 that the distribution of the luminosity difference is similar regardless of the emission line, we now select the Hα as the representative emission line and investigate how the luminosity distribution shifted with respect to the true luminosity differs according to the characteristics of the conditioned test models. First, we compare the case when the selected test object actually has only one star cluster (i.e., a single-cluster model) with the case when the object has two or more star clusters (i.e., a multicluster model). As mentioned above, 40 of our 100 validation test models are single-cluster models, whereas the remaining 60 are multicluster models. Figure 2.11 shows the distribution of the Hα luminosity difference between the predicted posterior samples and conditioned test models of each case. The distribution of single-cluster models (left panel) exhibits a better result considering the smaller offset and width compared to the multicluster models (right panel). The offset of multicluster models ($\sim$ 0.005 dex) is slightly larger than that of single-cluster models or that of all models combined in Figure 2.10, but it is still a very small value. Compared to the Hα distribution of entire models in Figure 2.10, the width of the single-cluster case is narrower by 0.06 dex, whereas the width of the multicluster case is wider by $\sim$0.16 dex. It is expected that our network performs better for single-cluster models because, in the previous section, we find more

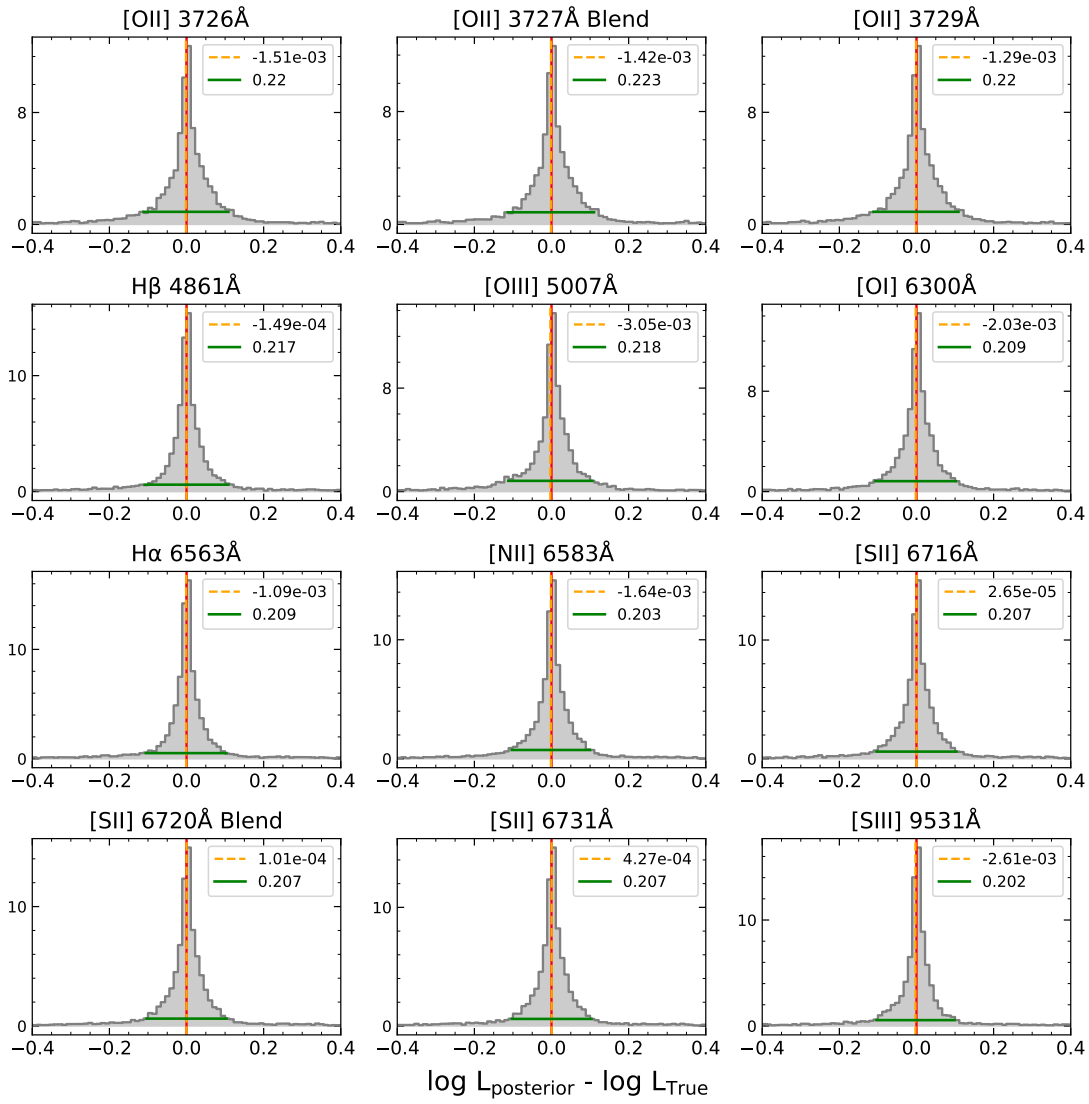

Figure 2.10: Density histograms of the logarithmic difference between the re-simulated emission-line luminosity of the posterior samples and true emission-line luminosity values of the test models. Twelve panels show the distribution of different emission lines required for our cINN model. For our 100 test models given in Figure 2.8, we draw 100 posterior samples per test model, so histograms are based on the luminosity of 10,000 posterior samples in total. We calculate the density-weighted first moment and second moment of each distribution. The vertical dashed orange line indicates the first moment and the horizontal green bar denotes twice the second moment, representing the offset and width of the distribution respectively. The corresponding values are presented by the legend in each panel.

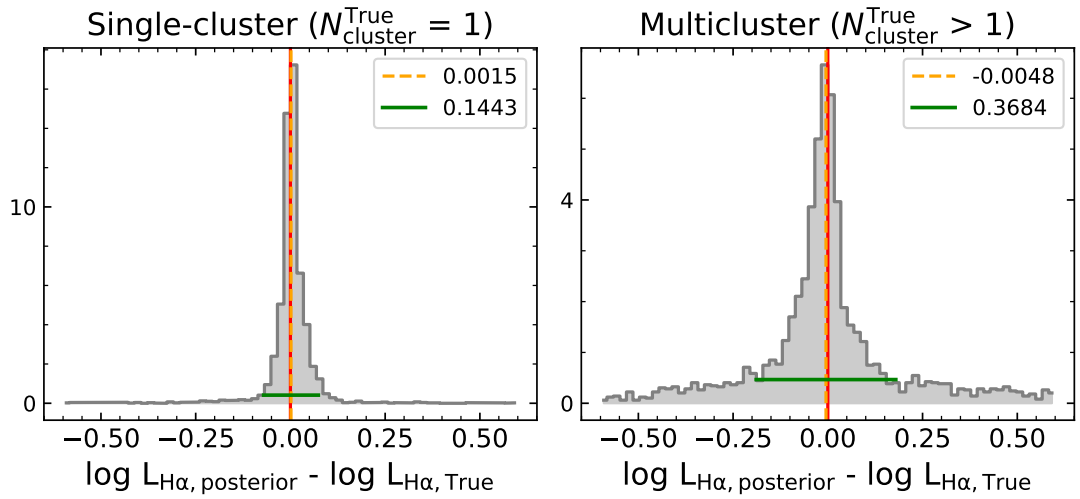

Figure 2.11: Density histograms of the logarithmic difference between the Hα re-simulated luminosity from the posterior samples and the true Hα luminosity of the test models. We divide the 100 test models into two groups depending on the true value of $N_{\rm cluster}$. The left panel shows the result of 6,000 posterior samples from the 60 test models, which have only one cluster and the right panel shows the result of 4,000 posterior samples from the 40 test models that contain more than one cluster. The colours and lines are the same as in Figure 2.10.

accurate parameter predictions for single-cluster models. Figure 2.11 reflects that even when our network shows relatively poor performance for multi-clusters, our network provides reliable predictions with the luminosity difference of -0.0048±0.1842 dex on average with respect to the conditioned observations.

Additionally, we divide the test samples according to various other features and compare the distribution of the Hα luminosity difference. For example, we split the samples according to the cloud age, where the cloud age of the first group is less than 10 Myr and that of the other group is older than 10 Myr. We confirm that the distribution of the Hα luminosity difference for the younger group has a better result than that of the older group. When the observed H II region is older than 10 Myr, the luminosity difference of the posterior samples with respect to the true luminosity value is -0.0001±0.157 dex on average. On the other hand, we divide samples into bright and dark models, where bright models have luminosities larger than 10^{34} erg/s in all twelve emission lines. This criterion serves to probe the influence of uncommon characteristics in the observation space because more than 60% of our training data is classified as bright models. We confirm that bright models have a luminosity error of 0.003±0.086 dex, while dark models have an average luminosity difference of -0.01±0.183. In conclusion, the performance of our network varies depending on the characteristics of the target we want to analyse. However, even in the poorly performing cases, the re-simulated emission line luminosity of the posterior samples predicted by our network is very similar to the conditioned luminosity with an average error of less than ±0.2 dex.

2.6 Degenerate Prediction

The shapes of the 1D posterior distributions as shown in Figure 2.3 mainly have the following three characteristics. First, a unimodal distribution is common but some posterior distributions have two or more distinguishable modes. Second, for a given observation, the number of modes in the posterior distribution is different for each parameter. Third, as mentioned in Section 2.4.2, the multi-modality in the posterior distribution of $M_{\rm cl}$, star formation efficiency, $n_{\rm H,0}$, cloud age, or the age of the youngest cluster is often due to the degeneracy in the $N_{\rm cluster}$ prediction. In this section, we perform a statistical analysis of the modality of the posterior distribution. Rather than interpreting physical meanings of the degeneracy of the network prediction, we focus on the shape of the 1D posterior distributions from a statistical point of view, such as how many modes are in the posterior distribution in general, and how the number of modes differs depending on the parameters.

2.6.1 Method of counting the number of modes

For discretized parameters such as $N_{\rm cluster}$ and phase, we simply measure the number of modes in the posterior distribution by rounding the posteriors and count the number of unique values. For non-discretized parameters, we analyse the number of modes in the posterior distribution in 3 steps. First, we bin the posteriors of each parameter to make a probability distribution. Next, we fit the posterior distribution with multiple Gaussians and in the last step, we measure the number of modes by counting the number of peaks in the posterior distribution according to three different definitions of peaks. The reasoning behind the binning method in the first step is that different binning can affect the fitting result.

We sample the posterior 2048 times per model in the test set and divide the predicted posteriors into bins with regular intervals to obtain a posterior probability distribution for the non-discretized parameters. We first bin with a number equal to the square root of the number of posterior samples. If it fails to fit the posterior distribution, we bin again according to the Freedman-Diaconis rule (i.e., FD rule: Freedman & Diaconis 1981) and retry the fitting. The FD rule takes into account the interquartile range (IQR) of the data (x) to determine the binning width (h):

$$\begin{aligned} h &= 2\frac{\mathrm{IQR}(x)}{\sqrt[3]{n}}, \\ n_{\rm bin} &= \frac{\max(x) - \min(x)}{h}, \end{aligned} \tag{2.6}$$

where n is the number of data.

We start to fit the posterior distribution with one Gaussian model and increase the number of Gaussian components if the integral of the root mean square of the residual is larger than 3%

of the full integral of the posterior distribution. The maximum number of Gaussian components per posterior distribution is limited to six. For each model, we fit the posterior distribution for the five parameters ($M_{\rm cl}$, SFE, $n_{\rm H,0}$, t, and $t_{\rm youngest}$) independently.

In order to determine the number of modes in one posterior distribution, we employ three different definitions for the peak of the mode and count the number of peaks according to each definition. First, we treat the number of Gaussian components fitted in the posterior distribution as the number of fitted peaks. In the second definition, we determine the number of visible peaks, which are defined as the local maxima of the fitted function. We use the fitted curve and the first derivative of it to find visible peaks. For the third one, we compute the number of separated peaks, where "separated" means that the distance between the centres of two Gaussian components is larger than the L2 norm of the two dispersions. If the distance between the centres is smaller than the L2 norm, we regard the pair of Gaussian components as part of one extended peak. In the previous two definitions, the position of the peak is accurately identified on the fitted posterior distribution, whereas the extended peak does not indicate a certain position but provides the range of possible peak positions.

2.6.2 Visible modes in posterior distributions

For the entire 101,149 test models, we count the number of peaks in the posterior distribution following the above steps and according to the three different definitions. As mentioned, we simply count the number of peaks for $N_{\rm cluster}$ and phase, so the number of peaks for these two parameters does not depend on the definition of the peak. In this section, we only discuss the number of visible peaks (i.e., $n_{\rm vpeak}$), which is the most conservative and straightforward among the three definitions. We provide an investigation of the other two approaches in Appendix B.2. The blue histogram in Figure 2.12 represents the distribution of the number of visible peaks in the posterior distributions. In the previous section (Section 2.4.2), we have mentioned that degenerate predictions (i.e., multi-modality in the posterior distribution) in the non-discretized five parameters are attributed to the degeneracy in the $N_{\rm cluster}$ prediction. So, we divide the test set into two groups: one with a unimodal $N_{\rm cluster}$ posterior distribution (orange histogram in Figure 2.12) and the other one with multimodal $N_{\rm cluster}$ posterior distribution (green histogram). Please note that the criterion for dividing models into two groups is not based on the true $N_{\rm cluster}$ value or the posterior value of $N_{\rm cluster}$, but on the number of modes in the posterior distribution of $N_{\rm cluster}$.

From Figure 2.12, we find that the distribution of $n_{\rm vpeak}$ differs for each parameter. However, in all parameters, about 95% of models have one visible peak if the posterior distribution of $N_{\rm cluster}$ is not degenerate. This shows that the multi-modality in the posterior distribution of $N_{\rm cluster}$ obviously affects the number of visible peaks in the posterior distribution of other parameters. However, the remaining 5% of the models whose posterior distribution is non-

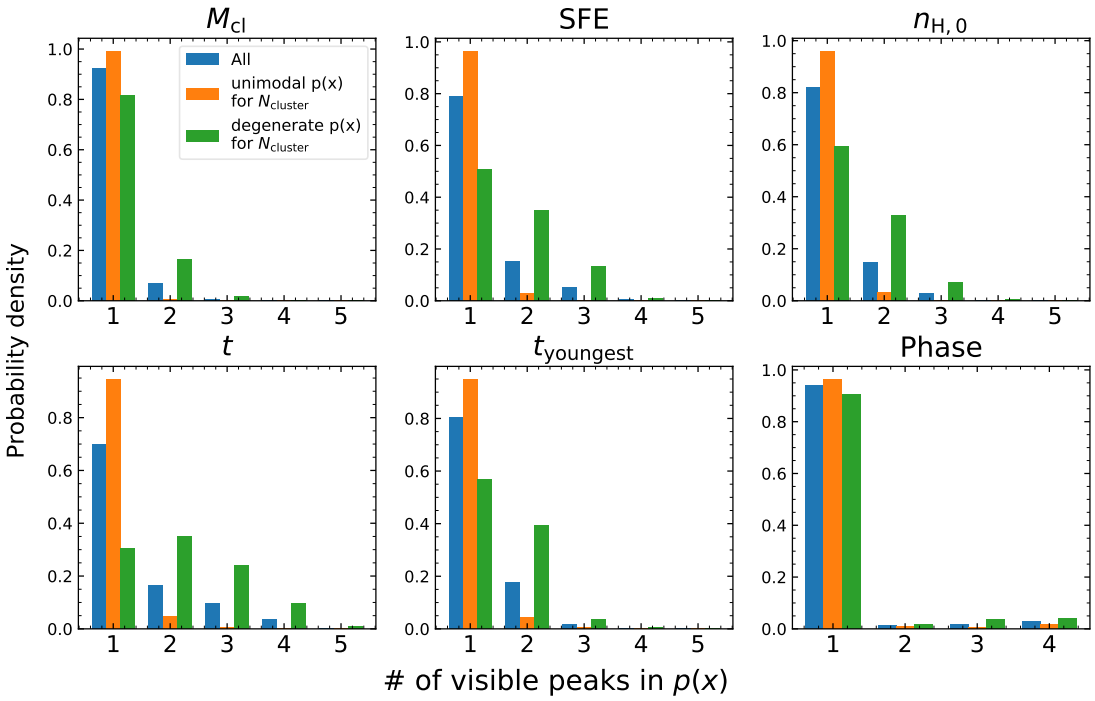

Figure 2.12: Density histograms of the number of visible peaks in the posterior distributions for six parameters. The blue histogram shows the results for all models in the test set. The orange histogram represents models whose posterior distribution of $N_{\rm cluster}$ are unimodal (61.4% of test set), whereas the green histogram represents the rest of models with multimodal posterior distribution in $N_{\rm cluster}$.

degenerate for $N_{\rm cluster}$ but degenerate for other parameters implies that the number of clusters is not the only cause of degeneracy. As expected from the results in the previous section, the posterior distribution of the phase is unimodal in most cases. The fraction of models with only one visible peak slightly decreases if the $N_{\rm cluster}$ posterior distribution is degenerate but it is still larger than 90%. We confirm that degeneracy rarely occurs in phase posteriors so we focus more on the result of the other five parameters.

Among the five parameters, the fraction of models with one visible peak is the highest in $M_{\rm cl}$ and the lowest in age. First, in the case of $M_{\rm cl}$, more than 90% of all models have only one visible peak. When the posterior distribution of $N_{\rm cluster}$ is degenerate, about 20% of models have two or more visible peaks, which is smaller compared to those of the other parameters. In the case of age, about 70% of the models have unimodal posterior distributions. If the $N_{\rm cluster}$ posterior distribution is not degenerate, more than 90% of models have one visible peak. On the other hand, if the $N_{\rm cluster}$ prediction is degenerate, 70% of models have two or more visible peaks, which implies that multi-modality of the age posterior distribution based on the visible peak is significantly influenced by the degeneracy in the $N_{\rm cluster}$ posterior distribution.

For all parameters, it is more probable to have multiple visible peaks in the posterior distribution if the posterior distribution of $N_{\rm cluster}$ is degenerate. Even when the $N_{\rm cluster}$ prediction is degenerate, the number of visible peaks is two in most cases. However, in the case of age,

half of the models with multiple visible peaks have three or more peaks. This means that, when the $N_{\rm cluster}$ prediction is degenerate, the age posterior distributions have more distinct and not blended multi-modes compared to other parameters.

In conclusion, the $N_{\rm cluster}$ prediction is not the only source of degeneracy in the posterior but it is the most influential one. As mentioned in the previous section (Section 2.4.3), it is difficult to break out the degeneracy remaining in the $N_{\rm cluster}$ posterior because the information on the number of clusters is not well reflected in the selected 12 emission lines used for the input. The amounts of emissions of these lines are contributed mostly by the youngest cluster with bright O- and B-types stars and they do not depend much on the other old stellar generations. These results suggest that we need an additional observable which is sensitive to the number of clusters such as the high-resolution photometry observations obtained with the Hubble Space Telescope (HST) to enhance the performance of our network by breaking out the degeneracy aroused from $N_{\rm cluster}$ prediction.

2.7 Posterior distributions considering luminosity errors

So far, we presented posteriors conditioned on the luminosity of 12 emission lines, but we did not take into account the fact that in real observations, these luminosities cannot be measured with arbitrarily high precision. Here, we explain how to obtain the posterior distribution by taking into account the luminosity error, and how the posterior distribution changes depending on the amount of the error of observations.

In this study, we use a Monte Carlo approach to sample posteriors while accounting for luminosity errors arising from different signal-to-noise levels of the observations. Suppose that one set of luminosities ($\mathbf{y}$) has a 1σ error in per cent unit for each of the 12 emission lines we consider (σ_1, ..., σ_{12}). The first step is to generate a sufficient number $N_{\rm MC}$ of mock luminosity sets ($\mathbf{y}'_1, ..., \mathbf{y}'_{N_{\rm MC}}$) by adding a small amount of random noise to each emission line luminosity considering the corresponding error. The second step is to sample the posterior $N_{\rm cINN}$ times conditioned on each mock luminosity set so that we draw $N_{\rm MC} \times N_{\rm cINN}$ posterior samples in total. We found that producing a sufficient number of mock luminosity sets ($N_{\rm MC}$) at least around 3000$\sim$5000 is important to obtain a smooth posterior distribution. In this study, we decided to sample the posterior 300,000 times in total for one observation by making 3000 mock luminosity sets and sampling the posterior 100 times per mock luminosity set. However, our network sometimes returns extremely extrapolated posterior samples that are physically incorrect (e.g., negative age or negative star formation efficiency) especially when the luminosity errors are large, and so we exclude all physically unrealistic posterior samples.

As we use synthetic H II region models, which by definition are fully accurate, we add a simple noise model to make mock errors of the emission line luminosity. In real observations, we have

to consider errors from diverse sources such as the Poisson error, calibration error, or systematic errors, e.g., arising from incomplete sky subtraction. However, in this study, we ignore error sources other than Poisson noise. We also ignore any covariances between different emission lines including the physical relations between blended lines and their individual components. When the 1σ per cent error of the brightest emission line of one observation is given, the errors of the other 11 emission lines are automatically determined according to

$$\sigma_{\rm line} = \sigma_{\rm min} \times \sqrt{\frac{L_{\rm brightest}}{L_{\rm line}}}, \tag{2.7}$$

where $\sigma_{\rm min}$ is the error of the brightest emission line, which is the smallest among the luminosity errors of the 12 emission lines. In real observations, this is typically the Hα line.

2.7.1 Statistical analysis

We expect the posterior distribution to change depending on the characteristics of the observed target as well as the magnitude of the luminosity error. So, we utilise the same test sample of 100 models used for network validation in Section 2.5. For each model, we obtain 16 posterior distributions for each parameter by varying the luminosity error of the brightest emission line in a logarithmic scale from 0.01% to 10% with 0.2 dex intervals. As mentioned above, we draw 300,000 posterior samples to construct the full posterior distribution for each of these 1600 cases.

To evaluate the accuracy and precision of the posterior, we use the MAP estimate and uncertainty at a 68% confidence interval (u_{68}) in the same way as we did to evaluate the posterior without considering luminosity errors (i.e., $p_0(x)$). In this section, we take the logarithm of the ratio between the posterior estimate and the true value (i.e., log deviation, $\log\frac{X}{X^*}$) as a proxy of the network accuracy. We define accuracy in two ways either by using only the MAP estimate as a representative value of the posterior distribution (i.e., MAP accuracy), or by using all of the posterior estimates. However, in the case of the $N_{\rm cluster}$ and phase, we use a linear deviation (i.e., $X - X^*$) instead. The precision of the posterior distribution is represented by u_{68}.

Using the 1600 posterior distributions, we first examine how the accuracy of the posterior depends on the luminosity error. The violin plots in Figure 2.13 show how the accuracy distribution changes with increasing minimum luminosity error. We present the accuracy using only MAP estimates in the first row of Figure 2.13 and accuracy using all posterior estimates in the second row. Blue histograms in the panel show the accuracy of the posterior distribution for 100 models with the same minimum luminosity error. Please note that we apply the same luminosity error only for the brightest emission line so each model has different luminosity errors for the other emission lines. The yellow histogram located at the bottom of each panel represents the accuracy of posterior distributions without luminosity errors ($p_0(x)$). In order to maintain the same sampling size, we draw 300,000 posterior samples per test model to make a posterior

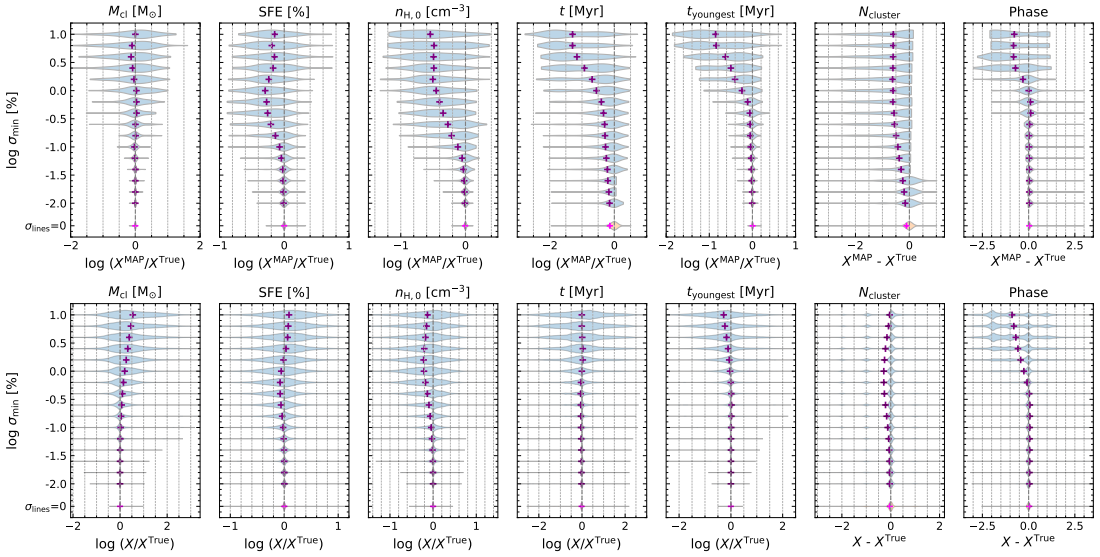

Figure 2.13: Using 1600 posterior distributions (100 test models and 16 different luminosity errors of the brightest emission line for each model), we present histograms of the logarithmic difference between MAP estimates and true values of each parameter (blue histograms in the first row). Please note that for $N_{\rm cluster}$ and phase, we use the difference in linear scale. The yellow histogram at the bottom of each panel is the distribution obtained from the posterior without luminosity error. The cross symbol in each histogram represents the average of the distribution. In the same manner, the lower panels show distributions of the logarithmic difference between all posterior estimates and true values instead of MAP estimates.

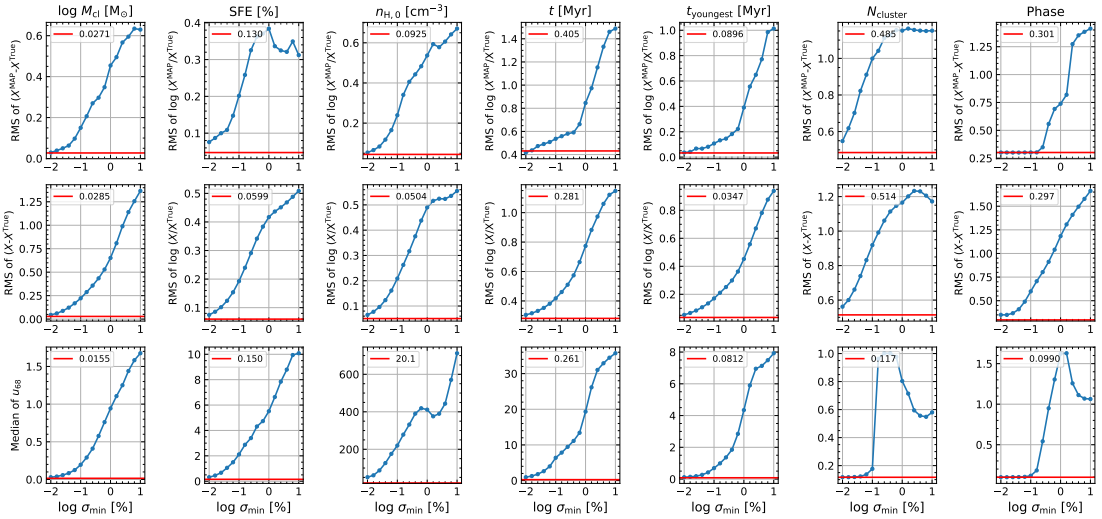

Figure 2.14: Accuracy and precision of our network as a function of the luminosity error of the brightest emission line ($\sigma_{\rm min}$) using 100 test models. Please note that the luminosity error is given in per cent. We use the RMS of MAP accuracy in the first row and the RMS of accuracy using all posterior estimates in the second row. The last row shows the median uncertainty at 68% confidence interval (u_{68}) which represents the precision of our network. The red horizontal lines and corresponding values on the upper left corner in each panel indicate the value obtained from posterior distributions without luminosity error ($\sigma_{\rm lines} = 0$).

distribution without luminosity errors. The average of each distribution in the figure is denoted by a cross symbol.

From Figure 2.13, we find that the accuracy distribution using all posterior estimates and that using only MAP estimates are significantly different. First, in the upper panels, the MAP accuracy distributions are on average shifted to negative values, except for $M_{\rm cl}$, and the magnitude of the shift increases with increasing luminosity error. This feature is especially evident in the distribution of $n_{\rm H,0}$, cloud age, and the youngest cluster age. The star formation efficiency exhibits a similar trend, but when the luminosity error ($\sigma_{\rm min}$) is within a range of 1$\sim$10% the distributions become less biased in comparison to those with smaller luminosity error. On the other hand, when using all posterior estimates, the accuracy distributions show different trends. In the case of $M_{\rm cl}$, the distribution gradually shifts to the positive direction on average as the luminosity error increases ($\sim$0.1 dex). The distributions of star formation efficiency, $n_{\rm H,0}$, cloud age, and the youngest cluster age become wider but do not shift to one side even when the luminosity error is large. The density distribution shows a negative shift around 0.2 dex but this is smaller than the amount of shift found in the MAP accuracy distributions. This implies that the posterior distributions of these four parameters are skewed towards the range below the true values, but have a long tail-like distribution with a lower probability in the range above the true values so that the distributions are not shifted on average. In particular, it is expected that density, cloud age, and the youngest cluster age will clearly show this characteristic when the minimum luminosity error is 1% or more. In the case of $M_{\rm cl}$ showing the opposite trend, it is expected that a large number of posterior estimates have larger values than the true one from the positive shifts in accuracy distributions. From the distribution of MAP accuracy, we infer that the posteriors of $M_{\rm cl}$ have multi-modal or spiky distributions so that the peak of the distribution does not have a systematic bias to a certain direction on average.

To measure the average accuracy as a function of the minimum luminosity error, we calculate the root mean square (RMS) value of the accuracy shown in Figure 2.13. The first and second rows in Figure 2.14 present the RMS of the MAP accuracy and the RMS of the accuracy using all posteriors estimates, respectively. Additionally, in the third row, we examine the median of the u_{68} values from 100 models as a function of the minimum luminosity error to quantify the average width of the posterior distribution. The red horizontal line in each panel represents the values from the posterior without luminosity errors ($p_0(x)$). Figure 2.14 quantitatively shows how much our network performance decreases with increasing luminosity error. In the case of $M_{\rm cl}$, star formation efficiency, and $n_{\rm H,0}$, the RMS of MAP accuracy gradually increases up to 0.4$\sim$0.6 dex. The RMS value of the star formation efficiency slightly decreases when the minimum luminosity error is larger than 1%, which is also confirmed in Figure 2.13. In the case of cloud age and the youngest cluster age, the slope of the RMS curve becomes larger around a luminosity error of 1%. The RMS value of the youngest cluster age is smaller than other parameters when

the minimum luminosity error is smaller than 1%, but thereafter the RMS value rapidly increases to 1 dex. This indicates that the network has more difficulty in finding accurate MAP estimates of the age and age of the youngest cluster than other parameters when the minimum luminosity error is larger than 1%. In the case of the number of clusters, the RMS value increases steadily up to the minimum luminosity error of 1%, and thereafter, it is almost constant. In the case of phase, on the contrary, there is no change up to the minimum luminosity error of 0.1%, but after that, the RMS value increases significantly.

There is no significant difference in the RMS curve when the whole posterior estimates are considered (second row in Figure 2.14) or when only the MAP estimates are used (first row in Figure 2.14). Although the increment is different, the RMS values of the seven parameters gradually increase with increasing luminosity error. This represents the increase in the overall width of the distribution shown in Figure 2.13. The median of u_{68} also shows the change in the width of the posterior distribution. The unit of the u_{68} in the third row of Figure 2.14 is the same as the physical unit of the corresponding parameter. Compared to the width of the posterior distribution without luminosity errors, $p_0(x)$, represented by red lines, the posterior distributions become much wider with increasing luminosity error. When the minimum luminosity error is around 10% the average width of the posterior distribution is almost the same as the entire parameter range of the training data (see Figure 2.1 to compare with the parameter ranges of the training data).

2.7.2 Change of the posterior distribution for individual models

In order to investigate the characteristics of the posterior distribution expected from Figure 2.13 and to examine the change of the posterior distribution depending on the luminosity error in detail, we select three from the 100 test models as examples. For these three models, we show the posterior distribution of each parameter for five different minimum luminosity errors (0.01, 0.1, $10^{-0.4}$, 1, and 10%, respectively) as well as the posterior distribution without luminosity error in Figures 2.15, 2.16, and 2.17 respectively.

In Figure 2.15, we show the posterior distributions of the first model which is the same model shown in the first column of Figure 2.3. When we do not apply luminosity errors (i.e., $p_0(x)$), this model has a common unimodal posterior distribution which is accurate and precise to the true value (the first column in Figure 2.3 and 2.15). The brightest emission line of this model is Hα and the faintest line is [O I] 6300Å with a factor of 12.7 larger uncertainty. The posterior distributions do not change much at a minimum luminosity error of 0.01%. From the u_{68} values of each distribution, we notice that the width of the distribution is 2-3 times wider than that of $p_0(x)$, but the width is still narrow enough. Even when the minimum luminosity error increases to 0.1%, the MAP value is close to the true value although the width of the posterior distribution considerably widens. However, if the minimum luminosity error is larger than 0.1%, the peak of

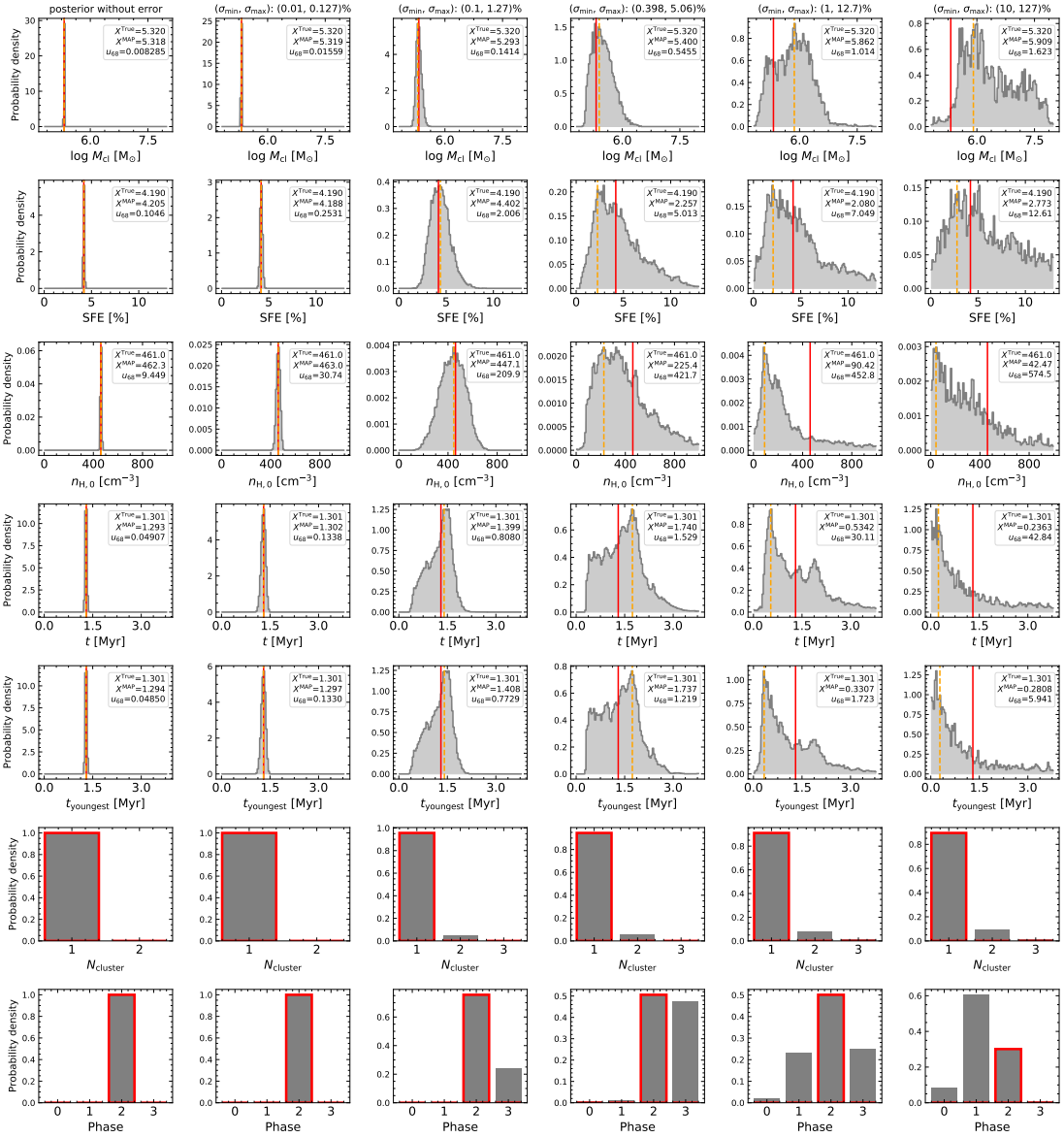

Figure 2.15: Posterior probability distributions (grey) of the first example model. This model is the same as the model in the first column of Figure 2.3. The first column shows the posterior distribution when no luminosity error is applied. From the second column to the last column, we present the posterior distribution with luminosity errors of the brightest emission-line of 0.01, 0.1, $10^{-0.4}$, 1, and 10% respectively. The luminosity errors of the brightest and faintest emission lines are indicated at the top of each column. Red vertical lines or red edges in the bar-shaped histograms indicate the true values of the model. On the upper right corner of each panel, we present true values, MAP estimates and their 1σ uncertainties, and u_{68} values of each posterior distribution.

the distribution moves further away from the true value and the width widens.

In the case of density, cloud age, and the youngest cluster age, the posterior distributions at $\sigma_{\rm min}$ of 1% and 10% show a skewed shape. As we expected from Figure 2.13, the distributions have MAP estimates smaller than the true value but exhibit a tail-like shape toward values larger than the true values. For cloud age and the youngest cluster age, the posterior distributions show blended multi-modality at a value of 0.1% $\sigma_{\rm min}$ because they begin to have a number of younger age estimates. As $\sigma_{\rm min}$ increases the level of skewness of the posterior distribution increases as well. Similarly, in the case of $N_{\rm cluster}$ and phase, the posterior distributions become more degenerate with increasing luminosity error. But, unlike $N_{\rm cluster}$, which maintains an accurate prediction even when the luminosity error increases to 10%, the accuracy of the phase deteriorates as the luminosity error increases.

The second example model is the one in the third column of Figure 2.3. For this model, the luminosity error of the faintest emission line, [O I] 6300Å, is a factor of 10.4 larger than that of the brightest line, Hα. Even without luminosity error, the posterior of this model has a bimodal distribution arising from the degeneracy in the $N_{\rm cluster}$ prediction. The two modes widen as the luminosity error increases and merge at $\sigma_{\rm min} \lesssim 1\%$. In the case of star formation efficiency and $n_{\rm H,0}$, the MAP estimate does not change significantly even when the minimum luminosity error increases by 1%. However, in the case of cloud age and the youngest cluster age, the MAP estimates at a $\sigma_{\rm min}$ of 1% or larger are significantly different from those at smaller luminosity errors. As the two degenerate modes in p_0 merge together, the MAP moves to the middle position between the two modes. But as the minimum luminosity error increases by 1%, the MAP moves to the peak of the newly formed mode located at a significantly younger age range. As the minimum luminosity error increases by 10%, the MAP is shifted again toward a much younger range. The result that the posterior distribution changes to a skewed distribution is consistent with the result of the first example and the trend revealed in Figure 2.13.

For the third example, we select a model that undergoes one recollapse phase and consequently contains two generations of stars, or in our terminology has two clusters. Before applying luminosity errors (the first column in Figure 2.17), the posterior distribution of this model has similar characteristics to the first example, which has a unimodal distribution and is very close to the true value. The differences are that the third model has two clusters and older age than the first model. The overall change of posterior distributions with increasing luminosity error is similar to that of the other models as well. Especially at a minimum luminosity error of 10%, the shape of the posterior distribution of cloud age and the youngest cluster age is similar in all three models. On the other hand, the posterior of $M_{\rm cl}$ and star formation efficiency shows a degenerate bimodal distribution at a $\sigma_{\rm min}$ of 0.1$\sim$1%.

The interesting point of this model in Figure 2.17 is the change of the $N_{\rm cluster}$ posterior distribution. From the minimum luminosity error of 0.1%, the degeneracy appears in the $N_{\rm cluster}$

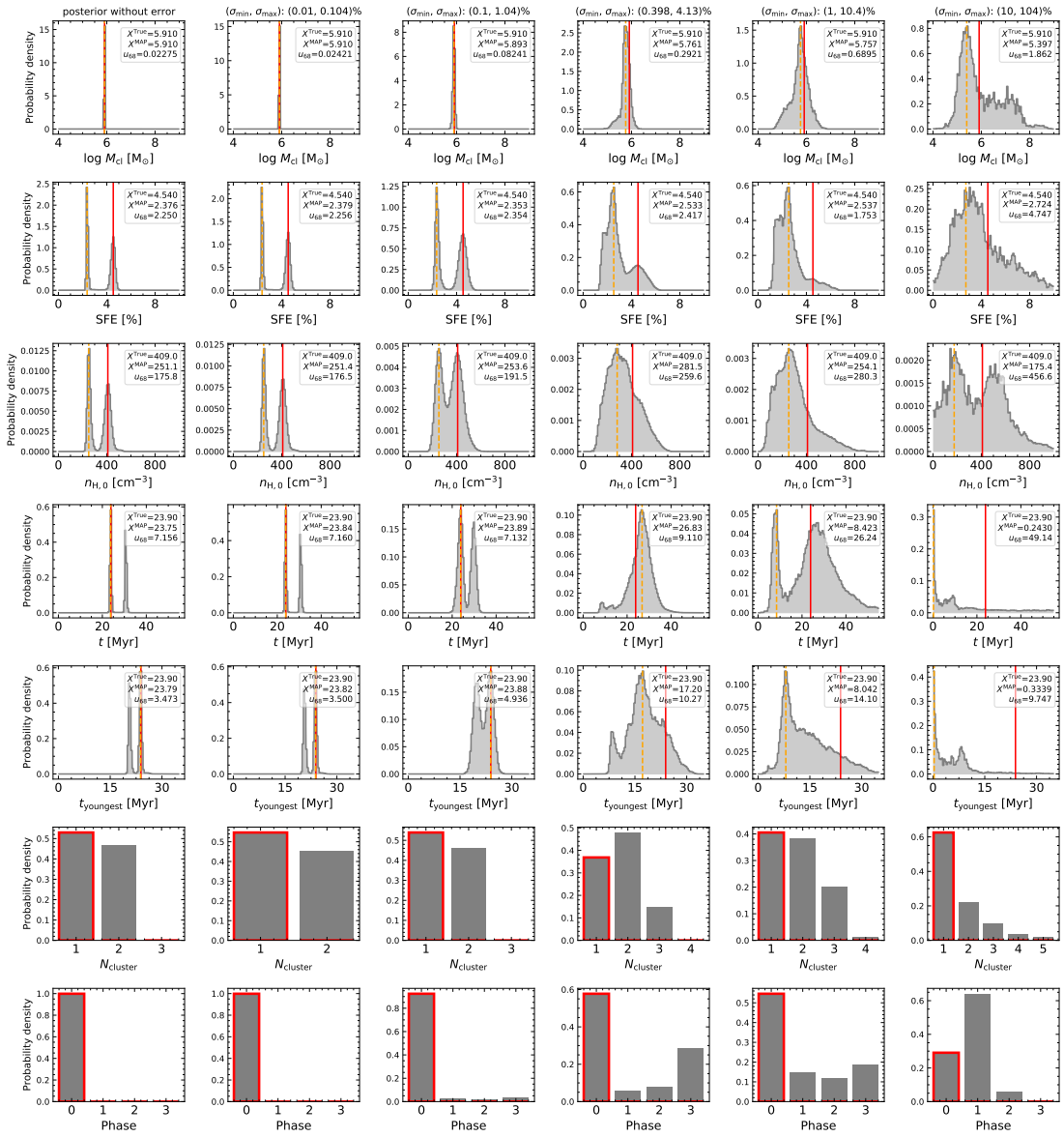

Figure 2.16: Posterior distributions of the second example model corresponding to the model in the third column of Figure 2.3. Colour codes and lines are the same as in Figure 2.15.

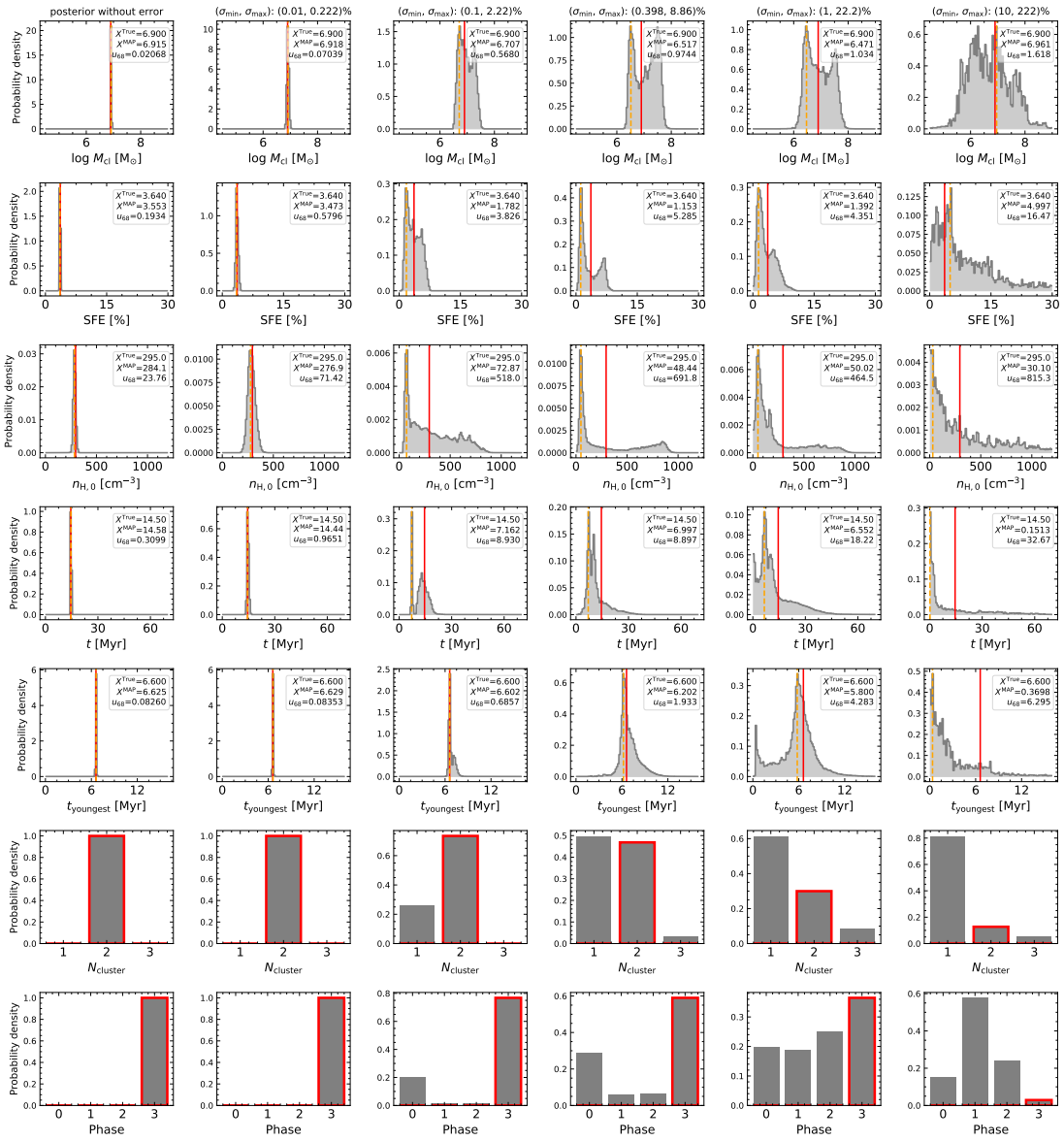

Figure 2.17: Posterior distributions of the third example model. Colour codes and lines are the same as in Figure 2.15.

prediction and the fraction of posterior samples that predict an $N_{\rm cluster}$ of 1 increases with increasing luminosity error. Finally, the posterior distribution at the largest luminosity error is almost the same as that of the other two models. However, this means that the network performed poorly compared to the other two cases because this model actually has two clusters. Thus, we examine the change of the posterior distribution of $N_{\rm cluster}$ for other test models that have two or more clusters like this model. We notice that when the minimum luminosity error is large, around 1∼10%, the posteriors samples that estimate $N_{\rm cluster}$ as 1 are dominant regardless of the intrinsic characteristics of the target. The MAP estimates of $N_{\rm cluster}$ with a minimum luminosity error larger than 1% are always 1 for all 100 test models. This means that in the previous two cases, the network seemed to predict the number of clusters reliably, but in fact, it just always provides a similar posterior distribution.

We can interpret the change of the posterior distributions of the cloud age, the youngest cluster age, and the phase in the same manner. For example, the posterior distribution of phase at a 10% minimum luminosity error is similar in all three models in that the fraction of phase 1 predictions is the highest. We check the MAP estimate of phase for 100 models and find that when the minimum luminosity error is larger than $\sim 3\%$, more than 85% of models have a phase MAP value of 1. Even for models whose true phase is not 1, still more than 85% of models have a phase MAP estimate of 1. This means that at least the peak of the phase posterior distribution is always similar regardless of the conditioned observations if the luminosity error is large enough.

We infer that the consistent posterior distribution when the luminosity error is large is influenced by the bias in our training data. As mentioned in Section 2.4.3, even when we do not consider luminosity errors, posterior distributions are frequently degenerate and inaccurate in the case of H II regions that either are old or have more than one cluster. This is because the fraction of single clusters or young H II regions is high in our training data so our network is well-trained and performs well for these kinds of models. We suppose that the bias of the training data also affects the posteriors with luminosity errors as well. Our network delivers good predictions for young and single cluster H II regions but frequently provides degenerate solutions for old, multi-cluster H II regions which include young, single cluster posterior samples. If the luminosity errors are large, mock luminosity sets are more likely to encompass the luminosity of diverse H II regions with various characteristics. Therefore the posteriors with a large luminosity error eventually have a higher proportion of younger and single-cluster posterior samples. For this reason, we consider that the skewed posterior distributions in the cloud age, the youngest cluster age, and the $N_{\rm cluster}$ are influenced by the characteristics of the training data.

On the other hand, the posterior distribution of phase at a minimum luminosity error of 10% is not similar to the phase distribution of our training data which has the highest fraction at phase 3. Phase posteriors without luminosity errors were not significantly influenced by the bias of the training data and were accurate most times regardless of the conditioned model. The

reason why the fraction of phase 1 in the posterior distribution is high when the luminosity error is large is that very young H II regions, younger than 1 Myr, are mostly in phase 1. As the fraction of posterior samples with a very young age increases when the luminosity error is large, the fraction of phase 1 predictions also increases.

To summarise, the posterior distribution gradually widens and shows a skewed shape with inaccurate MAP estimates as the luminosity error increases. However, the network guarantees the reliable MAP prediction at a $\sigma_{\rm min} \lesssim 0.1\%$ or $< 1\%$ depending on the parameters. This suggests an approximate minimum signal-to-noise level for the brightest emission line, which is typically the Hα line, to obtain reliable posterior samples when we apply our tool to the real observations. In this study, we ignore any covariance between different lines, e.g., a relation between a blended line and individual components, and produce mock luminosities as randomised as possible. We expect that our network will perform even better despite the same amount of uncertainties if we consider the covariances between lines because it reduces the randomness in mock luminosities.

2.8 Discussion

2.8.1 Major assumptions inherent in the training data

The WARPFIELD-EMP models we used as training data make several simplifying assumptions. In the previous sections, we demonstrated that the network learned the hidden rules in the training data well, but we did not validate our network using real observations beyond the comparisons already published in Pellegrini et al. (2020). As our synthetic models are not a perfect representation of nature, we should consider the main assumptions of the WARPFIELD-EMP model in interpreting the posterior distribution if we apply the network to real observations.

One of the main approximations made within WARPFIELD is the assumption of spherical symmetry. The effects of small structures such as pillars, or larger three-dimensional inhomogeneities in the cloud are not taken into account in WARPFIELD's model for the evolution of the H II region. Though it has limitations, this approximation dramatically reduces the computational cost of the calculation compared to a correspondingly detailed 3D simulation, allowing us to produce a large number of synthetic models applicable for machine learning techniques.

Another important assumption is that the clouds are isolated. In reality, some star-forming regions are close enough that their evolution may affect each other. In addition, even if nearby regions evolve independently, they may be blended together in observations that have limited spatial resolution. This is particularly an issue in optical surveys of H II regions in nearby galaxies, as even the highest resolution examples of such surveys struggle to reach spatial resolutions better than ~ 50 pc in all but the closest galaxies. Our assumption that the clouds are isolated also implies that we do not account for external effects such as the influence of the large-scale galactic potential or contamination of the observations by emission from diffuse ionised gas.

The WARPFIELD models also make use of a highly idealised treatment of star formation. All of the stars in each cluster are assumed to form instantaneously, i.e., there is no gradual star formation. Moreover, in cases where the cloud re-collapses and forms a new cluster, we assume that the star formation efficiency is the same as for the original cluster. This assumption is made purely on the grounds of simplicity and it is unclear how well it matches what happens in reality.

WARPFIELD also makes several assumptions that are commonly used in other feedback models. For example, we assume that the shell surrounding the inner bubble is in quasi-hydrostatic equilibrium. The WARPFIELD cloud is in virial equilibrium so that there is no partial gravitational collapse in the cloud. We also apply photoionisation and chemical equilibrium to model the cluster and cloud evolution. The mass of stars in the cluster follows the Kroupa initial mass function (Kroupa 2001) with an upper stellar mass limit of 120 $M_{\odot}$. As the time-dependent evolution of the cluster is calculated from STARBURST99 (Leitherer et al. 1999, 2014), we accordingly accepted the physical assumptions used in it.

In addition to the above, we also made assumptions about several parameters when constructing the WARPFIELD-EMP database, resulting in additional constraints. As shown in Figure 2.1, the range of each parameter in the training data is restricted. We confined the range of $M_{\rm cl}$, SFE, and $n_{\rm H,0}$ when we randomly sampled the initial clouds and limited the maximum age of the cloud to 30 Myr. On top of these, we assumed a constant radial profile for the initial cloud density and did not account for the effects of the magnetic field and turbulent pressure. As the network is trained on the models within the database, care must be taken when applying our network to objects whose conditions are significantly different from our database models.

2.8.2 Effect of noise augmentation on the network performance

We mentioned in Section 2.3.2 that smoothing out the discretized parameter distribution by adding a small amount of artificial noise not only helps the network converge in training, but also improves the prediction performance of the network. For this reason, we added Gaussian noise with a standard deviation of 0.05 to smooth out the distribution of $N_{\rm cluster}$ and phase to train the main network introduced so far in this paper. To assess the effect of the smoothing process on network performance, we compare the network used in this study with the network trained without smoothing the distribution of $N_{\rm cluster}$ and phase. For convenience, we refer to the former network as Network 1 and refer to the latter network as Network 2. We trained Network 2 with the exact same settings as Network 1 except for smoothing.

To compare the performance of the two networks, we evaluate Network 2 in the same way as described in Section 2.4.1. We sample posteriors for the entire test set by using Network 2 and measure MAP values for each posterior distribution. We first plot the predicted posteriors or MAP estimates against the true values in Figures A.2 and A.3. The direct comparison with the results of Network 1 (Figures 2.6 and 2.7) confirms that the prediction performance is improved by

Table 2.4: Comparison of the network performance between two networks: the main network introduced in the paper to which we applied noise augmentation on $N_{\rm cluster}$ and phase (Network 1), and the network trained without any noise augmentation (Network 2). The first value in each item shows the performance of Network 2 and the second value shows the performance of Network 1 which is the same value shown in Table 2.3.

Performance measure (Network 2 / Network 1)	log $M_{\rm cl}$ log [$M_\odot$]	SFE [%]	$n_{\rm H,0}$ [cm^{-3}]	t [Myr]	$t_{\rm youngest}$ [Myr]	$N_{\rm cluster}$	Phase
$e_{\rm cal}^{\rm med}$	4.7 / 0.44	4.9 / 0.26	6.7 / 0.87	6.2 / 1.3	7.4 / 1.1	18 / 2.3	29 / 0.12
$u_{68}^{\rm med}$ ($\hat{x}$)	0.056 / 0.028	0.32 / 0.095	0.48 / 0.17	0.58 / 0.018	0.035 / 0.0097	0.00088 / 0.15	0.00035 / 0.086
$u_{68}^{\rm med}$ (x)	0.031 / 0.015	0.74 / 0.22	54 / 19	4.9 / 0.15	0.23 / 0.064	0.00065 / 0.11	0.0004 / 0.1
RMSE ($\hat{x}$)	0.093 / 0.082	0.35 / 0.25	0.46 / 0.28	0.81 / 0.56	0.16 / 0.12	0.98 / 0.64	0.22 / 0.17
RMSE (x)	0.051 / 0.045	0.8 / 0.58	51 / 32	6.8 / 4.7	1.1 / 0.79	0.72 / 0.47	0.25 / 0.2

applying smoothing. This holds not only for the two parameters to which we applied smoothing, but also for the other five parameters. Especially for the star formation efficiency, we find that the arrow-shaped structure is more prominent in Figure A.2 than in Figure 2.6. We interpret this structure as the degeneracy revealed in the posterior distribution. The more conspicuous V-shaped structure shown in Figure A.2 reflects that degeneracy is more frequently observed in the posterior distributions of Network 2 and it is also common in the overall star formation efficiency range.

To compare the two networks more quantitatively, we evaluate the performance of Network 2 in the same way as Network 1 and provide the result in Table 2.4. The first entry in each column is the evaluation result of Network 2 and the second value after the slash is the result of Network 1, note that this last entry to the number provided in Table 2.3. This clearly shows that Network 1 with smoothing applied has significantly improved performance compared to Network 2 without smoothing. Network 2 has larger median calibration errors in all parameters. Specifically, the calibration errors of $N_{\rm cluster}$ and phase reach 18% and 29%, respectively, which are significantly larger than those of Network 1. Although the error of $N_{\rm cluster}$ is the largest in Network 1 as well, it is less than 2.5%. From the $u_{68}^{\rm med}$ representing the average width of posterior distributions, the smoothing process decreases the width of the posterior distribution for five parameters ($M_{\rm cl}$, SFE, $n_{\rm H,0}$, cloud age, and the youngest cluster age). The posterior distributions of $N_{\rm cluster}$ and phase widen by smoothing but this is an expected result. By changing the quantised values to a continuous, wider distribution in training, the network learns to avoid narrow and delta-function-like distributions and chooses wider but less degenerate posterior distributions. In addition, the increased width is still sufficiently small compared to the sampling interval of 1. Last, the RMSE values also demonstrate that the accuracy of Network 2 is inferior to that of Network 1.

These results prove that augmenting discontinuous parameter distributions with artificial noise can improve the overall performance of the network. By smoothing out the distribution of $N_{\rm cluster}$ and phase, the accuracy of $N_{\rm cluster}$ is improved, reducing the frequency of degeneracy in the predicted posterior distributions.

2.9 Summary

In this paper, we introduce the novel method of applying a conditional invertible neural network (cINN) to predict the fundamental physical parameters of H II regions from spectral observations. When solving the inverse problem to infer the underlying physical parameters ($\mathbf{x}$) from observational data ($\mathbf{y}$), intrinsic degeneracies make the solution ambiguous. During the forward process, which translates the parameters into observations, inevitable information loss occurs so that different physical systems are mapped onto identical observations. By introducing the latent variables ($\mathbf{z}$) which capture the information loss, a cINN learns the bijective forward mapping between $\mathbf{x}$ and $\mathbf{z}$ conditioned on $\mathbf{y}$, $\mathbf{z} = f(\mathbf{x}; \mathbf{y})$. The invertibility of the cINN architecture automatically provides the inverse mapping $\mathbf{x} = f^{-1}(\mathbf{z}; \mathbf{y})$. Once the network is trained through the forward process, we can produce the posterior distributions of $\mathbf{x}$ conditioned on $\mathbf{y}$ ($p(\mathbf{x}|\mathbf{y})$) by sampling the latent variables.

As it is difficult to collect the enormous amount of data required for network training, we used a database of synthetic H II region models produced by the WARPFIELD-EMP pipeline (Pellegrini et al. 2020). WARPFIELD-EMP evolves an isolated massive star-forming cloud using the 1D stellar feedback modelling code WARPFIELD (Rahner et al. 2017) and calculates several observable quantities for the evolving H II region, such as the luminosities of various emission lines, by processing the WARPFIELD output with CLOUDY (Ferland et al. 2017) and POLARIS (Reissl et al. 2016). The first WARPFIELD-EMP database introduced in Pellegrini et al. (2020) successfully mimicked the BPT diagram of H II regions observed in NGC628 (Rousseau-Nepton et al. 2018) but the number of models in that database and the sampling interval of each parameter was not enough to train a network. In this paper, we introduced a new, extended database that consists of 505,748 H II region models evolved from 10,000 randomly sampled initial clouds and used this new database to train and evaluate the network.

The network introduced in this paper is originally designed for the SDSS-V LVM survey, but it can easily be adapted to data from other instruments or telescopes. Using the luminosity of 12 optical emission lines (Table 2.2) observable within the wavelength coverage of LVM, our network predicts seven physical parameters of the H II region: initial mass of the star-forming cloud $M_{\rm cl}$, star formation efficiency, initial cloud density $n_{\rm H,0}$, age of the cloud t which means the age of the first generation stars, age of the youngest cluster (i.e., age of the youngest generation of stars in the system) $t_{\rm youngest}$, number of clusters (i.e., distinct stellar populations) $N_{\rm cluster}$, and evolutionary phase of the cloud (see also Table 2.1). We trained the network using 80% of the database and used the remaining 20% to evaluate the trained network with various methods. We validated the network performance with WARPFIELD-EMP synthetic models, focusing on the learning ability of the cINN architecture. An application of our newly developed tool to the analysis of real observational data of H II regions from various large-scale surveys will be

presented in follow-up studies. Our main results of testing the network performance are the following:

1. The trained network is able to predict the posterior distribution very fast and efficiently. On average, the cINN can predict posterior distributions for 170 observations per second, sampling each posterior 4096 times with an NVIDIA GeForce RTX 2080 Ti graphic card. With the same graphic card, training of the network takes only a few hours (2–6 hours in most cases) depending on the adopted hyperparameters of the network configuration.

2. Our network predicts each physical parameter very accurately and precisely. The posteriors commonly show a clear, unimodal distribution with narrow width around the true value. We evaluated the overall performance using three different methods (median calibration error, uncertainty at 68% confidence interval, and the RMSE between the MAP estimate and the true value) and confirmed that our network typically predicts the physical parameters of H II regions very well and accurately. The most difficult properties to determine precisely are the number of clusters and the age of the cloud.

3. In some cases, the posteriors are degenerate, showing a multimodal posterior distribution. This degeneracy worsens the performance in terms of parameter prediction. However, degenerate posteriors are not incorrect or wrong, instead, they are physically valid alternatives that satisfy the same observational constraints. The network understands the hidden rules in the training data and suggests physically reasonable possibilities. Distinguishing between these possibilities may require additional data currently not considered in our network (e.g. broad-band optical or infrared luminosities).

4. The main source of degeneracy that occurs in our network is caused by multiple star clusters (stellar generations) in a cloud leading to similar emission properties. If the posteriors of $N_{\rm cluster}$ are degenerate and exhibit multiple peaks, the other parameters are prone to have degenerate posterior distributions as well. In particular, the cloud age is highly sensitive to variations in $N_{\rm cluster}$. We confirm that more than 90% of the test models have non-degenerate posterior distributions for other parameters if the $N_{\rm cluster}$ prediction is not degenerate.

5. The performance of the network varies with the characteristics of the observed target. As mentioned above, for clusters with multiple stellar populations ($N_{\rm cluster}> 1$) it is more difficult to get the right value of $N_{\rm cluster}$ and the correct cloud age t. Also, our network performs better for young H II regions than for old H II regions. This is because multicluster or old H II regions are more likely to have degenerate posterior distributions. Even if the posterior provides an accurate estimate of the cloud age, the posterior distribution is wider if the estimated age is older. When the posterior distribution of the cloud age is multimodal,

the younger mode is usually narrower than the older mode. So the MAP estimate is frequently dominated by the peak of the younger mode even if the fraction of posteriors in the older mode is higher. The poorer performance for multicluster H II regions or old H II regions can in part be attributed to the biased parameter distributions in our training data.

6. We validated the network performance by comparing the luminosity of the predicted posteriors with the corresponding true luminosity. To get this luminosity information, we selected 100 test models, sampled the posterior 100 times per model, and re-run WARPFIELD-EMP for the corresponding 10,000 posterior samples. This final test confirmed that the re-simulated luminosity of the posterior samples is close to the true luminosity with an average offset less than 3×10^{-3}dex and scatter of 0.11 dex.

The initial evaluation of the network did not take into account the non-negligible uncertainties present in observations of real H II regions. Therefore, in the latter part of the paper, we introduced a Monte Carlo-based method of sampling the posterior distributions that takes luminosity errors into account, and tested the performance of the network for a wide range signal-to-noise levels, i.e., for different luminosity errors. For this, we added a simple Poisson noise model to the original training data generated from WARPFIELD-EMP. And we normalize our approach to the adopted uncertainty in the brightest emission line, which is typically Hα in real data. The influence of observational errors on posterior distributions can be summarized as followed.

7 The width of the posterior distribution gradually increases as a function of the luminosity error. However, the MAP estimates remain accurate until the error of the brightest emission line (i.e., minimum luminosity error, $\sigma_{\rm min}$) increases up to 0.1%.

8 If the minimum luminosity error is 1% or more, the posterior distributions show a skewed shape with a MAP estimate smaller than the true value and a long tail stretched to larger values than the true value. The skewness of the posterior distribution becomes more pronounced with increasing luminosity error.

9 If the luminosity error is too large ($\sigma_{\rm min} \sim 10\%$), the network provides similar posterior distributions regardless of the input observations. Specifically, the posterior distributions for five non-discretized parameters ($M_{\rm cl}$, SFE, $n_{\rm H,0}$, t, and $t_{\rm youngest}$) are significantly skewed and as wide as the entire parameter ranges in the training data. For discretized two parameters ($N_{\rm cluster}$ and phase), the MAP estimates are determined as the same value with $N_{\rm cluster}$ as 1 and Phase as 1 in most cases.

Overall the results of this study demonstrate that the cINN is a time-efficient and powerful tool to predict the fundamental parameters from the observations. We confirmed that with enough training data, the cINN learns the hidden rules within the training data well and can

provide accurate predictions of the physical parameters of H II regions from observations of key diagnostic emission lines.

3

Noise-Net: determining physical properties of H II regions reflecting observational uncertainties

This chapter is based on the paper Kang et al. (2023) published in Monthly Notices of the Royal Astronomical Society (MNRAS) in 2023. I am the first author and carried out all of the data analysis and writing of this paper. In the same manner, as in Chapter 2, we used the FREIA (Framework for Easily Invertible Architectures; Ardizzone et al. 2019b) to construct networks based on the cINN architecture.

Abstract

Stellar feedback, the energetic interaction between young stars and their birthplace, plays an important role in the star formation history of the universe and the evolution of the interstellar medium (ISM). Correctly interpreting the observations of star-forming regions is essential to understand stellar feedback, but it is a non-trivial task due to the complexity of the feedback processes and the degeneracy in observations. In our recent paper, we introduced a conditional invertible neural network (cINN) that predicts seven physical properties of star-forming regions from the luminosity of 12 optical emission lines as a novel method to analyse degenerate observations. We demonstrated that our network, trained on synthetic star-forming region models produced by the WARPFIELD-Emission predictor (WARPFIELD-EMP), could predict physical properties accurately and precisely. In this paper, we present a new updated version of the cINN that takes into account the observational uncertainties during network training. Our new network named Noise-Net reflects the influence of the uncertainty on the parameter prediction by using both emission-line luminosity and corresponding uncertainties as the necessary input information of the network. We examine the performance of the Noise-Net as a function of the uncertainty and compare it with the previous version of the cINN, which does not learn uncertainties during the training. We confirm that the Noise-Net outperforms the previous network

for the typical observational uncertainty range and maintains high accuracy even when subject to large uncertainties.

3.1 Motivation

Newly-formed stars and Giant Molecular Clouds (GMCs), the stellar birthplace, interact with each other via stellar feedback. Young massive stars born in a GMC inject a large amount of energy and momentum into the surrounding environment via stellar winds, radiation, thermal pressure from photoionised gas or supernovae (Krumholz et al. 2014; Klessen & Glover 2016). The feedback from young stars can disrupt further star formation by destroying the GMC or can locally promote new star formation (Shetty & Ostriker 2008; Dale et al. 2013; Rahner et al. 2019; Chevance et al. 2020b; Kim et al. 2021; Grudić et al. 2022).

To understand the influence of stellar feedback and the physical properties of the star-forming region on star formation, it is essential to correctly interpret the observed star-forming regions. However, the overall physical process occurring in the star-forming region is very complicated because the multiple feedback mechanisms are non-linearly coupled together and act simultaneously. Moreover, observations of star-forming regions are usually degenerate which means that different physical systems look similar in the observational space. Observational uncertainties as well as the degeneracy and complex physics make it even more difficult to describe the observation with theoretical models by using a classical fitting method.

We apply artificial neural networks (NNs; Goodfellow et al. 2016) to solve this complicated problem in star-forming regions. NNs can connect physical parameters and observational measurements through a statistical model, without designating a specific physical model. Recently, various machine learning techniques including NNs have been utilised in many astronomical fields e.g., to classify observations (Wu et al. 2019; Walmsley et al. 2021; Whitmore et al. 2021), to identify structures or exoplanets (Abraham et al. 2018; de Beurs et al. 2022), and to predict physical parameters (Fabbro et al. 2018; Ksoll et al. 2020; Olney et al. 2020a; Sharma et al. 2020a; Shen et al. 2022). In this study, we adopt the supervised learning approach, a type of machine learning that trains the network using labelled data sets, as we want to estimate specific physical parameters we want from quantities we can measure from observed star-forming regions.

In degenerate systems, the forward process that translates the physical parameters to observations is usually well-defined but involves information loss, from which the degeneracy arises. Due to the lost information, it is difficult to solve the inverse problem using a classical neural network trained through the inverse process. Therefore, we need a neural network that provides a full posterior probability distribution of the physical system conditioned on the given observation.

A conditional invertible neural network (cINN; Ardizzone et al. 2019a, 2021), a type of invert-

ible neural network (INN; Ardizzone et al. 2019b), is a deep learning architecture specialised for the inverse inference of degenerate systems. Unlike classical neural networks, a cINN is trained through the forward process of the system but also learns about the inverse process for free due to its invertibility. By using the additional latent variables in the network, the cINN captures the information otherwise lost during the forward process training and provides a full posterior distribution of physical parameters through the inverse process.

In Kang et al. (2022, hereafter Paper 1), we introduced a cINN that predicts seven physical parameters of H II regions using 12 optical emission-line luminosities. In this first study, we trained our network using synthetic H II region models produced by WARPFIELD-Emission predictor (WARPFIELD-EMP; Pellegrini et al. 2020), the pipeline modelling the synthetic observation based on the 1-dimensional stellar feedback code WARPFIELD (Rahner et al. 2017, 2018). We confirmed that our network predicts each parameter very accurately and precisely and validated the network predictions by re-simulating the emission-line luminosity of predicted models. Although the cINN presented in Paper 1 did not consider observational error in its prediction by definition, we introduced a way to obtain the posterior distribution reflecting the observational errors by modifying the posterior sampling method. Large observational errors worsened the overall performance, but we confirmed that our network was accurate in most cases unless the smallest error among 12 lines was larger than 0.1%.

In this study, we introduce a new type of cINN, which we term Noise-Net, that always reflects observational uncertainties in the prediction. Unlike our first cINN in Paper 1, Noise-Net learns not only about the relationship between physical parameters and luminosities, but also about the influence of luminosity errors during the network training. In this paper, we focus on comparing the performance of Noise-Net with the type of cINN used in Paper 1.

The paper is structured as follows. In Section 3.2, we explain the overall methodology used in this paper, including the structure and training of the cINNs and synthetic H II region database. In Section 3.3, we compare the performance of two cINNs as a function of the observational error using a statistical approach and individual models. We discuss the implication of our main results on the physical aspects and machine learning aspects in Section 3.4 and summarise the results of this paper in Section 3.5.

3.2 Methodology

3.2.1 cINN and Normal-Net

The cINN architecture (Ardizzone et al. 2019a, 2021) is an inverse problem solver specialised in degenerate systems. We assume that, in a degenerate system, different sets of physical parameters ($\mathbf{x}$) can be mapped onto an identical observation ($\mathbf{y}$) because of the information loss during the forward process. To capture the lost information, the cINN introduces the third component,

the latent variables ($\mathbf{z}$), and builds a bijective mapping between $\mathbf{x}$ and $\mathbf{z}$. The observation $\mathbf{y}$ is applied as a condition $\mathbf{c}$ to this mapping in both the forward (f) and inverse process:

$$\begin{aligned} \mathbf{z} &= f(\mathbf{x}; \mathbf{c} = \mathbf{y}), \\ \mathbf{x} &= g(\mathbf{z}; \mathbf{c} = \mathbf{y}), \end{aligned} \tag{3.1}$$

where $g = f^{-1}$ (Ardizzone et al. 2021). Due to the bijective mapping, the dimension of $\mathbf{z}$ is the same as the dimension of $\mathbf{x}$.

We train the network through the forward process to learn f and prescribe the latent variables to follow a standard multivariate normal probability distribution $p(\mathbf{z}) = N(0, \mathbf{I})$ with zero mean and unit covariance matrix, where $\mathbf{I}$ is the identity matrix with a dimension of $\dim(\mathbf{z}) \times \dim(\mathbf{z})$. As the cINN consists of invertible affine coupling blocks, the cINN automatically learns the inverse process g from the forward training. Following the inverse process g and sampling the latent variables from the prescribed distribution $p(\mathbf{z})$, we can obtain the posterior distribution of $\mathbf{x}$ for the given condition ($\mathbf{c}$), $p(\mathbf{x}|\mathbf{c} = \mathbf{y})$.

In Paper 1, we introduced a cINN that predicts seven physical parameters from the luminosity of 12 optical emission lines. The seven parameters are initial cloud mass $M_{\rm cl}$, star formation efficiency ϵ, cloud density $n_{\rm H,0}$, age of the first generation cluster (i.e., the oldest cluster), age of the youngest cluster, the number of clusters and the evolutionary phase of the cloud. The network presented in Paper 1, which we will refer to as **Normal-Net** in the following, used a basic cINN structure and training method predicting physical parameters from the given observations without considering the observational errors, $p(\mathbf{x}|\mathbf{y})$. Although the Normal-Net does not learn about observational errors ($\boldsymbol{\sigma}$) during the training, in Paper 1, we introduced a method to account for the observational uncertainties, $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\sigma})$, through a modification of the posterior sampling procedure and analysed the performance of the network as a function of the observational error (see Section 7 of Paper 1).

In this paper, we introduce **Noise-Net**, a variant of the cINN architecture and training method that can predict posterior distributions accounting for the observation errors, $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\sigma})$, without the additional steps required for the Normal-Net by learning the influence of errors during the training. In the following (Section 3.2.2), we describe the structure and training procedure of the Noise-Net in order to predict the parameters considering both observations and corresponding errors.

3.2.2 Noise-Net and noise training

Noise-Net and its training method are based on SoftFlow introduced by Kim et al. (2020). Kim et al. (2020) used the following ideas in their training to improve the performance of the network that reconstructs a two-dimensional pattern such as a spiral line from the latent variables

prescribed to an isotropic 2D Gaussian distribution. Firstly, Kim et al. (2020) randomly sampled the error (i.e., 1-σ width of the Gaussian distribution) and perturbed the original pattern (X) by adding the Gaussian noise based on the sampled error. Then they trained the network with the perturbed data (X') and use the sampled error as a condition. Kim et al. (2020) found that the pattern restoration accuracy of the network depended on the amount of error given by the condition. The network successfully reconstructed a clean pattern given the small error, and this result was better than those of the other networks trained without error. The situation in our study is slightly different to Kim et al. (2020) in that we use the cINN that already has a condition $\mathbf{y}$ and that we want to deal with the error of $\mathbf{y}$, not the error of $\mathbf{x}$. However, the idea of SoftFlow allowed us to design the Noise-Net that takes into account observation errors during the training.

We want the Noise-Net to predict physical parameters considering the given observation (luminosity) and the uncertainty of the observations (i.e., luminosity error). Thus, the Noise-Net, by definition, uses both $\mathbf{y}$ and the error of $\mathbf{y}$, $\boldsymbol{\sigma}$, as a condition of the network: $\mathbf{c} = [\mathbf{y}, \boldsymbol{\sigma}]$. In this study, we define the luminosity error $\boldsymbol{\sigma}$ as a fractional 1-sigma unitless uncertainty which is normalised by the corresponding luminosity value. Because of the additional condition, the dimension of $\mathbf{c}$ in the Noise-Net is twice as large as in the Normal-Net. As we use the information on 12 emission lines, the dimension of $\mathbf{c}$ of the Noise-Net in this paper is 24.

For the Noise-Net to learn the influence of the observational uncertainties, introducing the error as an additional condition of the network alone is not enough. We also need to modify the training strategy of the network. In this paper, we refer to the method of training Noise-Nets as noise training. The differences between noise training and normal training are two additional steps processed during the training on the fly.

The first step is to randomly sample the luminosity errors for each training model. For the network to learn about various $\boldsymbol{\sigma}$ values, the $\boldsymbol{\sigma}$ is not included in the training data but is randomly sampled from a given distribution at every training epoch and for every training model during the training. In this paper, we sample the errors in a logarithmic scale because the range of error values we want to train is wide. For each training model, the luminosity error of the $i-$th emission line, σ_i, is sampled from the uniform distribution,

$$p(\log \sigma_i) = U(a, b). \tag{3.2}$$

We sample the luminosity error of all 12 emission lines from one probability distribution (Eq. 3.2) with the same minimum (a) and maximum value (b) to simplify the training setup but it is also possible to use a different probability distribution for each emission line. In this study, we use the minimum error of -5 and maximum error of -0.5 which are equivalent to 0.001 and 31.6% error respectively.

The second step is to perturb the luminosity of the training model by adding random Gaussian noise to the true luminosity values ($\mathbf{y}^*$) based on the $\boldsymbol{\sigma}$ sampled in the first step. The perturbed luminosity of the $i-$th emission line (y'_i) is calculated by

$$y'_i = y^*_i \ (1 + r_i), \text{where } r_i \in N(0, \sigma_i^2). \tag{3.3}$$

In order to prevent the perturbed luminosity value from being negative because of a large amount of noise, we clip the perturbed luminosity to the minimum value of 1.

After these two steps, we train the network by using the true parameter values ($\mathbf{x}^*$) as an input and the group of perturbed luminosity and randomly sampled luminosity error ($[\mathbf{y}', \boldsymbol{\sigma}]$) as a condition to the forward process. In Paper 1, we trained the network (Normal-Net) by using the true parameters values and true luminosity values ($\mathbf{x}^*$ and $\mathbf{y}^*$) as an input and a condition (i.e., normal training method), so that the Normal-Net learned about the same values for each training epoch repeatably during the training. On the other hand, during the noise training, the Noise-Net learns about various luminosity and error values from an identical training model because errors and perturbation noises are randomly sampled at every training epoch. Consequently, the prediction power of the trained Noise-Net varies as a function of the luminosity error.

3.2.3 Training data: WARPFIELD-EMP

To train the cINN, which adopts a supervised learning approach, we need numerous sets of data containing both the observable quantities ($\mathbf{y}$) and physical parameters that we want to predict ($\mathbf{x}$). However, it is difficult to collect a sufficient number of well-analysed H II regions from real observations. Hence, to train and test the networks, we use the same database used in our first paper (Paper 1), which consists of 505,748 synthetic H II region models that we produced by using the WARPFIELD-EMP pipeline (Pellegrini et al. 2020). Based on the evolution of a massive star-forming cloud described by the 1D stellar feedback code WARPFIELD (Rahner et al. 2017, 2018), WARPFIELD-EMP produces the continuum and line emission of the model cloud at a large series of output times by using CLOUDY (Ferland et al. 2017) to calculate the continuum and line emissivities and POLARIS (Reissl et al. 2016) to compute the transfer of the resulting radiation through the cloud. In this section, we briefly summarise our synthetic H II region models and database that is described in detail in Paper 1.

Synthetic H II regions

WARPFIELD (Rahner et al. 2017), which is the basis of our synthetic model, simulates the evolution of an isolated massive cloud acted on by stellar feedback. At the beginning of the evolution ($t = 0$), a star cluster with mass M_* is formed at the centre of the cloud. The cluster mass is determined by two initial conditions, the initial cloud mass ($M_{\rm cl}$) and the star formation

efficiency (ϵ): $M_* = \epsilon M_{\rm cl}$. WARPFIELD treats star formation in a highly idealised way and assumes that all stars in the cluster form instantaneously. Due to the stellar feedback produced by the cluster, the cloud is separated into distinct zones: a diffuse central bubble (i.e., inner free wind zone), a dense shell surrounding the bubble with hot shocked wind materials, and a static cloud region outside of the shell. The evolution of the cloud is described by solving the equation of motion of the dense shell considering several feedback mechanisms such as stellar winds, supernovae, thermal gas pressure, radiation pressure, and gravity. WARPFIELD assumes that within the shell the gas is in quasi-hydrostatic equilibrium and that the evolution of the cloud can be distinguished into four distinct evolutionary phases according to the dynamics of the shell: Phase 1, 2, 3, and 0.

In the first evolutionary phase, Phase 1, the shell expands rapidly, driven by the thermal pressure of the hot shocked wind material. During this phase, the evolution of this hot gas is adiabatic and the influence of gravity and radiation pressure can be ignored. Phase 1 ends once the inner bubble loses its hot gas, either because the hot gas radiatively cools (which occurs after a time $t_{\rm cool}$) or because the bubble bursts and the hot gas leaks out. In WARPFIELD (Rahner et al. 2017), it is assumed for simplification that the bubble bursts only when the shell sweeps up the entire natal cloud ($t_{\rm sweep}$, i.e., when $R_{\rm shell} > R_{\rm cloud,\ initial}$). If the bubble cools down before the shell sweeps up the whole materials in the cloud ($t_{\rm cool} < t_{\rm sweep}$), the evolutionary phase turns into Phase 2, otherwise it turns into Phase 3 directly. After Phase 1, the shell expansion is dominated by radiation pressure and the ongoing ram pressure exerted by supernovae and stellar winds. The counteracting effects of the gravity of the star cluster and the self-gravity of the shell are now no longer negligible.

The fate of the clouds is divided into two options, depending on the balance between stellar feedback and gravity. If the feedback is more dominant than gravity, the shell continues to expand into the low-density interstellar medium after sweeping up the whole material in the cloud (Phase 3). The density of the shell gradually decreases and the shell finally dissolves into the ambient ISM. The evolution ends when the maximum number density of the shell is smaller than 1 cm^{-3} for more than 1 Myr. On the other hand, if gravity becomes more dominant than the feedback, the shell stops expanding and recollapses to the centre. This collapse triggers the formation of a new star cluster when the shell radius becomes smaller than 1 pc. The recollapsing phase until the birth of the new cluster is labelled as Phase 0. For simplicity, it is assumed that the star formation efficiency of the new birth is the same as the first burst ($M_{*,2} = \epsilon(M_{\rm cl} - M_{*,1})$). The evolution continues with the new shell formed by the stellar feedback dominated by the new cluster.

In the WARPFIELD model, stars within the cluster follow a Kroupa initial mass function (Kroupa 2001) and the time-dependent effect of the stellar feedback from the evolving star cluster is calculated with STARBURST99 (Leitherer et al. 1999, 2014) by using the Geneva

evolution tracks (Ekström et al. 2012; Georgy et al. 2012, 2013). Hence, the spectral energy distribution (SED) of the WARPFIELD model is time-dependent due to the evolving cluster and cloud and especially complex if the SED of multiple clusters with different ages are combined within one cloud due to the recollapse.

Based on the time-dependent evolution information of the WARPFIELD model, WARPFIELD-EMP uses CLOUDY (C17, Ferland et al. 2017), a spectral synthesis code, to calculate emissivities of various lines and continuum processes. We first run CLOUDY for the dense shell to obtain the emissivities as a function of radial position within the shell. If the natal cloud surrounding the shell still remains, we run CLOUDY a second time to calculate the emissivities from the static cloud by using the output of the first CLOUDY run for the shell as an incident flux. Lastly, the radial information on the emissivities is passed on to POLARIS (Reissl et al. 2016, 2019) to calculate the final luminosity information considering the attenuation within the shell and the natal cloud. In WARPFIELD-EMP, POLARIS is used in ray-tracing mode to solve the radiative transfer equation for the rays passing through a 3D grid of the shell and cloud. We use a 3D spherical grid for POLARIS because the WARPFIELD model is a 1D spherical symmetric model. Finally, the output of POLARIS is projected onto a 2D space and the 2D map is spatially integrated to obtain the velocity-integrated 1D luminosity of lines and continuum of a given WARPFIELD model at a certain time.

Our synthetic model contains information on many fundamental physical parameters together with the corresponding continuum emission and a series of lines for a given WARPFIELD model at a given time. We only use seven parameters and 12 emission lines in this study, but WARPFIELD-EMP originally provides many other lines within a wide frequency range from optical to radio listed in Table D1 and D2 in Pellegrini et al. (2020).

Training set and test set

In our first study (Paper 1), we built and introduced a new database that consists of 505,748 WARPFIELD-EMP synthetic H II region models evolved from 10,000 initial WARPFIELD clouds. In this study, we use the same database to train and evaluate our networks.

A WARPFIELD-EMP model in our database is uniquely determined by four independent physical parameters: initial mass of the WARPFIELD cloud ($M_{\rm cl}$), star formation efficiency (SFE), initial cloud density ($n_{\rm H,0}$), and the age of the cloud (t). The first three parameters are the initial conditions of the WARPFIELD calculation. As the evolution of the WARPFIELD cloud begins, the age becomes the fourth independent parameter, which is equivalent to the age of the first cluster, because the evolution of WARPFIELD starts with the formation of the first cluster. There are several other parameters that can be varied in the initial conditions of a WARPFIELD cloud (e.g., metallicity), but we fixed all of these parameters at constant values. In particular, we use solar metallicity and a constant radial profile of the initial density and turn

off the effects of turbulence and the magnetic field for all models in our database.

For the initial 10,000 WARPFIELD clouds, we randomly sampled $M_{\rm cl}$, SFE, and $n_{\rm H,0}$ of the clouds within the range of 10^{5-7} $M_\odot$, 2–10 %, and 100–500 cm^{-3}, respectively. We evolve the clouds until they reach an age of 30 Myr, although depending on the physical conditions of the cloud, the evolution can end earlier if the shell dissolves into the ambient ISM. WARPFIELD stores the time-dependent evolution in the prescribed time interval and we adopt a 0.1 Myr interval for our clouds. However, for computational efficiency reasons, we do not calculate the emission for all saved models, but instead run CLOUDY and POLARIS only when the physical conditions of the cloud (e.g., shell density, shell mass, shell radius, and ionising photon flux) change by a large enough amount to cause a significant change in the emission or when the evolutionary phase of the cloud changes. Therefore, the time interval of the database is not constant due to the adoptive time sampling method. In particular, the time intervals tend to grow longer as the cloud ages, because the emission changes more rapidly in the early evolutionary phase. For more details of the time distribution of the outputs, we refer the reader to Figure 1 in Paper 1.

Although the initial cloud parameters are uniformly distributed, the physical parameters of the models in the database are not evenly distributed (see both Figure 3.1 in this work and Figure 1 in Paper 1), because each cloud evolves differently and terminates its evolution at different ages. For example, if the gravity of the cloud is more dominant than the stellar feedback and the cloud recollapses repeatedly, this cloud survives longer and saves more models than clouds with strong stellar feedback. Hence, the distribution of $M_{\rm cl}$ and SFE are biased toward a large mass and small SFE, respectively. Additionally, the proportion of models with only one cluster (single cluster model) and the proportion of models in Phase 3 is much higher than in the other cases.

The distribution of training data affects network performance. In the previous study, we demonstrated that the network trained with this database performs better for H II regions with more common characteristics in the database. Specifically, the posterior distribution was usually more accurate and narrower for young and single-cluster models rather than old models with multiple clusters. For this reason, it is better to sample the training data as evenly as possible or to make the distributions even through post-processing. However, it is also not easy, because the evolution of a cloud is a very complex function of the four independent parameters so it is difficult to predict the evolution from the given initial conditions. If we additionally sample more clouds with less populated characteristics in the current distributions, this could in turn introduce new biases in the other parameters. Therefore, we use the current database without additional modification, although the current distribution is not entirely optimal in terms of training.

We randomly divide the database with 505,748 models into a training set and a test set, using the former to train the network, while the latter serves to evaluate the trained network. In this

Table 3.1: List of the seven physical parameters (i.e., **x** of the network) and list of the twelve emission lines whose luminosities are used as **y**. The information listed in this table is the same as Tables 1 and 2 in Paper 1.

Parameter	Symbol
initial mass of star-forming cloud	$M_{\rm cl}$ [$M_\odot$]
initial star formation efficiency	SFE [%]
initial cloud number density	$n_{\rm H,0}$ [cm^{-3}]
age of the first cluster	t [Myr]
age of the youngest cluster	$t_{\rm youngest}$ [Myr]
number of star clusters	$N_{\rm cluster}$
evolutionary phase of the cloud	Phase
Line	**Wavelength**
[O II]	3726Å
[O II] (blend)	3727Å
[O II]	3729Å
Hβ	4861Å
[O III]	5007Å
[O I]	6300Å
Hα	6563Å
[N II]	6583Å
[S II]	6716Å
[S II] (blend)	6720Å
[S II]	6731Å
[S III]	9531Å

study, we use 90% of the database (455,174 models) to train the network and use the remaining 10% (50,574 models) to evaluate the performance of the networks.

Our database contains various physical parameters and luminosity of several lines and continuum for each WARPFIELD-EMP model but we only use 12 optical emission lines and seven physical parameters in our study (Table 3.1), which are the same choices as in Paper 1. The seven target parameters consist of initial cloud mass ($M_{\rm cl}$), star formation efficiency (SFE), initial cloud density ($n_{\rm H,0}$), age of the first cluster (t), age of the youngest cluster ($t_{\rm youngest}$), the number of the clusters ($N_{\rm cluster}$), and the evolutionary phase of the cloud. We choose 10 optical lines within the range of 3700–9000 Å: [O II] 3726Å, [O II] 3729Å, Hβ 4861Å, [O III] 5007Å, [O I] 6300Å, Hα 6563Å, [N II] 6583Å, [S II] 6716Å, [S II] 6731Å, and [S III] 9531Å. We also add the total strength of the [O II] and [S II] doublets, referred to as [O II] 3727Å (blend) and [S II] 6720Å (blend).

3.2.4 Network setup

The goal of this study is to introduce the new Noise-Net together with the noise training and to compare the performance of the Noise-Net and Normal-Net. Thus we train one Normal-Net and one Noise-Net using the same network setup except for the intrinsic differences between the Normal-Net and the Noise-Net described in the previous section (e.g., the dimension of $\mathbf{c}$). In this section, we explain the network architecture of our two networks and the data pre-processing steps required to train or use the network. Most of the setup used in this study is the same as in Paper 1, except for some hyperparameters.

Network construction

To construct the cINN architecture, we use the FrEIA (Framework for Easily Invertible Architectures; Ardizzone et al. 2019b, 2021) which is based on the 'PyTorch' library (Paszke et al. 2019) as in Paper 1.

The cINN consists of a series of affine coupling blocks that follow the architecture proposed by Dinh et al. (2016a). In this study, we use 16 affine coupling blocks to build a network, two times more than that of the network in Paper 1. Each affine coupling block splits the input $\mathbf{u}$ into two parts (i.e., $\mathbf{u_1}$ and $\mathbf{u_2}$) and passes each part through affine transformations following

$$\begin{aligned} \mathbf{v_1} &= \mathbf{u_1} \odot \exp(s_2(\mathbf{u_2}, \mathbf{c})) + t_2(\mathbf{u_2}, \mathbf{c}), \\ \mathbf{v_2} &= \mathbf{u_2} \odot \exp(s_1(\mathbf{v_1}, \mathbf{c})) + t_1(\mathbf{v_1}, \mathbf{c}). \end{aligned} \tag{3.4}$$

The outputs of the two affine transformations (i.e., $\mathbf{v_1}$ and $\mathbf{v_2}$) are coupled into the final output $\mathbf{v}$.

The invertibility of the cINN architecture comes from the invertibility of each affine coupling block. After training the cINN through the forward process, we can use the inverse process of the trained network following

$$\begin{aligned} \mathbf{u_2} &= (\mathbf{v_2} - t_1(\mathbf{v_1}, \mathbf{c})) \odot \exp(-s_1(\mathbf{v_1}, \mathbf{c})), \\ \mathbf{u_1} &= (\mathbf{v_1} - t_2(\mathbf{u_2}, \mathbf{c})) \odot \exp(-s_2(\mathbf{u_2}, \mathbf{c})). \end{aligned} \tag{3.5}$$

The internal transformations s_i and t_i do not need to be invertible themselves, because they are only evaluated in the forward direction in both the forward and inverse process of the cINN. In this paper, we adopt a single sub-network as an internal transformation of each affine coupling block (i.e., the GLOW configuration; Kingma & Dhariwal 2018b). For each sub-network, we apply a simple fully connected architecture with 6 layers and a width of 256, using the rectified linear units (ReLU) as the activation functions. After each affine coupling block, we add an invertible permutation layer to mix the information stream. The permutation layer is a random

orthogonal matrix and is fixed during the training.

As shown in Eq. 3.4 and 3.5, the condition ($\mathbf{c}$) of the cINN architecture is always used as an additional input for each transformation in both the forward and inverse processes. However, before applying the condition to the affine coupling blocks, we pass the condition through an additional feed-forward network, the conditioning network, to extract higher-level features. According to Ardizzone et al. (2019a), using a conditioning network in complex systems helps the efficient conditioning of the cINN by reducing the burden of the main network (i.e., a series of invertible blocks) to re-learn higher-level features in each affine coupling block. The conditioning network can be either pretrained or trained together with the main network (Ardizzone et al. 2019a). In this study, we jointly train the main network and the conditioning network because the latter method helps extract features that are more relevant to $\mathbf{x}$. For the conditioning network, we adopt a simple fully connected feed-forward network with three layers and a width of 512.

Compared to Paper 1, we double the number of affine coupling blocks and the number of layers of each internal sub-network, because deepening the network into the current setup improves the performance, especially for the Noise-Net. In Appendix B.2, we will cover the influence of network depth on the prediction power of Noise-Nets and Normal-Nets in detail.

After constructing the network with the above setup, we train the network to minimize the maximum log-likelihood loss,

$$\mathcal{L} = \mathbb{E}_i\Big[\frac{\|f(\mathbf{x}_i;\mathbf{c}_i,\theta)\|_2^2}{2} - \log|J_i|\Big], \tag{3.6}$$

as described in Ardizzone et al. (2019a) and Ksoll et al. (2020), where $|J_i|$ denotes the determinant of the Jacobian matrix $|J_i| = \det\left(\frac{\partial f}{\partial \mathbf{x}}|_{\mathbf{x}_i}\right)$ and $\mathbf{x}_i$ is the physical parameters of some training sample with the corresponding condition $\mathbf{c}_i$. During the training, we calculate the test loss, that is the loss calculated with the test set, as well as the training loss calculated with the training set. The network is trained until the deviation between the training loss and test loss is small and both losses converge. The training time of the network depends on the batch size and the number of training epochs. Training one network for 100 epochs using a batch size of 256 took about 4 hours with NVIDIA GeForce RTX 2080 Ti graphic card and used about 1500 MB GPU memory and the size of the trained network is about 130MB. Training time and the size of the network also depend on the depth of the network (i.e., the number of affine coupling blocks and the number of layers of internal sub-network). By doubling both the number of layers and blocks compared to Paper 1, training time and the network size have increased by about 2 and 4 times, respectively.

Data pre-proccessing

When training the cINN or using the trained cINN, we use pre-processed physical parameters, observations and observation errors as described in Paper 1. For $N_{\rm cluster}$ and phase that have discretized distributions with a sampling interval of 1, we smooth out the distributions by adding a small Gaussian noise with a standard deviation of 0.05. In Paper 1, we demonstrated that smoothing out the discretized distribution improves the prediction power of the network (see Section 8.2. of Paper 1 for details). Please note that we smooth out these parameters only when we train the network.

For variables with a relatively broad range of values in linear space such as emission line luminosity ($\mathbf{y}$), we convert them into logarithmic scale. In the case of the noise training, we convert the luminosity after perturbation ($\mathbf{y}'$ from Eq. 3.3). As we sample the luminosity error $\boldsymbol{\sigma}$ from the wide distribution (Eq. 3.2), we transform $\boldsymbol{\sigma}$ in log-scale as well. Next, we re-scale the distribution of $\mathbf{x}$, $\mathbf{y}$, and $\boldsymbol{\sigma}$ by using linear transformations. For physical parameters, x_i, we transform x_i to $\hat{x}_i$ that has zero mean and unit standard deviation following

$$\hat{x}_i = (x_i - \mu_{x_i}) \cdot \frac{1}{s_{x_i}}, \tag{3.7}$$

where μ_{x_i} and s_{x_i} are the mean and the standard deviation of the physical parameter x_i calculated using the entire database. In the case of observation (y_i), we first centre them ($\tilde{y}_i = y_i - \mu_{y_i}$) and whiten the observation matrix following Equation 35 in Hyvärinen & Oja (2000) ($\hat{\mathbf{Y}} = \mathbf{W}_{\tilde{\mathbf{Y}}}\tilde{\mathbf{Y}}$) to have distributions with unit variance for each emission line and identity matrix for the covariance matrix of observations. To calculate μ_{y_i} and $\mathbf{W}_{\tilde{\mathbf{Y}}}$, we use true values ($\mathbf{y}^*$) of the entire database, not using the perturbed values, even for the Noise-Net.

For the observation error σ_i, we apply the same linear transformation used for physical parameters following

$$\hat{\sigma}_i = (\sigma_i - \mu_{\sigma_i}) \cdot \frac{1}{s_{\sigma_i}} \tag{3.8}$$

and change the mean and standard deviation of the distribution to zero and unity. As the errors are randomly sampled during the noise training from $p(\log \sigma)$ of Eq. 3.2 unlike the physical parameters and observations, we use the mean and the standard deviation of Eq. 3.2 for μ_{σ_i} and s_{σ_i}. We transform x_i, y_i and σ_i to $\hat{x}_i$, $\hat{y}_i$, and $\hat{\sigma}_i$, respectively, when training the network. When using the inverse process to obtain a posterior distribution, we transform y_i and σ_i to $\hat{y}_i$, and $\hat{\sigma}_i$ and transform $\hat{x}_i$ calculated by the network to x_i.

3.2.5 How to sample posterior estimates from the network

As explained in Section 3.2.1, the inverse process of the cINN calculates $\mathbf{x}$ from the corresponding condition $\mathbf{c}$ and latent variable $\mathbf{z}$, following Eq. 3.1. We obtain the posterior distribution $p(\mathbf{x}|\mathbf{c})$ by

sampling the latent variables N_z times from the prescribed probability distribution $p(\mathbf{z}) = N(0, \mathbf{I})$ and then calculating the corresponding $\mathbf{x}$ using the inverse process g in Eq 3.1. Thus, the number of obtained posterior estimates is N_z. In this way, we obtain $p(\mathbf{x}|\mathbf{y})$ from the Normal-Net and $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\sigma})$ from the Noise-Net. The standard Normal-Net provides a posterior distribution without considering the error as it does not learn the error during the training. However, in Paper 1, we introduced a Monte Carlo-based marginalisation method that allows us to obtain $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\sigma})$ from the Normal-Net. In this study, we only use the posterior distribution considering the observation error to compare the performance of the Noise-Net and the Normal-Net as a function of the error. Here, we summarise the marginalisation method introduced in Paper 1 to obtain posterior distributions that account for the errors from the Normal-Net.

The overall process of the marginalisation method for the Normal-Net is described by

$$p(\mathbf{x}|\mathbf{y}, \boldsymbol{\sigma}) = \int p(\mathbf{x}|\mathbf{y}')\, q(\mathbf{y}'|\mathbf{y}, \boldsymbol{\sigma})\, d\mathbf{y}', \tag{3.9}$$

where q is a profile of a luminosity error, which is a Gaussian distribution in our paper, i.e., $N(\mathbf{y}, \boldsymbol{\sigma}^2)$. Given a set of 12 emission line luminosities ($\mathbf{y}$) and the corresponding 1-sigma unitless errors ($\boldsymbol{\sigma}$) measured from observation, the first step is to perturb the luminosity in the same way that we do in the noise training, i.e. following Eq 3.3. In both cases, we assume that the perturbed luminosity follows a Gaussian distribution centred on the value y with a standard deviation of σ. By sampling Gaussian noise for each emission line N_p times, we obtain N_p sets of perturbed luminosity $(\mathbf{y}'_1, ..., \mathbf{y}'_{N_p})$ from a given observation. Then, for the $i-$th perturbed luminosity set $\mathbf{y}'_i$, we sample the latent variables $N_{z,\mathrm{normal}}$ times to get a posterior distribution $p(\mathbf{x}|\mathbf{y}'_i)$. By superimposing the posterior distributions of all perturbed luminosity sets, we finally obtain the $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\sigma})$ that consists of $N_p \times N_{z,\mathrm{normal}}$ posterior samples from the Normal-Net.

From Paper 1, we found that it is important to produce a sufficient number of perturbed luminosity sets (i.e., N_p) to get a smooth posterior distribution. In the previous study, we used $N_p = 3000$ and $N_{z,\mathrm{normal}} = 100$, which are large enough to ensure a smooth distribution even when the $\boldsymbol{\sigma}$ is large. However, in the case of small $\boldsymbol{\sigma}$, a smaller N_p is enough to produce a smooth distribution because the range of the perturbation is narrow. To reduce the computation time required for the posterior sampling and analysis, we flexibly adjust N_p and $N_{z,\mathrm{normal}}$ values depending on the network performance and the magnitude of the error. Considering the range of error values used in this study, we keep $N_{z,\mathrm{normal}}$ fixed at 50 and vary N_p between 1000 and 2500 depending on the error. If the luminosity error is too large, the posterior sometimes shows a very spiky distribution despite large N_p and $N_{z,\mathrm{normal}}$. We confirm that in such cases, the spiky shape is not due to the lack of N_p or $N_{z,\mathrm{normal}}$ but an inevitable result of the large errors. In the case of the Noise-Net, we only need to decide the number of latent variable samples, $N_{z,\mathrm{noise}}$. In this study, we use $N_{z,\mathrm{noise}}$ of 20,000 when we use the Noise-Net.

3.3 Noise-Net vs. Normal-Net

In this section, we compare the performance of the Normal-Net and the Noise-Net as a function of the observation error. As mentioned in Section 3.2.4, we trained both networks with the same setup except for the fundamental differences between the Normal-Net and the Noise-Net. We use the synthetic H II region models of the test set instead of real observations to concentrate on the validation of the cINN rather than the validation of the synthetic training data.

3.3.1 Experiment setup

Sample selection

To reduce the computation time for the evaluations, we only use 100 models among the 50,574 models in the test set as a representative subset. We randomly select these 100 test models based on the following criteria. Please note that we use the same selection criteria used in Paper 1 but select a new set of 100 test models. We present the parameter distributions of the selected models as well as their locations in the BPT diagram (Baldwin et al. 1981) in Figure 3.1.

First, we exclude models with [N II]/H$\alpha < 10^{-3}$ or [O III]/H$\beta < 10^{-2}$, considering the typical line-ratio values of observed star-forming galaxies (Kauffmann et al. 2003; Kewley et al. 2006) and H II regions (Sánchez et al. 2015; Rousseau-Nepton et al. 2018). Next, we select 4 of the 100 models from the extreme area on the BPT diagram beyond the revised demarcation curve between active galactic nuclei (AGNs) and starburst galaxies proposed by Kauffmann et al. (2003). As shown in Figure 3.1, one of the four models is beyond the more extreme demarcation curve of Kewley et al. (2001). For the other 96 models, we set a maximum fraction for N_{cluster} and phase to prevent too many similar models from being selected. The distributions of the entire training data (red line in the left panel of Figure 3.1) for these two parameters have biases toward the single cluster and Phase 3 models, respectively. We limit the fraction of single cluster models to 60% and the fraction of Phase 3 models to 40% to reduce this bias.

Noise model and error selection

Our synthetic test models are, by definition, fully accurate without any uncertainty measurements. Thus, we assume a simple noise model, the same method used in Paper 1, to assign mock luminosity errors for our experiments. Although diverse sources affect errors measured in real observations, we adopt a noise model that only considers Poisson noise. We assume that the error of each emission line is an independent random variable and ignore any covariance between physically related lines such as blended lines and their components. In our noise model, given the 1-sigma error of the brightest emission line, the errors of the remaining 11 emission lines are

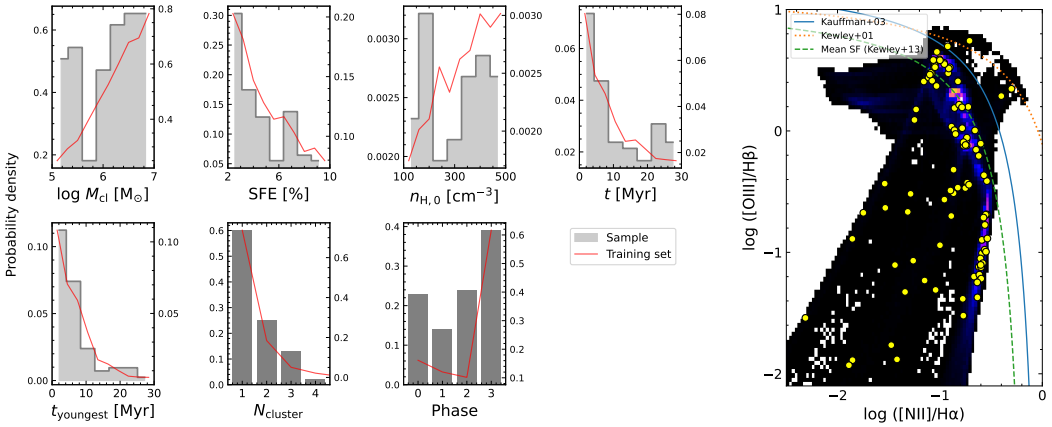

Figure 3.1: Distribution of seven physical parameters of the selected 100 test models (left) and their locations in the BPT diagram (right). In the left figure, the grey histogram and left y-axis of each panel show the distribution of the selected sample, and the red line and right y-axis show the distribution of 455,174 models of the training set. In the right figure, the background two-dimensional colour histogram shows the number density of the entire training set and yellow dots indicate the locations of 100 test models.

automatically determined following

$$\sigma_{\rm line} = \sigma_{\rm b} \times \sqrt{\frac{L_{\rm brightest}}{L_{\rm line}}}, \tag{3.10}$$

where $\sigma_{\rm b}$ is the error of the brightest emission line of the observation, which is the smallest error among 12 emission lines.

Using the error of the brightest emission line ($\sigma_{\rm b}$) as the representative error of one observation, we investigate the influence of the error on the posterior distribution by increasing $\sigma_{\rm b}$. We select 16 values from 0.01% to 10% in 0.2 dex intervals on a logarithmic scale. For each network, we sample the posterior distributions of each parameter for the 100 test models with varying $\sigma_{\rm b}$. As mentioned in Section 3.2.5, the number of posterior samples for one observation varies depending on the network and the size of the error. In Table 3.2, we list the numbers of sampling used in this experiment depending on the $\sigma_{\rm b}$ value. Especially when the given error is large, our network sometimes returns physically incorrect or extremely extrapolated posterior estimates such as negative star formation efficiencies or cluster ages larger than 100 Myr. We exclude all of these unrealistic posterior estimates before our analysis.

Although we use $\sigma_{\rm b}$ as the representative error of one observation, the errors of the other 11 emission lines of the 100 test models with the same $\sigma_{\rm b}$ value are all different, because these errors are determined by the luminosity ratio between the line and the brightest emission line, which is typically the Hα line. The error of the faintest emission line, usually the [O I] 6300Å, is on average 17 times larger than the error of the brightest emission line. The error of the remaining 11 lines excluding the brightest emission line is on average 3.8 times larger than $\sigma_{\rm b}$.

Table 3.2: List of the numbers used to sample a posterior distribution depending on the error of the brightest emission line ($\sigma_{\rm b}$).

$\sigma_{\rm b}$ [%]	$N_{\rm p}$ [1]	$N_{\rm z,normal}$ [2]	$N_{\rm z,noise}$ [3]
$\sigma_{\rm b} \leq 0.1$	1000	50	20000
$0.1 < \sigma_{\rm b} \leq 1$	2000	50	20000
$1 < \sigma_{\rm b} \leq 10$	2500	50	20000

[1] the number of perturbed mock luminosity sets for the Normal-Net
[2] the number of latent variable sampling for each mock luminosity set for the Normal-Net
[3] the number of latent variable sampling for the Noise-Net

Evaluation methods

To evaluate the prediction power of the network, we introduce two evaluation indices used in this study that describe the accuracy and precision of the network prediction, respectively.

The first index, which represents the accuracy of the network, is given by the deviation between the posterior estimate and the ground truth value of the test model. For $N_{\rm cluster}$ and phase, we use the linear deviation ($X - X^*$) and for the other five parameters, we use the logarithmic deviation $\left(\log \frac{X}{X^*}\right)$. To calculate the accuracy index we either use all posterior samples or only one representative estimate from the posterior distribution. For the representative value, we measure the maximum a posteriori (MAP) point estimates by performing a Gaussian kernel density estimation on the 1D posterior distribution of each parameter and finding the maximum of the derived probability density.

The second index, the precision index, is the uncertainty interval at the 68% confidence level (i.e., u_{68}). The u_{68} value represents the width of the 1D posterior distribution similar to the width of ± 1 standard deviation.

For the 3200 posterior distributions obtained from the two networks for our 100 test models with 16 different $\sigma_{\rm b}$ values, we measure the accuracy and precision indices of each parameter. In the following sections, we compare the Normal-Net and the Noise-Net by using the measured performance indices.

3.3.2 Statistical comparison

Figure 3.2 shows the histograms of the two accuracy indices for the 16 different $\sigma_{\rm b}$ values for the Normal-Net (red) and the Noise-Net (blue). We use the MAP accuracy in the upper panels and the accuracy using the entire posterior estimates in the lower panels. In the upper panels, the MAP accuracy distribution of the Normal-Net becomes wider and shifted with increasing error. The shift of the distribution is well described by the median value of the distribution denoted by the red x-shape mark. In particular, in the case of the first cluster age (t) and the youngest

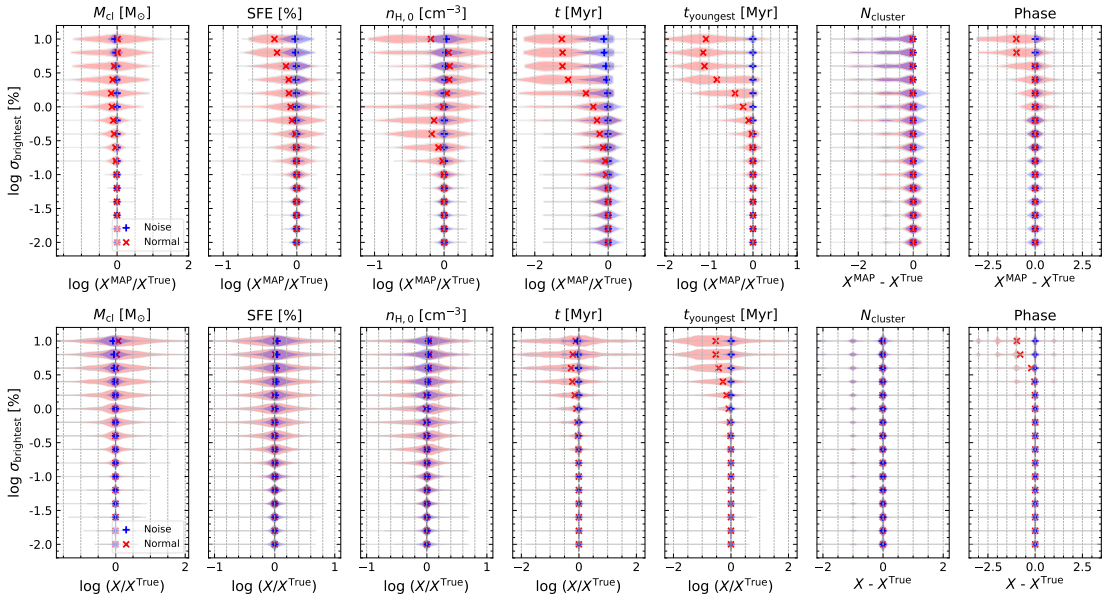

Figure 3.2: Histograms of two accuracy measures using 3200 posterior distributions (100 test models, 16 luminosity errors of the brightest emission line) obtained from the Noise-Net (blue histograms) and the Normal-Net (red histograms). We use the MAP estimates in the first row and use the entire posterior estimates in the second row. The blue plus marks (Noise-Net) and the red cross marks (Normal-Net) indicate the median values of the histograms. We use the logarithmic deviation between the posterior estimates and the ground truth values of the test models except for $N_{\rm cluster}$ and phase where we use the linear deviation.

cluster age ($t_{\rm youngest}$), the median value is shifted by more than 1 dex toward negative value when the error is larger than 2.5% (log $\sigma_{\rm b} > 0.4$). This trend was already revealed in Paper 1, where we found the Normal-Net usually returned a very young age estimate with a single cluster when the error is large.

On the other hand, the prediction of the Noise-Net is on average more accurate than the Normal-Net for all seven parameters. The blue histogram is overall much narrower than the red histogram and the difference in width becomes more clear when the error is larger than 1%. The accuracy distributions for $M_{\rm cl}$ and the youngest cluster age clearly show this trend. Contrary to the Normal-Net, the median value of the Noise-Net is always close to 0 without an offset even for the largest error of 10%. The age of the first cluster is the only parameter that shows a shift of the median value for a large error, but the offset of around 0.2 dex is still small compared to the Normal-Net where the median value is shifted by about 1.3 dex at $\sigma_{\rm b}$ of 10%. Unlike for the age of the first cluster, the Noise-Net shows an overwhelmingly accurate prediction for the youngest cluster age, even at an error of 10% compared to the Normal-Net.

The difference in the width between the blue and red histograms is noticeable as well when we evaluate the accuracy of the entire posterior estimates (lower panels in Figure 3.2). The difference in the width is distinct when the error is large, especially in the case of $M_{\rm cl}$, the oldest cluster age and the youngest cluster age. However, in the case of the density, the width of the Noise-Net and the Normal-Net appears fairly similar. The median value of the histogram

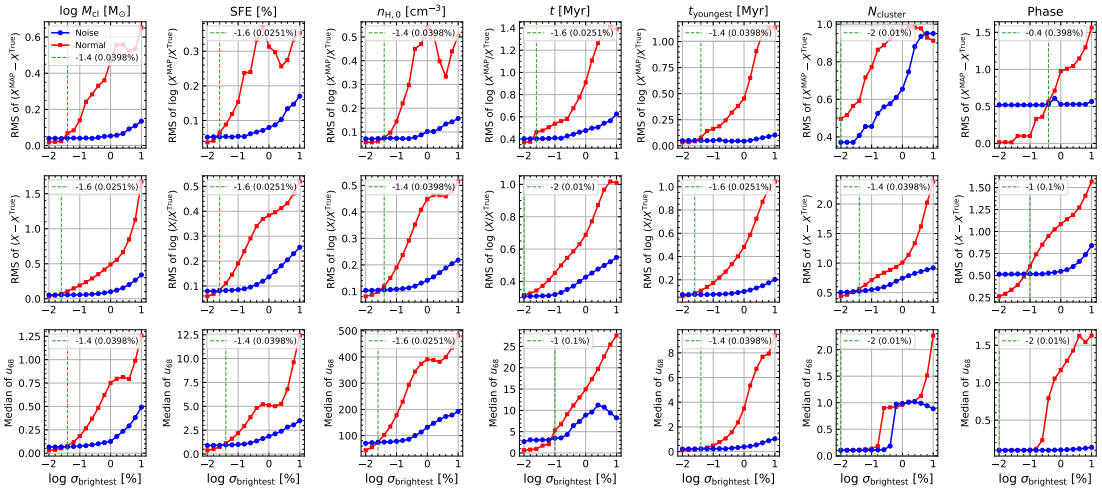

Figure 3.3: Accuracy and precision of the Noise-Net (blue lines) and the Normal-Net (red lines) as a function of the luminosity error of the brightest emission line ($\sigma_{\rm brightest}$) using 100 test models. We present the RMS of two accuracy measures: MAP accuracy in the first row and accuracy using all posterior estimates in the second row. The last row shows the median of the uncertainty at a 68% confidence interval (u_{68}) which represents the precision of our network. Vertical green dashed lines show the error values at the turning points where the Noise-Net begins to perform better than the Normal-Net.

mostly exhibits no offset for both Noise-Net and Normal-Net, except for the distributions of the youngest cluster age and phase of the Normal-Net. Based on the two violin plots in Figure 3.2, we discover that the predicted posterior distributions from both networks widen with increasing error, but the Noise-Net results remain much narrower than the Normal-Net. Furthermore, the Noise-Net maintains the accurate MAP prediction at large error, whereas the Normal-Net does not.

Figure 3.2 reveals that the Noise-Net performs better than the Normal-Net overall, especially for large errors, and resolves the offset issues of the Normal-Net. In Figure 3.3, we demonstrate how the accuracy and precision indices on average change with increasing error in a more quantitative way. The first two rows show the root mean square (RMS) value of the two accuracy indices, respectively, and the third row presents the median value of the precision index (u_{68}).

As already indicated by Figure 3.2, Figure 3.3 reveals that the Normal-Net performs better than the Noise-Net for small errors up until a critical error value is reached, where the Noise-Net then surpasses the Normal-Net. In each panel, we present the turning point, from which the Noise-Net predicts better than the Normal-Net, with the green vertical dashed line and in the legend. The turning point falls mostly around a value of 0.025% – 0.04% (log $\sigma_{\rm b}$: -1.6 – -1.4). In the case of phase accuracy, the turning point is larger than the other parameters. However, the RMS value of the Noise-Net at the error smaller than the turning point is around 0.5 which is still acceptable.

The larger the error after the turning point, the larger the performance gap between the two networks. This is because the performance of the Normal-Net deteriorates significantly as the

error increases, whereas the performance change of the Noise-Net is small. In the second row, the curves for $M_{\rm cl}$ and the youngest cluster age show the clear gap between the two networks where the accuracy differences between the two networks at the error of 10% are 1.4 and 0.8 dex, respectively.

We measure the error when the performance of the Normal-Net is comparable to the performance of the Noise-Net at $\sigma_{\rm b}$ of 10%. In most cases, the performance of the Noise-Net at a 10% error is similar to the performance of the Normal-Net at an error of 0.4%. In the previous paper, we demonstrated that the Normal-Net provides reliable prediction when the error of the brightest line is smaller than 0.1∼1%. This result shows that the Noise-Net works reliably even with large errors of around 10%.

Figures 3.2 and 3.3 demonstrate that the Noise-Net works significantly better than the Normal-Net after the turning point located at a small error of around 0.04%. It is also noteworthy that the offset issues and the difficulty of predicting ages at large errors in the Normal-Net improves a lot in the Noise-Net.

3.3.3 Individual posterior distribution

In this section, we compare the one-dimensional posterior distributions of the Normal-Net and the Noise-Net on two test models, which have a typical shape of posterior distributions.

The first example is a model at the age of 1.4 Myr in Phase 3, meaning that the expanding shell swept away all of the gas in the natal cloud. As there is only one cluster inside, the age (t) and the age of the youngest cluster ($t_{\rm youngest}$) are the same. In Figure 3.4, we select four $\sigma_{\rm b}$ values (0.01, 0.1, 1 and 10%) and present the corresponding predicted posterior distributions in each column. Each panel shows the posterior distribution of the Noise-Net in the upper sub-panel and the posterior distribution of the Normal-Net in the lower one.

The first column clearly shows that when the error is small (around 0.01%) the posterior distribution of the Normal-Net is narrower than the Noise-Net. The difference is especially apparent in the case of $M_{\rm cl}$ and the star formation efficiency. Both networks predict accurately but the Normal-Net performs slightly better for some parameters like $M_{\rm cl}$ and SFE. However, the situation is already reversed at an error of 0.1%. The posteriors predicted by the Normal-Net are wider than the Noise-Net for all five continuous parameters. The posterior distributions of the Normal-Net become less smooth but the corresponding MAP estimates are still accurate except for the two age parameters now showing bimodal distributions. On the other hand, the posterior of the Noise-Net shows a unimodal distribution similar to the case at 0.01% error and maintains accuracy although the width of the distribution does slightly increase.

As the error increases to 1 and 10%, the posterior distributions predicted by the Normal-Net for all parameters except $N_{\rm cluster}$ and phase change significantly. The posterior distributions are spiky and irregular and for the ages, they now exhibit an offset towards lower values similar to

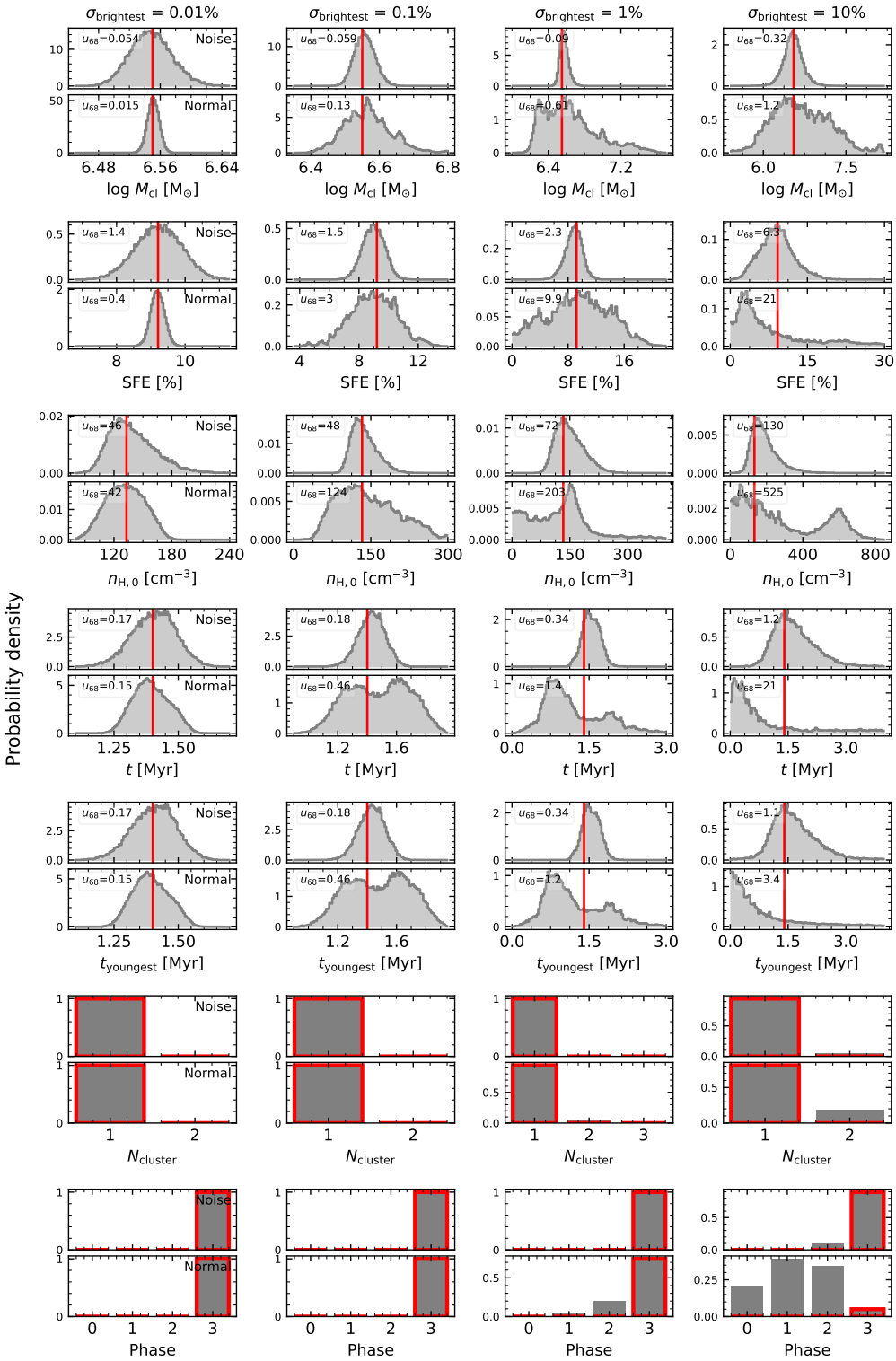

Figure 3.4: Posterior probability distributions (grey histograms) of the seven physical parameters of the first example model estimated by the Noise-Net and the Normal-Net for four different errors of the brightest emission line values (0.01, 0.1, 1 and 10%). Each column corresponds to the result for each error. Please note that the range of the x-axis is different in all columns. Each panel is divided into two: the posterior distribution from the Noise-Net (upper sub-panel) and the posterior distribution from the Normal-Net (lower sub-panel). The red vertical lines indicate the true value of the example model. In the upper left corner, we present the u_{68} value, the width of the distribution.

the one in the violin plots (Figure 3.2). While the network still predicts $N_{\rm cluster}$ well, the phase estimation becomes degenerate, especially when the error is 10%. In contrast, the posterior distribution of the Noise-Net appears almost completely unaffected even when the error increases to 10%. Only the width gradually increases but the MAP estimate remains accurate. At the largest error, the phase prediction does become degenerate but the MAP estimate is still accurate with a 90% probability.

For the second example, we pick a model in a different evolutionary stage. The second model is in Phase 2 and has two clusters due to the stellar feedback of the first-generation cluster being weaker than gravity, resulting in a recollapse. The age of the older generation cluster is 9 Myr and that of the younger one is 1.3 Myr. At the lowest error (0.01%, the first column in Figure 3.5), the posterior distributions are similar to those of the first example. The Noise-Net shows slightly wider and less accurate distributions than the Normal-Net. The difference to the first example is that both networks have a weak degeneracy in the $N_{\rm cluster}$ estimation despite the small error. This behaviour is expected because in Paper 1 we showed that $N_{\rm cluster}$ is the most degenerate parameter and induces degeneracies in the other parameters as well. Moreover, degeneracy is more likely to occur when the target object has more than one cluster. As shown in the age posterior distribution of the Noise-Net, the degeneracy in the $N_{\rm cluster}$ leads to a degenerate prediction in the age of the first cluster.

If the error increases to 0.1%, the posterior estimates of the Normal-Net show wide distributions and the fraction of degenerate predictions increases in $N_{\rm cluster}$ and age, whereas the posterior distributions of the Noise-Net are almost identical to the result for the 0.01% error. From an error of 1% onwards, most of the posterior estimates of the Normal-Net show a wide and skewed distribution similar to the first example, losing accuracy in most parameters. In the previous paper, we demonstrated that the Normal-Net tends to predict extremely young age with a single cluster regardless of the target objects if the error is large (around 10%). This means that the prediction of the Normal-Net at a large error is unreliable and cannot be explained as physically valid alternatives to the target system. Contrary to the Normal-Net, the posterior distributions predicted by the Noise-Net remain accurate and narrow despite the large error. Although the fraction of the degenerate estimates (i.e., $N_{\rm cluster}$ of 1) increases in $N_{\rm cluster}$ and the age if the error is large (10%), the other five parameters still have very accurate predictions with a unimodal distribution.

In many test models including these two examples, we confirmed that the performance of the Noise-Net is inferior to the Normal-Net for errors between 0.01 and 0.1%, which is consistent with the turning point of 0.04% found in the statistical evaluation (Figure 3.3). Additionally, the accuracy and precision of both networks deteriorate with increasing error, but the degree of deterioration is significantly different and the posterior distributions change differently. The Noise-Net does not return extremely wide or skewed posterior distributions like the Normal-Net

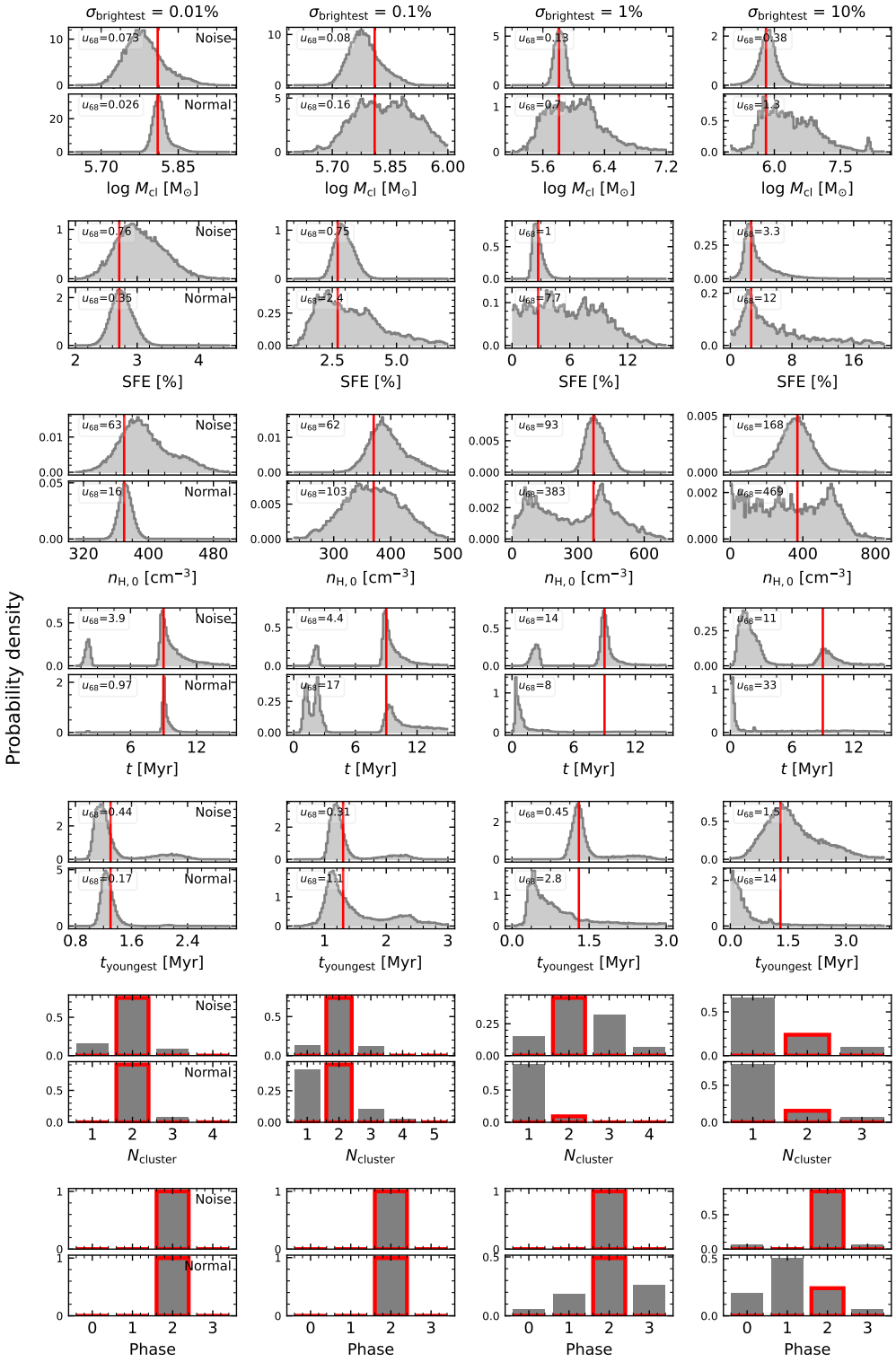

Figure 3.5: The posterior distributions of the second example model. Colour codes and lines are the same as in Figure 3.4.

and maintains an accurate prediction even at the largest error although the fraction of degenerate predictions increases slightly.

3.4 Discussion

3.4.1 Skewness and degeneracy

If network prediction is degenerate, the posterior distribution shows multiple modes, one of which indicates the true value of the test model. In Paper 1, we demonstrated the modes other than the true mode in a degenerate posterior distribution are not wrong but are physically valid alternatives satisfying the same luminosity by re-simulating the posterior models via WARPFIELD-EMP and comparing the re-simulated luminosity with the true luminosity. In this study, we do not re-simulate the posterior estimates. However, based on the results of Paper 1, we regard that a multimodal posterior distribution reflects physically valid degeneracy remaining in the prediction, especially when the multimodality is due to the $N_{\rm cluster}$.

In Paper 1, we showed that the most difficult parameters for our network to predict, even without considering errors, are the number of clusters ($N_{\rm cluster}$) and the age of the first generation cluster (t). We also found that more than 90% of the degeneracy in the posterior distribution was caused by the degeneracy in $N_{\rm cluster}$. If the posterior of $N_{\rm cluster}$ has a multimodal distribution, the posterior distribution of the age also has multiple modes in most of the cases, which lowers the (point estimate) accuracy. In particular, a degenerate $N_{\rm cluster}$ prediction was more common if the target H II region had multiple clusters, so the prediction was usually more accurate for single cluster models than multicluster models.

There are two main reasons why it is difficult to break the degeneracy in the $N_{\rm cluster}$ prediction. One is the biased distribution of our training data. More than 70% of our synthetic H II region models have only one cluster and the age distribution is also biased towards a young age, so that our network usually learns better about young and single cluster models. The second reason is the selection of the emission lines. We use 12 optical emission lines that are tracers of ionized gas and that therefore are mostly influenced by the youngest cluster. The contribution of older clusters to the luminosity of these lines is often insignificant (see Figure 4 in Pellegrini et al. 2020) so it is hard for the network to identify how many old clusters are in the H II region based only on these 12 lines.[1]

As the networks in this study use the same database and the same emission lines, they essentially have the same difficulties. Therefore, we further examine the results presented in Section 3.3 by separating the sample into two groups, single cluster models and multicluster

[1]Supplementing these diagnostics with additional observables sensitive to the older clusters (e.g., broad-band optical or near-IR colours) may allow many of these degeneracies to be broken, but is outside of the scope of our current study.

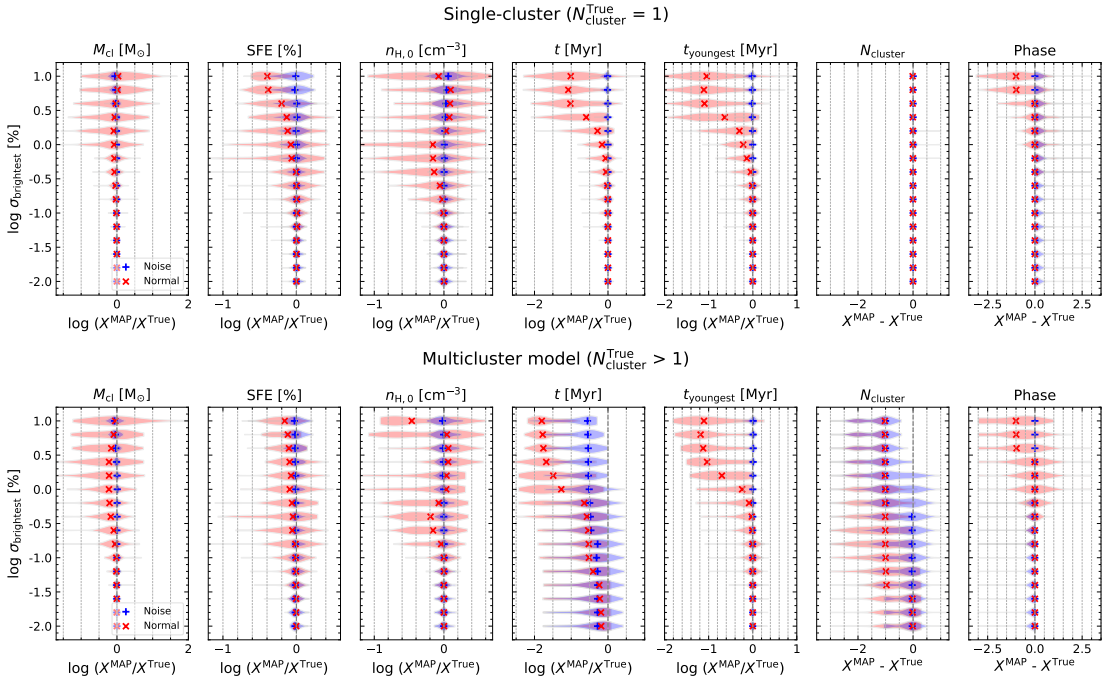

Figure 3.6: We divide the MAP accuracy violin plot in Figure 3.2 into two depending on the true $N_{\rm cluster}$ value of test models: single cluster models (upper panels) and multicluster models (lower panels). 60 models have only one cluster and the other 40 models have more than one cluster. Colour codes and symbols are the same as in Figure 3.2.

models, according to the true $N_{\rm cluster}$ value of each model. In particular, we focus on the large error range ($\sigma_{\rm b} \geq 1\%$), where the Normal-Net and the Noise-Net begin to show a significant difference in average performance and individual predicted posterior distributions. According to the selection criteria in Section 3.3.1, 60 out of our 100 models have only one cluster, whereas the other 40 models have more than one cluster. In Figure 3.6, we subdivide the violin plot of MAP accuracy from Figure 3.2 into single cluster models (upper panels) and multicluster models (lower panels), revealing a clear difference between the predictions of the Normal-Net and the Noise-Net at large error.

Firstly, the prediction results of the Normal-Net are similar between the two groups regardless of the true $N_{\rm cluster}$ value except for the histograms of the $N_{\rm cluster}$ estimation. When the error is larger than 1%, the MAP estimates of the age of the oldest cluster, the age of the youngest cluster, and the phase are always notably offset towards smaller values in both groups. This feature is well revealed in the example posterior distributions (Figures 3.4 and 3.5) by the notable skewness. The difference in the $N_{\rm cluster}$ histograms occurs, because the MAP estimates of the Normal-Net for $N_{\rm cluster}$ are always 1 when the error is large. Consequently, the histogram is close to 0 for single cluster models, because the true value is 1, but for the multicluster models, the MAP estimates are always smaller than the true value. In Paper 1, we also confirmed that the Normal-Net returns similar posterior distributions regardless of the target objects if the error is large, meaning that the Normal-Net returns wrong estimates when subject to large errors. This

is different to a true degeneracy in the prediction that shows the physically valid alternatives satisfying the same observation.

On the other hand, the Noise-Net shows different performance for N_{cluster} and the age of the first cluster depending on the true N_{cluster} value. The Noise-Net performs very well for all seven parameters in the case of single cluster models even at the large error of 10%. The Noise-Net predicts the age of the youngest and oldest cluster and the phase well unlike the Normal-Net which shows systematic offsets. In the case of multicluster models, however, the Noise-Net also exhibits an offset towards lower values in the oldest cluster age and N_{cluster}, but it recovers the age of the youngest cluster and phase well. In the case of the N_{cluster} prediction, the Noise-Net also determines the MAP value as 1 in most cases regardless of the target model when the error is large ($\sigma_{\mathrm{b}} > 1\%$). While the estimates of the oldest cluster age are also shifted towards lower values for the multicluster models even for the smaller error range, the magnitude of the shift is smaller than that of the Normal-Net. When the error is small, both networks exhibit a shift of less than 0.5 dex, but when the error is large the Normal-Net offset increases to about 2 dex, whereas the Noise-Net shift remains at around 0.5 dex.

Based on Figure 3.6 and the examples of the predicted posterior distributions (Figures 3.4 and 3.5), the likelihood of the Noise-Net to return degenerate predictions increases with the error. However, unlike the Normal-Net the Noise-Net does not return wrong predictions. As previously mentioned, due to the intrinsic difficulty in predicting N_{cluster} in our cINNs, the degeneracy in the Noise-Net is mostly caused by degenerate N_{cluster} predictions. As the prediction of the Noise-Net is degenerate unlike the Normal-Net, the estimation of the oldest cluster age by the Noise-Net is not extremely offset towards younger models but shifted close to the true value of t_{youngest}. In addition, the Noise-Net always returns the correct information about the youngest cluster and its evolutionary phase for both single cluster models and multicluster models.

While degenerate predictions are less accurate in terms of finding the unique true value, they are still meaningful in that they suggest other possible solutions that satisfy the same observation. As most of the degeneracy lies within the N_{cluster} prediction, we can filter out the correct posterior estimates if we can constrain the N_{cluster} by another method or observation. We expect that by using more unbiased training data or including emission lines that are contributed by old clusters, we can improve the degeneracy in our networks.

3.4.2 Pros and cons of Noise-Net and Normal-Net

In Section 3.3, we directly compared the performance of the Noise-Net and the Normal-Net. In this section, we compare the methodological differences between the two networks and discuss which network is more practical when applied to real observational data based on the results we presented. The two networks differ in the method of training the network and sampling the posterior estimates. The advantages and disadvantages of the two networks that arise from the

methodological aspect can be summarised as follows.

In the case of the Normal-Net, we train the network using pure training data. To consider the observational error in the posterior distribution, $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\sigma})$, we have to introduce the additional Monte Carlo-based marginalisation method (Eq. 3.9). We generate a sufficient number of perturbed luminosity sets by adding random Gaussian noise, assuming that the luminosity errors follow a Gaussian profile. As this method uses a trained network, we are free to choose the error profile to marginalise in this method. It is relatively easy to apply different profiles such as an asymmetric error profile or consider physical relationships between emission lines if needed depending on the object of interest. However, the disadvantage of this method is that it requires a lot of posterior estimates to fully sample the posterior distribution. Generating perturbed luminosity sets (N_p) and sampling the latent variables for each luminosity set (N_z), we generate $N_\mathrm{p} \times N_\mathrm{z}$ posterior estimates to make one posterior distribution. As it is important to use a sufficiently large N_p, we sample about 100,000 posterior estimates per one test model in this paper. This increases the overall computation time for posterior prediction and data analysis when using the Normal-Net.

On the other hand, the methodological advantages and disadvantages of the Noise-Net are opposite to the case of the Normal-Net. We can obtain a full posterior distribution with only a small number of posterior estimates because the Noise-Net does not require a marginalisation process (see Table 3.2). In other words, the Noise-Net is advantageous in terms of the computation time for sampling the posterior estimates and processing the data. The disadvantage of the Noise-Net is that we have to set the range of the luminosity error and the error profile in advance when training the network. Consequently, if we want to apply error values that are smaller or larger than the trained ranges or use a different error profile, we have to train a new network. In this paper, for simplicity, we assume that the errors of all 12 emission lines are independent to each other when using both Noise-Net and Normal-Net. If we want to consider any covariance between the lines, then we need to train a new Noise-Net that applies the relations between the lines during the training.

Methodologically, Noise-Net and Normal-Net have advantages and disadvantages that are opposite to each other. However, based on the results in Section 3.3, which network has better performance depends on the amount of error. According to Figure 3.3, the Noise-Net begins to perform better than the Normal-Net on average when the error of the brightest emission line (σ_b) is larger than 0.04%. At σ_b of 0.1%, the performance difference between the two networks is already noticeable and this performance gap widens as the error increases. This means that, if we apply our cINNs to real observations, the Noise-Net is a more suitable method in terms of performance when the observation error corresponding to the σ_b in our experiment is not less than 0.04%.

To determine which network is more practical considering the error of real observations,

Table 3.3: The average performance of the Noise-Net and the Normal-Net for 100 test models when σ_b is 0.1%. The first value in each entry shows the performance of the Noise-Net whereas the second value is that of the Normal-Net. The MAP accuracy denotes the deviation between the MAP estimates and the true value, while the accuracy in the second row refers to the deviation of all posterior estimates from the true value. We use a linear deviation for $N_{\rm cluster}$ and phase and use a logarithmic deviation for the other five parameters. The last row indicates the average width of the posterior distribution. The listed values are the same as the data points in Figure 3.3 at σ_b of 0.1%.

Performance index at $\sigma_b = 0.1\%$ (Noise-Net / Normal-Net)	log $M_{\rm cl}$	SFE	$n_{\rm H,0}$	t	$t_{\rm youngest}$	$N_{\rm cluster}$	Phase
RMS of MAP accuracy	0.04 / 0.14	0.055 / 0.15	0.073 / 0.14	0.41 / 0.54	0.05 / 0.16	0.46 / 0.77	0.52 / 0.1
RMS of accuracy	0.058 / 0.19	0.085 / 0.19	0.11 / 0.19	0.32 / 0.45	0.074 / 0.17	0.55 / 0.71	0.52 / 0.6
Median of u_{68}	0.076 / 0.18	1 / 2.2	78 / 180	3.5 / 5.3	0.22 / 0.49	0.11 / 0.14	0.098 / 0.1

we refer to the H II region catalogue of Santoro et al. (2022) based on the PHANGS-MUSE survey (Emsellem et al. 2022). In this catalogue, Santoro et al. (2022) lists observations for about 23000 H II regions in 19 nearby spiral galaxies including the fluxes and corresponding errors for a set of strong lines observable within the wavelength range 4850–7000Å. In our experiment in Section 3.3, we use the error of the brightest line, i.e., the smallest error among the 12 emission lines by definition following Eq. 3.10. We confirm that, in the H II region catalogue, both the brightest line and the line with the smallest flux error in per cent unit is always the Hα emission line. This is because the nebulae with emission lines brighter than the Hα were usually located outside of the H II region area in the BPT diagram and excluded in the catalogue. The error of the Hα line in the catalogue is $0.302 \pm 0.294\%$ on average and the minimum and maximum values are 0.011 and 3.08%, respectively. 98% of the H II regions have an Hα error larger than 0.04% and 87% of them have an Hα error larger than 0.1%. In other words, for most of the H II regions in this survey, we expect the Noise-Net to perform better than the Normal-Net.

Based on the typical error of the real observations, we list the average performance (two accuracy indices and the precision index) at a σ_b of 0.1% presented in Figure 3.3, in Table 3.3 to quantitatively examine the performance gap between the two networks. The first entry denotes the performance of the Noise-Net while the second lists that of the Normal-Net. This demonstrates that the Noise-Net performs better than the Normal-Net in all indices for all seven parameters. Excluding the phase and $N_{\rm cluster}$, the accuracy gap between the two networks is about 0.1 dex in each parameter. This suggests that for the 85 per cent of H II regions in Santoro et al. (2022)'s catalogue, the Noise-Net will at least give better results within this performance gap. As the average Hα error of the catalogue is 0.3%, the average performance gap will be larger than Table 3.3 based on Figure 3.3.

In conclusion, Normal-Net and Noise-Net have opposite advantages and disadvantages in terms of methodology, but considering the range of the luminosity errors in practice, we expect that the Noise-Net is a better choice than the Normal-Net. We only gave one survey example but as we have the fiducial error to determine which method performs better, users can choose their best option considering their observational data. If it is reasonable to use identical error

profiles for different H II regions of interest and the error of the brightest line (usually the Hα) is sufficiently larger than 0.04 per cent, we propose that Noise-Net would be a more robust method.

3.4.3 Error larger than the training range

The Noise-Net learns about the observation error within the fixed range during the training. As mentioned in Section 3.2.2, we trained the Noise-Nets in this paper on errors ranging from 0.001 to 31.6 % and applied the same error range for all 12 emission lines. However, in our experiment in Section 3.3, there are many cases where the luminosity error exceeds the maximum training error (i.e., $\sigma_{\rm max}$) of 31.6%, especially when the $\sigma_{\rm b}$ is larger than 1%. When $\sigma_{\rm b}$ is 1%, 28 of 100 test models have an emission line whose luminosity error is larger than $\sigma_{\rm max}$. When $\sigma_{\rm b}$ is 10%, at least one emission line has an error larger than $\sigma_{\rm max}$ in all test models. While the Noise-Net can somehow handle errors larger than the maximum it has learned, we need to examine if the Noise-Net processes the error properly.

We hypothesise that the Noise-Net either handles the large errors that it has never learned properly or simply self-clips large errors to the training limit ($\sigma_{\rm max}$). To investigate this hypothesis, we re-sample the posterior estimates for the 100 test models for 16 different $\sigma_{\rm b}$ values as we did in Section 3.3, but clip the luminosity error to $\sigma_{\rm max}$ if it is larger than the maximum error. We analyse the newly obtained posterior distributions in the same manner as in Section 3.3.1 and compare the result with the original outcome without error clipping from Section 3.3.

Figure B.1 shows the difference between the unclipped original result (i.e., blue curve in Figure 3.3) and the new result with error-clipping. Differences begin to appear around a $\sigma_{\rm b}$ value of 1%, but the deviation is very small. Even when $\sigma_{\rm b}$ is 10%, the deviation of the two results is less than 0.02 dex in accuracy indices. The difference in median u_{68} value is also very small considering the physical unit of the parameter. The difference between the original result without clipping and the result after clipping large errors to the maximum value is almost negligible.

In addition to the average performance differences, we present one example to investigate if there is any noticeable difference in the posterior distributions for individual test models. In Figure 3.7, we present the predicted posterior distributions for one test model at a $\sigma_{\rm b}$ of 10%. In this case, 5 of 12 emission lines have a luminosity error larger than the maximum value ([O II] 3726Å, [O I] 6300Å, [S II] 6716Å, [S II] 6731Å and [S II] blend 6720Å). For the posterior distributions in the left panels, we use the large errors without clipping and for the right panels, we clip the 5 large error values to the maximum error of 31.6%. As shown in Figure 3.7, the two distributions are almost the same. While the left distributions without clipping appear a bit wider based on the u_{68} values, the magnitude of the deviation is negligible.

Figures B.1 and 3.7 demonstrate that the Noise-Net handles errors larger than its training range by self-clipping the large errors close to the upper limit of the training range. Accordingly,

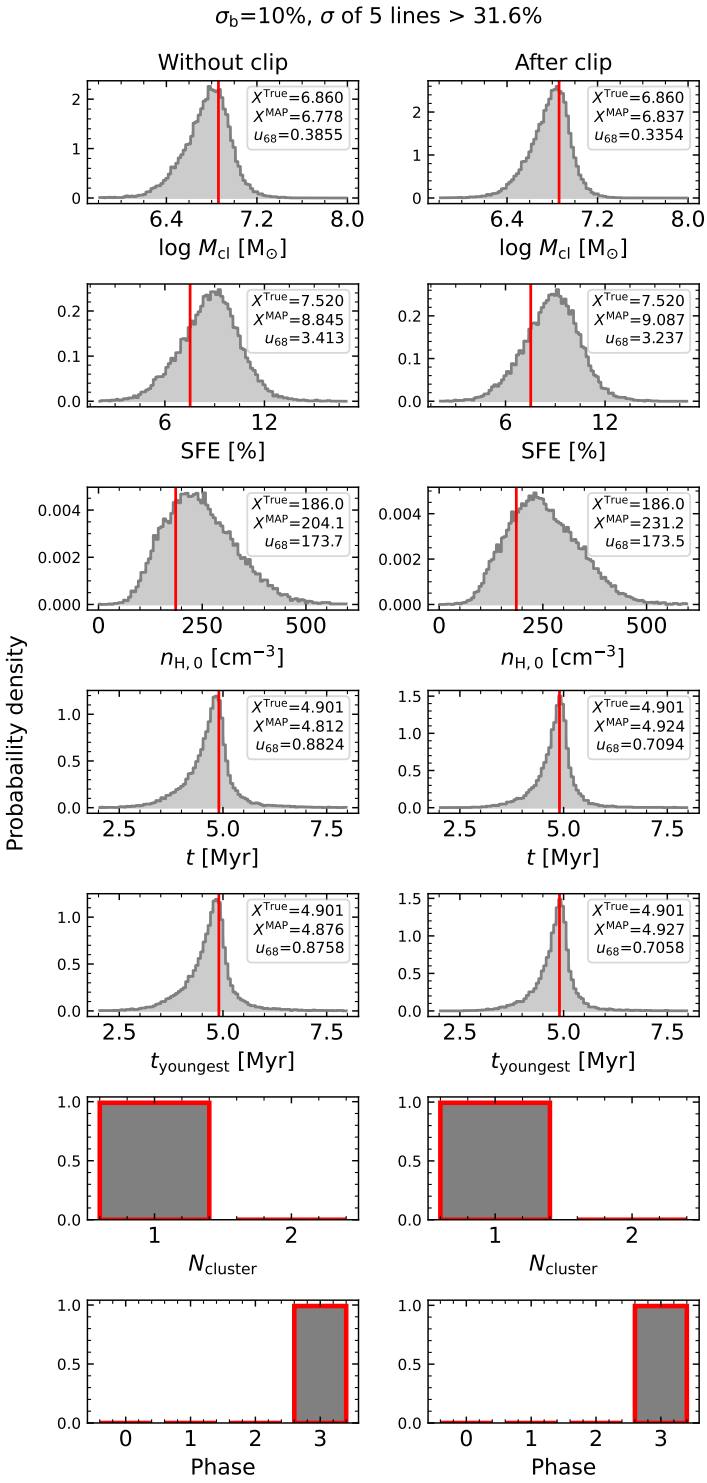

Figure 3.7: Predicted posterior probability distributions (grey histograms) for the seven physical parameters for one test model estimated by the Noise-Net at σ_b of 10%. For this model, the luminosity errors of 5 emission lines are larger than the maximum error of the training range (31.6%). The left panels show the posterior distribution using the luminosity errors without clipping, whereas the right panels present the posterior distribution after clipping the large errors to the maximum error. The red vertical lines indicate the true value of the model. In the upper right corner the true value, the MAP estimate, and the u_{68} value of the posterior distribution are listed.

the direct comparison between the Noise-Net and the Normal-Net may not be fair because the Noise-Net treats large errors as a smaller value by itself, while the Normal-Net does not. Therefore, we also re-sample the posterior estimates using the Normal-Net after clipping the large errors to the maximum error value and re-compare the new result of the Normal-Net with the result of the Noise-Net. We confirm that there is no change in the overall results presented in Section 3.3. This is because the new result from the Normal-Net as well is almost the same as the original result of the Normal-Net without clipping. In Figure B.2, we compare two results from the Normal-Net: unlcipped original result (red curve in Figure 3.3) and the new result after clipping the large errors. Similar to the case of the Noise-Net shown in Figure B.1, the recognizable difference begins to appear around a $\sigma_{\rm b}$ of 1%. The differences between unclipped and clipped results are less than 0.03 dex in most cases except for the $M_{\rm cl}$ where the maximum difference is around 0.07 dex, which is still small enough. This implies that there is no significant difference in the posterior distribution from the Normal-Net at large error range (i.e., larger than few tens per cent).

We propose the following reasons to explain the small differences in the results of the Normal-Net. Firstly, at large errors, the Normal-Net always returns a wide and skewed posterior distribution as shown in Figures 3.4 and 3.5. As mentioned in Section 3.4.1, the Normal-Net tends to predict similar posterior distributions at the largest error, regardless of the characteristics of the target model. The distribution is already as wide as the entire training range in Figure 3.1, so even if we use a larger error, the posterior distribution does not widen any further or become more skewed. Secondly, in the marginalisation process for the Normal-Net, we randomly perturb the luminosity by sampling from a Gaussian noise profile following Eq. 3.3. However, we clip the perturbed luminosity to a minimum value of 1 to avoid unrealistic estimations.

Thirdly, when using the posterior distribution for the analysis, we exclude all unrealistic posterior estimates such as negative densities or ages larger than 100 Myr. We define the acceptance rate ($f_{\rm physical}$) as the fraction of posterior estimates that are not regarded as unrealistic over the total number of the posterior samples. In Figure 3.8, we compare the median acceptance rate for 100 test models of the two networks as a function of the luminosity error ($\sigma_{\rm b}$). Figure 3.8 clearly shows that the acceptance rate of the Normal-Net significantly decreases with increasing error whereas the Noise-Net maintains an acceptance rate close to 100%. The acceptance rate of the Normal-Net begins to decrease at an error of 0.25% ($\log \sigma_{\rm b} = -0.6$) and decreases to 34% for an error of 10%. The acceptance rate of the Noise-Net is always larger than 99.5%. This result explains why the posterior distribution of the Normal-Net does not change much at large error. Moreover, this demonstrates that the Noise-Net provides physical estimates even when the error is large, although degeneracy remains in the posterior distribution, as discussed in Section 3.4.1.

In summary, we find that the Noise-Net clips large errors outside the training range by itself but this does not change the overall trend between the Noise-Net and the Normal-Net shown in

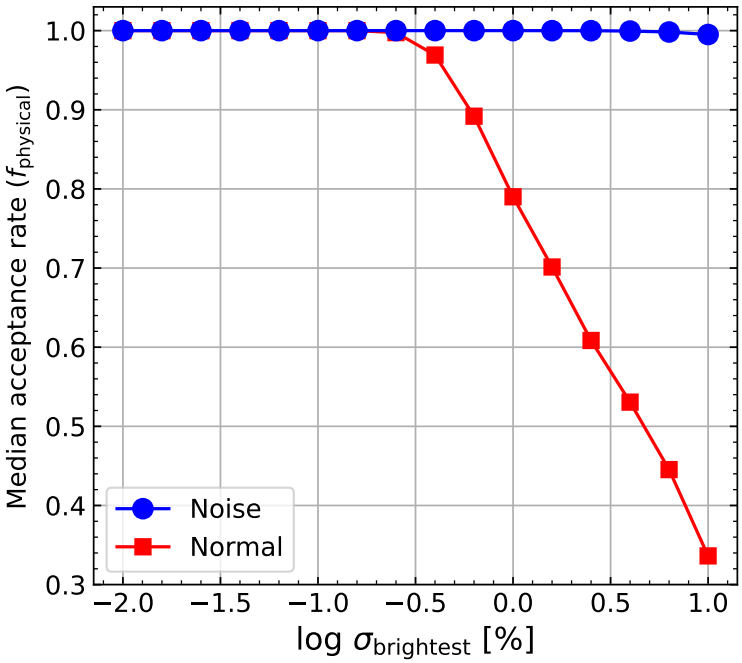

Figure 3.8: Median acceptance rate for 100 test models of the Noise-Net and the Normal-Net as a function of the luminosity error of the brightest emission line. The acceptance rate is the fraction of the posterior estimates that are not excluded as physically incorrect or extremely extrapolated values (e.g., negative age or star formation efficiency or age larger than 100 Myr) over the total number of the sampled posterior estimates.

Section 3.3. Because the posterior distributions of the Normal-Net do not change significantly at large error range. Considering the behaviour of the Noise-Net and Normal-Net at the large error range, it is better not to use our cINNs to analyze objects with too large luminosity errors above a few tens per cent.

3.4.4 Outperformance and saturation of the Noise-Net

In Figure 3.3, we showed that the performance of the Noise-Net and Normal-Net change differently as a function of the observational error. For example, the RMS deviation of the Normal-Net steadily increases from the minimum error of 0.01% to the maximum error of 10%. In contrast, the RMS deviation of the Noise-Net gradually increases after 0.1%, maintaining a small value in the small error range. Therefore, there is a turning point in which the performance of the Noise-Net overtakes the Normal-Net.

The turning point varies slightly depending on the parameter and performance index, but as mentioned in Section 3.3.2, it mainly falls around a value of 0.025% $\sim$ 0.04% (log σ_b: -1.6 $\sim$ -1.4). The performance of the two networks is similar at the turning point, but according to Table 3.3 and Figure 3.3, the performance difference between the two networks is evident even at the error of 0.1%. The larger the error, the larger the gap between the two networks.

We speculate that the reason why the Noise-Net outperforms the Normal-Net at large error is because of the differences in training methods. In the case of the Normal-Net, the network learns from the same training data for each training epoch repeatedly during the training. On the other hand, the Noise-Net learns about various luminosity and error values from the identical training models because the perturbations are randomly sampled at every epoch. This implicitly increases the number of training examples for the Noise-Net even if the number of actual training models is the same as in the normal training. Furthermore, as the luminosity of the training model is perturbed by the randomly sampled noise during the noise training, the Noise-Net learns more diverse cases of degeneracy and wider training distributions. For these reasons, the Noise-Net performance deteriorates less than that of the Normal-Net at a larger error.

On the other hand, the performance of the Noise-Net does not change as much as the Normal-Net when the observational error decreases. In Figure 3.3, the performance of the Noise-Net appears to almost saturate at around 0.1% and does not decrease much further with decreasing error. This also can be partly explained by the difference in the training methods. No matter how small the errors sampled during the noise training become, the Noise-Net has never seen a version of the data as precise as the pure training data. Therefore, the predicted posterior distributions may have a basic degeneracy even at the minimum error, showing the saturation in Figure 3.3. However, the Noise-Net starts to saturate at around 0.1%, which is a fairly large value considering the minimum training error of 0.001%. The reason for the early saturation has not become clear within the scope of this study, but we will examine this trend in further experiments such as changing the profile of the error in the noise training in future works.

3.5 Summary

In this paper, we introduce a new type of cINN, the Noise-Net, that predicts physical parameters ($\mathbf{x}$) of H II regions from the emission-line luminosities ($\mathbf{y}$), considering the uncertainty of the observations ($\boldsymbol{\sigma}$, emission-line luminosity error). In our recent paper (Paper 1), we first introduced a cINN that predicts the seven physical parameters of the H II region from the luminosity of 12 optical emission lines. The type of cINN used in Paper 1 is Normal-Net which estimates posterior distributions of the physical parameters conditioned only on the luminosities ($p(\mathbf{x}|\mathbf{y})$), but we presented a method of considering luminosity errors in the Normal-Net through a modification of the posterior sampling procedure (Section 3.2.5) and showed the performance of the Normal-Net as a function of the luminosity error. The Noise-Net, newly introduced in this paper, always reflects the luminosity error in the prediction of the parameters by using both luminosities and corresponding errors as an input of the network ($p(\mathbf{x}|\mathbf{y}, \boldsymbol{\sigma})$). In this paper, we introduce the Noise-Net and its training method (i.e., noise training) and compare the performance of the Noise-Net with the Normal-Net.

For the Noise-Net to learn the influence of observational errors, the training method of the Noise-Net is slightly different to the Normal-Net. At each training epoch, we first randomly sample the errors of each emission-line luminosity from the prescribed probability distribution. In this paper, we sample the error in the logarithmic scale from the uniform distribution (Eq. 3.2) with minimum and maximum errors of 0.001% and 31.6%, respectively. Next, we perturb the luminosities of the training data by adding the random Gaussian noises based on the sampled errors. Due to these two steps processed on the fly during the training, the Noise-Net learns about the different training data every epoch whereas the Normal-Net learns about the pure training data repeatedly.

We use synthetic H II region models produced by WARPFIELD-EMP (Pellegrini et al. 2020) to train the cINNs because it is hard to collect lots of well-interpreted real data required for the training. WARPFIELD-EMP is a pipeline that calculates emission from isolated massive star-forming clouds using CLOUDY (Ferland et al. 2017) and POLARIS (Reissl et al. 2016, 2019), based on the one-dimensional semi-analytic stellar feedback code WARPFIELD (Rahner et al. 2017). The database of WARPFIELD-EMP models used in this paper is the same as the database introduced in Paper 1. The database consists of 505,748 synthetic H II region models evolved from 10,000 initial star-forming clouds. We use 90% of the models to train the network and use the rest (test set) to evaluate the network performance.

In this paper, we present two cINNs, one Noise-Net and one Normal-Net, that predict seven physical parameters from the information on 12 emission lines (see Table 3.1) trained on the WARPFIELD-EMP models. To compare the performance of the Noise-Net and the Normal-Net as a function of the luminosity error, we evaluate the accuracy and precision of two cINNs at 16 different levels of luminosity error using the error of the brightest emission line ($\sigma_{\rm b}$) as a representative error. Our main results of the comparison between the Noise-Net and the Normal-Net are the following:

1. As the error increases, the performance of both networks gradually deteriorates, but the two networks show different trends as a function of the error. The Normal-Net predicts more accurately and precisely than the Noise-Net when the error is very small ($\sigma_{\rm b} \sim 0.01\%$). However, the performance of the Normal-Net steadily degrades significantly compared to the Noise-Net. On the other hand, the Noise-Net maintains good performance with increasing error at a small error range. The performance of the Noise-Net slowly and gradually degrades at large errors.

2. From a certain point (i.e., turning point), the Noise-Net outperforms the Normal-Net and the larger the error, the larger the gap between the Noise-Net and the Normal-Net. The turning point occurs at an error of around 0.025–0.04%.

3. The Noise-Net estimates parameters with sufficient accuracy and precision even when the

luminosity error is large. The performance of the Noise-Net when the error of the brightest line is 10% is similar to that of the Normal-Net when the error of the brightest line is only 0.4%.

4. The Noise-Net predicts parameters much better than the Normal-Net even when the error of the brightest line is 0.1%. Considering the luminosity error of the observed H II regions from the PHANGS-MUSE survey (Emsellem et al. 2022) as an example, the Hα luminosity error of 0.1% is a small enough value because 89% of the H II regions had the Hα luminosity error larger than 0.1%. This means that the Noise-Net is a more appropriate tool when applied to real observations that have a similar level of uncertainty to the PHANGS-MUSE survey.

5. As the error increases, degeneracy becomes common in the posterior distribution of the Noise-Net. In the case of the Normal-Net, the posterior estimates at a large error are not degenerate like the Noise-Net but are unphysical. The fraction of the unphysical posterior estimates increases significantly as a function of the luminosity error in the case of the Normal-Net, whereas the fraction for the Noise-Net is almost zero even at the large error, meaning that the Noise-Net always provides physically valid estimates although the degeneracy remains.

6. The Noise-Net learns about errors within a given training range (i.e., 0.001 – 31.6 %). If the Noise-Net receives an error larger than the training range, the Noise-Net self-clips the large errors close to the maximum of the training range and provides a posterior distribution using the clipped error.

The Noise-Net, newly presented in this paper, predicts parameters accurately and precisely even when the observation error is large. Our results of comparison between the Noise-Net and the Normal-Net will help select an appropriate network type according to the observed luminosity errors. Based on our results, we suggest utilising the Noise-Net if the error of the brightest emission line (e.g., the Hα line) is 0.1% or more.

4

Spectral classification of young stars using conditional invertible neural networks - I. Introducing and validating the method

This chapter is based on the paper Kang et al. (2023, *submitted*) submitted to Astronomy & Astrophysics in 2023. I carried out network construction and training and most of the data analysis. The text is majorly written by me but some sections are written by co-authors. Leonardo Testi developed the interpolation pipeline to easily generate numerous Phoenix models and wrote Section 4.3.1. Victor F. Ksoll produced the databases of Phoenix models using three different Phoenix grids: Settl, NextGen, and Dusty and carried out resimulations of posterior models and wrote Sections 4.2 and 4.3.2 and resimulation parts in Section 4.5. Dominika Itrich prepared the catalogue of Class III template stars and wrote Section 4.4.

Abstract

We introduce a new deep learning tool that estimates stellar parameters (such as effective temperature, surface gravity, and extinction) of young low-mass stars by coupling the Phoenix stellar atmosphere model with a conditional invertible neural network (cINN). Here we discuss cINNs trained on three different Phoenix grids: Settl, NextGen, and Dusty. They allow us to infer the posterior distribution of each stellar parameter from the optical spectrum. Our networks are time-efficient tools applicable to large amounts of observations. Evaluating the performance of these cINNs on unlearned Phoenix synthetic spectra, we confirm that the cINNs estimate the considered stellar parameters almost perfectly. We validate our approach further by applying the cINNs to the spectra of 36 Class III template stars with well-characterised stellar parameters and find good agreement with deviations of at most 5–10 per cent. The cINNs perform slightly better for earlier-type stars than for later-type stars like late M-type stars, but we conclude that estimations of effective temperature and surface gravity are reliable for all spectral types within the network's training range. Among the three networks, we recommend using the cINN trained

on the Settl library (Settl-Net), as it provides the best performance across the largest range of temperature and gravity.

4.1 Motivation

In star-forming regions, it is massive stars that influence the surrounding environment energetically and dynamically during their short lifetime, but the majority of stars formed in star-forming regions are low-mass stars similar to or less than the solar mass. These low-mass stars are not only the most numerous in the star-forming region (Bochanski et al. 2010) but also account for about half of the total stellar mass (Kroupa 2002; Chabrier 2003). Living longer than massive stars, these low-mass stars still remain in the pre-main-sequence phase even when the massive stars are dead. These young low-mass stars provide important information for studying stellar evolution and planet formation.

Stellar parameters (e.g., effective temperature, surface gravity, luminosity, etc.) are estimated from photometric or spectroscopic data by various methods. These methods are usually based on characteristic spectral features that vary depending on the type of stars. Therefore, it is important to adopt the appropriate method to the star under consideration and to the observed wavelength range.

As the volume of accumulated observations ever-expand in recent days, it has become important to develop time-efficient tools that analyse large amounts of data in a faster and more consistent way. This is why artificial neural networks (NNs; Goodfellow et al. 2016) have been utilised in many astronomical fields these days. For instance, NNs have been used to predict physical parameters (e.g., Fabbro et al. 2018; Ksoll et al. 2020; Olney et al. 2020b; Kang et al. 2022) or to efficiently analyse images such as identifying structures (e.g., Abraham et al. 2018) and exoplanets (e.g., de Beurs et al. 2022) or classifying observations (e.g., Wu et al. 2019; Walmsley et al. 2021; Whitmore et al. 2021). In this study, we develop NNs that can efficiently analyse numerous spectra in the optical wavelength range of young low-mass stars. We prepare our networks to analyse VLT/MUSE observations adopting the wavelength coverage and spectral resolution of MUSE. In the follow-up paper, we will apply our tool to the spectra of young stars in the Carina Nebula observed with VLT/MUSE.

We adopt conditional invertible neural network (cINN) architecture developed by Ardizzone et al. (2021). Estimating physical parameters from observed measurements is a non-trivial task. As the information we obtain from observation is limited due to the information loss during the forward process (i.e., translation from physical systems to observations), different physical systems can be observed similarly or almost identically, which we call a degenerate system. cINN architecture is specialised in solving the inverse problem of the degenerate system (i.e., from observations to physical systems). In particular, cINN has its own advantage in that

cINN always provides a full posterior distribution of the physical system without any additional computations. In astronomy, the cINN approach has been used so far to characterise the internal properties of planets (Haldemann et al. 2022), analyse photometric data of young stars (Ksoll et al. 2020), study emission lines in H II regions (Kang et al. 2022), or infer the merger history of galaxies (Eisert et al. 2023).

The cINN architecture adopts a supervised learning approach that learns the hidden rules from a number of well-labelled data sets of physical parameters and observations. As it is difficult to collect a sufficient number of well-interpreted real observations, synthetic observations have been usually used instead to generate enough training data. In this study, we utilise Phoenix stellar atmosphere libraries (e.g., Allard et al. 2012; Husser et al. 2013; Baraffe et al. 2015) to train cINNs. Selecting Settl, NextGen, and Dusty Phoenix libraries, we introduce three cINNs (Settl-Net, NextGen-Net, and Dusty-Net) trained on each library.

A few studies have developed NNs to analyse low-mass stars from photometric or spectroscopic data (e.g., Ksoll et al. 2020; Olney et al. 2020b; Sharma et al. 2020b). For example, Ksoll et al. (2020) developed a network using cINN architecture to estimate the physical parameters of individual stars from HST photometric data and Olney et al. (2020b) used a convolutional neural network (CNN) to estimate physical parameters (e.g., effective temperature, surface gravity, and metallicity) from near-infrared spectra observed with Apache Point Observatory Galactic Evolution Experiment (APOGEE) spectrograph. Sharma et al. (2020b) used CNN as well to diagnose the optical spectra of stars in a wide range of spectral types but their network only estimates the spectral type of the stars, not the other physical parameters. On the other hand, in this paper, our networks directly estimate the stellar parameters from the optical spectrum of low-mass stars, including the stars in both the main sequence and pre-main sequence phases. Moreover, our network provides a posterior distribution by adopting cINN architecture, which is useful to study the degeneracy between parameters.

In this paper, we focus on validating the performance of three cINNs. We evaluate our networks not only on Phoenix synthetic observations but also on real spectra of 36 young low-mass stars to investigate how well our cINNs work on real observations. These stars are template stars in the Class III phase, well-interpreted by literature (e.g., Manara et al. 2013, 2017; Stelzer et al. 2013).

The paper is structured as follows. In Section 4.2, we describe the structure and principles of cINN and explain implementation detail on the machine learning side. In Section 4.3, we introduce our three networks and three training databases. In the following section (Section 4.4), we describe the Class III template stars used in this paper. Our main results are in Section 4.5. We validate our networks using synthetic Phoenix spectra and 36 template stars. We not only evaluate the parameter prediction power of the cINN but also check whether the predicted parameters do explain the input observations. Section 4.6 present which parts of the spectrum

cINN relies mostly upon. In Section 4.7, we investigate the gap between Phoenix synthetic spectra and real observations. We summarise the results in Section 4.8.

4.2 Conditional invertible neural network

The conditional invertible neural network (cINN; Ardizzone et al. 2019c,d) is a deep learning architecture that is well suited for solving inverse problems, i.e. tasks where the underlying physical properties $\mathbf{x}$ of a system are to be recovered from a set of observable quantities $\mathbf{y}$. In nature, recovering the inverse mapping $\mathbf{x} \leftarrow \mathbf{y}$ is often challenging and subject to degeneracy due to an inherent loss of information in the forward mapping $\mathbf{x} \rightarrow \mathbf{y}$, such that multiple sets of physical properties may appear similar or even entirely the same in observations.

To tackle these difficulties, the cINN approach introduces a set of unobservable, latent variables $\mathbf{z}$ with a known, prescribed prior distribution $P(\mathbf{z})$ to the problem in order to encode the information that is otherwise lost in the forward mapping. The cINN achieves this by learning a mapping f from the physical parameters $\mathbf{x}$ to the latent variables $\mathbf{z}$ *conditioned* on the observations $\mathbf{y}$, i.e.

$$f\left(\mathbf{x}; \mathbf{c} = \mathbf{y}\right) = \mathbf{z}, \tag{4.1}$$

capturing all the variance of $\mathbf{x}$ not explained by $\mathbf{y}$ in $\mathbf{z}$, while enforcing that $\mathbf{z}$ follows the prescribed prior $P(\mathbf{z})$. Given a new observation $\mathbf{y}'$ at prediction time, the cINN can then query the encoded variance by sampling the latent space according to the known prior distribution and by making use of its invertible architecture run in reverse to estimate the full posterior distribution $p\left(\mathbf{x}|\mathbf{y}'\right)$ as

$$p\left(\mathbf{x}|\mathbf{y}'\right) \sim g\left(\mathbf{z}; c = \mathbf{y}'\right), \text{ with } \mathbf{z} \propto P\left(\mathbf{z}\right), \tag{4.2}$$

where $f^{-1}(\cdot, \mathbf{c}) = g(\cdot, \mathbf{c})$ represents the inverse of the learned forward mapping for fixed condition $\mathbf{c}$. In practice, $P(\mathbf{z})$ is usually prescribed to be a multivariate normal distribution with zero mean and unit covariance, and the dimension of the latent space is chosen to be equal to that of the target parameter space, i.e. $\dim(\mathbf{z}) = \dim(\mathbf{x})$.

The invertibility of the cINN architecture is achieved by chaining so-called (conditional) affine coupling blocks (Dinh et al. 2016b). Each of these blocks performs two complementary affine transformations on the halves $\mathbf{u}_1$ and $\mathbf{u}_2$ of the block input vector $\mathbf{u}$, following

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{u}_1 \odot \exp\left(s_2(\mathbf{u}_2, \mathbf{c})\right) \oplus t_2(\mathbf{u}_2, \mathbf{c}) \\ \mathbf{v}_2 &= \mathbf{u}_2 \odot \exp\left(s_1(\mathbf{v}_1, \mathbf{c})\right) \oplus t_1(\mathbf{v}_1, \mathbf{c}). \end{aligned} \tag{4.3}$$

As the equation shows, these two transformations are easily inverted given the halves $\mathbf{v}_1$, $\mathbf{v}_2$ of

the output vector $\mathbf{v}$ according to

$$\begin{aligned}\mathbf{u}_2 &= (\mathbf{v}_2 \ominus t_1(\mathbf{v}_1, \mathbf{c})) \odot \exp(-s_1(\mathbf{v}_1, \mathbf{c})) \\ \mathbf{u}_1 &= (\mathbf{v}_1 \ominus t_2(\mathbf{u}_2, \mathbf{c})) \odot \exp(-s_2(\mathbf{u}_2, \mathbf{c})) .\end{aligned} \tag{4.4}$$

In both sets of Equations (4.3) and (4.4), s_i and t_i ($i \in \{1, 2\}$) denote arbitrarily complex transformations, which need not themselves be invertible (as they are only ever evaluated in the forward direction) and can also be learned by the cINN itself when realised as small sub-networks (Ardizzone et al. 2019c,d).

Another advantage of the cINN architecture is that, as the observations are treated as a condition and simply concatenated to the input of the subnetworks s_i and t_i in each affine coupling layer, it allows for a) an arbitrarily large dimension of the input $\mathbf{y}$ and b) the introduction of a conditioning network h (trained together with the cINN itself), which transforms the input observation into a more helpful, learned representation $\tilde{\mathbf{y}} = h(\mathbf{y})$ for the cINN (Ardizzone et al. 2019d).

4.2.1 Implementation Details

In this paper, we employ a cINN consisting of 11–16 conditional affine coupling layers in the GLOW (Generative Flow; Kingma & Dhariwal 2018a) configuration, where the transformation outputs $s_i(\cdot)$ and $t_i(\cdot)$ are estimated by a single subnetwork $r_i(\cdot) = (s_i(\cdot), t_i(\cdot))$. The latter choice, reduces the number of sub-networks per affine layer from four to two, reducing network complexity and computation time. As sub-networks r_i we employ simple fully-connected architectures consisting of 5–7 layers of size 256 using the rectified linear unit (ReLU, $\mathrm{ReLU}(x) = \max(0, x)$) as activation function.

The affine coupling layers are, furthermore, alternated with random permutation layers, which randomly (but in a fixed and, thus, invertible way) permute the output vector in between coupling layers to improve the mixing of information between the two streams u_1 and u_2 (Ardizzone et al. 2019c,d). For the conditioning network h, we also employ a three-layer fully-connected architecture with layer size 512 and ReLU activation, extracting 256 features in the final layer.

Prior to training, we perform a linear scaling transformation on both the target parameters $\mathbf{x} = \{x_1, \ldots, x_N\}$ and input observations $\mathbf{y} = \{y_1, \ldots, y_M\}$, where each target property x_i and input feature y_i is modified according to

$$\begin{aligned}\hat{x}_i &= \frac{x_i - \mu_{x_i}}{\sigma_{x_i}}, \\ \hat{y}_i &= \frac{y_i - \mu_{y_i}}{\sigma_{y_i}},\end{aligned} \tag{4.5}$$

where μ_{x_i}, μ_{y_i} and σ_{x_i}, σ_{y_i}, denote the means and standard deviations of the respective param-

eter/feature across the training data set. These transformations ensure that the distributions of individual target parameters/input features have zero mean and unit standard deviation, and are trivially inverted at prediction time. The transformation coefficients μ_{x_i}, μ_{y_i} and σ_{x_i}, σ_{y_i} are determined from the training set and applied in the same way to new query data.

We train the cINN approach for this problem by minimisation of the maximum likelihood loss as described in (Ardizzone et al. 2019d) using the Adam (Kingma & Ba 2014) optimiser for stochastic gradient descent with a step-wise learning rate adjustment.

4.3 Training data

4.3.1 Stellar photosphere models

The approach used to train the cINN is to use libraries of theoretical models for stellar photospheres. Our goal is to use the cINN to be able to classify and derive photospheric parameters from medium to low-resolution optical spectroscopy. To this purpose, we selected the most extensive set of available models that offer a spectral resolution better than R∼10000. The most extensive, homogeneous, tested, and readily available[1] library of theoretical photospheric spectra, including different treatments of dust and molecules formation and opacities, applicable in the range of effective temperatures covering the range from ∼2000 to ∼7000 K and gravities appropriate for pre-main sequence stars and brown dwarfs are the Phoenix spectral libraries (e.g., Allard et al. 2012; Husser et al. 2013; Baraffe et al. 2015). In this study, we have used the NextGen, Dusty, and Settl models, the latter is expected to provide the best description of the atmospheric characteristics in most cases of interest (Allard et al. 2012). We have included the older NextGen models as a comparison set, and the Dusty models as they seem to more accurately describe photospheres in the range of 2000 K $\leq T_{\rm eff} \leq$ 3000 K (e.g. Testi 2009). For a more detailed description and comparison of the physical assumption in the models, see the discussion and references in Allard et al. (2012).

The grid of synthetic spectra is available for regularly spaced values of $T_{\rm eff}$ and $\log g$, with steps of 100 K in $T_{\rm eff}$ and 0.5 in $\log g$. To compute a synthetic spectrum for a given set of (arbitrary but within the grid ranges) values of ($T_{\rm eff}$, $\log g$, and $A_{\rm V}$) we set up the following procedure: first, we identify the values of $T_{\rm eff}$ and $\log g$ in the grid that bracket the requested values, then we interpolate linearly in $\log g$ at the values of the two bracketing $T_{\rm eff}$ values, then we interpolate linearly the two resulting spectra at the requested $T_{\rm eff}$ value, finally we compute and apply the extinction following the Cardelli et al. (1989) prescription, with $R_{\rm V}$ as a user selectable parameter (in this study we use $R_{\rm V}$=4.4, see Section 4.3.2). The resulting spectrum is then convolved at the MUSE resolution, using a Gaussian kernel, and resampled on the MUSE

[1]We downloaded the theoretical spectra from the websites: https://osubdd.ens-lyon.fr/phoenix/ and http://svo2.cab.inta-csic.es/theory/newov2/

wavelength grid.

4.3.2 Databases and networks

In this study, we analyse the cINN performance based on each of the three spectral libraries described in the previous section. Accordingly, we construct a training data set for each spectral library using the interpolation scheme we have outlined. For the target parameter space, we adopt the following limits:

For NextGen and Settl we limit $T_{\rm eff}$ to a range of 2600 to 7000 K and $\log(g/{\rm cm\,s^{-2}})$ from 2.5 to 5. The Dusty library has an overall smaller scope, therefore we can only probe from 2600 to 4000 K in $T_{\rm eff}$ and from 3 to 5 in $\log(g/{\rm cm\,s^{-2}})$ here. For $A_{\rm V}$ we select the same range of 0 to 10 mag for all three libraries, where we use the Cardelli et al. (1989) extinction law with $R_{\rm V} = 4.4$ to artificially redden the model spectra. We choose $R_{\rm V} = 4.4$ considering the application of our networks to the Carina Nebula (Hur et al. 2012) in the follow-up study. As some of the template stars used in this paper (Section 4.4) are dereddend assuming $R_{\rm V} = 3.1$, we have also experimented with training data sets using $R_{\rm V} = 3.1$. We have not found a significant difference in our main results, therefore we keep using $R_{\rm V} = 4.4$ in this study.

In terms of wavelength coverage, we match the range of the template spectra described in Section 4.4, i.e. $\sim$ 5687 to $\sim$ 9350 Å, and adopt the MUSE spectral resolution subdividing the wavelength interval into a total of 2930 bins with a width of 1.25 Å. Additionally, we normalise the spectra to the sum of the total flux across all bins.

To generate the training data we opt for a uniform random sampling approach, where we sample both $T_{\rm eff}$ and g in log-space and only $A_{\rm V}$ in linear space within the above-specified limits for the three libraries. We generate a total of 65,536 synthetic spectra models for each library. Note that we have also experimented with larger training sets, but have not found a significant increase in the predictive performance of our method, such that we deemed this training set size sufficient.

Finally, we randomly split each of these three initial databases 80:20 into the respective training and test sets for the cINN. The former subsets mark the data that the cINN is actually trained on, whereas the latter are withheld during training and serve to quantify the performance of the trained cINN on previously unseen data with a known ground truth of the target parameters.

We first train 50 networks for each library with randomised hyper-parameters of cINN and we select the best network based on the performance on the test set and template stars. We train the network until both training loss and test loss converge or either of them diverges, where the latter cases are discarded. It takes about 50 min to train one network (6 hours for 50 networks using 7 processes in parallel) with an NVIDIA GeForce RTX 2080 Ti graphic card. Once trained, our networks can sample posterior estimates very efficiently. Using the same graphic card (NVIDIA GeForce RTX 2080 Ti graphic card) and sampling 4096 posterior estimates per observation, it

takes about 1.1 sec to sample posterior distributions for 100 observations (91 observations per second). When tested with M1 pro CPU with 8 cores, it takes about 13 sec for 100 observations (7.6 observation/sec).

4.4 Class III templates

The set of observations on which we validate our networks contains 36 spectra of well-known Class III stars observed with VLT/X-Shooter (Manara et al. 2013, 2017). We refer the reader for details of observations and data reduction to original papers. Templates come from different star-forming regions (Taurus, Lupus, Upper Scorpius, σ Orionis, TW Hydrae Association, Chameleon I) and span a broad range of effective temperatures (2300 – 5800 K), as well as spectral types (M9.5 - G5.0). In this work we use their properties provided by Manara et al. (2013, 2017); Stelzer et al. (2013).

Spectral types for stars later than K5 were obtained based on the depth of molecular absorption bands (TiO, VO and CaH) and a few photospheric lines (e.g., Na I, Ca I, Mg I, etc.) present in the optical part of the spectra (Manara et al. 2013). Earlier K-type stars were identified using the spectral indices introduced by Herczeg & Hillenbrand (2014), while G-type stars were identified based on the difference at 5150 Å of continuum estimated between 4600 and 5400 Å, and 4900 and 5150 Å (Herczeg & Hillenbrand 2014). Effective temperatures ($T_{\rm eff}$) were derived from spectral types using relations from Luhman et al. (2003) for M-type objects and Kenyon & Hartmann (1995) for K- and G-type stars. Most of the templates have none or negligible extinction ($A_{\rm V} < 0.5$ mag, Manara et al. 2017); those with $A_{\rm V} > 0.3$ were dereddened before analysis assuming the extinction law from Cardelli et al. (1989) and $R_{\rm V} = 3.1$.

Surface gravity ($\log g$) of Class III sources was estimated using the ROTFIT tool (Frasca et al. 2003). It compares the observed spectrum with the grid of referenced spectra and finds a best-fit minimising the χ^2 of difference between the spectra in specific wavelength ranges. Stelzer et al. (2013) and Manara et al. (2017) used BT-Settl spectra in a $\log g$ range of 0.5 – 5.5 dex as reference. The tool also provides $T_{\rm eff}$, radial and rotational velocities; but we use $T_{\rm eff}$ derived from spectral types in the subsequent analysis. Table 4.1 provides a summary of the Class III stars and their stellar parameters. We exclude from the original paper sources, which are suspected to be unresolved binaries or their youth is doubtful due to the lack of lithium absorption line at 6708 Å (Manara et al. 2013).

X-Shooter has higher spectral resolution than MUSE, thus template spectra were degraded to MUSE resolution (R∼4000) using a Gaussian kernel and re-sample on MUSE spectra within the range of 5687.66 – 9348.91 Å (the common spectral range of MUSE and optical arm of X-Shooter). Subsequently, spectra are normalised to the sum of the total flux of the stellar spectrum within the analysed spectral range.

Table 4.1: Stellar parameters of Class III template stars. The last column indicates the literature source of the $\log(g)$ values, where "fixed" indicates that no measurement was available in the literature and we assumed a fixed value of $\log(g/\mathrm{cm\,s^{-2}}) = 4.0$ instead.

Object Name	Region	Spectral Type	$T_{\rm eff}(K)$	$\log(g/\mathrm{cm\,s^{-2}})$	Reference $\log(g)$
RXJ0445.8+1556	Taurus	G5.0	5770	3.93	Manara et al. (2017)
RXJ1508.6-4423	Lupus	G8.0	5520	4.06	Manara et al. (2017)
RXJ1526.0-4501	Lupus	G9.0	5410	4.38	Manara et al. (2017)
HBC407	Taurus	K0.0	5110	4.33	Manara et al. (2017)
PZ99J160843.4-260216	Upper Scorpius	K0.5	5050	3.48	Manara et al. (2017)
RXJ1515.8-3331	Lupus	K0.5	5050	3.86	Manara et al. (2017)
PZ99J160550.5-253313	Upper Scorpius	K1.0	5000	3.81	Manara et al. (2017)
RXJ0457.5+2014	Taurus	K1.0	5000	4.51	Manara et al. (2017)
RXJ0438.6+1546	Taurus	K2.0	4900	4.12	Manara et al. (2017)
RXJ1547.7-4018	Lupus	K3.0	4730	4.22	Manara et al. (2017)
RXJ1538.6-3916	Lupus	K4.0	4590	4.21	Manara et al. (2017)
RXJ1540.7-3756	Lupus	K6.0	4205	4.42	Manara et al. (2017)
RXJ1543.1-3920	Lupus	K6.0	4205	4.12	Manara et al. (2017)
SO879	σ Orionis	K7.0	4060	3.90	Stelzer et al. (2013)
Tyc7760283_1	TW Hydrae	M0.0	3850	4.70	Stelzer et al. (2013)
TWA14	TW Hydrae	M0.5	3780	4.70	Stelzer et al. (2013)
RXJ1121.3-3447_app2	TW Hydrae	M1.0	3705	4.60	Stelzer et al. (2013)
RXJ1121.3-3447_app1	TW Hydrae	M1.0	3705	4.80	Stelzer et al. (2013)
CD_29_8887A	TW Hydrae	M2.0	3560	4.40	Stelzer et al. (2013)
CD_36_7429B	TW Hydrae	M3.0	3415	4.50	Stelzer et al. (2013)
TWA15_app2	TW Hydrae	M3.0	3415	4.60	Stelzer et al. (2013)
TWA7	TW Hydrae	M3.0	3415	4.40	Stelzer et al. (2013)
TWA15_app1	TW Hydrae	M3.5	3340	4.50	Stelzer et al. (2013)
SO797	σ Orionis	M4.5	3200	3.90	Stelzer et al. (2013)
SO641	σ Orionis	M5.0	3125	3.80	Stelzer et al. (2013)
Par_Lup3_2	Lupus	M5.0	3125	3.70	Stelzer et al. (2013)
SO925	σ Orionis	M5.5	3060	3.80	Stelzer et al. (2013)
SO999	σ Orionis	M5.5	3060	3.80	Stelzer et al. (2013)
Sz107	Lupus	M5.5	3060	3.70	Stelzer et al. (2013)
Par_Lup3_1	Lupus	M6.5	2935	3.60	Stelzer et al. (2013)
LM717	Chameleon I	M6.5	2935	3.50	Stelzer et al. (2013)
J11195652-7504529	Chameleon I	M7.0	2880	3.09	Manara et al. (2017)
LM601	Chameleon I	M7.5	2795	4.00	fixed
CHSM17173	Chameleon I	M8.0	2710	4.00	fixed
TWA26	TW Hydrae	M9.0	2400	3.60	Stelzer et al. (2013)
DENIS1245	TW Hydrae	M9.5	2330	3.60	Stelzer et al. (2013)

4.5 Validation

4.5.1 Validations with synthetic spectra

In this section, we validate whether the trained networks well learned the physical rules hidden in the synthetic Phoenix models or not. We use the test set of each database, the synthetic models that are not used for the training but share the same physics as the training data. As mentioned in Section 4.3.2, we only used 80% of the database for training and remained the rest for validation. Each test set consists of 13,107 test models.

Prediction performance

We introduce an accuracy index for evaluating the parameter prediction performance of the network. The accuracy of the prediction is defined as the deviation between the posterior estimate of the parameter and the ground true value (x^*) of the test model. In this section, we calculate the accuracy in the same physical scales we used to build the databases in Section 4.3.2, meaning that we use the logarithmic scales for the effective temperature and surface gravity and use the linear scale for the extinction magnitude. We use either all posterior estimates sampled for one test model or the maximum a posteriori (MAP) point estimate as a representative. To determine the MAP estimate from the posterior distribution, we perform a Gaussian kernel density estimation on a 1D posterior distribution and find the point where the probability density maximises, similar to the method used in Ksoll et al. (2020) and Kang et al. (2022). In most parts of this paper, we use the MAP estimate to quantify the accuracy of the prediction.

We evaluate the three networks (Settl-Net, NextGen-Net, and Dusty-Net) by using all 13,107 test models in the corresponding test set. For each test model, we sample 4096 posterior estimates and measure the MAP estimates for three parameters from the 1D posterior distributions. In Figure 4.1, we present 2D histograms comparing the MAP values estimated by the Settl-Net with the true values of the entire test models. The Settl-Net predicts all three parameters extremely well so the data points are all lying on the 1-to-1 correspondence line. The NextGen-Net and Dusty-Net as well show extremely good results on the test set. The results of the other two networks are presented in Figures C.1 and C.2.

To quantify the average accuracy of the network for multiple test models, we measure the root mean square error (RMSE) following,

$$\mathrm{RMSE} = \sqrt{\frac{\Sigma_{i=1}^{N}(x_i^{\mathrm{MAP}} - x_i^*)^2}{N}}. \tag{4.6}$$

In the case of the Dusty-Net, the training ranges of the effective temperature and surface gravity are narrower than that of the other two networks. As the total number of models is the same

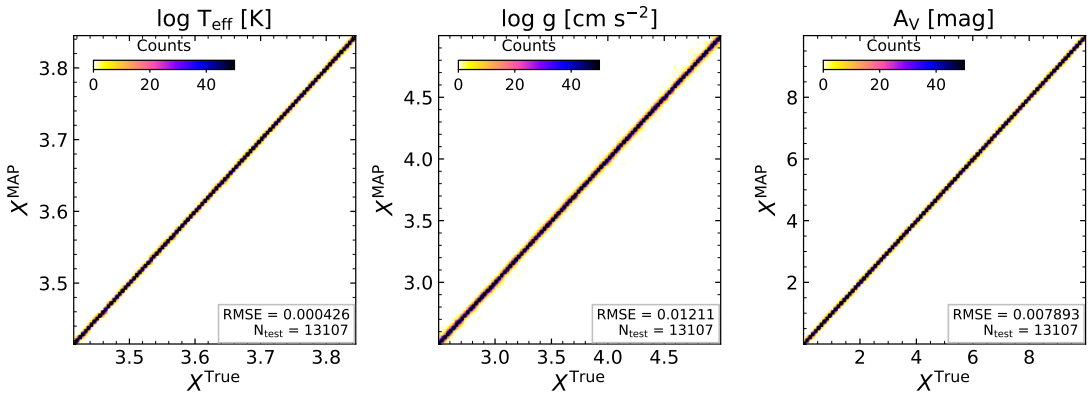

Figure 4.1: 2-dimensional histograms comparing the MAP values estimated by the Settl-Net and the true values for the entire test models of the Settl database. The colours indicate the number of models at each point in the 2D histograms. In the lower right corner, we present the root mean square error (RMSE) and the number of test models used (N_{test}).

Table 4.2: Average prediction performance of three networks (Settl-Net, NextGen-Net, and Dusty-Net) on 13,107 Phoenix synthetic models in the test set. For each parameter and each network, we present the RMSE, the mean accuracy of the MAP estimates, and the RMSE normalised by the parameter range covered in the training data (NRMSE). The test set of each network is drawn from the corresponding synthetic database.

	RMSE			NRMSE		
Network	$\log T_{\rm eff}$	$\log(g)$	$A_{\rm V}$	$\log T_{\rm eff}$	$\log(g)$	$A_{\rm V}$
Settl	4.260×10^{-4}	1.211×10^{-2}	7.893×10^{-3}	9.904×10^{-4}	4.846×10^{-3}	7.893×10^{-4}
NextGen	3.064×10^{-4}	6.742×10^{-3}	6.499×10^{-3}	7.123×10^{-4}	2.697×10^{-3}	6.499×10^{-4}
Dusty	7.274×10^{-5}	1.573×10^{-3}	2.517×10^{-3}	3.888×10^{-4}	7.863×10^{-4}	2.517×10^{-4}

for all three databases (i.e., 65,536 models), the number density of the model for the effective temperature and surface gravity in the Dusty database is higher than the other two. On this account, we define the normalised RMSE (NRMSE),

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{x_{\mathrm{max}}^{\mathrm{training}} - x_{\mathrm{min}}^{\mathrm{training}}}, \tag{4.7}$$

by dividing the RMSE by the training range.

In Table 4.2, we list the RMSE and NRMSE of each parameter for three networks. As already shown in the comparisons between the MAP values and true values (Figures 4.1, C.1, and C.2), the RMSE and NRMSE for all three networks are very small around $10^{-4} \sim 10^{-2}$. The Dusty-Net has the smallest RMSE and NRMSE for all three parameters among the three networks. In the case of the effective temperature and extinction, the differences in NRMSE between the networks are very small, whereas the difference in the NRMSE in the case of surface gravity is relatively noticeable among the three parameters. Although the Dusty-Net has the best results, small values in Table 4.2 demonstrate that all three networks perfectly learned about the synthetic spectra.

Resimulation

To further validate the prediction results of the cINN on the synthetic test data, we verify if the spectra that correspond to the MAP estimates match the respective input spectrum of each test example. We do so by feeding the MAP predictions for the stellar parameters of the 13,107 test examples as an input to our spectra interpolation routine, which we introduced for the training set generation in Section 4.3.1, in order to resimulate the corresponding spectra. Afterwards, we compute the residuals, RMSEs and R^2 scores of the resimulated spectra in comparison to the corresponding input spectra. The latter serves as a goodness-of-fit measure and is defined as

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{4.8}$$

for a set of N observations y_i with corresponding predictions $\hat{y}_i$, where $\bar{y} = \frac{1}{N}\sum_i^N y_i$ denotes the mean of the observations. It takes on values between 0 and 1, with the latter indicating a perfect match (James et al. 2017).

Figure 4.2 summarises the results for Settl-Net, showing the median relative residual against the wavelength in the left panel and the distribution of RMSEs in the right one. The corresponding plots for NextGen-Net and Dusty-Net can be found in Figures C.3 and C.4 in the Appendix. Out of the 13,107 test cases, we could not resimulate spectra for only 52, 32 and 9 MAP predictions for the Settl-Net, NextGen-Net and Dusty-Net, respectively. Only in these few instances either the predicted temperature or gravity (or both) fall outside the interpolation

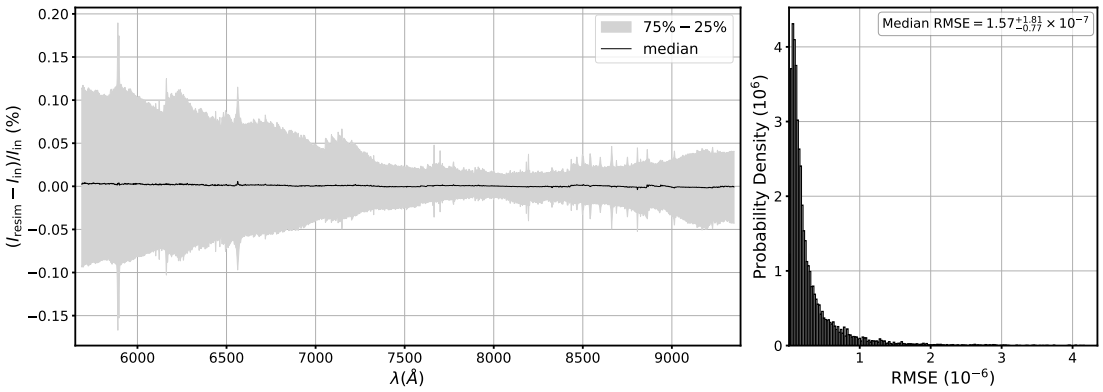

Figure 4.2: Left: Median relative error across the wavelength range of the resimulated spectra based on the MAP predictions of the cINN trained on the Settl models averaged over the 13,107 synthetic spectra in the test set. Here the grey envelope indicates the interquantile range between the 25% and 75% quantiles. Right: Histogram of the RMSEs of the 13,107 resimulated spectra. The mean resimulation RMSE across the test set is $3.01 \pm 4.35 \times 10^{-7}$.

limits of the respective spectra library, such that the spectrum cannot be resimulated. Notably all of these cases are extreme edge cases, i.e. right at the training boundaries of either $T_{\rm eff}$ or $\log(g)$ such that the cINN MAP estimates fall ever so slightly outside the limits while still being an excellent match to the ground truth.

Figure 4.2 confirms the excellent precision of the MAP predictions demonstrated in the ground truth comparison in Figure 4.1. With a median RMSE of the resimulated spectra of $1.57^{+1.81}_{-0.77} \times 10^{-7}$ (and median R^2 score of 1), we find that the resimulated spectra are practically spot-on to the corresponding input. In the left panel of Figure 4.2 we can also see that, while the overall median residual is very small, there is a systematic trend towards a larger discrepancy between resimulation and input within a shorter wavelength regime (< 7250 Å). This is likely an effect of the overall low flux in the short wavelength regime for the colder stars (< 4000 K), such that even a small deviation in flux results in a comparably larger value of the relative residual. Although, it has to be noted again that with most relative deviations falling below 0.2% the discrepancy is overall marginal even in the short wavelength regime.

As Figures C.3 and C.4 show, NextGen-Net and Dusty-Net exhibit similar behaviour in the resimulation test, although we find slightly lower mean RMSEs with $2.28 \pm 2.48 \times 10^{-7}$ and $9.01 \pm 7.34 \times 10^{-8}$, respectively. Given that the mean RMSEs across the three different spectral libraries agree within one σ, however, it is safe to say that all three networks achieve equally excellent performance in the resimulation test.

4.5.2 Validations with Class III template stars

In this section, we investigate how well our cINNs predict each parameter when applied to real observations by analysing Class III template stars introduced in Section 4.4. Stellar parameter values (i.e., effective temperature, surface gravity, and extinction) provided by previous

Table 4.3: Summary of cINN MAP predictions for the Class III template spectra for the cINN models based on the three different spectral libraries. For $T_{\rm eff}$ and $\log(g)$ the value in parenthesis indicates the difference $x_{\rm lit} - x_{\rm MAP}$ to the literature stellar parameters listed in Table 4.1. Since all Class III templates are assumed to be at zero extinction, for $A_{\rm V}$ the value itself is identical to the difference.

	MAP estimate								
	$T_{\rm eff}$ (K) [$\Delta_{\rm lit}$]			$\log(g/{\rm cm\,s^{-2}})$ [$\Delta_{\rm lit}$]			$A_{\rm V}$ (mag)		
Object Name	Settl	NextGen	Dusty	Settl	NextGen	Dusty	Settl	NextGen	Dusty
RXJ0445.8+1556	5391 [379]	5692 [78]	4161 [1609]	4.28 [-0.35]	4.13 [-0.20]	4.14 [-0.21]	0.21	0.38	-0.02
RXJ1508.6-4423	5069 [451]	5434 [86]	4141 [1379]	4.10 [-0.04]	4.16 [-0.10]	4.13 [-0.07]	-0.31	-0.04	-0.13
RXJ1526.0-4501	5150 [260]	5443 [-33]	4170 [1240]	4.25 [0.13]	4.21 [0.17]	4.13 [0.25]	-0.02	0.19	0.14
HBC407	5129 [-19]	5497 [-387]	4165 [945]	4.71 [-0.38]	4.64 [-0.31]	4.26 [0.07]	0.17	0.37	0.02
PZ99J160843.4-260216	5006 [44]	5366 [-316]	4154 [896]	4.43 [-0.95]	4.42 [-0.94]	4.28 [-0.80]	0.15	0.38	-0.09
RXJ1515.8-3331	4895 [155]	5248 [-198]	4177 [873]	4.25 [-0.39]	4.32 [-0.46]	4.31 [-0.45]	0.00	0.32	0.27
PZ99J160550.5-253313	4759 [241]	5168 [-168]	4192 [808]	4.02 [-0.21]	4.19 [-0.38]	4.34 [-0.53]	0.09	0.40	0.21
RXJ0457.5+2014	4644 [356]	5105 [-105]	4123 [877]	4.37 [0.14]	4.63 [-0.12]	4.47 [0.04]	-0.13	0.34	-0.17
RXJ0438.6+1546	4588 [312]	4992 [-92]	4177 [723]	4.01 [0.11]	4.20 [-0.08]	4.50 [-0.38]	0.01	0.44	0.20
RXJ1547.7-4018	4615 [115]	5015 [-285]	4185 [545]	4.15 [0.07]	4.40 [-0.18]	4.52 [-0.30]	-0.02	0.26	0.13
RXJ1538.6-3916	4464 [126]	4830 [-240]	4180 [410]	4.17 [0.04]	4.38 [-0.17]	4.69 [-0.48]	0.01	0.30	0.21
RXJ1540.7-3756	4225 [-20]	4260 [-55]	4115 [90]	4.22 [0.20]	4.17 [0.25]	4.92 [-0.50]	-0.11	0.12	0.22
RXJ1543.1-3920	4269 [-64]	4299 [-94]	4132 [73]	4.34 [-0.22]	4.32 [-0.20]	5.00 [-0.88]	0.03	0.28	0.39
SO879	4106 [-46]	4027 [33]	3909 [151]	3.96 [-0.06]	4.09 [-0.19]	4.78 [-0.88]	0.22	0.29	-0.12
Tyc7760283_1	3881 [-31]	3748 [102]	3742 [108]	5.00 [-0.30]	4.99 [-0.29]	5.23 [-0.53]	-0.17	-0.34	-0.52
TWA14	3819 [-39]	3739 [41]	3677 [103]	5.07 [-0.37]	4.87 [-0.17]	5.09 [-0.39]	-0.32	0.19	-0.30
RXJ1121.3-3447_app2	3797 [-92]	3622 [83]	3635 [70]	4.78 [-0.18]	4.68 [-0.08]	5.13 [-0.53]	0.38	0.30	0.02
RXJ1121.3-3447_app1	3719 [-14]	3559 [146]	3564 [141]	4.90 [-0.10]	4.77 [0.03]	5.16 [-0.36]	0.01	0.04	-0.07
CD_29_8887A	3670 [-110]	3483 [77]	3491 [69]	4.79 [-0.39]	4.57 [-0.17]	5.05 [-0.65]	0.56	0.51	0.07
CD_36_7429B	3423 [-8]	3264 [151]	3262 [153]	4.70 [-0.20]	4.44 [0.06]	4.82 [-0.32]	0.52	0.50	0.13
TWA15_app2	3467 [-52]	3289 [126]	3306 [109]	4.93 [-0.53]	4.71 [-0.31]	5.02 [-0.62]	0.17	0.31	0.09
TWA7	3519 [-104]	3321 [94]	3316 [99]	4.83 [-0.23]	4.45 [0.15]	4.80 [-0.20]	0.41	0.94	0.14
TWA15_app1	3469 [-129]	3285 [55]	3310 [30]	5.01 [-0.51]	4.79 [-0.29]	5.08 [-0.58]	0.06	0.20	0.10
SO797	3248 [-48]	3225 [-25]	3078 [122]	3.93 [-0.03]	3.47 [0.43]	4.03 [-0.13]	1.07	1.48	0.73
SO641	3129 [-4]	3237 [-112]	2997 [128]	3.86 [-0.06]	3.20 [0.60]	3.81 [-0.01]	0.68	1.46	0.43
Par_Lup3_2	3181 [-56]	3245 [-120]	3048 [77]	3.96 [-0.26]	3.29 [0.41]	4.00 [-0.30]	0.72	1.29	0.40
SO925	3008 [52]	3277 [-217]	2961 [99]	3.76 [-0.06]	2.92 [0.78]	3.61 [0.09]	0.97	2.01	0.76
SO999	3079 [-19]	3294 [-234]	2979 [81]	3.68 [0.12]	2.85 [0.95]	3.58 [0.22]	0.69	1.60	0.54
Sz107	2981 [79]	3272 [-212]	2935 [125]	3.69 [0.11]	2.85 [0.95]	3.50 [0.30]	0.56	1.67	0.35
Par_Lup3_1	2739 [196]	3170 [-235]	2868 [67]	3.53 [-0.03]	2.37 [1.13]	3.04 [0.46]	2.74	3.62	2.47
LM717	2714 [221]	3218 [-283]	2903 [32]	3.46 [0.14]	2.37 [1.23]	2.84 [0.76]	1.83	3.08	1.82
J11195652-7504529	2629 [251]	3165 [-285]	2864 [16]	3.50 [-0.41]	2.27 [0.82]	2.75 [0.34]	2.11	3.43	2.24
LM601	2601 [194]	3137 [-342]	2807 [-12]	3.62 [-]	2.28 [-]	2.98 [-]	1.79	3.16	2.00
CHSM17173	2539 [171]	3096 [-386]	2773 [-63]	3.50 [-]	2.18 [-]	2.61 [-]	1.66	3.45	2.31
TWA26	2477 [-77]	2959 [-559]	2625 [-225]	3.46 [0.14]	1.83 [1.77]	2.56 [1.04]	2.64	3.92	2.92
DENIS1245	2453 [-123]	2924 [-594]	2590 [-260]	3.45 [0.15]	1.71 [1.89]	2.58 [1.02]	2.34	3.74	2.82

papers (Manara et al. 2013, 2017; Stelzer et al. 2013) are listed in Table 4.1. Among the 36 template stars, there are cases where the literature value of effective temperature is out of the training range of the cINNs, or where the literature value of gravity is missing. Two out of 36 stars have temperatures below 2600 K, outside the temperature range of all three databases. Also, 14 stars with temperatures between 4000 K and 7000 K are out of the training range of the Dusty-Net. These stars will be excluded from some analyses in the following sections.

Using each network, we sample 4096 posterior estimates per star and measure MAP estimation for three parameters. We list the MAP values predicted by three networks in Table 4.3.

Parameter comparison between literature and cINN

In Figure 4.3, we compare the stellar parameter values from literature ($x_{\rm lit}$) with MAP predictions ($x_{\rm MAP}$). Each row shows the result of different cINNs. The first two columns are the results of effective temperature and surface gravity. As the extinction value of template stars is negligible, we compare the literature value of the temperature with the MAP estimation of extinction. We calculate the uncertainty of the MAP estimate based on the width of the posterior distribution, but as the uncertainties are all very small, we did not present the uncertainty of the MAP estimate in the figure. For the uncertainty of the literature values, we adopt a 1-subclass temperature interval as the uncertainty of temperature and use the surface gravity uncertainty provided by the literature (Stelzer et al. 2013; Manara et al. 2017). According to the literature, the 1-σ uncertainty of extinction is $\sim$ 0.1–0.2 mag, so we indicate from -0.2 to 0.2 mag range in grey to show the uncertainty range.

In this section, we do not use some stars in our analyses where the stellar parameter value from literature is out of the training range or where any stellar parameter value is missing, although they are presented in Figure 4.3 by triangle symbol. We use 34, 34, and 20 stars for Settl-Net, NextGen-Net, and Dusty-Net, respectively when analysing temperatures or extinction, and use 32, 32, and 18 stars respectively when analysing gravity.

Comparing the temperature MAP estimates with the literature values, we confirm that the majority of stars are lying close to the 1-to-1 correspondence line. We calculate the RMSE for each network by only using stars whose temperature literature values are within the training range (i.e., circle markers in Figure 4.3). Considering that the average of the 1-subclass temperature interval of these stars is about 140 K, the RMSE values of 175.3 K, 192.3 K, and 94.02 K for Settl-Net, NextGen-Net, and Dusty-Net, respectively, are well within 1 to 2 subclasses interval. As shown in the figure and RMSE values, Dusty-Net has the best agreement with the literature value when the temperature is within its training range of 2600 – 4000 K. However, Dusty-Net shows very poor agreement with the literature values when the temperature is outside the training range. This implies the caution of using cINN to analyse stars far from the training range. Comparing Settl-Net and NextGen-Net having the same training range, MAP estimates

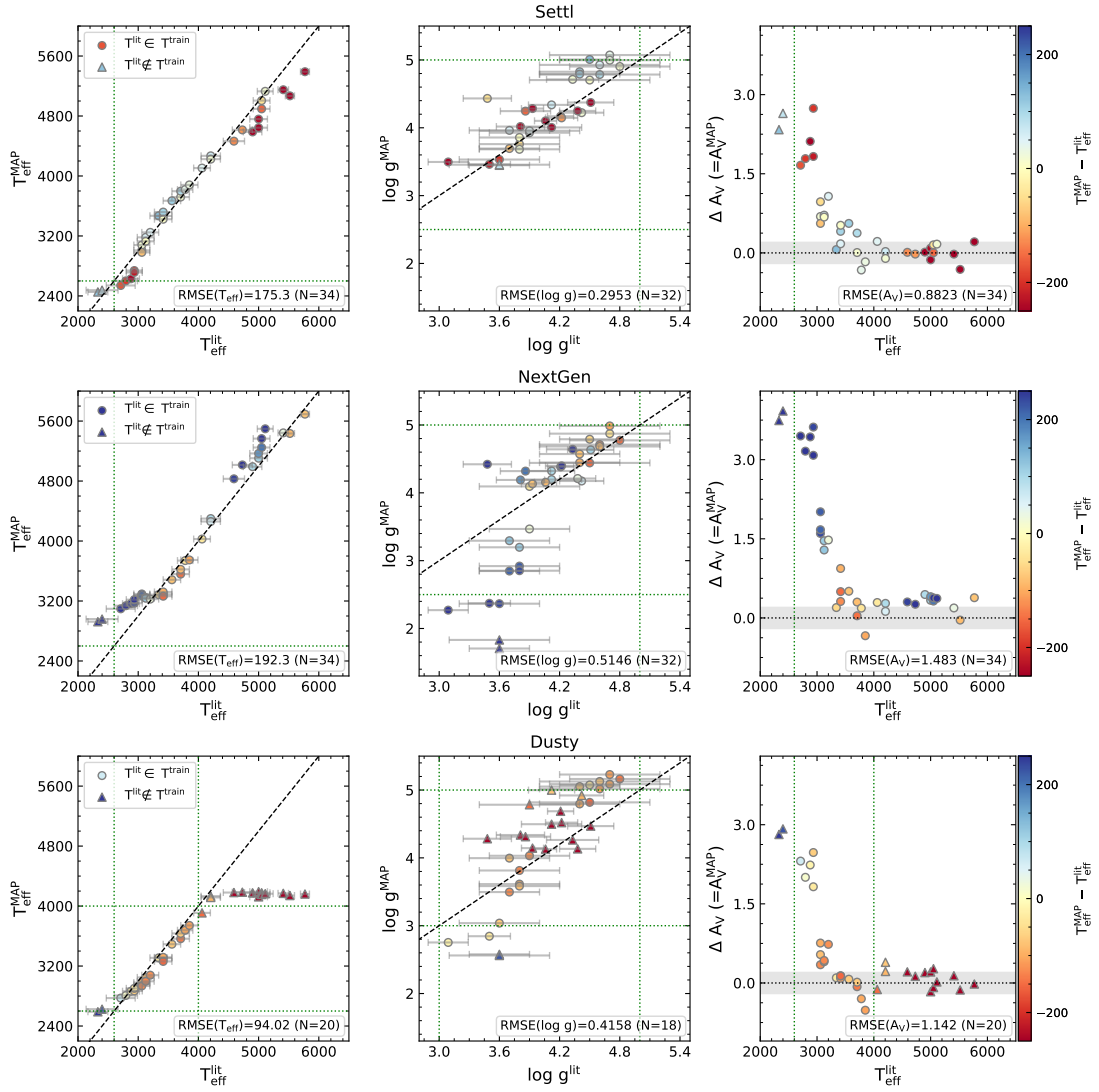

Figure 4.3: Comparison of MAP predictions with literature values in Table 4.1. Stars are basically denoted by circle symbols but triangle symbols denote stars excluded in analyses such as RMSE calculation either because their literature values of temperature are out of the cINN training range or because their literature values of surface gravity are missing. The colour indicates the temperature deviation between the MAP estimate and the literature value. We indicate the training range of each parameter with green dotted lines. In the third column, the grey horizontal area presents the 1-σ uncertainty (i.e., 0.2 mag) of extinction provided by literature.

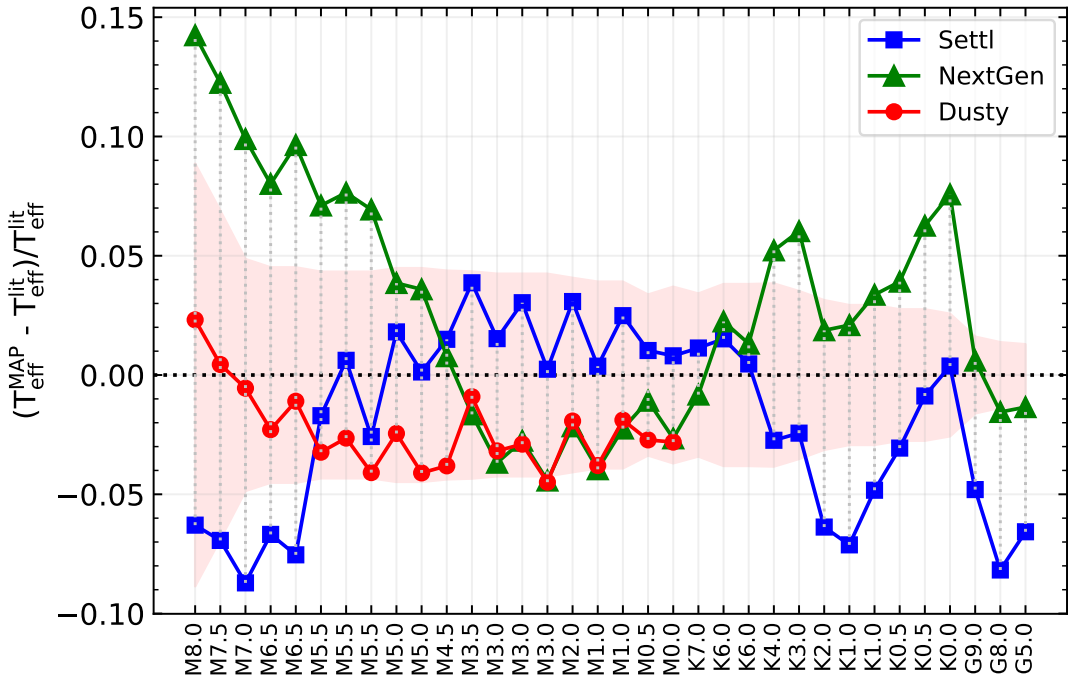

Figure 4.4: Relative temperature deviations of the template stars between the MAP estimates and the literature values sorted by their spectral type. Different colours and symbols indicate the results of three different cINNs. The pink area indicates the uncertainty of the literature value of temperature. We only present template stars whose literature value of temperature is within the network training range.

of Settl-Net are closer to the literature values.

To compare the performance of the three networks on the temperature in more detail, we present the relative temperature deviations between the MAP predictions and the literature values sorted by their spectral type. Figure 4.4 as well shows that MAP estimates from Dusty-Net are in good agreement with the literature value within a 5 per cent discrepancy. In the case of Dusty-Net, all but one star have a deviation within the 1-subclass interval. In the case of Settl-Net and NextGen-Net, 23 and 16 stars out of 34, respectively, have a deviation less than a 1-subclass interval. MAP estimates of Settl-Net and NextGen-Net have a relatively poor agreement with the literature values for hot stars of 4500 K (e.g., K4.0 type) or higher. However, the discrepancies are still within 10 per cent. The average absolute relative deviations when only using the templates within the training range of each network are 3.28, 4.49, and 2.58 per cent for Settl-Net, NextGen-Net, and Dusty-Net, respectively (Table 4.4). These average errors are equivalent to 1.08, 1.12, and 0.601 subclasses.

In the case of surface gravity, the RMSE of Settl-Net, NextGen-Net, and Dusty-Net are 0.30, 0.51, and 0.42 dex, respectively. However, because the surface gravity value from previous studies (Stelzer et al. 2013; Manara et al. 2017) is obtained by fitting the spectrum on the Settl models, the MAP estimate of Settl-Net is essentially the closest to the literature value. Although Settl-Net has the smallest RMSE value, considering the uncertainty of literature values, the other

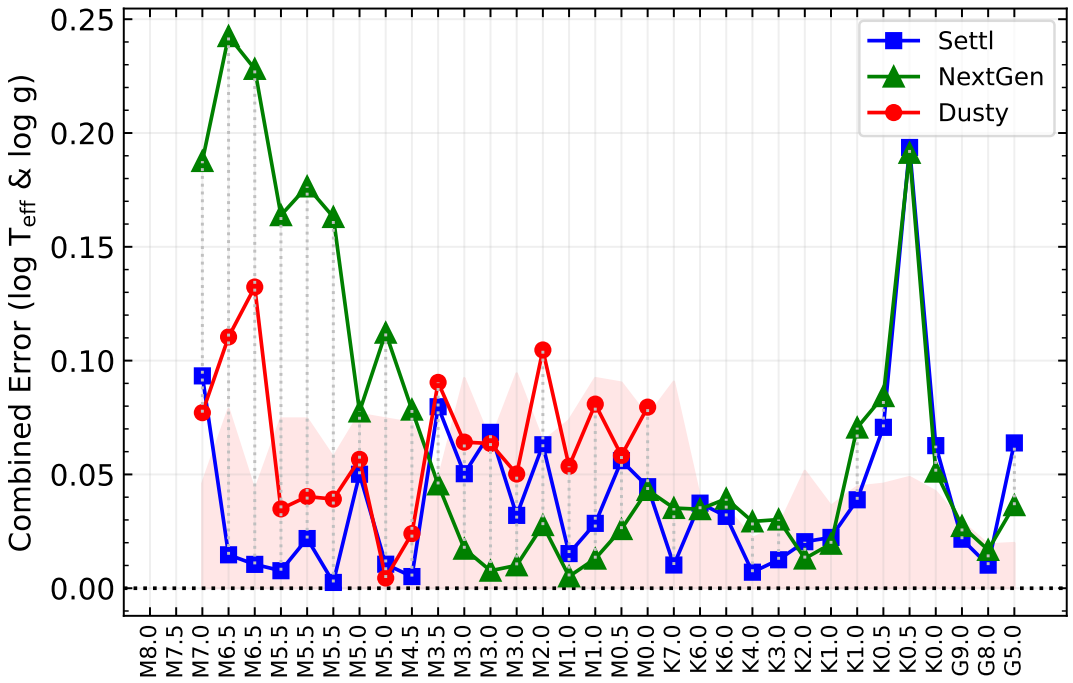

Figure 4.5: Average relative error of the template stars between the MAP estimates and the literature values sorted by their spectral type. The average error is calculated as a root mean square of the relative errors of temperature and gravity, both in log-scale (Eq. 4.9). The pink area indicates the 1-σ uncertainty of the literature value. We only present template stars whose literature value of temperature is within the network training range and whose literature value of gravity is presented. Colour codes are the same as in Figure 4.4.

two networks also have good agreement with the literature value.

To combine the results of temperature and surface gravity, we define the combined error of two parameters as,

$$\text{Combined error} = \sqrt{\frac{1}{2}\left(\left(\frac{\Delta T_{\rm eff}}{\log T_{\rm eff}^{\rm lit}}\right)^2 + \left(\frac{\Delta g}{\log g^{\rm lit}}\right)^2\right)},$$

$$\text{for} \tag{4.9}$$

$$\Delta T_{\rm eff} = \log T_{\rm eff}^{\rm MAP} - \log T_{\rm eff}^{\rm lit},$$

$$\Delta g = \log g^{\rm MAP} - \log g^{\rm lit},$$

and present the combined error of each template star. We use the effective temperature in the logarithmic scale to match the scale with surface gravity. The overall result using combined error presented in Figure 4.5 are not significantly different from Figure 4.4, but by adding the gravity error, Settl-Net shows better performance than Dusty-Net even for low-temperature stars. In the case of NextGen-Net, the combined error is larger than the other two networks because there are cases where temperature and gravity errors are both large. The average combined errors across the stars of Settl-Net NextGen-Net, and Dusty-Net are 3.93, 7.20, and 6.47 per cent,

Table 4.4: Average absolute relative error between cINN predictions and literature values for template stars. We calculate the errors by dividing the absolute difference between the MAP estimate and literature value either by literature values (i.e., errors in per cent unit) or by 1-σ uncertainty of literature value (i.e., errors in 1-σ unit). In the case of the effective temperature, the 1-σ uncertainty corresponds to the temperature interval of one subclass. For each network and parameter, we only use template stars whose literature values are within the training range of the network to calculate the errors.

	Average relative error [%]			Average relative error [σ]		
Network	$T_{\rm eff}$	$\log(g)$	$A_{\rm V}$	$T_{\rm eff}$	$\log(g)$	$A_{\rm V}$
Settl	3.28	5.5	-	1.08	0.809	2.78
NextGen	4.49	10.2	-	1.12	1.38	4.95
Dusty	2.58	9.13	-	0.601	1	3.87

respectively.

In the case of Settl-Net, all but 7 stars are in good agreement with the literature values within 1-σ uncertainty. Excluding one star with a large error, most of the stars have errors of less than 5 per cent and a maximum of 10 per cent. Dusty-Net also has small errors (<15 per cent) but Dusty-Net has a disadvantage in that it is inherently less versatile than the other two networks because of its training range. NextGen-Net also shows an error of less than 10 per cent for stars with spectral type earlier than M5.0.

Lastly, in the case of extinction, the deviation between MAP estimates and literature values varies depending on the temperature. For stars hotter than about 3400 K (i.e., M3.0 type), all three networks predict near-zero extinction, with little deviation from literature values. In the case of NextGen-Net, there are stars that are slightly outside the error range but their MAP estimates are sufficiently small. On the other hand, for cool stars below 3400 K, the discrepancy between the MAP value and the literature value gradually grows. In the case of Settl-Net and Dusty-Net, the MAP estimate does not exceed the maximum of 3, but in the case of NextGen-Net, the MAP estimates are slightly larger than the other two networks.

In this section, we showed that the discrepancy between the network MAP prediction and literature value varies with the characteristics of the stars. Based on the overall results, a star of

- M6.5 – K1.0 (2935 – 5000 K) for Settl-Net,
- M4.5 – K1.0 (3200 – 5000 K) for NextGen-Net,
- M5.5 – M0.0 (3060 – 4000 K) for Dusty-Net

shows especially high agreement with the literature values. Settl-Net showed the best agreement with the literature values overall. Dusty-Net also shows a good agreement for stars whose temperature is within the Dusty database of 2600 – 4000 K. NextGen-Net has relatively large errors compared to the other two, but it still shows reliable performance for early-type stars. Given that Settl-Net and NextGen-Net cover a wider range of temperature (i.e., 2600 – 7000 K) and gravity

(2.5 – 5 $\log(\mathrm{cm\,s^{-2}})$) than Dusty-Net, Settl-Net is the best choice among the three networks. However, all three networks showed good agreement with the literature values considering their uncertainty.

This result shows how well our cINN predictions are in good agreement with values obtained with the classical methods in previous studies. Differences between literature values and network predictions do not demonstrate that the network prediction is wrong. For example, in the case of surface gravity, because the literature value was also obtained by fitting spectra based on the Settl model, there is inevitably a larger discrepancy between the literature values and MAP predictions of NextGen-Net and Dusty-Net. This means that we need to consider methods used in the literature, and additional analysis is required to judge whether the cINN prediction is really wrong or not. The resimulation following in the next section will provide a better clue to determine the correctness of our cINN predictions.

Resimulation

As we have done for the synthetic test data in Section 4.5.1, we also evaluate the accuracy of the cINN predictions on the Class III template by resimulation to quantify the agreement between the spectra corresponding to the MAP estimates with the input spectra. In this case, we also run a resimulation for the nominal literature stellar parameters of the Class III sources listed in Table 4.1 for comparison. Some of the Class III template sources in our sample do not have an estimate for $\log(g)$ in the literature. For these sources, we assume a fixed value of $\log(g/\mathrm{cms^{-2}}) = 4.0$ in our resimulation, which is a reasonable guess for the spectral types in our sample. The sources in question are marked as "fixed" in the last column of Table 4.1. There are a few templates (7 for Settl, 1 for NextGen and 8 for Dusty; see Table 4.3), where the cINN extinction MAP estimate has a non-physical negative value. Since most of these are only barely below zero, we decide to allow these negative values to be accounted for during the resimulation.

Figure 4.6 shows an example result of the resimulation for the M4-type template star SO797 for all three spectral libraries with the top panels comparing the resimulated spectra to the input spectrum and bottom panels showing the corresponding residuals. Here the red curve indicates the resimulation result derived from the cINN MAP estimates, whereas the blue curve marks the literature-based outcome. In this particular example, the cINN recovers both T_{eff} and $\log(g)$ quite accurately for all three spectral libraries, but overestimates A_V for this supposedly zero extinction template Class III source by 1.07, 1.48 and 0.73 mag based on Settl, NextGen and Dusty, respectively. Interestingly, however, we find that the resimulated spectrum based on the cINN MAP prediction with the supposedly wrong A_{V} matches the input spectrum better than the spectrum derived from the literature value in all three examples as e.g. attested by the smaller RMSE and better R^2 score of 2.7×10^{-5} and 0.98 compared to 3.77×10^{-5} and 0.97 in the Settl case. Figure C.5 in the Appendix shows another such example, where it is immediately apparent

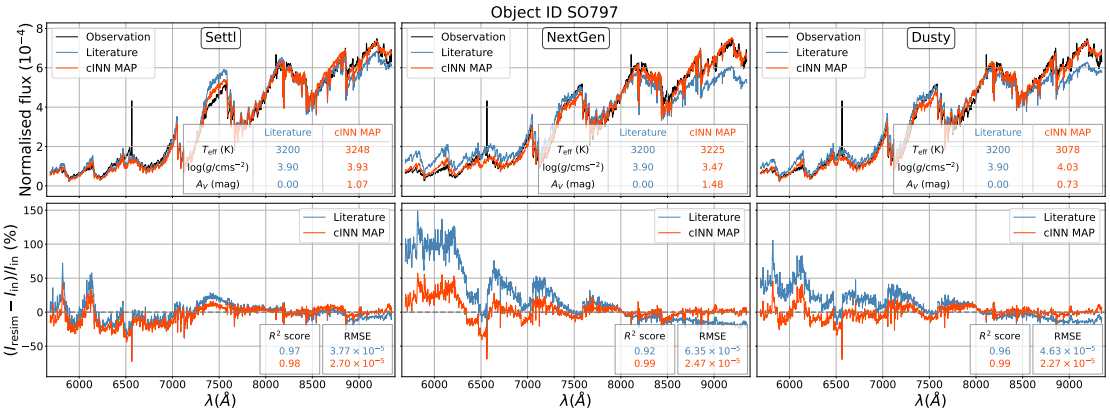

Figure 4.6: Resimulation results for Class III star SO797. The columns show in order the results for the three different spectral libraries Settl, NextGen and Dusty. Top: Comparison of resimulated spectrum. The blue spectrum indicates the resimulation derived from the literature stellar parameters from Table 4.1. The red spectrum shows the corresponding resimulation based on the cINN MAP prediction. The respective input parameters for the resimulation are summarised in the table in the bottom right corner. The relative residuals $(I_{\text{resim}} - I_{\text{in}})/I_{\text{in}}$ of the resimulated spectra compared to the input spectrum are shown in the bottom panels, respectively.

that the cINN-based resimulated spectrum matches the input observation much better than the literature-based solution, which evidently does not capture the slope of the observed spectrum correctly.

Figures 4.7, 4.8 and Table C.1 in the Appendix summarise the resimulation results across the entire Class III template sample, showing the median relative residuals against the wavelength, the distributions of RMSEs and R^2 scores, and a table of all RMSEs and R^2 scores, respectively. Note that the resimulation statistics vary between the libraries here. Given the lower effective temperature limits of the libraries (i.e. 2600 K) 2 of the 36 templates, namely TWA26 and DENIS1245, can a priori not be resimulated with Settl and NextGen. For Dusty the literature sample is even smaller with only 20 out of 36 templates due to the low upper temperature limit of 4000 K. For the resimulation of the MAP estimates we can use 31 templates with the Settl-Net, 29 with NextGen-Net and only 17 with Dusty-Net. For more details, we refer to Table C.1. Note that for the Dusty resimulation there are actually 7 templates, where the $\log(g)$ prediction is above the training set limit of 5. However, since the Dusty spectral library does actually extend to $\log(g/\text{cms}^{-2}) = 5.5$, we decide to run the resimulation for these 7 templates anyways, in particular since for most of those the $\log(g)$ prediction is only barely above 5 (see Table 4.3).

Figure 4.7 shows that for all three libraries we find that our observation from Figure 4.6, i.e. that the resimulated spectrum based on the cINN prediction fits the input spectrum better than the literature-based resimulation, holds on average across the entire template sample. The distributions of the RMSEs and R^2 scores of the resimulated spectra in Figure 4.8 further confirm this, as the cINN-based resimulated spectra tend towards smaller RMSEs and slightly better R^2 scores compared to the literature-based spectra for all three spectral libraries.

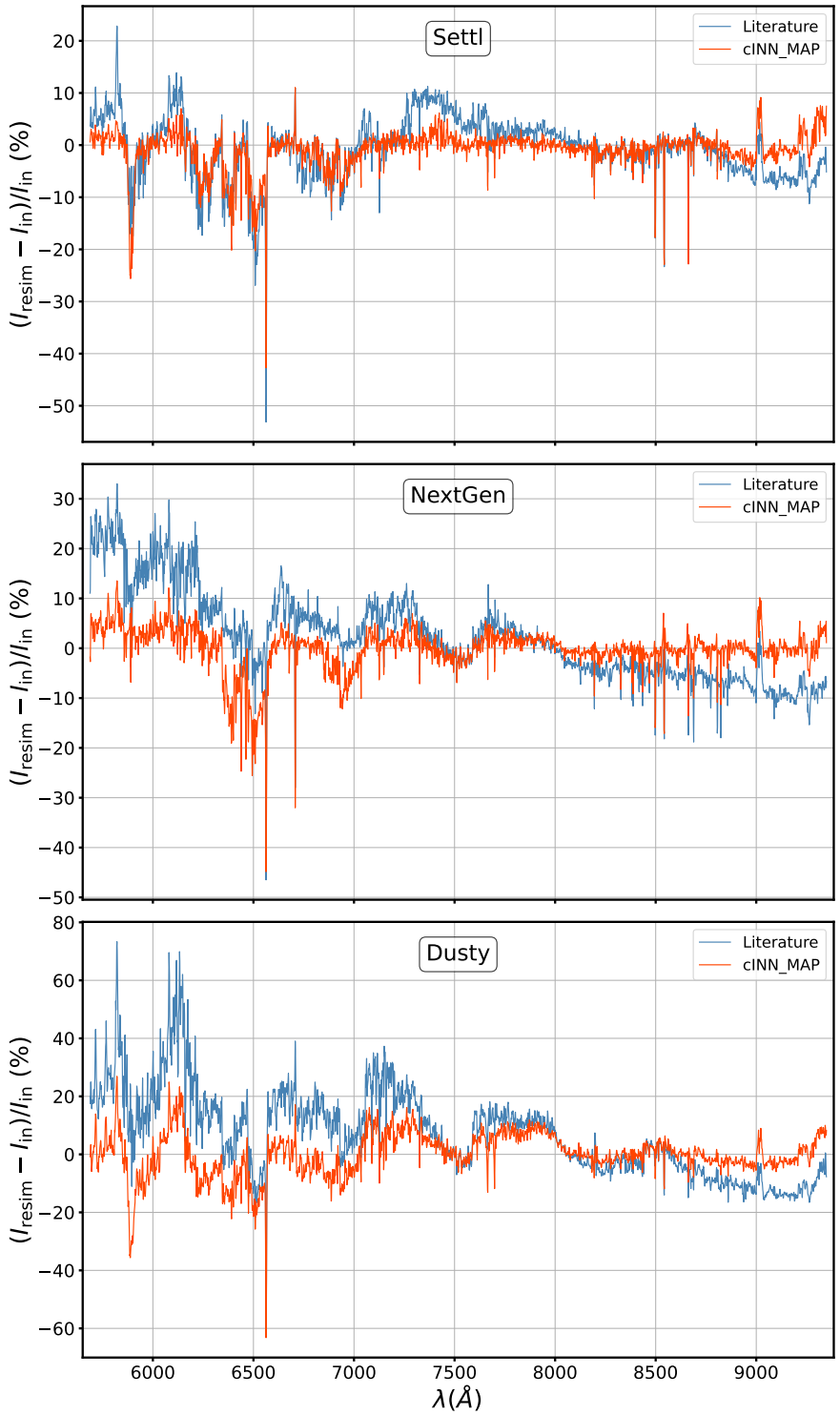

Figure 4.7: Comparison of the median relative error of the resimulated spectra for the Class III template stars between the resimulations based on the literature stellar parameters (blue, see Table 4.1) and the cINN MAP predictions (red). From top to bottom, the panels show the corresponding results for the three tested spectral libraries Settl, NextGen and Dusty.

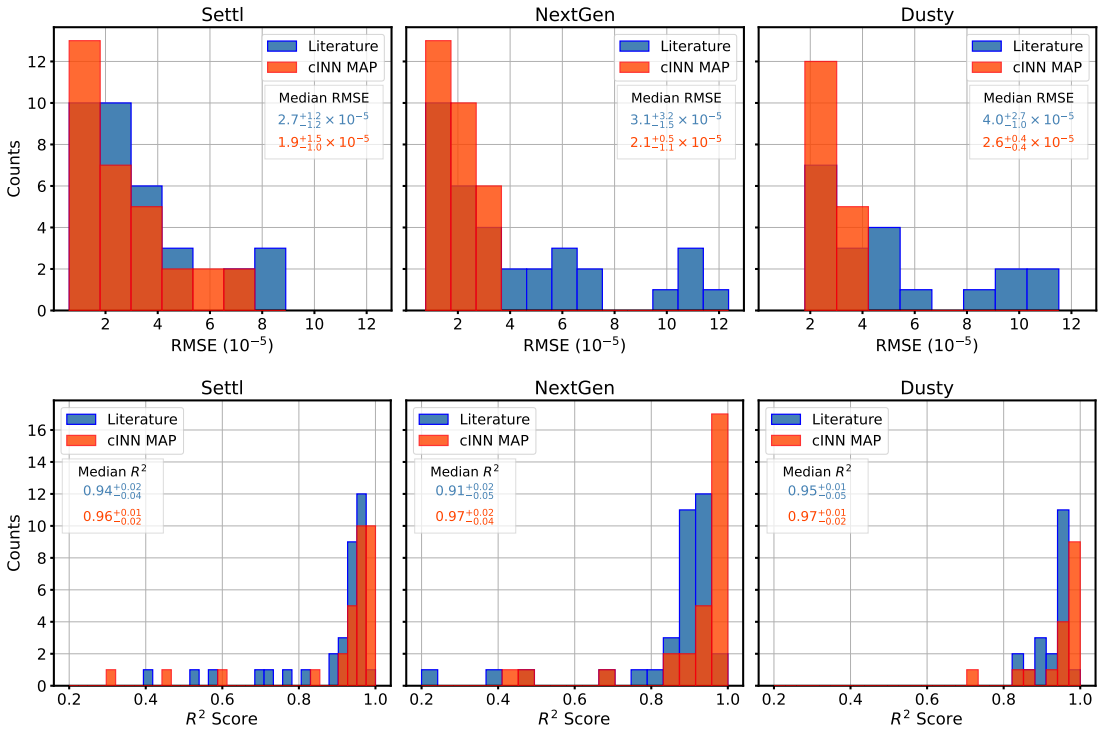

Figure 4.8: Top: Histograms of the RMSEs for the resimulation on the Class III template spectra for the three different spectral libraries. Bottom: Histograms of the corresponding R^2 scores for the resimulated spectra.

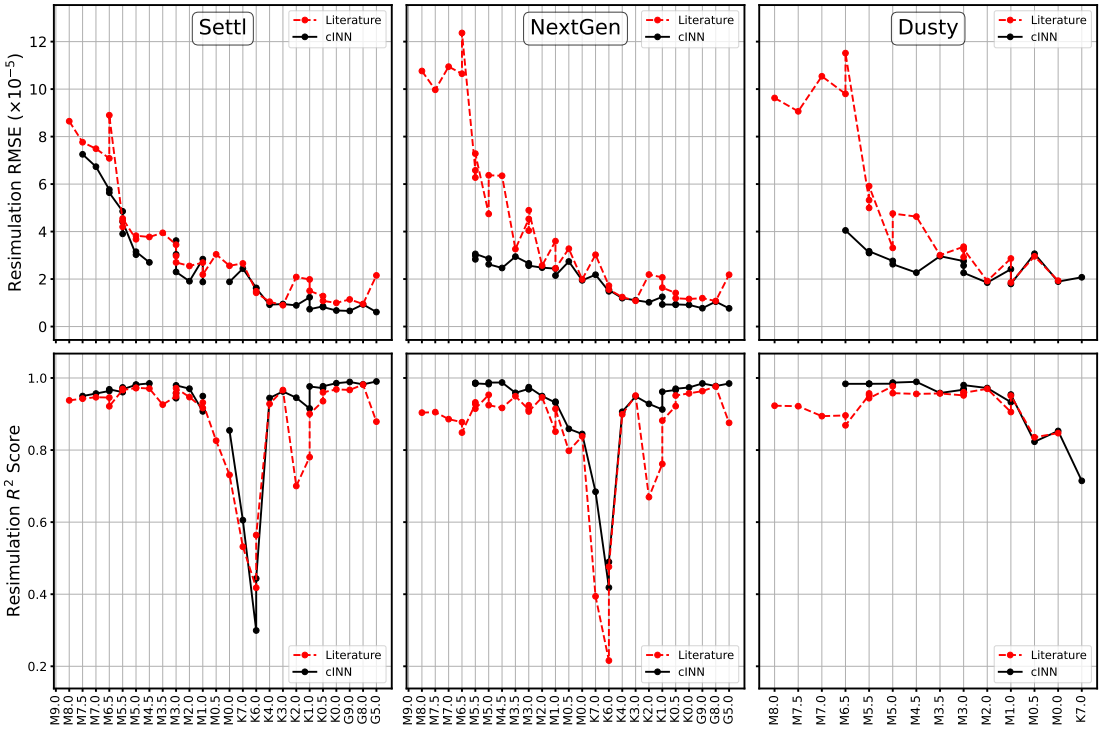

Figure 4.9: Comparison of the resimulation accuracy measures (RMSE in the top row, R^2 score in the bottom) for the three spectra libraries to the spectral type of the Class III templates. In all panels, the dotted red line indicates the results for the resimulation based on the literature stellar properties, while the black line shows the cINN-based outcomes.

Examining the 7 templates, for which the Dusty-based cINN prediction of $\log(g)$ exceeds the learned upper limit of 5 (i.e. the cINN extrapolated), more closely, the resimulation results show that even when the cINN extrapolates, the set of predicted parameters corresponds to a spectrum, which matches the input observation quite well and, in particular, equally if not better than the respective spectrum resimulated from the literature values as indicated by the R^2 scores (see Table C.1 and Figure C.6 for an example). This result shows that the cINN prediction is actually fairly robust even in the event of slight extrapolation.

Comparing our chosen resimulation accuracy measures to the spectral types of the Class III templates in Figure 4.9, we find that the RMSEs exhibit an increasing trend towards the M-types for all three spectral libraries. For the R^2 scores, we find a notable dip in the goodness of fit for the intermediate spectral types, i.e. between M2 to K3, in both the resimulation of the literature and cINN-based values for Settl and NextGen. The beginning of this dip can also be seen in the Dusty-based results up to the temperature limit of this library at the K7 type. Interestingly, when compared to Figure 4.5 in this spectral type range the discrepancy between the cINN prediction and literature stellar properties is relatively low, i.e. both cINN and literature values correspond to an equally sub-optimal fit to the observed spectra.

Overall the resimulation test shows that the cINN approach predicts parameters for the real Class III template spectra that correspond to spectra, which not only fit the input observations very well (as shown by the good R^2 scores in Figure 4.8 and Table C.1), but also match better than the spectra resimulated from the literature values in most instances.

4.6 Feature Importance

4.6.1 Importance calculation

In this section, we evaluate which parts of the spectra the cINN prediction relies the most upon. To do so we measure the so-called *permutation feature importance*, an approach first described by Breiman (2001) for random forest models and later generalised by Fisher et al. (2018). In this study we implement the Fisher et al. (2018) algorithm as described in Molnar (2022), operating as follows:

First, we compute the error on the original held-out test set

$$e_{\text{orig}} = L\left(\mathbf{X}, g(\mathbf{Y})\right), \tag{4.10}$$

where g represents the inverse translation ($\mathbf{x} \leftarrow \mathbf{y}$) of the trained cINN, $\mathbf{X}$ denotes the matrix of the target parameters of the test set ($n_{\text{test}} \times n_{\text{parameters}}$), $\mathbf{Y}$ is the $n_{\text{test}} \times n_{\text{features}}$ feature matrix of the test set and L represents a loss measure. In our case, L is the RMSE of the MAP estimates.

Next, for each feature $j \in \{1, \ldots, n_{\text{features}}\}$, we generate a feature matrix $\mathbf{Y}_{\text{perm},j}$ via random permutation of the j-th column in order to break the association between feature j and the target parameters $\mathbf{x}$, estimate the prediction error $e_{\text{perm},j} = L\left(\mathbf{X}, g\left(\mathbf{Y}_{\text{perm},j}\right)\right)$ based on the permuted data set, and compute the feature importance of feature j as the quotient

$$\text{FI}_j = \frac{e_{\text{perm},j}}{e_{\text{orig}}}. \tag{4.11}$$

The larger FI_j is, the worse the model prediction becomes if feature j is scrambled via permutation, i.e. the more important feature j is to the model's decision making. The closer FI_j is to 1, on the other hand, the less feature j affects the predictive performance and, thus, the less relevant it is to the model's reasoning.

In our particular case, the feature space is very high dimensional with 2930 spectral bins per spectrum. Consequently, computing the individual per spectral bin feature importance is rather computationally expensive as it requires generating the posteriors and determining the MAP estimates for each of the 2930 bins. Although the computational cost alone is not prohibitive in this case given the cINNs great efficiency, we still opt for a slightly different approach, because the spectral bins themselves are also not necessarily independent of each other. Instead of using the individual bins, we group them together into combined features, for which we then estimate the importance. In practice, this means that we permute multiple columns at once (each column with its own permutation seed though) corresponding to the spectral bins in a given group. For the setup in this study in particular we decide to evaluate the feature importance across the wavelength range using groups of 10 bins, which corresponds to a spectral width of 12.5 Å. We set all groups to overlap by 5 bins (i.e., 6.25 Å) with the preceding and following groups. We average feature importance for overlapping bins.

4.6.2 Important features for M-, K-, and G-type stars

We draw three groups from the test set according to the temperature of the test model: M-type (2600K–3850 K) group, K-type (3900K–5110 K) group, and G-type (5150K–6000 K) group, and evaluate the feature importance across the wavelength for each group per network. In the case of Dusty-Net, we only evaluate for the M-type group because the maximum temperature of the Dusty database is 4000 K.

Figure 4.10 presents the feature importance of Settl-Net for M-type stars. To compare the important features with the locations of stellar parameter tracers existing in the real spectrum, we plot the median flux of M-type template stars in the first row, and indicate the locations of several tracers of stellar parameters (Table 4.5): Na I doublet 5890, 5896 Å (T_{eff} and $\log g$, Allen & Strom 1995), Ca I 6122, 6162, 6439 Å ($\log g$, Allen & Strom 1995), Ba II, Fe I, and Ca I blend 6497 Å (T_{eff} and $\log g$, Allen & Strom 1995; Herczeg & Hillenbrand 2014), Hα 6563 Å (T_{eff},

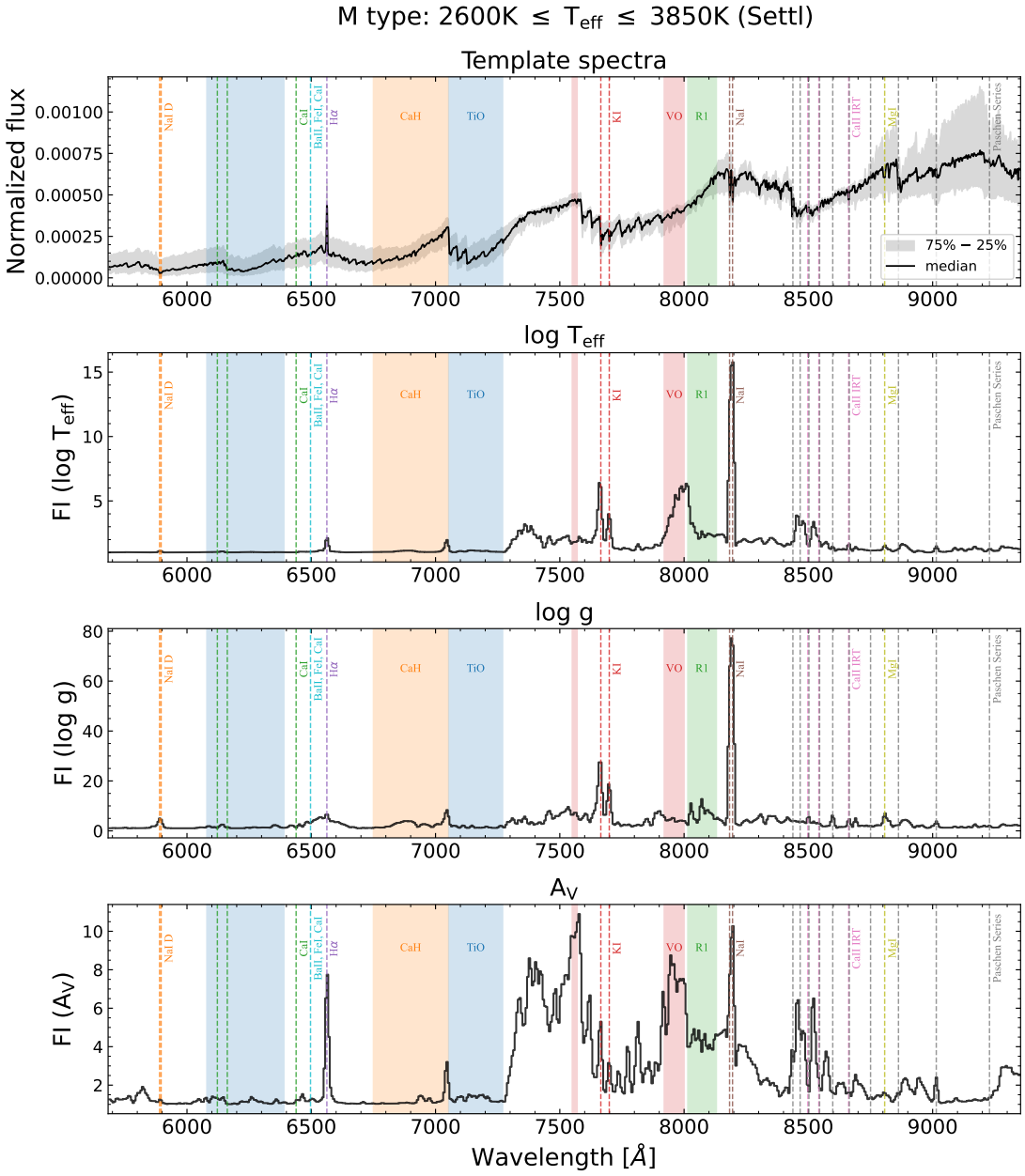

Figure 4.10: Feature importance evaluation for M-type synthetic models in the test set using Settl-Net. We present the median flux of M-type Class III template stars in the first row. The grey area indicates the interquantile range between the 25% and 75% quantiles. The other three rows show the feature importance across the wavelength for each stellar parameter. Vertical lines and shades indicate the location of typical tracers of stellar parameters listed in Table 4.5.

Luhman et al. 2003), K I doublet 7665, 7699 Å ($T_{\rm eff}$ and $\log g$, Manara et al. 2013, 2017), Na I doublet 8183, 8195 Å ($T_{\rm eff}$ and $\log g$, Kirkpatrick et al. 1991; Allen & Strom 1995; Riddick et al. 2007), Ca II IR triplet 8498, 8542, 8662 Å ($T_{\rm eff}$, Kirkpatrick et al. 1991; Allen & Strom 1995; Luhman et al. 2003), Mg I 8807 Å ($T_{\rm eff}$, Manara et al. 2013; Herczeg & Hillenbrand 2014), hydrogen Paschen series ($A_{\rm V}$, Edwards et al. 2013), CaH 6750–7050 Å ($T_{\rm eff}$ and $\log g$, Kirkpatrick et al. 1993; Allen & Strom 1995), TiO 6080–6390, 7053–7270 Å ($T_{\rm eff}$, Kirkpatrick et al. 1991; Henry et al. 1994; Jeffries et al. 2007), ViO 7550–7570, 7920–8000 Å ($T_{\rm eff}$, Allen & Strom 1995; Riddick et al. 2007; Manara et al. 2013), and R1 8015–8130 Å ($T_{\rm eff}$, Riddick et al. 2007) .

To evaluate whether these observational tracers act as important features in our networks, we check whether the feature importance value corresponding to each tracer's wavelength is larger than a fiducial value. We use the value of median plus one standard deviation over the entire wavelength range as a fiducial value to determine an important tracer. For tracers with multiple lines or molecular bands, we average the feature importance for each line or over the wavelength range. In Table 4.5, we mark tracers whose average importance is larger than the fiducial value. We also indicate for which parameters these lines and bands trace in real observations.

Figure 4.10 shows that Na I doublet 8183, 8195 Å lines are the most important feature for Settl-Net to predict stellar parameters of M-type stars. In the case of extinction, there are two wide peaks near 7500 Å, where the redder peak overlaps with the VO molecular band. However, Na I has a similarly large importance value. In the case of temperature and gravity, K I doublet 7665, 7699 Å lines play a second important role, and in extinction, Hα does. VO and R1 molecular absorption bands as well act as important features to determine the temperature and extinction.

We present the feature importance evaluated for NextGen-Net and Dusty-Net in Figure C.8. The fact that Na I, K I, and Hα are important features for M-type stars is the same for all three networks. However, for NextGen-Net, there is a large bump at 7500 Å in the case of temperature. The results of NextGen-Net in overall are spikier than in the other two networks. In the case of Dusty-Net, the importance value of Na I doublet 5890, 5896 Å (Na I D) is relatively large compared to the other networks, and there is a very wide bump around Na I doublet 8183, 8195Å.

Given the fact that extinction affects the overall shape of the spectrum, it is interesting that the Settl-Net relies a lot on a few certain lines. Broad bumps exist in the red part of the spectrum, but there are particularly important lines and areas such as the Na I, Hα, and near VO bands. The result of NextGen-Net is similar to Settl-Net but shows a little more spiky trend with wider peaks. Dusty-Net shows a more wavy shape across the entire wavelength range compared to the others.

Next, in the case of K-type stars, the results of Settl-Net and NextGen-Net are similar to each other, unlike the case of M-type stars, so we only present the result of Settl-Net in this

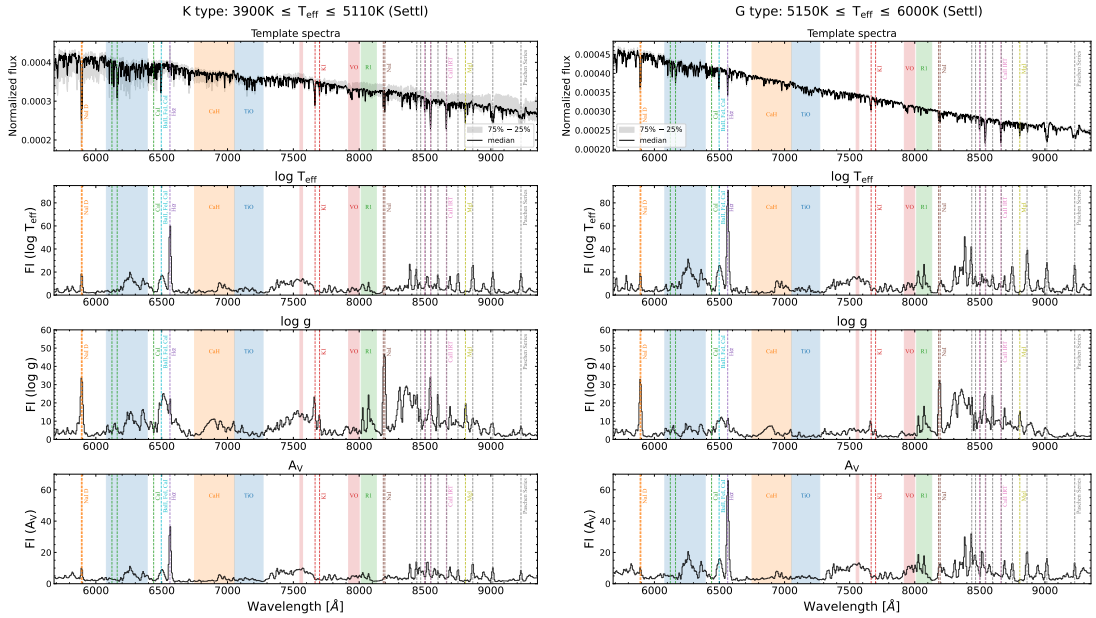

Figure 4.11: Left: Feature importance evaluation for K-type synthetic models in the test set using Settl-Net. Right: Feature importance evaluation for G-type synthetic models in the test set using Settl-Net. The panels in the first row show the median flux of K-type and G-type Class III template stars, respectively. Lines and shades are the same as Figure 4.10.

paper (left panels in Figure 4.11). Compared to the results of M-type stars, it is noticeable that important features are different for each parameter. In the case of temperature and extinction, the overall shapes are similar, with the Hα line being the most important feature. Na I doublet 8183, 8195 Å are no longer so important to determine temperature and extinction for K-type stars. In addition, Na I D lines and hydrogen Paschen series have relatively high importance values. On the other hand, in the case of surface gravity, the Na I doublet 8183, 8195 Å lines still play the most important role. The importance of Na I D in gravity becomes noticeable in K-type stars compared to M-type stars. Additionally, there are several peaks at K I, Mg I 8807 Å used as important features to determine gravity.

The result of G-type stars (i.e., right panels in Figure 4.11) is similar to the K-type stars. The Hα is still the most important feature for temperature and extinction and there are several peaks at the Paschen series as well. For gravity, Na I D becomes more important in G-type stars and has an importance value comparable to that of Na I doublet 8183, 8195Å. These sodium lines are the most important features to determine gravity. On the other hand, the importance of K I lines decreases in G-type stars compared to K-type stars.

These results show that the features that our networks rely on to determine parameters vary depending on the input object. In particular, when changing from M- to K-type, important features change noticeably. For example, the Na I doublet 8183, 8195 Å lines are essential features for networks to understand M-type stars, sensitive to all three stellar parameters, but for earlier type stars (K- and G-types), it is important only to determine gravity. Similarly, the K I doublet

Table 4.5: We mark tracers whose feature importance values are larger than the fiducial value of median plus 1 standard deviation, meaning that marked tracers are significantly important features to determine each stellar parameter. For tracers with multiple lines (e.g., doublets) or molecular bands, we average the feature importance values. The results are based on the feature importance evaluation of Settl-Net (Figures 4.10 and 4.11).

		M-type			K-type			G-type		
Tracers	used in observations for	$T_{\rm eff}$	$\log(g)$	$A_{\rm V}$	$T_{\rm eff}$	$\log(g)$	$A_{\rm V}$	$T_{\rm eff}$	$\log(g)$	$A_{\rm V}$
Na I doublet 5890, 5896 Å	$T_{\rm eff}$, $\log(g)$	-	-	-	✓	✓	✓	✓	✓	✓
TiO 6080–6390, 7053–7270 Å	$T_{\rm eff}$ (M- and late K-type)	-	-	-	-	-	-	-	-	-
Ca I 6122, 6162, 6439 Å	$\log(g)$	-	-	-	-	-	-	-	-	-
Ba II, Fe I, and Ca I blend 6497 Å	$T_{\rm eff}$, $\log(g)$	-	-	-	✓	-	✓	✓	-	✓
Hα 6563 Å	$T_{\rm eff}$ (early type)	-	-	✓	✓	✓	✓	✓	✓	✓
CaH 6750–7050 Å	$T_{\rm eff}$ (M-type), $\log(g)$	-	-	-	-	-	-	-	-	-
VO 7550–7570, 7920–8000 Å	$T_{\rm eff}$ (M-type)	✓	✓	✓	✓	✓	✓	✓	-	✓
K I doublet 7665, 7699 Å	$T_{\rm eff}$, $\log(g)$	✓	✓	✓	-	-	-	-	-	-
R1 8015–8130 Å	$T_{\rm eff}$ (M-type)	✓	✓	✓	-	-	-	✓	✓	-
Na I doublet 8183, 8195 Å	$T_{\rm eff}$ (M-type), $\log(g)$	✓	✓	✓	✓	✓	-	-	✓	✓
hydrogen Paschen series	$A_{\rm V}$	-	-	-	✓	✓	✓	✓	✓	✓
Ca II IR triplet 8498, 8542, 8662 Å	$T_{\rm eff}$ (early type)	✓	✓	-	✓	✓	-	✓	✓	✓
Mg I 8807 Å	$T_{\rm eff}$	-	-	-	-	✓	-	-	✓	-

lines are gravity-sensitive features for late-type stars but they are less essential for earlier types. In the case of Na I doublet 5890, 5896 Å lines, on the other hand, they are more important for hot stars rather than for cold stars to determine gravity.

Please note that the feature-importance tests presented in this section indicate the features that affect the network's judgment which is based on the Phoenix models. Some of the important features (that are essential for the network) behave very similarly to our knowledge, but some do not. Above all, the behaviour of the Na I doublet 8183, 8195 Å lines in the feature importance test agrees well with our knowledge. The Na I line, tracing the gravity (Riddick et al. 2007; Herczeg & Hillenbrand 2014; Manara et al. 2017) and the temperature of late-type stars (Kirkpatrick et al. 1991; Allen & Strom 1995; Riddick et al. 2007), is also essential for networks to determine stellar parameters of late-type stars and gravity. Based on Table 4.5, we find that the R1 8015–8130 Å, K I doublet 7665, 7699 Å, and Ba II, Fe I, and Ca I blend 6497 Å as well behave similarly to our knowledge. On the other hand, unlike our knowledge that the Ca II IR triplet 8498, 8542, 8662 Å and Mg I 8807 Å trace the temperature (Kirkpatrick et al. 1991; Allen & Strom 1995; Luhman et al. 2003; Manara et al. 2013; Herczeg & Hillenbrand 2014), the networks do not rely much on these lines to estimate the temperature.

In the feature-importance results of extinction, we showed the interesting results that there are particularly influential features although the extinction affects the overall shape of the spectrum, not the particular lines. One of the possible causes is the degeneracy between temperature and extinction. In our results, the features influential in determining the temperature tend to have high importance in extinction as well (e.g., Na I doublet 8183, 8195 Å, VO band, and Hα). Due to the degeneracy between the two parameters, the over- or under-estimation of temperature

can be compensated by over- or under-estimate of extinction. So, if the features important for temperature are scrambled, it can also affect the determination of extinction. Another possible cause is that the network determines extinction based on correlations between multiple features. For example, if the network relies on the ratios between several features to the Hα, Hα may have relatively higher importance than others, because scrambling the Hα affects all these ratios.

The feature importance only shows how much the error increases by scrambling a certain feature. Therefore, it is not easy to clearly understand the reasons for the error increment. Compared to the spectra of template stars, however, it is obvious that cINN captures important information from the point where absorption or emission exists. There are many features used to predict parameters besides the major features indicated in the figures or in the table, but the important point is that the most influential features are the same as the tracers we already know. This confirms that even though we do not exactly know how cINNs learn the hidden rules from the training data, what cINNs learned is very close to the physical knowledge we have.

4.7 Simulation gap and the best network

In sections 4.5.1 and 4.5.2, we showed that, for the synthetic models, our cINNs predict stellar parameters perfectly and for the template stars, network predictions are in good agreement with literature values within a 5 to 10 per cent error. The difference between literature values and network predictions slightly varies depending on the characteristics of the template stars. In sections 4.5.1 and 4.5.2, we confirmed that resimulation of the spectrum based on the network prediction well restored the original input spectrum. This means that the network successfully finds the most suitable model that satisfies the given observational data, as the network is designed to. In other words, the very good resimulation results indicate that cINNs provided us with the best results within the physics it has learned.

Interestingly, the resimulated spectrum based on the network prediction is closer to the original input spectrum than the resimulated spectrum based on the literature values for template stars (see Figure 4.6 and Table C.1), despite the discrepancy between the network prediction and literature value. This can be considered to be one of the following two cases. One is because there is a simulation gap, i.e., a gap between the physics within training data (i.e., the Phoenix atmosphere models), and the physics of the real world. The other is because of misclassification, meaning the literature value used as a reference in this paper is inaccurate. In the former case, no matter how perfectly trained the network is in terms of machine learning, it encounters inherent limitations. The simulation gap can be improved if we use better training data.

The three Phoenix libraries used in this paper reflect lots of important physics and characteristics of stellar atmosphere, but, of course, do not perfectly reflect reality. Therefore, we suspect that it is because of the simulation gap that the parameter predictions differ from the literature

values even though the resimulation results are almost perfect. In this section, we will introduce how we can quantify the simulation gap using the trained cINN and determine how large the gap is between the Phoenix models and reality. Finally, we will draw comprehensive conclusions about the performance and usage of our cINNs.

4.7.1 Quantifying simulation gap

As explained in section 4.2, cINN consists of the main network that connects parameters ($\mathbf{x}$) and latent variables ($\mathbf{z}$) and conditioning network (h) that transforms the input observation ($\mathbf{y}$) to the useful representative (i.e., condition, $\mathbf{c}$). Both are trained together, and the conditioning network in this paper compresses 2930 features ($y_1, \ldots, y_{2930}$) included in one spectrum into 256 conditions ($c_1, \ldots, c_{256}$). If the condition of the real observational data passed through the conditioning network ($\mathbf{c}_{\rm obs}$) follows the same probability distribution as the condition of the training data ($\mathbf{c}_{\rm train}$) this means there is no simulation gap. Because the conditioning network extracts only important features from the spectrum.

However, unlike the latent variables set up to follow a prescribed distribution (i.e., a standard normal distribution), the distribution of conditions does not follow a certain known distribution. Therefore, we build a network (k) that transforms the distribution of conditions ($p(\mathbf{c})$) into a prescribed probability distribution. The k network based on the cINN architecture is described as $k(\mathbf{c}) = \mathbf{s}$, and the output $\mathbf{s}$ is trained to follow a standard normal distribution. By definition of the cINN architecture, the dimensions of $\mathbf{c}$ and $\mathbf{s}$ are the same.

Using the conditioning network h and transformation network k, we check the simulation gap between the Phoenix models and template stars by comparing the distribution of the transformed condition of template stars $k(h(\mathbf{y}_{\rm tpl})) = \mathbf{s}_{\rm tpl}$ with the distribution of transformed condition of the training data $\mathbf{s}_{\rm train}$ which follows a known distribution. We evaluate the simulation gap based on the R^2 score between two probability distributions, $p(\mathbf{s}_{\rm train})$ and $p(\mathbf{s}_{\rm tpl})$. The bigger the R^2 value, the smaller the simulation gap.

4.7.2 Simulation gap

We trained transformation networks (k) for each cINNs (Settl-Net, NextGen-Net, and Dusty-Net) and compare the probability distribution of the transformed conditions of the training data and template stars. Figure 4.12 shows that the distribution of the training data (blue line) well follows the prescribed standard normal distribution (pink line) but the distribution of template stars (black) differs from that of the training data. There are 256 condition components for each star, but we present all these components in one distribution. The R^2 scores for all template stars are 0.805, 0.709, and 0.425 for Settl, NextGen, and Dusty, respectively. Dusty model seems to have the widest simulation gap, but we need to consider that Dusty-Net has a narrower training range than the parameter space of the template stars.

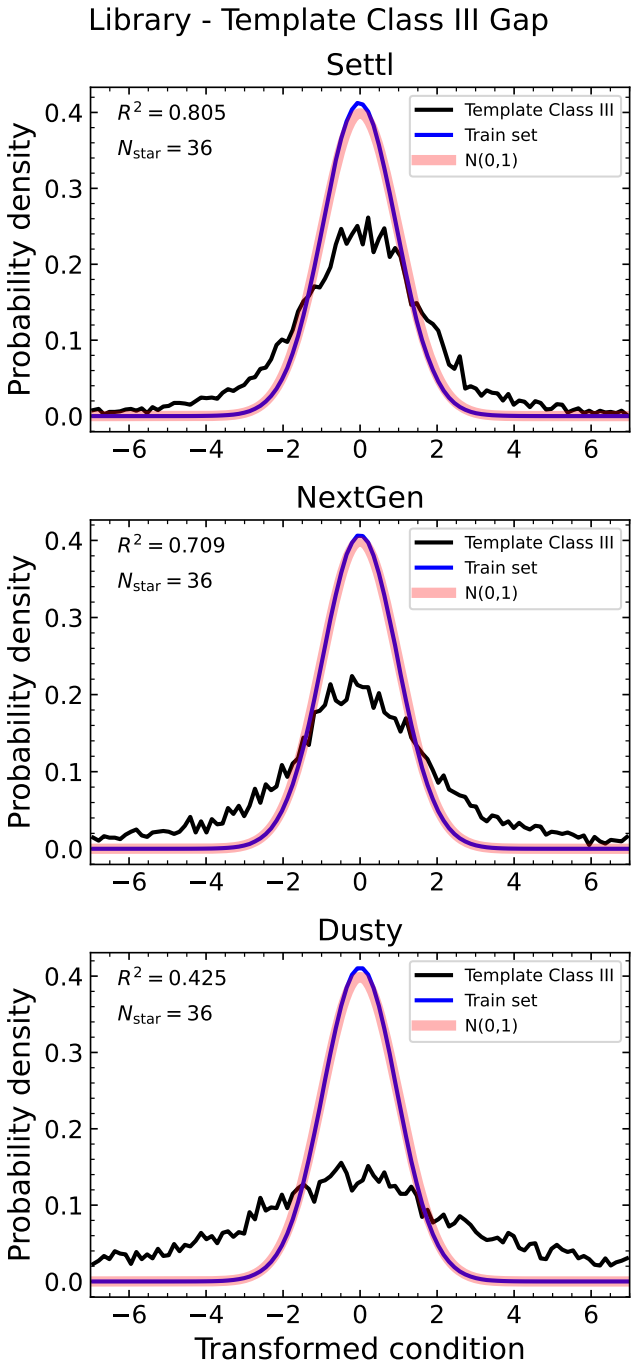

Figure 4.12: Probability distributions of transformed conditions of the training data (blue) and template stars (black) for three networks. The gap between the blue and black lines means the gap between the Phoenix model and the template spectrum. The R^2 value between the blue and black line and the number of template stars used are presented in the upper left corner of each panel.

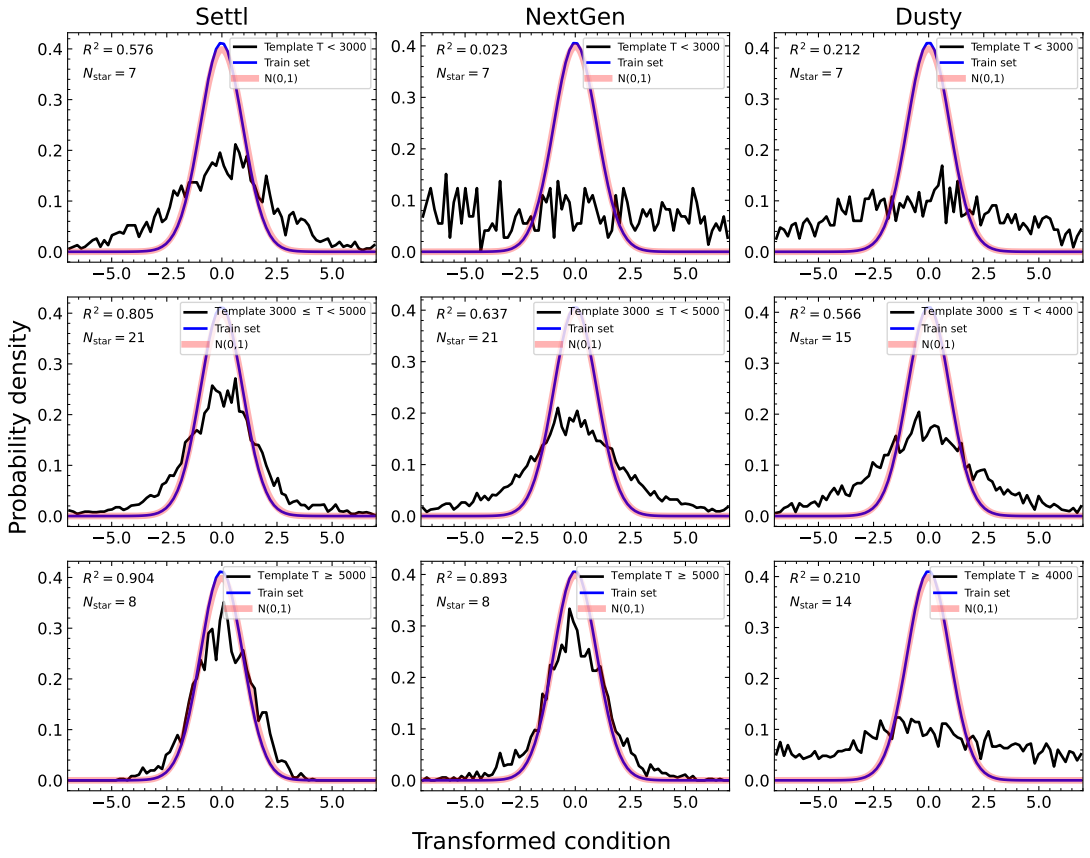

Figure 4.13: Probability distributions of transformed conditions of the training data and template stars. Each column represents three networks (Settl-Net, NextGen-Net, and Dusty-Net) and each row represents the group of template stars depending on their temperature ($T_{\rm eff}^{\rm lit}$). Colour codes are the same as in Figure 4.12.

As the performance of the cINN varies depending on the temperature of the template star, we divided the stars into three groups based on the prediction performance of the networks shown in section 4.5.2 (see Figures 4.4 and 4.5). For example, Settl-Net and NextGen-Net predicted parameters with good agreement with literature values, especially for stars with temperatures between ∼3000 K and ∼5000 K. So we divided the stars into 3 groups based on 3000 K and 5000 K for Settl-Net and NextGen-Net. In the case of Dusty-Net, due to the temperature upper limit of 4000 K for the Dusty training set, we divided groups based on 3000 K and 4000 K.

In the case of Settl and NextGen libraries (Figure 4.13), the earlier the spectral type, the smaller the gap and Settl has a smaller gap than NextGen in the overall temperature range. While the simulation gap is small for hot stars above 3000 K, the gap is large for later-type stars below 3000 K. In the case of NextGen, in particular, the simulation gap is very large for stars below 3000 K. In the case of Dusty, the simulation gap for the coldest group ($T < 3000$ K) is also very large and comparable to that for hot stars ($T > 4000$ K), out of the temperature range of the Dusty library.

The large gap for the lowest temperature group ($T < 3000$ K) is an obvious result because

perfectly implementing the atmosphere of late-type stars through the simulation is a much more difficult task than the earlier type stars. For late-type stars, condensation of vapour is important but the relevant physical processes are complex, making it very difficult to produce a good atmosphere model. Thus, these results demonstrate the inherent limitations of modelling low-temperature stars. These results show that the degree of simulation gap varies with the characteristics of the star, just as the difference between the prediction of cINN and the literature value varies, as shown in section 4.5.2.

Interestingly, both Settl and NextGen have the smallest simulation gaps for the early-type stars with temperatures above 5000 K. However, in Figures 4.4 and 4.5, the difference between the MAP prediction and the literature value of this group is slightly larger than that of the intermediate temperature group (3000–5000 K). The smallest simulation gap (Figure 4.13) and good resimulation results better than the resimulation of literature values (Figure C.5 and Table C.1) imply that MAP estimates of our networks for early-type stars above 5000 K are sufficiently reliable. Therefore, we suggest that the parameter estimations by our networks may be more accurate than the literature values for early-type stars above 5000 K.

4.7.3 Best network

It is clear that the simulation gap is large for late-type stars. Interestingly, however, our cINNs nevertheless predict the temperature and surface gravity well. First of all, all three networks had poor predictions of extinction for late-type stars below 3000 K. It is therefore very difficult for the network to estimate extinction accurately for stars in this temperature range, and the estimated extinction is not very reliable compared to the other two stellar parameters. However, Settl-Net, NextGen-Net, and Dusty-Net estimated the temperature accurately with maximum errors of less than 10, 5, and 15 per cent, respectively, despite the large simulation gap. This is a sufficiently accurate prediction considering the temperature interval between 1 subclass of stellar spectral type (see Figure 4.4). Using combined error in Figure 4.5, we demonstrated that Dusty-Net and Settl-Net predict surface gravity and temperature accurately within 5 per cent for late-type stars as well as early-type stars, despite the simulation gap of late-type stars. This shows that our networks are still applicable to low-temperature stars despite the limitations of training data. In the case of NextGen-Net, its performance was relatively poor for low-temperature stars compared to the other two networks, which is explained by the large simulation gap shown in Figure 4.13.

On the other hand, for earlier type stars with relatively small simulation gaps, the network performs more reliably. Except for one or two outliers, both Settl-Net and NextGen-Net accurately predict temperature and gravity within a maximum error of 5 to 10 per cent. NextGen-Net tends to estimate extinction and temperature slightly higher than Settl-Net. This seems that NextGen-Net is adopting a degenerate solution that satisfies the same input spectrum by increasing both extinction and temperature slightly. Overall, Settl-Net, with the smallest simulation

gap, shows the best performance among the three networks.

We conclude that Settl-Net is the best network considering both parameter prediction performance and simulation gap. Against low-temperature stars (e.g., M-type stars), Dusty-Net also shows comparable performance to Settl-Net. However, given that the stellar parameter coverage (i.e., temperature and gravity) of Settl-Net is wider than that of Dusty-Net, Settl-Net is more versatile and usable. Therefore, based on our overall results, we recommend using Settl-Net when applying the network to real observations. The only limitation to be cautious of is the estimation of extinction. Regardless of the spectral type of the stars, cINN estimates temperature and gravity accurately, but it should be cautious of using estimated extinction when the estimated temperature is below 3000 K.

4.8 Summary

In this paper, we introduce a novel tool to estimate stellar parameters from the optical spectrum of an individual young, low-mass star. cINN is one of the deep learning architectures specialised in solving a degenerate inverse problem. The degenerate problem here means that, due to the inevitable information loss during the forward process from the physical system to observation, different physical systems are mapped onto similar or almost identical observations. Many of the major tasks in astrophysics are solving degenerate inverse problems like estimating physical properties from observations. In this work, we develop a cINN for young low-mass stars to efficiently diagnose their optical spectra and estimate stellar parameters such as effective temperature, surface gravity, and extinction.

cINN adopts a supervised learning approach, meaning that the network is trained on the database consisting of numerous well-labelled data sets of physical parameters and observations. However, it is difficult to collect a sufficient number of well-interpreted observations in real. Therefore, we use synthetic observations instead to generate enough training data. In this work, we utilise three Phoenix stellar atmosphere libraries (i.e., Settl, NextGen, and Dusty) to produce the database for training and evaluation of the network. Interpolating the spectrum on the temperature – gravity space and adding the extinction effect on the synthetic spectra, we produce a database for each Phoenix library consisting of 65,536 synthetic models. To produce databases, we randomly sampled three parameters from the given parameter ranges. Settl and NextGen databases cover the temperature range of 2600–7000 K and $\log(g/\mathrm{cm\,s^{-2}})$ range of 2.5–5. In the case of the Dusty database, it covers the temperature of 2600–4000 K and $\log(g/\mathrm{cm\,s^{-2}})$ of 3–5. All three databases have extinction values within 0–10 mag. Then, we build and train cINNs using each database but only use 80% of the synthetic models in the database to train the network and remain the rest for evaluation. In this paper, we present three cINNs that learned about different Phoenix atmosphere models: Settl-Net, NextGen-Net, and Dusty-Net.

We validated the performance of our cINNs in various methods. Our main results are the following:

1. All three networks provide perfect predictions on the test set with the RMSE of less than 0.01 dex for all three parameters, demonstrating that the cINNs are well-trained. Additionally, we resimulate the spectrum using the parameters estimated by the network using our interpolation method and compare it with the original input spectrum. The resimulated spectra perfectly match the input spectra of the test models with RMSE of about 10^{-7}. These results prove that our three cINNs perfectly learned the hidden rules in each training data.

2. To test the performance on the real observational data, we analyse 36 Class III template stars well-interpreted by Manara et al. (2013, 2017); Stelzer et al. (2013) with our cINNs. We demonstrate that stellar parameters estimated by our cINNs are in good agreement with the literature values.

3. Each network has a slightly different error depending on the temperature of the given star. Settl-Net works especially well for M6.5 – K1.0 (2935 – 5000 K) stars and NextGen-Net works well for M4.5 – K1.0 (3200 – 5000 K) stars. Dusty-Net works well for M5.5 – M0.0 (3060 – 4000 K) stars. Given that the temperature upper limit of Dusty training data is 4000 K, Dusty-Net works well for stars within its training range. For stars in other temperature ranges, three networks perform well with an error of less than 10 per cent.

4. The most difficult parameter for cINNs to predict is the extinction of cold stars with temperatures less than 3200 K. All three networks tend to estimate extinction higher than the literature value for cold stars. However, cINNs estimate extinction well for hot stars with temperatures above 3200 K.

5. We resimulate spectra based on cINN estimations and literature values and compare them with the original input spectrum. Interestingly, most of the resimulated spectra based on cINN estimations are closer to the input spectra than the resimulated spectra derived from literature values. This implies that our cINNs well understand the physics in each Phoenix library and are able to find the best-fitting Phoenix model (i.e., parameters) for the given observation.

6. Results that the resimulations are perfect even though the prediction of the network is slightly different from the literature can be explained by a gap between the Phoenix model and reality, so-called the simulation gap. We quantify the simulation gap between each library and template stars using the conditioning networks included in our cINNs. We confirm that the simulation gaps are relatively large for cold stars below 3000 K where the

cINNs have difficulty estimating extinction. We confirm that the simulation gap is small for hot stars where cINNs predict parameters well.

7. The overall results imply that although there is an obvious gap between the Phoenix model and reality, especially for cold stars below 3000 K, our networks can nonetheless provide reliable predictions for all stars within 5–10 per cent error, especially for temperature and gravity. Extinction estimated by cINN is also reliable unless the estimated temperature is less than 3200 K.

8. We investigate which parts of the spectrum cINN relies mostly upon to predict stellar parameters and compare the important features with typically used stellar parameter tracers. We find that cINN relies on different features depending on the physical parameters and on the input observations (e.g., spectral types). We confirm that the major features are equivalent to the typically used tracers such as Hα 6563 Å and Na I doublet 8183, 8195 Å.

Our overall results show that our cINNs present reliable enough performance applicable to real observational data. Among the three networks introduced in this paper, we recommend Settl-Net trained on the Settl library as the best network because of its remarkable performance and versatility on the parameter space.

5

Conclusions

We aimed to develop deep learning tools that effectively and efficiently analyse observed star-forming regions (i.e., physical parameters) to study star formation. Adopting the cINN architecture, we presented networks that analyse cloud-scale observations and networks that analyse individual star-scale observations, and carefully evaluated their performance. Here, we summarise three studies presented in this thesis.

5.1 Summary

cINNs for cloud-scale observations

In Chapter 2, we introduced the first version of the network that analyses cloud-scale observations of star-forming regions. We built and trained the network that provides a full posterior distribution of seven physical parameters using the luminosity of twelve optical emission lines commonly observable in H II regions as the input of the network. The seven physical parameters are initial cloud mass, star formation efficiency, initial cloud number density, age of the first generation cluster, age of the youngest cluster, the number of clusters and the evolutionary phase of the cloud. To train the network, we produced a database of numerous synthetic H II region models by using the WARPFIELD-EMP pipeline (Pellegrini et al. 2020) that is based on a one-dimensional stellar feedback code, WARPFIELD (Rahner et al. 2017). We split the database into train and test sets for network training and performance evaluation. Using the unlearned synthetic models (test set), we demonstrated that our network provides a very accurate and precise posterior distribution for the seven physical parameters. We further validated the network performance by resimulating the posterior models (i.e., sets of physical parameters estimated by the network) through the WARPFIELD-EMP pipeline and confirming that posterior models satisfied the original input luminosity. Sometimes networks provided a degenerate posterior distribution with a multi-modal shape. We statistically analysed degenerated posterior

distributions and found out that most of the degeneracy occurs because of the degeneracy in the number of clusters prediction. The number of clusters and the age of the first generation cluster (i.e., the oldest cluster) are the most difficult parameters for the network to accurately predict because of the following two reasons. The first reason is that the contribution of the older generation clusters to the luminosity of the 12 emission lines we selected is much smaller than that of the youngest generation cluster so it is hard to distinguish how many old generation clusters are there. The second reason is that parameter distributions of our database are biased toward young and single cluster H II regions rather than old or multi-cluster models so the network learned relatively better for young and single cluster H II regions. Additionally, we introduced the method of obtaining a posterior distribution considering the observational uncertainties by modifying the posterior sampling method of the trained network and showed how the posterior distribution changes as a function of the observational uncertainties.

In Chapter 3, we introduced Noise-Net, an updated version of the cINN for cloud-scale observations introduced in Chapter 2. The Noise-Net uses not only the luminosity of the 12 emission lines but also the uncertainty of the luminosity measurement as the input of the network and learns the influence of the error on the parameter prediction during the training. Consequently, the Noise-Net always provides a posterior distribution reflecting the influence of the observational uncertainties. As we already introduced the method of sampling the posterior distribution considering uncertainties of observation using the first version of the network, which we referred to as Normal-Net, in Chapter 2, we mainly compared the performance of Noise-Net and Normal-Net as a function of the luminosity error. We showed that the Normal-Net performs slightly better when the error is very small, but after the turning point, the Noise-Net outperforms the Normal-Net. The turning point occurs when the error of the brightest emission line (usually the smallest error) is around 0.025–0.04 per cent and the performance gap between the two networks significantly widens as the error increases. Since the error value of the turning point is sufficiently small compared to the usual amount of errors measured from the real observational data, we concluded that Noise-Net would be a better choice when we apply our networks to the real observational data. Through further investigations, we found that Normal-Net tends to give unphysical posterior estimates when the error is large whereas Noise-Net provides physically valid estimates even though the posterior distribution becomes wider or degenerate.

cINNs for individual young stars

In Chapter 4, we introduce the cINNs for individual stars that diagnose the optical spectrum of young low-mass stars and predict three stellar parameters: effective temperature, surface gravity, and extinction. In this work, for the network training, we adopt a Phoenix stellar atmosphere model (Allard et al. 2012) that contains both physical parameters and spectral information, selecting three different Phoenix libraries: Settl, NextGen, and Dusty. We produced a database

for each library and train three networks based on different databases: Settl-Net, NextGen-Net, and Dusty-Net. Our networks are trained on young stars with a temperature range of 2600 to 7000 K and a surface gravity ($\log(g/\mathrm{cm\,s^{-2}})$) range of 2.5 to 5. In the same way, as in the previous two studies (Chapters 2 and 3), we evaluated the performance using the unlearned synthetic models and confirmed that three networks perform perfectly without any degeneracy in the posterior distributions. Furthermore, we applied our networks to the real observational data of Class III stars whose stellar parameters are well-characterised by literature. By comparing the known values of stellar parameters with our network estimations, we confirmed that our network estimation agrees well with the literature values within the error of up to 5–10 per cent. The performance of the networks slightly varies depending on the spectral type, and our networks showed the best performance for stars in the temperature range of 3000 to 5000 K corresponds to M6–K1 type. Despite a small discrepancy in the stellar parameters, the resimulated spectrum of the posterior model was very close to the input spectrum. We found out that the resimulations of the posterior models were even closer to the input spectrum than the resimulations based on literature values.

In this study, we conducted two additional experiments using the trained networks. The first one is the feature importance test that examines the influence of each spectral part (input of the network) on the prediction of each stellar parameter. We compared the important features of the network with stellar parameter tracers commonly used to analyse observation data and found that most of those tracers play important roles in the network to estimate stellar parameters as well. In the second experiment, we quantified the gap between the theoretical model (synthetic observations, Phoenix models in this study) used to train the network and the real observations and investigated how this gap, which we referred to as the simulation gap, affects the network performance. Through this study, we confirmed that even though there is a clear gap between reality and the theoretical models and the gap is related to the network performance, our networks nevertheless are able to estimate the stellar parameters accurately.

5.2 Discussion and future works

Through these three studies, we presented a methodology and applicability of using cINN architecture (deep learning technique) to analyse observational data. We introduced two case studies, cINNs for cloud-scale star-forming region observations by utilising WARPFIELD-EMP synthetic H II region models and cINNs for individual stellar spectra by using Phoenix stellar atmosphere models. However, cINN architecture can be utilised to interpret various observations if it is coupled with proper theoretical models. Here, we summarise the strengths and limitations of applying cINN to astrophysics.

In this thesis, we demonstrated that cINN can understand the hidden rules within the complex

processes of star formation and link observations with physical properties. One of the advantages of adopting cINN is that cINN provides a full posterior distribution of physical parameters rather than simply estimating a single value. In addition, once trained, neural networks are very time-efficient tools that can quickly and consistently analyse large amounts of observations. For example, our cINNs for cloud-scale H II region observations (Chapters 2 and 3) can analyse 170 observations per second. As we train the network based on the theoretical model, it is possible to build a network that estimates physical properties that are hardly measured by classical methods. For instance, we can build a network predicting initial conditions (e.g., the initial density of the star-forming region) or the age of the cluster because theoretical models describe the overall evolution of the object. Additionally, the cINN can estimate the desired parameters even from data with an insufficient resolution to use classical analysis methods. Our networks for individual young stars (Chapter 4) can measure the surface gravity although the networks use the stellar spectrum with low spectral resolution insufficient to measure the surface gravity with classical methods. Moreover, we showed in Chapter 3 that cINN makes sufficiently good predictions even when taking into account considerable observational error. Trained networks also can be used for various experiments such as finding significant observables through feature importance tests or evaluating the theoretical model by measuring the simulation gap.

However, there are some limitations to applying cINNs to solve astrophysical problems. Since we use theoretical models to collect sufficient training data, it is not possible to avoid the limitations inherent in the theoretical models just like other methods that analyse observations through theoretical models. There is no theoretical model that perfectly mimics reality, which inevitably results in the gap between the model and reality. Although cINN can perfectly understand the hidden rules in the theoretical model if well-trained, the network cannot overcome the inherent limitations of the theoretical model. This gap is likely to become larger as the scale of astronomical objects enlarges because more physical assumptions are used and the ranges of the parameters are limited when modelling larger-scale observations. As the number of physical processes and parameters to be considered increases, the simulation of the forward process becomes computationally expensive, making it difficult to generate enough training data.

Although there are limitations in our method but these are common difficulties experienced in all methods of interpreting observations with theoretical models. Moreover, in Chapter 4, we showed that cINN can play a sufficient role in estimating physical parameters even though such a gap between the theoretical model and reality exists. Therefore, cINN is a good enough tool applicable to interpret observational data.

We will continue working on follow-up studies with the development of relevant theoretical models. The cINN for young stars (Chapter 4) will be applied to young stars in the Carina Nebula observed with VLT/MUSE. In this follow-up study, we plan to update our theoretical models (Phoenix model) by adding more physical parameters such as the influence of veiling.

The cINNs for the cloud-scale observations introduced in Chapters 2 and 3 (both Normal-Net and Noise-Net) were first applied to thousands of H II regions observed from Physics at High Angular resolution in Nearby GalaxieS–Multi Unit Spectroscopic Explorer (PHANGS) survey (Santoro et al. 2022; Emsellem et al. 2022). We compared the stellar mass measured by Lee et al. (2022) based on the HST observations and stellar mass estimated by our networks and found that network estimations were on average larger by 1 dex. The reasons for the large discrepancy we discussed were the following two. Firstly, based on the location of the training data and observation data on the BPT diagram, the metallicity assumption used for our training data did not match well with the real metallicity of the observations. Secondly, the clusters of the training data were much more massive than the average stellar mass of the observed H II regions. We concluded that we need to update the WARPFIELD-EMP pipeline to generate the proper training data for these observations. We expect that we can obtain better results when the training data is updated and can implement various experiments (e.g., measuring the simulation gap and examining feature importance) using the new networks.

A

Appendix for Chapter 2

A.1 Supplemental materials

The following are supplementary figures of the training evaluation of our main network (Section 2.4.1) and the performance of Network 2 without noise augmentation (Section 2.8.2). Figure A.1 shows the covariance matrix and the probability distributions of latent variables, which are obtained from the forward process of the network for the entire test set ($\mathbf{Z_{test}}$). As mentioned in Sections 2.3.3 and 2.4.1, latent variables should follow the prescribed Gaussian normal distribution and their covariance matrix should be close to a unit matrix if the network is well-trained. Though the covariance matrix and probability distributions are not perfectly equal to the desired results, residuals are small enough to conclude that the network converged to a good solution.

Figures A.2 and A.3 show the prediction performance of Network 2 introduced in Section 2.8.2. We did not augment the distributions of N_{cluster} and phase with artificial noise in training Network 2. After drawing 4096 posterior samples for each model in the entire test set, we compare the true values of the models with all posterior estimates or with the MAP estimates as the representative in Figure A.2 and Figure A.3, respectively. As already discussed in the previous section, adding artificial noise to smooth out the discretized parameter distribution improves the overall prediction performance of the network.

A.2 Fitting and determining peaks of the posterior distribution

In this appendix, we provide an example of the posterior fitting result and the histogram of the number of peaks using different definitions of the peak. We present in Figure A.4 the results of fitting the posterior distribution of one model in the test set. As mentioned in Section 2.6, we fit

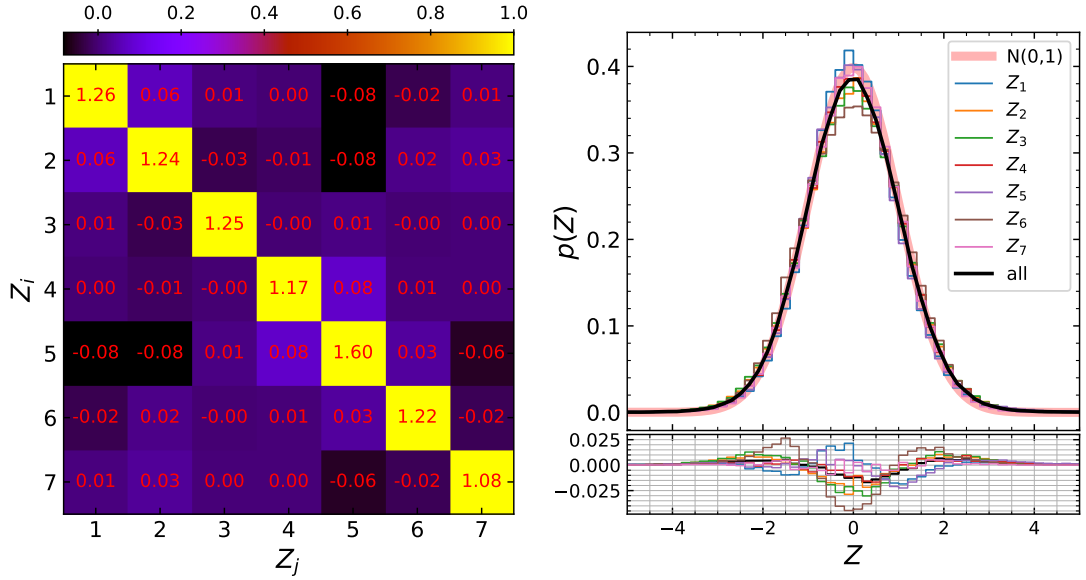

Figure A.1: The covariance matrix of the latent variables (left) and distributions of each latent variable (right) as evaluated on 101,149 test set models. In the right panel, the black line shows the distribution of all seven latent variables in one line and the red line indicates the standard normal distribution. The small panel below shows residuals between distributions of latent variables and the standard normal distribution. If the network is well-trained, the covariance matrix is close to the unit matrix and the distribution of latent variable follows the standard normal distribution.

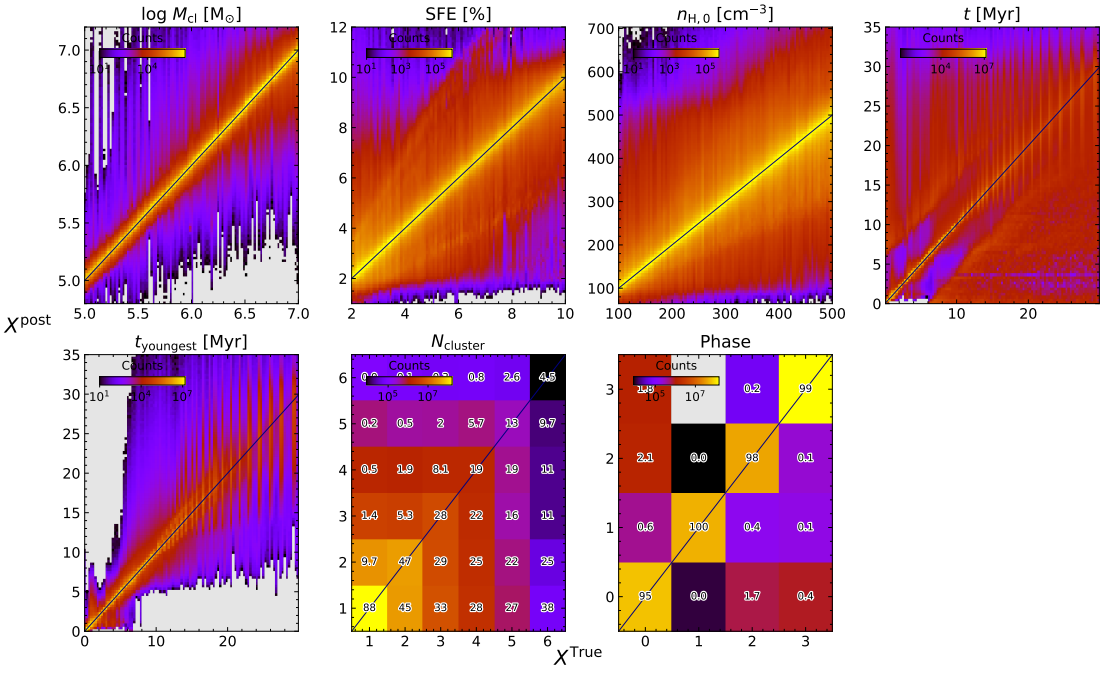

Figure A.2: Comparison of true parameter values and all posteriors obtained from the cINN model trained without noise augmentation (Network 2), using the entire 101,149 test models. Figure 2.6 has a similar but better result which is obtained from our main cINN model, trained with noise augmentation. Colour code indicates the counts of the 2D histogram.

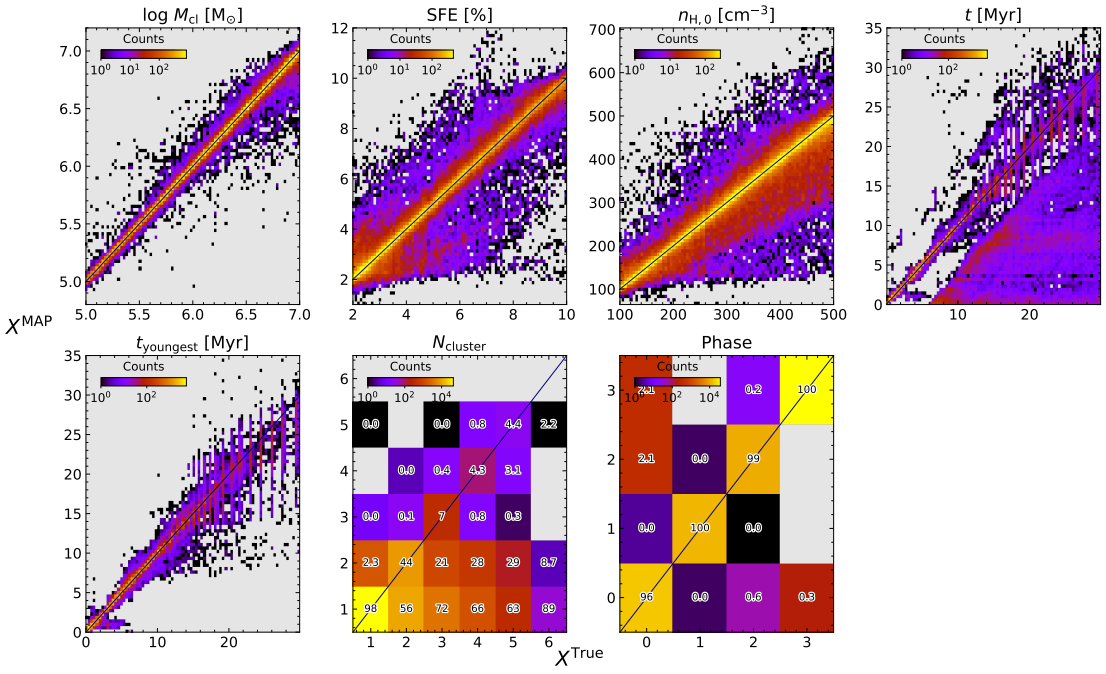

Figure A.3: Comparison of true parameter values and MAP estimates obtained from the cINN model trained without noise augmentation using all 101,149 test models. Figure 2.7 shows the similar result obtained from our main cINN model. Colour code is the same as in Figure 2.7.

the distribution using up to 6 Gaussian functions, and the number of Gaussian functions used varies even in one observation. In addition to the final fitting curve (blue solid line), individual Gaussian components used in the fitting are shown with green dotted lines. Considering the posterior distribution as a multimodal distribution, we provide the number of peaks according to the three different definitions of the peak of the mode. In Figure A.4, the number of Gaussian components (n_g), the number of visible peaks (i.e., the number of red circles, n_v), and the number of separated peaks (n_s) are all the same for M_{cl}, star formation efficiency, and the cloud age. The case of $n_{H,0}$ shows that even if the distribution is fitted with two Gaussian components, it is considered as one visible peak or one separated peak. On the other hand, the case of the youngest cluster age is where two Gaussian components are considered as one visible peak but two separated peaks depending on the distance between two Gaussian components.

In Figures A.5 and A.6, we present the histogram of the number of peaks using the separated peaks and the number of Gaussian components, respectively. We exclude the result of phase because the number of peaks in the phase posterior distributions does not depend on the definition of the peak so it is the same as shown in Figure 2.12. These results differ from those obtained using the visible peaks in detail, but the overall trends are similar to Figure 2.12, including the influence of the degenerate $N_{cluster}$ prediction on the degeneracy in the posterior distribution of other parameters.

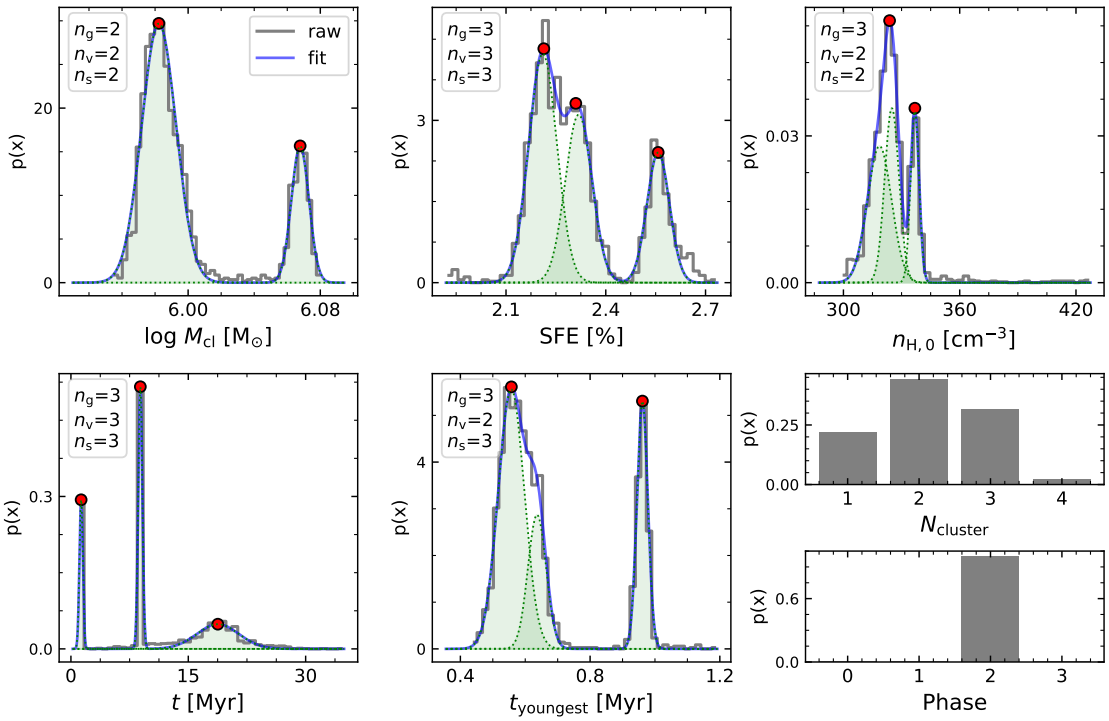

Figure A.4: An example of fitting posterior distributions for five parameters. The posterior distribution (grey histogram) is fitted with multiple Gaussian functions. The blue solid line represents the fit result and the green shades with dotted edges represent individual Gaussian components of the fitting curve. We do not fit for $N_{\rm cluster}$ and phase. The number of modes in the posterior distribution measured according to the three different definitions of the peak of the mode is listed on the upper left side of each panel: the number of Gaussian components used for the fitting ($n_{\rm g}$), the number of visible peaks ($n_{\rm v}$), and the number of separated peaks ($n_{\rm s}$). The visible peaks of each posterior distribution are denoted by red circles.

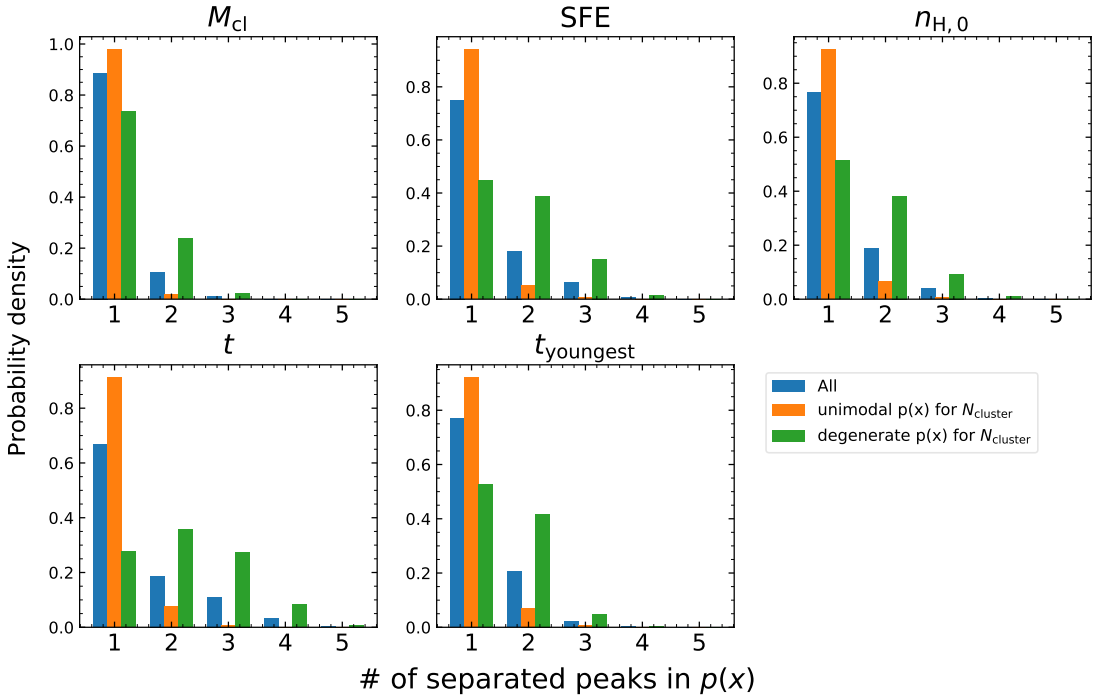

Figure A.5: Density histograms of the number of separated peaks in the posterior distributions for five parameters. The colour codes are the same as in Figure 2.12: results of all models in the test set (blue), results of models whose $N_{\rm cluster}$ posterior distributions are not degenerate (orange), and results of models whose $N_{\rm cluster}$ posterior distributions are degenerate (green).

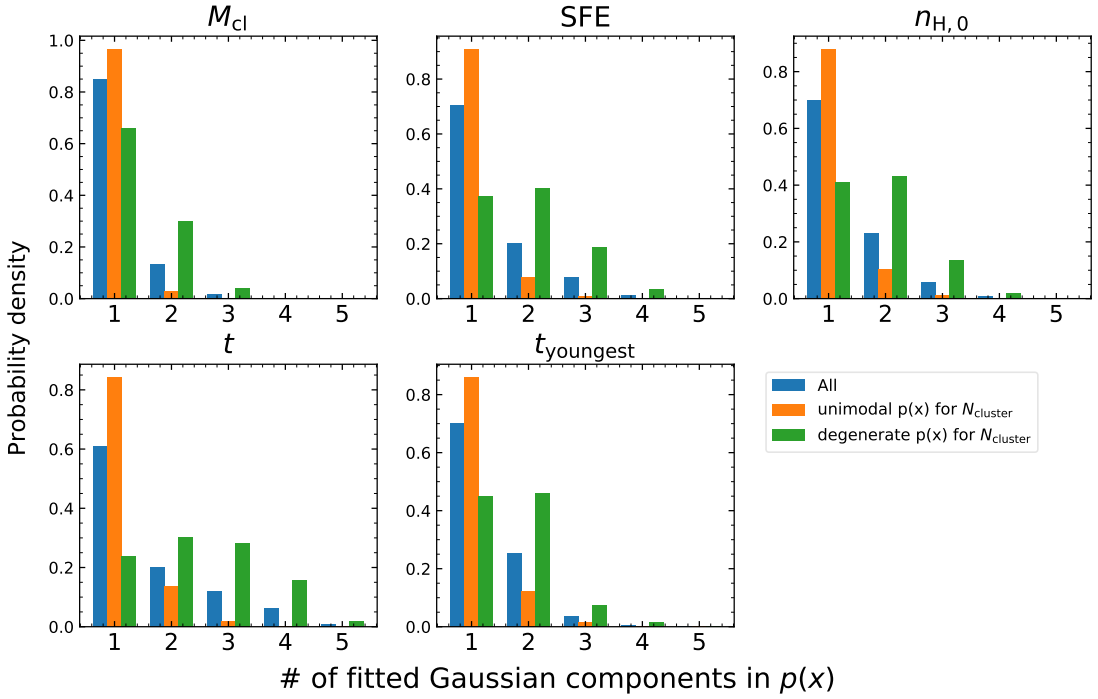

Figure A.6: Density histograms of the number of Gaussian components used in the fit of posterior distributions for five parameters. The colour codes are the same as in Figure 2.12 and Figure A.5.

B

Appendix for Chapter 3

B.1 Supplemental materials

In this appendix, we present supplementary figures for the error clipping tests for the Noise-Net and the Normal-Net discussed in Section 3.4.3. We re-sampled the posterior estimates for the 100 test models for 16 different σ_{b} levels after clipping the luminosity errors larger than the maximum training error (σ_{max}) of 31.6%. We present the average performance differences between the original unclipped result (Figure 3.3) and the clipped result of the Noise-Net in Figure B.1 and of the Normal-Net in Figure B.2. As mentioned in Section 3.4.3, the deviation between unclipped and clipped results is very small for both the Noise-Net and the Normal-Net.

B.2 Deeper networks

In this appendix, we provide the results of additional experiments that investigate the influence of network depth on prediction performance. Here the depth of the network refers to the number of affine coupling blocks (N_{block}) and the number of layers of the internal sub-network (N_{layer}). As the Noise-Net processes more information than the Normal-Net by definition, we investigate whether network performance improves by increasing the depth of the network. The networks used in the main paper are the best choices based on the results of this experiment.

In Paper 1, we used eight affine coupling blocks and internal sub-networks with three layers to build the cINN (N_{block}=8, N_{layer}=3). Before the experiment, we confirmed that the Noise-Net performed better than the Normal-Net in general, similar to the results in Section 3.3.2, even when we use the setup of Paper 1. However, with the possibility of performance improvement in mind, we investigated the influence of N_{block} and N_{layer}. To evaluate the network performance we used the same 100 test models and the same methodology as in Section 3.3. For each network, we sample posterior distributions at 16 different σ_{b} for the 100 test models and measure two

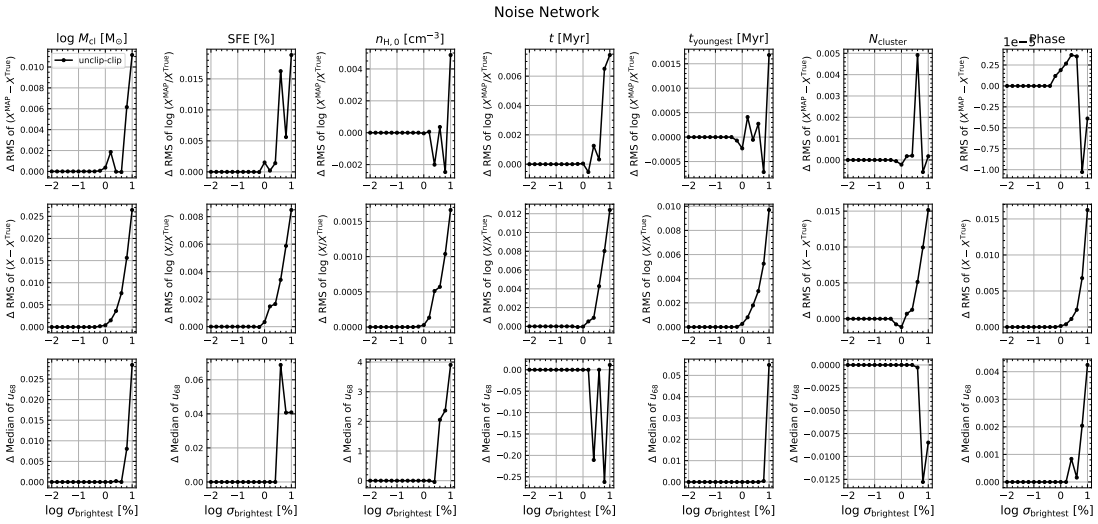

Figure B.1: The average performance difference of the Noise-Net between the original result (blue line in Figure 3.3) and the results after error clipping. For errors larger than the training range of the Noise-Net, we clip the error to the maximum value (31.6%) and re-sample the posterior distributions for all test models.

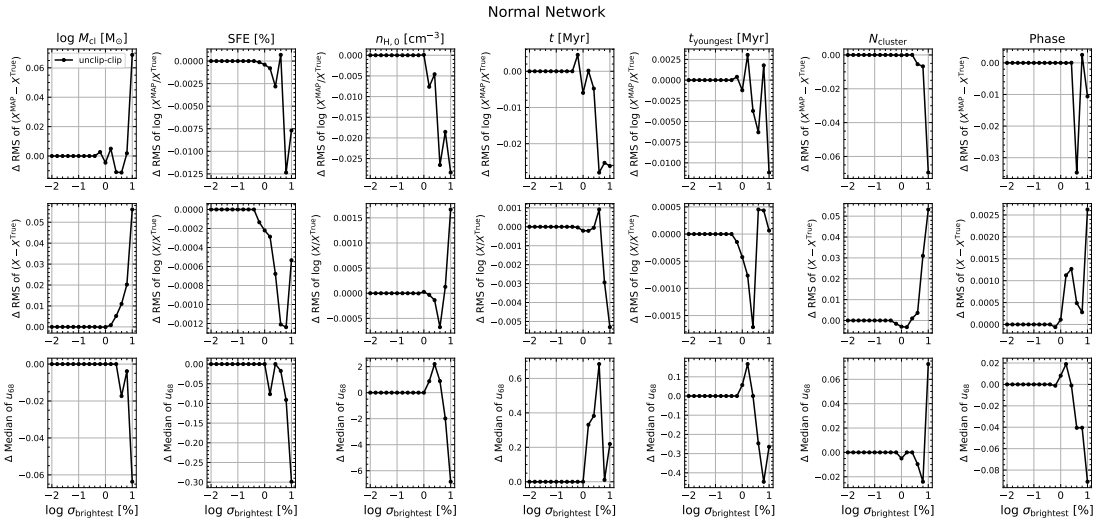

Figure B.2: Similar to Figure B.1, we present the average performance difference of the Normal-Net between the original result (red line in Figure 3.3) and re-sampled posterior distributions after clipping the large errors.

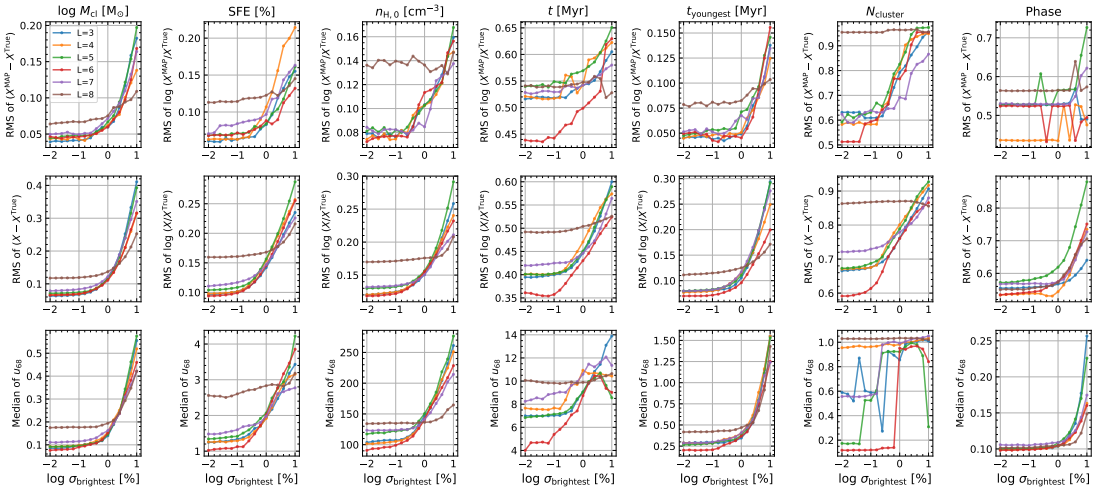

Figure B.3: We compare the two accuracy indices (the first and the second row) and precision index (the third row) of six Noise-Nets for different numbers of layers in the internal sub-network. All six networks have 8 affine coupling blocks. We use the same sample of 100 test models and the same 16 different errors as used in Section 3.3.

accuracy indices and the precision index for each posterior distribution. We compare the average performance of the network as a function of σ_b like the curves presented in Figure 3.3.

In the first experiment, we train and compare 6 Noise-Nets by increasing the N_{layer} from three to eight (Figure B.3). For these six networks, we fix N_{block} at eight. Except for the worst network with eight layers (brown curve), the performance of the other networks is similar overall. It is not easy to identify the relation between the performance and the number of layers because the performance difference between networks slightly varies depending on the parameters and the range of the error. However, the performance gap between the network with six layers (red curve) and the rest is noticeable in the case of the oldest cluster age (t), especially when the error is small. For the other parameters, the red curve shows similar or slightly better results than the others. As we discovered in our first study (Paper 1), the age of the oldest cluster is the most difficult parameter for our network to predict, so the improvement in t prediction is meaningful. On this account, we choose the network with 6 layers as the best result of the first experiment.

In Figure B.3, we present the results of 6 networks but we also tried to train the network with 9 layers. However, the latter experiment failed because the loss (Eq. 3.6) did not decrease at all. In addition to the results for the networks with more than 6 layers in Figure B.3 (purple and brown), this implies that increasing the number of layers does not always improve the prediction.

In the second experiment, we increase the number of affine coupling blocks (N_{block}) and fix the N_{layer} to three. In Figure B.4, we compare the performance of three networks with N_{block} of 8, 12 and 16, respectively. All three networks perform similarly, but the network with 16 blocks (green curve) performs the best, especially in the case of the oldest cluster age (t). This network performs better than the other two at σ_b smaller than 1% but performs poorly at large errors.

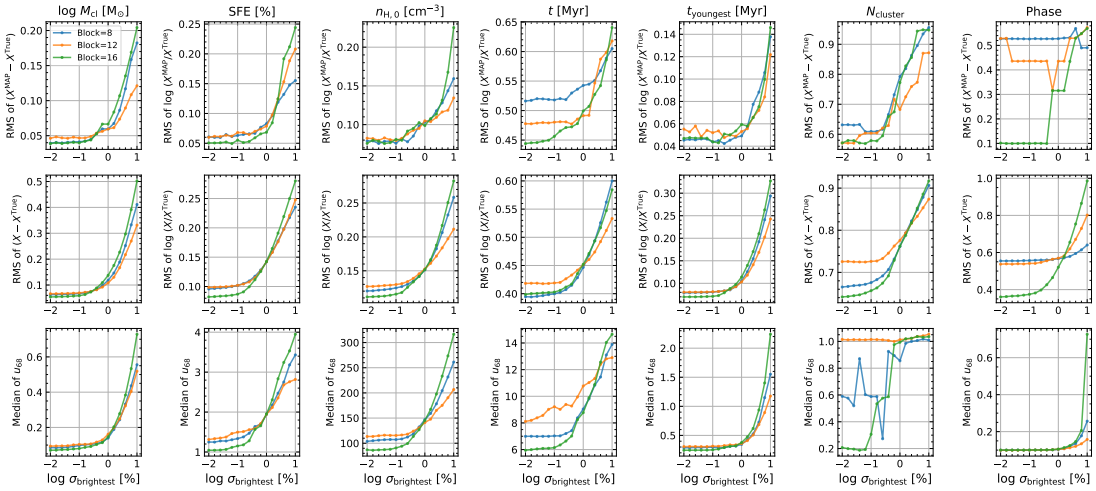

Figure B.4: Comparison of the performance of 3 Noise-Nets with the different numbers of affine coupling blocks. All three networks use 3 layers in their internal sub-network.

As in the first experiment, we focus on improving the t prediction and choose the network with 16 blocks as the best result.

Lastly, we build the deepest cINN combining the best options from the previous two experiments ($N_{\text{layer}} = 6$ and $N_{\text{block}} = 16$) and compare the performance of the following four networks: the network with the Paper 1 setup (N_{layer}=3, N_{block}=8), the best network of the first experiment (N_{layer}=6, N_{block}=8), the best network of the second experiment (N_{layer}=3, N_{block}=16) and the deepest network (N_{layer}=6, N_{block}=16). In Figure B.5, the deepest network (red curve) performs the best in the case of M_{cl}, SFE, $n_{\text{H},0}$ and t_{youngest}, but the gap between the curves is not large. However, the deepest network shows a noticeable improvement in the oldest cluster age prediction and the accuracy of the N_{cluster} prediction. In the case of the phase, the network with 3 layers and 16 blocks (green curve) performs the best, but considering the physical unit of the phase by definition, the performance of the other networks is still acceptable. Taking into account the overall performance, we choose the deepest network with 6 layers and 16 blocks as the best network and use this network for the main paper. So, the red curve in Figure B.5 is the same as the blue curve in Figure 3.3.

Comparing the various Noise-Nets with different N_{layer} and N_{block}, we found the following. Firstly, deepening the network can improve the prediction power, but it does not always guarantee improvement. Moreover, the performance does not change proportionally to the depth and training may fail if we deepen the network too much. Secondly, in the case of parameters that the cINN predicts relatively well (e.g., M_{cl}, t_{youngest}), there is no significant change depending on the network depth, but deepening the network improves the performance in the case of t and N_{cluster} which are relatively difficult for the cINN to accurately predict due to the degeneracy. In Paper 1, we demonstrated that our cINN has difficulty resolving the degeneracy in N_{cluster} prediction because of the biased parameter distribution of the training data and our selection of

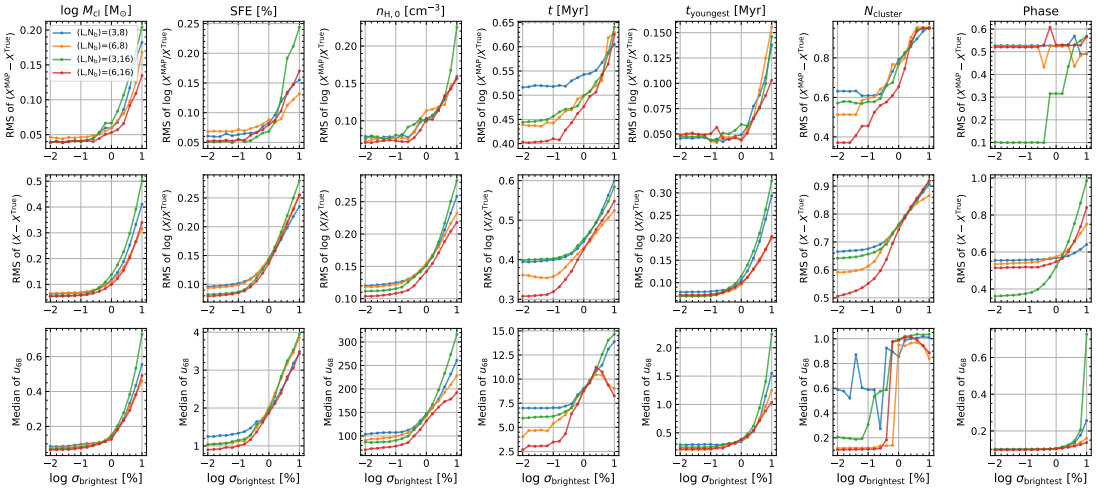

Figure B.5: Comparison of the performance of four Noise-Nets with different combinations of the number of affine coupling blocks and the number of layers of the internal sub-network. The red line with 6 layers and 16 blocks is the same Noise-Net used in the main paper (i.e., the blue line in Figure 3.3).

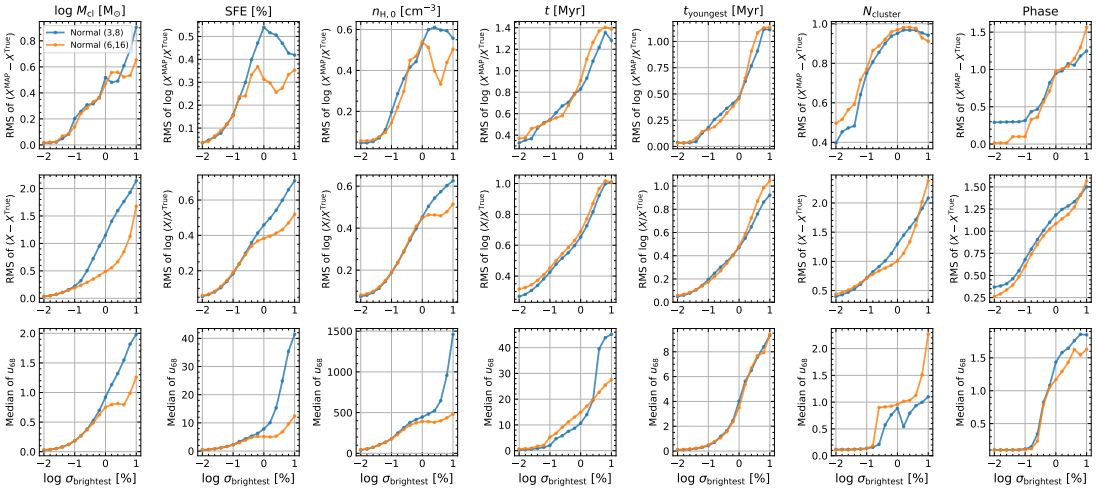

Figure B.6: Comparison of two Normal-Nets with different numbers of blocks and layers of the internal sub-network. The orange line with 6 layers and 16 blocks is the same Normal-Net used in the main contents (i.e., the red line in Figure 3.3).

optical emission lines. This result implies that we can enhance the degenerate prediction of the Noise-Net by properly increasing the network depth.

Additionally, we investigate whether the performance of the Normal-Net changes when increasing N_{layer} and N_{block}. We only compare two networks using the setup of Paper 1 (N_{layer}=3, N_{block}=8) and the setup of the best Noise-Net (N_{layer}=6, N_{block}=16). Figure B.6 shows that the overall change of the Normal-Net is different to the case of the Noise-Net. The performance gap between the two networks becomes noticeable for M_{cl}, SFE and $n_{\mathrm{H,0}}$, when the error is large. On the other hand, there is no significant change in the N_{cluster} and t prediction and sometimes the deeper network performs slightly worse than the other. In conclusion, we choose the deeper Normal-Net (orange curve) and keep the network setup the same as the Noise-Net for the main paper because the performance of the deeper Normal-Net is not significantly worse, but rather exhibits some improvement for some parameters.

C

Appendix for Chapter 4

C.1 Supplemental materials

In this appendix, we present supplementary figures and table mentioned in our main results (sections 4.5–4.6).

C.1.1 Prediction performance

We evaluate the performance of NextGen-Net and Dusty-Net on 13,107 synthetic test models by comparing the MAP predictions from the network and the true values of the models. Figures C.1 and C.2 show that both networks estimate all three parameters perfectly with very small RMSEs.

C.1.2 Resimulation

We validate the cINN predictions on both the synthetic test data (Section 4.5.1) and real template spectra (Section 4.5.2) by resimulating the spectra corresponding to the MAP estimates with our spectral library interpolator (Section 4.3.1) and comparing the result to the respective input spectra.

Analogous to Figure 4.2, Figures C.3 and C.4 show the median relative error of the resimulated spectra (left panel) and distributions of the RMSEs (right panel) for the 13,107 synthetic test spectra when evaluated with the cINN models trained on NextGen and Dusty, respectively.

Table C.1 provides a summary of the resimulation results for the cINN predictions on the Class III template spectra (see Section 4.5.2 and also Tables 4.1 and 4.3). Here we list the RMSEs and R^2 scores of the resimulated spectra with respect to the corresponding input spectra for the resimulation based on the literature and cINN-predicted parameters for all three spectral libraries.

Figures C.5 and C.6 provide additional examples of the resimulation results, comparing the resimulated spectra to the input spectra and the outcomes between the three libraries, anal-

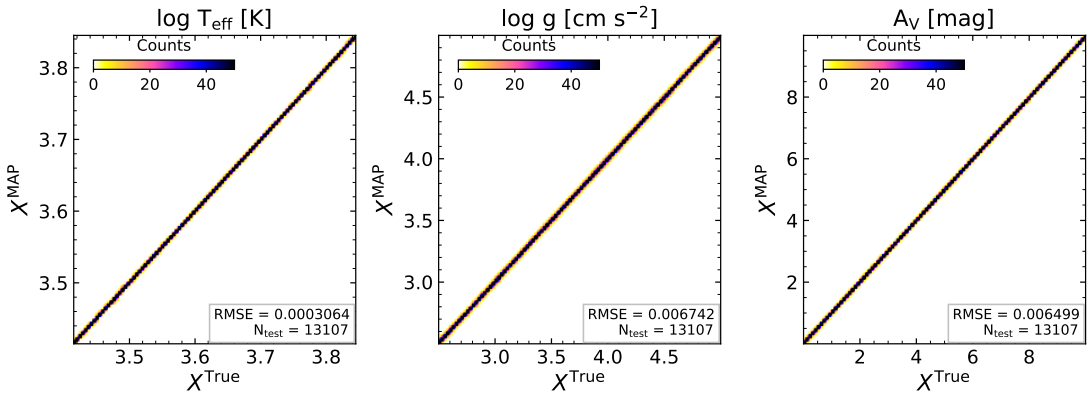

Figure C.1: 2-dimensional histograms comparing the MAP predictions by NextGen-Net and the true values for the entire test models of the NextGen database. The colours indicate the number of models at each point in the 2D histograms. In the lower right corner, we present the root mean square error (RMSE) and the number of test models used (N_{test}).

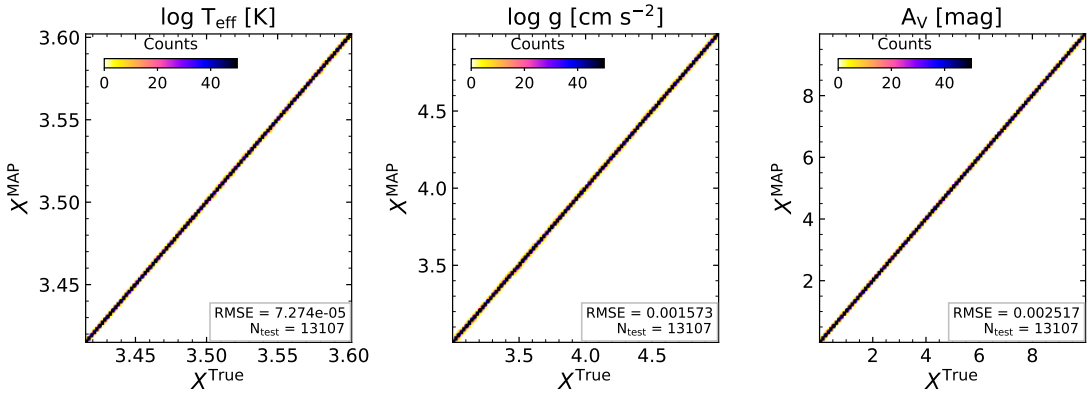

Figure C.2: 2-dimensional histograms comparing the MAP predictions by Dusty-Net and the true values for the entire test models of the Dusty database. The colours indicate the number of models at each point in the 2D histograms. In the lower right corner, we present the root mean square error (RMSE) and the number of test models used (N_{test}).

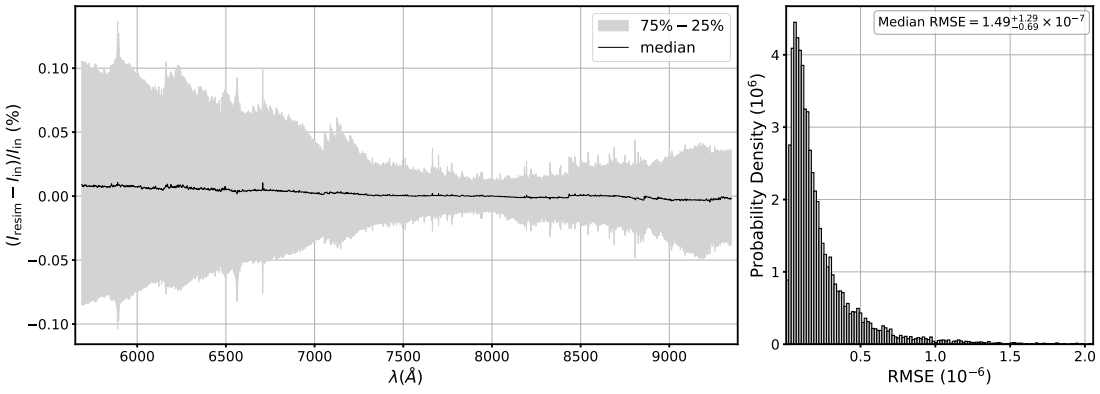

Figure C.3: Left: Median relative error across the wavelength range of the resimulated spectra based on the MAP predictions of the cINN trained on the NextGen models averaged over the 13,107 synthetic spectra in the test set. Here the grey envelope indicates the interquantile range between the 25% and 75% quantiles. Right: Histogram of the RMSEs of the 13,107 resimulated spectra. The mean resimulation RMSE across the test set is $2.28 \pm 2.48 \times 10^{-7}$.

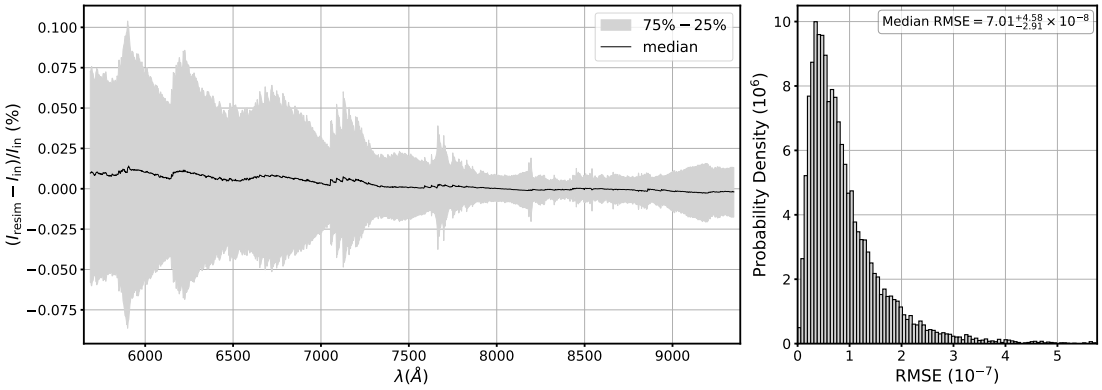

Figure C.4: Left: Median relative error across the wavelength range of the resimulated spectra based on the MAP predictions of the cINN trained on the Dusty models averaged over the 13,107 synthetic spectra in the test set. Here the grey envelope indicates the interquantile range between the 25% and 75% quantiles. Right: Histogram of the RMSEs of the 13,107 resimulated spectra. The mean resimulation RMSE across the test set is $9.01 \pm 7.34 \times 10^{-8}$.

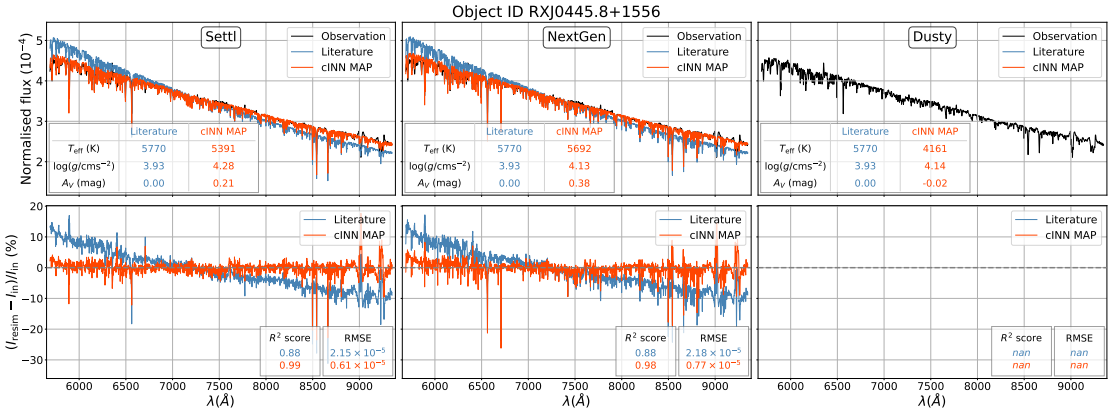

Figure C.5: Resimulation results for Class III star RXJ0445.8+1556. Same as Figure 4.6.

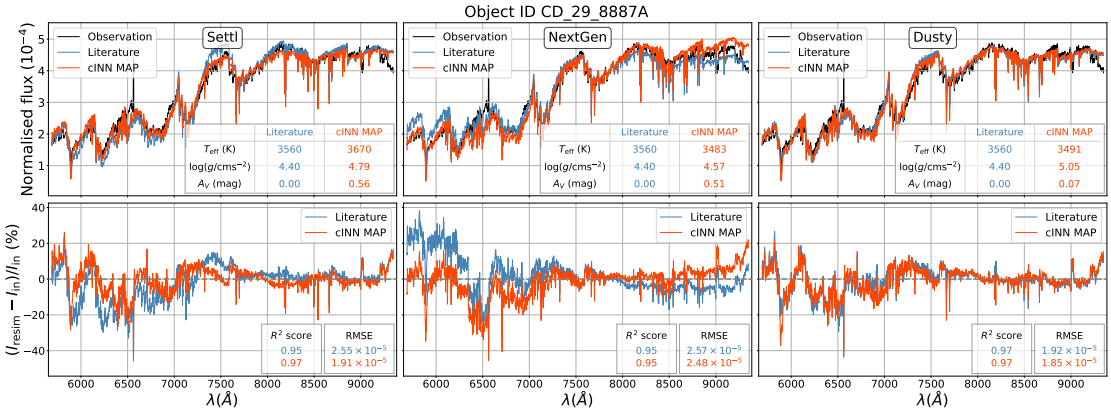

Figure C.6: Resimulation results for Class III star CD_29_8887A. Same as Figure 4.6.

Table C.1: Summary of the resimulation test for the literature values and cINN MAP predictions for the three different spectral libraries, listing the RMSEs and R^2 scores of the resimulated spectra. The comment column indicates the reason why the cINN prediction could not be resimulated. Note that for SO879 the cINN prediction can be resimulated with the Dusty library despite the literature temperature being above 4000 K, because the cINN underestimates $T_{\rm eff}$ by 151 K here, thus falling into the Dusty temperature boundaries.

	Resimulation RMSE ($\times 10^{-5}$) / R^2 Score								
	Settl			NextGen			Dusty		
Object Name	Literature	cINN	Comment	Literature	cINN	Comment	Literature	cINN	Comment
RXJ0445.8+1556	2.15 / 0.88	0.61 / 0.99	-	2.18 / 0.88	0.77 / 0.98	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ1508.6-4423	0.95 / 0.98	0.93 / 0.98	-	1.08 / 0.98	1.05 / 0.98	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ1526.0-4501	1.14 / 0.97	0.66 / 0.99	-	1.20 / 0.96	0.77 / 0.98	-	- / -	- / -	$T_{\rm eff} > 4000$ K
HBC407	1.00 / 0.97	0.68 / 0.99	-	1.16 / 0.96	0.91 / 0.97	-	- / -	- / -	$T_{\rm eff} > 4000$ K
PZ99J160843.4-260216	1.08 / 0.96	0.82 / 0.98	-	1.19 / 0.95	0.93 / 0.97	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ1515.8-3331	1.28 / 0.94	0.85 / 0.97	-	1.41 / 0.92	0.92 / 0.97	-	- / -	- / -	$T_{\rm eff} > 4000$ K
PZ99J160550.5-253313	1.50 / 0.90	0.73 / 0.98	-	1.64 / 0.88	0.93 / 0.96	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ0457.5+2014	1.98 / 0.78	1.23 / 0.92	-	2.07 / 0.76	1.25 / 0.91	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ0438.6+1546	2.09 / 0.70	0.89 / 0.95	-	2.19 / 0.67	1.02 / 0.93	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ1547.7-4018	0.90 / 0.97	0.95 / 0.96	-	1.08 / 0.95	1.11 / 0.95	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ1538.6-3916	1.05 / 0.93	0.92 / 0.94	-	1.24 / 0.90	1.20 / 0.91	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ1540.7-3756	1.42 / 0.56	1.61 / 0.44	-	1.56 / 0.48	1.54 / 0.49	-	- / -	- / -	$T_{\rm eff} > 4000$ K
RXJ1543.1-3920	1.48 / 0.42	1.63 / 0.30	-	1.72 / 0.22	1.48 / 0.42	-	- / -	- / -	$T_{\rm eff} > 4000$ K
SO879	2.66 / 0.53	2.44 / 0.61	-	3.02 / 0.39	2.18 / 0.68	-	- / -	2.08 / 0.71	-
Tyc7760283_1	2.56 / 0.73	1.88 / 0.85	-	1.99 / 0.84	1.95 / 0.84	-	1.93 / 0.85	1.89 / 0.85	$5 < \log(g) < 5.5$
TWA14	3.04 / 0.83	- / -	$\log(g) > 5$	3.28 / 0.80	2.74 / 0.86	-	2.96 / 0.84	3.07 / 0.82	$5 < \log(g) < 5.5$
RXJ1121.3-3447_app2	2.19 / 0.93	1.88 / 0.95	-	2.45 / 0.91	2.15 / 0.93	-	1.86 / 0.95	1.80 / 0.95	$5 < \log(g) < 5.5$
RXJ1121.3-3447_app1	2.69 / 0.92	2.84 / 0.91	-	3.60 / 0.85	2.44 / 0.93	-	2.87 / 0.91	2.42 / 0.93	$5 < \log(g) < 5.5$
CD_29_8887A	2.55 / 0.95	1.91 / 0.97	-	2.57 / 0.95	2.48 / 0.95	-	1.92 / 0.97	1.85 / 0.97	$5 < \log(g) < 5.5$
CD_36_7429B	2.70 / 0.97	2.30 / 0.98	-	4.90 / 0.91	2.57 / 0.97	-	3.26 / 0.96	2.26 / 0.98	-
TWA15_app2	2.98 / 0.96	3.04 / 0.96	-	4.04 / 0.92	2.59 / 0.97	-	2.93 / 0.96	2.57 / 0.97	$5 < \log(g) < 5.5$
TWA7	3.45 / 0.95	3.62 / 0.94	-	4.53 / 0.91	2.66 / 0.97	-	3.36 / 0.95	2.76 / 0.97	-
TWA15_app1	3.95 / 0.93	- / -	$\log(g) > 5$	3.26 / 0.95	2.95 / 0.96	-	3.01 / 0.96	2.96 / 0.96	$5 < \log(g) < 5.5$
SO797	3.77 / 0.97	2.70 / 0.98	-	6.35 / 0.92	2.47 / 0.99	-	4.63 / 0.96	2.27 / 0.99	-
SO641	3.83 / 0.97	3.15 / 0.98	-	6.37 / 0.92	2.62 / 0.99	-	4.76 / 0.96	2.63 / 0.99	-
Par_Lup3_2	3.68 / 0.97	3.03 / 0.98	-	4.74 / 0.95	2.86 / 0.98	-	3.31 / 0.98	2.76 / 0.98	-
SO925	4.55 / 0.97	4.42 / 0.97	-	7.28 / 0.91	3.06 / 0.98	-	5.91 / 0.94	3.17 / 0.98	-
SO999	4.20 / 0.97	3.90 / 0.97	-	6.27 / 0.93	2.99 / 0.98	-	5.00 / 0.96	3.10 / 0.98	-
Sz107	4.44 / 0.97	4.85 / 0.96	-	6.58 / 0.93	2.83 / 0.99	-	5.32 / 0.95	3.11 / 0.98	-
Par_Lup3_1	8.90 / 0.92	5.64 / 0.97	-	12.4 / 0.85	- / -	$\log(g) < 2.5$	11.5 / 0.87	4.05 / 0.98	-
LM717	7.08 / 0.95	5.77 / 0.96	-	10.1 / 0.88	- / -	$\log(g) < 2.5$	9.80 / 0.90	- / -	$\log(g) < 3.0$
J11195652-7504529	7.49 / 0.95	6.73 / 0.96	-	10.9 / 0.89	- / -	$\log(g) < 2.5$	10.1 / 0.89	- / -	$\log(g) < 3.0$
LM601	7.76 / 0.94	7.26 / 0.95	-	9.97 / 0.91	- / -	$\log(g) < 2.5$	9.06 / 0.92	- / -	$\log(g) < 3.0$
CHSM17173	8.65 / 0.94	- / -	$T_{\rm eff} < 2700$ K	10.1 / 0.90	- / -	$\log(g) < 2.5$	9.63 / 0.92	- / -	$\log(g) < 3.0$
TWA26	- / -	- / -	$T_{\rm eff} < 2700$ K	- / -	- / -	$\log(g) < 2.5$	- / -	- / -	$T_{\rm eff} < 2700$ K
DENIS1245	- / -	- / -	$T_{\rm eff} < 2700$ K	- / -	- / -	$\log(g) < 2.5$	- / -	- / -	$T_{\rm eff} < 2700$ K
Resimulated Spectra	34	31	-	34	29	-	20	17	-

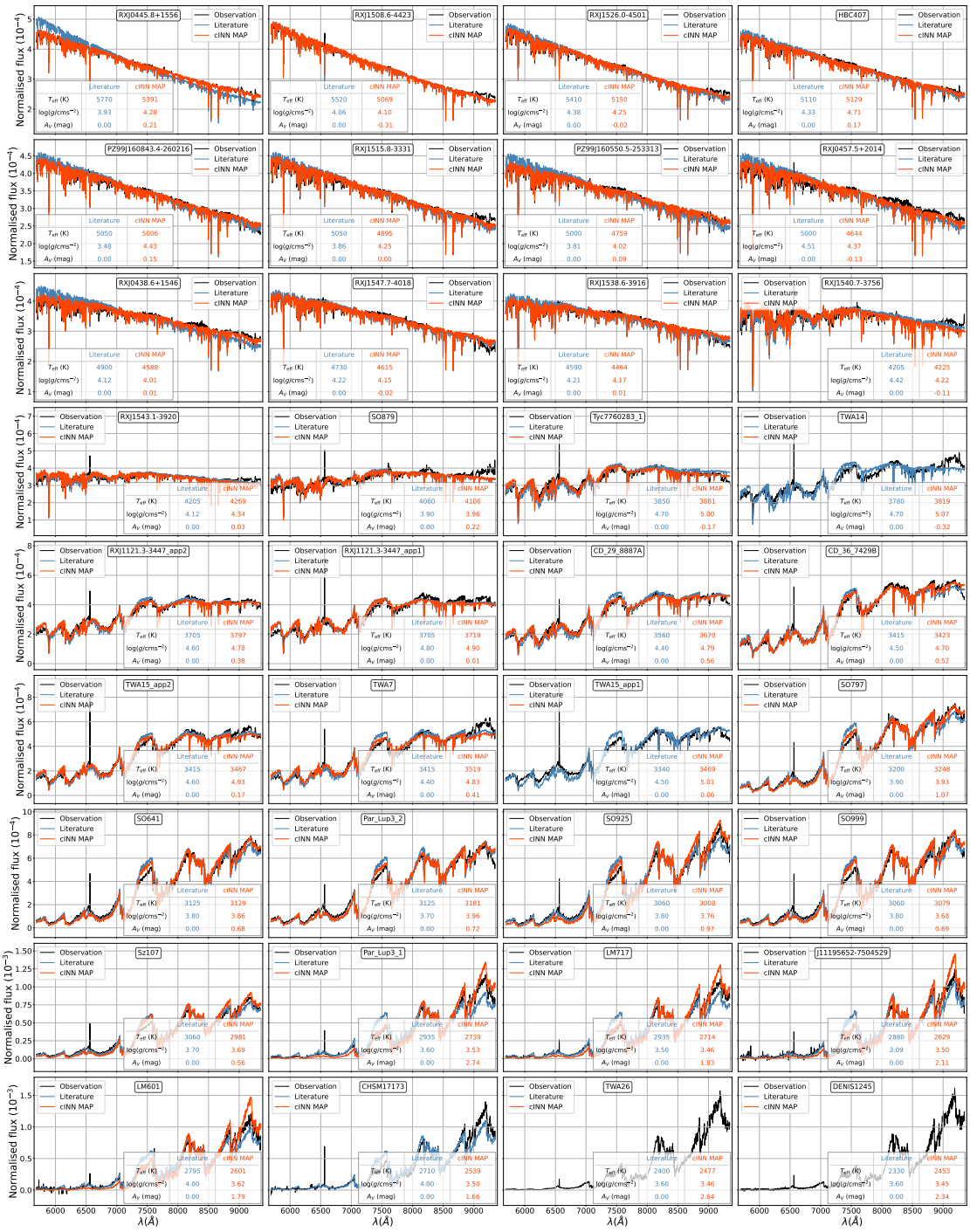

Figure C.7: Resimulation results for all Class III templates for the cINN trained on the Settl library. In each panel, the black curve indicates the observed spectrum, while the red and blue curves correspond to the spectra resimulated based on the cINN MAP estimates and literature properties, respectively. The latter values are summarised in the table in each panel. Note that if either the red or blue or both curves are missing that the corresponding set of parameters could not be resimulated. For the RMSEs and R^2 scores of the resimulated spectra see Table C.1.

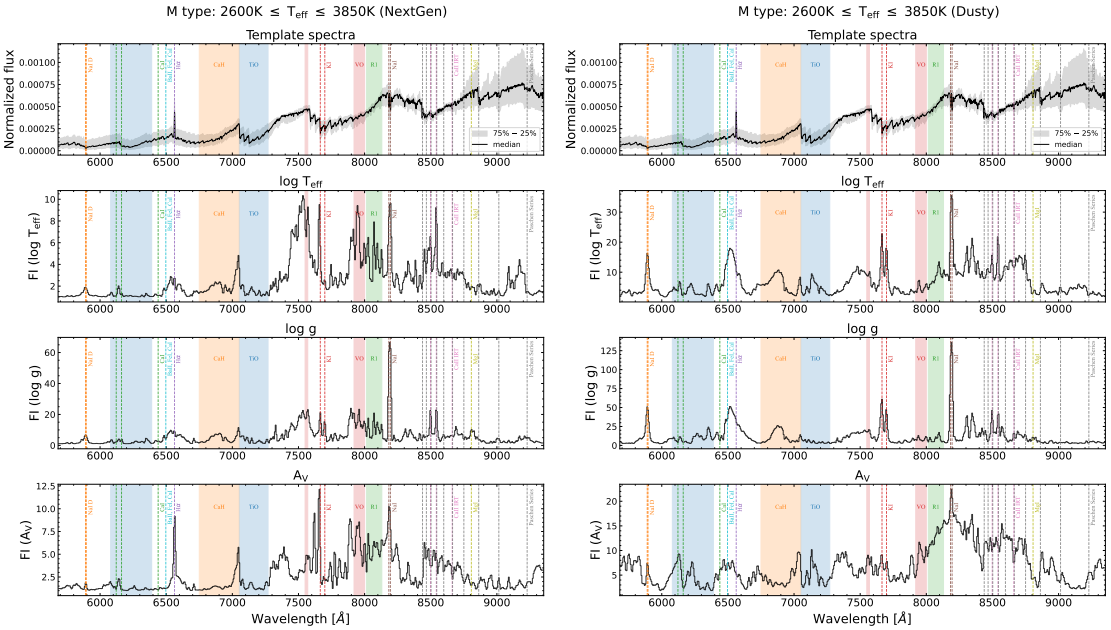

Figure C.8: Feature importance evaluation for M-type synthetic models in the test set using NextGen-Net (left) and Dusty-Net (right), respectively. The first row shows the median flux of M-type Class III template stars. Lines and shades are the same as Figure 4.10.

ogous to Figure 4.6. In particular, these two Figures show examples, where the resimulated spectra based on the cINN MAP estimates seem to match the input spectra notably better than the respective resimulation outcome based on the literature properties of the given Class III templates.

Lastly, Figure C.7 provides an overview of the resimulation results for all Class III template spectra for the cINN trained on the Settl library, corresponding to the top left panels in Figures 4.6, C.5 and C.6.

C.1.3 Feature importance

We investigate the important feature where NextGen-Net and Dusty-Net rely mostly upon. We divided the synthetic observations into three groups depending on their spectral types (e.g., M-, K-, and G-types). We present the results of NextGen-Net and Dusty-Net for M-type stars in Figure C.8. We do not present the results of NextGen-Net for K- and G-type stars because overall results are similar to that of Settl-Net presented in Figure 4.11.

Acknowledgements

My time in Heidelberg over the past four years has been an invaluable and meaningful period in my life both as an astronomer and as one person. Without the help and support of many people around me, it would not have been possible to successfully and pleasantly finish the challenging PhD program. I would like to express my gratitude to these people.

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Dr. Ralf Klessen, who guided me and give persistent support and encouragement throughout these years. I am grateful for all the nice discussions and for his constant patience and trust. I would like to extend my gratitude to my two other supervisors, Prof. Dr. Ullrich Koethe and Prof. Dr. Simon Glover, for their valuable advice and help. I would also like to thank Prof. Dr. Eva Grebel for agreeing to be the examiner of this thesis.

I want to thank my colleagues and collaborators related to my work: Eric Pellegrini, Victor Ksoll, Lynton Ardizzone, Dominika Itrich, and Leonardo Testi. The studies contained in this thesis would not have been possible without their cooperation, fruitful discussions, and help from them. And I would like to thank the past and present team members at ITA. Despite the long pandemic period, I had a great time with my team which will remain a good memory for me.

Finally, I sincerely thank all the people who cheer me up from Korea: my family, my partner and my friends. I thank my parents and brother for their constant support, not only during my PhD study but throughout my entire life. I especially thank my partner, Chikyun, for always encouraging and trusting me in good and bad times.

I am grateful for the financial support from the European Research Council via the ERC Synergy Grant "ECOGAL" (project ID 855130), from the German Research Foundation (DFG) via the Collaborative Research Center "The Milky Way System" (SFB 881 – funding ID 138713538 – subprojects A1, B1, B2 and B8) and from the Heidelberg Cluster of Excellence (EXC 2181 - 390900948) "STRUCTURES", funded by the German Excellence Strategy.

Bibliography

Abraham S., Aniyan A. K., Kembhavi A. K., Philip N. S., Vaghmare K., 2018, MNRAS, 477, 894

Ali A. A., Harries T. J., 2019, MNRAS, 487, 4890

Allard F., Homeier D., Freytag B., 2012, Philosophical Transactions of the Royal Society of London Series A, 370, 2765

Allen L. E., Strom K. M., 1995, AJ, 109, 1379

Appenzeller I., Mundt R., 1989, The Astronomy and Astrophysics Review, 1, 291

Ardizzone L., et al., 2019b, International Conference on Learning Representations

Ardizzone L., Kruse J., Rother C., Köthe U., 2019c, in International Conference on Learning Representations. https://openreview.net/forum?id=rJed6j0cKX

Ardizzone L., Lüth C., Kruse J., Rother C., Köthe U., 2019a, arXiv

Ardizzone L., Lüth C., Kruse J., Rother C., Köthe U., 2019d, CoRR, abs/1907.02392

Ardizzone L., Kruse J., Lüth C., Bracher N., Rother C., Köthe U., 2021, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12544 LNCS, 373

Baldwin J. A., Phillips M. M., Terlevich R., 1981, PASP, 93, 5

Baraffe I., Homeier D., Allard F., Chabrier G., 2015, A&A, 577, A42

Barnes A. T., et al., 2021, MNRAS, 508, 5362

Bastian N., Covey K. R., Meyer M. R., 2010, ARA&A, 48, 339

Bellagente M., Butter A., Kasieczka G., Plehn T., Rousselot A., Winterhalder R., Ardizzone L., Köthe U., 2020, SciPost Physics, 9, 074

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

Bochanski J. J., Hawley S. L., Covey K. R., West A. A., Reid I. N., Golimowski D. A., Ivezić Ž., 2010, AJ, 139, 2679

Botev Z. I., Grotowski J. F., Kroese D. P., 2010, The Annals of Statistics, 38, 2916

Breiman L., 2001, Machine Learning, 45, 5

Cardelli J. A., Clayton G. C., Mathis J. S., 1989, ApJ, 345, 245

Chabrier G., 2003, PASP, 115, 763

Chevance M., et al., 2020a, Space Science Reviews, 216

Chevance M., et al., 2020b, MNRAS, 493, 2872

Crowther P. A., et al., 2016, MNRAS, 458, 624

Dale J. E., 2015, New Astronomy Reviews, 68, 1

Dale J. E., Ercolano B., Bonnell I. A., 2012, MNRAS, 424, 377

Dale J. E., Ercolano B., Bonnell I. A., 2013, MNRAS, 430, 234

Dale J. E., Ngoumou J., Ercolano B., Bonnell I. A., 2014, MNRAS, 442, 694

Dinh L., Sohl-Dickstein J., Bengio S., 2016a, arXiv

Dinh L., Sohl-Dickstein J., Bengio S., 2016b, arXiv e-prints, p. arXiv:1605.08803

Draine B. T., 2011, Physics of the Interstellar and Intergalactic Medium. Princeton University Press, doi:10.1515/9781400839087, https://www.degruyter.com/document/doi/10.1515/9781400839087/html

Dunham M. M., et al., 2014, in , Protostars and Planets VI. University of Arizona Press (arXiv:1401.1809), doi:10.2458/azu_uapress_9780816531240-ch009, http://muse.jhu.edu/books/9780816598762/9780816598762-15.pdf

Edwards S., Kwan J., Fischer W., Hillenbrand L., Finn K., Fedorenko K., Feng W., 2013, ApJ, 778, 148

Eisert L., Pillepich A., Nelson D., Klessen R. S., Huertas-Company M., Rodriguez-Gomez V., 2023, MNRAS, 519, 2199

Ekström S., et al., 2012, A&A, 537, A146

Emsellem E., et al., 2022, A&A, 659, A191

Fabbro S., Venn K. A., O'Briain T., Bialek S., Kielty C. L., Jahandar F., Monty S., 2018, MNRAS, 475, 2978

Ferland G. J., et al., 2017, Revista Mexicana de Astronomia y Astrofisica, 53, 385

Fisher A., Rudin C., Dominici F., 2018, arXiv e-prints, p. arXiv:1801.01489

Frasca A., Alcalá J. M., Covino E., Catalano S., Marilli E., Paladino R., 2003, A&A, 405, 149

Freedman D., Diaconis P., 1981, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete, 57, 453

Geen S., Pellegrini E., Bieri R., Klessen R., 2020, MNRAS, 492, 915

Georgy C., Ekström S., Meynet G., Massey P., Levesque E. M., Hirschi R., Eggenberger P., Maeder A., 2012, A&A, 542, A29

Georgy C., et al., 2013, A&A, 558, A103

Girichidis P., et al., 2020, Space Science Reviews, 216

Glover S. C. O., Mac Low M.-M., 2011, MNRAS, 412, 337

Goldsmith P. F., Velusamy T., Li D., Langer W. D., 2010, ApJ, 715, 1370

Goodfellow I., Bengio Y., Courville A., 2016, Deep Learning. MIT Press, Cambridge, MA, http://www.deeplearningbook.org

Goodwin S., 2013, in Oswalt T. D., Barstow M. A., eds, , Planets, Stars and Stellar Systems. Springer Netherlands, Dordrecht, pp 243–277, doi:10.1007/978-94-007-5615-1_5, https://doi.org/10.1007/978-94-007-5615-1_5http://link.springer.com/10.1007/978-94-007-5615-1_5

Grudić M. Y., Guszejnov D., Offner S. S. R., Rosen A. L., Raju A. N., Faucher-Giguère C.-A., Hopkins P. F., 2022, MNRAS, 512, 216

Guo C., Pleiss G., Sun Y., Weinberger K. Q., 2017, 34th International Conference on Machine Learning, ICML 2017, 3, 2130

Haldemann J., et al., 2022, arXiv

Henry T. J., Kirkpatrick J. D., Simons D. A., 1994, AJ, 108, 1437

Herczeg G. J., Hillenbrand L. A., 2014, ApJ, 786, 97

Hobbs A., Read J., Nicola A., 2015, MNRAS, 452, 3593

Hur H., Sung H., Bessell M. S., 2012, The Astronomical Journal, 143, 41

Husser T. O., Wende-von Berg S., Dreizler S., Homeier D., Reiners A., Barman T., Hauschildt P. H., 2013, A&A, 553, A6

Hyvärinen A., Oja E., 2000, Neural Networks, 13, 411

James G., Witten D., Hastie T., Tibshirani R., 2017, An introduction to statistical learning, corrected at 8th printing edn. Springer texts in statistics, Springer, New York ; Heidelberg ; Dordrecht ; London, doi:10.1007/978-1-4614-7138-7

Jeans J., 1902, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 199, 1

Jeffries R. D., Oliveira J. M., Naylor T., Mayne N. J., Littlefair S. P., 2007, MNRAS, 376, 580

Kang D. E., Pellegrini E. W., Ardizzone L., Klessen R. S., Koethe U., Glover S. C. O., Ksoll V. F., 2022, MNRAS, 512, 617

Kang D. E., Klessen R. S., Ksoll V. F., Ardizzone L., Koethe U., Glover S. C. O., 2023, MNRAS, 520, 4981

Kauffmann G., et al., 2003, MNRAS, 346, 1055

Kenyon S. J., Hartmann L., 1995, ApJS, 101, 117

Kewley L. J., Dopita M. A., Sutherland R. S., Heisler C. A., Trevena J., 2001, ApJ, 556, 121

Kewley L. J., Groves B., Kauffmann G., Heckman T., 2006, MNRAS, 372, 961

Kewley L. J., Dopita M. A., Leitherer C., Davé R., Yuan T., Allen M., Groves B., Sutherland R., 2013, ApJ, 774, 100

Kim J.-G., Kim W.-T., Ostriker E. C., 2016, ApJ, 819, 137

Kim J.-G., Kim W.-T., Ostriker E. C., 2018, ApJ, 859, 68

Kim H., Lee H., Kang W. H., Lee J. Y., Kim N. S., 2020, arXiv

Kim J., et al., 2021, MNRAS, 504, 487

Kingma D. P., Ba J., 2014, arXiv e-prints, p. arXiv:1412.6980

Kingma D. P., Dhariwal P., 2018a, arXiv e-prints, p. arXiv:1807.03039

Kingma D. P., Dhariwal P., 2018b, Advances in Neural Information Processing Systems, 2018-Decem, 10215

Kippenhahn R., Weigert A., Weiss A., 2012, Stellar Structure and Evolution. Astronomy and Astrophysics Library, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-642-30304-3, http://link.springer.com/10.1007/978-3-642-30304-3

Kirkpatrick J. D., Henry T. J., McCarthy Donald W. J., 1991, ApJS, 77, 417

Kirkpatrick J. D., Kelly D. M., Rieke G. H., Liebert J., Allard F., Wehrse R., 1993, ApJ, 402, 643

Klessen R. S., Glover S. C. O., 2016, in Revaz Y., Jablonka P., Teyssier R., Mayer L., eds, , Star Formation in Galaxy Evolution: Connecting Numerical Models to Reality: Saas-Fee Advanced Course 43. Swiss Society for Astrophysics and Astronomy. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 85–249, doi:10.1007/978-3-662-47890-5_2, https://doi.org/10.1007/978-3-662-47890-5_2http://link.springer.com/10.1007/978-3-662-47890-5_2

Kollmeier J. A., et al., 2017, arXiv e-prints, p. arXiv:1711.03234

Kroupa P., 2001, MNRAS, 322, 231

Kroupa P., 2002, Science, 295, 82

Krumholz M. R., Tan J. C., 2007, ApJ, 654, 304

Krumholz M. R., et al., 2014, in , Protostars and Planets VI. University of Arizona Press, Tucson, AZ, p. 243 (arXiv:1401.2473), doi:10.2458/azu_uapress_9780816531240-ch011, http://arxiv.org/abs/1401.2473http://dx.doi.org/10.2458/azu_uapress_9780816531240-ch011http://muse.jhu.edu/books/9780816598762/9780816598762-17.pdf

Ksoll V. F., et al., 2020, MNRAS, 499, 5447

Lada C. J., Lombardi M., Alves J. F., 2010, ApJ, 724, 687

Lee J. C., et al., 2022, ApjS, 258, 10

Leitherer C., et al., 1999, ApjS, 123, 3

Leitherer C., Ekström S., Meynet G., Schaerer D., Agienko K. B., Levesque E. M., 2014, ApjS, 212, 14

Leroy A. K., Walter F., Brinks E., Bigiel F., de Blok W. J. G., Madore B., Thornley M. D., 2008, AJ, 136, 2782

Leroy A. K., et al., 2012, AJ, 144, 3

Luhman K. L., Briceño C., Stauffer J. R., Hartmann L., Barrado y Navascués D., Caldwell N., 2003, ApJ, 590, 348

Manara C. F., et al., 2013, A&A, 551, A107

Manara C. F., Frasca A., Alcalá J. M., Natta A., Stelzer B., Testi L., 2017, A&A, 605, A86

Mathews W. G., 1967, ApJ, 147, 965

McLeod A. F., et al., 2021, MNRAS, 508, 5425

Molnar C., 2022, Interpretable Machine Learning, 2 edn. https://christophm.github.io/interpretable-ml-book

Murray N., 2011, ApJ, 729, 133

Olney R., et al., 2020a, AJ, 159, 182

Olney R., et al., 2020b, AJ, 159, 182

Oort J. H., Spitzer L. J., 1955, ApJ, 121, 6

Paszke A., et al., 2019, in Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., Garnett R., eds, , Advances in Neural Information Processing Systems 32. Curran Associates, Inc., Red Hook, NY, pp 8024–8035, http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pellegrini E. W., Baldwin J. A., Ferland G. J., 2011, ApJ, 738, 34

Pellegrini E. W., Rahner D., Reissl S., Glover S. C. O., Klessen R. S., Rousseau-Nepton L., Herrera-Camus R., 2020, MNRAS, 496, 339

Pudritz R. E., Ouyed R., Fendt C., Brandenburg A., 2006, arXiv

Rahner D., Pellegrini E. W., Glover S. C. O., Klessen R. S., 2017, MNRAS, 470, 4453

Rahner D., Pellegrini E. W., Glover S. C. O., Klessen R. S., 2018, MNRAS, 473, L11

Rahner D., Pellegrini E. W., Glover S. C. O., Klessen R. S., 2019, MNRAS, 483, 2547

Reissl S., Wolf S., Brauer R., 2016, A&A, 593, A87

Reissl S., Brauer R., Klessen R. S., Pellegrini E. W., 2019, ApJ, 885, 15

Riddick F. C., Roche P. F., Lucas P. W., 2007, MNRAS, 381, 1067

Rousseau-Nepton L., Robert C., Martin R. P., Drissen L., Martin T., 2018, MNRAS, 477, 4152

Rousseau-Nepton L., et al., 2019, MNRAS, 489, 5530

Rugel M. R., et al., 2019, A&A, 622, A48

Sánchez S. F., et al., 2015, A&A, 574, A47

Santoro F., et al., 2022, A&A, 658, A188

Sharma K., Kembhavi A., Kembhavi A., Sivarani T., Abraham S., Vaghmare K., 2020a, MNRAS, 491, 2280

Sharma K., Kembhavi A., Kembhavi A., Sivarani T., Abraham S., Vaghmare K., 2020b, MNRAS, 491, 2280

Shen H., Huerta E. A., O'Shea E., Kumar P., Zhao Z., 2022, Machine Learning: Science and Technology, 3, 015007

Shetty R., Ostriker E. C., 2008, ApJ, 684, 978

Silverman B. W., 1986, Density Estimation for Statistics and Data Analysis. Chapman & Hall, London

Stelzer B., et al., 2013, A&A, 558, A141

Strömgren B., 1939, ApJ, 89, 526

Testi L., 2009, A&A, 503, 639

Trofimova D., Adler T., Kausch L., Ardizzone L., Maier-Hein K., Köthe U., Rother C., Maier-Hein L., 2020, NeurIPs Medical Imaging

Walmsley M., et al., 2021, MNRAS, 509, 3966

Watkins E. J., Peretto N., Marsh K., Fuller G. A., 2019, A&A, 628, A21

Wei W., et al., 2020, MNRAS, 493, 3178

Whitmore B. C., et al., 2021, MNRAS, 506, 5294

Wolfire M. G., Hollenbach D., McKee C. F., 2010, ApJ, 716, 1191

Wu C., et al., 2019, MNRAS, 482, 1211

de Beurs Z. L., et al., 2022, AJ, 164, 49