# Understanding the Predication Mechanism of Deep Learning through Error Propagation among Parameters in Strong Lensing Case (Postprint)

**Authors:** Xilong Fan, Peizheng Wang, Jin Li and Nan Yang

**Date:** 2024-01-09T00:00:00+00:00

## Abstract

The error propagation among estimated parameters reflects the correlation among the parameters. We study the capability of machine learning of "learning" the correlation of estimated parameters. We show that machine learning can recover the relation between the uncertainties of different parameters, especially, as predicted by the error propagation formula. Gravitational lensing can be used to probe both astrophysics and cosmology. As a practical application, we show that the machine learning is able to intelligently find the error propagation among the gravitational lens parameters (effective lens mass ML and Einstein radius E) in accordance with the theoretical formula for the singular isothermal ellipse (SIE) lens model.

## Full Text

## Preamble

**Research in Astronomy and Astrophysics**, 23:125022 (11pp), 2023 December

https://doi.org/10.1088/1674-4527/ad0498

**Understanding the Prediction Mechanism of Deep Learning through Error Propagation among Parameters in Strong Lensing Case**

Xilong Fan[1], Peizheng Wang[2], Jin Li[3],[5], and Nan Yang[4],[5]

[1] Department of Physics, Chongqing University, Chongqing 401331, China; cqujinli1983@cqu.edu.cn
[4] Department of Electronic Information Science and Technology, Xingtai University, Xingtai 054001, China; cqunanyang@hotmail.com

Received 2023 August 5; accepted 2023 September 13; published 2023 November 28

## Abstract

The error propagation among estimated parameters reflects the correlation among the parameters. We study the capability of machine learning to "learn" the correlation of estimated parameters. We show that machine learning can recover the relation between the uncertainties of different parameters, especially as predicted by the error propagation formula. Gravitational lensing can be used to probe both astrophysics and cosmology. As a practical application, we demonstrate that machine learning is able to intelligently find the error propagation among the gravitational lens parameters (effective lens mass $M_L$ and Einstein radius $\theta_E$) in accordance with the theoretical formula for the singular isothermal ellipse (SIE) lens model. The relation of errors of lens mass and Einstein radius (e.g., the ratio of standard deviations) predicted by the deep convolutional neural network are consistent with the error propagation formula of the SIE lens model. As a proof-of-principle test, a toy model of linear relation with Gaussian noise is presented. We find that the predictions obtained by machine learning indeed indicate information about the law of error propagation and the distribution of noise. Error propagation plays a crucial role in identifying the physical relation among parameters rather than a coincidental relation. Therefore, we anticipate that our case study on the error propagation of machine learning predictions could extend to other physical systems for searching the correlation among parameters.

**Key words:** gravitational lensing: strong -methods: data analysis -Galaxy: fundamental parameters

## 1. Introduction

Since 1979 [**?**], gravitational lensing effects have been used as a practical approach in numerous researches of astrophysics and cosmology (e.g., see recent reviews [**?**, **?**]). As an important tool in astronomical research, gravitational lenses can generate multiple images of galaxies, quasars and supernovae, Einstein crosses and Einstein rings, which contain very important information about luminous objects [**?**, **?**, **?**, **?**, **?**, **?**, **?**]. Furthermore, gravitational lensing effects also play an important role in the study of cosmology. By using gravitational lenses, astronomers and cosmologists can determine the distribution of baryonic matter and dark matter in galaxies and clusters of galaxies more precisely, and then determine some important parameters of cosmology [**?**, **?**, **?**, **?**].

Although many lensing systems have been found through traditional searches (e.g., [**?**]), with rapidly increasing data sets, the enhancement of automated methods to discover lens candidates and estimate the relationship among parameters has become highly necessary [**?**]. Besides searching for candidates, modeling is executed by running maximum likelihood algorithms that are computationally expensive (e.g., [?, ?, ?]), and traditional parameter estimation methods are time-consuming [?]. Convolutional neural networks (CNNs), known as a class of deep learning networks, can be trained to identify characteristics of specific images. Recently, CNNs have been used to study lens modeling as a more efficient parametric method [?, ?, ?]. Furthermore, the authors of [?] have extended the work to estimate the uncertainties in parameters with neural networks [?], which was produced by using dropout techniques that evaluate the deep neural network from a Bayesian perspective [?, ?]. Some latest related works [?, ?] have demonstrated that Neural Networks can be used as a powerful tool for uncertainty inference. The main purpose of their work is to improve prediction accuracy by eliminating some unrepresentative prior deviations. Whether the uncertainties of estimated parameters by the Network reflect a correlation among the parameters plays an important role in understanding the prediction mechanism.

Being different from the above works on estimation error, in this paper we focus on the relation of errors of prediction results by machine learning. Namely, we test whether the error relation can reflect the correlation among the predicted objects through the prediction errors from deep neural networks (DNNs). We compare the estimation results of the effective lens mass (which cannot be observed directly from the lens image) and Einstein radius (which can be approximately measured from the lens image) to find the error propagation among the parameters predicted by machine learning, then to find the potential relationship between the two parameters. To our knowledge, the current work is the first one which shows that machine learning can recover the relation between the uncertainties of different parameters, especially as predicted by the error propagation formula. In fact, the correlation of the uncertainty in each parameter can be learned by general Neural Networks automatically, which will be demonstrated by means of machine learning on a linear relation toy model and strong lens data in Section 2. The detailed information about simulation and results in two models (toy model and lens model) is discussed in Section 3. Summary is drawn in Section 4 with some additional discussion.

## 2. Error Propagation Among Parameters

Traditionally, given the known physical relation of parameters or an analytic likelihood function, the relation of uncertainties of parameter estimation is presented by the error propagation formula, the Fisher matrix approach, or a Bayesian posterior distribution of multi-parameters. The error propagation formula is quite common and the most simple approach when one cannot directly measure some parameters. By using a particular relation of two parameters, the differentiation law and the Taylor expansion, one can derive the error propagation formula of two parameters on their standard deviation $\sigma$:

The effective lens mass $M_L$ (see definition in Equation (6)) in a strong lens system is an example. In traditional estimation approach, people could directly measure the $\theta_E$ and estimate the lens mass $M_L$ based on a lens model. The errors of $M_L$ (e.g., the standard deviation $s_{ML}$) are calculated through the error propagation law with the error of $\theta_E$ ($s_{\theta_E}$). While the Neural Networks do not need this known relation to get $\sigma^2(y)$ and $\sigma^2(x)$, since in the supervised-training step one can directly design any label. Then the Neural Networks could directly show the results of $\sigma^2(y)$ and $\sigma^2(x)$. Note that, when individual label relates to individual parameter, the relation of parameters is not indicated in the training process. We highlight the difference of the error propagation approach and the Neural Networks approach in Figure 1 [Figure 1: see original paper].

We will demonstrate the error propagation of Neural Networks in two cases: a linear relation toy model with Multi-layer Perceptron based networks in Section 2.1 and the lens model with convolution-based networks in Section 2.2.

Here we summarize the main results using the symbols in Table 1 . Taking the lens model into consideration, the relation from two Networks does not have clear relation, since the Network is not trained with information on the data noise distribution and we do not know the systematical error of the Network itself. So it is not trivial to check if the relation of $s_{\theta_E}$ and $s_{ML}$ derived from a Network follows the error propagation law. In fact, if the prediction error is only from the Network itself (e.g., the label value is exactly equal to the data value), the $s_{\theta_E}$-$s_{ML}$ relation does not follow the error propagation law assuming Gaussian noise. On the other hand, if the data noise is dominated, the $s_{\theta_E}$-$s_{ML}$ relation follows the error propagation law (see details in Figure 7 [Figure 7: see original paper]). This consistency is a puzzle for us, since one did not label the $\theta_E$-$M_L$ relation in the training process but only separately label the true parameters of $\theta_E$ and $M_L$.

### 2.1. Toy Model

We design a linear relation with Gaussian noise $n$ as follows:

$$X = d + n, \quad Y_i = d^i + n, \quad i = 1, 2, 3$$

where $d$ is an arbitrary value and $n$ is a Gaussian noise $n \sim N(0, \sigma)$, $X$ is an input datum with three labels $Y_i$, $\hat{Y}_i$ is the predicted value of $Y_i$, and $N_i$ is the prediction error from the neural network. For each $X$, we assign a label set $\{Y_1, Y_2, Y_3\}$, which has values $\{d, d^2, d^3\}$. By this type of labels, we not only try to check if the Network could overcome the Gaussian noise $n$ and predict the true value $\{d, d^2, d^3\}$ from $X$, but also to investigate the relation of prediction errors $s(\hat{Y}_i)$. The predicted values $\{\hat{Y}_1, \hat{Y}_2, \hat{Y}_3\}$ given by the neural network can be regarded as a function of the testing variable $X$, which is determined by an unknown prediction mechanism of the neural network. Therefore the properties of prediction errors $\{N_1, N_2, N_3\}$, e.g., their distribution, are unclear. However, we could directly compare the prediction results among them with the well-defined error propagation relations of $\{X, X^2, X^3\}$.

Through the definition of standard deviation by the error propagation law:

$$s_{\hat{Y}i} = \mathcal{F}_i(\sigma, d)$$

where $\sigma(n) = \sigma$ is the standard deviation of the Gaussian noise $n$ and $s_{\hat{Y}i}$ allows us to use the covariance properties of any variable with zero mean, $\mathrm{Cov}(n, n^2) = 0$.

At low noise limit (e.g., $\sigma \ll d$), we have:

$$s_{\hat{Y}1} \approx \sigma, \quad s_{\hat{Y}2} \approx 2d\sigma, \quad s_{\hat{Y}3} \approx 3d^2\sigma$$

It is worth checking if the noise $n$ could affect the relation of errors $s(\hat{Y}_i)$ from the networks:

$$\mathcal{M}(\hat{Y}_i, \hat{Y}_j) = \frac{s_{\hat{Y}i}}{s_{\hat{Y}j}}$$

### 2.2. The Lens Model

For the lens system $X = d + n$, the data $X$ is an observed image (e.g., see the first supernova lens image in [?]). The image could be reconstructed by a lens model $d$, while the noise $n$ is more complex than Gaussian noise. We adopt an SIE lens model, which is described by five parameters: the Einstein radius $\theta_E$, the complex ellipticity $(\epsilon_x, \epsilon_y)$ and the position of lens center $(x, y)$. For both training and testing data, those parameters are drawn from the uniform distribution shown in Table 2 with different random seeds. The network could directly predict those five parameters $\{\theta_E, \epsilon_x, \epsilon_y, x, y\}$. For the lens model, the effective lens mass $M_L$ (the mass enclosed inside the Einstein radius) is related to the Einstein radius $\theta_E$ [?]:

$$M_L = \frac{c^2 D_l D_s}{4G D_{ls}} \theta_E^2$$

where $D_l$, $D_s$ are the angular diameter distances of lens and source respectively, $D_{ls}$ is the angular diameter distance between lens and source. Traditionally, those distances are inferred by redshifts of lens and source through a cosmology model (see a recent review in [?]). Therefore, the model for the Network could also be described by parameters $\{M_L, \epsilon_x, \epsilon_y, x, y\}$, if we can measure the redshift of the source and lens by emission lines.

Unlike Einstein radius, the lens mass is a strongly model-dependent parameter and could not be observed by telescope directly. However, all parameters could be predicted from the input directly. Shown in the toy model case, deep learning is able to extract deep information from input and predict any designated source parameters. We attempt to check whether it can also learn the association among parameters indicated in the propagation of uncertainty of predictions for the lens model. Here we take $\theta_E$ and $M_L$ into consideration (Equation (6)), and the predicted errors ratio by the theoretical error propagation formula is:

$$\frac{s_{M_L}}{s_{\theta_E}} = 2\frac{c^2 D_l D_s}{4G D_{ls}} \theta_E$$

Since we only try to recover the relation of errors of two parameters ($M_L$ and $\theta_E$), here we do not infer the redshifts of lens and source, but simply assume $D_l$, $D_s$ and $D_{ls}$ are known constants as did in [?], e.g., fixed $z_s = 0.5$ and $z_l = 0.2$ (see more details in Section 3.2). The parameters $\{\epsilon_x, \epsilon_y, x, y\}$ are not fixed for the training and testing processes to make sure that all results could be used as an astrophysical application in our next work. It should also be noted that although we generate the labels of $M_L$ and $\theta_E$ for the training data sets with the SIE lens model, the prediction processes of DNN are not informed of any relationship between these parameters.

Similarly to the error relation of $\hat{Y}_1$ and $\hat{Y}_2$ in the toy model, the error relation of $\hat{M}_L$ and $\hat{\theta}_E$ is the target of this section. Therefore we could also design three label sets: $\{\theta_E, \epsilon_x, \epsilon_y, x, y\}$, $\{M_L, \epsilon_x, \epsilon_y, x, y\}$, and $\{M_L, \theta_E, \epsilon_x, \epsilon_y, x, y\}$, for one observed image $X$. For the same data $X$, three networks shown in Table 2 are adopted to predict the common parameters $\{\epsilon_x, \epsilon_y, x, y\}$, and: - Network VGG16($\theta_E$) for $\{\theta_E, \epsilon_x, \epsilon_y, x, y\}$ also gives the prediction $\hat{\theta}_E$ with prediction error $N^A_{\theta_E}$ - Network VGG16($M_L$) for $\{M_L, \epsilon_x, \epsilon_y, x, y\}$ also gives the prediction $\hat{M}_L$ with prediction error $N^B_{M_L}$ - Network VGG16($\theta_E$, $M_L$) for $\{M_L, \theta_E, \epsilon_x, \epsilon_y, x, y\}$ also gives the predictions $\hat{\theta}_E$ and $\hat{M}_L$ with prediction errors $N^C_{\theta_E}$ and $N^C_{M_L}$

Prediction errors $N^j_i$ ($i = \theta_E, M_L$; $j = A, B, C$) are caused by lens data noise and unknown network prediction mechanism, therefore we do not know the statistical properties of $N^j_{\theta_E}$ and $N^j_{M_L}$.

### 2.3. Neural Networks

**For the toy model:** We train the networks and predict $\{\hat{Y}_1, \hat{Y}_2, \hat{Y}_3\}$ with a two-layer fully connected neural network. One layer has 32 fully connected units and another has one fully connected unit for Network A and B while three fully connected units for Network C. We choose mean squared error (MSE) and Rectified Linear Unit (ReLU) as the loss function and activation function respectively, optimize the network by the ADAM algorithm, set the batch size to be 512, and adjust the learning rate to be $10^{-3}$ for 100 epochs. As shown in Table 1 , we design three types of networks for different prediction parameter sets: Network A for only $\{\hat{Y}_1\}$, Network B1 for $\{\hat{Y}_2\}$, Network B2 for $\{\hat{Y}_3\}$, and Network C for $\{\hat{Y}_1, \hat{Y}_2, \hat{Y}_3\}$.

**For the lens model:** Besides the AlexNet used in [?], we adopt the VGG16 network [?] for the lens parameters. VGG16 network is a common deep learning structure and sometimes outperforms AlexNet on computer vision tasks. We adjust the final layer to the fully connected layer to regress the parameters in VGG16. The structure of our VGG16 network is shown in Figure 2 [Figure 2: see original paper].

In the training process, we choose averaged MSE and ReLU as the loss function and activation function respectively, initialize the weights using ImageNet' s pretrained model, optimize the network by the ADAM algorithm, set the batch size to be 50, and adjust the learning rate to be $10^{-4}$ for the first $10^4$ epochs and to be $10^{-6}$ for another $10^4$ epochs. The training process lasts several hours for AlexNet and twenty hours for VGG16 with GPU RTX 2080 Ti single card.

## 3.1. The Toy Model

We test two cases: one is the noise case ($n \neq 0$ with different noise levels) used to test the consistency between Equations (3) and (5), and the no-noise ($n = 0$) case to test Equations (4) and (5).

For both cases, $d$ is drawn from a uniform distribution $U[-2, 2]$. For the noise case, Gaussian noise $n \sim N(0, \sigma)$ is adopted to generate a sample $X = d + n$ with three labels $Y_i$, which have values $\{Y_1 = d, Y_2 = d^2, Y_3 = d^3\}$. To avoid over-fitting, 200,000 $X$ with $\sigma = 0.05$ and 20,000 $X$ with $\sigma = 0.5$ are adopted for training and testing processes, respectively. It is worth noting that the feature distribution of the training set and the test set is not independent and identically distributed. The noise level of the training set is smaller than that of the test set. This is because we want to ensure that the model can learn accurate mappings. The function $\mathcal{M}$ is calculated from predictions $\{\hat{Y}_i\}$, which are evaluated in 20 bins of $d$.

Figure 3 [Figure 3: see original paper] shows the prediction results of the toy model for the no-noise case (in upper panel) and comparison of the function $\mathcal{M}$ (red points, Equation (5)), $\mathcal{F}$ (blue lines, Equation (3)) and $\mathcal{F}$ (green lines, Equation (4)) for both no-noise (in middle panel) and noise cases (in lower panel). For the no noise ($n = 0$) case, according to Equation (2) $X = Y$, model prediction errors $\{N_1, N_2, N_3\}$ are only determined by the network itself, and the predictions of the Network almost do not have errors ($\{N_1, N_2, N_3\} \sim 0$ shown in the upper panel of Figure 3). The propagation of errors of $\hat{Y}$ caused by unknown network prediction mechanism (the function $\mathcal{M}$ in the middle panel of Figure 3) does not follow the law of error propagation, e.g., $\mathcal{M} \neq \mathcal{F}$. On the other hand, for the noise $n \neq 0$ case shown in lower panel of Figure 3, the propagation of errors of $\hat{Y}$ follows the law of error propagation quite well, e.g., $\mathcal{M} \approx \mathcal{F}$ for all noise levels. Note that, since we fixed the noise level $\sigma$, larger $|d|$ represents smaller noise case (Equation (4)). Those trends do not depend on the prediction parameter set, since Network A, B and C return almost identical results.

The results of noise case $\mathcal{M} \sim \mathcal{F}$ indicate that the network "knows" the distribution of noise $n$ and the relation of label numbers $\{Y_1, Y_2, Y_3\}$, although there is no information on $n$ in the labels and the relation itself is also not included in the labels. Although we do not know the prediction mechanism of the neural network for parameter prediction but only the structure of the networks, it seems that the neural network could guarantee the association among parameters. This performance of the neural network is more interesting when it is applying for more sophisticated physics models, e.g., the gravitational lens model.

## 3.2. The SIE Lens Model

Following [?], we consider the singular isothermal ellipse (SIE) lens model (Equation (6)) and fix the redshift of lens $z_l = 0.5$, the redshift of source $z_s = 2$ and adopt the Planck 2016 cosmology model (i.e., $h = 0.7$, $\Omega_m = 0.3$, $\Omega_\Lambda = 0.7$) [?]. The values of Einstein' s radius $\theta_E$, the complex ellipticity $(\epsilon_x, \epsilon_y)$ and the position of lens center $(x, y)$ for both training and testing data are drawn from the uniform distribution shown in Table 2 . We simulated the lensed images for training and testing based on source images from COSMOS-23.5, COSMOS-25.2 in GREAT3 data. To test the generalization of networks trained by the images with high quality from GREAT3, we also use data from Galaxy Zoo as source images to produce another test data set. With the VGG16 and AlexNet trained by GREAT3 data (two million samples in total), we estimate the parameters $\{\theta_E, \epsilon_x, \epsilon_y, x, y\}$ of other branches of GREAT3 data (ten thousand samples in total), labeled such as VGG16($\theta_E$) and AlexNet($\theta_E$) in Table 2 , respectively. More details on data, training, testing, robustness, accuracy of estimation and error propagation can be found in the following content.

The source images for training data are from COSMOS-23.5 and COSMOS-25.2. All source images are first convolved by the point-spread function (PSF) supported in GREAT3 data to improve image quality. These images are used to produce two million lensed images with parameters shown in Table 2 . Each lensed image undergoes the following operations before being fed into the network to avoid over-fitting. First, add random Gaussian noise to the lensed image. The root mean square value of the noise is randomly selected from a uniform distribution, and its value is 1%–10% of the signal. Then, we use a factor of 501,000 to convert the image to photon counts, and use these values as the $\lambda$ to generate a Poisson realization map, effectively adding Poisson noise to the image. We use the 400,000 images including simulated hot pixels and cosmic rays provided by [?] to make the network insensitive to pixel artifacts and cosmic rays. Then we use a random root mean square Gaussian filter to convolve the image to simulate the blurring effect of the PSF that reveals the factors of atmosphere and the telescope itself. Finally, randomly translate the image for augmenting the data. The total training data samples are two million in total, of which ten thousand data sets are used for validation. Each training data sample is fed into the neural network with different data augmentation operations.

We use another branch of GREAT3 data (1.8 million samples) as source images to produce our test data set (ten thousand samples in total). To test the generalization of networks trained by the high-quality images from GREAT3, we use data from Galaxy Zoo (61 thousand samples) as source images to produce another test data set (ten thousand samples in total). The data in Galaxy Zoo are coming from the Sloan Digital Sky Survey (SDSS). There may be multiple galaxies and a higher noise level compared to the GREAT3 data. One training image from GREAT3 and one test image from Galaxy Zoo are illustrated in Figures 4(a) and (b) [Figure 4: see original paper], respectively.

To understand the robustness of networks, we use the VGG16($M_L$) and AlexNet($M_L$), which are trained by the GREAT3 data sets with higher image quality, to predict lens mass $M_L$ of test data sets in the Galaxy Zoo data group. As shown in Table 2 , the standard deviations of the parameters estimated by VGG16 are all slightly better than AlexNet. It should be noted that in [?] the errors of their AlexNet networks seem to be much better than ours. The reason is that the test data sets in [?] are unknown to us; we get corresponding results only considering the network (AlexNet) with their trained weights. The standard deviations of $\{\epsilon_x, \epsilon_y, x, y\}$ from VGG16($M_L$) are comparable to the results from VGG16($\theta_E$).

To check if the prediction also depends on the parameter value, we compare the estimated lens masses $\hat{M}_L$ by VGG16($M_L$) and Einstein' s radius $\hat{\theta}_E$ by VGG16($\theta_E$) with their true values with box plots for the VGG16 trained by GREAT3 data (Figure 5 [Figure 5: see original paper]). We divide the interval into 10 segments with equal width, and draw a box plot for each segment. For the box plot of Einstein' s radius, every bin has approximately the same data. But for the box plot of the lens mass, there are more data in the bins with small mass, because the mass is proportional to the square of Einstein' s radius. As shown in Figure 5 [Figure 5: see original paper], the mean value of predicted mass $\hat{M}_L$ by VGG16($M_L$) recovers the true value $M_L$ better, although there are more outliers than the Einstein' s radius $\hat{\theta}_E$ by VGG16($\theta_E$). For massive galaxies, more outliers are in smaller prediction values compared with the true value, while for less massive galaxies more outliers are in larger predicted values.

The results from the Galaxy Zoo data are shown in Figure 6 [Figure 6: see original paper]. The value of $M_L$ and the residuals of lens mass predicted by VGG16($M_L$) are shown in Figures 6(a) and (b) [Figure 6: see original paper], respectively. Although the average value of predicted parameters represents the true value quite well, there are more outliers resulting in larger standard deviations for both networks (see Table 2 ). This is partly because the data sets in the Galaxy Zoo data group are sources with irregular shapes and have very noisy background as the image shown in Figure 4(b) [Figure 4: see original paper]. It can be seen that the reason for clustering to the average value is that the noise of the image is too large to contain useful information (see the image data of outliers in Figures 6(d) and (f) [Figure 6: see original paper] corresponding to the outliers in Figures 6(c) and (e) [Figure 6: see original paper]). In order to minimize the overall loss, neural networks tend to output the average value of the sample.

The detailed comparison of estimated lens masses with their true values is shown in Figure 6 [Figure 6: see original paper], which indicates that the predicted value of small mass tends to be greater than the real value, while the predicted value of large mass is less than the real value (also see the residuals plot in

Figure 6(b) [Figure 6: see original paper]). This tendency is also found for $\{\epsilon_x, \epsilon_y, x, y\}$.

The standard deviation of all parameters predicted by VGG16 or AlexNet in three sets of labels for all test samples are shown in Table 2 . In order to investigate the relation among the prediction errors of $\theta_E$ and $M_L$, the ratio of the standard deviation of $\widehat{M}_L$ and $\widehat{\theta}_E$ by VGG16 networks A, B, C (test data set in 15 segments) as a function of the center value of $M_L$ is shown in Figure 7 [Figure 7: see original paper]. It can be found that the error propagation by VGG16($\theta_E$, $M_L$) is roughly consistent with the error propagation formula (the yellow line represents Equation (7) $\mathscr{F}$ in Figure 7 [Figure 7: see original paper]): $\mathscr{M} \sim \mathscr{F}$ as shown in the toy model case. Again, network C seems to "know" the noise distribution since the errors follow the theoretical error propagation formula. However, the ratio of the errors of $\widehat{\theta}_E$ and $\widehat{M}_L$ derived from networks A and B does not follow the theoretical error propagation formula, which is not the case in the toy model. The plausible reason for the difference between the toy model and lens model is that Equation (7) does not consider the non-Gaussian noise effects of input data.

This result enlightens us that as long as the accuracy of parameter estimation by the network is guaranteed, even if we do not know the physical relation between the parameters, the relation will be reflected through the input data, corresponding error of deep learning estimation. This feature of deep learning is valuable for further investigation on the parameter correlation with unknown theoretical model in advance. For example, if the lens mass is measured by gravitational wave [?], one could combine the lens mass estimation from gravitational wave, Einstein radius estimation from optical lens image and the redshifts $z$ from emission lines to investigate $M_L$-$\theta_E$-$z$ relation.

## 4. Discussion and Summary

Unlike traditional parameter estimation, parameter estimation by machine learning almost completely depends on the information in samples. Assuming the SIE lens model, the Einstein radius $\theta_E$ and effective lens mass $M_L$ are estimated by the convolutional neural network, and the capability of the network to acquire correlation information between parameters from the data is tested through the estimation errors. In this process, the Networks produce the relation of errors as the traditional error propagation law based on known $\theta_E$-$M_L$ relation. Such a correlation of estimated parameters provides a self-consistent result, so it is very important for further study on parameter estimation by machine learning.

In order to ensure the reliability of the above results, the accuracy of parameter estimation by the network also needs to be guaranteed. The convolutional neural network AlexNet is an effective approach for predicting parameters of the lens model [?]. Through applying the typical convolutional neural network VGG on the parameter estimation of the gravitational lens, the great performance on abstraction of features has been shown in our simulated lens data (results are shown in Table 2 and Figures 5 [Figure 5: see original paper] and 6 [Figure 6: see original paper]). Meanwhile, the robustness of such a network could also be guaranteed to a certain extent. From the results of Galaxy Zoo test data sets, it is found that for signals submerged in noise, neural networks tend to output the average of the training set to minimize the MSE. We also test the non-normal loss (MAE) as the loss function after the normal generative process, and the performance of the error propagation is similar as shown in the MSE case. Further study could also test more advanced networks on the performance of parameter estimation, such as ResNet [?], DenseNet [?], ViT [?], so as to get a more accurate error propagation correlation among parameters.

Although the MSE and MAE loss functions seem to only guarantee the accuracy of each parameter, they ensure that the model fits the functional relationship between input data and output data. According to the universal approximation theorem [?], the functional relationship should be presented by the perfect network structure and weights of neurons. The fact that the error of each parameter' s estimation by machine learning satisfies the error propagation formula is worth discussing. The general plausible reason for consistency is that a mapping exists between the predicted parameters and the input data. If there is noise in the input, the model will output biased predictions containing noise according to the accurate mapping. Therefore, different parameters will exhibit the law of error propagation due to the same input noise. In particular, for the toy model, there is a simple functional relationship between input and output data, i.e., $Y_i = X^i$. In training, the model tends to minimize the loss function in order to learn the functional relationship. After training, the functional relationship is stored in the weight of each neuron and the entire model has the functional relationship contained in the training data. When we add noise directly to $X$, since the new model' s mapping is almost the same as the mapping contained in the training data, the predicted results exhibit the error propagation relationship. In the lens model, it can be considered that the CNN layer performs feature extraction on the image. The output results of the last CNN layer are the latent variables representing the image. The fully connected layer is a function from latent variables to predicted results. The error of the image will cause the error of the latent variables, and the predicted values are functions of the latent variables. So the error propagation formula is satisfied between the predicted values and the latent variables, and the error propagation formula is therefore satisfied between the predicted values.

## Acknowledgments

We thank Kai Liao for helpful discussions. This work was supported by the National Natural Science Foundation of China (grant No. 11922303), the Natural Science Foundation of Chongqing (grant No. CSTB2023NSCQ-MSX0103), the Key Research Program of Xingtai 2020ZC005, and the Fundamental Research Funds for the Central Universities (grant No. 2042022kf1182).

## References

Atek, H., Richard, J., Kneib, J.-P., & Schaerer, D. 2018, MNRAS, 479, 5184
Blandford, R. D., & Narayan, R. 1992, Annual Review of Astronomy and Astrophysics, 30, 311
Bradač, M., Treu, T., Applegate, D., et al. 2009, ApJ, 706, 1201
Collett, T. E. 2015, ApJ, 811, 20
Diego, J. M., Protopapas, P., Sandvik, H. B., & Tegmark, M. 2005, MNRAS, 360, 477
Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2021, in Int. Conf. on Learning Representations
Frieman, J. A., Harari, D. D., & Surpi, G. C. 1994, PhRvD, 50, 4895
Gal, Y., & Ghahramani, Z. 2015, arXiv:1506.02142
He, K., Zhang, X., Ren, S., & Sun, J. 2015, arXiv:1512.03385
Helbig, P., & Kayser, R. 1996, A&A, 308, 359
Hezaveh, Y. D., Dalal, N., Marrone, D. P., et al. 2016, ApJ, 823, 37
Hezaveh, Y. D., Perreault Levasseur, L., & Marshall, P. J. 2017, Natur, 548, 555
Hornik, K., Stinchcombe, M., & White, H. 1989, NN, 2, 359
Hortúa, H. J., Volpi, R., Marinelli, D., & Malagò, L. 2020, PhRvD, 102,
Jullo, E., Natarajan, P., Kneib, J. P., et al. 2010, Sci, 329, 924
Kelly, P. L., Diego, J. M., Rodney, S., et al. 2018, NatAs, 2, 334
Kelly, P. L., Rodney, S. A., Treu, T., et al. 2015, Sci, 347, 1123
Kneib, J.-P., & Natarajan, P. 2011, A&ARv, 19, 47
Lefor, A. T., Futamase, T., & Akhlaghi, M. 2013, NewAR, 57, 1
Liao, K., Biesiada, M., & Zhu, Z.-H. 2022, ChPhL, 39, 119801
Metcalf, R. B., & Petkova, M. 2014, MNRAS, 445, 1942
Morningstar, W. R., Hezaveh, Y. D., Perreault Levasseur, L., et al. 2018, arXiv:1808.00011
Oguri, M. 2019, RPPh, 82, 126901
Oguri, M., & Marshall, P. J. 2010, MNRAS, 405, 2579
Park, J. W., Wagner-Carena, S., Birrer, S., et al. 2021, ApJ, 910, 39
Perreault Levasseur, L., Hezaveh, Y. D., & Wechsler, R. H. 2017, ApJL, 850, L7
Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2016, A&A, 594, A19
Schneider, P., Ehlers, J., & Falco, E. E. 1992, Gravitational Lenses (Berlin: Springer)
Schuldt, S., Suyu, S. H., Meinhardt, T., et al. 2021, A&A, 646, A126
Simonyan, K., & Zisserman, A. 2014, arXiv:1409.1556
Spiniello, C., Agnello, A., Napolitano, N. R., et al. 2018, MNRAS, 480, 1163
Hou, S., Fan, X.-L., Liao, K., & Zhu, Z.-H. 2020, PhRvD, 101, 064011
Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. 2016, arXiv:1608.06993
Vegetti, S., Lagattuta, D. J., McKean, J. P., et al. 2012, Natur, 481, 341
Wagner-Carena, S., Park, J. W., Birrer, S., et al. 2021, ApJ, 909, 187
Walsh, D., Carswell, R. F., & Weymann, R. J. 1979, Natur, 279, 381