MLody - Deep Learning Generated Polarized Synchrotron Coefficients

J. Davelaar [iD][1, 2, 3, 4]

[1] Department of Astrophysical Sciences, Peyton Hall, Princeton University, Princeton, NJ 08544, USA
[2] NASA Hubble Fellowship Program, Einstein Fellow
[3] Center for Computational Astrophysics, Flatiron Institute, 162 Fifth Avenue, New York, NY 10010, USA
[4] Department of Astronomy and Columbia Astrophysics Laboratory, Columbia University, 550 W 120th St, New York, NY 10027, USA

ABSTRACT

Polarized synchrotron emission is a fundamental process in high-energy astrophysics, particularly in the environments around black holes and pulsars. Accurate modeling of this emission requires precise computation of the emission, absorption, rotation, and conversion coefficients, which are critical for radiative transfer simulations. Traditionally, these coefficients are derived using fit functions based on precomputed ground truth values. However, these fit functions often lack accuracy, particularly in specific plasma conditions not well represented in the datasets used to generate them. In this work, we introduce MLody, a deep neural network designed to compute polarized synchrotron coefficients with high accuracy across a wide range of plasma parameters. We demonstrate MLody's capabilities by integrating it with a radiative transfer code to generate synthetic polarized synchrotron images for an accreting black hole simulation. Our results reveal significant differences, up to a factor of two, in both linear and circular polarization compared to traditional methods. These differences could have important implications for parameter estimation in Event Horizon Telescope observations, suggesting that MLody could enhance the accuracy of future astrophysical analyses.

Keywords: Radiation transfer – Machine Learning – synchrotron emission – plasma physics – black holes

1. INTRODUCTION

Polarized synchrotron emission is a cornerstone of high-energy astrophysics, providing critical insights into some of the universe?s most extreme environments, such as the regions around black holes and pulsars. This type of emission is generated when charged particles undergo an acceleration perpendicular to their motion. High-energy astrophysical sources produce synchrotron emission via electrons gyrating around magnetic field lines. A key characteristic of synchrotron emission is its highly polarized nature (Rybicki & Lightman 1979), making it a widely used tool for probing magnetic field structures and understanding the dynamics of these extreme astrophysical systems.

Corresponding author: J. Davelaar
jdavelaar@princeton.edu

The Event Horizon Telescope (EHT) produced the first historic pictures of black holes, either in the center of Messier 87 (Event Horizon Telescope Collaboration et al. 2019a) or from the center of our own galaxy, Sagittarius A* (SgrA*) (Event Horizon Telescope Collaboration et al. 2022a). These observations, conducted at millimeter wavelengths, revealed polarized synchrotron emission originating from a relativistic population of electrons within the accretion flow surrounding the black hole. In the years following these historic results, the EHT also analyzed the polarization signatures coming from both Messier 87 (Event Horizon Telescope Collaboration et al. 2021, 2023) and SgrA* (Event Horizon Telescope Collaboration et al. 2024a). These polarization studies provide critical insights into the magnetic field structures near event horizons, which are essential for understanding the dynamics of accretion flows and jet formation.

For the interpretation of the observations, the EHT uses general relativistic magnetohydrodynamical simulations of the accretion flow, which are post-processed by general relativistic ray tracing codes to compute synthetic images and spectra (Event Horizon Telescope Collaboration et al. 2019b, 2022b, 2024b). These radiative transfer codes generally all follow a similar algorithmic design (Dexter 2016; Mościbrodzka et al. 2017; Bronzwaer et al. 2018, 2020). In this approach, the code defines a virtual camera positioned outside of the simulations consisting of a predefined amount of pixels; each pixel is assigned a wave vector, which is used as the initial condition for solving the geodesic equation. The resulting trajectories (or rays) are then used to solve the radiation transfer equation

$$\frac{\mathrm{d}\boldsymbol{\mathcal{S}}}{\mathrm{d}\lambda} = \begin{pmatrix} j_I \\ j_Q \\ j_U \\ j_V \end{pmatrix} - \begin{pmatrix} \alpha_I & \alpha_Q & \alpha_U & \alpha_V \\ \alpha_Q & \alpha_I & \rho_V & -\rho_U \\ \alpha_U & -\rho_V & \alpha_I & \rho_Q \\ \alpha_V & \rho_U & -\rho_Q & \alpha_I \end{pmatrix} \begin{pmatrix} \mathcal{I} \\ \mathcal{Q} \\ \mathcal{U} \\ \mathcal{V} \end{pmatrix}. \quad (1)$$

The left hand side of Eqn. 1, contains; λ the affine parameter, and $\boldsymbol{\mathcal{S}}$ the Lorentz invariant Stokes vector given by

$$\boldsymbol{\mathcal{S}} = \frac{\mathbf{S}_\nu}{\nu^3} = \frac{1}{\nu^3} \begin{pmatrix} I_\nu \\ Q_\nu \\ U_\nu \\ V_\nu \end{pmatrix} \quad (2)$$

here S_ν is the specific intensity for every Stokes component I, Q, U, V, and ν is the frequency. The right-hand side of Eqn. 1 depends on the polarized synchrotron coefficients, first the emission coefficients j_I, j_Q, j_U, j_V, secondly absorption coefficients $\alpha_I, \alpha_Q, \alpha_U, \alpha_V$, and finally the rotation and conversion coefficients ρ_Q, ρ_U, ρ_V. Polarized radiative-transfer computations depend on the orientation of the frame in which the Stokes vector can be defined. This requires constructing a frame at the observer's location (the camera) and any location where the ray interacts with radiating matter. A suitable choice for the latter is one where Stokes U is aligned with the magnetic field vector, resulting in $j_U = \alpha_U = \rho_U = 0$, and $j_Q < 0$.

At every time step along the ray, the coefficients are computed based on the local plasma conditions. Calculating these coefficients involves an integral over momentum space and a sum over the synchrotron harmonics, see, e.g., Leung et al. (2011); Dexter (2016). The integral over momentum space depends on the shape of the electron distribution function, which gives the particle number at a given energy (or momentum). Com-

mon choices for the electron distribution function include the relativistic Maxwellian, κ-distribution functions, and power-law distribution functions. Computing this integral accurately over the full range of possible plasma conditions is computationally very demanding. Therefore, many forms of approximate "fit formula" are used instead of performing the full complex computation for every step. Common fit formulas in the literature that are tested against the exact numerical solution are Shcherbakov (2008); Leung et al. (2011); Dexter (2016); Pandya et al. (2016); Marszewski et al. (2021). The weakness of this approach is that fit functions are by construction approximate and are only constructed using a limited range of the parameter domain, e.g., Pandya et al. (2016)[1] only considered dimensionless electron temperatures between 2 and 40, which limits their accuracy in regions where absorption and Faraday rotation dominate at lower temperatures. Additionally in the ultra-relativistic synchrotron limit the integral over the distribution function can be analytically approximated, e.g. Mahadevan et al. (1996); Dexter (2016).

To address the limitations of traditional fit functions, we developed MLody, a deep neural network designed to enhance the accuracy of polarized synchrotron coefficient computations across a diverse range of plasma conditions, extending high accuracy to the lower temperature regime. The code is publicly availbe on GitHub[2]. In Section 2, we will outline the training data generation, the network architecture, and how we interfaced MLody with RAPTOR[3] (Bronzwaer et al. 2018, 2020). In Section 3, we will show the network's training results, and present a use case of MLody's capabilities by computing synthetic images of black hole accretion flow simulations. In Section 4 we will draw our conclusions.

2. METHODS

2.1. *Training Data Generation*

We used the publicly available code symphony to generate a large sample of polarized synchrotron radiation coefficients. For a detailed explanation of symphony, see Pandya et al. (2016) and references therein. The coefficients generally depend on four variables: the electron number density n_e, the frequency ν, the magnetic field strength, B, and the angle between the wave vector k and magnetic field, referred to as the pitch angle, θ_B. Additionally, certain variables are connected to the spe-

---

[1] We will compare our results mainly to the work by Pandya et al. (2016) since it is the most commonly used fit formula within the Event Horizon Telescope Collaboration.

[2] https://github.com/jordydavelaar/mlody

[3] Publicly available at https://github.com/jordydavelaar/raptor

cific distribution function. For a thermal Maxwellian, this includes the dimensionless electron temperature $\Theta_e = \frac{kT}{m_e c^2}$, with k the Boltzmann constant, m_e the electron mass, and c the speed of light.

Naively this would require a five-dimensional data cube for our input vector. However, by utilizing the exact form of the integrals solved by `symphony`, we can reduce this to three dimensions. First, all coefficients depend linearly on electron number density; therefore, we can always multiply with density after computing the coefficients. Secondly, we can define $X_\nu = \nu/\nu_c$, where $\nu_c = \frac{eB}{4\pi mc}$ is the electron cyclotron frequency, this combines the frequency and magnetic field, resulting in another reduction of our input dimension. The coefficients can then be written in a universal form

$$j_S = \frac{n_e e^2 \nu}{c} J_S(X, \theta_B, \theta_e) \tag{3}$$

$$\alpha_S = \frac{n_e e^2}{\nu_c m_e c} A_S(X_\nu, \theta_B, \Theta_e) \tag{4}$$

$$\rho_S = n_e R_S(X, \theta_B, \theta_e), \tag{5}$$

where J_S and A_S are dimensionless functions. For training data of our network we will computed J_S, A_S and R_S by sampling the three input variables randomly over the following ranges $X_\nu \in [1, 10^6]$, $\theta_B \in [0, 2\pi]$, and $\Theta_e \in [10^{-2}, 10^2]$. In total, the training data includes nearly two million coefficients for each Stokes parameter.

2.2. Network Architecture and Learning Strategy

To build our neural network we use `Keras` (Chollet et al. 2015) with a `TensorFlow` (Abadi et al. 2015) backend. The network architecture is a fully connected deep neural network with nine Dense layers. These layers are divided into three groups, each with 100, 200, and 300 neurons, respectively. For every Dense layer, we use a rectified linear unit (ReLU) activation function, described by $f(x) = \max(0, x)$ (Nair & Hinton 2010). ReLU activation functions are essential for introducing non-linearities into the network, enabling it to learn complex underlying functions. After the last group of Dense Layers, we have our output layer which consists of eight neurons and utilizes a linear activation function. For a summary of our network architecture see Table 1.

We split the data 80/20 between training and validation. Additionally, we keep 20,000 randomly selected samples aside for testing after training is finished. The validation set is used to monitor for overfitting. Overfitting occurs when the network perfectly replicates the training data without learning how to generalize, which would result in poor performance on unseen data. By

Layer	Neurons	Activation
Input	3	-
Dense 1-3	100	ReLU
Dense 4-6	200	ReLU
Dense 7-9	300	ReLU
Output	8	linear

Table 1. Summary of the network architecture. Listed are the various layers used, the number of neurons per layer, and the activation functions.

comparing the loss on the training set with the validation set, we ensure the network is learning the true underlying correlations rather than simply memorizing the training data. If the training loss decreases while the validation loss increases, this indicates overfitting.

We employed the Adam optimizer (Kingma & Ba 2014), and the network uses an exponential-decay learning-rate schedule with an initial learning rate of 10^{-4}, a decay rate of 0.95, and a decay step of 10,000. This means the learning rate decreases by a factor of 0.95 for every 10,000 gradient descent steps. The loss function used was mean squared error:

$$L = \frac{1}{N} \Sigma_i^N |y_i - \hat{y}_i|^2, \tag{6}$$

where y_i is the true value of the i'th sample, $\hat{y}_i$ is corresponding network prediction, and N is the total number of data points. The network was trained for 1,000 epochs and used a batch size of 2,000.

Before training, the input vector X and output $\hat{y}$ are transformed and rescaled so that their values map between -1 and 1 and their mean is zero. In the case of X_ν, Θ_e, J_S, A_S, we 1) take the logarithm, 2) recenter them to have a mean of zero, and 3) normalize with their maximum value. For θ_B, we train on $\cos\theta_B$, which is naturally normalized between $[-1, 1]$. Since ρ_Q and ρ_V, both have positive as well as negative values, we did not use a logarithmic function; instead, their scaling was performed by first applying $X = \text{sign}(\rho_S)|\rho_S|^{0.25}$, then recentering and normalizing by the maximum value.

2.3. Integration with RAPTOR

After the network was successfully trained, we interfaced `MLody` with `RAPTOR` by using the publicly available `keras2cpp` [4] library. This library is a `C++` framework that reads in a model file that contains all the information on the network architecture and the neuron weights. The library also provides functions that can be called by `RAPTOR` to compute new outputs. For this final step, the

---

[4] See https://github.com/gosha20777/keras2cpp
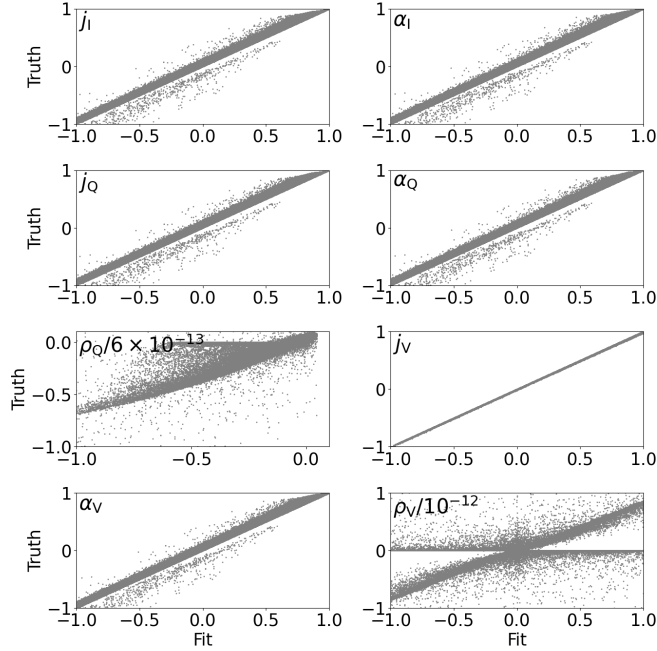
Figure 1. The ground truth values of `MLody` are plotted as a function of their corresponding fit function values. A perfect diagonal would mean identical values for both methods. The largest discrepancies are found for the rotation and conversion coefficients, ρ_Q and ρ_V.

code converts the plasma quantities to the input vector format of `MLody`, obtains an output vector, and computes J_S, A_S and R_S by rescaling the output and add units via Equations 3-5. In case the input variables for `MLody` are outside the range of the training data, we default to using the fit functions.

3. RESULTS

3.1. *Network Training and Performance Assessment*

The training data set, computed with `symphony`, consists of nearly two million coefficients for each Stokes parameter. To assess the performance of the training data against the fit formula from Pandya et al. (2016), we compared the results using both approaches, as illustrated in Fig. 1. In an ideal scenario, all values would align along the diagonal, indicating perfect agreement between the fit function on the x axis and the `symphony` output on the y axis. The emission and absorption coefficients all show a wider band around the optimal diagonal, with the largest discrepancies at low X_ν and electron temperatures, which are outside the data range on which the fit functions were constructed. Moreover, the rotation and conversion coefficients display significant deviations across the entire parameter range, including incorrect signs.
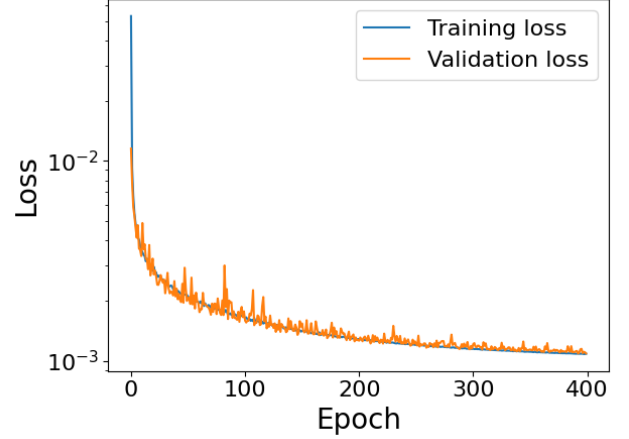

Figure 2. Loss as a function of epoch for the training and validation set. Both sets perform equally well, which is a clear indicator that we do not overfit the training data.

Next, we train the network for 1,000 epochs and evaluate its performance. Fig. 2 shows the training and validation Loss as a function epoch. Both losses steadily decrease with epoch and follow similar trajectories, indicating good network performance and a lack of overfitting. At the end of the training, the Loss reaches a value of 5×10^{-4}. We additionally also compute the mean absolute relative error of the network, and for both the training as well as the validation set, this is below one percent at the end of training.

To further evaluate `MLody`?s performance, we compared the network?s predictions to the ground truth from `symphony` in Fig. 3 by using the 20,000 samples in the test data set. Compared to Fig. 1, `MLody` outperforms the traditional fit functions across all coefficients. The most notable improvements can be seen in the rotation and conversion coefficients ρ_Q and ρ_V.

3.2. *Performance Comparison for Black Hole Accretion Modeling*

As an example of `MLody`'s capabilities, we integrated the network with `RAPTOR` to compute synthetic images of a black hole accretion flow simulation. We use a simulation snapshot run with the GRMHD code `BHAC` (Porth et al. 2017). The black hole simulation is in the Magnetically Arrested Disk (MAD) regime Narayan et al. (2003); Tchekhovskoy et al. (2011); see Davelaar et al. (2023) for simulation details. Meaning that the black hole reaches a saturated state in the amount of magnetic flux on the horizon. The spin of the black hole is $a = 0.9375$. Ideal GRMHD simulations are scale-free and typically do not evolve the electron thermodynamics explicitly. Therefore, plasma variables must be rescaled to achieve a physical scale in post-processing.
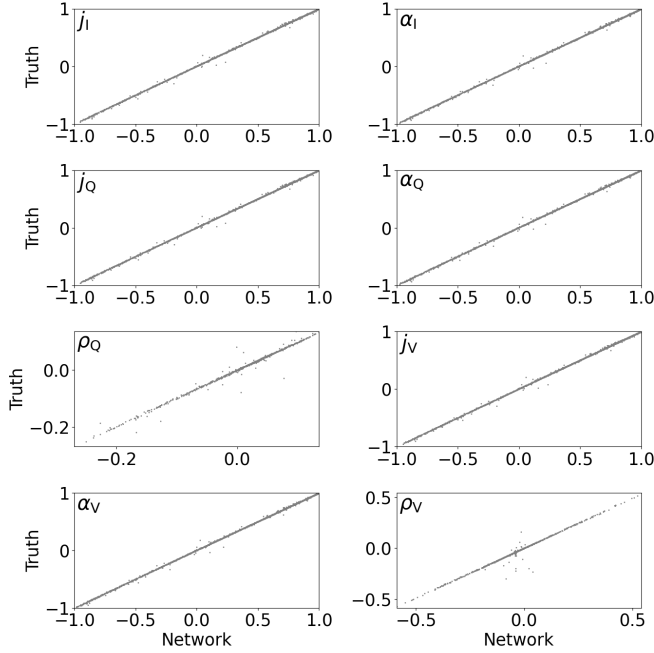
Figure 3. The same as Fig. 1 but now as a function of the predictions of `MLody`. Compared to Fig. 1 the performance is much higher, especially for the conversion and rotation coefficients.

For this test, we chose scaling parameters to model the black hole at the center of our galaxy, Sagittarius A*. The black hole mass and distance are constrained by long-term observations of stellar orbits in the galactic nucleus, $M_{\mathrm{BH}} = 4.14 \times 10^6 M_\odot$, $d = 8.127$ kpc (Do et al. 2019; GRAVITY Collaboration et al. 2019). The mass accretion rate we set to $\dot{M} = 10^{-8} M_\odot \mathrm{yr}^{-1}$, resulting ≈ 1 Jy of flux at 230 GHz.

Finally, the electron temperature was parameterized using the prescription from Mościbrodzka et al. (2016) which relates the electron temperature T_{e} to the proton temperatures T_{p} via

$$T_{\mathrm{ratio}} = \frac{1}{1 + \beta_{\mathrm{p}}^2} + R\frac{\beta_{\mathrm{p}}^2}{1 + \beta_{\mathrm{p}}^2}, \tag{7a}$$

$$\Theta_{\mathrm{e}} = k_{\mathrm{b}} T_{\mathrm{e}}/m_{\mathrm{e}} c^2 = \frac{U(\gamma_{\mathrm{ad}} - 1) m_{\mathrm{p}}/m_{\mathrm{e}}}{\rho\,(1 + T_{\mathrm{ratio}})}, \tag{7b}$$

where m_{p} the proton mass, m_{e} the electron mass, U the internal energy, and Θ_{e} the dimensionless electron temperature. The free parameter R determines the temperature ratio in regions where $\beta_{\mathrm{p}} = 2P/B^2 \gg 1$. The choice of R will change the morphology of the resulting image, e.g., the disk dominates the emission for $R = 1$ while the jet dominates if $R \gg 1$. For our test case, we will set $R = 40$, meaning a jet-dominated model. Finally, we set the inclination of the observer relative to
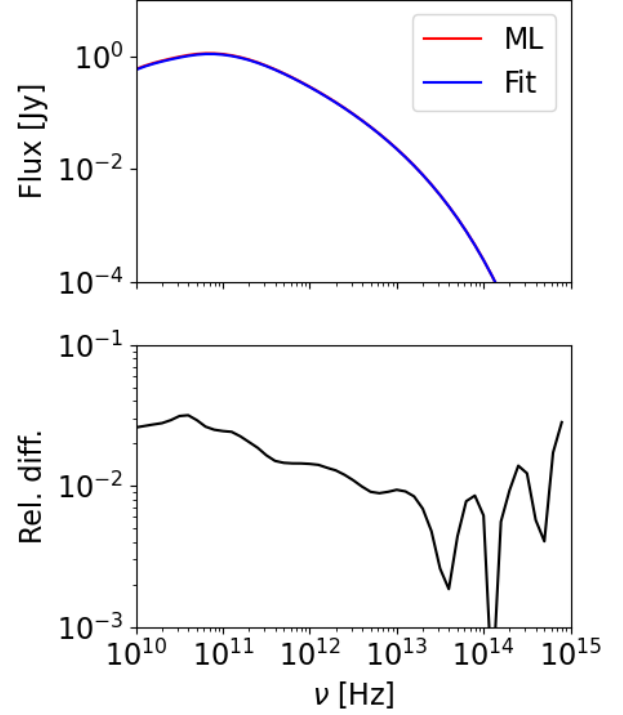

Figure 4. Spectrum of Stokes I for both the `MLody` generated coefficients as well as the fit formula (top panel). The bottom panel shows the relative difference between the two spectra. The overall difference is small, and nearly independent of frequency.

the black hole spin axis at 30 degrees. We then generate synthetic polarized spectra and images between $10^{10} - 10^{14}$ Hz using both `MLody` and the fit function from Pandya et al. (2016).

We computed spectra for all Stokes variables and compared them to results obtained using the fit function method. In Fig. 4 we show the spectra for Stokes I. Overall, we found good agreement between both methods, with relative differences remaining close to one percent across the entire frequency range, as shown in the bottom panel. This small discrepancy is not unexpected given that there is a relatively small difference between the ground truth and the fit formula for Stokes I (see Fig. 1) in contrast to the rotation and conversion coefficients which only affect Stokes Q, U and V.

Given the poorer performance of the fit function for the rotation and conversion coefficients, we expect larger discrepancies in the polarized quantities. We computed both image-integrated and pixel-integrated values of linear and circular polarization. The difference lies in when the summation over pixel values occurs: we either sum Stokes Q and U first and then compute the total linear polarization $\sqrt{Q^2 + U^2}$, or we compute $\sqrt{Q^2 + U^2}$ per
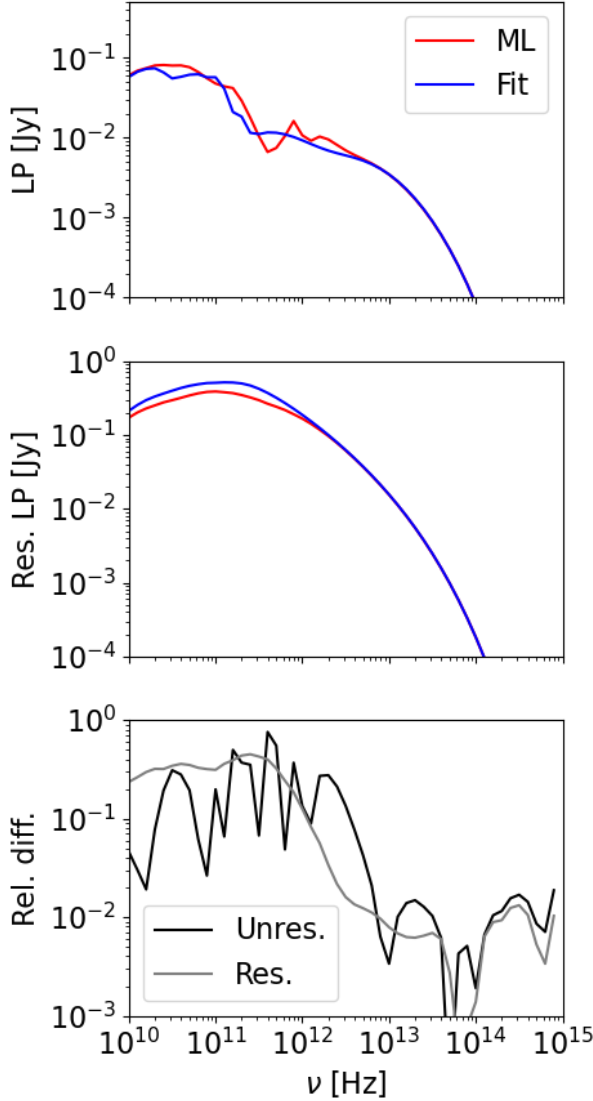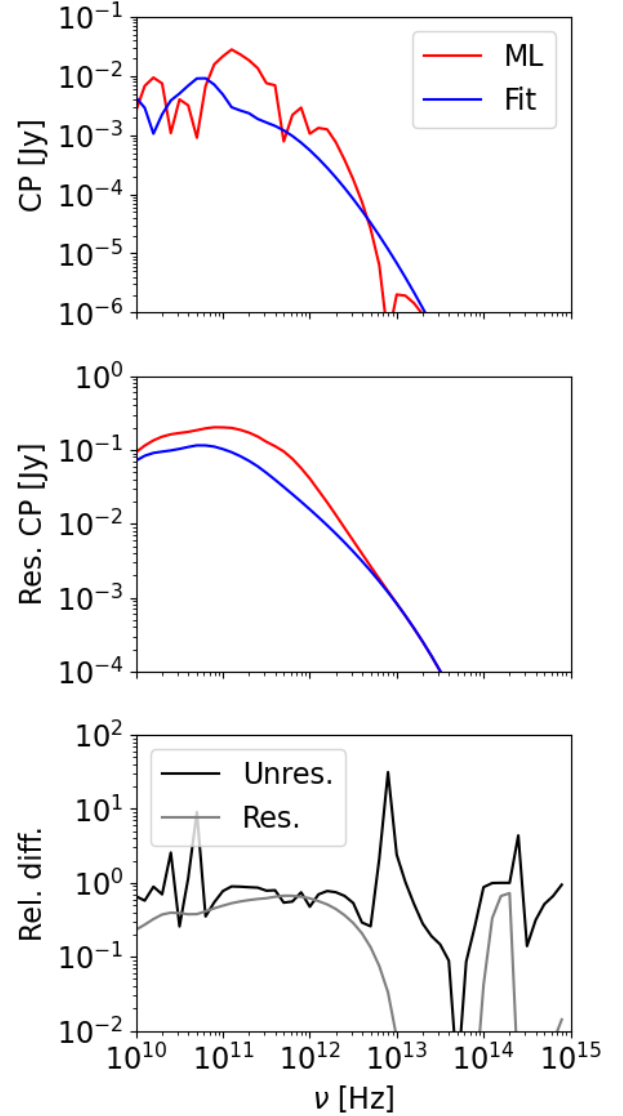
Figure 5. Same as Fig. 4 but now for linear polarization, $\sqrt{S_Q^2 + S_U^2}$. Here, we find larger relative differences compared to Stokes I, up to 10-30%, peaking at 10^{11} Hz, and 1% at higher frequencies.

Figure 6. Same as Fig. 4 but now for circular polarization, $|S_V^2|$. Compared to linear polarization the relative difference is comparable in the majority of the spectrum, but includes outliers to large values, due to a difference in overall shape at higher frequencies ($\nu > 10^{13}$ Hz.).

pixel and then the sum. The former approach mimics an under-resolved measurement, such as one from a single telescope with a large field of view, while the latter represents a perfectly resolved measurement.

Fig. 5 shows the linear polarization $\sqrt{Q^2 + U^2}$ for both unresolved (top panel) and resolved (middle panel) cases, with the relative difference between the two methods shown in the bottom panel. In both cases, we find maximum differences of 10%, where MLody generates less linear polarization than the fit functions.

Fig. 6 presents the circular polarization results, $|V|$. The top panel shows the unresolved circular polariza-

tion, the middle panel the resolved circular polarization, and the bottom panel the relative difference between the two models. The relative difference with the fit function is largest for Stokes V, reaching up to 100% for most frequencies, with MLody generating more circular polarization than the fit functions.

In Fig.7, we show the pixel-to-pixel residuals between synthetic images computed with and without MLody at $\nu = 10^{10}, 10^{11}, 10^{12}, 10^{13}, 10^{14}$ Hz. Each image is normalized to its largest pixel value, with an inset showing the absolute error divided by the maximum pixel value
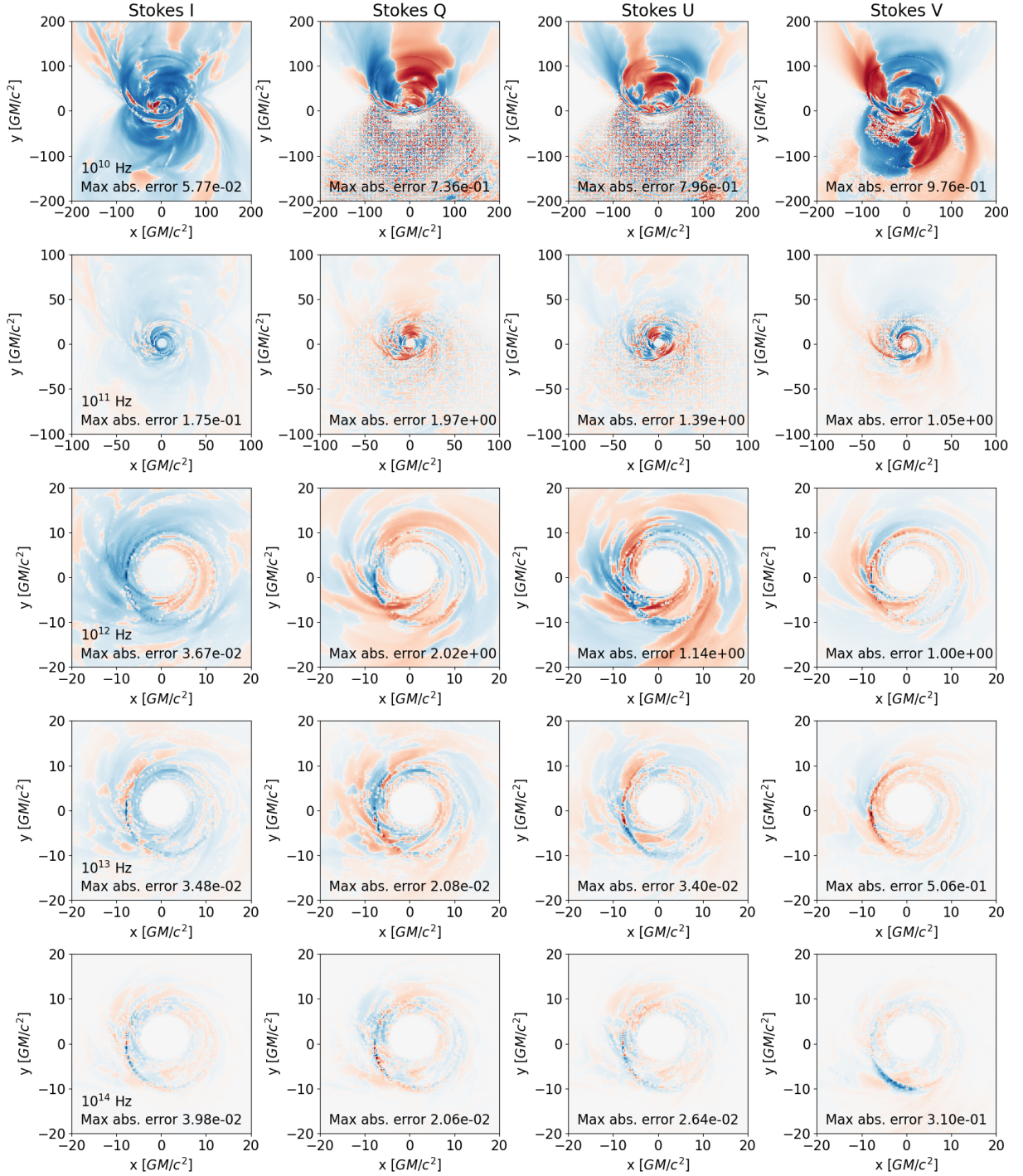
Figure 7. Pixel-to-pixel difference maps on synthetic polarized synchrotron images either computed with `MLody` or fit functions. The first, second, third, and fourth columns are Stokes I, Q, U, and V, respectively. Rows correspond to difference frequencies, top to bottom; $\nu = 10^{10}, 10^{11}, 10^{12}, 10^{13}, 10^{14}$ Hz. Each image contains the maximum difference normalized by the maximum pixel value. The largest differences are found for Stokes Q, U, and V at frequencies $\nu = 10^{11} - 10^{12}$ Hz.

of the image. The first column shows residual images for Stokes I. As expected from the spectra, the residuals are small, generally below 0.1. For Stokes Q, U, and V, larger differences are observed. This is in agreement with the larger relative differences found in the linear and circular polarization spectra, see Figs. 5 and 6. The maximum differences are found around 10^{12} Hz and decrease at higher frequencies. The largest discrepancies are seen either far out in the jet, see 10^{10} Hz, where the temperature and X_ν decrease, or close to the black hole for higher frequencies.

The computational speed of `RAPTOR` is affected by the current implementation of `MLody`, making the code a factor of a few slower compared to when it is run with the fit functions. However, the keras2cpp library used to connect `MLody` with `RAPTOR` is not yet optimized for our specific use case. Improving the computational performance of the network is ongoing, and speed-ups are expected to be needed to make the network more efficient for EHT analysis. However, improvements observed with `MLody` suggest that deep learning techniques could become integral to modeling efforts for the study of black hole environments.

4. CONCLUSION

In this letter, we introduced `MLody`, a novel deep neural network specifically designed to generate polarized synchrotron coefficients with high accuracy, particularly tailored for black hole accretion studies. Trained on millions of coefficients from the publicly available `symphony` code, `MLody` outperforms traditional fit functions across a wide range of plasma conditions. Our results not only demonstrate the increased performance of `MLody` over traditional methods but also highlight its potential to increase the accuracy of the analysis of future EHT observations and related astrophysical research.

This work focused on the thermal Maxwellian electron distribution function, demonstrating significant improvements in the accuracy of polarization modeling. Future developments will extend `MLody` to support κ and power-law distribution functions, as well as other more complex electron distributions, such as anisotropic distribution function (Galishnikova et al. 2023) or distribution function computed by first-principle kinetic plasma simulations. These expansions will further enhance `MLody`?s applicability to a broader range of astrophysical scenarios.

By interfacing `MLody` with the radiative transfer code `RAPTOR`, we generated synthetic images from a black hole accretion simulation and observed discrepancies in polarization spectra?ranging from 10% to 100%?compared to results derived from fit functions. These differences are substantial and could critically impact the parameter estimations used in EHT observations, particularly for M87 and SgrA* (Event Horizon Telescope Collaboration et al. 2021, 2023, 2024a), where models are often validated against specific polarization thresholds. While a comprehensive comparison with EHT data is beyond the scope of this letter, our findings suggest that incorporating `MLody` could lead to more accurate model-to-data comparisons and potentially refine the interpretation of EHT results.

In conclusion, `MLody` demonstrates enhanced accuracy in modeling polarized synchrotron emission, providing a more precise tool for studying black hole environments. Its application in upcoming observational campaigns, including those by the next-generation Event Horizon Telescope, may lead to more accurate interpretations of black hole observations.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/

Bronzwaer, T., Davelaar, J., Younsi, Z., et al. 2018, A&A, 613, A2, doi: 10.1051/0004-6361/201732149

Bronzwaer, T., Younsi, Z., Davelaar, J., & Falcke, H. 2020, A&A, 641, A126, doi: 10.1051/0004-6361/202038573

Chollet, F., et al. 2015, Keras, https://keras.io

Davelaar, J., Ripperda, B., Sironi, L., et al. 2023, ApJL, 959, L3, doi: 10.3847/2041-8213/ad0b79

Dexter, J. 2016, MNRAS, 462, 115, doi: 10.1093/mnras/stw1526

Do, T., Hees, A., Ghez, A., et al. 2019, Science, 365, 664, doi: 10.1126/science.aav8137

Event Horizon Telescope Collaboration, Akiyama, K., Alberdi, A., et al. 2019a, The Astrophysical Journal, 875, L1, doi: 10.3847/2041-8213/ab0ec7

Event Horizon Telescope Collaboration, Akiyama, K., Alberdi, A., et al. 2019b, ApJL, 875, L5, doi: 10.3847/2041-8213/ab0f43

Event Horizon Telescope Collaboration, Akiyama, K., Algaba, J. C., et al. 2021, The Astrophysical Journal, 910, L12, doi: 10.3847/2041-8213/abe71d

Event Horizon Telescope Collaboration, Akiyama, K., Alberdi, A., et al. 2022a, ApJL, 930, L12, doi: 10.3847/2041-8213/ac6674

Event Horizon Telescope Collaboration, Akiyama, K., Alberdi, A., et al. 2022b, The Astrophysical Journal, 930, L16, doi: 10.3847/2041-8213/ac6672

Event Horizon Telescope Collaboration, Akiyama, K., Alberdi, A., et al. 2023, ApJL, 957, L20, doi: 10.3847/2041-8213/acff70

—. 2024a, ApJL, 964, L25, doi: 10.3847/2041-8213/ad2df0

—. 2024b, ApJL, 964, L26, doi: 10.3847/2041-8213/ad2df1

Galishnikova, A., Philippov, A., & Quataert, E. 2023, ApJ, 957, 103, doi: 10.3847/1538-4357/acfa77

GRAVITY Collaboration, Abuter, R., Amorim, A., et al. 2019, A&A, 625, L10, doi: 10.1051/0004-6361/201935656

Hunter, J. D. 2007, Computing in Science and Engineering, 9, 90, doi: 10.1109/MCSE.2007.55

Jones, E., Oliphant, T., Peterson, P., et al. 2001, SciPy: Open Source Scientific Tools for Python

Kingma, D. P., & Ba, J. 2014, arXiv e-prints, arXiv:1412.6980, doi: 10.48550/arXiv.1412.6980

Leung, P. K., Gammie, C. F., & Noble, S. C. 2011, ApJ, 737, 21, doi: 10.1088/0004-637X/737/1/21

Mahadevan, R., Narayan, R., & Yi, I. 1996, ApJ, 465, 327, doi: 10.1086/177422

Marszewski, A., Prather, B. S., Joshi, A. V., Pandya, A., & Gammie, C. F. 2021, ApJ, 921, 17, doi: 10.3847/1538-4357/ac1b28

Millman, K. J., & Aivazis, M. 2011, Computing in Science & Engineering, 13, 9, doi: 10.1109/MCSE.2011.36

Mościbrodzka, M., Dexter, J., Davelaar, J., & Falcke, H. 2017, Monthly Notices of the Royal Astronomical Society, 468, 2214, doi: 10.1093/mnras/stx587

Mościbrodzka, M., Falcke, H., & Shiokawa, H. 2016, Astronomy & Astrophysics, 586, A38, doi: 10.1051/0004-6361/201526630

Nair, V., & Hinton, G. E. 2010, in Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10 (Madison, WI, USA: Omnipress), 807?814

Narayan, R., Igumenshchev, I. V., & Abramowicz, M. A. 2003, Publications of the Astronomical Society of Japan, 55, L69, doi: 10.1093/pasj/55.6.L69

Oliphant, T. E. 2007, Computing in Science & Engineering, 9, 10, doi: 10.1109/MCSE.2007.58

Pandya, A., Zhang, Z., Chandra, M., & Gammie, C. F. 2016, ApJ, 822, 34, doi: 10.3847/0004-637X/822/1/34

Porth, O., Olivares, H., Mizuno, Y., et al. 2017, Computational Astrophysics and Cosmology, 4, 1, doi: 10.1186/s40668-017-0020-2

Rybicki, G. B., & Lightman, A. P. 1979, Radiative Processes in Astrophysics (Wiley)

Shcherbakov, R. V. 2008, ApJ, 688, 695, doi: 10.1086/592326

Tchekhovskoy, A., Narayan, R., & McKinney, J. C. 2011, Monthly Notices of the Royal Astronomical Society, 418, L79, doi: 10.1111/j.1745-3933.2011.01147.x

van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, Computing in Science and Engineering, 13, 22, doi: 10.1109/MCSE.2011.37