Identification of Lensed Gravitational Waves with Deep Learning

Kyungmin Kim

Korea Astronomy and Space Science Institute (KASI)

In collaboration with

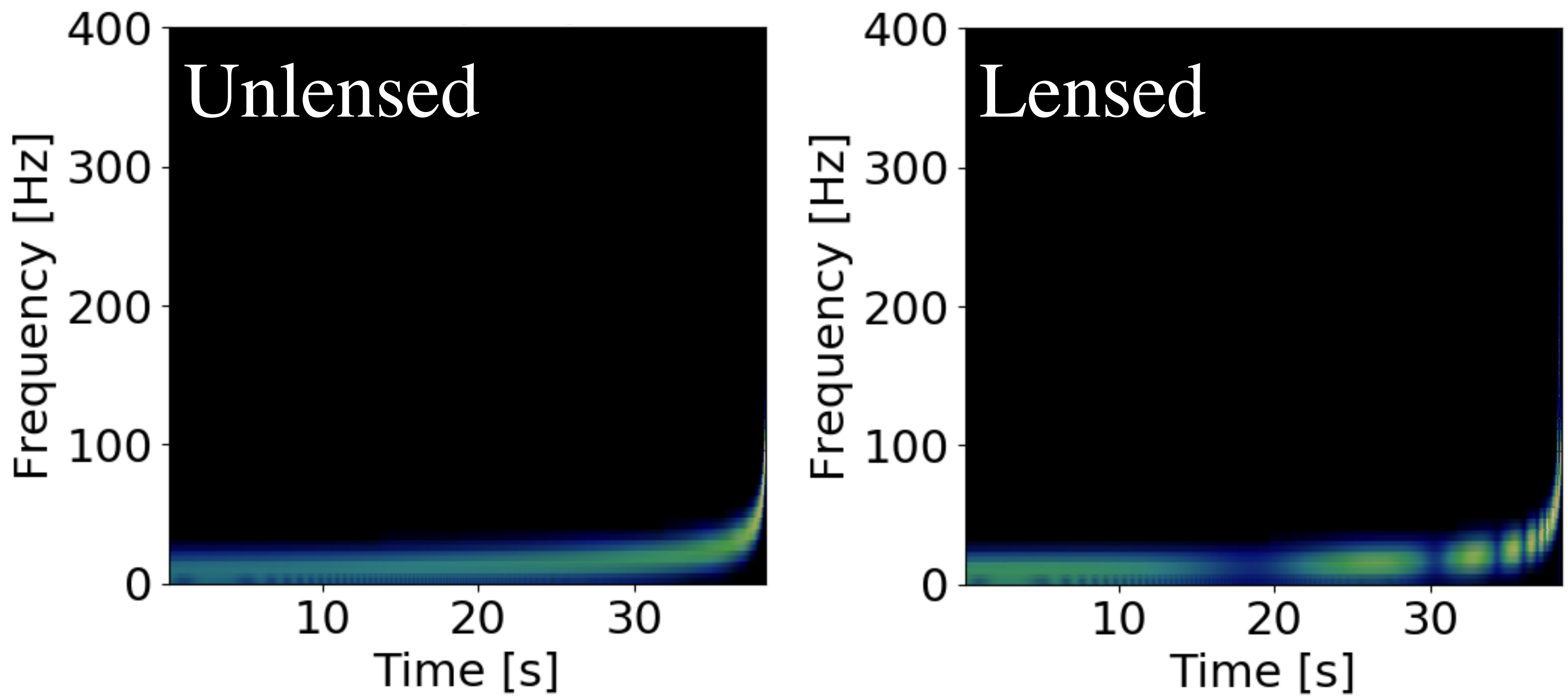
Amit Jit Singh (CUHK), Ivan S. C. Li (Imperial), Robin S. H. Yuen (CUHK),
Otto A. Hannuksela (Nikhef), Joongoo Lee (KASI/SNU), and Tjonnie G. F. Li (CUHK)

Motivation

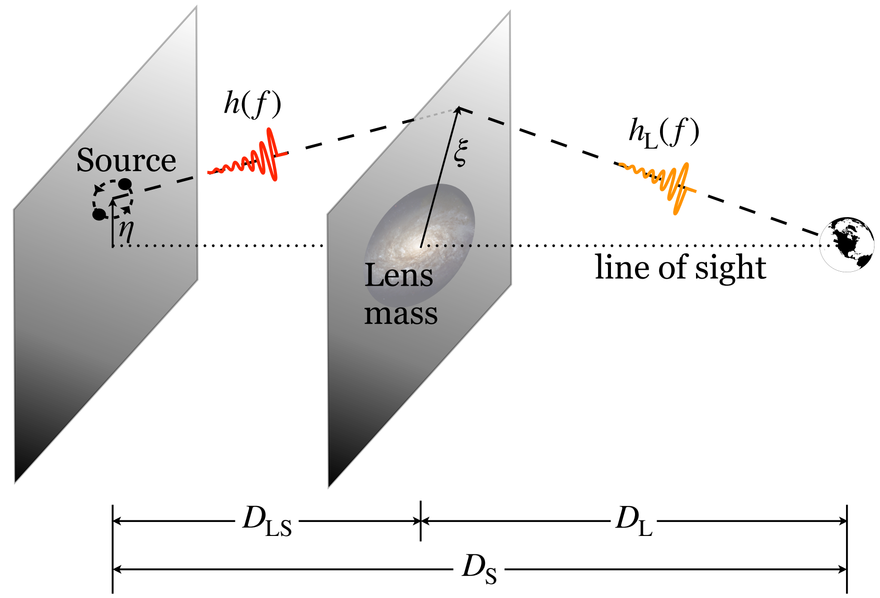
- Expect "gravitational lensing" effect in gravitational waves (GWs) like gravitational lensing in astronomical EM observations.
- If time delay were small enough to make "beating pattern", may identify lensed GW from a spectrogram, time-frequency map, of the chirp signal of GWs from compact binary mergers by using Machine Learning or Deep Learning.

Motivation

- Expect "gravitational lensing" effect in gravitational waves (GWs) like gravitational lensing in astronomical EM observations.
- If time delay were small enough to make "beating pattern", may identify lensed GW from a spectrogram, time-frequency map, of the chirp signal of GWs from compact binary mergers by using Machine Learning or Deep Learning.



Lensing Model [Takahashi & Nakamura, ApJ 2003]



- Strain amplitude of lensed GW in frequency domain

$$h_{\rm lens}(f) = F(f)h(f)$$

where $F(f)$ is the ***amplification factor*** which is written (under the geometrical optics limit) as

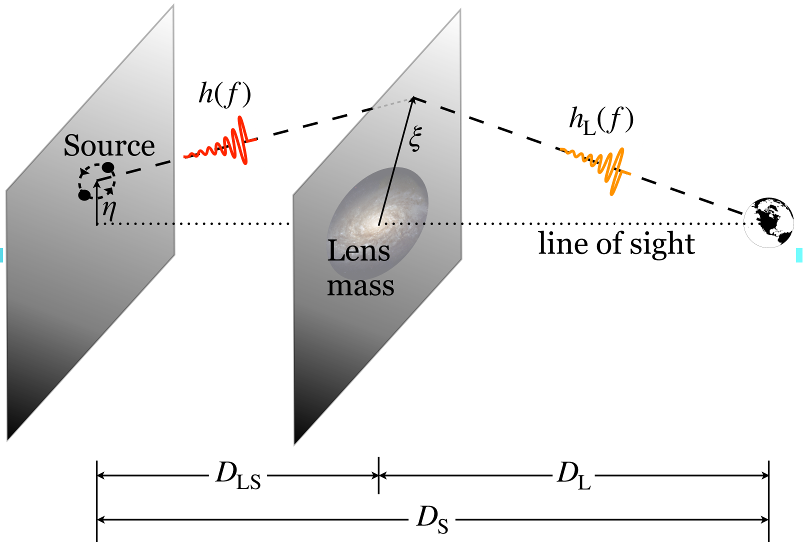
$$F(f) = \sum_j |\mu_j|^{1/2} e^{2\pi i f \Delta t_{d,j} - i\pi n_j}$$

model dependent number

time delay of j-th image

magnification factor

Lensing Model [Takahashi & Nakamura, ApJ 2003]



- Point-Mass (PM) Lens
  - Lens mass: 2-dim Dirac delta function on lens plane

$$F(f) = |\mu_+|^{1/2} - i\,|\mu_-|^{1/2} e^{2\pi i f \Delta t_d}$$

$$M_{Lz} = M_L(1+z)$$

$$\mu_\pm = \frac{1}{2} \pm \frac{(y^2+2)}{(2y\beta)} \qquad \Delta t_d = 4M_{Lz}\left[\frac{y\beta}{2} + \ln\left(\frac{\beta+y}{\beta-y}\right)\right]$$

$$\beta = \sqrt{y^2+4}$$

$$y = \frac{\eta D_L}{\xi_0 D_S}$$

- Singular Isothermal Sphere (SIS)
  - Lens mass: circular and symmetric distribution on lens plane

$$F(f) = \begin{cases} |\mu_+|^{1/2} - i|\mu_-|^{1/2} e^{2\pi i f \Delta t_d} & \text{if } y < 1, \\ |\mu_+|^{1/2} & \text{if } y \geq 1 \end{cases}$$

$$\mu_\pm = \frac{1}{y} \pm 1 \qquad \Delta t_d = 8M_{Lz}y \qquad y = \frac{\eta D_L}{\xi_0 D_S}$$

Strategy

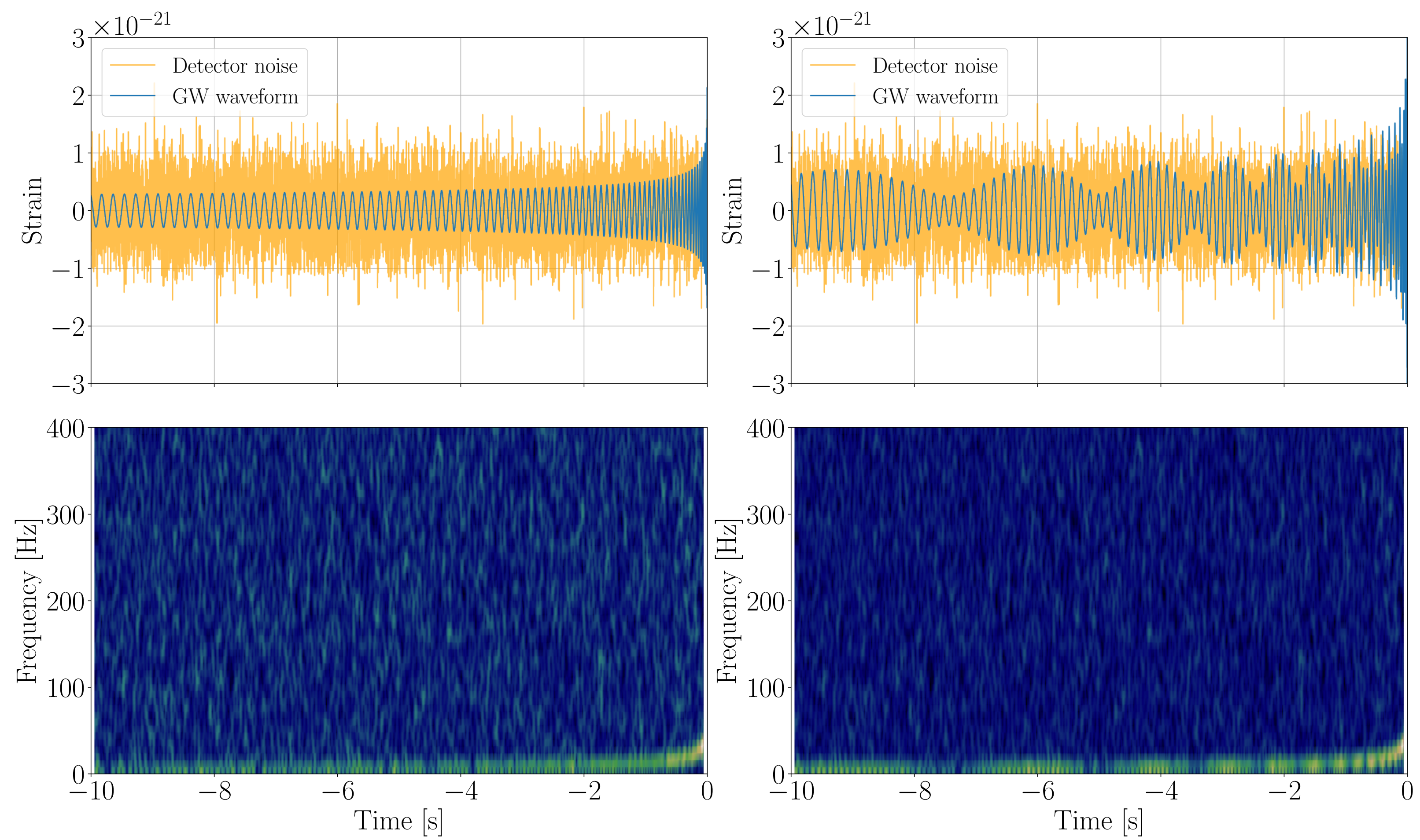
- **Classification**
  - classifies lensed signal from unlensed signal.
  - w/ Scikit-Learn and TensorFlow

↓

- **Regression**
  - estimates lensing parameters
    - source mass, lens mass, distance, and so on.
  - w/ TensorFlow

Classification with Machine Learning w/ Scikit-Learn [Singh et al., AJUR 2019, arXiv:1810.07888]

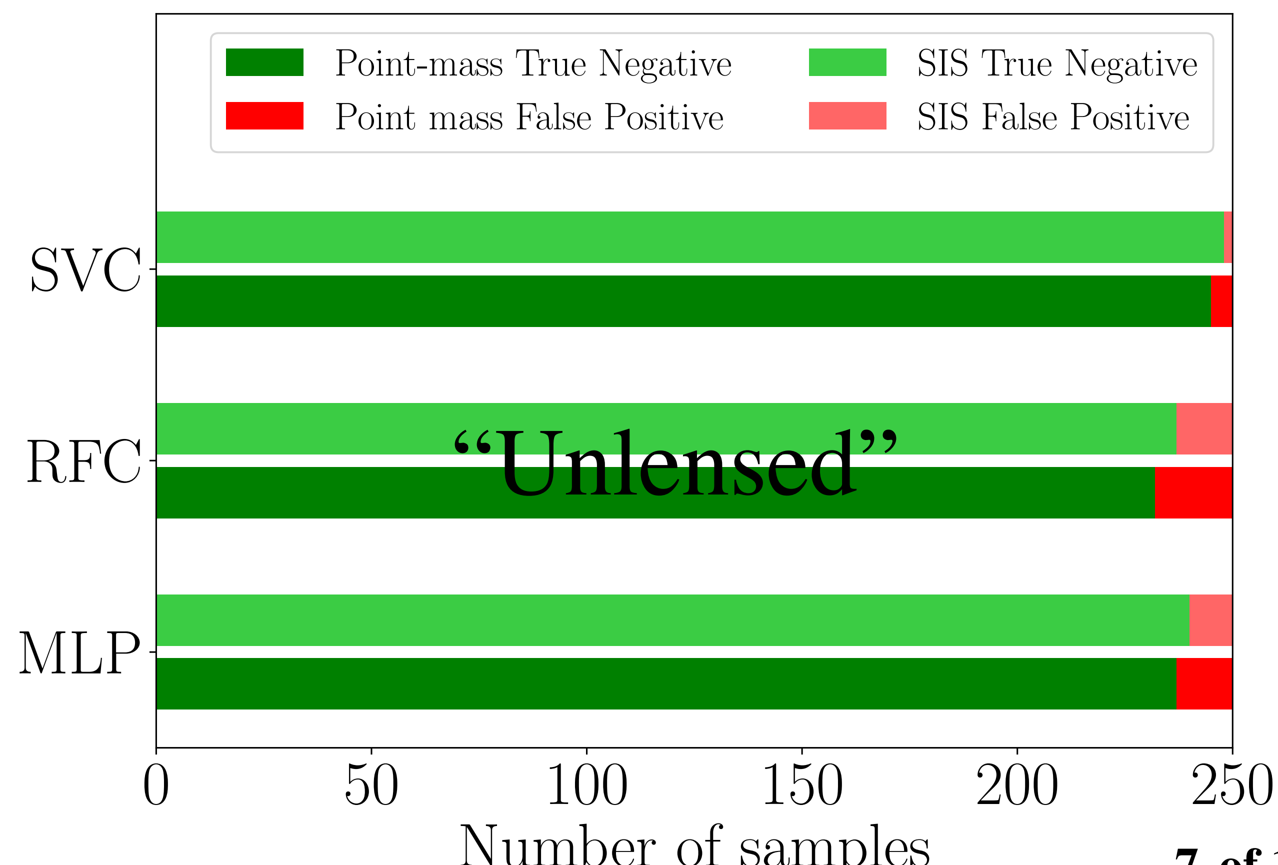
- Input data: 2-dim spectrogram images using Short-Time-Fourier-Transform
  - 2 classes: unlensed vs. lensed
  - # of samples: 1000 samples for unlensed / 1000 samples for lensed (PM or SIS)
  - Parameters
    - m_1, m_2: 4-35 solar mass
    - D_L: 10-1000 Mpc
    - D_{LS}: 10-1000 Mpc
    - M_L: 10-10^7 solar mass
    - ε: 0-0.5 pc
    - redshift factor, z: 0-2
  - Noise: random white
    - SNR: ≤ 80 (c.f. ≤ 23.6 for O1&O2 BBHs in GWTC-1)



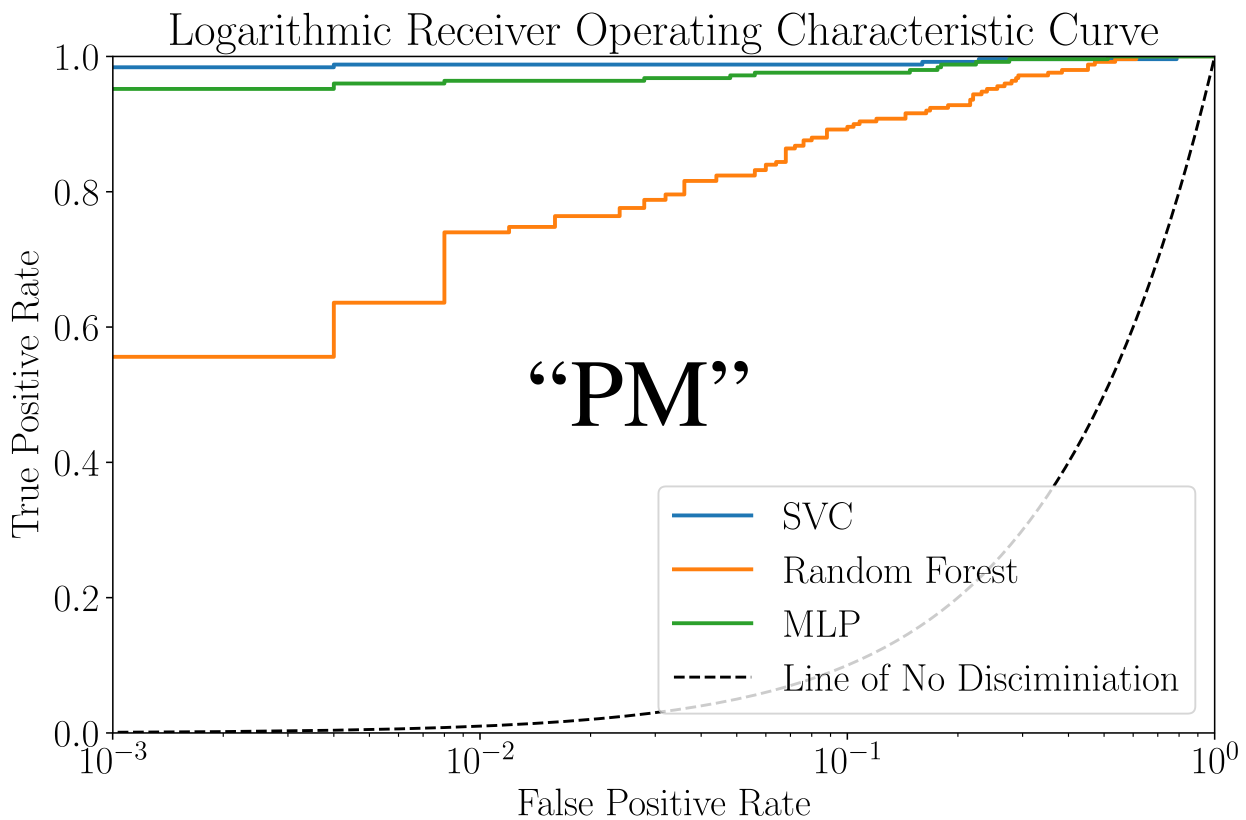
Classification Performance [Singh et al., AJUR 2019, arXiv:1810.07888]

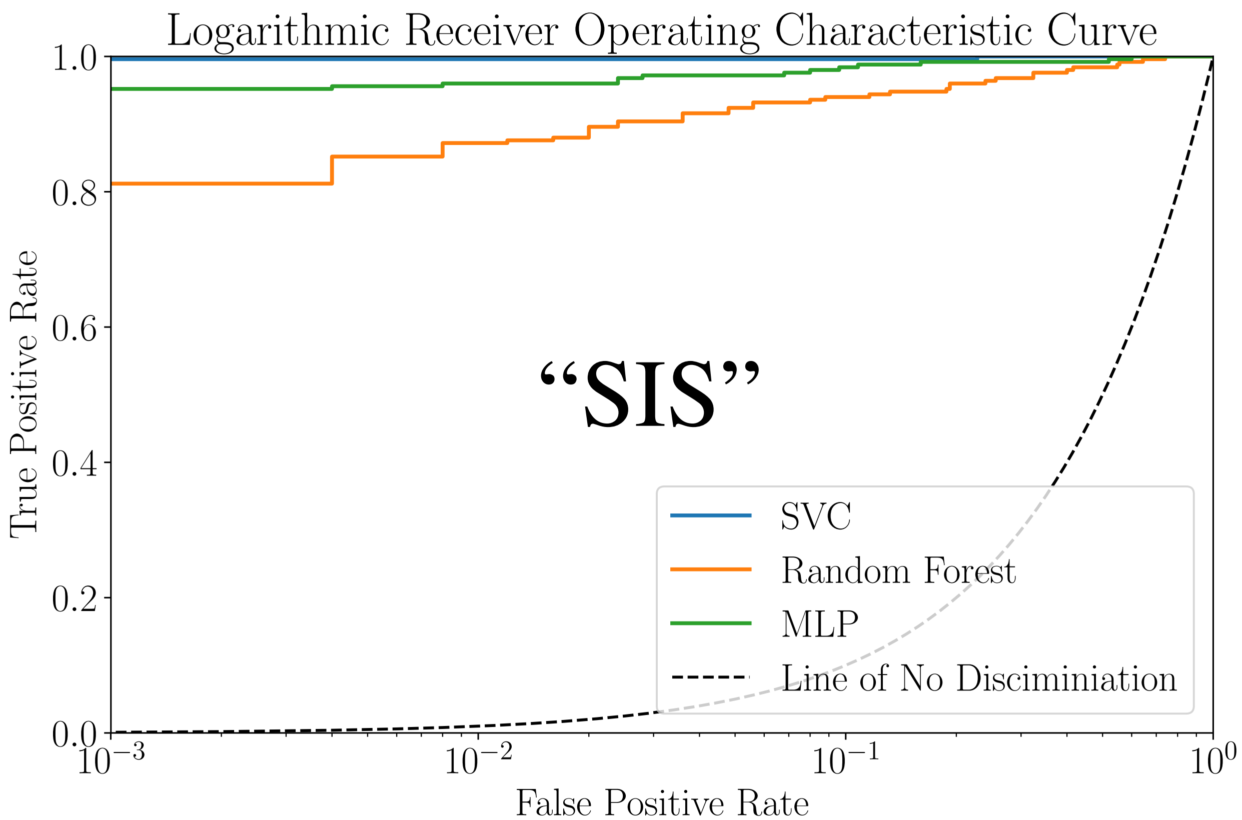
Classifiers	Hyper-parameters	Value
SVC	C	1000
	γ	5×10^{-7}
RFC	N_t	1000
	$N_{\rm mss}$	2
MLP	α	0.72
	$N_{layer\_neurons}$	1000

- ML algorithms
  - Support Vector Machine (SVC)
  - Random Forest (RF)
  - Multi-Layer Perceptron (MLP) ↔ Neural Network
- Confusion Matrix
  - Can say performance is good if
    - True Positive ~ 1, False Negative ~ 0, True Negative ~ 1, False Positive ~ 0



ROC Curve [Singh et al., AJUR 2019, arXiv:1810.07888]





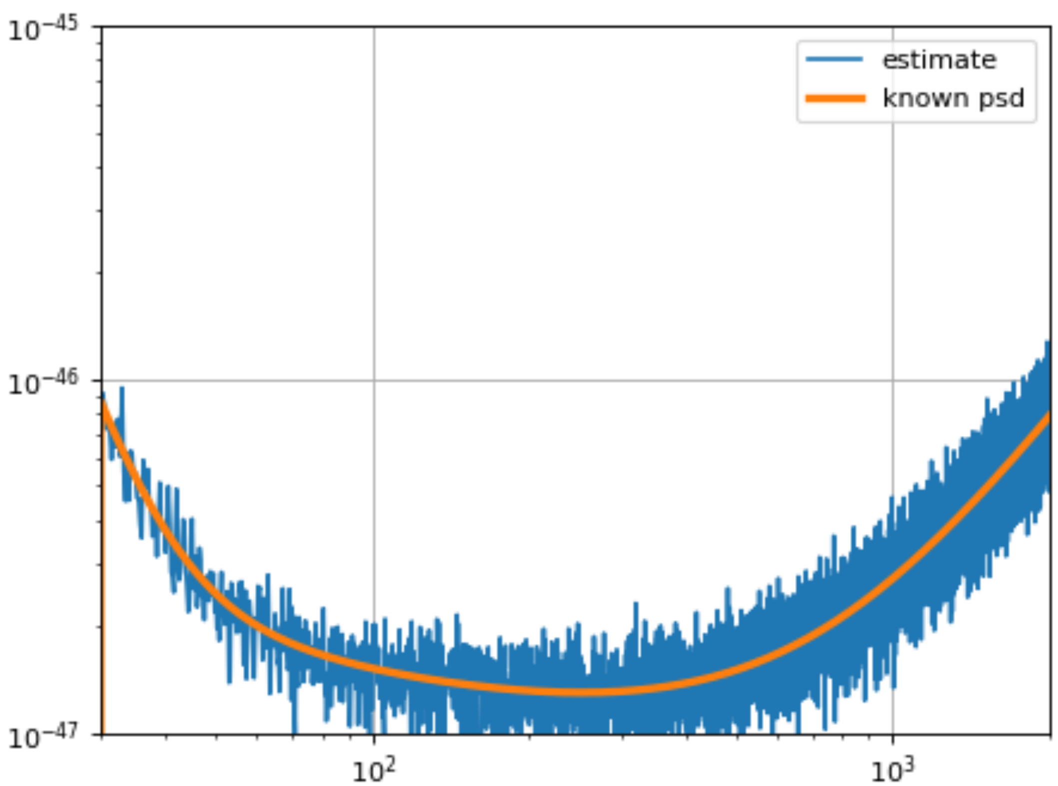
- Area under Curve: $P(\mathrm{rank}_{\mathrm{signal}} > \mathrm{rank}_{\mathrm{noise}})$
  - PM
    - SVC: 0.995 / RF: 0.962 / MLP: 0.993
  - SIS
    - SVC: 0.999 / RF: 0.976 / MLP: 0.993

Summary

- Seems promising to classify lensed GWs from spectrograms of unlensed GWs even with simple ML models.
- For practical application, needs to
  - use LIGO/Virgo noise profile w/ proper restriction on observable SNR in the generation of input image samples and
  - use Qscan spectrogram which has been used in the representation of the chirp signal of detected GWs.

Summary

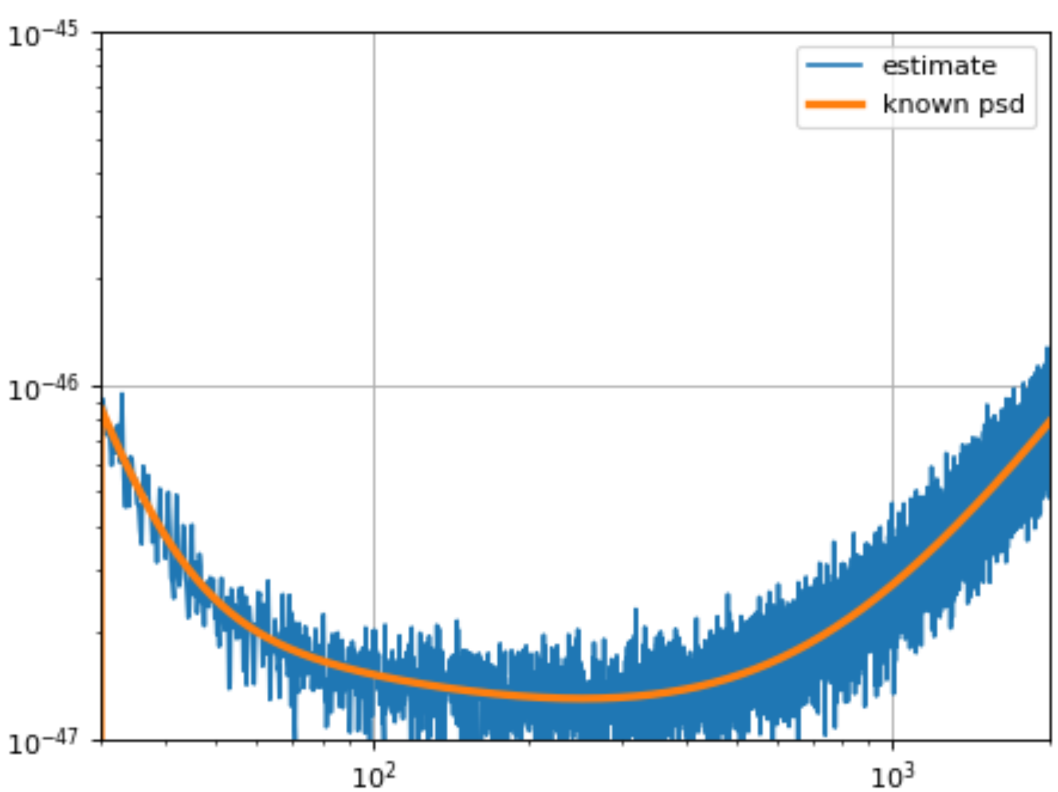
- Seems promising to classify lensed GWs from spectrograms of unlensed GWs even with simple ML models.
- For practical application, needs to
  - use LIGO/Virgo noise profile w/ proper restriction on observable SNR in the generation of input image samples and
  - use Qscan spectrogram which has been used in the representation of the chirp signal of detected GWs.



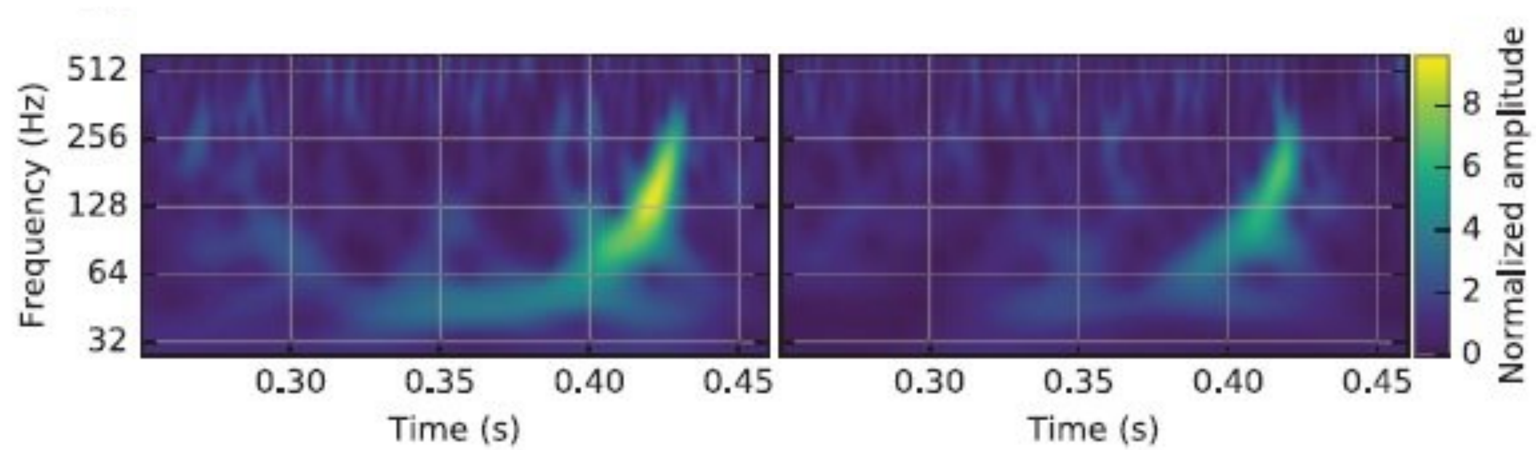
[from PyCBC documentation]

Summary

- Seems promising to classify lensed GWs from spectrograms of unlensed GWs even with simple ML models.
- For practical application, needs to
  - use LIGO/Virgo noise profile w/ proper restriction on observable SNR in the generation of input image samples and
  - use Qscan spectrogram which has been used in the representation of the chirp signal of detected GWs.



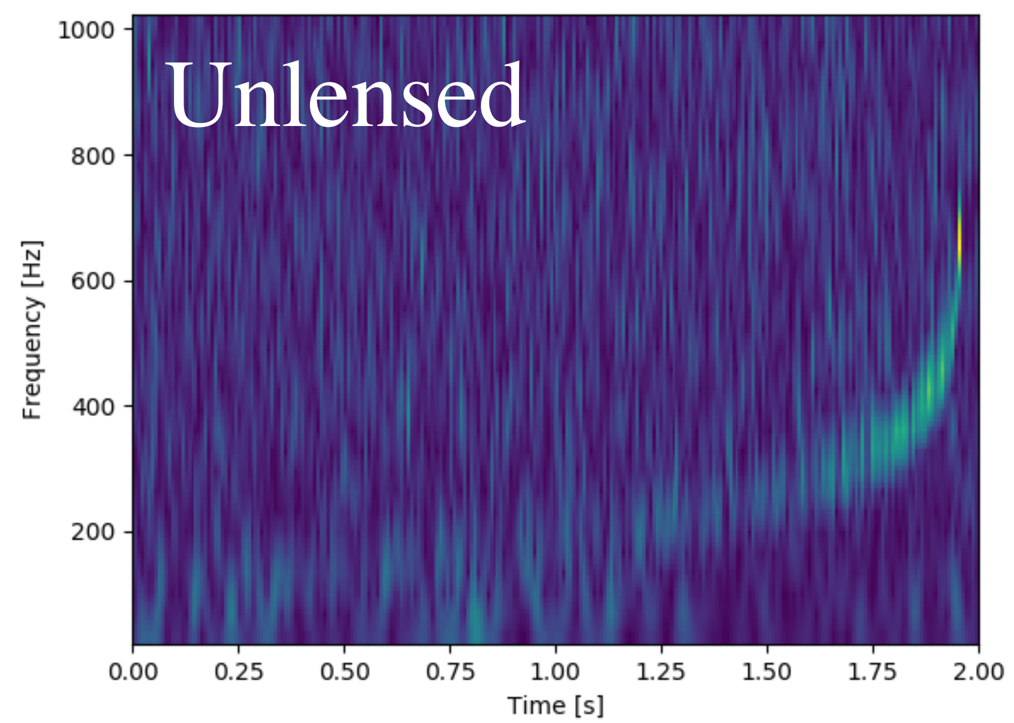
[from PyCBC documentation]

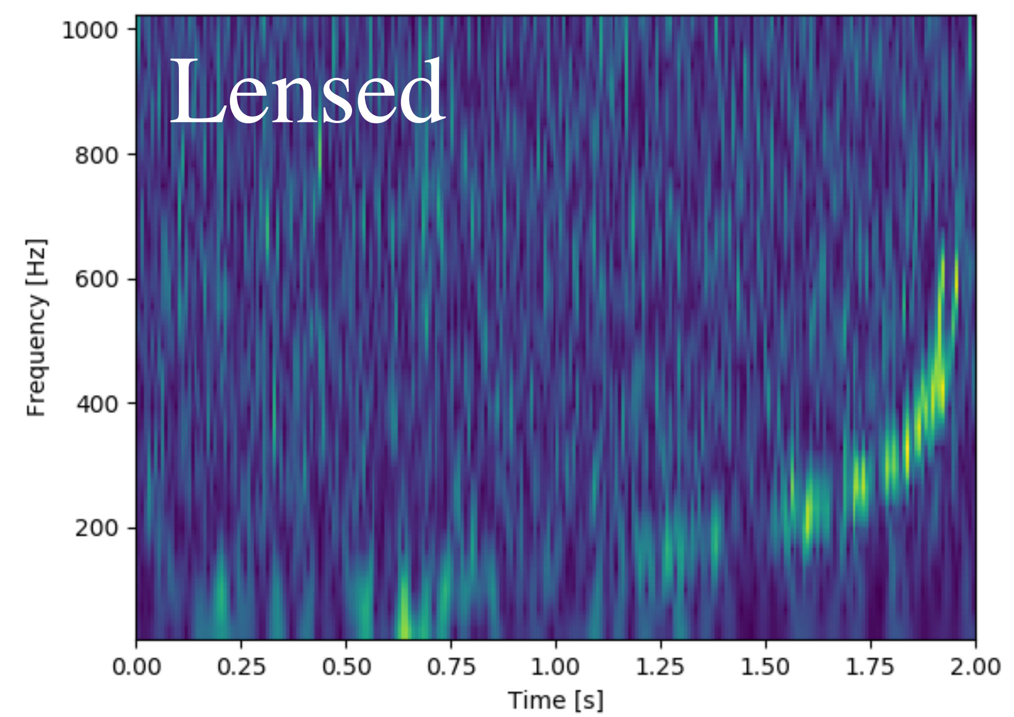


[from PRL **116**, 061102 (2016)]

Classification with Deep Learning w/ TensorFlow [Kim et al., *in progress*]

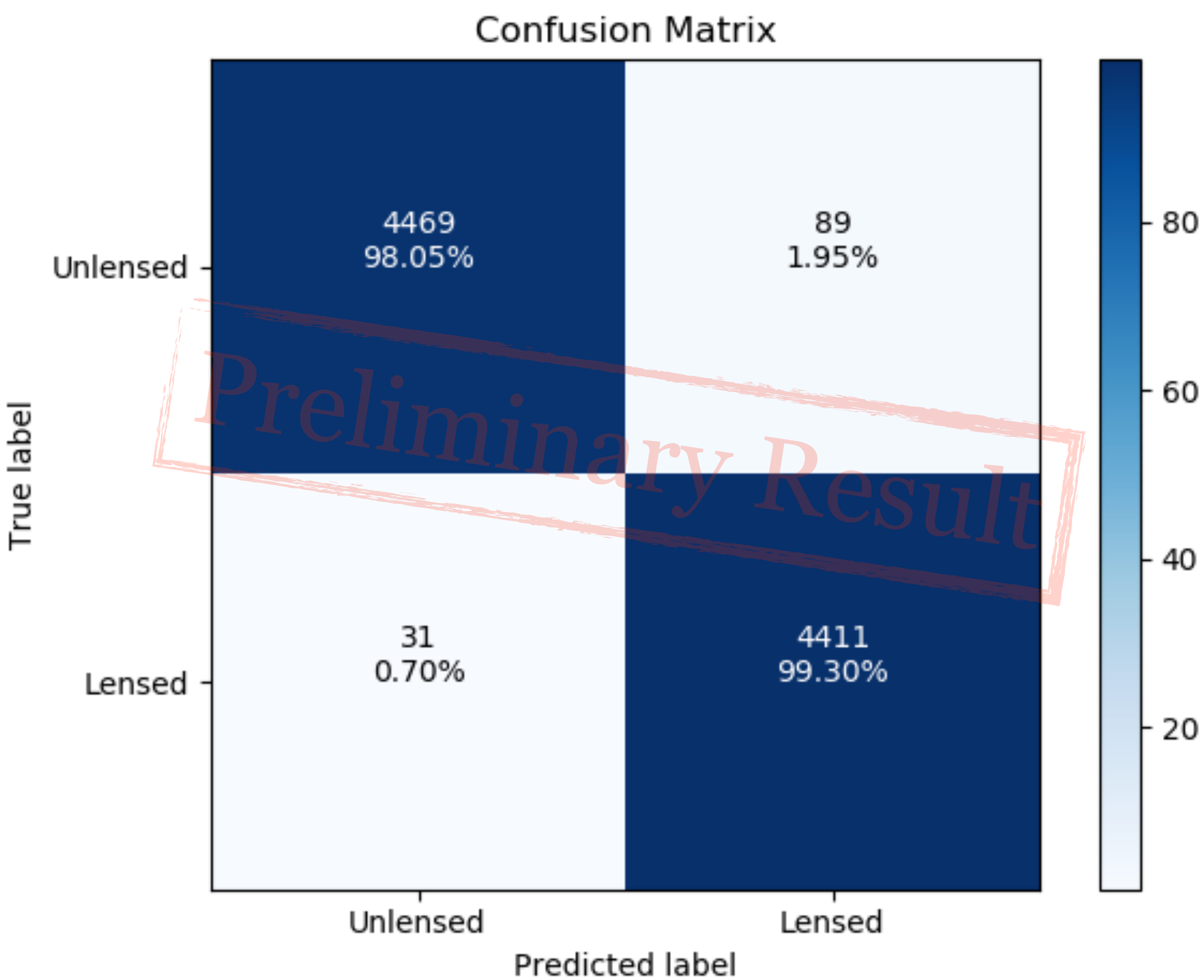
- Input data: 2-dim spectrogram using Qscan
  - # of samples: 45,000 (Training/Validation/Test: 27,000/9,000/9,000)
  - PM only
  - Parameters
    - m_1, m_2: 5 - 55 solar mass
    - D_L: 10 - 1000 Mpc
    - D_{LS}: 10 - 1000 Mpc
    - M_L: 10^3 - 10^5 solar mass
    - ε: 10^{-6} - 0.5 pc
    - redshift factor, z: $H_0 \times D$ / c (H_0 = 70 km/s/Mpc)
  - Noise: aLIGOZeroDetHighPower
    - $10 \leqq \mathrm{SNR} \leqq 50$ (c.f. ≤ 23.6 for O1&O2 BBHs in GWTC-1)

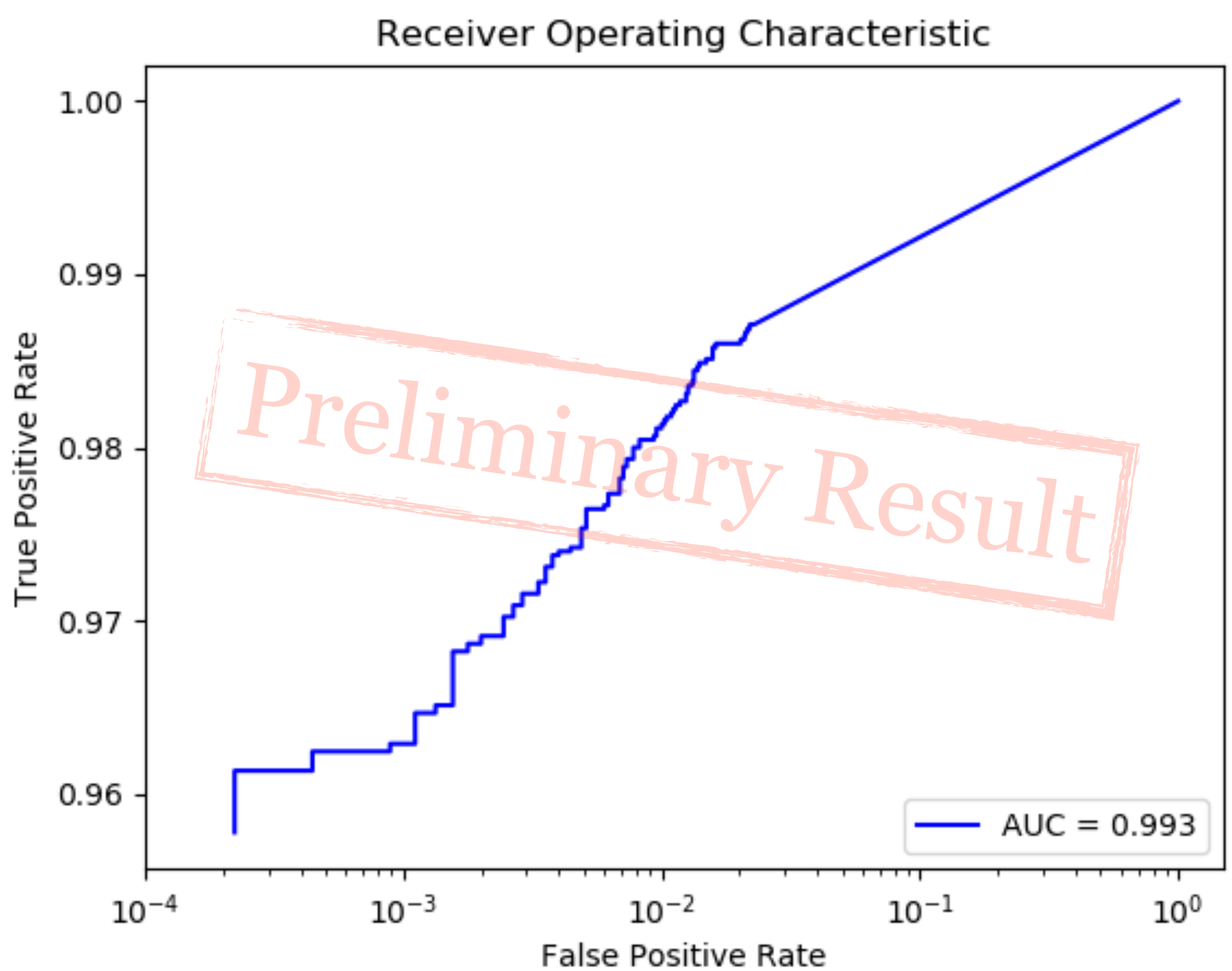




Classification Performance [Kim et al., *in progress*]

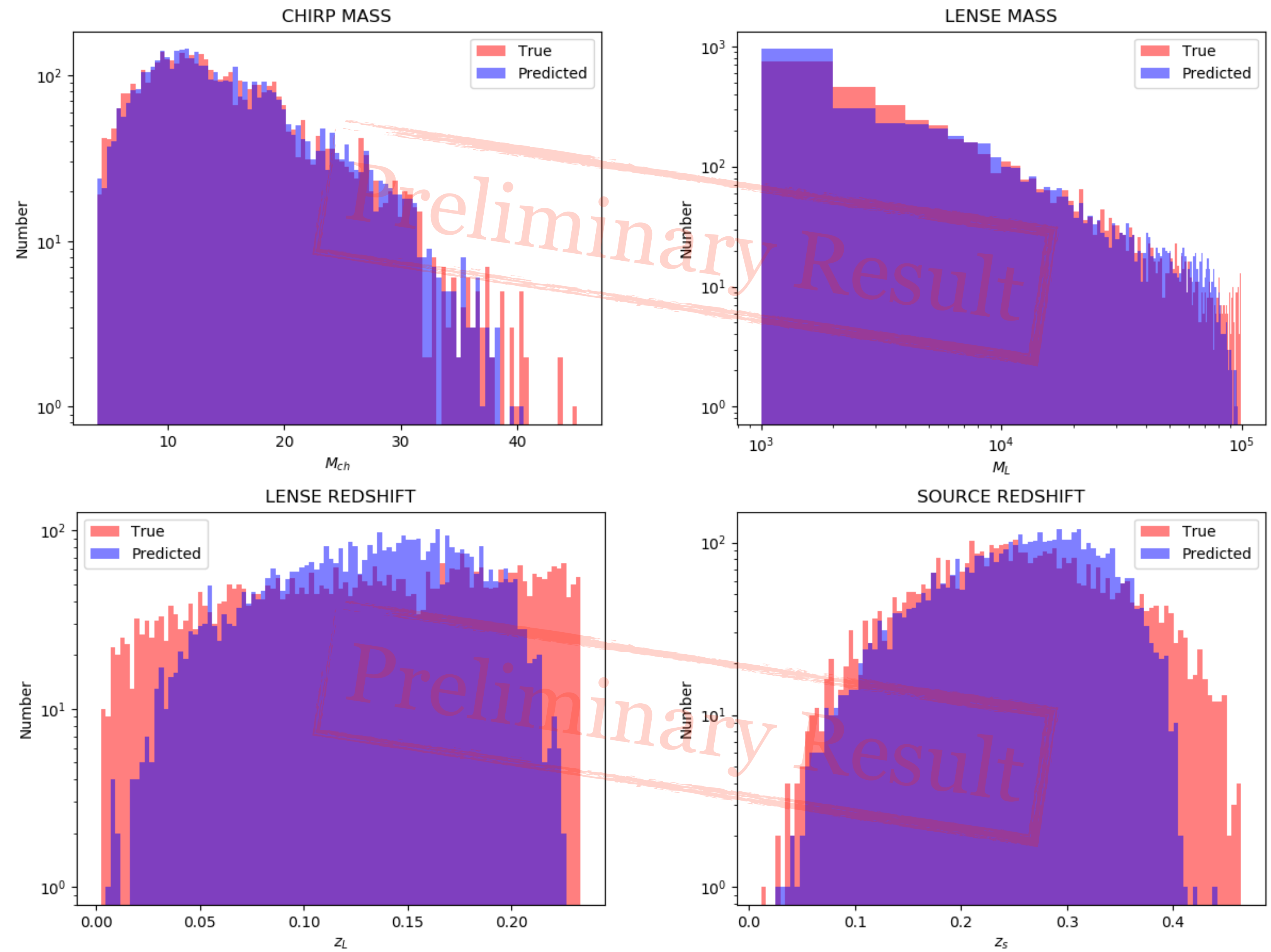
- DL model: VGG11
  - Activation function between hidden layers: Rectified Linear Unit (ReLU)
  - Output activation function: Sigmoid
  - Loss function for error measurement: Mean-Squared-Error





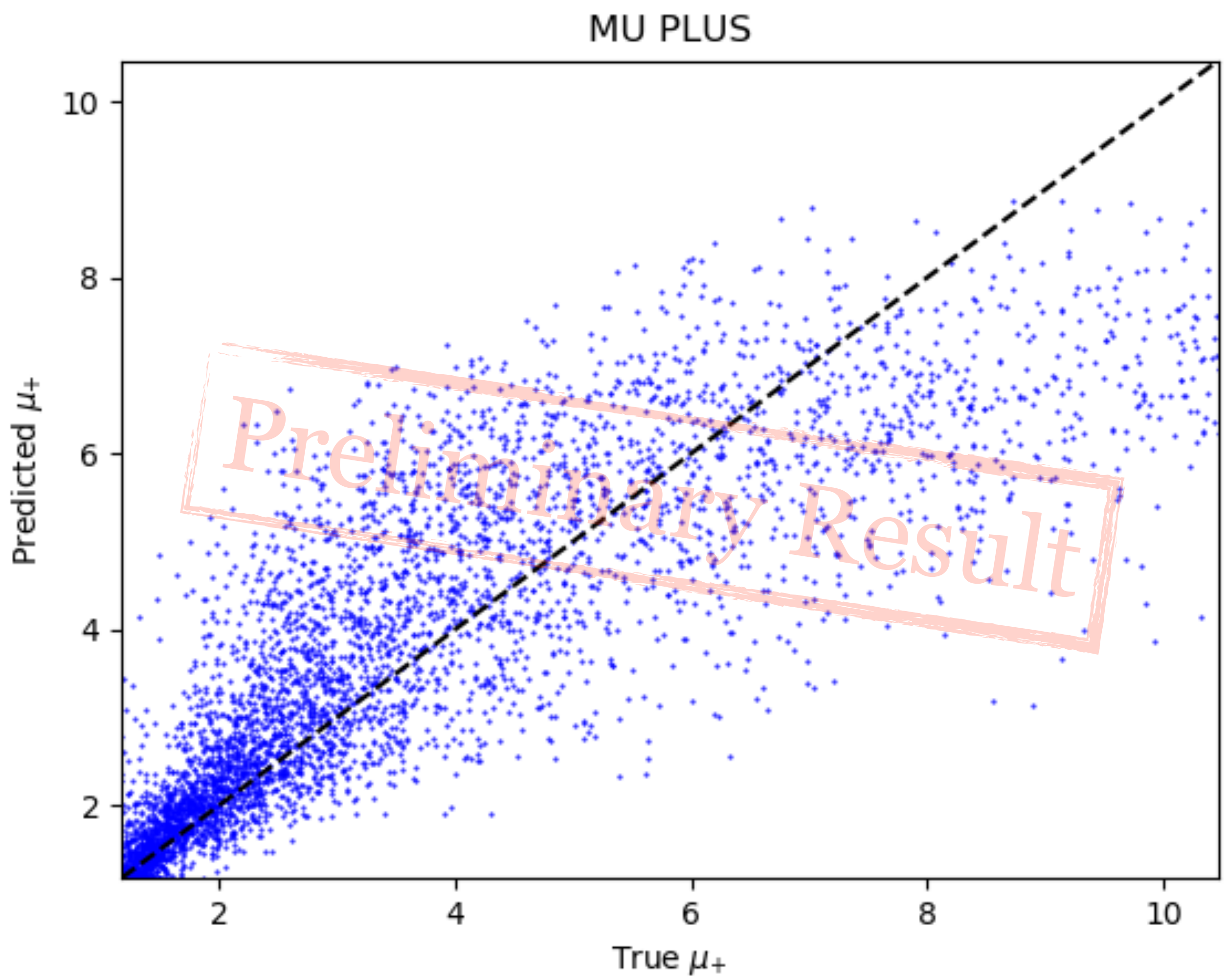
Regression with Deep Learning [Kim et al., *in progress*]

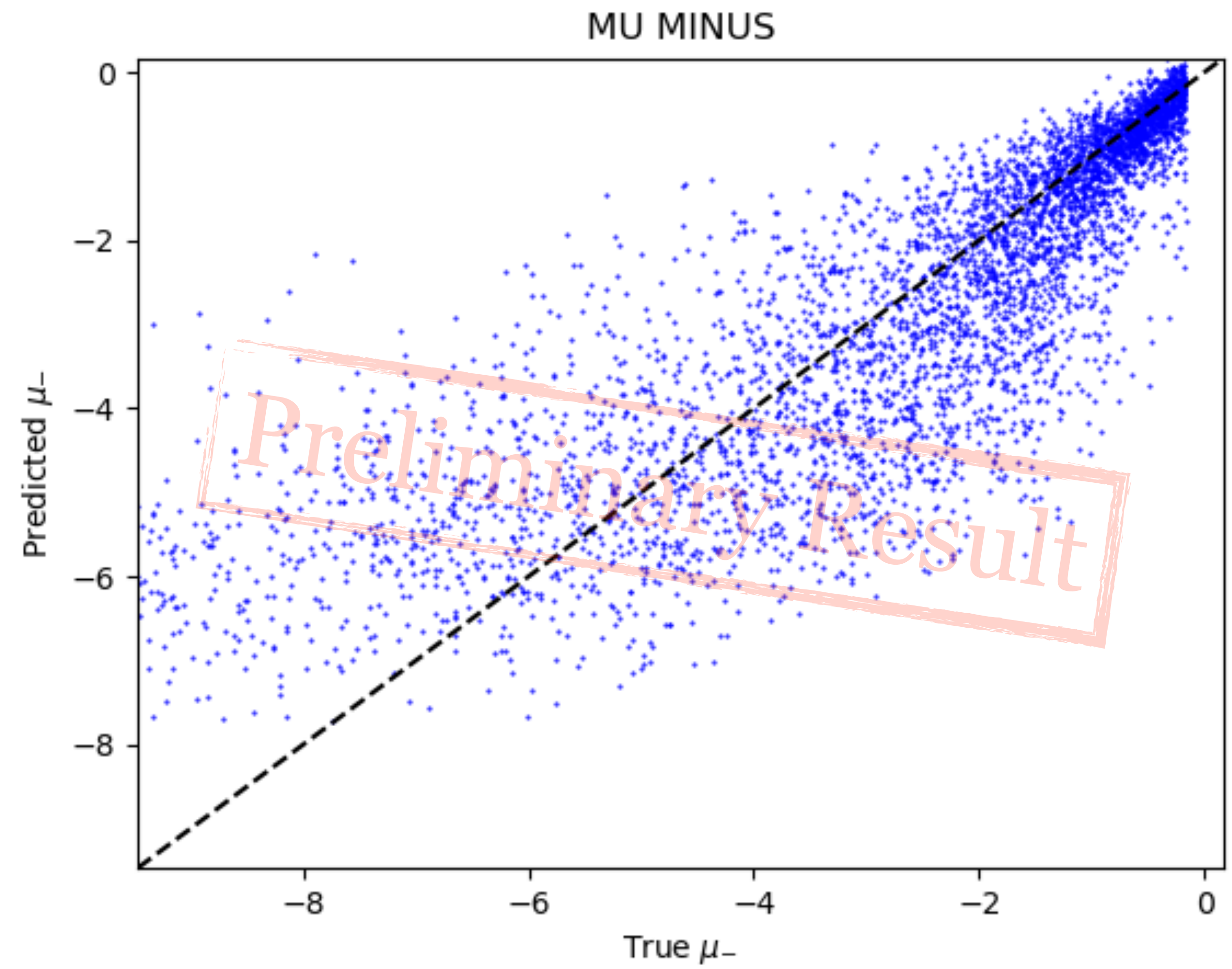
- Input data
  - The same 2-dim Qscan spectrograms used in classification
- DL model: VGG11 (the same with classification)
  - Activation function: ReLU (for both hidden layers and output layer)
  - Loss function: Smooth L1 loss



Comparison between True and Predicted Parameters [Kim et al., *in progress*]

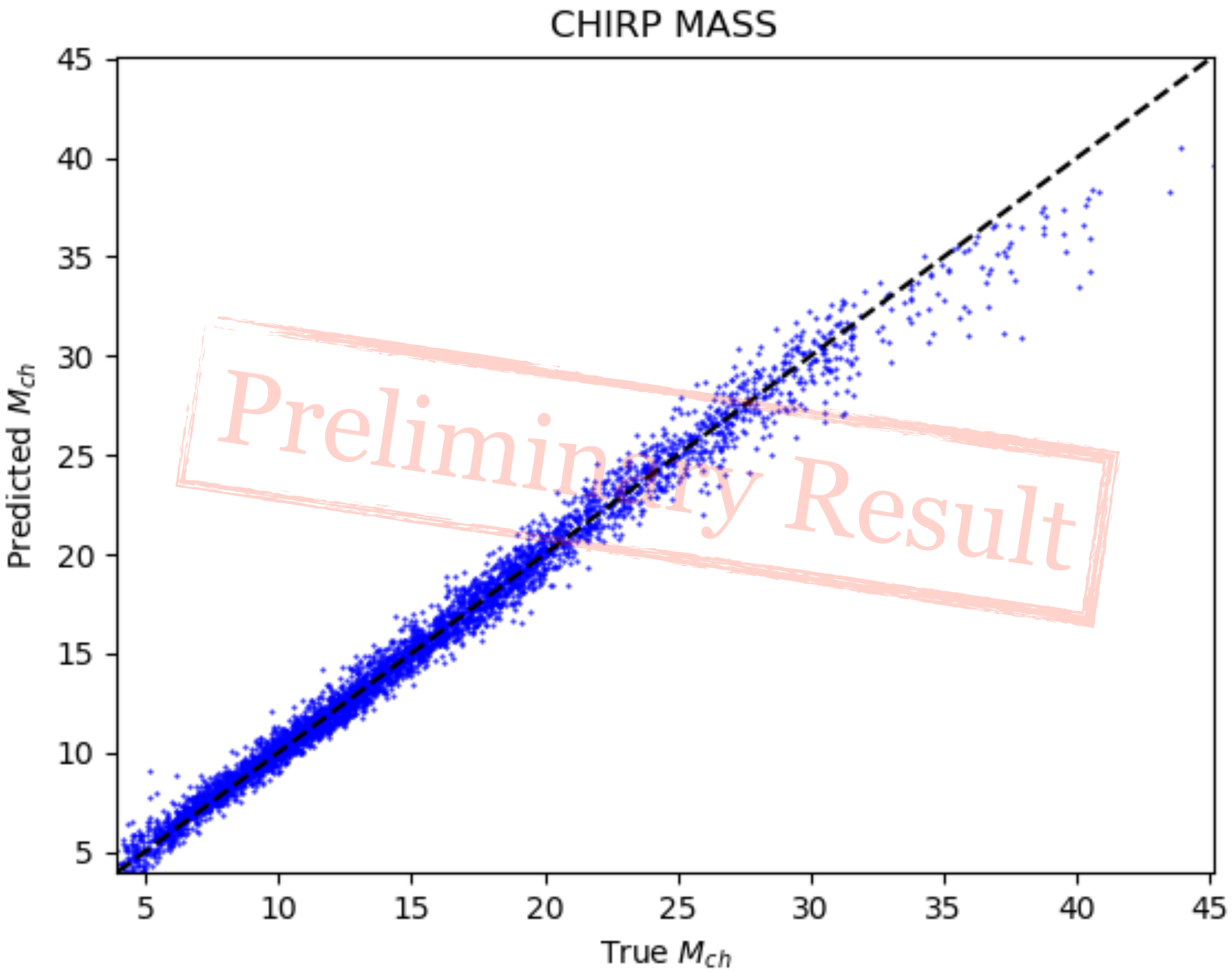
- Lensing factors

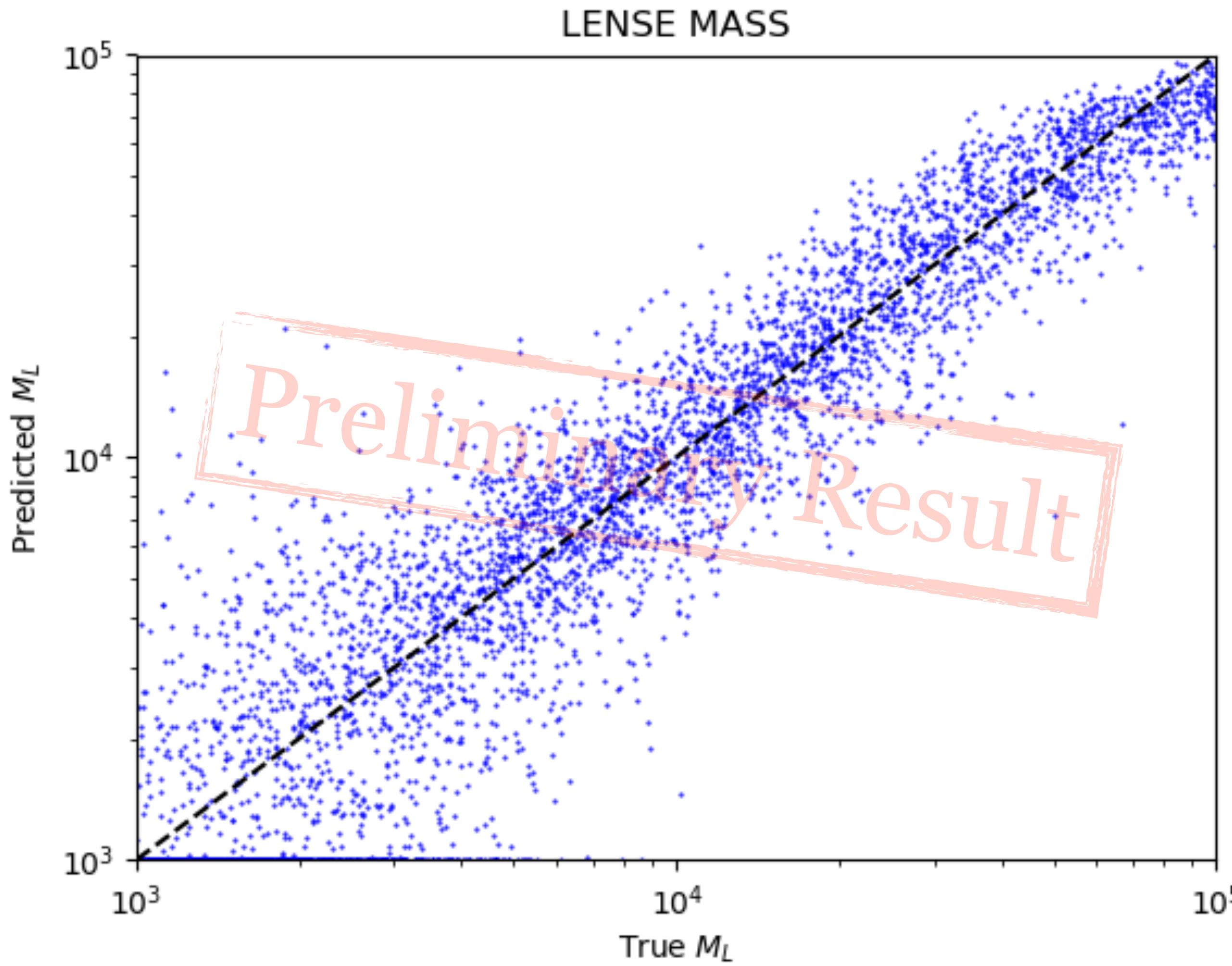




Comparison between True and Predicted Parameters [Kim et al., *in progress*]

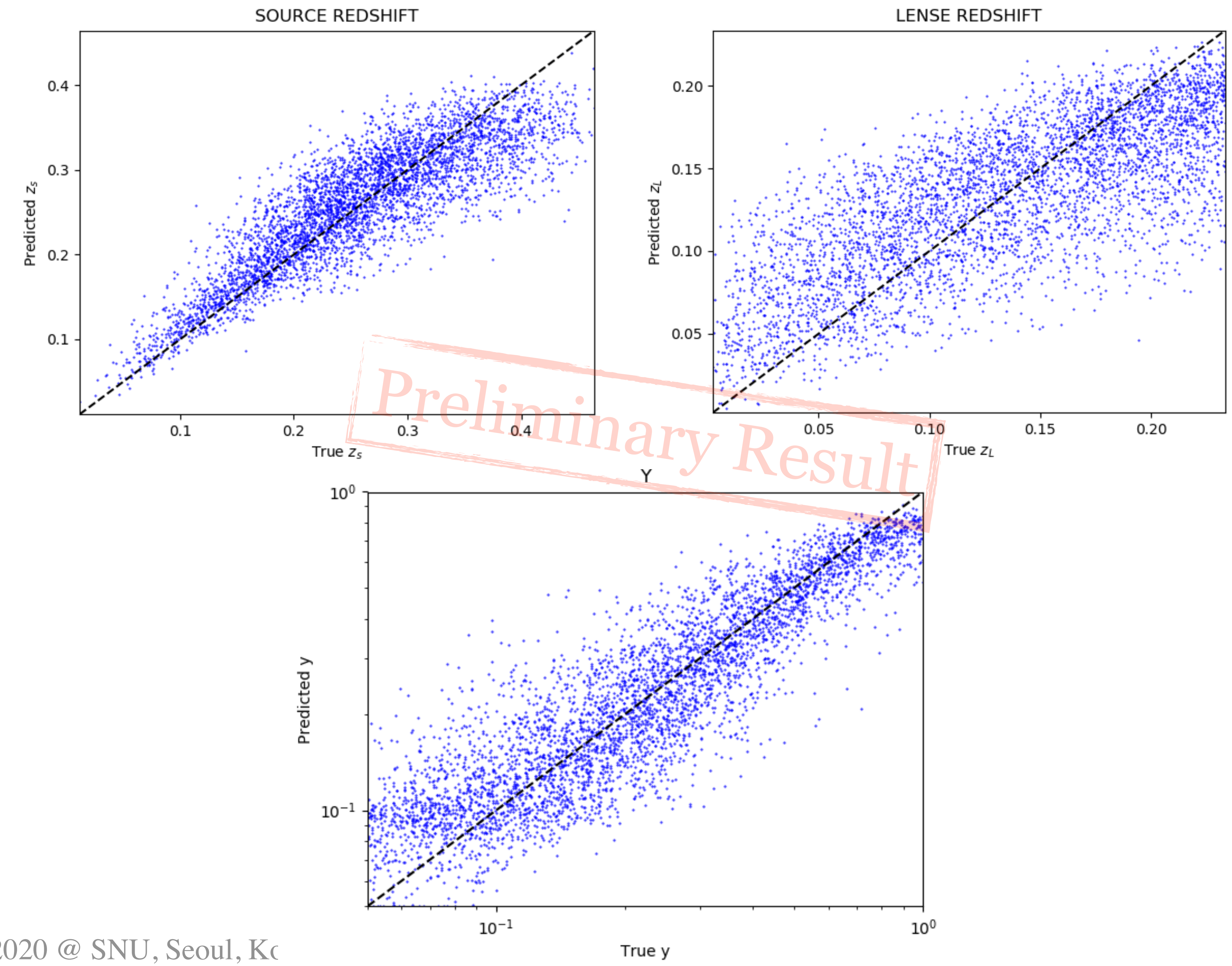
- Masses





Comparison between True and Predicted Parameters [Kim et al., *in progress*]

- Distances and Location



Summary

- DL is good at classification too.
- Seems possible to estimate lensing parameters from regression with DL.
  - Still needs to increase estimation power for poorly estimated parameters.
  - May needs to compare the estimation performance to a popular statistical method.
- Directions of future work
  - Tests SIS model too.
  - Studies the feasibility in wave optics.
  - Explores different physical parameter space with regarding astrophysical objects.
  - Builds a mature end-to-end framework from classification to regression.

Thank you! :)

Question?