DeepQuasar: Simulation-based inference for lensed quasar modeling

E. Danilov\textsuperscript{1,*}, A. Ciprijanovic\textsuperscript{2} and B. Nord\textsuperscript{2,3,4}

1. Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland
2. Fermi National Accelerator Laboratory, USA
3. Department of Astronomy and Astrophysics, University of Chicago, USA
4. Kavli Institute for Cosmological Physics, University of Chicago, USA

Research supported by Deep Skies Lab
Funding by Fermilab and U.S. Department of Energy
Contact email: egorssed@gmail.com
Hubble constant $H_0$ in LCDM

Expansion of the Universe
$$a(t) = a_0 e^{H_0 t} + ...$$

Insight into dark energy
$$\Lambda = 3H_0^2 + ...$$

Critical density of the Universe
$$\rho_c = \frac{3H_0^2}{8\pi G}$$

Fate of the Universe
- Collapse
- Expansion
- Big rip

Images credit: NASA, ESO Supernova
Measurement discrepancies

Hubble rate $H_0$

- Direct methods
  - $H_0 = 74.03 \pm 1.42 \text{ km s}^{-1} \text{ Mpc}^{-1}$

- Indirect methods
  - $H_0 = 67.4 \pm 0.5 \text{ km s}^{-1} \text{ Mpc}^{-1}$

$>3\sigma$ discrepancy

- Mistakes in detectors
- New physics

Solution: Time delay cosmology
Gravitational lensing

Ordinary lensing

Gravitational lensing

Quasar
Background

Galaxy
Foreground

Our Vantage
Point

Light travels out from Quasar

Galaxy’s gravity acts as a lens, bending Quasar’s diverging light paths back towards us

From our vantage point, Quasar’s light arrives from four different directions

A telescope sees the single Quasar as four objects surrounding the Galaxy

Time delay:

\[ t(\vec{\theta}) = \frac{D \Delta t}{c} \phi(\vec{\theta}) \propto \frac{1}{H_0} \phi(\vec{\theta}) \]
Gravitationally Lensed Quasars (LQSO)

Lensed quasar light curves

Magnitude

2000 2001 2002 2003
Time, year

18.0 18.3 18.6
Magnitude

\[ \Delta t \propto \frac{1}{H_0} \]

Lensing
Shifts: time and magnitude

Earth

Image credit: fftech.net
Outline of the presentation

1. Challenges of inference
2. Neural networks-based method
3. Performance of neural networks
Challenges of inference
Quasar model - accretion + reverberation

\[ I(t) \propto \int_0^\infty f_{\text{acc}}(t - t_\lambda \rho) P_{\text{disk}}(\rho) d\rho \sim \mathcal{N}(\mu, \Sigma[\hat{\sigma}^2, \tau, t_\lambda]) \]

Quasar light curve

Accretion onto black hole

Reverberation of the disk

Gaussian stochastic process

Images credit: NASA
Quasar model - Reverberating Damped Random Walk

Autocorrelation:

\[ \int_{-\infty}^{\infty} f(t' + \Delta t) f(t') dt' \propto \tau \cdot \exp \left( -\frac{\Delta t}{\tau} \right) - 4t_\lambda \cdot \exp \left( -\frac{\Delta t}{4t_\lambda} \right) \]

Correlation time \( \tau \)

Global evolution (years)

Reverberation time \( t_\lambda \)

Local evolution (days)
Challenge of inference

\[ \tau = 1000 \text{ days}, \ t_\lambda = 30 \text{ days} \]

1σ constraints:

\[ 200 < \tau < 20000 \text{ days} \]
\[ 25 < t_\lambda < 125 \text{ days} \]

1σ constraints:

\[ 990 < \tau < 1410 \text{ days} \]
\[ 29 < t_\lambda < 40 \text{ days} \]
Neural networks-based method

Data → Summary → Temporal summary → Posterior predictions

- Dimensionality reduction
- Posterior predictions
Dimensionality reduction with VAE

Data → Encoder → Latent summary → Decoder → Reconstruction

Loss function

\[ \mathcal{L} = \text{MSE}(x, x') + \beta \cdot D_{KL}(p(z|x) \parallel p(z)) \]

- Quality of reconstruction
- Amount of information
Posteriors from Normalizing flow

Summary of data → Importance picking → Temporal summary → Normalizing flow → Posteriors of temporal parameters

Summary of data

Importance picking

Temporal summary

Normalizing flow

Posteriors of temporal parameters
Result performance
Variational autoencoder’s reconstruction

Accurate curves reconstruction

Relative error: 4%

Error: RMSE=0.08
Quasar parameters inference

Truth:
\[ \tau = 1000 \text{ days} \]
\[ t_\lambda = 30 \text{ days} \]

1σ constraints:
\[ 990 < \tau < 1410 \]
\[ 29 < t_\lambda < 40 \]
Errors of prediction

Prediction:
\[ x = x_{pred} \pm \sigma_x \]

Absolute error:
\[ \Delta x \rightarrow 0 \]

Normalized error (Chi):
\[ \chi(x) = \frac{x_{pred} - x_{g.t.}}{\sigma_x} \]

Should be:
\[ \chi(x) \rightarrow \mathcal{N}(0, 1) \]
Statistical distribution of errors

Distribution of absolute errors

\[ x_{pred} - x_{g.t.} \]

Conclusion: accurate prediction

\[ 0.5 < \frac{x_{pred}}{x_{g.t.}} < 2 \]

Distribution of normalized errors

\[ \chi(x) = \frac{x_{pred} - x_{g.t.}}{\sigma_x} \]

Conclusion: reliable uncertainties

\[ \sigma_{pred} \approx 0.93 \cdot \sigma_{truth} \]
Lensing inference

Posterior results:

- Negligible magnification error
- Accurate time delay prediction

\[ \Delta M = -0.5^{+0.001}_{-0.003} \]
\[ \Delta t = -200.8^{+1.6}_{-1.7} \]
Comparison with standard software

PyCS3 setup:
- Adjusted necessary hyperparameters
- The rest left to default

PyCS3 warnings:
- does not guarantee realistic time-delay uncertainty estimates
- Please try to avoid publishing time delays with overly optimistic uncertainty estimates

<table>
<thead>
<tr>
<th>Method</th>
<th>Inference time</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our method</strong></td>
<td><strong>20 sec</strong></td>
<td>±1.6 days</td>
</tr>
<tr>
<td>PyCS3 Splines</td>
<td>13 min</td>
<td>±7.4 days</td>
</tr>
<tr>
<td>PyCS3 Gauss proc.</td>
<td>9 min</td>
<td>±8.2 days</td>
</tr>
</tbody>
</table>

Comparison with PyCS3 package (no fine-tuning)

- Package credit: Millon et al. 2020, Tewes et al. 2013
Improvement of Hubble rate $H_0$

Uncertainty from time delay:

$$H_0 \pm \sigma_H \propto \frac{\Delta \dot{\phi}_{g.t.}}{\Delta t \pm \sigma_t}$$

PyCS3: $\sigma_H = 2.1 \text{ km s}^{-1} \text{ Mpc}^{-1}$

Our method: $\sigma_H = 0.6 \text{ km s}^{-1} \text{ Mpc}^{-1}$

3.5 times less
Summary

Quasar light curves
- Relative error: \(0.5 < \frac{x_{\text{pred}}}{x_{\text{g.t.}}} < 2\)
- Uncertainty \(\sigma_{\text{pred}} \approx 0.93 \cdot \sigma_{\text{truth}}\)

Lensing inference
- Magnification uncertainty 0.001
- Time delay uncertainty \(\pm 1.6\) days

Hubble rate improvement
- Uncertainty from time delay 0.6
- Uncertainty improved in 3.5 times

Contact email: egorssed@gmail.com
Thank you for your attention

Contact email: egorssed@gmail.com
Additional materials
Data preprocessing

<table>
<thead>
<tr>
<th>Target parameter</th>
<th>Time scale</th>
<th>Image axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation time $\mathcal{T}$</td>
<td>Years</td>
<td>Y axis</td>
</tr>
<tr>
<td>Reverberation time $t_\lambda$</td>
<td>Days</td>
<td>X axis</td>
</tr>
</tbody>
</table>
Inference pipeline architecture

Observations

Annual splines

Latent summary

Splines reconstruction

Temporal evolution summary

Reconstruction

Pipeline diagram

Data encode Summary decode Data reconstruction

Temporal summary

importance picking

Temporal parameters

Normalized flow

Temporal parameters

Posteriors of temporal parameters

Inference pipeline architecture

Observations

Annual splines

Latent summary

Splines reconstruction

Temporal evolution summary

Reconstruction

Pipeline diagram

Data encode Summary decode Data reconstruction

Temporal summary

importance picking

Temporal parameters

Normalized flow

Temporal parameters

Posteriors of temporal parameters
Variational autoencoder’s reconstruction

Dataset:
- 12 years-long mock light curves
- 800k / 100k / 100k split

Architecture:
- Latent space size 128
- Temporal summary size 24

Training:
- 9 epochs, batch size = 128
- Regularization $\beta = 0.05$
- Loss:
  \[ \mathcal{L} = \text{MSE}(x, x') + \beta \cdot D_{KL}(p(z|x)\|p(z)) \]

Result:
- RMSE = 0.04 for data in [0,1]
- Learning curve

![Learning curve](image)
Variational autoencoder

Reconstruction performance

Dataset:
- 12 years long light curves
- 800k / 100k / 100k split

Training:
- 9 epochs, batch size = 128
- Loss - MSE and KL divergence
- Regularization $\beta = 0.05$

Result:
- RMSE = 0.04 for data in $[0,1]$

Learning curve

Vary one latent variable

Decoder($z_0, 0, ..., 0)$ ~ $z_0 \cdot \sin(x + \phi)$
Variational autoencoder latent effects
Models of a quasar light curve

**Spline**

*Advantages:*  
- No hyperparameters except for degree

*Disadvantages:*  
- Not interpretable
- No uncertainties

**Gaussian process (GP)**

*Advantages:*  
- Physically interpretable parameters
- Get uncertainties for free

*Disadvantages:*  
- Degeneracies between parameters
- Difficult optimization for disbalanced gradients