Investigation of the solar atmosphere using machine learning techniques

Seminar presentation on master's thesis

Daniel Zahnd

May 27, 2024

Content

Introduction

Space weather relies on identification of precursors to solar eruptions.

Inversions (conversion of observations into physical parameters) are needed to identify those precursors.

Most inversion models need to be inverted numerically; conventional methods are computationally costly.

This thesis explores the application of a machine learning technique called "normalizing flows" to invert solar atmospheric models.

An atmospheric model $\boldsymbol{M}$ predicts observations $\boldsymbol{y}$ based on physical parameters $\boldsymbol{x}$.

Such a model is governed by physics; especially radiative transfer.

The basis of an atmospheric model is the generalized radiative transfer equation describing the evolution of the four Stokes parameters.

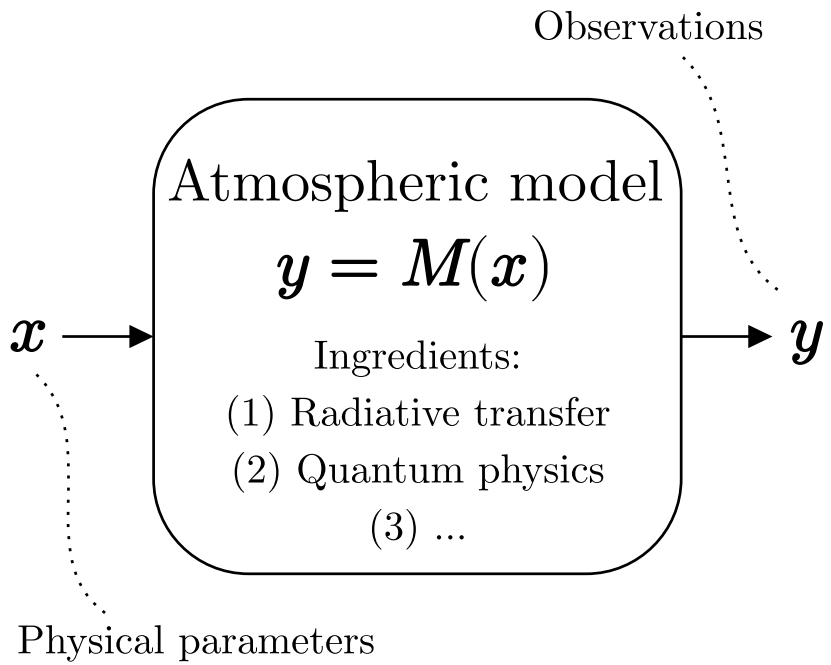


Figure 1: Sketch of a (solar) atmospheric model.

Generally speaking, the generalized radiative transfer equation cannot be solved analytically.

The Milne-Eddington assumptions as [Degl'Innocenti, 2005, p.411] gives them however allows for an analytical solution:

1. The radiative transfer system is assumed to be of a plane parallel nature and in a local thermodynamic equilibrium.
2. All parameters relevant to spectral line formation are required to be independent of all measures of height within the system.
3. The source function is required to be an affine function of the continuum optical depth.

The full Milne-Eddington model can be written as

$$\boldsymbol{M} : \mathbb{R}^9 \to \mathbb{R}^D, \quad \boldsymbol{x} = \begin{pmatrix} |\boldsymbol{B}| \\ \theta \\ \varphi \\ \vdots \end{pmatrix} \mapsto \boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{I}_{\lambda_{min}} \\ \vdots \\ \boldsymbol{I}_{\lambda} \\ \vdots \\ \boldsymbol{I}_{\lambda_{max}} \end{pmatrix}. \tag{1}$$

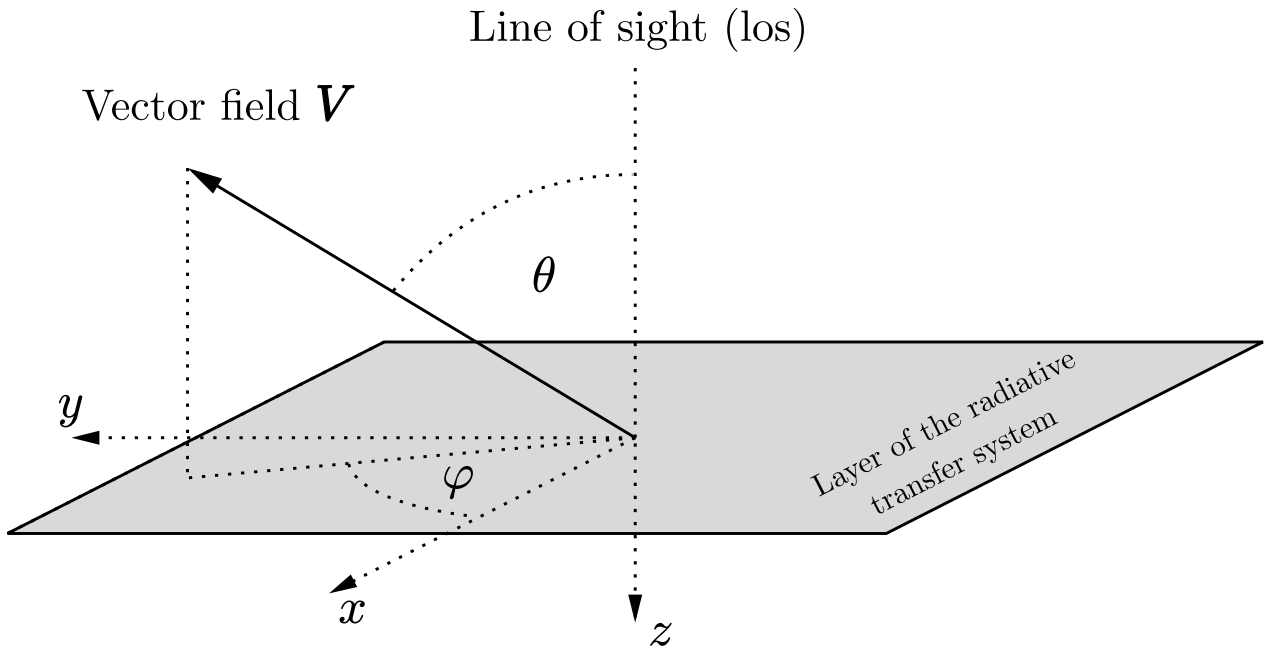


Figure 2: Visualization of the definitions for the inclination and azimuth angles θ and φ.

Normalizing flows

Why use normalizing flows for inversions?

There are two main reasons to use normalizing flows for inversions:

1. Only simple atmospheric models can be inverted analytically. Therefore, one needs to resort to numerical methods, which are computationally demanding. Normalizing flows are - once trained - comparatively faster.
2. Numerical methods usually return a single solution to an inversion problem. Normalizing flow however return an entire probability density.

Normalizing flows

What has been done yet using normalizing flows?

Normalizing flows were introduced in 2015 by [Rezende and Mohamed, 2015] as a new solution to approximate posterior probability densities $p(\boldsymbol{x}|\boldsymbol{y})$.

[Díaz Baso et al., 2022] applied normalizing flows to learn stellar atmospheric inversions using LTE and non-LTE models.

The thesis at hand comes in here to explore normalizing flows applied to learn the full Milne-Eddington inversion task (with magn. field parameters).

Normalizing flows

How does a normalizing flow work?

A flow is a generative model, which transforms a target probability density $p_{\boldsymbol{x}}(\boldsymbol{x})$ into a latent (base) probability density $p_{\boldsymbol{z}}(\boldsymbol{z})$.

Members $\boldsymbol{x}$ of the target probability density $p_{\boldsymbol{x}}(\boldsymbol{x})$ are optionally conditioned on data $\boldsymbol{y}$. (Sometimes $\boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x})$).

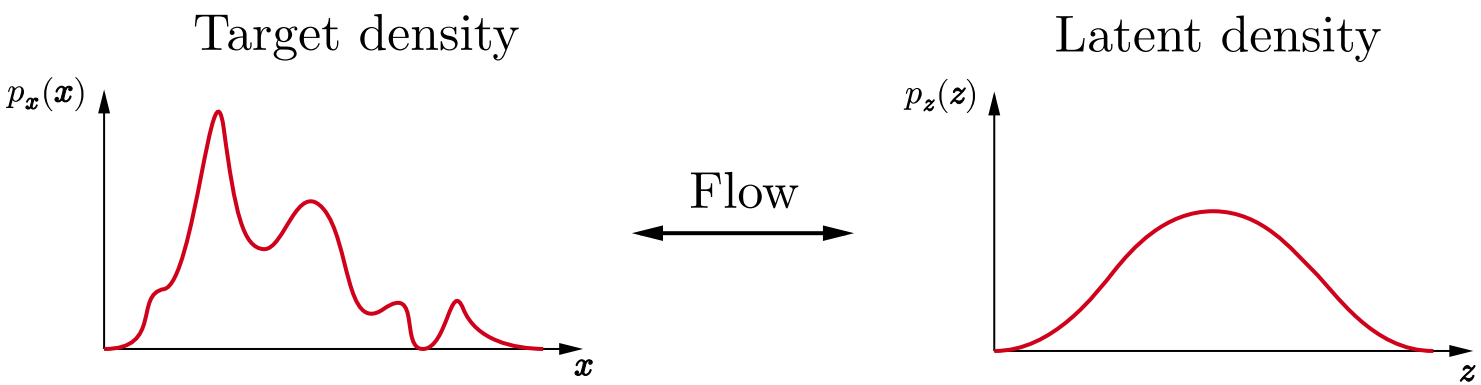


Figure 3: Simple illustration of the working principle of a normalizing flow.

Affine coupling layer flows are simple and straightforward implementations of normalizing flows.

Basically, one uses a concatenation of affine functions to transform the target density to the latent density.

Simple example with $f_2(x) = a_2 x + b_2$ and $f_1(x) = a_1 x + b_1$:

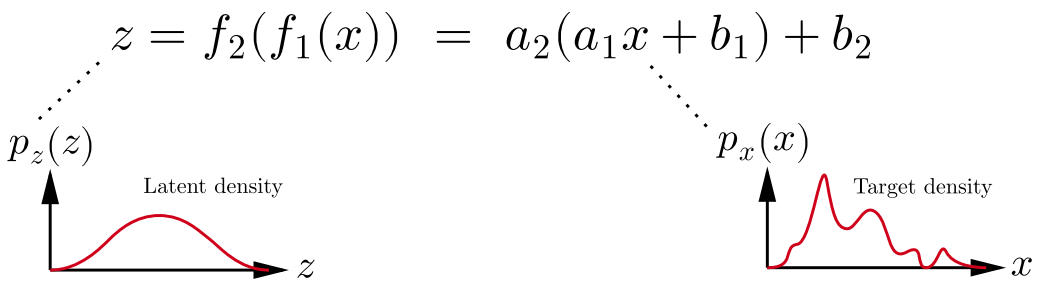


Figure 4: Simple example of an affine coupling layer flow.

While simple to implement, affine coupling layer flows lack flexibility, especially with multimodal density functions.

Therefore, [Durkan et al., 2019, p.4] suggest piecewise quadratic spline flows: Concatenation of quadratic functions.

Simple example with $f_2(x) = a_2x^2 + b_2$ and $f_1(x) = a_1x^2 + b_1$:

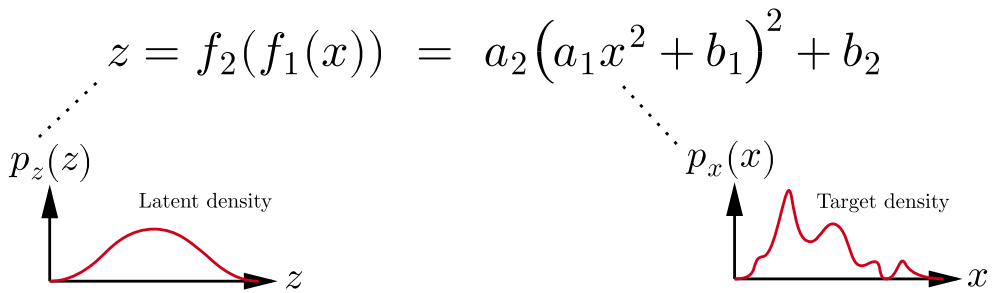


Figure 5: Simple example of a quadratic coupling layer flow.

Normalizing flows

Quadratic vs. affine coupling layer normalizing flows

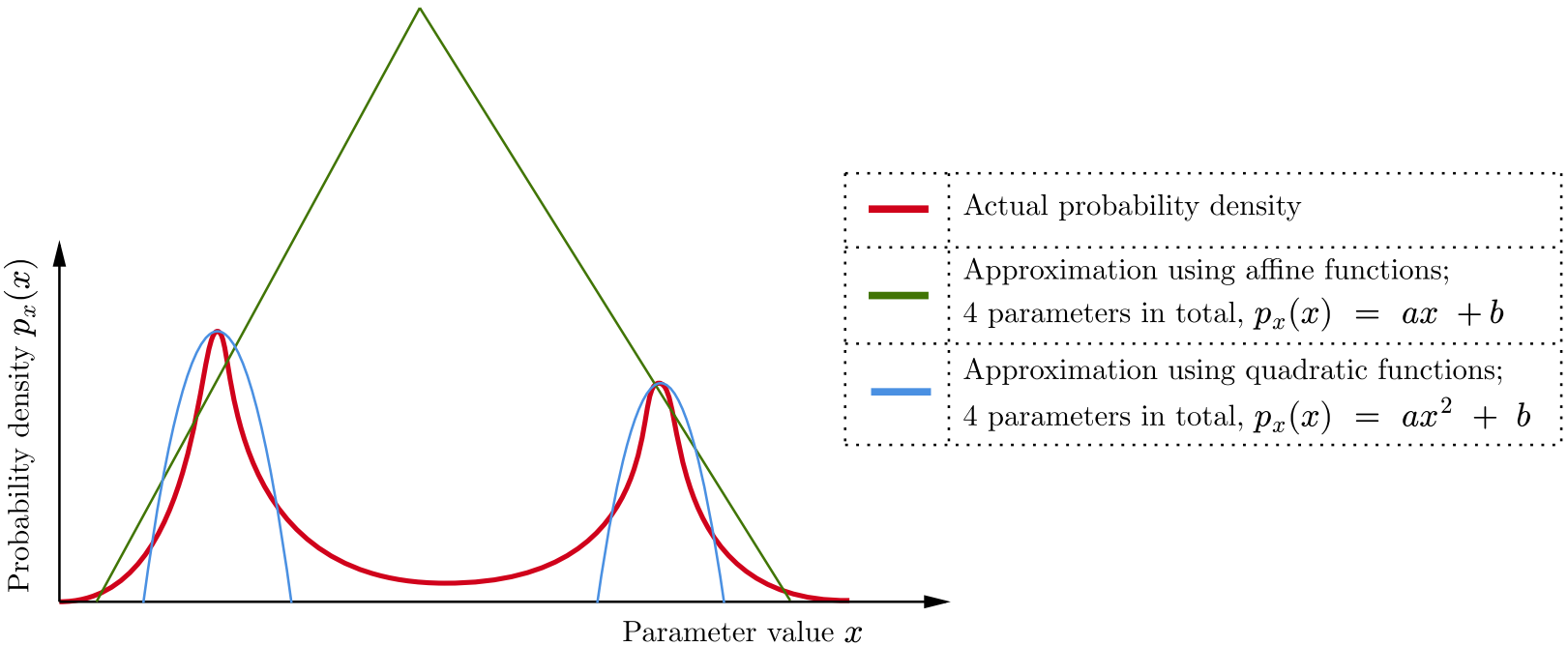


Figure 6: Approximating a probability density with affine or quadratic functions using the same amount of parameters in both cases.

Results and discussion

General information about experiments

The stellar atmospheric model for all of four presented experiments was the Milne-Eddington model. It extracts 9 atmospheric parameters from four Stokes profiles.

The experiments were performed on penumbra formation maps obtained by the space weather group at the Swedish Solar Telescope (SST).

This data contains a FOV of 550 by 600 pixels with four Stokes profiles consisting of 13 wavelength points each for each pixel.

The observations focus on the Fe I absorption line centered at $\lambda_0 = 6302.4931\,\text{Å}$.

The Milne-Eddington model for the experiments can be specified as

$$\boldsymbol{M} : \mathbb{R}^9 \to \mathbb{R}^{52}, \quad \boldsymbol{x} = \begin{pmatrix} |\boldsymbol{B}| \\ \theta \\ \varphi \\ \vdots \end{pmatrix} \mapsto \boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{I}_{\lambda_{min}} \\ \vdots \\ \boldsymbol{I}_{\lambda_0} \\ \vdots \\ \boldsymbol{I}_{\lambda_{max}} \end{pmatrix}, \qquad (2)$$

where $\boldsymbol{I}_\lambda$ are the normalized Stokes parameters at wavelength λ.

Results and discussion

Multiple map analysis: Experiment explanation

The penumbra formation dataset contains 16 frames. Frame 0 containing observational data $\hat{\mathbf{y}}$ was inverted using the Milne-Eddington algorithm, yielding corresponding atmospheric parameter data $\hat{\mathbf{x}}$.

A normalizing flow was trained on the data $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ to map a 9-dimensional standard normal distribution to the distribution of atmospheric parameters.

Given an observation $\boldsymbol{y}$ (Stokes profiles of one pixel) and many samples $\boldsymbol{z} \sim \mathcal{N}(0,1)^9$ of a 9-dimensional standard distribution, one obtains densities for the nine atmospheric parameters.

Figure fig. 7 shows, how the solution to an inversion problem is obtained given the output of a normalizing flow for one pixel.

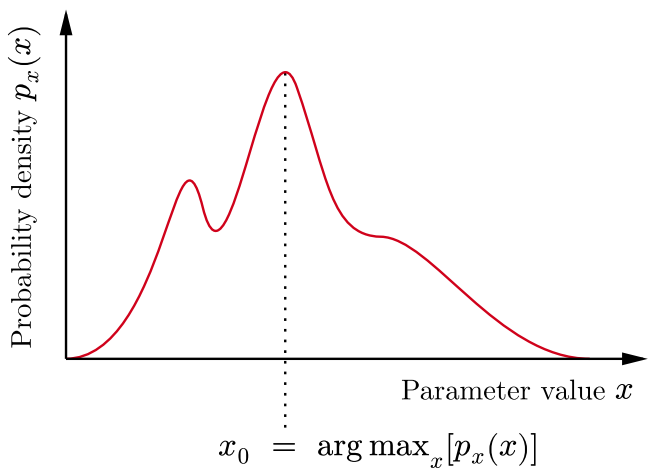


Figure 7: Determining the solution of an inversion problem using normalizing flows. Suppose, $x \sim p_x(x)$ is the output of the normalizing flow. In order to get the most probable solution x_0 as given by the normalizing flow, one can take x_0 to be the argument of maximal probability of $p_x(x)$. 500 samples per atmospheric parameter were used in experiments.

Results and discussion

Multiple map analysis: Results

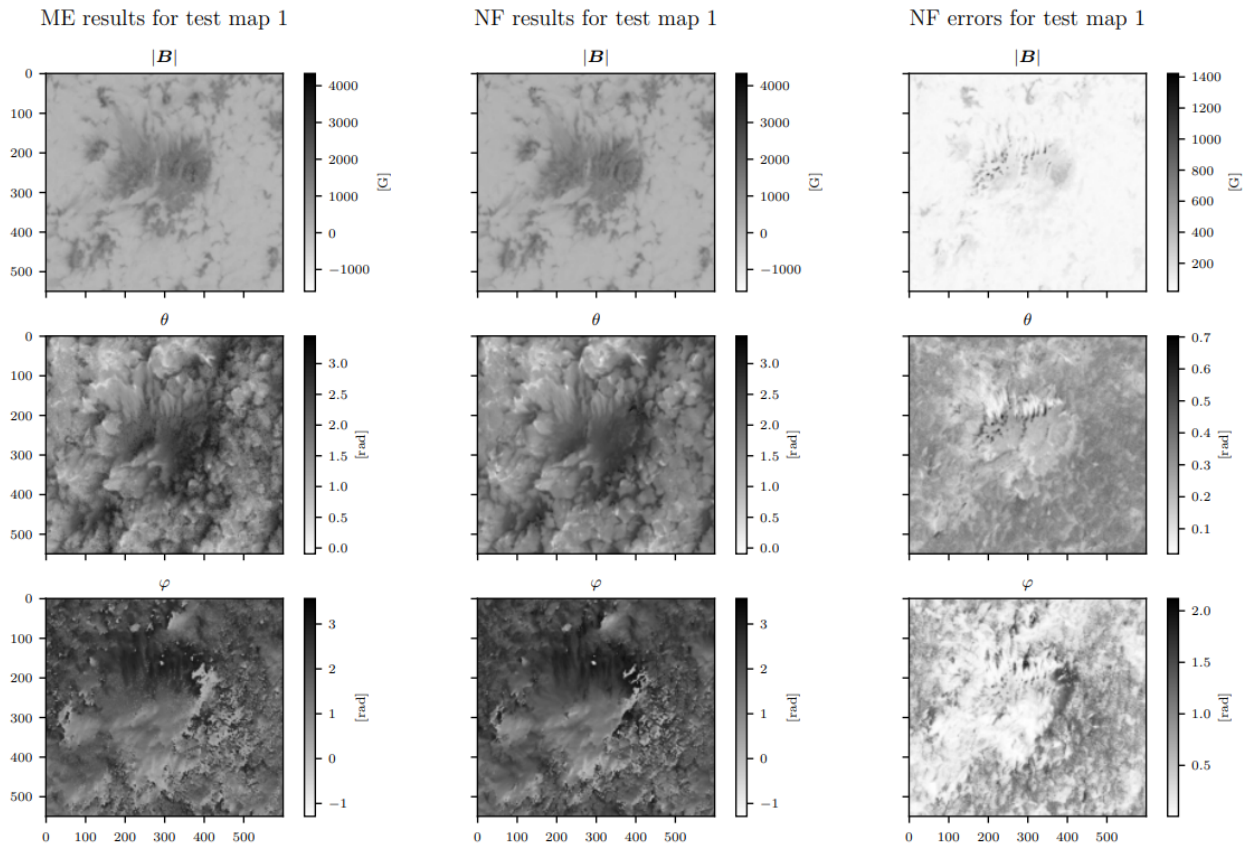


Figure 8: Results for multiple map analysis experiment.

There are two key points to take away from these results:

1. Normalizing flows used for atmospheric inversions seem to have a tendency to smooth out edges, corners and sharp transitions: The “smoothing property”.
2. Uncertainties calculated as standard deviations of the obtained probability densities for atmospheric parameters show expected behaviour for inclination and azimuth.

Results and discussion

Improving normalizing flow performance by enhancing the training dataset: Experiment explanation

This experiment consisted of successively increasing the size of the training dataset to see, whether the normalizing flow trained on it leads to "better" inversion results.

For this purpose, frame 0, frames 0 and 1 and frames 0, 1 and 2 of the penumbra formation dataset were used as training data (Stokes profiles $\hat{\mathbf{y}}$ with associated parameters $\hat{\mathbf{x}}$).

The three models were used to invert the last frame of the penumbra formation dataset.

Direct comparison to the Milne-Eddington inversion results for the same frame leads to an assessment of the results.

Results and discussion

Improving normalizing flow performance by enhancing the training dataset: Results

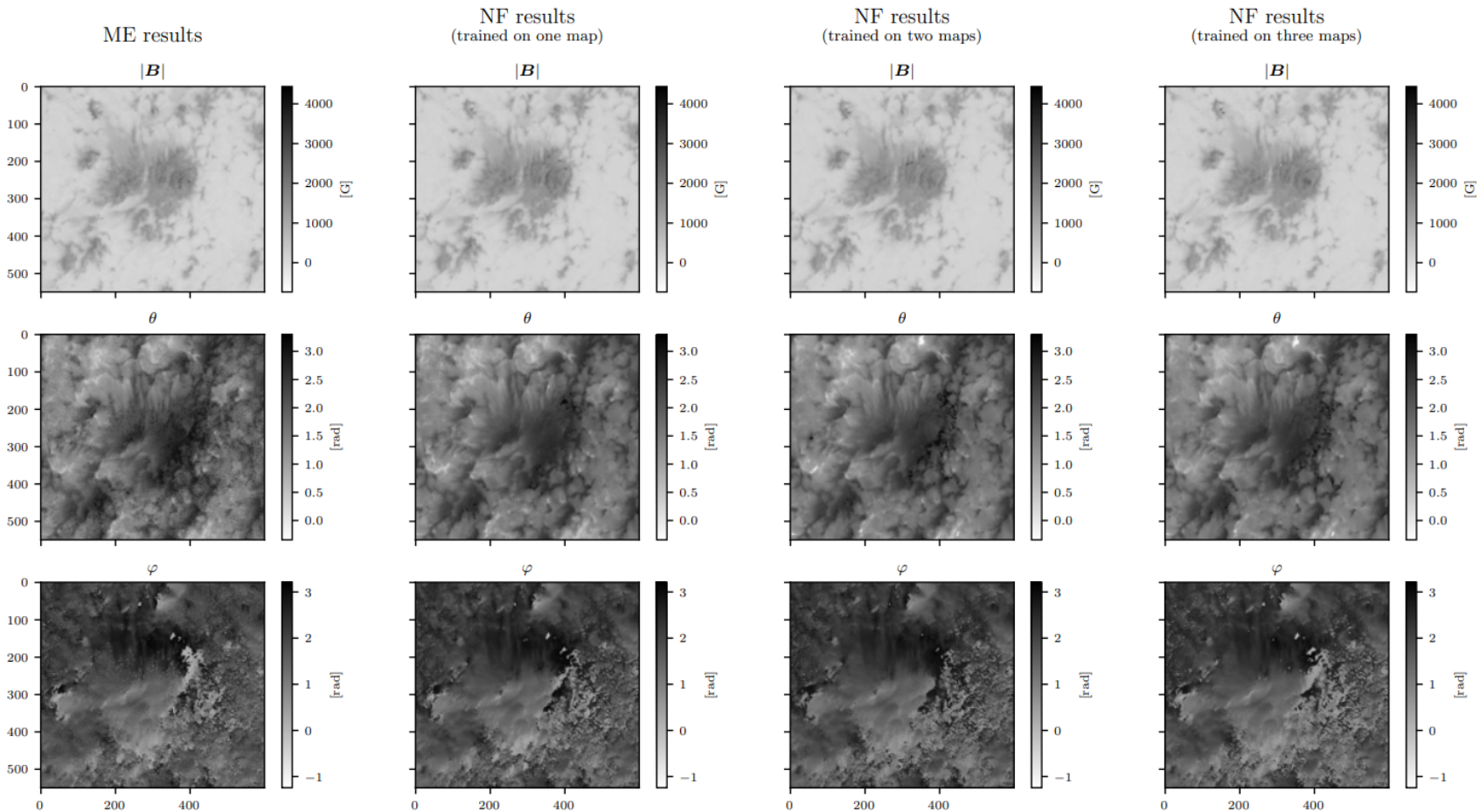


Figure 9: Resulting inversion for improving normalizing flow performance by enhancement of the training dataset experiment.

Results and discussion

Improving normalizing flow performance by enhancing the training dataset: Results

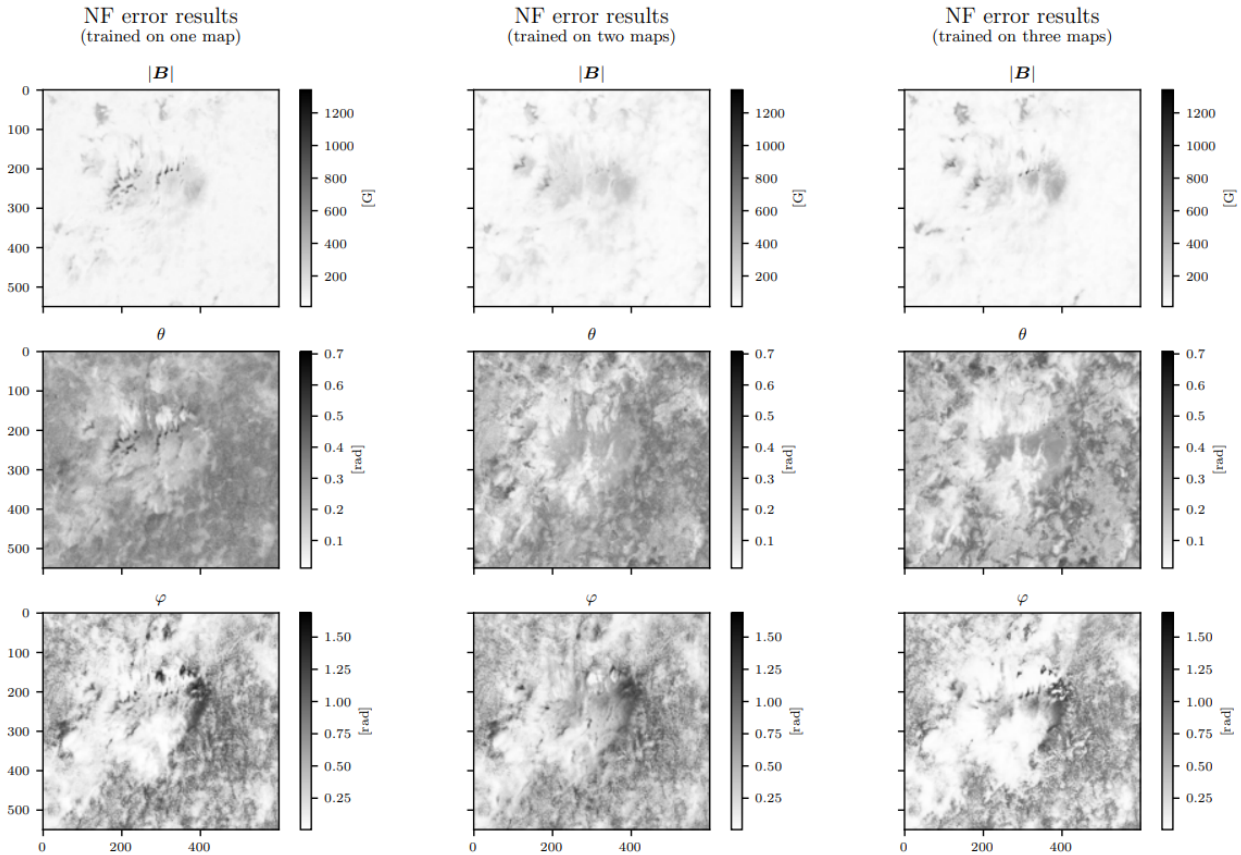


Figure 10: Resulting inversion errors for improving normalizing flow performance by enhancement of the training dataset experiment.

Results and discussion

Improving normalizing flow performance by enhancing the training dataset: Results

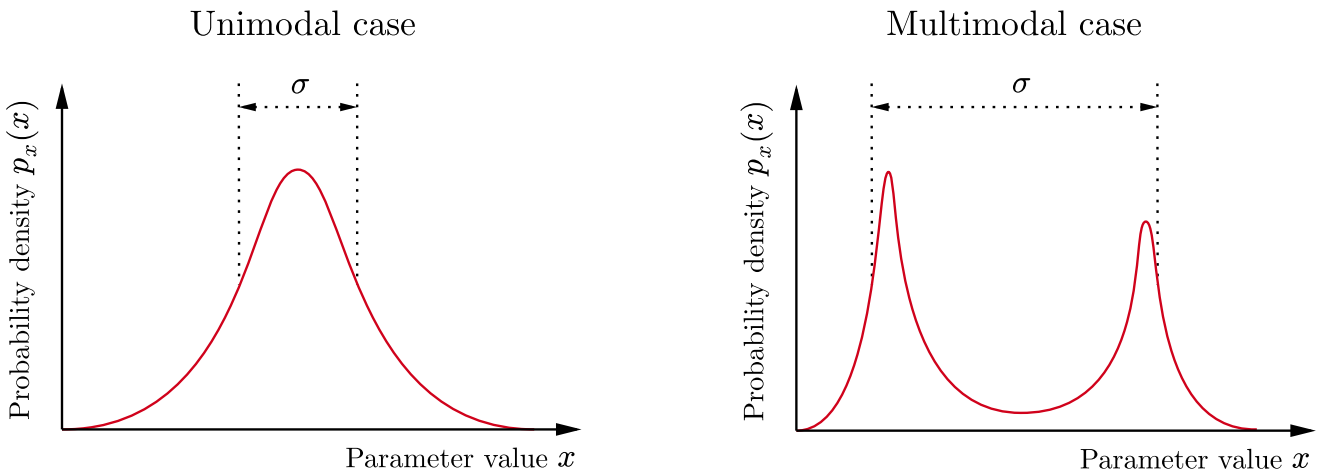


Figure 11: Visualization of the principle, that the standard variation σ of a multimodal probability density is larger than that of an unimodal one, even if the multiple modes are sharply peaked - as long as the multiple peaks are sufficiently separated.

There are two key points to take away from these results:

1. Overall uncertainties in normalizing flow inversions seem to decrease for increasing size in dataset. There are however areas, where errors increase.
2. Generally speaking one can say, that increasing the amount of "diversity" in the training dataset, e.g. by taking into account more data for training, increases the overall performance of normalizing flows.

Conclusions

There are several conclusions to be drawn from the presented material:

1. Normalizing flows seem to be a useful tool with many applications.
2. Normalizing flows seem to provide reasonable results for inversions of the full Milne-Eddington model (LTE).
3. Increasing the size and hence variation in the training dataset seems to improve normalizing flow performance (captures more possible solutions).

Outlook

There are many possibilities for further research, e.g.:

1. Build normalizing flow model solely trained on synthetic data.
2. Investigate performance of normalizing flows trained for more complex atmospheric models, e.g. SIR (LTE) or STiC (non-LTE).
3. Build normalizing flow model capable of inverting observations of any desired wavelength window.
4. Quantify, if normalizing flow results are accurate enough to identify possible precursors of solar eruptions.

Questions

Thank you for your time and attention!

Literature

Degl'Innocenti, E. L. (2005).

Polarization in Spectral Lines, volume 307 of *Springer eBook Collection Physics and Astronomy*. Springer Netherlands, Dordrecht.

Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020).

Mathematics for machine learning. Cambridge University Press, Cambridge and New York, NY and Port Melbourne and New Delhi and Singapore.

del Toro Iniesta, J. C. (2003).

Introduction to Spectropolarimetry. Cambridge University Press, Cambridge.

Díaz Baso, C. J., Asensio Ramos, A., and de La Cruz Rodríguez, J. (2022).

Bayesian stokes inversion with normalizing flows. *Astronomy & Astrophysics*, 659:A165.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016).

Density estimation using real nvp.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019).

Neural spline flows.

Froehlich, C. and Lean, J. (2004).

Solar radiative output and its variability: evidence and mechanisms. *The Astronomy and Astrophysics Review*, 12(4):273–320.

Literature

Kobyzev, I., Prince, S. J. D., and Brubaker, M. A. (2021).

Normalizing flows: An introduction and review of current methods.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(11):3964–3979.

Raschka, S. (2022).

Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python.
Packt Publishing Limited, Birmingham, 1 edition.

Rezende, D. J. and Mohamed, S. (2015).

Variational inference with normalizing flows.

Rutten, R. J. (2015).

Introduction to astrophysical radiative transfer: Lecture notes.

Weigert, A., Wendker, H. J., and Wisotzki, L. (2006).

Astronomie und Astrophysik: Ein Grundkurs.
Lehrbuch Physik. Wiley-VCH Verlag, Weinheim, 1. nachdr. der 4., völlig überarb. und erw. aufl. edition.

Weng, L. (2018).

Flow-based deep generative models.
lilianweng.github.io.

Appendix

Why bother about inversions?

The ultimate goal of space weather is to identify triggers of solar flares and coronal mass ejections.

Physical parameters in the Sun's atmosphere need to be known for that matter.

Measurement directly within the solar atmosphere are not possible due to extreme environment; inference of atmospheric parameters from remote-sensing observations (spectral lines) is necessary.

Appendix

The Sun's characteristics

Object of investigation of the thesis is the Sun's atmosphere.

The outer layers of the Sun are roughly divided into photosphere, chromosphere and corona.

Depth of photosphere defined by optical depth, such that photons from below the photosphere cannot escape to space [Weigert et al., 2006, p.135].

Interesting phenomena such as sunspots appear in the photosphere.

Appendix

The Sun's characteristics

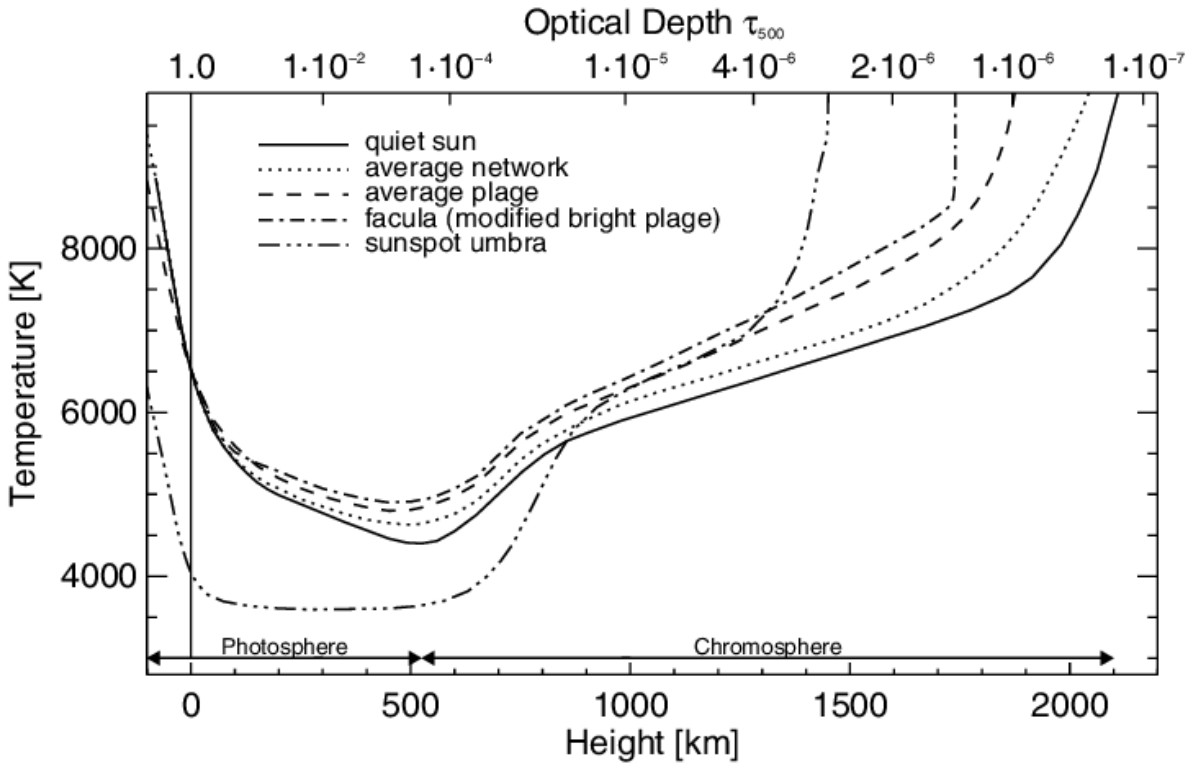


Figure 12: Temperature profile in the solar atmosphere. Figure taken from [Froehlich and Lean, 2004, p.281].

Appendix

The Sun's characteristics

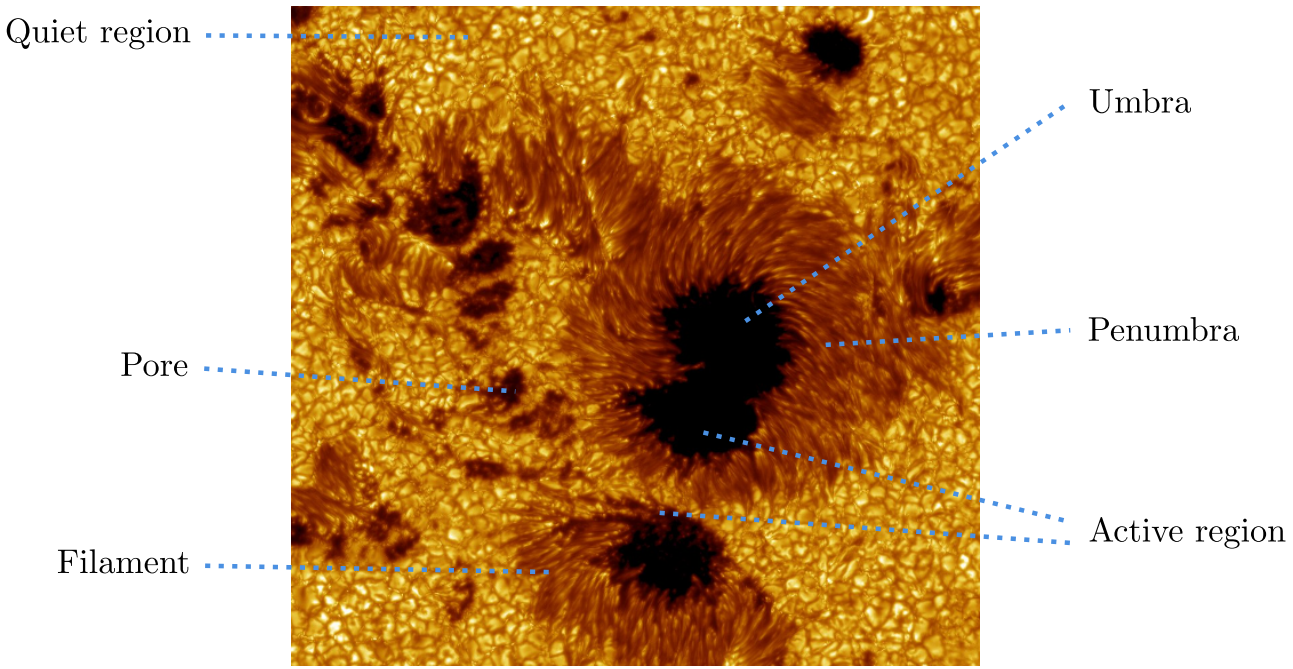


Figure 13: Sunspots in the active region 10030, observed on July 15, 2002. Image credit: NASA Goddard Space Flight Center, `https://www.flickr.com/photos/gsfc/5510488494`, accessed on November 07, 2023. Adapted by the author.

Fully developed sunspots exhibit an umbra surrounded by a filament-like structure, the penumbra.

A schematic visualization of a sunspot is given in the left panel of fig. 2; whereas the common bipolar sunspot arrangement is pictured in the right panel.

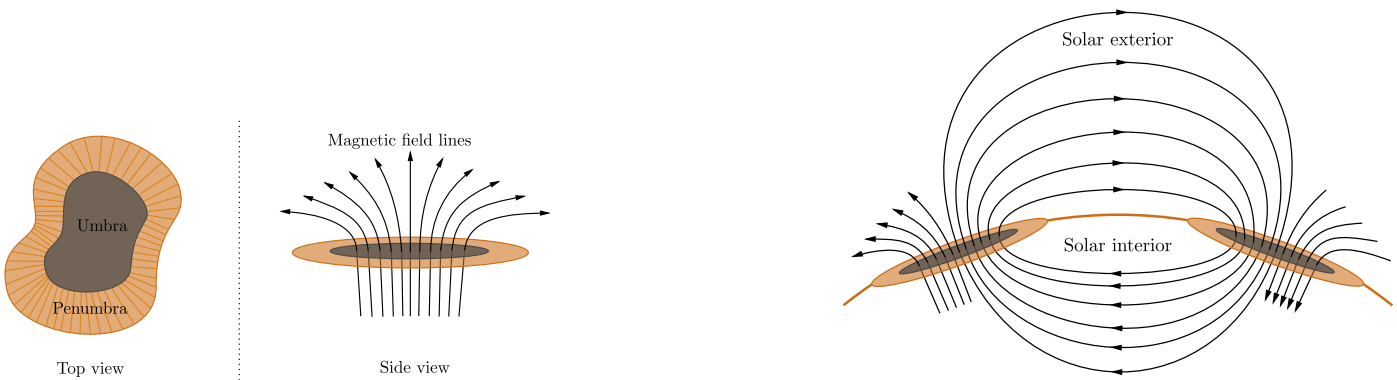


Figure 14: Schematic visualizations of sunspots. Note, that the magnetic field lines as drawn here could also be oriented exactly opposite.

Appendix

Radiative transfer equation

Using fig. 15, the radiative transfer equation can be derived to take the form

$$\cos(\theta)\frac{\mathrm{d}I_\lambda(\tau'_\lambda)}{\mathrm{d}\tau'_\lambda} = I_\lambda(\tau'_\lambda) - S_\lambda(\tau'_\lambda). \quad (3)$$

The optical depth τ'_λ is defined as $\mathrm{d}\tau'_\lambda \doteq \alpha_\lambda \,\mathrm{d}z$, whereas τ_λ is the optical pathlength; $S_\lambda = j_\lambda/\alpha_\lambda$.

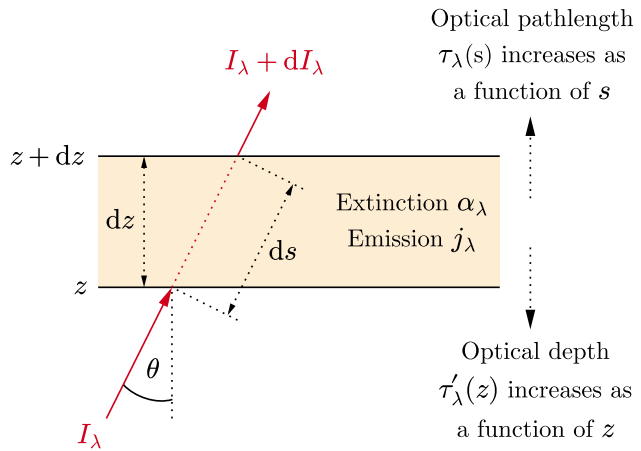


Figure 15: Illustration for the derivation of the radiative transfer equation. $[I_\lambda] = \mathrm{W\,sr^{-1}\,m^{-3}}$ denotes the spectral radiance measured along a ray of inclination θ with respect to a horizontal plane in the coordinate system under consideration.

The formal solution of eq. (3) is given by integration from τ'_λ to ∞ as

$$I^+_\lambda(\tau'_\lambda, \theta) = \frac{1}{\cos(\theta)} \int_{\tau'_\lambda}^{\infty} e^{\frac{\tau'_\lambda - t}{\cos(\theta)}} S_\lambda(t)\,\mathrm{d}t, \quad (4)$$

known as outward spectral radiance ("intensity").

$S_\lambda(\tau'_\lambda)$ is a source function describing photon emission at optical depth τ'_λ.

In thermodynamic equilibrium, $S_\lambda = B_\lambda$ with B_λ the Planck function holds.

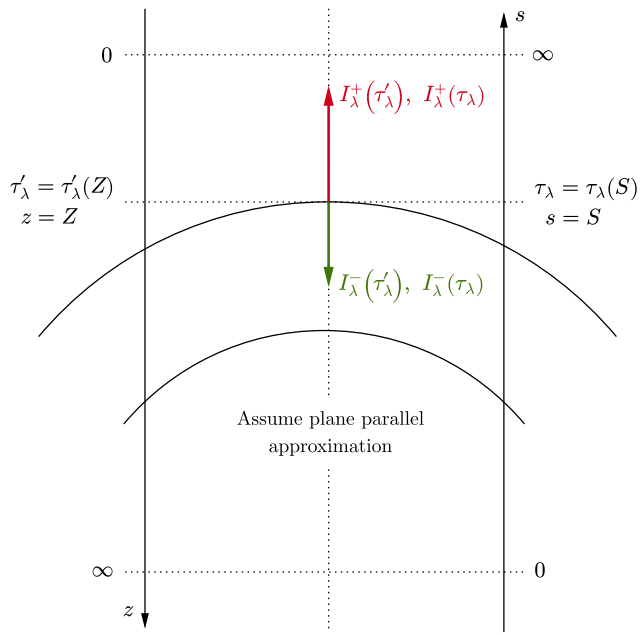


Figure 16: Illustration of coordinate systems for optical pathlength τ_λ parametrized by s and optical depth τ'_λ parametrized by z.

Appendix

Radiative transfer equation

In order to build an atmospheric model for a stellar atmosphere, radiative transfer is needed.

The propagation of electromagnetic radiation through matter is described by the radiative transfer equation.

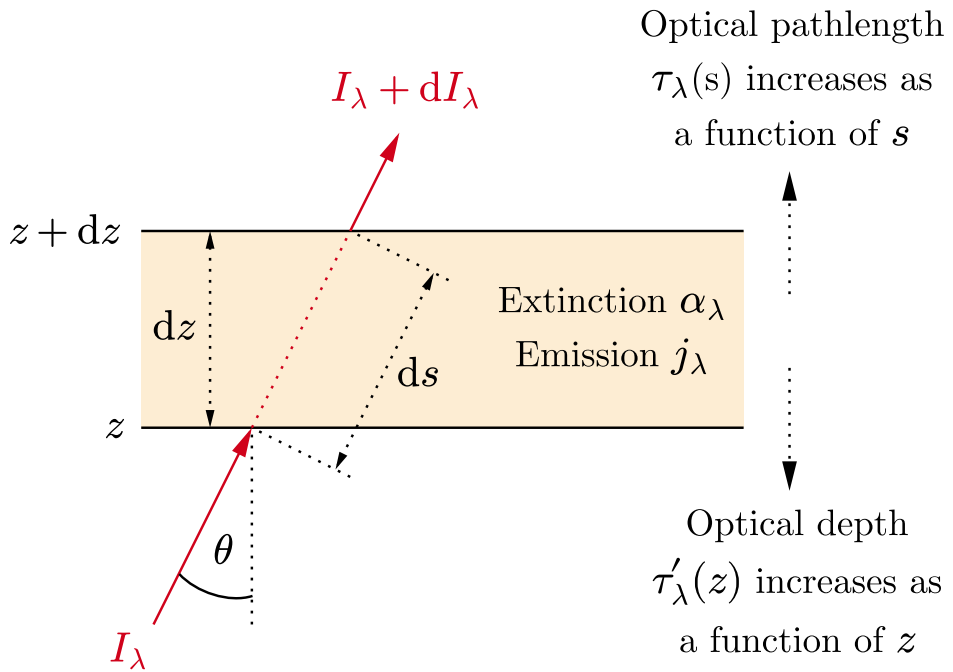


Figure 17: Illustration for the derivation of the radiative transfer equation. $[I_\lambda] = \mathrm{W\,sr^{-1}\,m^{-3}}$ denotes the spectral radiance measured along a ray of inclination θ with respect to a horizontal plane in the coordinate system under consideration.

Appendix

Zeeman effect

A radiative transfer system subject to a strong external magnetic field $\boldsymbol{B}$ will exhibit splitting of an absorption/emission line into several lines.

This effect is known as the Zeeman effect and allows for inference of magnetic field strength.

The "distance" between split energy levels is a function of magnetic field strength.

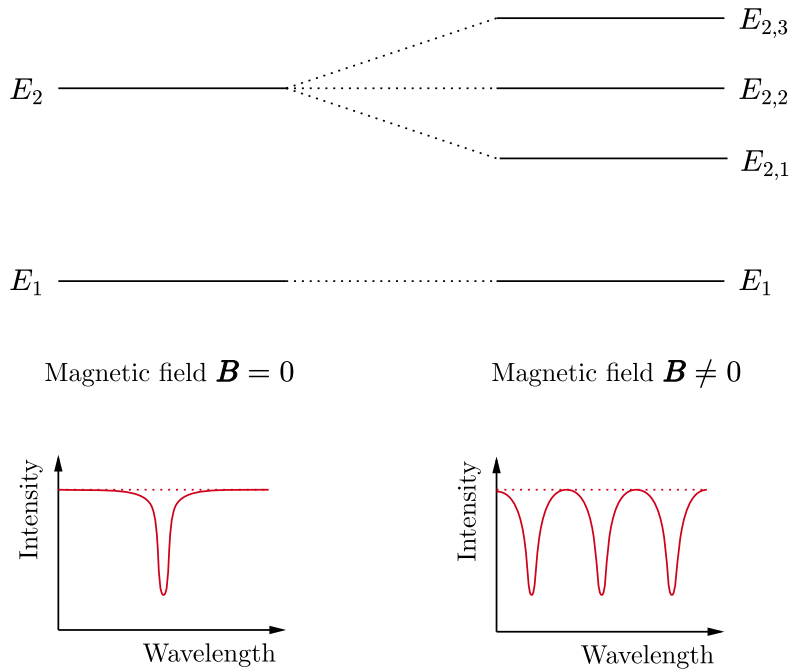


Figure 18: Visualization of the Zeeman effect. Due to coupling of the magnetic moment of electrons with the external magnetic field, energy levels split into several sub-levels.

Appendix

Stokes parameters and Stokes vector

The Stokes parameters I_λ, Q_λ, U_λ and V_λ contain information about the polarization properties of electromagnetic waves.

The Stokes vector $\boldsymbol{I}_\lambda = (I_\lambda, Q_\lambda, U_\lambda, V_\lambda)^\top$ contains all four Stokes parameters.

Values of the Stokes parameters in dependence of wavelength λ are called Stokes profiles. They can be used to infer physical parameters of a radiative transfer system from observations.

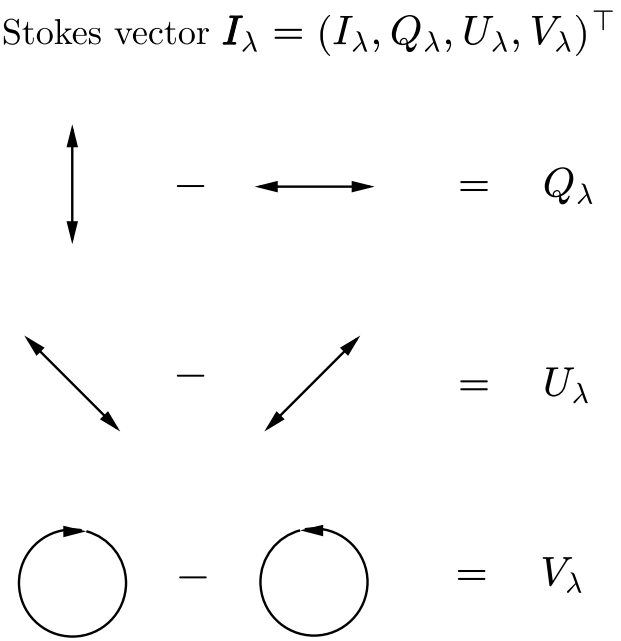


Figure 19: Meaning of the four Stokes parameters for an electromagn. wave propagating towards the observer. Figure inspired by [Degl'Innocenti, 2005, p.17].

In order to describe the change in all four Stokes parameters with optical depth, one needs to generalize the radiative transfer equation.

The generalized radiative transfer equation encompasses all information about how different atmospheric parameters change with respect to optical depth; [del Toro Iniesta, 2003, p.150] give it as

$$\frac{\mathrm{d}\boldsymbol{I}_\lambda(\tau'_\lambda)}{\mathrm{d}\tau'_\lambda} = \mathbf{K}_\lambda(\tau'_\lambda)\left[\boldsymbol{I}_\lambda(\tau'_\lambda) - \boldsymbol{S}_\lambda(\tau'_\lambda)\right], \tag{5}$$

were $\mathbf{K}_\lambda$ is the propagation matrix and
$\boldsymbol{S}_\lambda(\tau'_\lambda) = S_\lambda(\tau'_\lambda)(1,0,0,0)^\top$ is the generalized source function.

Appendix

Spectral line formation

The Eddington-Barbier approximation provides a means to qualitatively understand spectral line formation.

It is obtained as

$$I_\lambda^+(\tau'_\lambda = 0, \theta = 0) \approx S_\lambda(\tau'_\lambda = 1) \tag{6}$$

by $S_\lambda(\tau'_\lambda)$ as a power series in τ'_λ and truncating terms of order $\mathcal{O}(\cos(\theta)^2)$ and higher.

Essentially it says, that the intensity I_λ^+ perpendicular to the system's surface as measured by an observer outside the system is roughly given by the source function at optical depth $\tau'_\lambda = 1$.

Appendix

Spectral line formation

Eddington-Barbier relation:
$I_\lambda^+(\tau'_\lambda = 0) \approx S_\lambda(\tau'_\lambda = 1)$.

Assumptions:

1. Extinction coefficient α_λ independent of depth z; hence $\tau'_\lambda(Z) = \int_0^Z \alpha_\lambda \, \mathrm{d}z = \alpha_\lambda Z$ holds.
2. Source function S_λ independent of wavelength λ and affine in z, i.e. $S_\lambda = S = S_0 + S_1 z$.

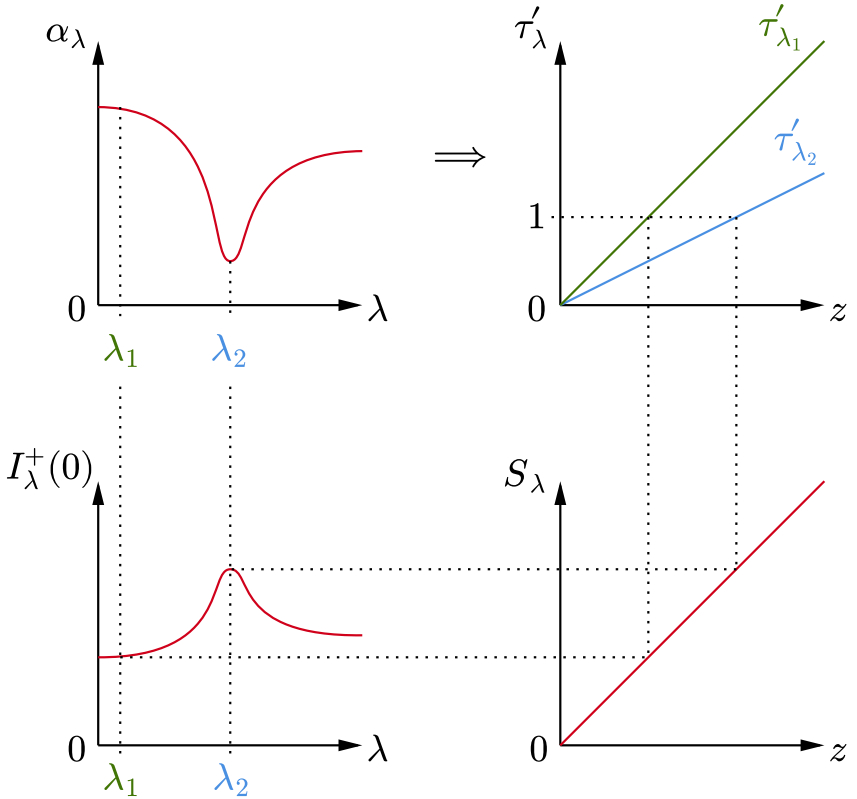


Figure 20: Illustration of a spectral line formation using the Eddington-Barbier relation. Figure inspired by [Rutten, 2015, p.38].

Appendix

Generalized radiative transfer equation

The radiative transfer equation can be generalized to encompass information about the full Stokes vector $\boldsymbol{I}_\lambda$.

The generalized radiative transfer equation is given as

$$\frac{\mathrm{d}\boldsymbol{I}_\lambda(\tau'_\lambda)}{\mathrm{d}\tau'_\lambda} = \mathbf{K}_\lambda(\tau'_\lambda)\left[\boldsymbol{I}_\lambda(\tau'_\lambda) - \boldsymbol{S}_\lambda(\tau'_\lambda)\right], \tag{7}$$

were $\mathbf{K}_\lambda$ is the propagation matrix and $\boldsymbol{S}_\lambda(\tau'_\lambda) = S_\lambda(\tau'_\lambda)(1,0,0,0)^\top$ is the generalized source function [del Toro Iniesta, 2003, p.150].

The propagation matrix $\mathbf{K}_\lambda$ encompasses all information about how the different atmospheric parameters affect the development of $\boldsymbol{I}_\lambda$ as a function of optical depth and wavelength.

The general formal solution to eq. (7) is derived to be

$$\boldsymbol{I}_\lambda(\tau'_\lambda) = \mathbf{P}\left(\tau'_\lambda, \infty\right) \boldsymbol{I}_\lambda(\infty) - \int_\infty^{\tau'_\lambda} \mathbf{P}\left(\tau'_\lambda, t\right) \mathbf{K}_\lambda(t) \boldsymbol{S}_\lambda(t)\, \mathrm{d}t, \quad (8)$$

where $\mathbf{P}(\kappa, \kappa')$ is the so-called evolution operator giving the evolution of the homogeneous part from a point κ' to κ.

At $\tau'_\lambda = 0$, i.e. at the position of an observer outside the radiative transfer system, the solution reduces to

$$\boldsymbol{I}_\lambda(\tau'_\lambda = 0) = \int_0^\infty \mathbf{P}(0, t) \mathbf{K}_\lambda(t) \boldsymbol{S}_\lambda(t)\, \mathrm{d}t. \quad (9)$$

[Degl'Innocenti, 2005, p.411] gives three assumptions pertaining to the Milne-Eddington approximations:

1. The radiative transfer system is assumed to be of a plane parallel nature and in a local thermodynamic equilibrium.
2. All parameters relevant to spectral line formation are required to be independent of all measures of height within the system.
3. The source function is required to be an affine function of the continuum optical depth.

Implementing these assumptions as simplifications to eq. (9), one arrives as

$$\boldsymbol{I}_\lambda(\tau'_\lambda = 0) = \boldsymbol{S}_{\lambda,0} - \mathbf{K}^{-1}_{\lambda,0}\boldsymbol{S}_{\lambda,1} \tag{10}$$

for the Milne-Eddington solution.

By assumption 1., $\mathbf{K}_{\lambda,0}$ is the isotropically constant propagation matrix at wavelength λ.

By assumption 2., the source function vector $\boldsymbol{S}_\lambda$ is given by $\boldsymbol{S}_\lambda(\tau'_\lambda) = \boldsymbol{S}_{\lambda,0} + \tau'_\lambda \boldsymbol{S}_{\lambda,1}$.

By assumption 1., $\boldsymbol{S}_{\lambda,0}$ and $\boldsymbol{S}_{\lambda,1}$ are connected to the Planck function $B_\lambda(T)$ as $\boldsymbol{S}_{\lambda,0} = B_\lambda(T)(1,0,0,0)^\top = (S_0,0,0,0)^\top$ and $\boldsymbol{S}_{\lambda,1} = B_\lambda(T)\beta(1,0,0,0)^\top = (S_1,0,0,0)^\top$, $\beta \in \mathbb{R}$.

Appendix

Milne-Eddington (ME) model

Parameter	Explanation
$\lvert\boldsymbol{B}\rvert$	Magnitude of the magnetic field vector.
θ	Inclination value of the magnetic field vector. $\theta = 0$ means orientation of the magnetic field towards the observer, $\theta = \pi$ away from the observer.
φ	Azimuth value of the magnetic field vector. $\varphi = 0$ means orientation of the magnetic field towards solar north, $\varphi = \pi/2$ towards solar east.
v_{los}	Line of sight velocity of emitting/absorbing matter.
v_{dop}	Doppler velocity corresponding to the Doppler broadening $\Delta\lambda_D = v_{dop}\lambda/c$ of spectral lines.
a	Damping parameter affecting the spectral line shape. The damping is strong for high values of a and vice versa.
η_0	Line strength parameter for spectral lines. A spectral line is highly pronounced relative to the continuum for high values of η_0 and vice versa.
S_0	Shift parameter for the source function as an affine function of optical depth.
S_1	Scale parameter for the source function as an affine function of optical depth.

Table 1: Summary of the nine Milne-Eddington atmospheric parameters.

Appendix

Neural networks

The basic building block of neural networks is given by the perceptron [Raschka, 2022, pp.338-342].

The perceptron is constructed somewhat analogous to a neuron in the human brain, as fig. 21 shows.

A deep neural network can be build by multiple layers of interconnected perceptrons.

The perceptron as seen in the right panel of fig. 21 has an input $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$ and an output $\bar{x}$.

The quantity $s = b + \sum_{i=1}^{d} x_i \phi_i$ is calculated with weights ϕ_i and a bias b.

The sum s is passed through an suitable (non-linear) activation function $a : \mathbb{R} \to \mathbb{R}$, $s \mapsto a(s)$, which generates the output value $\bar{x}$.

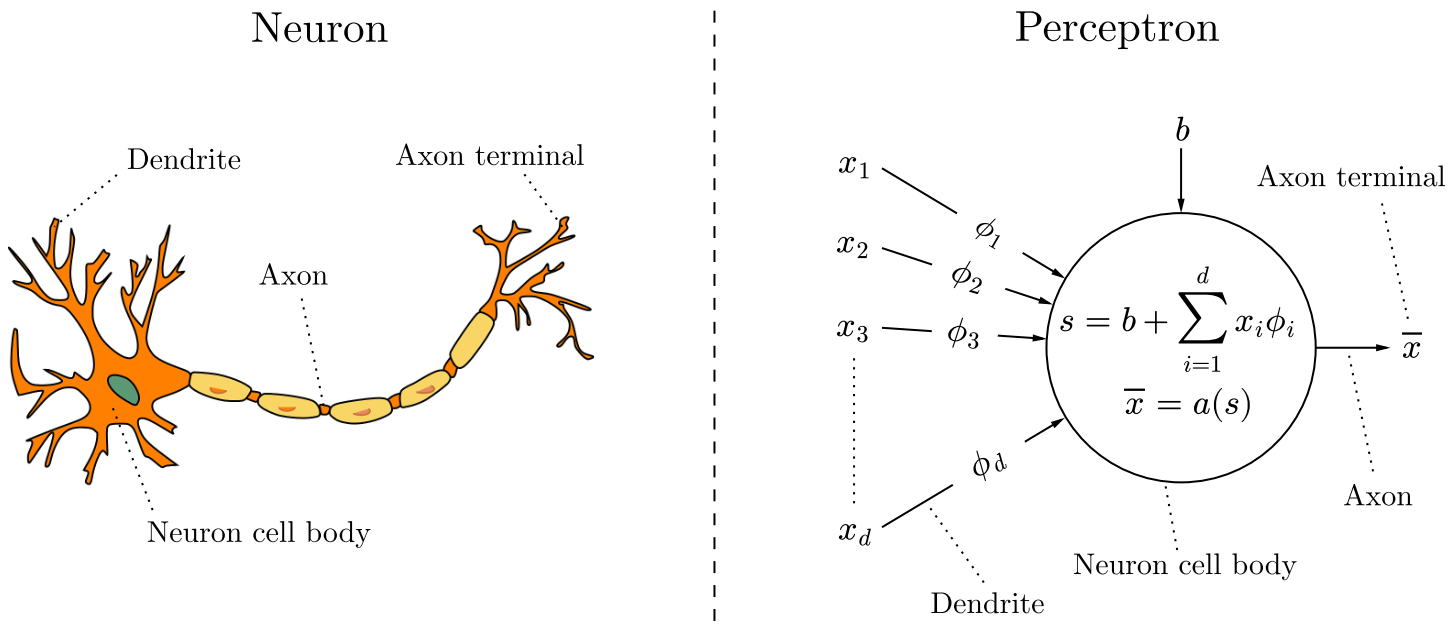


Figure 21: Visualization of the similarities between a human brain neuron and a perceptron. Image credit for image in left panel: Clker-Free-Vector-Images, `https://www.needpix.com/photo/170704/`, accessed on January 02, 2024. Adapted by the author.

In order for a neural network to learn anything, it has to be conditioned by data, whose interpretation is known. This process is known as training.

Training requires a mathematical framework allowing for optimization of the network parameters (weights and biases).

A loss function $\mathcal{L}_\phi$ depending on network parameters ϕ is defined, which is a measure for the discrepancy between network output and true value(s) of the training data.

Neural networks are commonly trained by gradient descent as visualized in fig. 22.

The training process works as follows:

1. Take a start value $\boldsymbol{\phi}_0$ and calculate the value of the loss function $\mathcal{L}_{\boldsymbol{\phi}_0}$ at this point.
2. Update the weights $\boldsymbol{\phi}_i$ according to the rule

$$\boldsymbol{\phi}_{i+1} = \boldsymbol{\phi}_i - \gamma \boldsymbol{\nabla}_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi}_i} \qquad (11)$$

   for $i \in \mathbb{N}$ individual steps and $\gamma \in \mathbb{R}$ a suitable step size as long as $\epsilon \leq |\mathcal{L}_{\boldsymbol{\phi}_{i+1}} - \mathcal{L}_{\boldsymbol{\phi}_i}|$ holds for $\epsilon \in \mathbb{R}$ an accuracy threshold.
3. Terminate, if $\epsilon > |\mathcal{L}_{\boldsymbol{\phi}_{i+1}} - \mathcal{L}_{\boldsymbol{\phi}_i}|$.

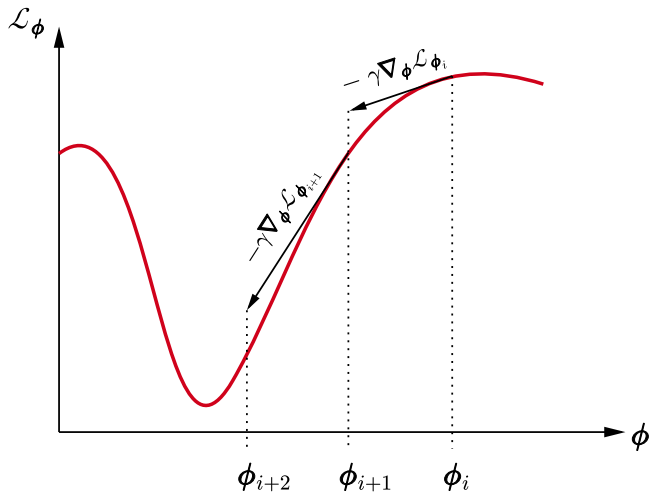


Figure 22: Visualization of a gradient descent routine.

Appendix

Learning curves

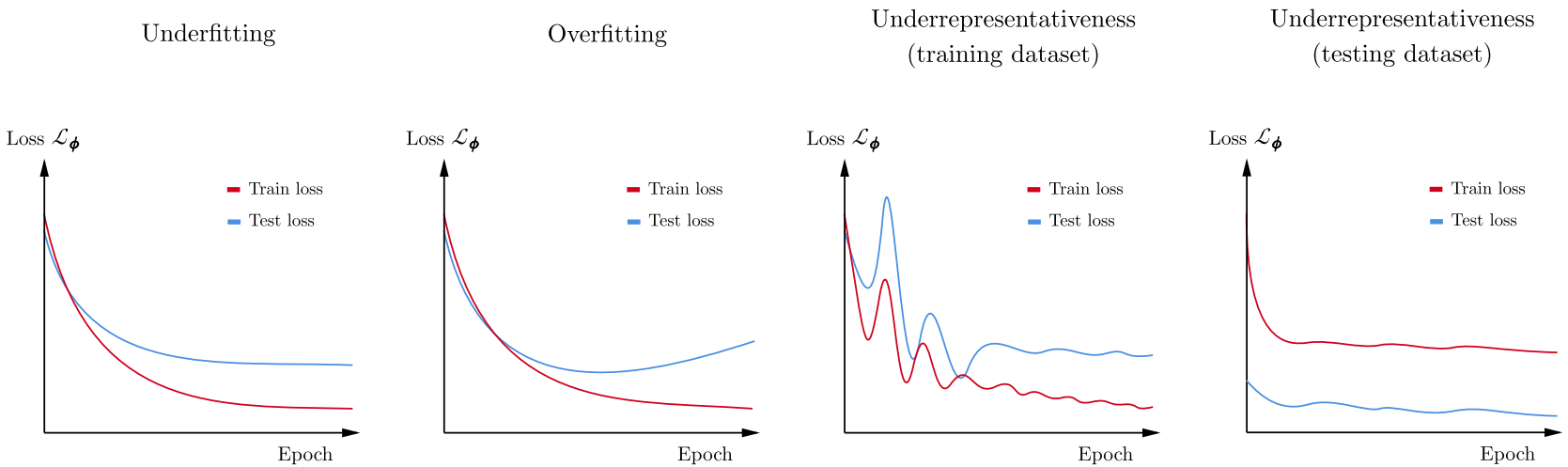


Figure 23: Examples for the four basic cases of non-ideal model performance. Sketches inspired by Jason Brownlee, `https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/`, accessed on December 18, 2023.

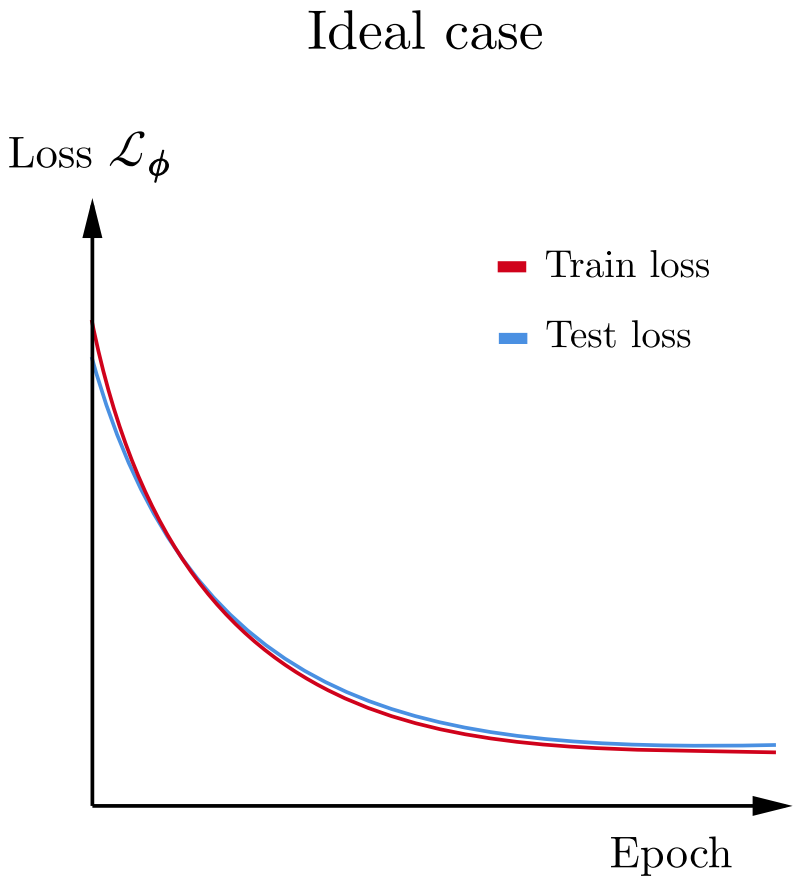


Figure 24: Example of an ideal learning curve.

A so-called flow (or normalizing flow) is a type of generative model, as e.g. GAN's and VAE's are [Weng, 2018].

The principle of a flow is visualized in fig. 25.

A flow is called a normalizing flow, if the latent distribution $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$ is chosen as a standard normal distribution.

An arbitrary probability density $p_{\boldsymbol{x}}(\boldsymbol{x})$ is mapped to the latent density $p_{\boldsymbol{z}}(\boldsymbol{z})$ by the diffeomorphic flow $\boldsymbol{f}_{\boldsymbol{\phi}}(\boldsymbol{x})$.

Flows can be trained by the negative log-likelihood, which can be exactly calculated using the specific transformation $\boldsymbol{f}_{\phi}$ used as a flow in combination with the change of variable theorem.

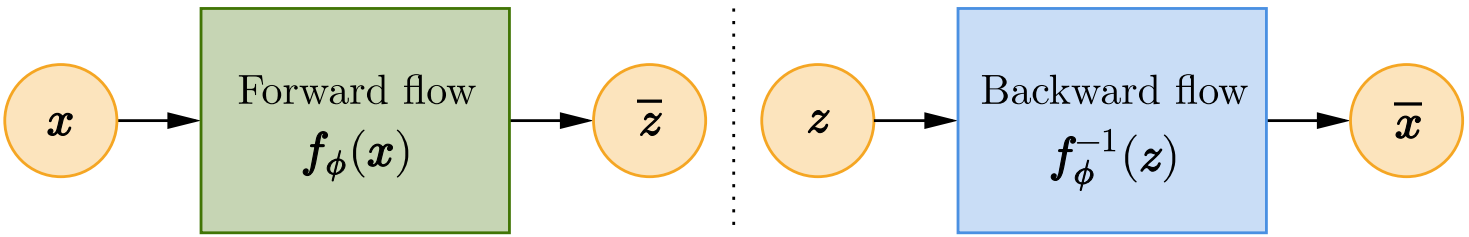


Figure 25: Working principle of a flow-based generative model. Figure inspired by [Weng, 2018].

Appendix

How does a normalizing flow work?

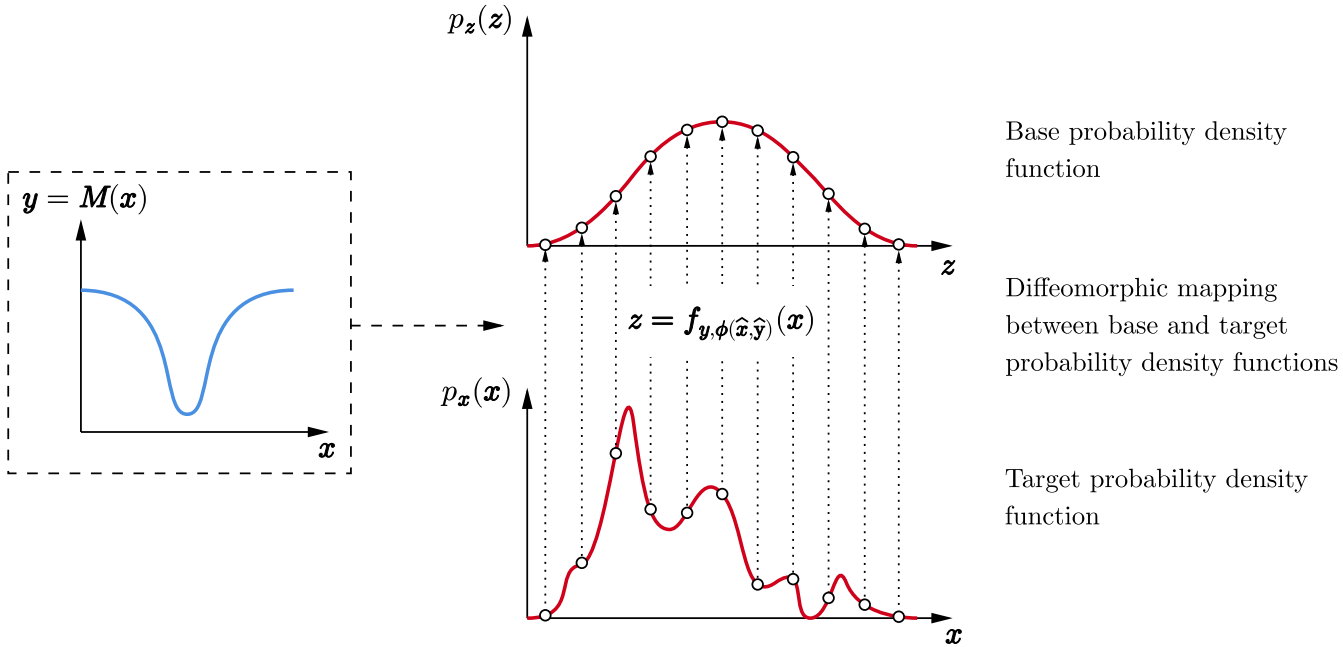


Figure 26: Conceptual principle of a conditional normalizing flow $\boldsymbol{z} = \boldsymbol{f}_{\boldsymbol{y},\boldsymbol{\phi}(\hat{\mathbf{x}},\hat{\mathbf{y}})}(\boldsymbol{x})$, transforming a target probability density function $p_{\boldsymbol{x}}(\boldsymbol{x})$ to a base probability density function $p_{\boldsymbol{z}}(\boldsymbol{z})$ conditional on $\boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x})$, where $\boldsymbol{M}$ is a model that relates the data $\boldsymbol{x}$ to the conditioning data $\boldsymbol{y}$.

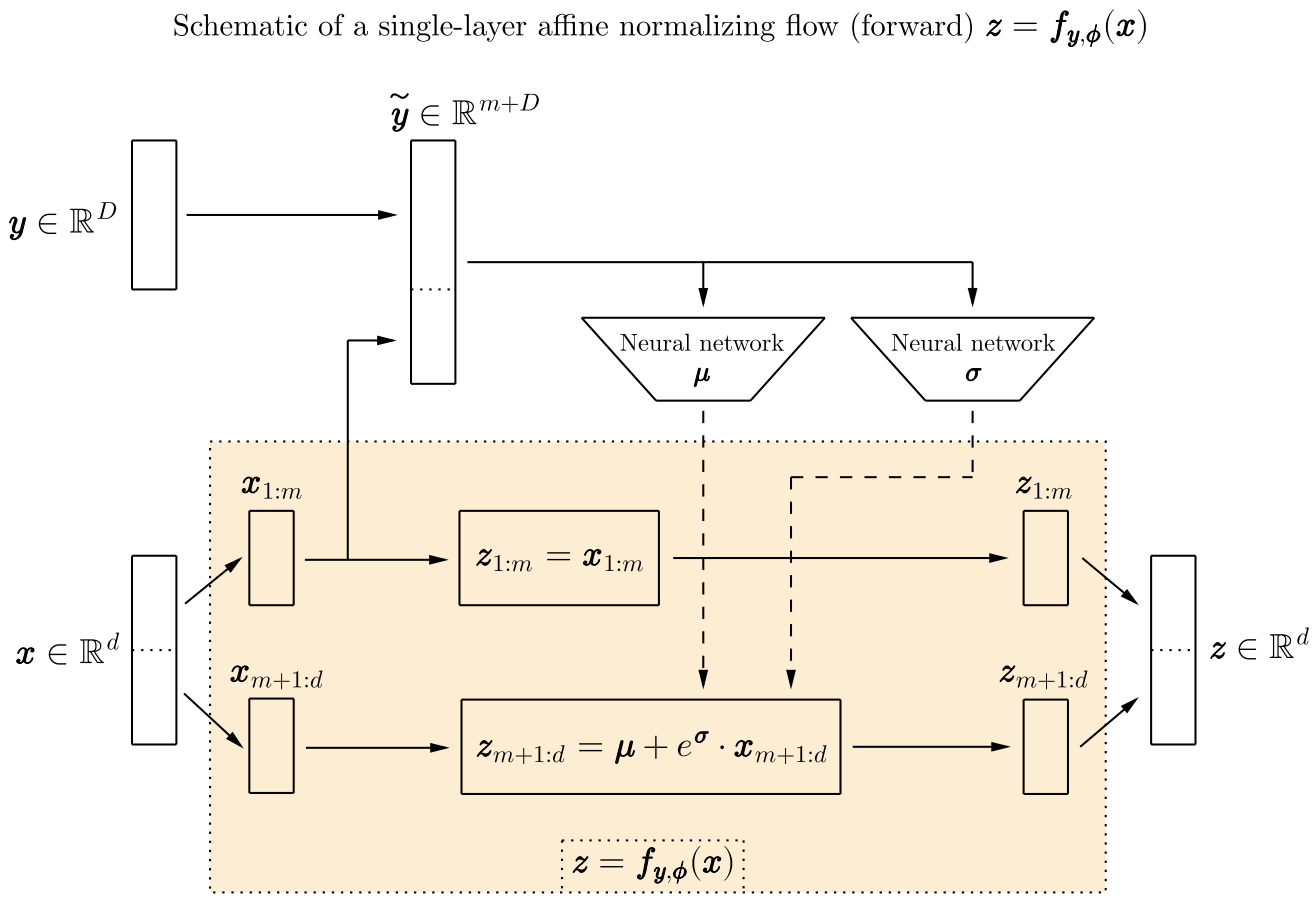


Figure 27: Single-layer affine (coupling layer) normalizing flow (forward).

Normalizing flows

Affine coupling layer normalizing flows

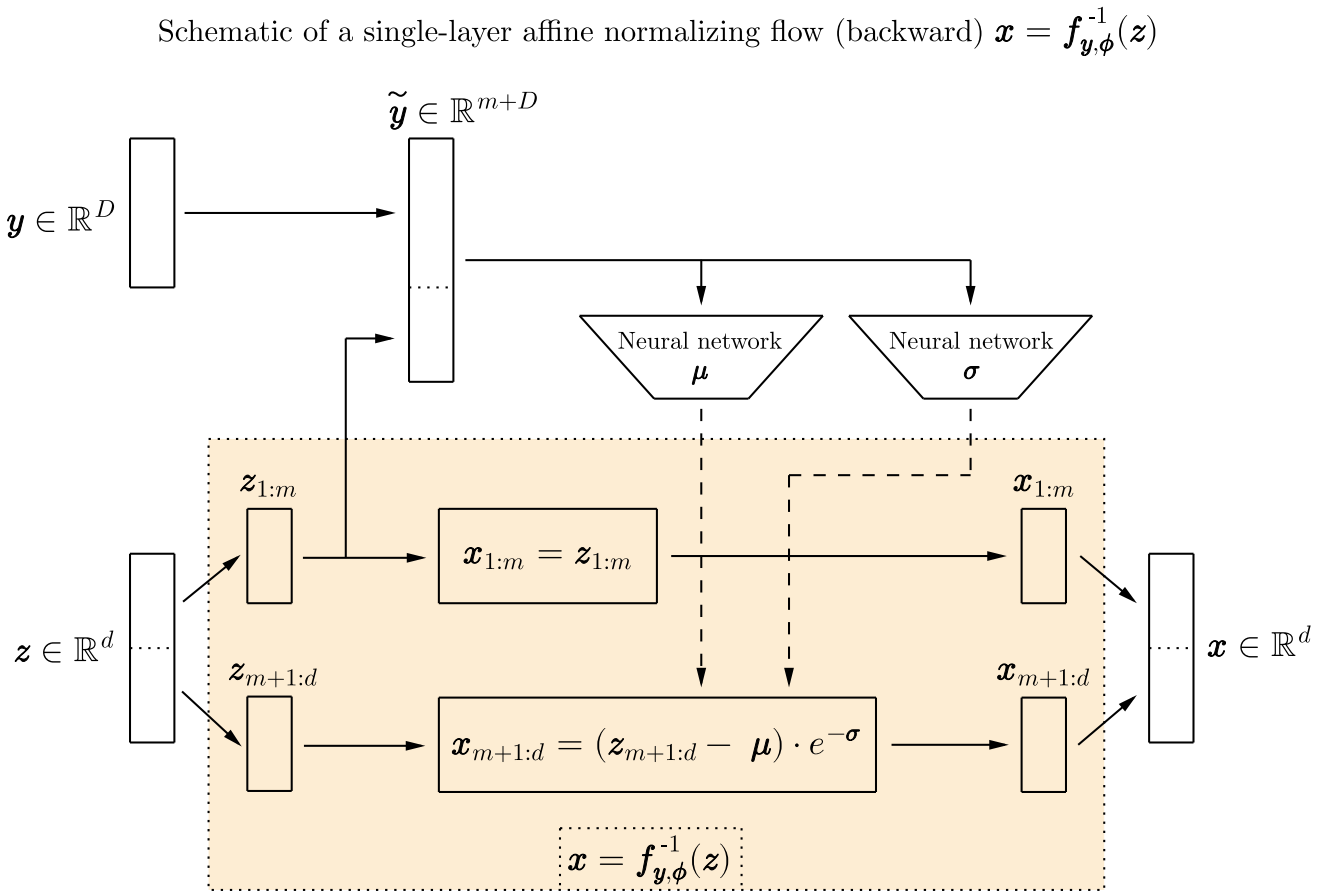


Figure 28: Single-layer affine (coupling layer) normalizing flow (backward).

Appendix

Piecewise quadratic spline coupling layer flows

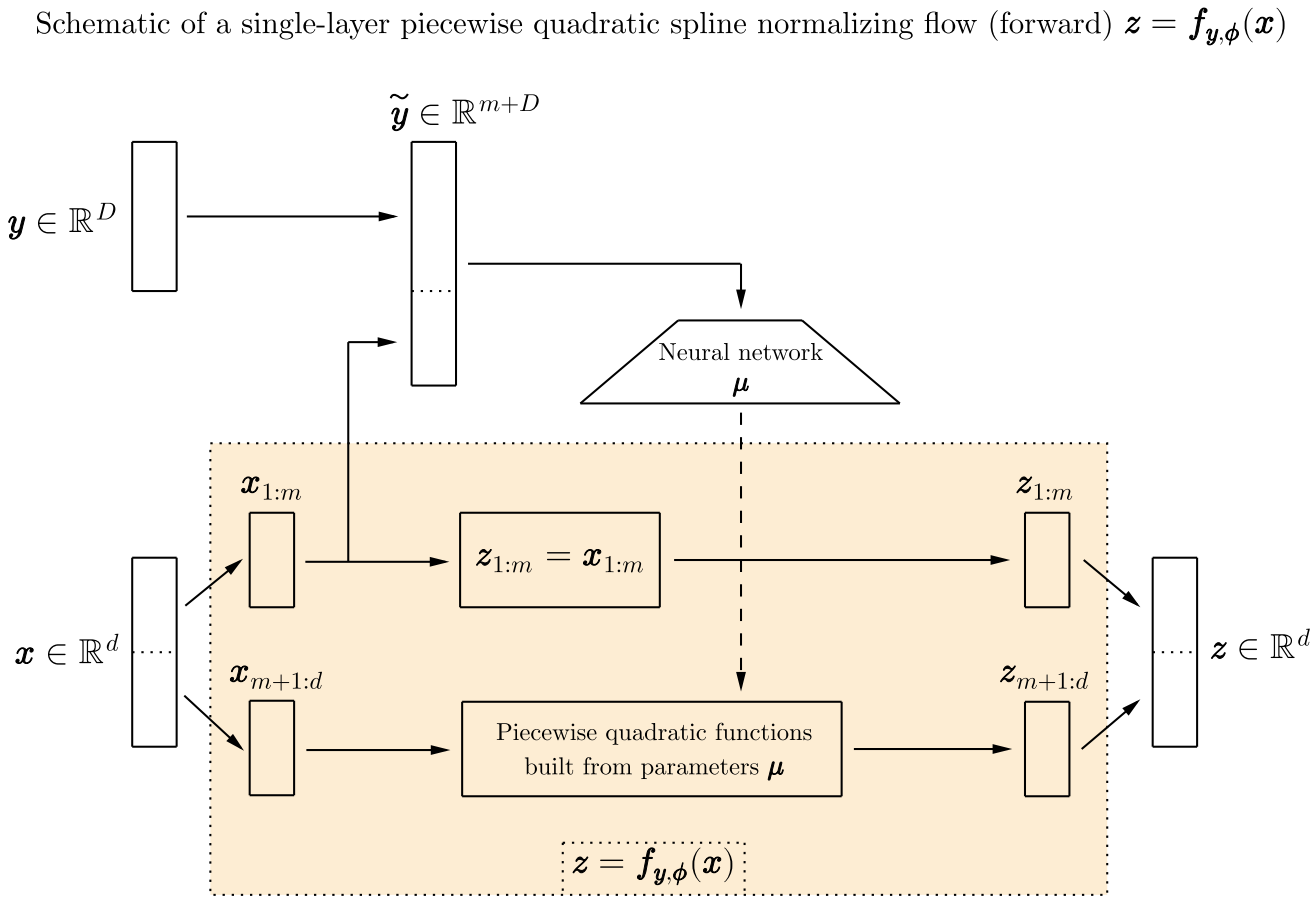


Figure 29: Single-layer piecewise quadratic spline (coupling layer) normalizing flow (forward).

Appendix

Piecewise quadratic spline coupling layer flows

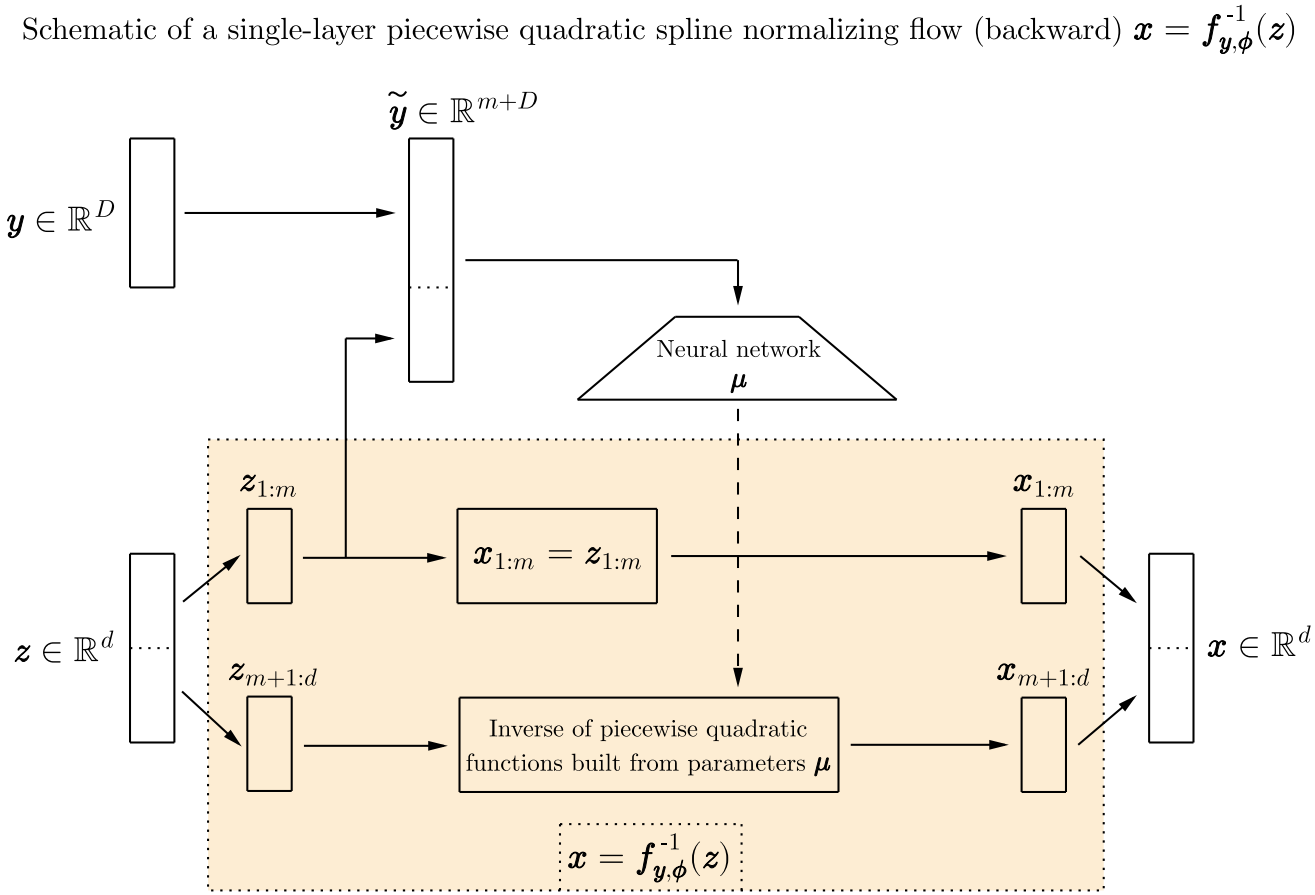


Figure 30: Single-layer piecewise quadratic spline (coupling layer) normalizing flow (backward).

Appendix

What has been done yet using normalizing flows?

Normalizing flows were introduced in 2015 by [Rezende and Mohamed, 2015] as a new solution to approximate posterior probability densities $p(\boldsymbol{x}|\boldsymbol{y})$.

A first attempt to apply normalizing flows as generative model for image modeling was made by [Dinh et al., 2016] in 2016. They demonstrated that normalizing flows can generate very similar images as those they were trained on.

[Durkan et al., 2019] introduced the piecewise rational quadratic spline coupling layer flows and showed, that they are very flexible for approximating multimodal posterior densities.

Finally, [Díaz Baso et al., 2022] applied normalizing flows to learn stellar atmospheric inversions. They successfully trained and evaluated normalizing flows using the scalar Milne-Eddington (no magn. field parameters) and a more complex non-LTE atmospheric model.

The thesis at hand comes in here to explore normalizing flows applied to learn the full Milne-Eddington inversion task (with magn. field parameters).

The change of variable formula is the centerpiece of the normalizing flow technique.

Suppose that $\boldsymbol{x}$, $\boldsymbol{z} \in \mathbb{R}^d$ are multidimensional random variables with associated probability densities $p_{\boldsymbol{x}}(\boldsymbol{x})$ and $p_{\boldsymbol{z}}(\boldsymbol{z})$. Let furthermore $\boldsymbol{f}$ be a function transferring the variable $\boldsymbol{x}$ into $\boldsymbol{z}$ as $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$.

[Deisenroth et al., 2020, p.196] give the change of variable theorem as

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = p_{\boldsymbol{z}}[\boldsymbol{f}(\boldsymbol{x})] \left| \det \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right|. \tag{12}$$

Appendix

Principle/requirements of/for a normalizing flow

Suppose, that a function $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$, $\boldsymbol{x} \mapsto \boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$ is a normalizing flow ($p_{\boldsymbol{z}}(\boldsymbol{z})$ being a standard normal distribution). The normalizing flow can be composed of several functions $\boldsymbol{f}_{(k)}$ such that $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}_{(n)} \circ \cdots \circ \boldsymbol{f}_{(1)}(\boldsymbol{x})$.

[Kobyzev et al., 2021] give three key requirements to the $\boldsymbol{f}_{(k)}$ such that they indeed compose a normalizing flow:

1. All $\boldsymbol{f}_{(k)}$ need to be invertible and differentiable.
2. The transformations $\boldsymbol{f}_{(k)}$ need to be expressive enough to model complex data distributions.
3. The transformations $\boldsymbol{f}_{(k)}$ are ideally computationally efficient, such that the computation of the Jacobian requires as little resources as possible.

Appendix

Moons experiment: Explanation

As an instructive example, an affine coupling layer normalizing flow was trained to map samples from a 2D "moons" target distribution to a 2D standard normal base distribution, as visualized in fig. 31.

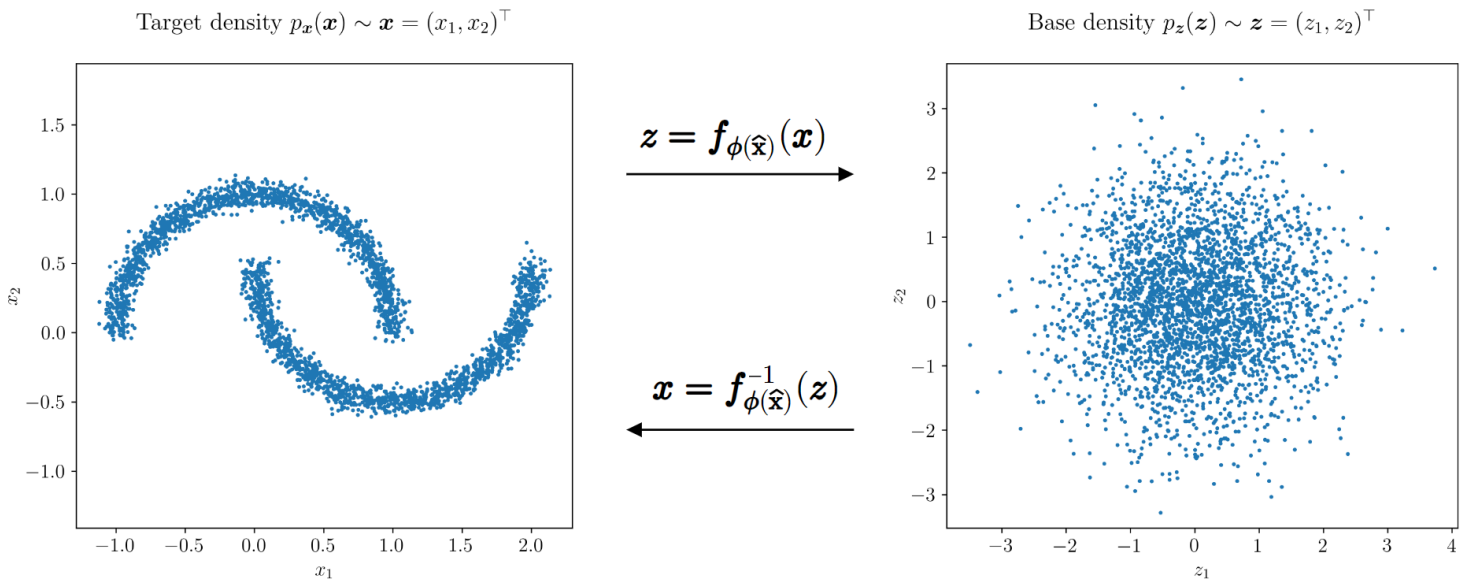


Figure 31: Visualization of a normalizing flow mapping the moons distribution to a standard normal distribution and vice versa in a plane.

The goal was to find a diffeomorphic (differentiable and invertible) function mapping a 2D standard normal distribution $\boldsymbol{z}$ to the distribution points $\boldsymbol{x}$ of the moons distribution as visualized in fig. 32.

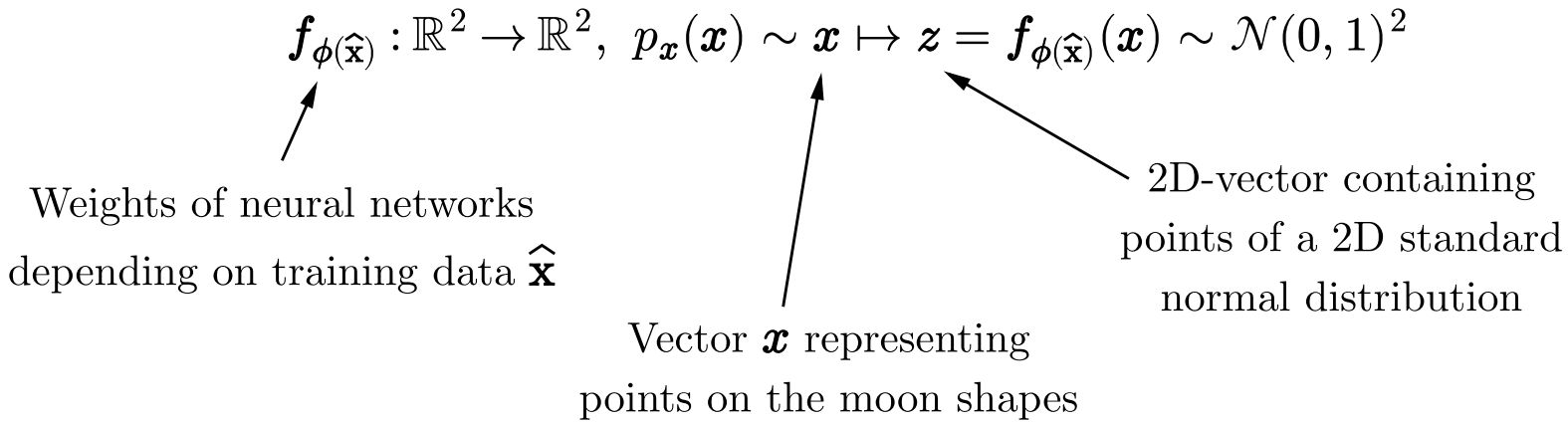


Figure 32: Goal for the normalizing flow applied to the moons experiment.

Appendix

Moons experiment: Results

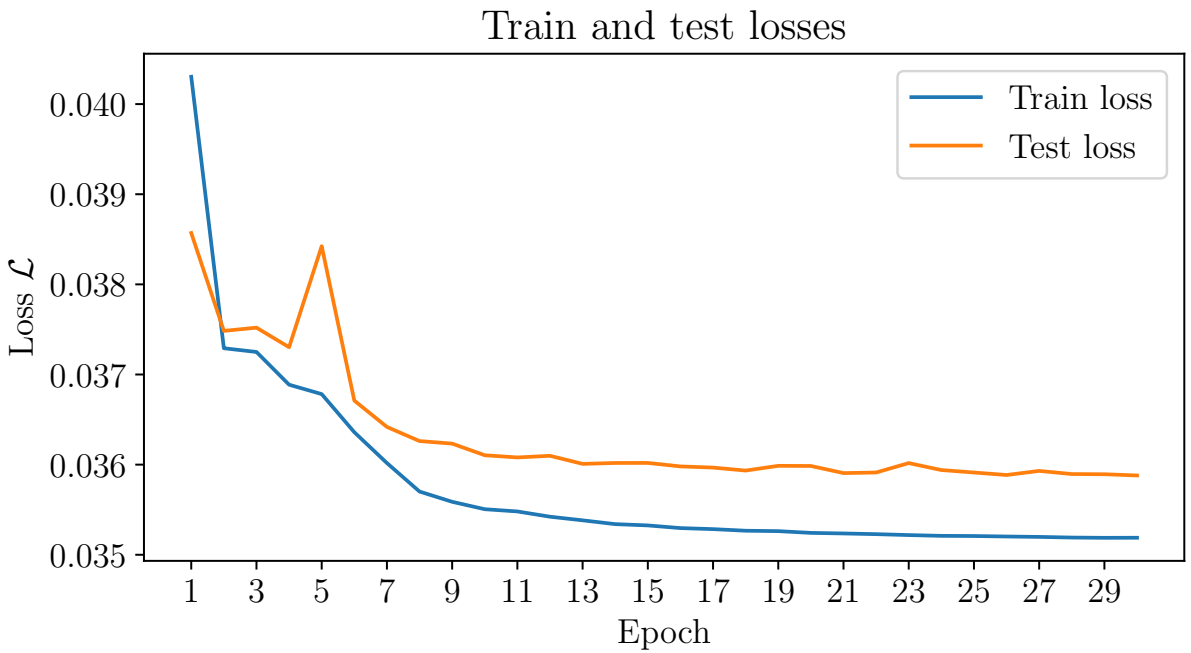


Figure 33: Learning curves for the model implementing a map from the moons distribution to a two-dimensional standard normal distribution.

Appendix

Moons experiment: Results

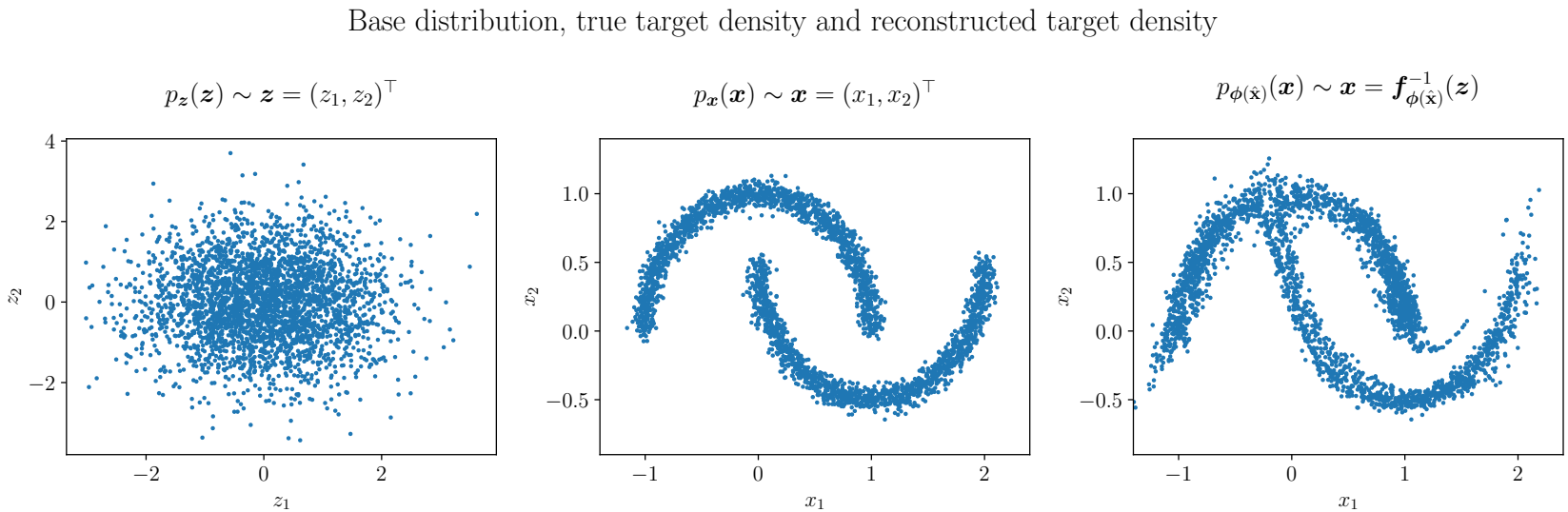


Figure 34: Results for a normalizing flow implementing a map from the moons distribution to a standard normal distribution in a plane (3000 samples).

There are three key points to take away from these results:

1. Model is possibly experiencing overfitting and potentially underrepresentativeness in training or testing data.
2. The reconstructed target density features struggles to separate the two half-moons close to the origin.
3. Generally speaking however, the normalizing flow technique seems to work as intended.

Appendix

Linear regression experiment: Explanation

As an example, an affine coupling layer normalizing flow was trained on sets of parameters $\boldsymbol{x} = (a, b)^\top$ and associated lines (context) $\boldsymbol{y} = (y_1, \ldots, y_{10})^\top$ with added noise.

$\boldsymbol{x} \sim p_{\boldsymbol{x}}(\boldsymbol{x})$ and $\boldsymbol{y} \sim p_{\boldsymbol{y}}(\boldsymbol{y})$ are intertwined by $y_j = M_j(\boldsymbol{x}) = au_j + b$ for $j \in \{1, \ldots, 10\}$.

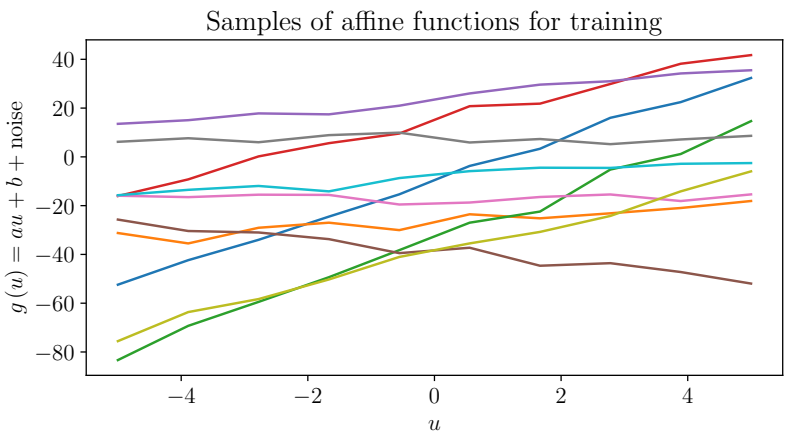


Figure 35: Samples from the training dataset of context used in the linear regression example.

Appendix

Linear regression experiment: Explanation

The goal was to find a diffeomorphic (differentiable and invertible) function mapping a 2D standard normal distribution $\boldsymbol{z}$ to the distribution of linear regression parameters $\boldsymbol{x}$ as specified in fig. 36.

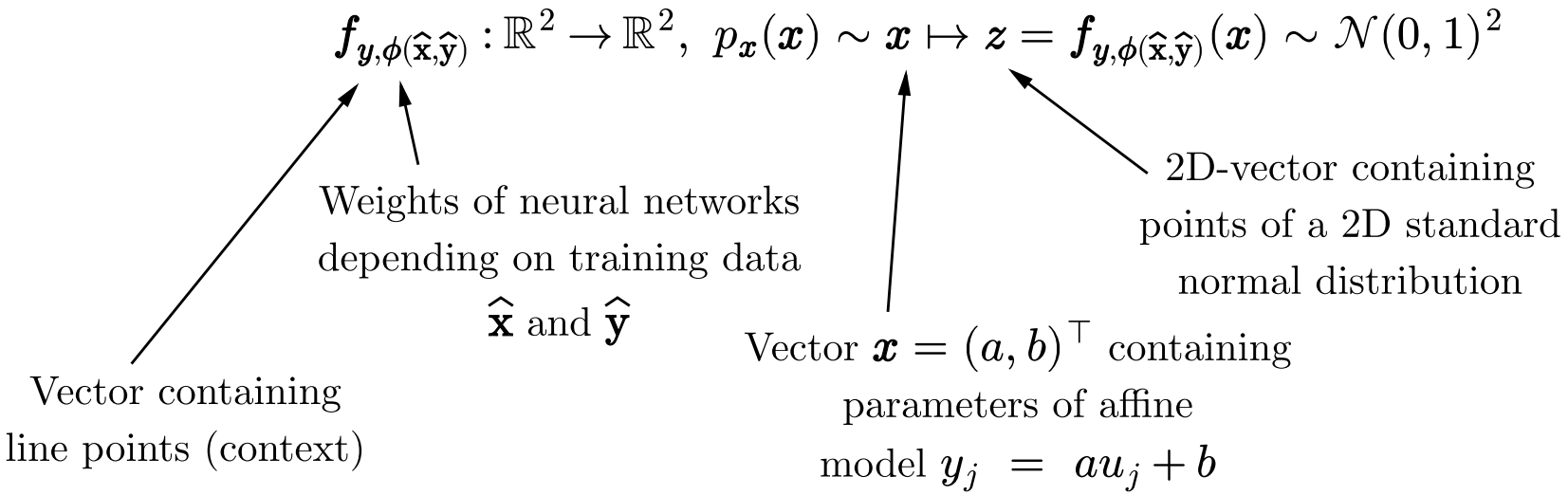


Figure 36: Goal for the normalizing flow applied to the linear regression experiment.

Appendix

Linear regression experiment: Results

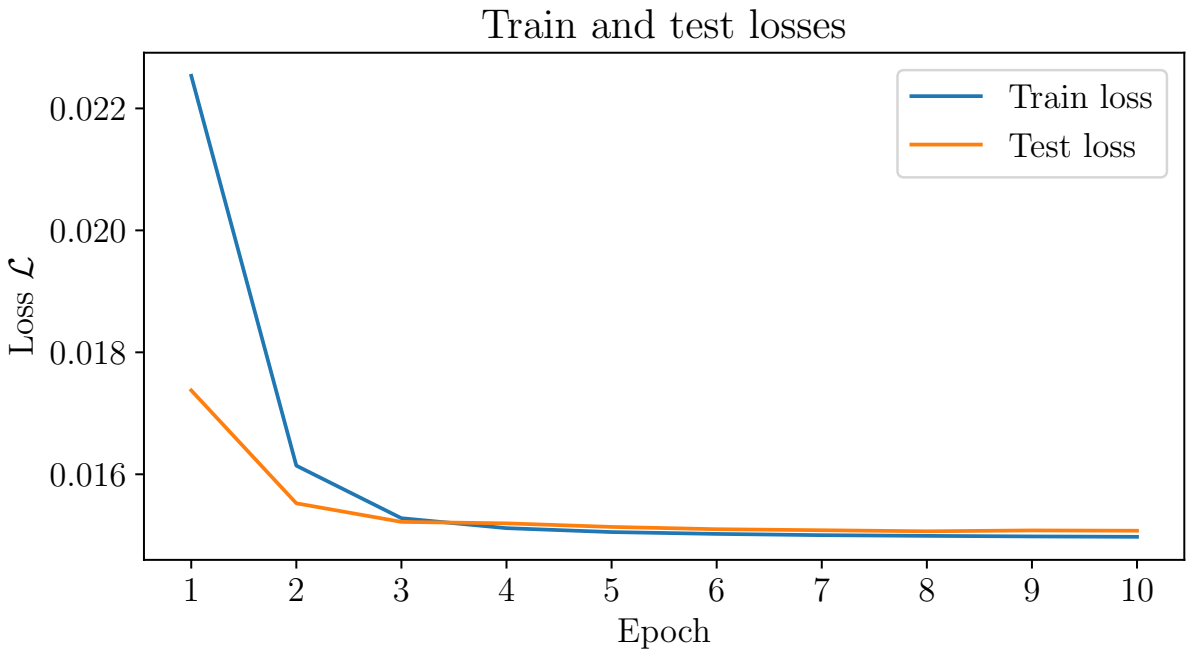


Figure 37: Learning curves for the model implementing a map from parameter distributions of an affine function to a two-dimensional standard normal distribution, given the affine function values associated with the function parameters as context.

Appendix

Linear regression experiment: Results

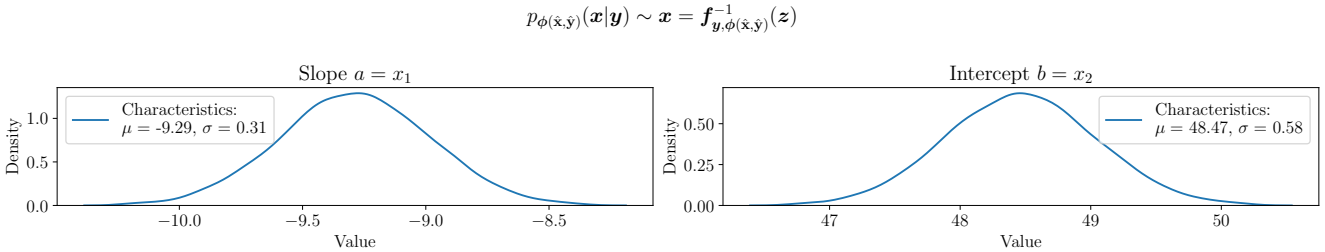


Figure 38: Calculated densities for a test observation $\boldsymbol{y} = \boldsymbol{y}_{test}$ generated by parameters $\boldsymbol{x}_{test} = (-9.34, 47.64)^\top$ given to the normalizing flow.

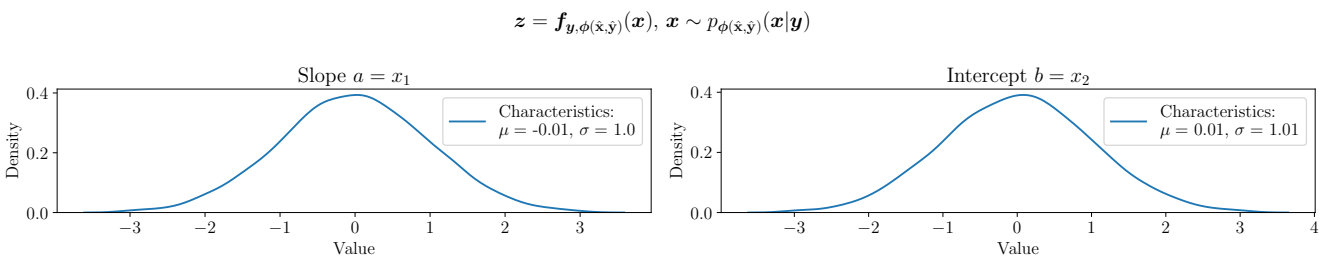


Figure 39: Calculated latent densities for the parameter densities in the training dataset.

Appendix

Linear regression experiment: Results

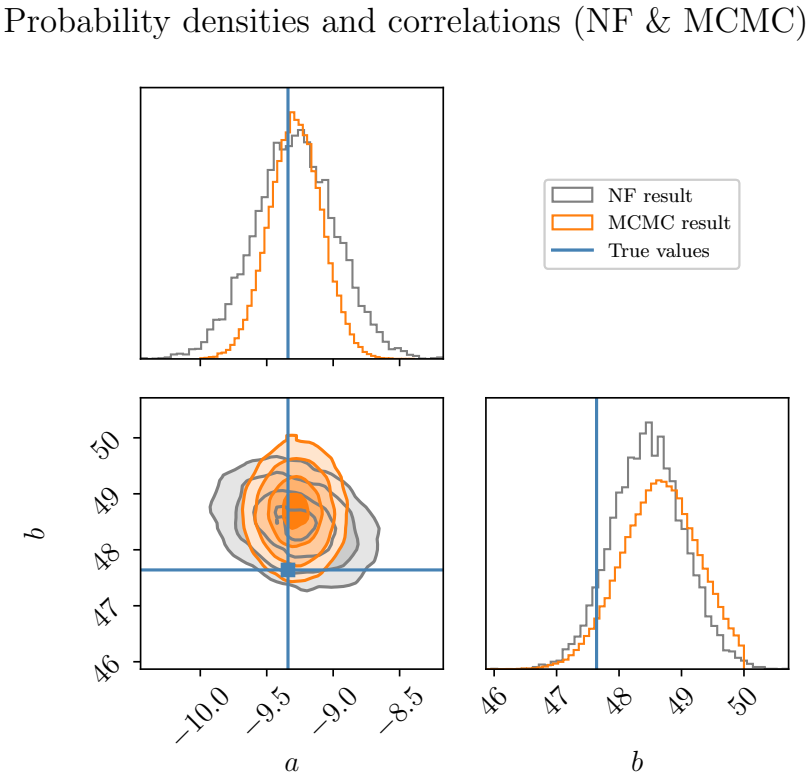


Figure 40: Comparison of normalizing flow (NF) and Markov Chain Monte Carlo simulation (MCMC) results for a test observation $\boldsymbol{y} = \boldsymbol{y}_{test}$ generated by test parameters $\boldsymbol{x} = \boldsymbol{x}_{test}$.

Appendix

Linear regression experiment: Results

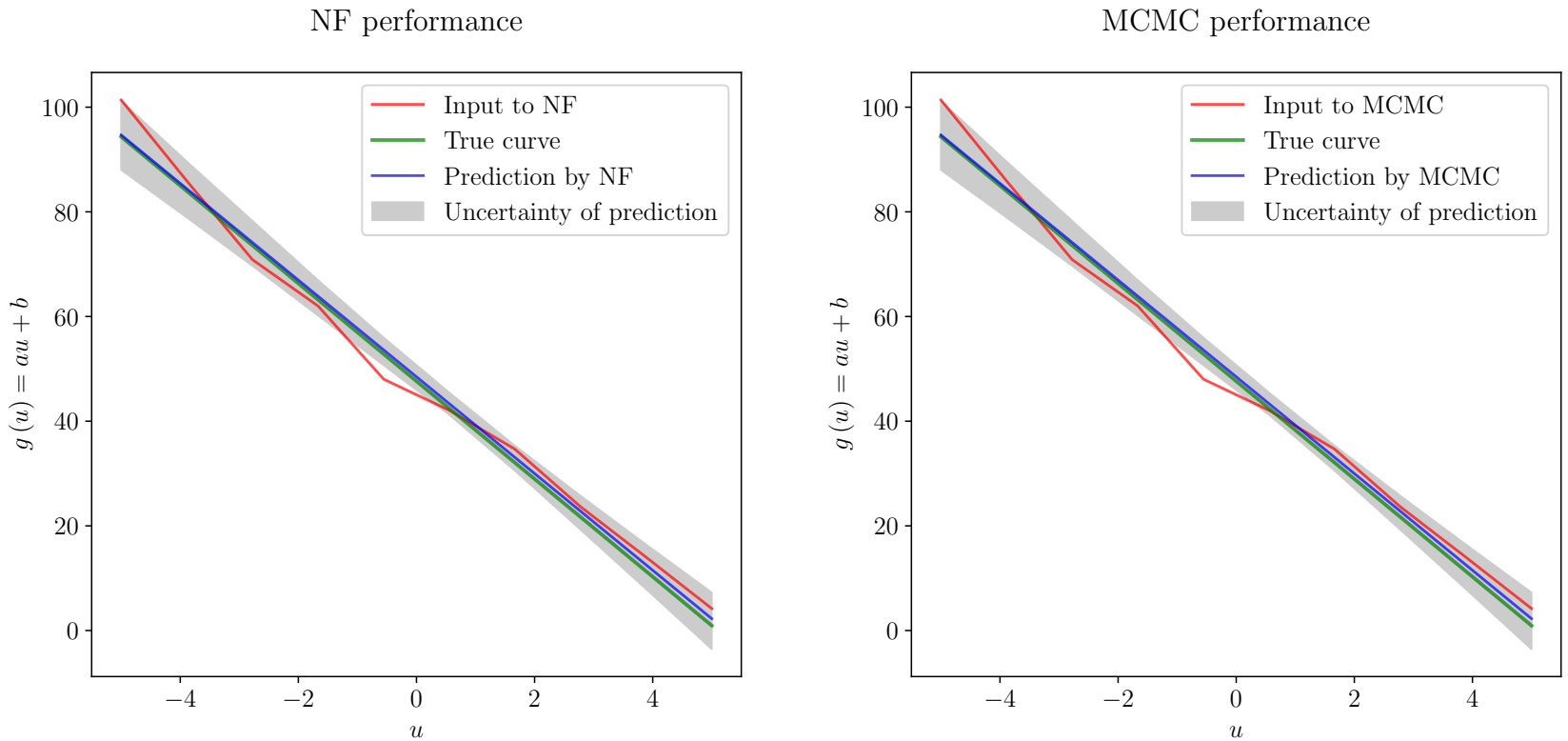


Figure 41: Left panel: Results obtained by the normalizing flow (NF) for $\boldsymbol{y} = \boldsymbol{y}_{test}$ and $\boldsymbol{x} = \boldsymbol{x}_{test}$. Right panel: Results obtained by the Markov Chain Monte Carlo simulation (MCMC) for the same test observation $\boldsymbol{y}_{test}$.

There are three key points to take away from these results:

1. The prediction by the normalizing flow almost everywhere partially to fully overlaps with the true curve.
2. The prediction and uncertainty regions as obtained by the normalizing flow agree with those obtained by the Markov Chain Monte Carlo simulation.
3. The uncertainty region and the prediction itself seems to fit well with the input data, on which the linear regression was carried out.

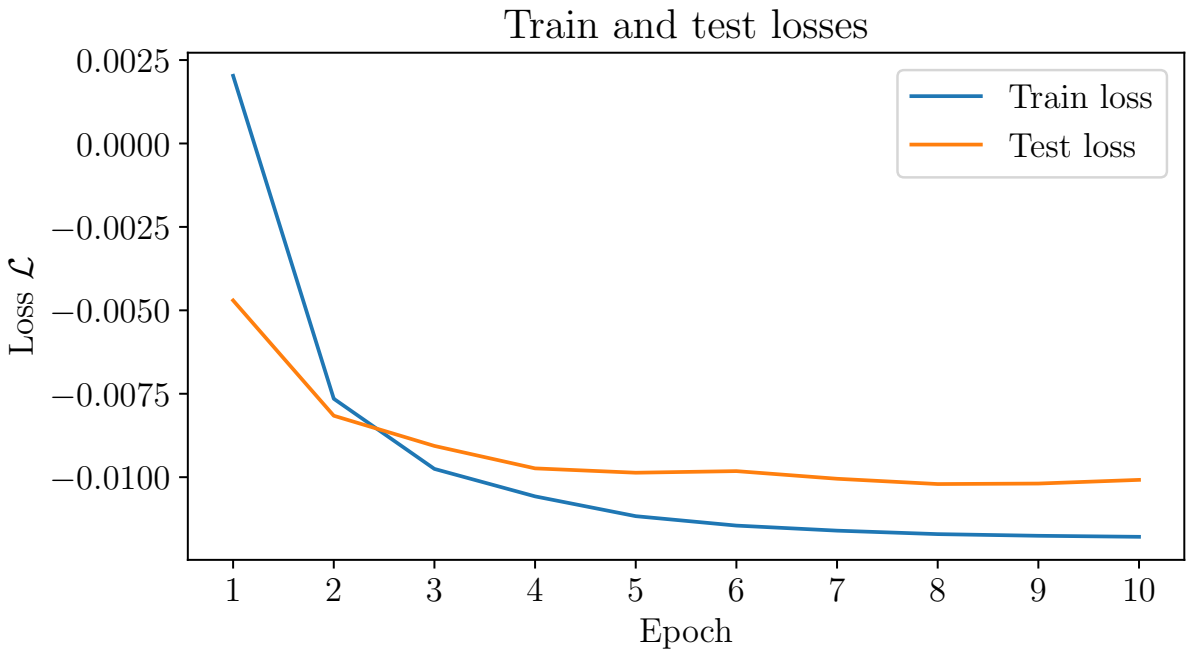


Figure 42: Learning curve for multiple map analysis experiment.

Appendix

Multiple map analysis using a different map for training: Experiment explanation

This experiment basically was carried out as the previous experiment, but instead of using frame 0 of the penumbra formation dataset for training, a frame from a different active region of the sun was used (Stokes profiles $\hat{\mathbf{y}}$ and associated atmospheric parameters $\hat{\mathbf{x}}$ obtained by the Milne-Eddington algorithm).

Once trained, the normalizing flow was applied to invert frames from the penumbra formation dataset.

Appendix

Multiple map analysis using a different map for training: Experiment explanation

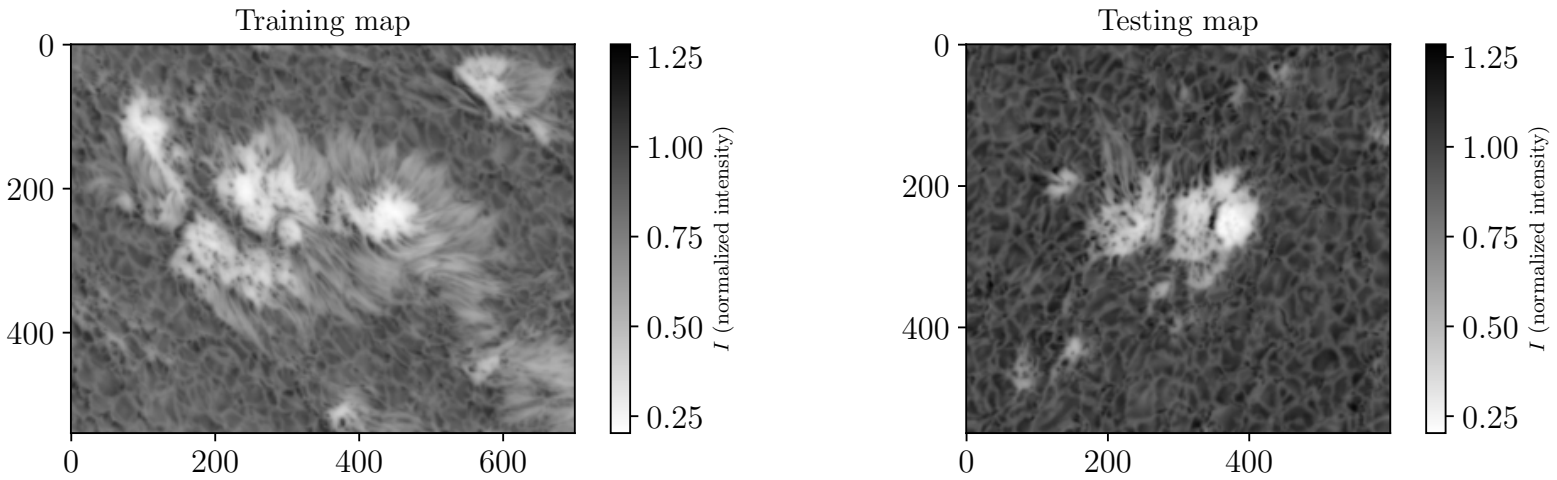


Figure 43: Left panel: Frame used for training the normalizing flow. Right panel: Frame to be inverted by trained normalizing flow.

Appendix

Multiple map analysis using a different map for training: Results

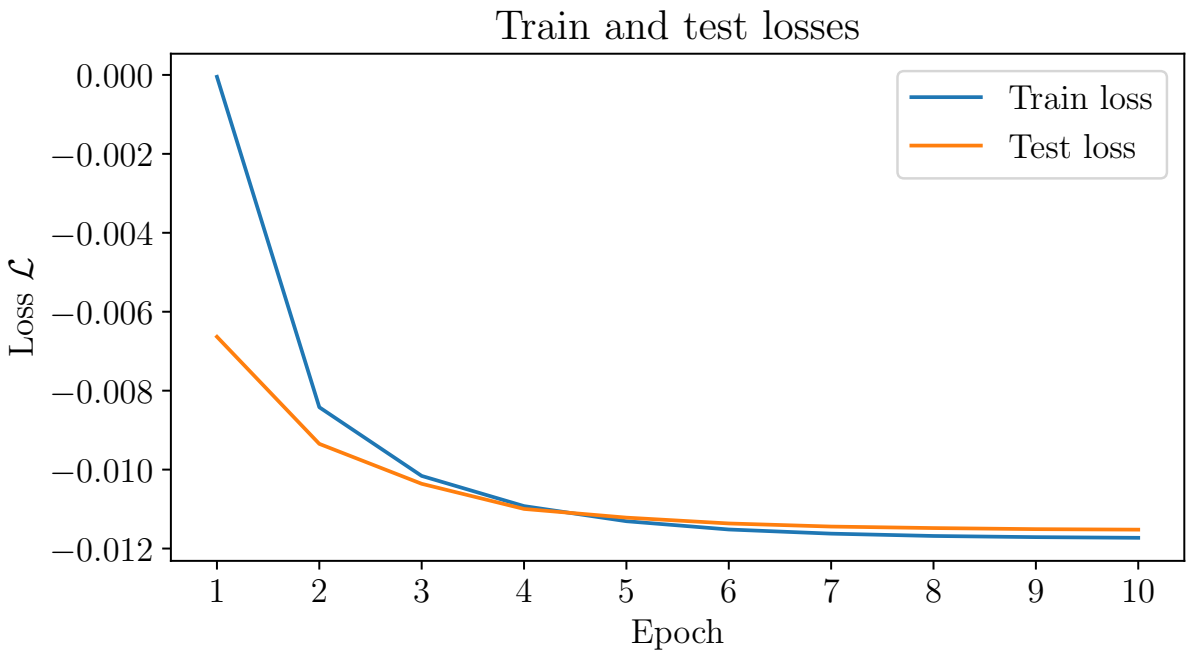


Figure 44: Learning curve for experiment 4 of normalizing flows applied to learn Milne-Eddington inversions.

Appendix

Multiple map analysis using a different map for training: Results

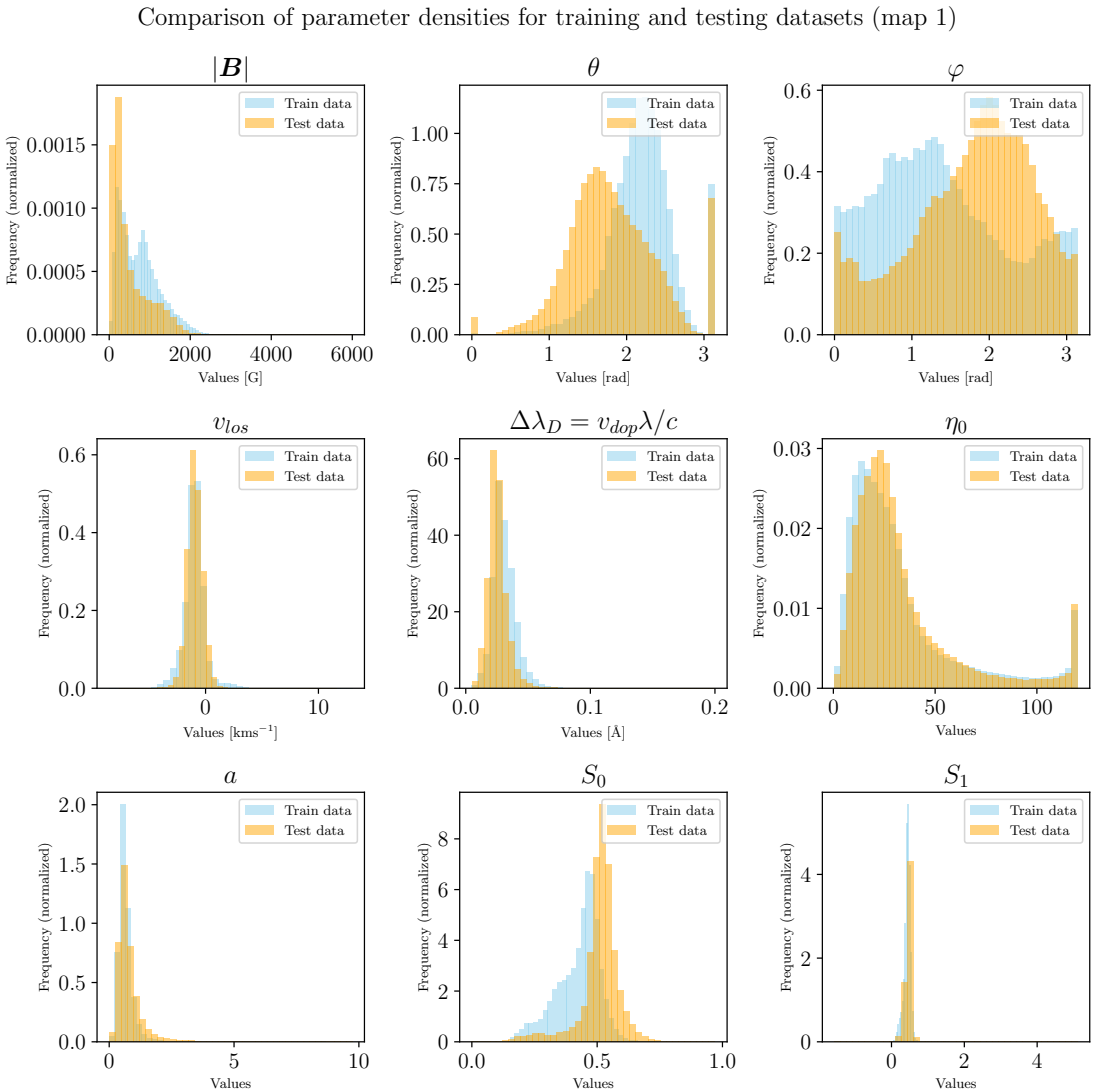
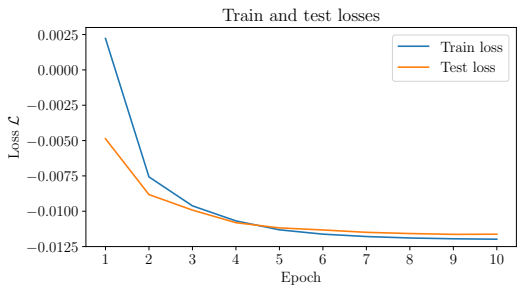


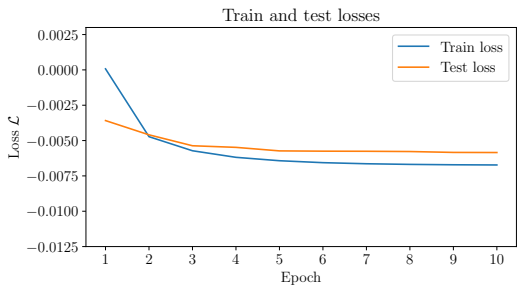
Figure 45: Visualization of variations in training and testing data. This plot shows data from the sunspot map as training data; and data of frame 0 (test map 1) from the penumbra formation dataset as testing data.

Results and discussion

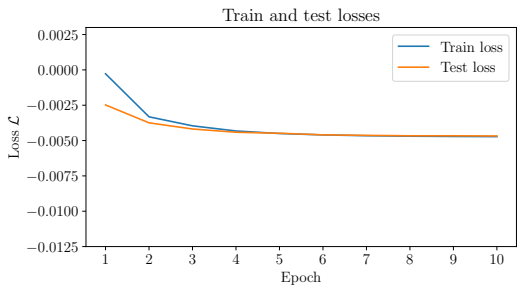
Improving normalizing flow performance by enhancing the training dataset: Results



(a) Normalizing flow trained on one frame.



(b) Normalizing flow trained on two frames.



(c) Normalizing flow trained on three frames.

Figure 46: Train & test losses for different sizes of train/test datasets.

Appendix

Multiple map analysis using a different map for training: Results

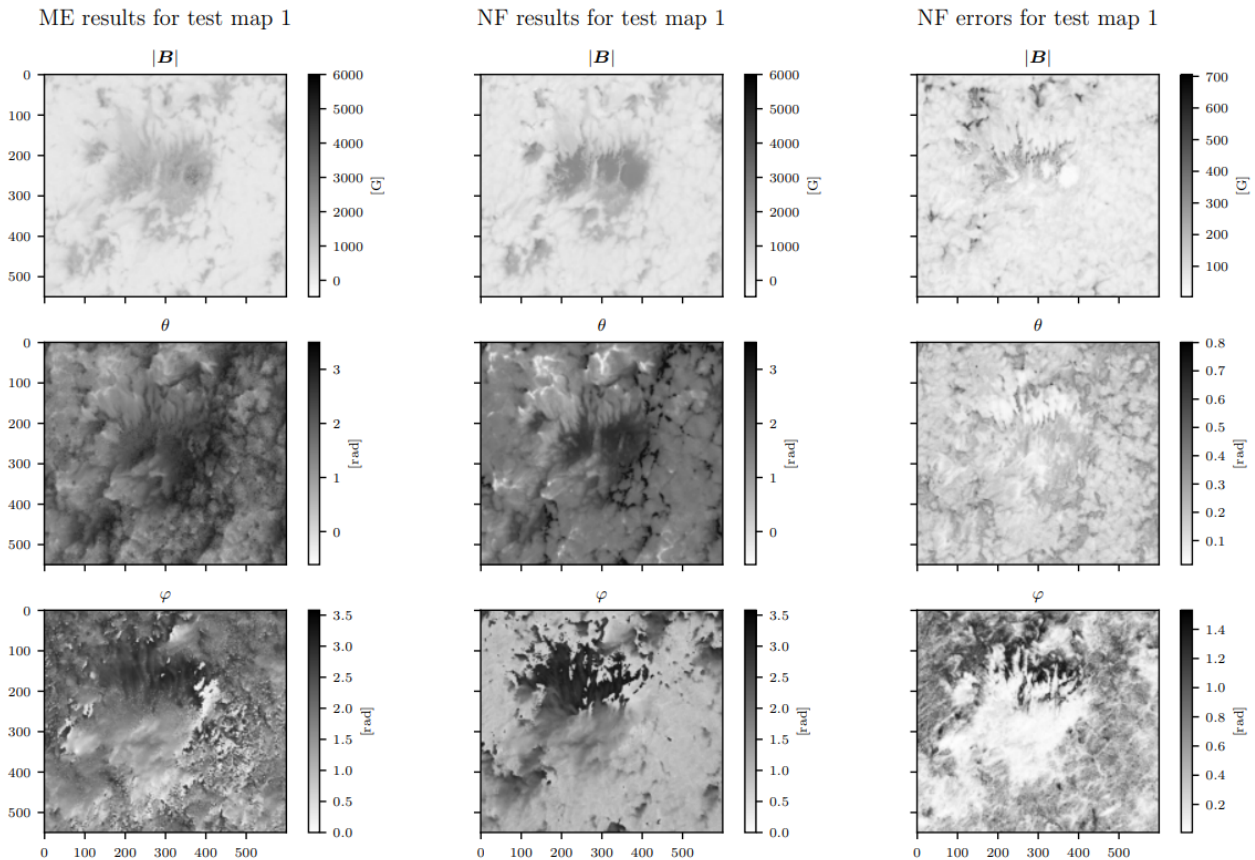


Figure 47: Results for multiple map analysis using a different map for training experiment.

There are three key points to take away from these results:

1. The training and testing losses show nearly ideal behaviour; after about 7 epochs, they seem to converge to a nearly constant value with little spacing.
2. The "smoothing property" seems to be stronger in this case than in the previous case.
3. Results for magnetic field parameters are of similar quality than in the previous case, except for the azimuth parameter φ.

Appendix

Tracking parameter evolution during penumbra formation: Experiment explanation

For this experiment, the observational data $\hat{\mathbf{y}}$ and the associated parameter data $\hat{\mathbf{x}}$ of frame 0 was used to train a normalizing flow.

Using this trained flow, all subsequent frames of the penumbra formation dataset were inverted.

In the penumbra formation dataset, a penumbra can be observed to form in the area of the red rectangle in fig. 48.

Appendix

Tracking parameter evolution during penumbra formation: Experiment explanation

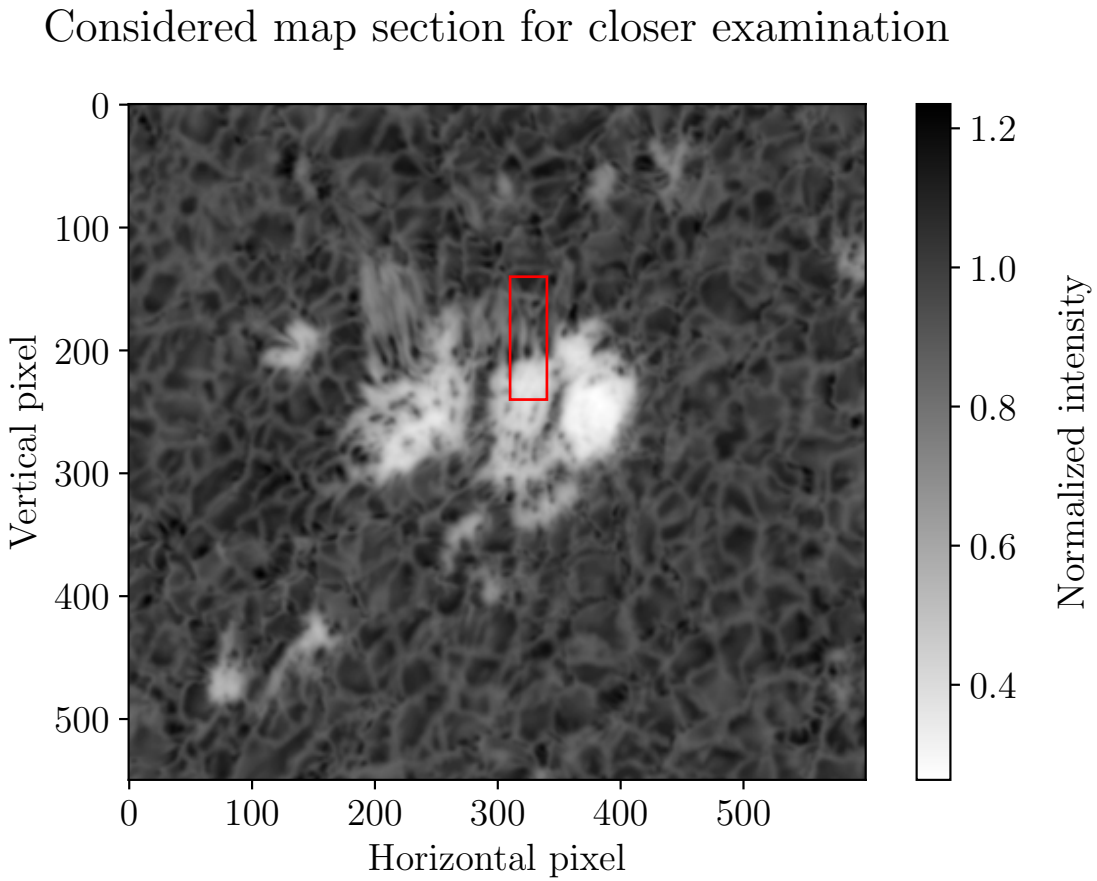


Figure 48: Area of particular interest for atmospheric parameter evolution during penumbra formation.

Appendix

Tracking parameter evolution during penumbra formation: Results

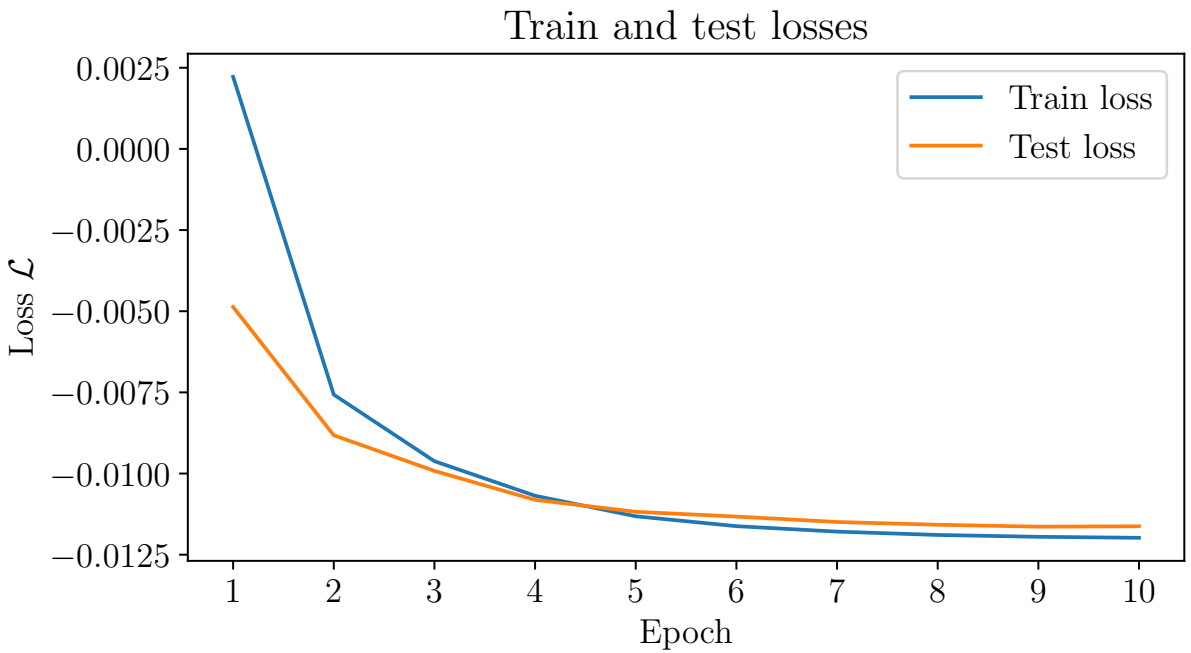


Figure 49: Train and test losses of normalizing flow training process for parameter evolution experiment.

Appendix

Tracking parameter evolution during penumbra formation: Results

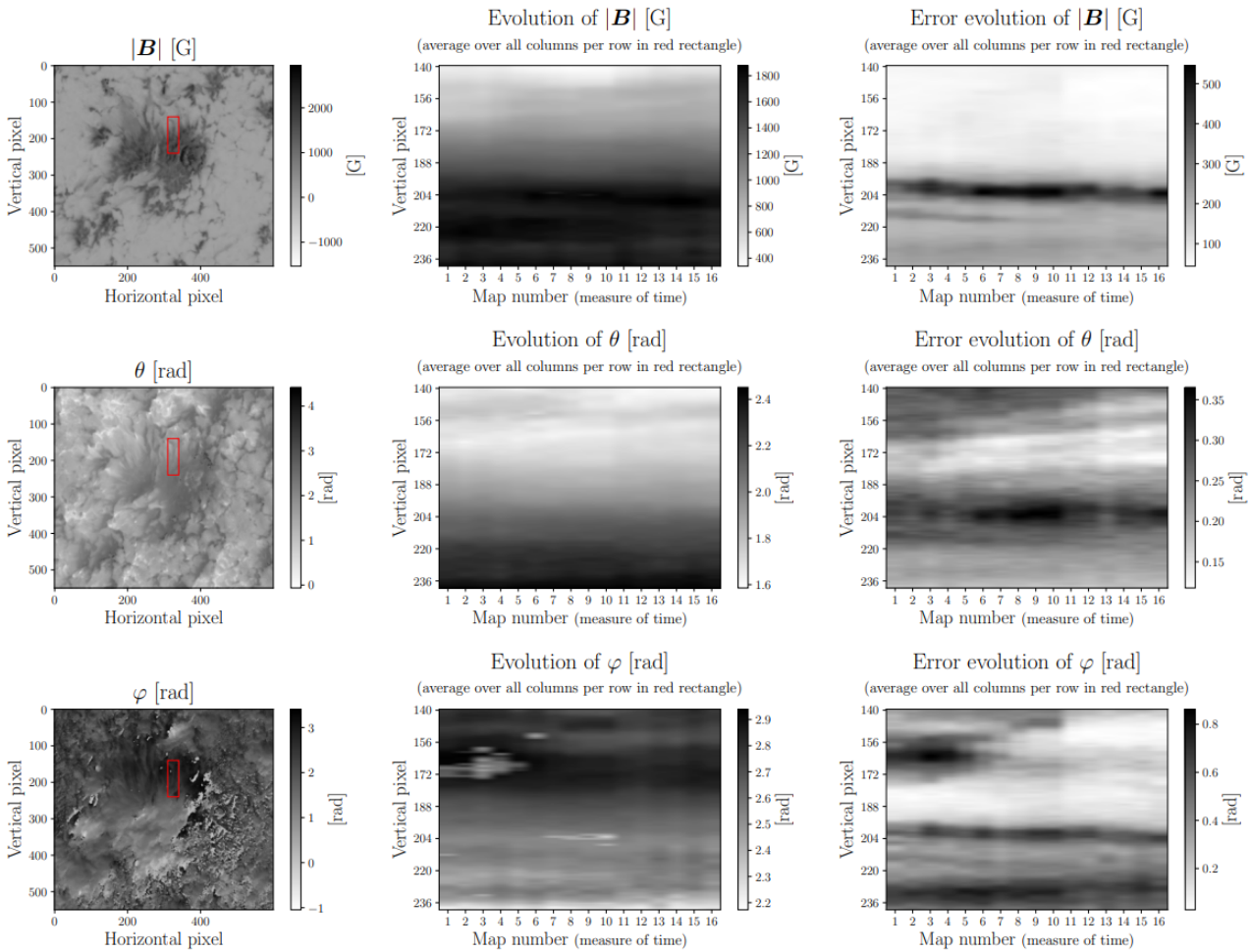


Figure 50: Results for parameter evolution experiment.

Appendix

Tracking parameter evolution during penumbra formation: Results

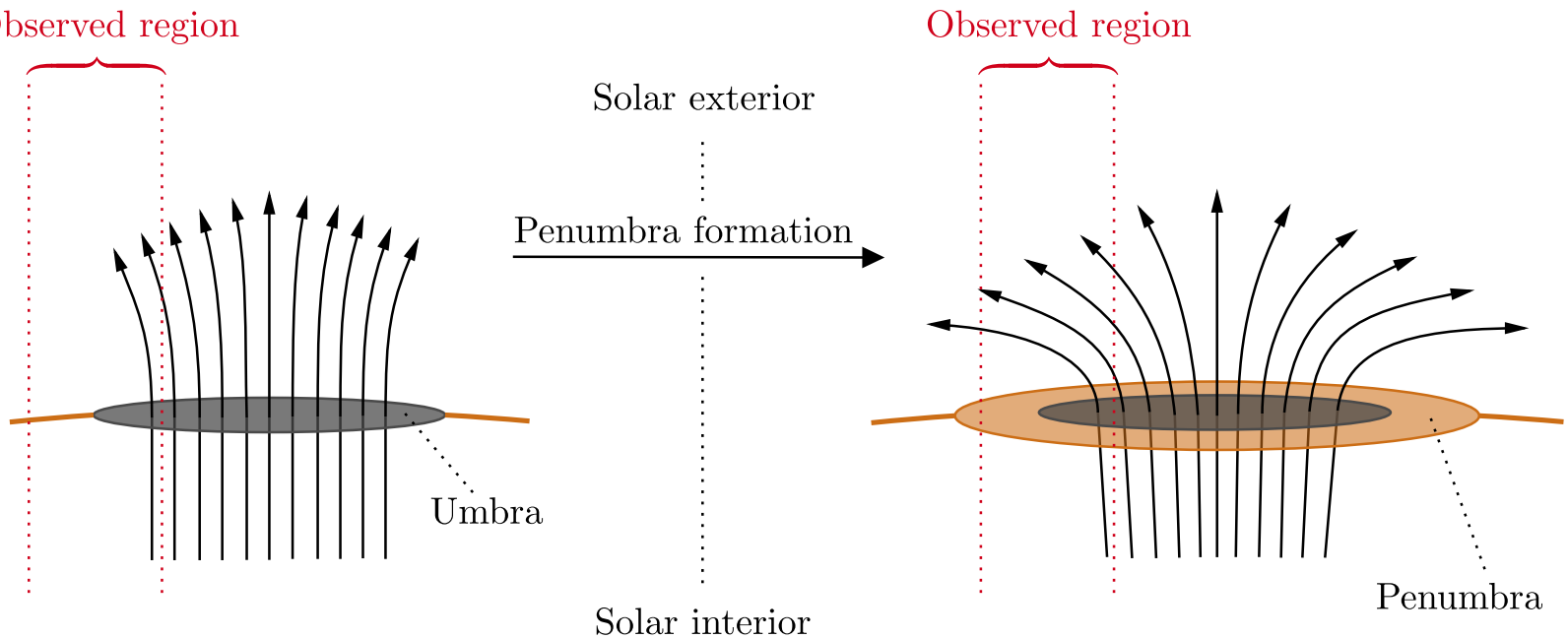


Figure 51: Schematic of what is expected to happen to the magnetic field lines during the penumbra formation process. Note, that the magnetic field lines as drawn here could also be oriented exactly opposite, i.e. facing towards the Sun's interior.

There are two key points to take away from these results:

1. Train and test loss curves show a desirable behaviour indicating, that the model has indeed learned the characteristic properties of Milne-Eddington inversions.
2. The evolution of magnetic field parameters as obtained by a normalizing flow inversion seems to match expectations.