Time delay estimation in unresolved lensed quasars

L. Biggio,[a] A. Domi,[b] S. Tosi,[c] G. Vernardos,[d] D. Ricci,[e] L. Paganin,[c] G. Bracco[c]
[a] Eidgenössische Technische Hochschule Zürich, Rämistrasse 101, CH-8092 Zürich, Switzerland
[b] University of Amsterdam, Science Park 105, 1098 XG Amsterdam, the Netherlands
[c] Università degli Studi di Genova and Istituto Nazionale di Fisica Nucleare (INFN)
- Sezione di Genova, via Dodecaneso 33, 16146, Genoa, Italy
[d] Institute of Physics, Laboratory of Astrophysics, Ecole Polytechnique Fédérale de Lausanne (EPFL),
Observatoire de Sauverny, 1290 Versoix, Switzerland
[e] Istituto Nazionale di Astrofisica (INAF), Osservatorio di Padova - Vicolo dell'Osservatorio, 5, 35122 Padova, Italy

4 February 2022

ABSTRACT

Time-delay cosmography can be used to infer the Hubble parameter H_0 by measuring the relative time delays between multiple images of gravitationally-lensed quasars. A few of such systems have already been used to measure H_0: their time delays were determined from the multiple images light curves obtained by regular, years long, monitoring campaigns. Such campaigns can hardly be performed by any telescope: many facilities are often over-subscribed with a large amount of observational requests to fulfill. While the ideal systems for time-delay measurements are lensed quasars whose images are well resolved by the instruments, several lensed quasars have a small angular separation between the multiple images, and would appear as a single, unresolved, image to a large number of telescopes featuring poor angular resolutions or located in not privileged geographical locations. Methods allowing to infer the time delay also from unresolved light curves would boost the potential of such telescopes and greatly increase the available statistics for H_0 measurements. This work presents a study of unresolved lensed quasar systems to estimate the time delay using a deep learning-based approach that exploits the capabilities of one-dimensional convolutional neural networks. Experiments on state-of-the-art simulations of unresolved light curves show the potential of the proposed method and pave the way for future applications in time-delay cosmography.

Key words: gravitational lensing: strong – software: data analysis – methods: statistical – distance scale

1 INTRODUCTION

The Hubble parameter H_0, quantifying the current expansion rate of the universe, is a major component of cosmological models, which can then be tested by its determination. To date, measurements of H_0 from different observations have led to a tension on its estimated value. In particular, early universe observations of the CMB anisotropies (e.g., from the Planck satellite, Aghanim et al. 2020) have measured $H_0 = 67.4 \pm 0.5$ km s^{-1} Mpc^{-1}, whereas late universe probes such as the distance ladder (Riess et al. 2019) give $H_0 = 74.03 \pm 1.42$ km s^{-1} Mpc^{-1}, resulting in a tension of about 4.4σ (Di Valentino et al. 2021; Beenakker & Venhoek 2021; Verde et al. 2019). As firstly pointed out by Refsdal (1964), an additional method to determine H_0 is time-delay cosmography, which exploits that fact that the time delay (ΔT) between multiple images of gravitationally lensed quasars (GLQs) is directly related to the Hubble parameter. The most relevant results obtained via time-delay cosmography come from the H0LiCOW collaboration (Wong et al. 2019), who has found $H_0 = 73.3^{+1.7}_{-1.8}$ km s^{-1} Mpc^{-1} from a sample of six GLQs monitored by the COSMOGRAIL project (Millon et al. 2020). This result, combined with the other late universe observations (Riess et al. 2019), enhances the H_0 tension up to 5.3σ. However, a more recent analysis of 40 strong gravitational lenses, from TDCOSMO+SLACS (Birrer et al. 2020), has found $H_0 = 67.4^{+4.1}_{-3.2}$ km s^{-1} Mpc^{-1}, relaxing the tension and demonstrating the importance of a better understanding of the mass density profiles of the lenses. In this context, further studies including more objects are needed for a more precise estimation of the H_0 parameter (Birrer & Treu 2020). The fractional error on H_0 for an ensemble of N GLQs is related to the uncertainties in the time-delay measurement, $\sigma_{\Delta T}$, line-of-sight convergence, $\sigma_{\rm los}$, and lens surface density, $\sigma_{[k]}$, as (Tie & Kochanek 2017):

$$\frac{\sigma_H^2}{H_0^2} \sim \frac{\sigma_{\Delta T}^2/\Delta T^2 + \sigma_{\rm los}^2 + \sigma_{[k]}^2}{N}, \quad (1)$$

where the first two terms are dominated by random uncertainties and their contributions scale as $N^{-1/2}$. There are therefore two ways of reducing the uncertainty on H_0: 1) reduce the contribution of random uncertainties, 2) increase the size N of the analysed GLQ sample. The main contribution on random uncertainties is given by the microlensing effect (Tie & Kochanek 2017): massive objects such as giant stars, black holes, etc, present in the lensing system,

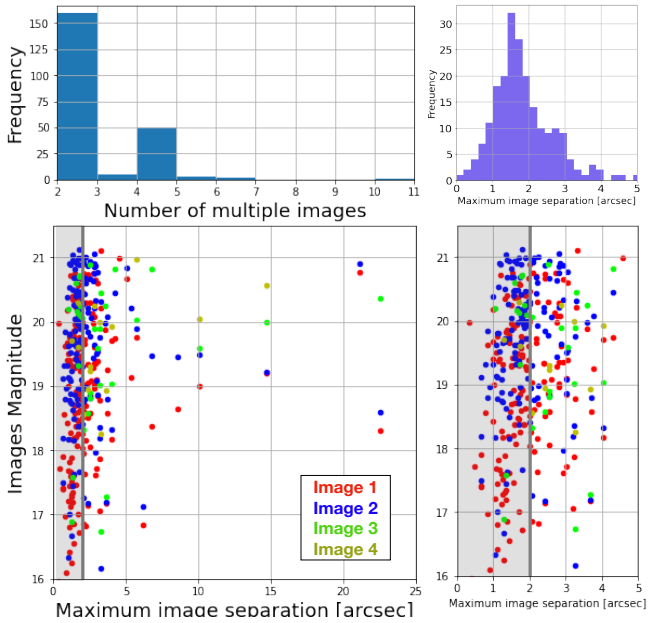

Figure 1. Number of images and image separation for the population of known gravitationally-lensed quasars. Top-left: distribution of known GLQs as a function of the number of multiple images. Top-right: distribution of known GLQs as a function of the maximum image separation. Bottom: Magnitude of the multiple images versus the maximum image separation for lensed systems with up to 4 multiple images (left); a zoom in the region 0-5 arcsec is shown on the right. The different colors of the dots identify each of the multiple images, from 1 to 4. The grey region contains 70% of the total GLQ sample.

can partially absorb, deflect or magnificate the light coming from the source. This results in changes in the light curves which can mistakenly be exploited to estimate Δt. With respect to the size N, to date, a sample of about 220 GLQs is available[1], however, only a very small subset with well separated multiple images has been used to measure H_0. Indeed, larger-separation systems benefit of better resolved space-based data, which in turn allow for better constraints on the mass models; moreover, it is easier to monitor brighter systems and to obtain their time delays. Therefore, it is easier and safer to extract information from such systems, and consequently reduce the uncertainty on H_0. Figure 1 (bottom) shows the magnitude of the multiple images versus the maximum image separation for the known GLQs. Systems falling in the grey region, which represents 70% of the total sample, have a maximum image separation below 2 arcsec. The image separation peaks indeed at around 1 arcsec (Oguri & Marshall 2010; Collett 2015), making the smaller and harder to observe systems the most numerous GLQs in future surveys.

The ideal instruments to perform lensed quasar monitoring have high sensitivity, an optimal geographical location (where the effects of atmospheric turbulence are less prominent), and a high angular resolution optimized with the usage of state-of-the-art adaptive optics systems. However, because of the time scales of the intrinsic variations of the sources, which can be of the order of years, such observation campaigns should last several observing seasons (Millon et al. 2020). Consequently, due to the amount of observational requests that the best performing telescopes have to fulfill, they can hardly be employed for such monitoring purposes. On the other hand, small/medium sized telescopes ($\approx$ 1-2m or smaller) can often guarantee a better availability of observational time for this purpose (Borgeest et al. 1996). Unfortunately, their already reduced sensitivity can be further worsened by their often less privileged geographical positions, in terms of clear nights and atmospheric seeing, which can reach up to 3 arcsec (Karttunen et al. 2017). While a few lensed quasars can be fully resolved by such facilities, and indeed time delay curves have been provided by them (for example the 1.2 m Euler Swiss telescope at ESO La Silla), still the majority of the already known GLQs, together with future discoveries, will mainly appear as a single image for these telescopes.

The identification of more strongly lensed quasars from unresolved light curves can clearly boost the outcome of upcoming surveys but it represents a challenging problem because of the limited angular resolution of wide surveys: proposals have been made to identify lensed systems even from not fully resolved light curves (see for example (Shu et al. 2021)). Light curves from resolved sources are then analysed using point estimators to derive time delays (Tewes et al. 2012); a recent proposal was advanced to deal with the unresolved cases, based on minimizing fluctuations in the reconstructed light curves (Bag et al. 2021).

This work proposes a novel approach to estimate the time delay in unresolved GLQ light curves based on deep learning algorithms. Deep Learning (DL) is an emerging field of Machine Learning that has reached state-of-the-art performance in several applications. Cosmology and astrophysics will also benefit from the application of deep learning techniques, in particular in light of the need for more efficient data analysis tools and the unprecedented amount of information expected from the launch of several upcoming surveys, such as the Vera Rubin Observatory (Abell et al. 2009).

For simplicity, the usage of DL on GLQ light curves in this work only focuses on unresolved image pairs, which can come either from doubly- or quadruply-lensed GLQs. This choice is further motivated by the fact that about 85 per cent of the already known systems are doubles (Oguri & Marshall 2010; Collett 2015), as shown in Fig. 1 (top-left).

The paper is structured as follows: Sec. 2 describes the deep learning-based method used for evaluating the time delay between unresolved multiple images, Sec. 3 describes generating the simulated light curves needed for training the DL algorithm, and Sec. 4 shows the results of the proposed method on a test dataset.

2 TIME DELAY ESTIMATION WITH DEEP LEARNING

The method we adopt here exploits the ability of modern Convolutional Neural Networks (CNNs) (Bengio & Lecun 1997) to extract informative features directly from raw data; they work in an end-to-end fashion given a supervised-learning task of interest that utilizes a pre-trained model. In our case, this task is the estimation of the time-delay between two unresolved quasar light curves.

The approach is motivated by the surprisingly good performance of Machine Learning and, in particular, Deep Learn-

[1] https://research.ast.cam.ac.uk/lensedquasars/index.html

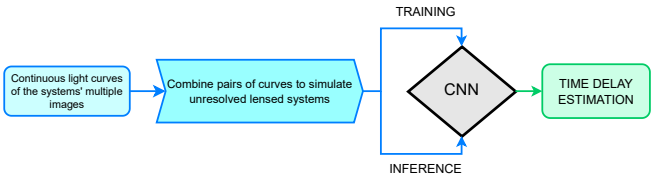

Figure 2. Overview of the proposed methodology.

ing methods in a wide range of engineering fields, including astrophysics and cosmology: in addition to automated tasks on the large datasets of wide survey experiments (see for example (Cabrera-Vives et al. 2016; Kimura et al. 2017; George et al. 2017; Schawinski et al. 2017; Sedaghat & Mahabal 2018; Shallue & Vanderburg 2018)), deep learning is also proposed to analyse time series (for example Reimers & Requena-Mesa (2020); Wei & Huerta (2021)).

Most of these techniques are based on the supervised learning paradigm, i.e. when labelled data are available and the algorithm can rely on explicit supervision signals. In the case of most deep learning algorithms, the extent of such supervision is often significant, meaning that large labelled datasets are needed for effective learning. This scenario often results in excellent performances when labelled data are abundant and their collection is easy and not expensive. However, these conditions are not always satisfied and, in absence of aforementioned labelled datasets, one must resort to either unsupervised or self-supervised learning strategies, for which only unlabelled data are used, or to synthetic data generation in order to produce the desired labelled datasets in such a way that the artificial data resemble the real ones as much as possible.

Our work follows the latter approach: the training data are manually constructed via a physics-based lens simulator described in Section 3. The benefit of such an approach is that, depending on the available computational resources, arbitrarily large datasets can be generated. On the other hand, the obvious downside is that the performance of the model at inference time, i.e. when tested on real data, will be strongly dependent on the degree of fidelity of the artificial data with respect to the real ones. This problem is also known as *sim2real gap* (Heiden et al. 2020; Zhao et al. 2020) and it is a very important aspect in many disciplines, including robotics and computer vision. In this paper, we show that, given an as accurate as possible simulator (in the sense specified above), a fully data-driven CNN is able to retrieve the time delay from a *single* time series representing the overlap between two unresolved light curves. The approach is modular in the sense that, if a more precise simulator is available, it can simply be replaced and used to generate new data to retrain our CNN models with it.

Furthermore, available models in the literature have yielded state-of-the-art results in the context of time-series classification and regression. Here, we use such approaches with very little variations compared to their original form; significant changes in the architectures will not be needed in order to obtain the desired results, even in presence of data generated from a different simulator. In the following, we motivate our choice of CNNs as our data-driven models and we introduce the basic principles behind their architecture. Then, we provide detail on the design choices and our training procedure.

2.1 Convolutional Neural Networks

CNNs (Bengio & Lecun 1997) have been initially proposed in the context of Computer Vision applications, such as image classification (Krizhevsky et al. 2012) and segmentation (Sultana et al. 2020). They differ from standard fully-connected neural networks (where each node in a certain layer is linked to any other node in the subsequent one) since they implement a convolution operation conferring them two biologically-inspired properties, namely weight sharing and local connectivity. The first results in the same weights being applied repeatedly to different areas of the input data, whereas the second imposes that the action of such weights is realized only locally, on small regions of the input space. Modern CNNs consist of multiple stacked *layers* implementing the aforementioned operation in a hierarchical fashion.

Besides Computer Vision, CNNs have been also fruitfully applied to time series regression and classification (He et al. 2015; Fawaz et al. 2020). The main difference compared to the standard CNNs applied to Computer Vision problems is that, in the case of time series, the filters used by the neural network are now one-dimensional. The choice of such networks for time series analyses is motivated by the structural assumptions (or inductive biases) at the basis of the design of CNN models. Indeed, deep CNNs implement a series of convolutions at each level of the hierarchy along their depth. They work by extracting *local* features from the input raw data, whose representation assumes increasingly higher levels of complexity as we move along the deeper layers of the network. Our basic assumption is, therefore, that eventual traces of the magnitude of the time delay between two curves manifest themselves at a local level, motivating the choice of CNN as feature extractor. The application of CNNs to the problem of time-delay estimation is described in the following paragraph.

2.2 Time-delay estimation with CNNs

The input of the CNN models consists of a time series representing an unresolved quasar light curve $\mathbf{x} = \{x_t\}_{t=1}^{T}$ where T is the length of the sequence. The output of the model is a real number $\hat{y} \in \mathbb{R}$ representing the time delay between the two superimposed curves that went into creating the input time series.

Models are trained by a generated dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where N is the total number of training examples and y_i is the ground-truth time delay associated with the i-th training instance. This initial dataset is split into three parts, namely a training dataset, $\mathcal{D}_{tr}$, a validation dataset, $\mathcal{D}_{val}$, and a testing dataset, $\mathcal{D}_{ts}$. The first is used to train the weights of our model, the second to check the generalization performance during training, and the last one to evaluate the model once the training phase is terminated. Such a splitting is necessary to monitor the occurrence of the so-called overfitting phenomenon, i.e. when the neural network simply memorizes the training set and does not generalize outside the training distribution. Figure 2 summarises our methodology. The artificial data produced as will be detailed later are used to train

the CNN model. Inference is then performed on the original non-resolved light curves using the trained model.

The mean-squared error (MSE) is used as a loss function, i.e. to measure the error the network is making in predicting $\hat{y}$ instead of y:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2. \quad (2)$$

During training, the weights of the network are varied so that the value of this loss is minimized. This process is realized by the *back-propagation algorithm*, which allows for the efficient calculation of the gradients of the loss function with respect to the weights in the network. The optimization algorithm used for minimizing the loss is called stochastic gradient descent, and the popular Adam (Kingma & Ba 2015) variant of this algorithm is used here: we use it with a learning rate of 10^{-3}; a batch size of 50 is selected. We also periodically evaluate the network on the validation set during training and check its performance in terms of MSE. As commonly done in practice, whenever the validation loss decreases, the weights of the network at that step of training are saved.

2.3 Models

Three different CNN models are tested to perform the time-delay estimation task. The first two, namely ResNet (He et al. 2015) and InceptionTime (Fawaz et al. 2020), are the results of recent research efforts in the area of time series classification and are adapted to our scope with minimal changes compared to their original implementation. The main innovation introduced by these two models stands in their use of the residual connections, which facilitate the propagation of gradients even in relatively deep networks, without incurring in the so-called vanishing/exploding gradient problems, by introducing elements like bottleneck layers and allowing for multiple filters of different lengths to be applied simultaneously to the same input time series.

Inception modules have a slightly more complex structure compared to Resnet modules which are mainly made of simple fully convolutional layers (McNeely-White et al. 2020). For more details on the specifics of the architectures of these two models we refer the interested reader to the works of He et al. (2015) and Fawaz et al. (2020). We adapt the original implementations of these models to the regression setting by changing the dimension of the output layer to one. To the best of the authors' knowledge, this is the first time ResNet and InceptionTime have been applied to a problem involving time-series in the context of cosmology and we hope our work can be inspirational for future applications of such models.

The third model, simply labelled generically "CNN" in the following, is a relatively standard deep fully convolutional network without residual connections and it is fine-tuned to maximize the performance on the validation set: in this way we can compare the results obtained with a fine-tuned network with those from not-fine-tuned ones. This CNN consists of 13 convolutional layers, each with 14 filters of size 17 and 3 linear layers mapping the output of the convolutional blocks into the final output. Batch-Normalization (Ioffe & Szegedy 2015) and ReLU (Agarap 2019) activations functions are used after each convolutional layer.

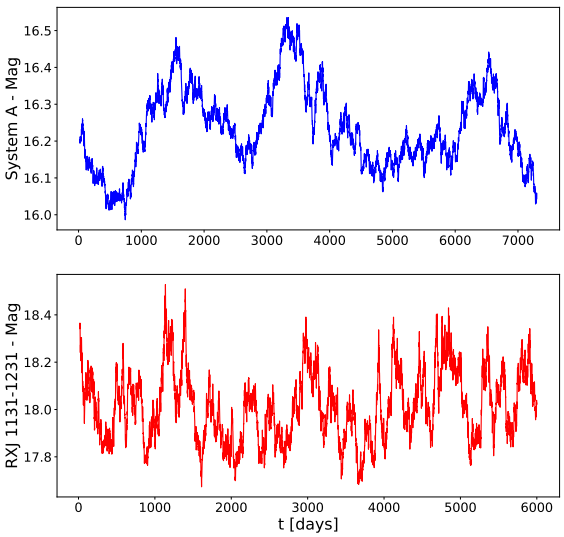

Figure 3. Assumed intrinsic variability of the quasar point source of system *A* (top) and *RXJ 1131-1231* (bottom).

3 GENERATION OF MOCK LIGHT CURVE DATASETS

We use mock data of the lensed systems to generate the training set required by our deep learning algorithm. Specifically, the *MOLET*[2] software package (Vernardos 2021) is used to generate light curves of GLQs multiple images. *MOLET* allows to include the microlensing effect that affects the time delay estimates and is thought to be constantly present (Millon et al. 2020). Fig. 1 in (Vernardos 2021) illustrates the flow chart of *MOLET*: it receives in input different information such as cosmological and astrophysical parameters of the lensed system, the intrinsic light curves of the source, magnification maps, the telescope parameters and a realistic observational plan accounting for daily and seasonal gaps.

We chose to simulated two systems with different features, broadly representative of the various known lensed quasars. The first one, hereafter denoted as system *A*, is a basic test-system for an AGN point source, with a simple intrinsic variability and with microlensing. The second system is RXJ 1131-1231 (Claeskens et al. 2006), hereafter denoted as system *B* for brevity. It consists of four multiple images: here they will be combined in pairs to mimic an unresolved doubly-imaged quasar.

To properly simulate the observed light curves, *MOLET* needs the intrinsic variability of the quasar sources. The two systems that we analyze feature distinct intrinsic variabilities, as shown in Fig. 3, representative of two typical regimes for lensed quasars. The magnification maps needed by the second step of the *MOLET* simulation are available for both systems from the GERLUMPH resource (Vernardos et al. 2014). Finally, the last step of the *MOLET* run accounts also for the assumed instrumental gaps (daily or seasonal) to simulate a

[2] https://github.com/gvernard/molet

realistic campaign from an optical telescope. More details on the simulation of system B can be found in Vernardos (2021).

To build our mock up data, we run *MOLET* several times, by fixing the value of the input time delay and extracting for each run random numbers in a given range of time delays (0-40 days): in this way we obtained 2000 simulations of system A, and 8000 simulations of system B. Each simulation produced four resolved light curves, one for each multiple image. The original output of the mock simulation is continuous light curves: these must then be made discrete and noise must be added to them as well. Since the goal of our work is to measure the time delay in unresolved doubly-imaged systems, the separate continuous mock data curves have to be added up in pairs to obtain a single light curve that mimics an unresolved doubly-imaged GLQ. In this way, a total of six realisations of a single unresolved light curve of a doubly-imaged quasar are obtained from the individual light curves of the four images of the original GLQ. Each of these six light curves is characterised by a different time delay, which is given by the absolute difference of the time delays of the underlying light curves. The combination of the light curves is performed adopting the following definition of the magnitude of a generic source X:

$$\mathrm{mag}_X = -2.5 \log_{10} F_X + K, \tag{3}$$

where F_X is the *flux* of the source X, i.e. the energy per unit time per unit area incident on the detector, and the constant K defines the zero point of the magnitude scale. When two images A and B of a GLQ cannot be resolved, a single image O will appear on the detector, with a flux given to a first approximation by the sum $F_O = F_A + F_B$ of the fluxes of the two overlapping images. According to Equation (3), the corresponding magnitude is:

$$\mathrm{mag}_O(t) = -2.5 \log_{10} \left(F_A(t) + F_B(t + \Delta t)\right). \tag{4}$$

If B is delayed by Δt with respect to A, the information about the time-delay Δt should still be present in the features of the light curve of the image sum O. Such information needs to be properly extracted as it is hidden by the microlensing effect.

Using all the various possible combinations in pairs of the multiple images has allowed to get $6 \cdot 2000 = 12000$ unresolved light curves for system A and $6 \cdot 8000 = 48000$ unresolved light curves for system B. Figure 4 shows ten light curves for each system, randomly chosen by the whole simulation sets, for illustrative purpose.

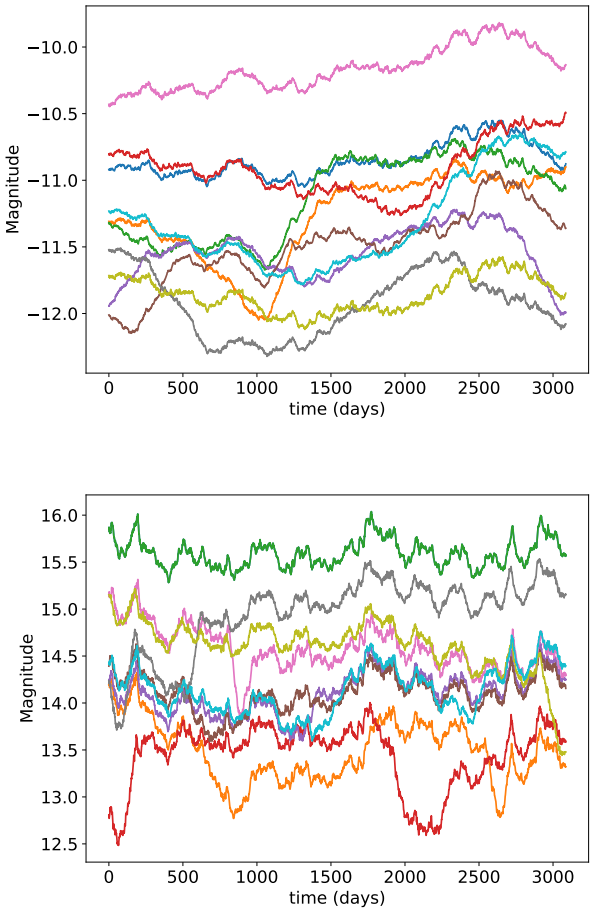

Figure 4. Continuous sample light curves for system A (top) and system B (bottom). The light curves have been randomly chosen from the whole simulation sets for illustrative purposes. Therefore, the observed differences are due to the fact that each of them comes from a different simulation, with different microlensing effects. The simulation period covers about 10 years of data taking.

4 RESULTS

Here we report our results on the performance of the three proposed CNN architectures on the *test data*, i.e. light curves that our models have never seen during their training phase. The goal is to verify their level of generalization on *new* test curves extracted from the same distribution as the training data. Note that, since the test data are generated with the same simulation engine used to produce the training set, here the out-of-distribution generalization ability of our algorithm is not tested, i.e. we do not test its robustness with respect to the sim2real gap phenomenon.

The performance of the models on both systems A and B is analysed at different levels. The first evaluation studies the error distribution $t_{predicted}$-t_{true} between the predicted time delay and the ground truth, i.e. the time delay used to construct the training set. Given that the optimizer we use has a stochastic component in its operation, the outcome of the training can be slightly different from one run to another, so several training runs are performed for each model and, for this experiment, the best performing one for each architecture is selected. The kernel density estimation (Silverman 1986; Scott 1992) is used to approximate the error distributions for each model and each system. The results are shown in Fig. 5. For both systems, the distribution provided by the CNN model is much narrower and symmetric around zero, that is the predicted time delay is closer to the input time delay. ResNet seems to outperform InceptionTime, even though both provide broader and more skewed curves.

As a second evaluation, the distribution of the r^2 coefficient of determination (Silverman 1986; Scott 1992) is investigated to establish to what extent prediction and ground truth are aligned. The r^2 score represents the fraction of the variance of a dependent variable that can be predicted from the independent variables and is then a metric used to evaluate the

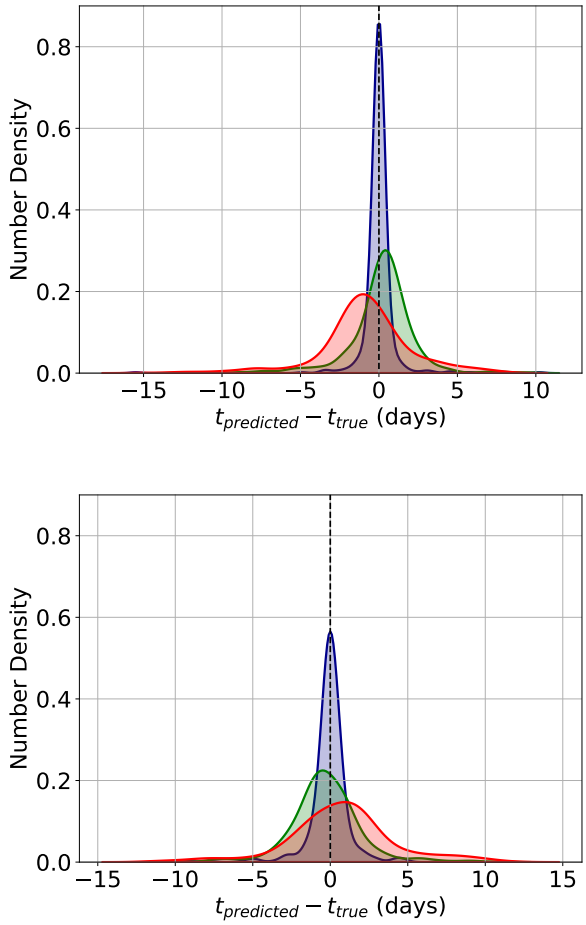

Figure 5. Kernel Density Estimation of the error distributions on the test set provided by our CNN model (blue), InceptionTime (red) and ResNet (green), for the system A (top) and system B (bottom).

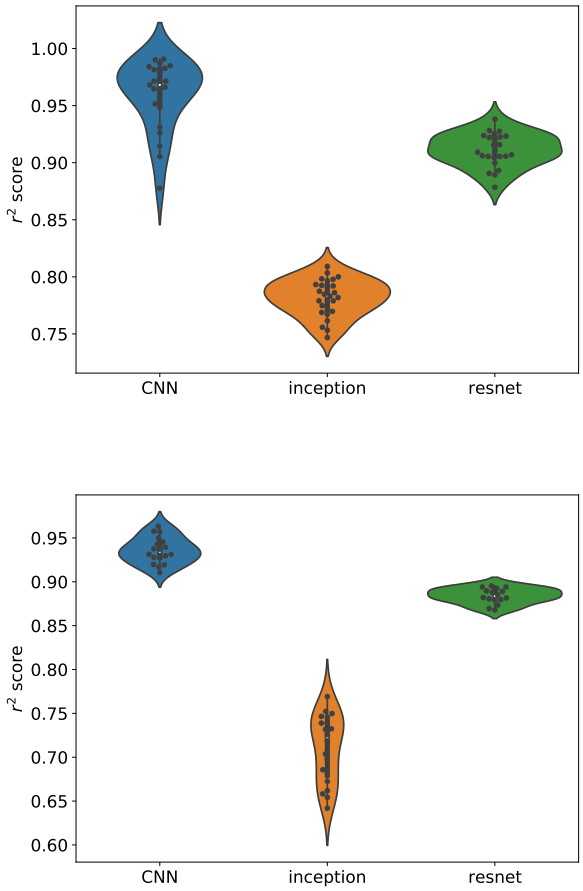

Figure 6. r^2 score distributions provided by the three considered models, for system A (top) and system B (bottom).

goodness of fit of a regression model, a value of 1 being the ideal case. For each system, each model is trained 20 times and the performance of each trained model is assessed on the test set in terms of the r^2 score. Figure 6 shows the resulting probability density of the data at different values, the so-called "violin plots": in both cases, our CNN yields r^2 scores closer to 1. However, in the case of system A, the r^2 distribution of our CNN is characterized by a slightly larger variance, resulting in an overlap with the ResNet r^2 distribution. InceptionTime is outperformed by the other two baselines in both systems and, in the case of system B, it produces a high variance distribution with realizations ranging between a minimum of 0.6 and a maximum of 0.8 r^2 scores.

In summary, the analysis in terms of error distribution and r^2 score highlights that all models provide very good performance on both systems. It is important to emphasize that InceptionTime and ResNet were not fine-tuned, in order to keep them the same as the original implementations from the literature. This was done on purpose to showcase the flexibility of these models to work on very heterogeneous types of data. We therefore expect their performance to further improve with more careful architectural and hyper-parameter design choices.

After having verified that the proposed methods generalize well on the test set, their robustness is assessed when noise is injected in the data. To this extent, we perturb the original time series with a zero-mean Gaussian noise with standard deviation $\sigma \in \{0.00001, 0.0001, 0.001, 0.002, 0.004, 0.006, 0.008, 0.009\}$. These values have been selected so that the main structural properties of the resulting light curves are not altered too much by the injection of noise and their macroscopic visual appearance is roughly preserved. This operation effectively introduces a bias between training and testing distributions, whose severity depends on the intensity of the perturbation. Figure 7 shows how the r^2 score of each model decreases as the standard deviation of the gaussian noise increases. Again, generally the CNN outperforms the other models. However, in system A, its curve tends to align with the InceptionTime one as the noise level increases. Interestingly, the CNN model seems to be more robust on system B, providing satisfactory values of the r^2 score even for relatively large noise standard deviations. In general, the performance of the three models on system A seems to be much more sensitive to noise perturbation than that obtained by the models on system

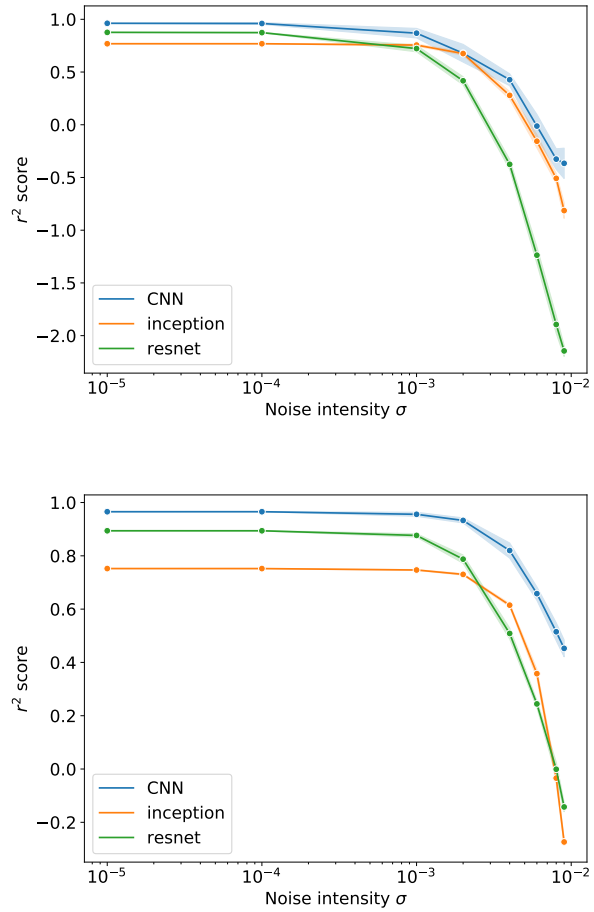

Figure 7. r^2 scores for the three models as injected noise standard deviation increases, for system A (top) and system B (bottom).

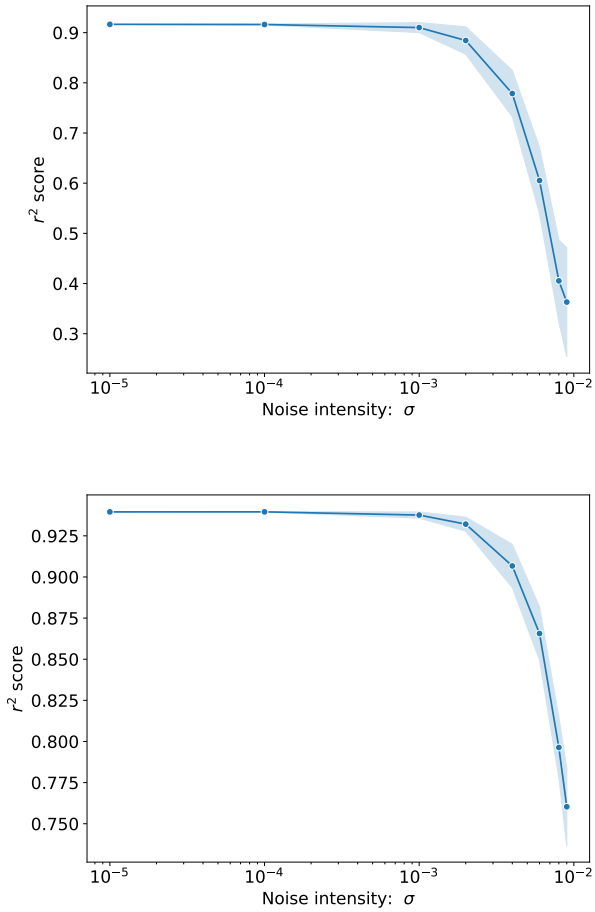

Figure 8. r^2 scores for our CNN model trained with random noise injection as noise standard deviation increases, for system A (top) and system B (bottom).

B. We do not have a satisfactory explanation for that yet: one likely possibility is that it depends on the difference in the size of the datasets associated with the two systems, but more investigations are needed to draw a firm conclusion.

As a last experiment, we study how the previous analysis changes if we train our best performing CNN on data perturbed by noise at variable standard deviations. To do so, at training time for each batch of data, we randomly sample a value for the noise standard deviation from a uniform distribution between 10^{-5} and 10^{-2} and we feed the corrupted data into the network. Once training is terminated, we repeat the previous experiments. The results for system A and system B are shown in Fig. 8.

The result of injecting noise at training time is a model which is more robust to perturbations in the test data. On the other hand, this positive effect is obtained at the price of a slight degradation in performance when the noise level is low. This experiment suggests that randomly injecting noise in the data at training time in this case represents an effective strategy to obtain more robust models.

In light of the presented experiments, the proposed CNN architectures appear to guarantee satisfactory performance on the task of predicting the time delay from unresolved quasar light curves. Overall, the CNN model proposed here seems to outperform the others even though more fine-tuning and more carefully designed choices might eventually close this gap.

5 CONCLUSIONS

In this work, a new class of deep learning-based methods has been used to extract the time delay between GLQs unresolved light curves. The method is motivated by the existing tension on the estimated values of H_0, which can only be resolved by reducing the uncertainty on H_0. In this respect, there is the necessity to increase the number of analysed GLQs by processing and extracting information from unresolved quasar light curves, which can be typically provided by small/medium telescopes. The obtained results show that the proposed approach performs nicely on mock data describing two different lensed quasar systems. Moreover, it has several advantages compared to classical approaches: first, it is designed to process unresolved light curves which represent the majority of the data that small/medium sized telescopes are expected to provide. Second, the method is fully data-driven: its performances scale easily with the dataset size

and it makes little to no assumptions on the nature of the time-delay estimation problem.

On the other hand, the current implementation is still affected by some limitations that open new interesting research questions. Most importantly, the method is highly reliant on the quality of the simulated data and on their level of fidelity with respect to real quasar light curves. It follows that, when applying the proposed approach to real data, a degradation in performance shows up. This problem is a manifestation of the sim2real gap phenomenon described in Sec. 4. In order to alleviate it, the gap between simulated and real data must be reduced as much as possible. One popular strategy to cope with this problem is the technique of *domain randomization* (Tobin et al. 2017; Peng et al. 2018; Chebotar et al. 2019; Prakash et al. 2020), where large and very diverse datasets are generated by randomizing the parameters of the simulator with the hope that, when the model is deployed on real data, the new observations will somewhat close to the randomized simulations the model has been trained on.

On the other hand, the sim2real gap phenomenon, and in particular the size of the discrepancies, can be exploited to assess the quality of the simulated data: the better the performance of the network, the closer the simulator grasps the details of the data it is trying to emulate.

Lastly, as a further improvement to the method that will be investigated in future works, is to enable it to process also irregularly sampled time series which are indeed commonly encountered in the context of cosmological and astrophysical applications, because of unavailabilities or outages of the instruments, for example. Finally, a future extension of this work will deal with systems having $N > 2$ images: in fact, a large amount of quadruply-imaged GLQs is expected to be detected in the future (Shu et al. 2021). We aim to investigate these new research directions in future works.

ACKNOWLEDGEMENTS

Part of this work was supported by the German *Deutsche Forschungsgemeinschaft, DFG* project number `Ts 17/2-1`. GV has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodovska-Curie grant agreement `No 897124`. Authors acknowledge support from the "Departments of Excellence 2018-2022" Grant (`L. 232/2016`) awarded by the Italian Ministry of University and Research (MUR).

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Abell P., et al., 2009, LSST Science Book, Version 2.0 (arXiv:0912)
Agarap A. F., 2019, Deep Learning using Rectified Linear Units (ReLU) (arXiv:1803.08375)
Aghanim N., et al., 2020, A &A, 641, A6
Bag S., et al., 2021, Identifying lensed quasars and measuring their time-delays from unresolved light curves (arXiv:2110.15315)
Beenakker W., Venhoek D., 2021, A structured analysis of Hubble tension (arXiv:2101.01372)
Bengio Y., Lecun Y., 1997, Conference: The Handbook of Brain Theory and Neural Networks
Birrer S., Treu T., 2020, A &A, 649, A61
Birrer S., et al., 2020, A &A, 643, A165
Borgeest U., et al., 1996, in , Examining Big Bang Diffus. Backgr. Radiations. Springer, pp 527–528
Cabrera-Vives G., et al., 2016, 2016 International Joint Conference on Neural Networks (IJCNN), pp 251–258
Chebotar Y., Handa A., Makoviychuk V., Macklin M., Issac J., Ratliff N., Fox D., 2019, Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience (arXiv:1810.05687)
Claeskens J. F., Sluse D., Riaud P., Surdej J., 2006, A&A, 451, 865
Collett T. E., 2015, ApJ, 811
Di Valentino E., et al., 2021, Class. Quantum Grav., 38, 153001
Fawaz I., Lucas B., Forestier G., et al., 2020, Data Min Knowl Disc, 34, 1936–1962
George D., Shen H., Huerta E. A., 2017, Physical Review D, 97
He K., Zhang X., Ren S., Sun J., 2015, Deep Residual Learning for Image Recognition (arXiv:1512.03385)
Heiden E., Millard D., Coumans E., Sukhatme G. S., 2020, Augmenting Differentiable Simulators with Neural Networks to Close the Sim2Real Gap (arXiv:2007.06045)
Ioffe S., Szegedy C., 2015, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift (arXiv:1502.03167)
Karttunen H., Kröger P., Oja H., Poutanen M., Donner K. J., 2017, Fundamental Astronomy. Springer-Verlag Berlin and Heidelberg Gm, doi:10.1007/978-3-662-53045-0
Kimura A., et al., 2017, IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)
Kingma D. P., Ba J., 2015, Adam: A Method for Stochastic Optimization (arXiv:1412.6980)
Krizhevsky A., Sutskever I., Hinton G. E., 2012, in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12. Curran Associates Inc., Red Hook, NY, USA, p. 1097–1105
McNeely-White D. G., Beveridge J. R., Draper B. A., 2020, in Samsonovich A. V., ed., Biologically Inspired Cognitive Architectures 2019. Springer International Publishing, Cham, pp 352–357
Millon M., et al., 2020, A &A
Oguri M., Marshall P. J., 2010, MNRAS, 405, 2579–2593
Peng X. B., Andrychowicz M., Zaremba W., Abbeel P., 2018, 2018 IEEE International Conference on Robotics and Automation (ICRA)
Prakash A., Boochoon S., Brophy M., Acuna D., Cameracci E., State G., Shapira O., Birchfield S., 2020, Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data (arXiv:1810.10093)
Refsdal S., 1964, MNRAS, 128, 307
Reimers C., Requena-Mesa C., 2020, in Škoda P., Adam F., eds, , Knowledge Discovery in Big Data from Astronomy and Earth Observation. Elsevier, pp 251–265, doi:https://doi.org/10.1016/B978-0-12-819154-5.00024-2, https://www.sciencedirect.com/science/article/pii/B9780128191545000242
Riess A. G., et al., 2019, ApJ, 876, 85
Schawinski K., et al., 2017, MNRAS, 467, 110
Scott D. W., 1992, Multivariate Density Estimation. Wiley
Sedaghat N., Mahabal A., 2018, MNRAS, 476
Shallue C. J., Vanderburg A., 2018, The Astronomical Journal, 155
Shu Y., Belokurov V., Evans N. W., 2021, MNRAS, 502, 2912–2921
Silverman B. W., 1986, Density Estimation for Statistics and Data

Analysis. Chapman and Hall
Sultana F., Sufian A., Dutta P., 2020, Knowledge-Based Systems, 201-202, 106062
Tewes M., et al., 2012, A &A, 553
Tie S. S., Kochanek C. S., 2017, MNRAS, 473, 80–90
Tobin J., Fong R., Ray A., Schneider J., Zaremba W., Abbeel P., 2017, Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World (arXiv:1703.06907)
Verde L., Treu T., Riess A., 2019, Nature Astronomy, 3, 891
Vernardos G., 2021, Simulating time-varying strong lenses (arXiv:2106.04344)
Vernardos G., et al., 2014, ApJS, 211, 16
Wei W., Huerta E. A., 2021, Physics Letters B, 816
Wong K. C., et al., 2019, MNRAS, 498, 1420–1439
Zhao W., Queralta J. P., Westerlund T., 2020, in 2020 IEEE Symposium Series on Computational Intelligence (SSCI). pp 737–744, doi:10.1109/SSCI47803.2020.9308468

This paper has been typeset from a TEX/LATEX file prepared by the author.