

Neural network analysis of X-ray polarimeter data

A.L.Peirson

Abstract This chapter presents deep neural network based methods for enhancing the sensitivity of X-ray telescopic observations with imaging polarimeters. Deep neural networks can be used to determine photoelectron emission directions, photon absorptions points, and photon energies from 2D photoelectron track images, with estimates for both the statistical and model uncertainties. Deep neural network predictive uncertainties can be incorporated into a weighted maximum likelihood to estimate source polarization parameters. Events converting outside of the fiducial gas volume, whose tracks have little polarization sensitivity, complicate polarization estimation. Deep neural network based classifiers can be used to select against these events to improve energy resolution and polarization sensitivity. The performance of deep neural network methods is compared against standard data analysis methods, revealing a $< 0.75\times$ improvement in minimum detectable polarization for IXPE-specific simulations. Potential future developments and improvements to these methods are discussed.

Keywords: Deep learning, ensemble learning, neural networks, uncertainty quantification, imaging X-ray polarimetry, soft X-ray polarimetry, machine learning

Abel Lawrence Peirson
Kavli Institute for Particle Astrophysics and Cosmology, Stanford University, Stanford, CA, 94305
e-mail: alpv95@stanford.edu

Contents

Neural network analysis of X-ray polarimeter data	1
A.L.Pearson	
1 Introduction	4
1.1 How this chapter is organized	6
2 Imaging X-ray polarimetry	8
2.1 Track reconstruction	8
2.2 Polarization estimation	12
3 Deep neural networks	18
3.1 Machine learning with deep neural networks	18
3.2 Convolutional Neural Networks	21
3.3 Multitask learning	22
3.4 Uncertainty quantification	22
4 Neural networks for track reconstruction	26
4.1 Dataset	26
4.2 Geometric bias	27
4.3 Deep ensemble setup	29
4.4 Removing tail tracks	31
4.5 Training and ensemble selection	33
4.6 Performance	34
5 Neural networks for polarization estimation	39
5.1 Modulation factor	40
5.2 Weighted maximum likelihood estimator	41
5.3 Performance	43
6 Conclusion and future directions	47
References	47
References	47

1 Introduction

Measuring X-ray polarization has been a major goal in astrophysics for the last 40 years. X-ray polarization measurements offer rich opportunities to probe the magnetic field topology and emission physics of high energy astrophysical sources, such as accreting black holes and astrophysical jets [1, 2]. The recent development of photoelectron tracking detectors [3] has greatly improved the prospects of doing so. The gas pixel detector (GPD) [4] has brought soft X-ray polarimetry (1-10 keV) to the PolarLight CubeSat test [5], the scheduled NASA IXPE mission [6], and the potential Chinese mission, eXTP [7].

X-ray polarization telescopes using GPDs directly image electron tracks formed from photoelectrons scattered by incoming X-ray photons. This technique has the capability to build up an image of extended sources (e.g., supernova remnants or pulsar wind nebulae). Fig. 1 shows a schematic of the GPD and fig. 2 gives example photoelectron tracks at various photon energies, measured by IXPE’s GPDs. GPD sensitivity is limited by the track analysis algorithm used to recover source polarization, spatial structure, and energy, given a measured set of electron track images.

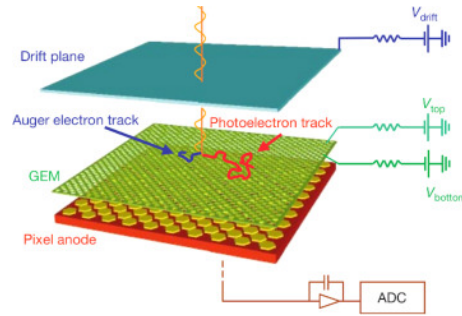
In the 1 – 15 keV range, the cross-section for photoelectron emission is proportional to $\cos^2(\theta - \theta_0)$, where θ_0 is the normal incidence X-ray’s electric vector position angle (EVPA) and θ the azimuthal emission direction of the photoelectron; see the previous chapter. Specifically, the emission angles θ follow the distribution

$$\theta \sim \frac{1}{2\pi} (1 + p_0 \cos[2(\theta - \theta_0)]). \quad (1)$$

By measuring a large number of individual photoelectron emission angles $\{\hat{\theta}_i\}_{i=1}^N$, one can recover the above distribution to extract the source polarization parameters: polarization fraction ($0 \leq p_0 \leq 1$) and EVPA ($-\pi/2 \leq \theta_0 < \pi/2$). In practice, the recovery of photoelectron emission angles from track images is imperfect. Track images are noisy due to Coulomb scattering and diffusion, and, especially for low

Fig. 1 Design of the GPD.

An incoming X-ray photon interacts with a gas molecule to produce a photoelectron between the entrance window and the gas electron multiplier (GEM). Secondary charges created along the photoelectron’s path are multiplied by the GEM and read out at the anode plane. This forms a 2D photoelectron track image on hexagonal pixel grid; fig. 2 gives some examples. Figure from [8].



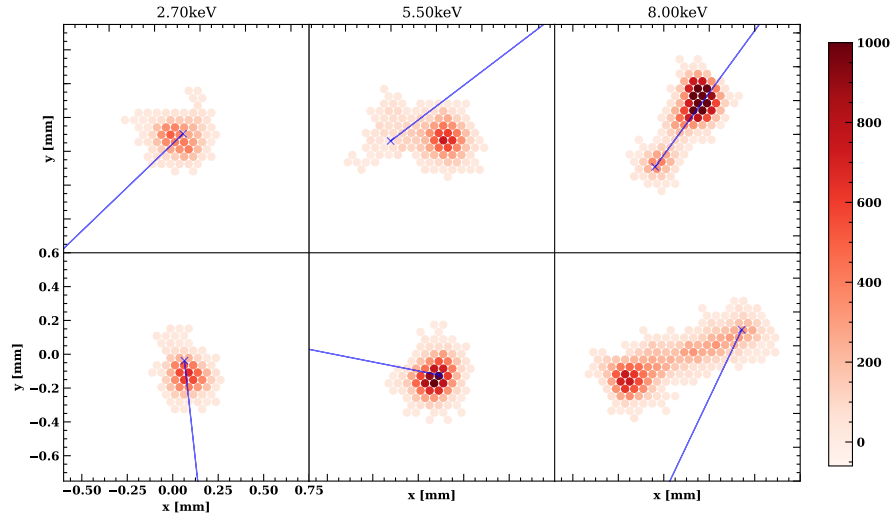


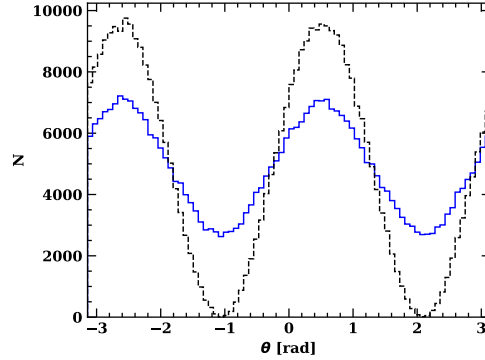
Fig. 2 A selection of measured photoelectron track events from IXPE GPDs. Pixel color density represents charge deposited, blue crosses show photon x-y absorption point and blue lines the initial photoelectron direction. Pixels are thresholded at 10 counts. Track morphologies vary widely and depend strongly on energy.

energies, are often barely resolved. For example, in fig. 3 the distribution of recovered emission angles using IXPE’s classical track reconstruction is significantly blurred compared to the true distribution.

The best classical track reconstruction method for GPDs is a moment analysis described by [3] and in the previous chapter. Impressive accuracies for the emission angle and photon absorption point are achieved from a simple weighted combination of track moments. Photon energy estimates are proportional to the total collected GPD charge for a track. The track ellipticity is quantified to provide a rough proxy for track reconstruction quality. High ellipticity tracks typically have more accurate angle estimates. However, simple moments cannot capture all image information, especially for long high energy tracks, and so a more sophisticated image analysis scheme can lead to improved track emission angle, absorption point, and energy recovery.

Once the emission angles have been reconstructed, $\{\hat{\theta}_i\}_{i=1}^N$, classical polarization estimation uses a maximum likelihood estimator (MLE) or direct curve fit to calculate $(\hat{p}_0, \hat{\theta}_0)$. This assumes individual tracks contribute equally to the final polarization estimate. In fact, photoelectron tracks are very morphologically diverse, even for the same photon energy (cf. fig. 2), and so emission angle estimates are highly heteroskedastic: some emission angles are much better estimated than others. This is especially true for tracks of different photon energies, important in the case of broadband polarization estimates. Assuming an equal contribution from all estimated

Fig. 3 Photoelectron angles for a 100% polarized ($p_0 = 1$) 6.4keV simulated line source. Black gives the true photoelectron angles, blue gives the recovered photoelectron angles for the standard moment analysis.



emission angles results in sub-optimal polarization recovery. To ameliorate this, IXPE's standard analysis performs an event cut removing tracks in the bottom 20% of estimated ellipticities; this does provide a marginal improvement in the recovered polarization signal. However, a detailed analysis with proper estimation of emission angle uncertainties and their inclusion in the likelihood function can substantially improve the signal-to-noise ratio of recovered polarization.

Neural networks are supervised learning models that are the state of the art for classification and regression problems in many fields, especially computer vision [9]. They are highly non-linear models trained on large datasets of labelled input-output example pairs. Neural networks are often criticized as black box models with no output interpretability. However, recent work in uncertainty quantification for neural networks [10] has demonstrated these models are capable of trustworthy uncertainty estimates on their predictions. Neural networks' success in dealing with image data and their advances in uncertainty quantification make them a perfect candidate for X-ray polarimetry track analysis [11, 12].

This chapter outlines neural networks approaches to track analysis algorithms, covering both track reconstruction and polarization estimation. The methods discussed in this chapter were developed primarily for GPDs, but should be applicable to all imaging X-ray polarimetry technologies, such as the time projection chambers proposed for the former GEMS/Praxys mission [13].

1.1 How this chapter is organized

- §2 contains an overview of the data analysis process necessary to go from telescope observations to polarization measurements and energy spectra, including some of the standard pipeline, covered in more detail in the previous chapter.
- §3 gives a brief overview of neural network models and the types of deep learning important for X-ray polarimetry, with a discussion of uncertainty quantification for neural networks.

- §4 describes an application of neural networks to track reconstruction: extracting emission angles, absorption points and photon energies from GPD track images. Comparison of neural network methods to the standard data analysis. A neural network approach to identifying and removing events outside of the main detector.
- §5 illustrates the use of neural network results for polarization estimation: using neural network predicted uncertainties on emission angles to form a proper maximum likelihood estimator of the polarization parameters. Comparison of neural network methods to the standard data analysis. Discussion of the impacts of events outside of the main detector gas.
- §6 contains the conclusion, limitations, and future directions in the field.

2 Imaging X-ray polarimetry

Measuring the X-ray polarization of a source using an imaging X-ray polarimeter is, from a data analysis perspective, a two step process.

1. *Track reconstruction.* The telescope is pointed at the source and collects photoelectron track images, see fig. 2. Individual track images are processed to extract all of the relevant features: emission angles, absorption points and photon energies.
2. *Polarization estimation.* The extracted emission angles from many individual photoelectron tracks are combined to estimate the source polarization parameters.

For spatially extended, time-varying or spectrally varying sources, the extracted emission angles could be grouped based on absorption points, arrival time or photon energy, i.e. in spatial, time and/or energy bins. Polarization estimation from these grouped emission angles follows the same procedure.

This chapter describes classical approaches for both steps, a prerequisite for the neural network approach. In classical polarization analysis the two steps are entirely disconnected. §5 will show how these two steps can be more closely connected using neural network uncertainty estimates.

2.1 Track reconstruction

Photoelectron track images contain all the extractable information in imaging X-ray polarimeters. More specifically, the various sensitivities of an imaging X-ray polarimeter can be attributed to a few photoelectron track features that can be extracted from individual track images:

- *Polarization sensitivity:* photoelectron emission angle θ , §2.1.1.
- *Spatial sensitivity:* photon absorption point (x, y) , §2.1.2.
- *Spectral sensitivity:* photon energy E , §2.1.3.

Tracks are imaged by charge deposition onto an array of pixels. For example, IXPE's GPD pixel array is a $\sim 15\text{mm} \times 15\text{mm}$ square tiled with hexagonal pixels on a $50\mu\text{m}$ pitch [8]; see fig. 4. The charge deposited in each pixel is measured in integer counts, typically ranging from 0 to 1000+, fig. 2. Track images are identified on the pixel array by clustering analysis and the region of interest is cropped. Each track image comes as a list of (x, y, c) tuples: the pixel coordinates x, y and charge deposited c .

There are a number of instrument specific settings that affect track morphology. For example, a pixelwise charge threshold is usually applied to individual track images to reduce small noisy pixel clusters. The GPD gas pressure and composition can also be varied; lower pressure allows for longer tracks that are easier to reconstruct but reduces the detector quantum efficiency. The methods presented in this chapter generalize to all imaging polarimeter settings, details on optimizing specific settings

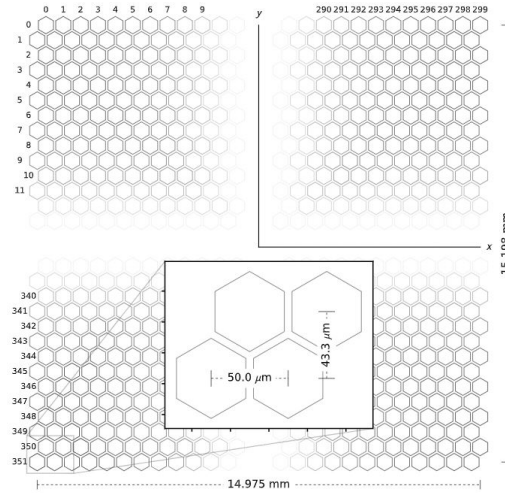


Fig. 4 Layout of IXPE GPD readout pixel array. There are approximately 90000 pixels. More pixels allow for better track reconstruction. Figure from [3].

can be found in chapters XY. The examples shown here assume IXPE-specific GPD settings.

2.1.1 Emission angle reconstruction

Source polarization is encoded in the photoelectron emission angles θ . These are the image plane projected initial directions of the photoelectron at the interaction point. Track images capture the entire track of the photoelectron in the image plane, so in principle it should be possible to recover the initial direction. In practice, track images are relatively noisy representations of the photoelectron path. Charge diffusion means the further away from the pixel array a photoelectron is emitted, the less well defined its track image will be.

Track images are typically asymmetric and follow a similar pattern: the photoelectron starts at the absorption point and moves along its initial direction. Sometimes an Auger electron is also released at the absorption point. As the photoelectron moves away from the absorption point, it is Coulomb scattered away from the initial direction. As the photoelectron slows down, it ionizes more gas molecules, depositing more charge in the pixel array and culminates in a final Bragg peak. Low energy photoelectrons travel less far from the absorption point and leave nearly circularly symmetric track images.

The scattering of the photoelectron from its initial direction and short, symmetric photoelectron tracks make emission angle reconstruction challenging. The moment analysis described in [3] uses the known structure of photoelectron tracks to improve their emission angle estimates. They initially calculate the principal axis of the track image by maximizing the second charge moment about the track barycenter. This provides a simple initial estimate of the emission angle, but is skewed by the Bragg

peak – the dense final part of the track that is usually not correlated to the initial direction because of Coulomb scattering. To correct for the Bragg peak bias, the location of the Bragg peak is calculated by observing the sign of the third charge moment about the track barycenter. This identifies which side of the track contains the Bragg peak. The absorption point is estimated as being a multiple of the second moment away from the track barycenter along the principal axis, away from the Bragg peak. Finally, the emission angle is calculated by using the second moment about the absorption point, using only pixels in the initial half of the track (those not on the Bragg peak side). See the previous chapter for more details on the moment analysis.

2.1.2 Absorption point reconstruction

Photon absorption points (x, y) on the image plane along with the known point spread function (PSF) of the telescope optics allow for the spatial resolution of extended sources. Better absorption estimates yield better spatial resolution. A simple estimator for the absorption point of an individual photon from its track image is the track barycenter, but this can be strongly biased by the photoelectron Bragg peak §2.1.1. In moment analysis [3], the absorption point is estimated jointly with the emission angle by identifying and excluding the Bragg peak. For low energy symmetric tracks where it is impossible to separate the Bragg peak, the track barycenter is used as the standard absorption point estimator.

2.1.3 Energy reconstruction

An X-ray photon's energy is proportional to the average charge deposited in the detector. Since the charge deposited is subject to statistical fluctuations, the energy resolution is limited and this limit depends on the effective detector Fano factor. Classical polarimetry uses a simple linear model calibrated on real detector track images to reconstruct the photon energy. Thus, the total track charge is assumed proportional to the photon energy. This method approaches but does not meet the theoretical best energy resolution.

2.1.4 Events converting outside of the gas volume

So far, the track images considered are formed when photons interact with the detector gas. However, incoming X-ray photons can also convert in detector components just outside of the main gas volume, with electrons penetrating the gas triggering the GPD and producing photoelectron tracks. For IXPE's GPDs, these external interactions occur in the thin beryllium entrance window at the top of the GPD and in the gas electron multiplier copper material at the bottom, fig. 1. The track images formed by these external interactions have different morphological properties compared to

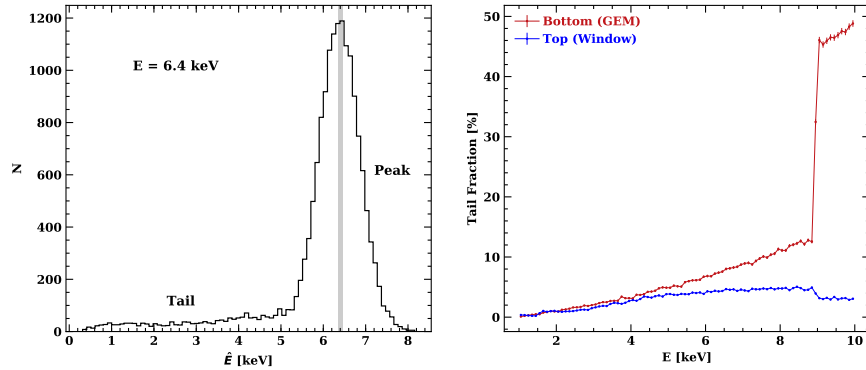


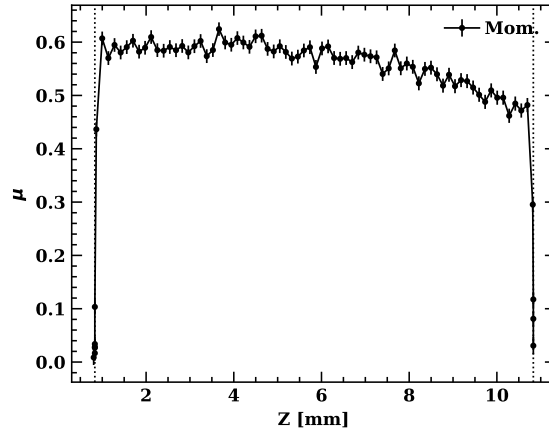
Fig. 5 *Left*: Recovered energy histogram for a 6.4keV line source. A simple linear function of total charge deposited is used here for recovered energy, as in the standard moment analysis. The long low energy tail is produced by events converting in the window or GEM. *Right*: Fraction of events that are ‘tails’ as a function of energy. Red and blue traces show the Be window and GEM conversion respectively. The jump in GEM conversions at 8.9 keV Cu edge is prominent.

those in the detector gas. Notably, they have less total charge deposited for the same photon energy, because some of the photoelectron energy can be lost in the solid window or GEM material. In addition they tend to have a denser ‘core’ since the photoelectrons surviving to the gas volume tend to have trajectories with a large vertical component. The left panel of fig. 5 shows the linearly reconstructed energy (§2.1.3) for a 6.4keV line source. External events form a low energy tail on the predicted energy histogram – external events are described here tail events, producing tail tracks, while events converting in the detector gas are called peak events. The right panel gives the population fraction of tail events as a function of photon energy. Tail tracks are mostly subdominant, but with a linearly increasing fraction as photon energy increases. At energies above 8.9 keV, the Copper K absorption edge causes a large increase in tail events in the GEM, increasing the tail track fraction.

Tail tracks also have significantly reduced polarization signal compared to peak tracks. The emitted photoelectron is scattered more strongly in the dense GEM and window materials than in the detector gas; as a result, the initial photoelectron direction is often irrecoverable. Fig. 6 gives the simulated polarization sensitivity as a function of vertical direction along IXPE’s GPD, using the moment analysis to estimate photoelectron emission angles. There is a clear drop in sensitivity at both vertical edges of the detector where tail tracks dominate the absorption. Additionally, this figure serves to illustrate the decrease in polarization sensitivity with increasing vertical distance from the pixel array. This is due to charge diffusion in the detector gas blurring tracks before they reach the pixel read-out array (§2.1.1).

If included, tail tracks can cause problems during polarization estimation. Not only do tail tracks degrade the detector polarization sensitivity, the energy-dependent tail fraction can make it difficult to properly calibrate the polarization sensitivity and degrade the energy resolution for continuous spectra. In classical imaging polarime-

Fig. 6 Polarization sensitivity, measured by the modulation factor μ (§2.2), as a function of absorption point height in IXPE’s GPD. Dotted lines denote the edges of the GPD, the top at $z=10.83\text{mm}$ is the Beryllium window, and the bottom at $z=0.83\text{mm}$ is the Cu-coated GEM.



try, tail tracks are cut from the line source calibration by excluding the low energy tail from the recovered energy histogram, fig. 5. For real, continuous sources this is not possible, so tail tracks remain an important problem for classical polarimetry approaches. §4.4 will describe how a NN analysis can identify tail tracks based on their morphological differences and decrease their contamination of the polarization signal.

2.2 Polarization estimation

Polarization estimation can begin once all relevant track features have been recovered. The basic problem is to estimate source polarization parameters (p_0, θ_0) and their uncertainties from a set of reconstructed emission angles $\{\hat{\theta}_i\}_{i=1}^N$. As described in the introduction, true emission angles θ exhibit a sinusoidal modulation with period π that depends on the source polarization

$$p(\theta|p_0, \theta_0) = \frac{1}{2\pi} (1 + p_0 \cos[2(\theta - \theta_0)]), \quad (2)$$

where $0 \leq p_0 \leq 1$, $-\pi/2 \leq \theta_0 < \pi/2$ and $-\pi \leq \theta < \pi$. Reconstructed emission angles $\hat{\theta}$ are imperfectly recovered, so these follow an adjusted distribution

$$p(\hat{\theta}|p_0, \theta_0) = \frac{1}{2\pi} (1 + p_0 \mu \cos[2(\hat{\theta} - \theta_0)]). \quad (3)$$

The modulation factor $0 \leq \mu \leq 1$ is a measure of the detector polarization sensitivity: how well the emission angles are recovered. A higher modulation factor means better polarization sensitivity and $\mu = 1$ means perfect emission angle reconstruction, $\theta = \hat{\theta}$. The modulation factor is the recovered polarization fraction \hat{p}_0 for a 100%

polarized source, $p_0 = 1$. Fig. 3 gives an example for a $p_0 = 1$ source, the moment analysis recovered emission angles follow the blue $\mu < 1$ distribution while the true emission angles have $\mu = 1$. The modulation factor is a function of photon energy, higher energy tracks have higher modulation factor because they have longer tracks, thus better emission angle estimates. Track reconstruction algorithms and detector hardware both affect the instrument modulation factor. The modulation factor can be considered an average of the reconstruction quality of all tracks for a particular source energy spectrum. In §5 we will revisit the modulation factor from first principles; for now it can be treated as a known constant for a given source energy spectrum, determined by instrument calibration.

2.2.1 Stokes parameters

It is usually simpler to estimate the normalized linear Stokes parameters (Q, \mathcal{U}) instead of the polarization fraction and EVPA (p_0, θ_0). These are an alternative representation of the source linear polarization. Disregarding circular polarization, the Stokes parameters are defined as

$$Q = I p_0 \cos 2\theta_0, \quad (4)$$

$$U = I p_0 \sin 2\theta_0, \quad (5)$$

where I is the source intensity, and the normalized Stokes parameters are

$$Q = p_0 \cos 2\theta_0, \quad (6)$$

$$\mathcal{U} = p_0 \sin 2\theta_0, \quad (7)$$

$-1 \leq Q \leq 1$, $-1 \leq \mathcal{U} \leq 1$, from which the polarization fraction and EVPA can be derived:

$$p_0 = \sqrt{Q^2 + \mathcal{U}^2}, \quad (8)$$

$$\theta_0 = \frac{1}{2} \arctan \frac{\mathcal{U}}{Q}. \quad (9)$$

Using Stokes parameters, the probability density eq.3 can be rewritten

$$p(\hat{\theta} | Q, \mathcal{U}) = \frac{1}{2\pi} (1 + Q\mu \cos 2\hat{\theta} + \mathcal{U}\mu \sin 2\hat{\theta}). \quad (10)$$

This is a more convenient form because the probability density is linear in the Stokes parameters, unlike for (p_0, θ_0) . Some researchers even prefer to work solely with Stokes parameters. Since likelihoods are invariant to reparameterization, estimating the Stokes parameters and converting back to (p_0, θ_0) is equivalent to estimating (p_0, θ_0) directly, although some care must be taken in propagating the uncertainties.

2.2.2 Methods

With the modulation factor known, (p_0, θ_0) and/or $(\mathcal{Q}, \mathcal{U})$ can be recovered from the emission angles $\{\hat{\theta}_i\}_{i=1}^N$ in a number of ways. The simplest method is a binned curve fit, i.e. directly fitting the probability density eq.10 to the histogram of emission angles $\{\hat{\theta}_i\}_{i=1}^N$, fig.3. Since the Stokes parameters are linear in the probability density eq.10 they can be fit using binned least squares. The resulting estimates $(\hat{\mathcal{Q}}, \hat{\mathcal{U}})$ are unbiased and equivalent to the maximum likelihood estimator (MLE), so long as the bin widths are small enough and each bin contains enough events such that Poisson statistics can be well approximated by Gaussian. This is not always the case, so directly applying the MLE is often a better approach.

The MLE is the estimator $(\hat{\mathcal{Q}}, \hat{\mathcal{U}})$ that maximizes the likelihood function:

$$L(\{\hat{\theta}_i\}_{i=1}^N | \mathcal{Q}, \mathcal{U}) = \prod_{i=1}^N \frac{1}{2\pi} (1 + \mathcal{Q}\mu \cos 2\hat{\theta}_i + \mathcal{U}\mu \sin 2\hat{\theta}_i). \quad (11)$$

Here L is a function of $(\mathcal{Q}, \mathcal{U})$ and the observations $\{\hat{\theta}_i\}_{i=1}^N$ are fixed. Computationally, it is easier to minimize the negative log-likelihood function

$$-\log L = N \log 2\pi - \sum_{i=1}^N \log (1 + \mathcal{Q}\mu \cos 2\hat{\theta}_i + \mathcal{U}\mu \sin 2\hat{\theta}_i). \quad (12)$$

This expression must be minimized numerically to find $(\hat{\mathcal{Q}}, \hat{\mathcal{U}})$. Confidence intervals are calculated by numerically integrating the negative log likelihood function in the vicinity of the final estimator. Stokes parameter estimators and their errors can be simply transformed into $(\hat{p}_0, \hat{\theta}_0)$ if desired.

It is often inconvenient to evaluate the MLE and its errors numerically. An analytical solution for the MLE exists if $|\mathcal{Q}\mu| \ll 1$, $|\mathcal{U}\mu| \ll 1$. In practical X-ray polarimetry this limit is nearly always satisfied; polarimeters typically don't achieve much better than $\mu \lesssim 0.5$ averaged over a continuous spectrum and real astrophysical sources generally have low polarization $p_0 \lesssim 0.3$. By applying the Taylor expansion $\log(1+x) \approx x - x^2/2$ to the negative log-likelihood eq. 12 and minimizing the resulting quadratic form one finds

$$\hat{\mathcal{Q}} = \frac{1}{\mu} \frac{(\sum_{i=1}^N \cos 2\hat{\theta}_i - \sum_{i=1}^N \cos 2\hat{\theta}_i \sin 2\hat{\theta}_i) \sum_{i=1}^N \sin^2 2\hat{\theta}_i}{\sum_{i=1}^N \cos^2 2\hat{\theta}_i \sum_{i=1}^N \sin^2 2\hat{\theta}_i - (\sum_{i=1}^N \cos 2\hat{\theta}_i \sin 2\hat{\theta}_i)^2}, \quad (13)$$

$$\hat{\mathcal{U}} = \frac{1}{\mu} \frac{(\sum_{i=1}^N \sin 2\hat{\theta}_i - \sum_{i=1}^N \cos 2\hat{\theta}_i \sin 2\hat{\theta}_i) \sum_{i=1}^N \cos^2 2\hat{\theta}_i}{\sum_{i=1}^N \cos^2 2\hat{\theta}_i \sum_{i=1}^N \sin^2 2\hat{\theta}_i - (\sum_{i=1}^N \cos 2\hat{\theta}_i \sin 2\hat{\theta}_i)^2}. \quad (14)$$

Notice that $\mathbb{E}[\sum_{i=1}^N \cos 2\hat{\theta}_i \sin 2\hat{\theta}_i] = 0$, $\mathbb{E}[\sum_{i=1}^N \cos^2 2\hat{\theta}_i] = N/2$ and $\mathbb{E}[\sum_{i=1}^N \sin^2 2\hat{\theta}_i] = N/2$. For the relatively large N in X-ray polarimetry, these terms are effectively constants, so the expressions reduce to

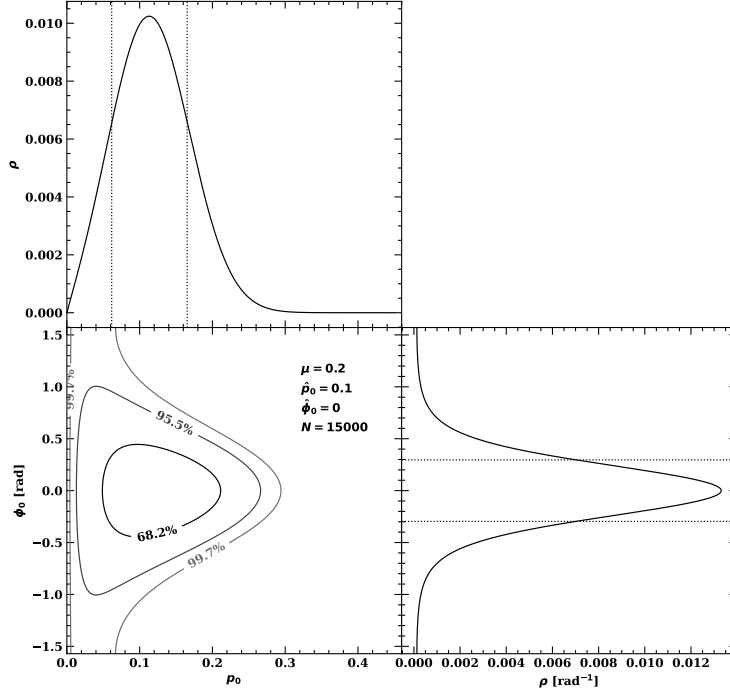


Fig. 7 Confidence intervals from full posterior distribution (eq.17) over polarization parameters (p_0, ϕ_0) given estimates $(\hat{p}_0, \hat{\phi}_0)$ using N events and instrument modulation factor μ . Confidence intervals for 68.2%, 95.5% and 99.7% (1σ , 2σ , 3σ) are displayed. The full marginal posterior distributions are plotted on the wings with 68.2% (1σ) confidence intervals drawn.

$$\hat{Q} = \frac{2}{N\mu} \sum_{i=1}^N \cos 2\hat{\theta}_i, \quad (15)$$

$$\hat{U} = \frac{2}{N\mu} \sum_{i=1}^N \sin 2\hat{\theta}_i. \quad (16)$$

Because these are unbiased and MLE, they are the minimum variance unbiased estimators for (Q, U) . In other words, so long as $|Q\mu| \ll 1$, $|U\mu| \ll 1$ and $N \gtrsim 1000$, these are the best possible estimators for (Q, U) .

Kislat et al. [14] derive the full analytical posterior distribution for both (Q, U) and (p_0, θ_0) given the estimators eqs.15,16. They find the (Q, U) posterior follows a bivariate normal distribution while the (p_0, θ_0) posterior is

$$\begin{aligned}
p(p_0, \theta_0 | \hat{p}_0, \hat{\theta}_0) &= \frac{\sqrt{N} \hat{p}_0 \mu^2}{2\pi\sigma} \times \\
&\exp \left[-\frac{\mu^2}{4\sigma^2} \left\{ \hat{p}_0^2 + p_0^2 - 2\hat{p}_0 p_0 \cos(2(\hat{\theta}_0 - \theta_0)) \right. \right. \\
&\quad \left. \left. - \frac{\hat{p}_0^2 p_0^2 \mu^2}{2} \sin^2(2(\hat{\theta}_0 - \theta_0)) \right\} \right], \tag{17}
\end{aligned}$$

where

$$\sigma = \sqrt{\frac{1}{N} \left(1 - \frac{p_0^2 \mu^2}{2} \right)}. \tag{18}$$

The posterior assumes a uniform prior over (p_0, θ_0) . With the posterior in hand, any desired confidence interval can be computed analytically or numerically. High μ and high N reduce the width of the posterior so are both desirable to minimize the errors on recovered polarization parameters. Note that the estimator \hat{p}_0 is not unbiased, its posterior is highly asymmetric for low p_0 since $p_0 \geq 0$. For $p_0 = 0$ some polarization fraction $\hat{p}_0 > 0$ will always be measured. The amount of polarization likely to be measured when $p_0 = 0$ depends on the width of the posterior, i.e. N and μ , and is often used as a measure of detector sensitivity or signal-to-noise ratio (SNR), §2.2.3. Fig. 7 shows the posterior distribution for an example observation.

The recommended approach to polarization estimation is using eqs. 15, 16 for point estimates and the derived posterior eq. 17 for confidence intervals. This approach is expedient and optimal given the instrumental constraints of imaging X-ray polarimetry. In §5 we extend eqs. 15 – 17 to include neural network uncertainty estimates on predicted emission angles $\hat{\theta}$.

2.2.3 Minimum detectable polarization (MDP)

The standard figure-of-merit used in X-ray polarimetry to compare instrument sensitivity is minimum detectable polarization (MDP) [15]. MDP_{99} is the polarization fraction that has a 1% probability of being exceeded by chance for an unpolarized ($p_0 = 0$) source. This can be found by integrating the posterior distribution eq.17:

$$\int_0^{\text{MDP}_{99}} \int_{-\pi/2}^{\pi/2} p(p_0, \theta_0 | \hat{p}_0, \hat{\theta}_0) d\theta_0 dp_0 = 0.99, \tag{19}$$

$$\text{MDP}_{99} = \frac{4.29}{\mu\sqrt{N}}. \tag{20}$$

On fig. 7, the MDP_{99} would be the one sided 99% confidence interval for the p_0 marginal distribution (top panel).

The MDP_{99} depends only on the modulation factor (signal) and the Poisson counting noise $1/\sqrt{N}$; it is effectively an inverse of the SNR. Polarimeters with lower

MDP₉₉ can expose for shorter times and get the same confidence on their polarization measurements. Track reconstruction and polarization estimation approaches that increase the modulation factor for a fixed number of tracks N will decrease the MDP₉₉. Sections 4 and 5 will describe how a neural network approach can improve both track reconstruction and polarization estimation to minimize the MDP₉₉.

3 Deep neural networks

Deep neural networks have achieved state-of-the-art performance on a wide variety of machine learning tasks and are becoming increasingly popular in domains such as speech recognition [16], natural language processing [17], bioinformatics [18] and especially computer vision [19]. Going from track images to emission angle estimates can be classified as a computer vision problem, so it is not surprising that neural networks would be well suited to track reconstruction. The Cherenkov Telescope Array (CTA) [20] team have applied related deep learning methods to differentiate between cosmic rays and gamma rays. Notably they also have to deal with a hexagonal pixel grid, also found in imaging X-ray polarimeters. The IceCube collaboration has begun the use of graph neural networks to identify 3D neutrino tracks with great success [21]. This section briefly covers deep neural network and machine learning concepts relevant to X-ray polarimetry, but for a proper introduction to this important field see [22, 23].

3.1 Machine learning with deep neural networks

Deep neural networks (DNNs) are a type of supervised machine learning algorithm. Machine learning algorithms learn from data, as opposed to a fixed algorithm programmed by human beings. Supervised machine learning algorithms learn from labelled data: known input-output pairs $\{x_i, y_i\}_{i=1}^N$. An example from X-ray polarimetry would be track image (input), emission angle (output). Supervised machine learning algorithms can be used to solve a variety of tasks. Most tasks belong to one of two groups: regression or classification. In regression the algorithm is trained to predict a continuous numerical value given an input. In classification, the algorithm predicts the probability an input belongs to a particular class from a preselected group of finite classes. In the case of track reconstruction, we are interested mainly in regression.

In essence, DNNs approximate the unknown function f^* that maps problem inputs to outputs $y = f^*(x)$. DNNs are organized as a forward network of function layers each with their own learnable parameters or weights w , e.g. $f(x) = f_3(f_2(f_1(x; w_1); w_2); w_3)$. The initial function layer f_1 is known as the input layer, the final layer f_3 is the output layer, and all layers in between are the hidden layers. Using many hidden layers tends to improve the predictive capacity of the DNN, whence the term 'deep learning'. During training, the DNN $f(x)$ is iteratively directed towards $f^*(x)$.

If the layer functions f_1, f_2, f_3 were linear then the DNN $y = f(x)$ would be a linear model $y = w^T x$, equivalent to a simple least squares approach. DNNs use non-linear layer functions. This allows the DNN to learn non-linear relationships between the input features $y = w_2^T f_1(x; w_1)$. Here $f_1(x; w_1)$ is a non-linear function with learnable parameters w_1 . In this way DNNs are able to learn what functions of the input features should be combined, unlike a linear model that linearly combines

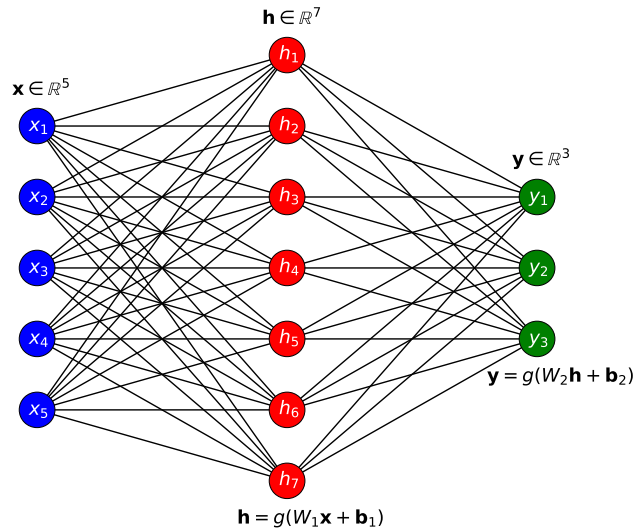


Fig. 8 Feed-forward neural network with one hidden layer. The neural network takes vector \mathbf{x} as input and produces vector \mathbf{y} as output.

input features directly. A simple example of a non-linear layer function used in DNNs is $f_1(x; w_1, b) = g(w_1^T x + b)$ where w_1 is a matrix and g is a non-linear function such as tanh, the sigmoid function or a rectified linear unit (ReLU). DNNs using this kind of layer are universal function approximators in the asymptotic limit [24], so they should in principle well approximate any $f^*(x)$. In practice, memory, training and data limitations make this difficult to achieve. Many DNNs use domain specific layer functions and architectures to reduce these limitations and converge to $f^*(x)$ more quickly, see §3.2. Figure 8 shows a schematic of a simple DNN.

3.1.1 Training

As in supervised linear models like least squares regression, DNNs are trained to minimize a specific loss function over the observed data. In most tasks this normally means minimizing the negative log-likelihood, as in §2.2. For regression tasks, if the output distribution family $p(y|x)$ is unknown, a Gaussian should usually be the default since it is the least informative choice. For a Gaussian likelihood, the negative log likelihood is

$$L(\{y_i, x_i\}_{i=1}^N | \mathbf{w}) \propto \sum_{i=1}^N \|y_i - \hat{y}(x_i | \mathbf{w})\|^2, \quad (21)$$

the mean squared error (MSE) of prediction for the dataset. The loss function $L(\mathbf{w})$ is minimized over the DNN weights \mathbf{w} . MSE is a common loss function for regression with DNNs.

For classification tasks, $p(y|x)$ is usually assumed to follow a Bernoulli (in binary, 2-class classification) or categorical distribution (in >2 class classification). In the Bernoulli case, the negative log-likelihood is

$$L(\{y_i, x_i\}_{i=1}^N | \mathbf{w}) \propto \sum_{i=1}^N -y_i \log(\hat{y}(x_i | \mathbf{w})) - (1 - y_i) \log(1 - \hat{y}(x_i | \mathbf{w})). \quad (22)$$

In order to ensure $0 \leq \hat{y}_i \leq 1$, DNNs for classification usually end with a softmax layer [25]. These project DNN outputs onto the appropriate simplex, i.e., makes sure all predicted probabilities sum to one. For K output classes

$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{i=1}^K e^{x_i}} \quad (23)$$

Since DNNs use non-linear functions, eq. 21 or 22 cannot be minimized analytically. Non-linearity makes the loss function complex with multiple local minima and saddle points in a very high dimensional weight space $\{\mathbf{w}\}$. This challenging minimization problem is usually tackled with variations on the stochastic gradient descent (SGD) algorithm [22]. At each iteration SGD approximates the gradient of the loss function using a subset of the training data

$$g_i = \nabla_{\mathbf{w}} L(\mathbf{w}) \approx \sum_{i=1}^n \nabla_{\mathbf{w}} \|y_i - \hat{y}(x_i | \mathbf{w})\|^2, \quad n \ll N \quad (24)$$

and takes a step in parameter (\mathbf{w}) space along the direction of the negative gradient

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \epsilon g_i. \quad (25)$$

The process is repeated until suitable convergence. The weights are initialized randomly [] and the learning rate ϵ is a tuneable scalar for the specific application. The gradient for each individual w_i can be calculated by using the chain rule through each layer of the DNN, in a process called backpropagation [22]. In order for SGD to work, the DNN has to use a differentiable loss function and layer functions f_1, f_2, f_3 . The whole dataset N is not used to calculate the gradient because it is too slow and often too large to fit into memory. Furthermore, DNNs benefit from using small batches n to approximate the gradient because the added stochasticity lets the minimization escape saddle points and local minima [22]. Usually DNNs are trained on graphical processing units (GPUs), specialized computer hardware for fast and parallelizable linear algebra operations.

3.1.2 Validation and model selection

DNNs typically have on the order of millions, sometimes billions [26], of trainable weights \mathbf{w} . The number of weights is often higher than the number of training examples, making overfitting the labelled dataset likely. This is the central challenge of machine learning: the algorithm must perform well on new inputs, not just those in the training dataset. To ensure model generalization to unseen examples, part of the training data is set aside for validation. The trainable weights \mathbf{w} are not trained on these examples. DNNs come with a number of additional model parameters – hyperparameters – that can be adjusted to improve model generalization; examples include the learning rate ϵ , the neural network architecture (type and number of layers) and any regularization. During training, the DNN is evaluated on the validation examples to tune the hyperparameters. This process is known as model selection.

Regularization is an important hyperparameter for improving model generalization. In exactly the same way as linear models can be regularized, for example in ridge regression, one can regularize a DNN by adding a penalty on the weights \mathbf{w} to the loss function. A common example is L2 regularization:

$$L(\mathbf{w}) \propto \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^N \|y_i - \hat{y}(x_i|\mathbf{w})\|^2. \quad (26)$$

The additional term restricts the model space by encouraging the weights \mathbf{w} to be close to zero. The hyperparameter λ can be tuned to maximize generalization on the validation examples. This type of regularization is equivalent to a Gaussian prior on the weights with mean zero. By restricting the model space, regularization helps prevent overfitting and improves generalization when λ is appropriately tuned.

3.2 Convolutional Neural Networks

Some DNNs have specialized layer functions that provide an inductive bias helpful for specific inputs. Convolutional neural networks (CNNs) [27] were designed for processing inputs that have neighbouring position correlations, for example 2D images, where adjacent pixel values are often correlated, or 1D time series where adjacent time points are often correlated. CNNs use layer functions that apply a convolution to the input. The convolution kernel function is learned during training. This kind of layer function is invariant under translation of the input, an important prior for working with image data. CNNs have revolutionized image processing and computer vision applications since winning the Imagenet benchmark competition in 2012 [19]. (Paragraph and plot of convolution here maybe)

Recently, a certain class of CNNs called residual networks (ResNets) have achieved state-of-the-art results in multiple image benchmark tasks [28]. ResNets introduce 'skip' connections between certain layers that allow the CNN to ignore the

layers in between. This alleviates many of the difficulties in training extremely deep neural networks.

3.3 Multitask learning

So far, only DNNs with single objectives and outputs have been considered. However, DNNs can be trained to solve multiple tasks at once, producing a vector of outputs. Solving multiple tasks at once often performs better than solving each task individually. This is partly because multiple tasks have a regularizing effect on each other, penalizing model complexity and overfitting on any single task leading to better overall model generalization [29]. Moreover, since each task shares the same input representation, e.g., an image, the features learned for a single task can help the others train faster and better.

In a multitask learning setting, the loss function includes the terms for each individual task. For example, in regression with three tasks and an MSE loss for each

$$L(\mathbf{w}) \propto \sum_{i=1}^N \|y_i^1 - \hat{y}^1(x_i|\mathbf{w})\|^2 + \alpha \sum_{i=1}^N \|y_i^2 - \hat{y}^2(x_i|\mathbf{w})\|^2 + \beta \sum_{i=1}^N \|y_i^3 - \hat{y}^3(x_i|\mathbf{w})\|^2, \quad (27)$$

where the DNN outputs concatenated vector $(\hat{y}^1, \hat{y}^2, \hat{y}^3)$. The relative size of each task loss term is controlled by the hyperparameters α, β . As usual, these hyperparameters should be tuned to maximize the model performance on unseen examples. More often in multitask learning, each task has a different form of loss function.

3.4 Uncertainty quantification

Once trained, DNNs can achieve state-of-the-art prediction accuracy in many domains. However, these models cannot capture the uncertainty inherent in their predictions. Predictive uncertainty quantification can be crucial, especially in applications where the real-world data distribution differs from the training data distribution. In this scenario, DNNs can extrapolate, producing overconfident but highly inaccurate predictions sometimes with disastrous consequences [30]. Predictive uncertainty is best represented by a posterior distribution on the predicted parameters given the inputs, however, simply quoting the variance of the posterior, often assumed to be Gaussian, is common practice.

There are two germane types of uncertainty one can model [31]. *Aleatoric uncertainty* captures noise inherent in the data. This is equivalent to statistical uncertainty or unexplained variance. On the other hand, *epistemic uncertainty* accounts for uncertainty in the model parameters – uncertainty which captures our ignorance about which model generated our collected data. This uncertainty can be reduced given the

appropriate choice of model or enough data, and is often referred to as model uncertainty, systematic uncertainty or explained variance. More explicitly [32], training a model $\hat{y} = f(\mathbf{x})$ to approximate $y = f^*(\mathbf{x}) + \epsilon$, where the statistical noise ϵ is normally distributed $\epsilon \in \mathcal{N}(0, \sigma_a^2)$, the expected error is

$$\begin{aligned} \mathbb{E}\|y - \hat{y}\|^2 &= \mathbb{E}\|y - f^*(\mathbf{x}) + f^*(\mathbf{x}) - f(\mathbf{x})\|^2 \\ &= \mathbb{E}\|y - f^*(\mathbf{x})\|^2 + \mathbb{E}\|f^*(\mathbf{x}) - f(\mathbf{x})\|^2 \\ &= \sigma_a^2 + \sigma_e^2, \end{aligned} \quad (28)$$

where σ_a^2 and σ_e^2 are the aleatoric and epistemic errors, respectively. Cross terms in eq. 28 go to zero since ϵ is independent of y and \hat{y} . Eq. 28 highlights how epistemic uncertainties arise from a systematic difference between the true and learned models, while aleatoric uncertainties are irreducible statistical fluctuations about the true model.

Aleatoric uncertainties can be further broken down into two types: homoskedastic and heteroskedastic uncertainties. Homoskedastic uncertainties are the same for all inputs, whereas heteroskedastic uncertainties depend on the input \mathbf{x} , so $\epsilon \in \mathcal{N}(0, \sigma_a^2(\mathbf{x}))$. Almost all real data has heteroskedastic uncertainty; some examples are better measured than others. Photoelectron tracks are extremely heteroskedastic.

Uncertainty quantification in DNNs is a quickly developing field with many possible approaches. Modelling aleatoric uncertainties is simpler and a similar method is shared between approaches. This involves training the DNN to predict its own aleatoric uncertainty by enhancing the loss function. The next section, §3.4.1, provides a concrete example. Epistemic uncertainty is more difficult to quantify since it usually involves sampling over different models. Bayesian neural networks [31] do this by randomly subsampling the trained DNN weights \mathbf{w} and making multiple predictions, while deep Gaussian processes [33] or mixture density networks [32] model the epistemic uncertainty intrinsically. For practical applications, which require robust and scalable uncertainty predictions as well as high prediction accuracy, deep ensembles [34] are the state of the art [35].

3.4.1 Deep ensembles

Deep ensembles are made up of an ensemble of individual DNNs, each trained independently on the same dataset to predict the desired output features. Different random initializations of the same NN at the start of training leads to widely different prediction functions [37]. Deep ensembles exploit this property by incorporating the results of many differently initialized NNs, increasing the diversity of predictors. This not only yields higher prediction accuracy, but also allows for state-of-the-art aleatoric and epistemic uncertainty quantification [38, 35].

To model the aleatoric uncertainty in a regression task, each individual DNN composing the ensemble is trained to minimize the following negative log-likelihood loss function over all training examples i [34]

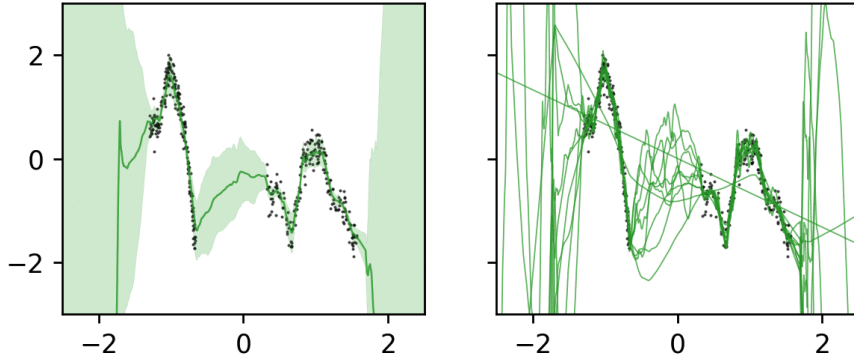


Fig. 9 Toy regression problem using a deep ensemble. Right panel shows individual DNN prediction functions. Left panel gives the estimated epistemic uncertainty of the deep ensemble. Figure from [36].

$$L(y_i | \mathbf{x}_i) = \frac{\log(\hat{\sigma}^2(\mathbf{x}_i))}{2} + \frac{\|y_i - \hat{y}(\mathbf{x}_i)\|_2^2}{2\hat{\sigma}^2(\mathbf{x}_i)}. \quad (29)$$

The DNN predicts the aleatoric uncertainty $\hat{\sigma}_a$ for each example prediction, as well as the example label \hat{y} as in standard DNN regression, eq. 21. This form of loss function assumes the aleatoric error has a Gaussian distribution, usually a reasonable approximation. In practice the DNNs are trained to predict the log variance $\hat{s}(\mathbf{x}_i) = \log(\hat{\sigma}^2(\mathbf{x}_i))$

$$L(y_i | \mathbf{x}_i) = \frac{\hat{s}(\mathbf{x}_i)}{2} + \frac{e^{-\hat{s}(\mathbf{x}_i)} \|y_i - \hat{y}(\mathbf{x}_i)\|_2^2}{2} \quad (30)$$

since this is more numerically stable [31].

During model prediction, predictions from all DNNs composing the ensemble are combined. The final predicted label is the mean over the ensemble predictions. For an ensemble with M DNNs $\hat{y}(\mathbf{x}) = (1/M) \sum_{j=1}^M \hat{y}^j(\mathbf{x})$. Similarly, the final aleatoric uncertainty is given by $\hat{\sigma}_a^2(\mathbf{x}) = (1/M) \sum_{j=1}^M \hat{\sigma}_a^2(\mathbf{x})^j$. The variance of the label predictions \hat{y}^j over the ensemble is used to model the epistemic uncertainty, resulting in total predicted uncertainty

$$\begin{aligned} \hat{\sigma}^2(\mathbf{x}) &= \hat{\sigma}_a^2 + \hat{\sigma}_e^2 \\ \hat{\sigma}^2(\mathbf{x}) &= \frac{1}{M} \sum_{j=1}^M \hat{\sigma}_a^2(\mathbf{x})^j + \frac{1}{M} \sum_{j=1}^M \left[\hat{y}^j(\mathbf{x}) - \frac{1}{M} \sum_{j=1}^M \hat{y}^j(\mathbf{x}) \right]^2. \end{aligned} \quad (31)$$

Very few DNNs are needed in the ensemble, $M \approx 5 - 10$, for the method to achieve state-of-the-art uncertainty quantification [34]. Figure 9 visualizes the epistemic uncertainty prediction for a toy regression problem. In the area far from the training

examples, the DNNs composing the ensemble strongly disagree in their predictions, giving a high epistemic uncertainty.

In multitask problems, the aleatoric uncertainties can actually replace the hyperparameters that determine the relative importance of individual tasks in the loss function [39]. The DNNs automatically tune their own hyperparameters by learning the aleatoric uncertainties during training.

4 Neural networks for track reconstruction

Track reconstruction is an image input, multi-output regression problem §2.1. From a machine learning perspective, this problem is perfectly suited to a multitask CNN approach. CNNs learn translation invariant representations §3.2, an important constraint for working with image inputs, and a multitask approach is not only more computationally efficient but also likely to improve the individual task performances §3.3. In practice, a deep ensemble of CNNs is better than a single CNN for track reconstruction:

- An ensemble of models provides better predictive performance and generalization, §3.4.1.
- Trustworthy uncertainty quantification will be important for the next step: polarization estimation, §5.
- Deep ensembles provide automatic hyperparameter tuning of individual task importance for the multitask loss function, §3.4.1.

Some authors [11] have argued for an end-to-end deep learning approach to polarization estimation that combines track reconstruction with polarization estimation into one huge supervised learning problem to go directly from a set of tracks to source polarization. Although initially attractive because one could use source polarization directly as the neural network learning signal, avoiding the need for simulated detector events, this approach is difficult. Training would be costly (perhaps infeasible) since many individual events would be required for a single training example. Moreover, the observer may wish to adjust partitioning of the data set into e.g., time, energy, and spatial subsets, with differing polarization. The best partition will often not be obvious before the analysis starts, thus combining the properties of individual tracks allows a more efficient exploration of binning options. Finally detector-dependent artifacts can best be handled from individual tracks (with fine spatial positioning), rather than point spread function (PSF) weighted groups of tracks.

The neural network approach presented in this section and the next splits imaging X-ray polarimetry into two mostly separate data analysis steps, as described in §2. This section describes step 1, track reconstruction §2.1: extracting the relevant features (emission angles, absorption points, energies, and their associated uncertainties) from photoelectron track images as well as possible. This section follows the current state-of-the-art deep ensemble approach for track reconstruction taken by [12, 40].

4.1 Dataset

In any supervised learning problem, a dataset of labelled examples is an essential first step, without this, training a model is not possible. In the case of track reconstruction, one needs photoelectron track images labelled with known photoelectron

2D emission directions, 2D absorption points on the detector grid and scalar photon energies. Unfortunately, these labelled events cannot be collected from the detector in the lab so must be simulated. Most telescope detectors have event simulators to study detector performance; for example, IXPE has a GPD Monte Carlo event simulator built in GEANT4 [41] as part of its software suite. In any application where simulated data is used to train a machine learning model, care must be taken to make sure the simulated training data truly matches the real data the algorithm is likely to see. Systematic difference between the two is known as covariate shift and can cause reductions in prediction accuracy and uncertainty quantification on real data compared to the simulated data. In the case of IXPE, simulated events have been rigorously verified to match with real lab-collected detector events [8] at the expected telescope settings, including photons absorbed outside the gas volume §2.1.4. For IXPE trained neural network approaches, the performance on polarization estimation for simulated events generalizes to real detector data [12, 40], suggesting minimal covariate shift.

Once labelled photoelectron track events can be reliably simulated for the X-ray polarimeter of choice, one needs to choose the amount and type of training and testing data to be simulated. For IXPE’s GPDs, [12, 40] use a training set of approximately 3.3 million tracks simulated uniformly for an unpolarized source between 1 - 11keV, for performance across a broad spectrum. They keep separate an additional 200k simulated tracks for validation and testing. The energy distribution of the training set does not make much difference to track reconstruction or polarization prediction, so long as no energies are greatly underrepresented nor labeled energy bins significantly larger than the energy resolution. An unpolarized source is simulated to make sure emission angles are uniformly distributed, so as not to cause any learning bias. The next section describes additional data pre-processing necessary for non-square grids.

In general, the larger the DNN (in terms of parameters and layers), the more data is required for training.

4.2 Geometric bias

Imaging polarimeters like IXPE use hexagonal pixels for two reasons:

1. Hexagonal arrays are the densest possible 2D packing, meaning more pixels per unit area thus improved track resolution.
2. Hexapolar grid effects are orthogonal to the quadrupolar polarization signal.

The discrete nature of the pixel grid means that track elongations and emission directions are better resolved along some axes of symmetry than others, potentially creating preferred directions for reconstruction. This bias would only be visible when the pixel size is significant compared to the track size, i.e., for small, low energy

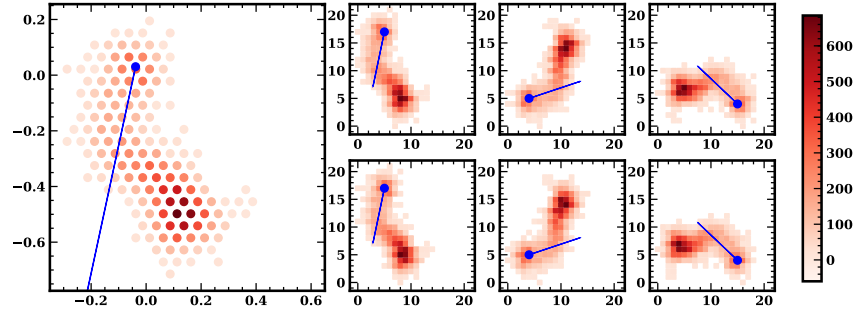


Fig. 10 Example square conversions of a 6.4 keV hexagonal track (left panel). The six panels to the right show shifts along the 120° GPD axes; shifting odd rows (upper) or even rows (lower). For each hexagonal track, CNNs are fed column-wise pairs of square conversions, along with the energy, absorption point (blue dot) and initial photoelectron direction (blue line) as labels.

tracks. The hexagonal grid used in the IXPE GPDs is designed to minimize any polarization systematics arising from these biases. Although not all imaging polarimeters use hexagonal pixels, this section describes some potential neural network solutions for dealing with non-square pixels.

A hexagonal grid is not natively compatible with standard CNN implementations for square matrices. It is possible to transform from a hexagonal to a square grid, however a naive transformation can lead to polarization systematics. Polarization estimation, step 2 §6, expects unbiased estimators for the emission angles. CNNs learn to prefer some emission angles over others, possibly interfering with the polarization signal. In a naive hexagonal to square conversion, emission angle bias happens partly because the CNN convolutional kernels are not spatially equivariant in hexagonal space. A possible solution is to work directly with hexagonal convolutions [42], but [12] describe an expedient method.

4.2.1 Hexagonal to square conversion

It is possible to leverage existing square CNN architectures by using an appropriate polarization-systematic-free hexagonal to square conversion. From [12]:

There are two main ways of converting between hexagonal and square grids: interpolation and pixel shifting. Interpolation places a fine square grid on top of the hexagonal image and interpolates. Interpolation should generally be avoided since it adds noise to the raw data and is not easily reversible. Pixel shifting rearranges pixels by shifting alternate rows and then rescaling [42] ... To avoid any polarization bias from the conversion, we pixel shift each track along each of the six hexagonal axes. Hexagonal tracks are rotated so that rows align horizontally (this can be done in three different ways, separated by 120°), then alternate rows are shifted (this can be done in two different ways, left and right) so that the track resembles a rectangular grid. We convert the rectangular grid into a square image by defining the leftmost track pixel and bottom track pixel as the left edge and the base of the image respectively. A square image size of 50x50 pixels is used to fit all track sizes for

energies up to 11 keV. Since square track images are defined independently of the absolute hexagonal coordinate values, the initial hexagonal track rotation can be performed about any axis.

A single hexagonal track thus produces six square conversions, fig. 10, two for each 120° angle. A single training example for the CNNs is formed by stacking the corresponding square conversion pair, similarly to color channels in a *rgb* image CNN problem – in this case with only two channels. At test time all 3 pairs are evaluated by the CNNs and the predicted angles are rotated back to their original direction.

This approach artificially approximates spatial equivariance of CNN convolution kernels in the hexagonal space and removes all relevant prediction bias on emission angles θ (and later p_0, θ_0) introduced when converting from hexagonal to square coordinates.

4.3 Deep ensemble setup

The specific CNN model that will compose the deep ensemble has to be chosen before any training begins. CNN architectures are usually compared by their performance on natural image classification datasets, like imagenet (224x224 pixels) or CIFAR (32x32 pixels) [43]. The residual network architecture (ResNet [28]) described in §3.2 and related architectures like DenseNet [44] became the state of the art for image processing benchmarks in 2017. [12] use the ResNet-18 CNN architecture to compose the deep ensemble; this has 18 total layers. The results shown in this section are using ResNet-18s, but any DNN appropriate for images could be used. Indeed, as of 2020, transformer architectures [45] have supplanted residual networks as the latest state of the art in image processing.

During training, each CNN minimizes a loss function. Since track reconstruction is a multitask problem, the loss function will include multiple terms for emission angle loss, absorption point loss, and energy loss. The DNNs will also predict the aleatoric uncertainty on each of its estimates. Each task loss function term is considered separately and they are all combined at the end of the section.

Emission angles. Since emission angles are periodic, a Gaussian aleatoric uncertainty is inappropriate. Instead of the Gaussian negative log-likelihood (NLL) used in §3.4.1, the von Mises (VM) distribution NLL is a better loss function choice. This is the maximum entropy distribution for circular data with specified expectation. For a random 2D unit vector \mathbf{x}

$$\text{VM}(\mu, \kappa) \equiv p(\mathbf{x}|\mu, \kappa) = \frac{\exp(\kappa \mu^T \mathbf{x})}{2\pi I_0(\kappa)}. \quad (32)$$

where I_0 is the modified Bessel function of the first kind. This can also be considered a 1D distribution over the polar angle θ of vector \mathbf{x} :

$$\text{VM}(\theta_\mu, \kappa) \equiv p(\theta|\theta_\mu, \kappa) = \frac{\exp(\kappa \cos(\theta - \theta_\mu))}{2\pi I_0(\kappa)}. \quad (33)$$

The VM distribution is parameterized by concentration parameter κ ; for large κ the VM converges to a Gaussian with variance $\sigma^2 = 1/\kappa$. For small κ the VM converges to a uniform distribution. This more appropriately reflects the distribution of predictions $\hat{\theta}$, which are clearly periodic.

Predicting scalar periodic values directly is tricky for standard DNNs, so the emission angle θ can be parameterized as a 2D vector $\mathbf{v} = (\cos 2\theta, \sin 2\theta)$. Now the DNNs attempt to predict the 2D vector $\hat{\mathbf{v}}$ [12]. Only $-\pi/2 \leq \hat{\theta} < \pi/2$, as opposed to $-\pi \leq \hat{\theta} < \pi$, are required for polarization estimation since the EVPA $-\pi/2 \leq \theta_0 < \pi/2$, i.e. polarization is a quadrupolar signal. However, the full 2π emission angle can be useful when dealing with systematics and extended source analysis. If predicting the full 2π emission angle is necessary, an additional loss term for $\mathbf{v}_2 = (\cos\theta, \sin\theta)$ can be included [12]. Predicting only \mathbf{v}_2 gives very poor results for low energy tracks because of direction ambiguity, and should be avoided.

Each DNN model in the ensemble computes the emission angle loss for track image \mathbf{x} with true direction θ as the VM NLL

$$L_\theta(\mathbf{v} | \mathbf{x}) = -\hat{\kappa}^a(\hat{\mathbf{v}} \cdot \mathbf{v}) + \log I_0(\hat{\kappa}^a). \quad (34)$$

where $\hat{\kappa}^a$ are the DNN predicted aleatoric VM uncertainties. The epistemic uncertainties can also be assumed to follow $\text{VM}(0, \kappa^e)$; κ^e can be estimated from the output of a deep ensemble with M DNNs $\{\hat{\theta}_j\}_{j=1}^M$ using the appropriate maximum likelihood estimator:

$$\bar{R}^2 = \left(\frac{1}{N} \sum_{j=1}^M \cos 2\hat{\theta}_j \right)^2 + \left(\frac{1}{N} \sum_{j=1}^M \sin 2\hat{\theta}_j \right)^2 \quad (35)$$

$$\frac{I_1(\hat{\kappa}^e)}{I_0(\hat{\kappa}^e)} = \bar{R}, \quad (36)$$

The total VM predictive error on each track angle $\hat{\theta}$ is approximated by summing the aleatoric and epistemic variances, as in §3.4.1:

$$\frac{1}{\hat{\kappa}} = \frac{1}{\hat{\kappa}^e} + \frac{1}{M} \sum_{j=1}^M \frac{1}{\hat{\kappa}_j^a} \quad (37)$$

Absorption points. To predict the (x,y) photon absorption point coordinates given a track image \mathbf{x} , a simple Gaussian NLL loss function suffices:

$$L_{\text{abs}}(x, y | \mathbf{x}) = \frac{1}{2} \log \hat{\sigma}_{\text{abs}}^2 + \frac{1}{2\hat{\sigma}_{\text{abs}}} \left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \right\|^2. \quad (38)$$

As in eq.30, the DNN actually predicts the log variance for numerical stability. If desired, the epistemic error for the absorption point predictions can be calculated in the same way as §3.4.1.

Photon energy. Similarly to the absorption point loss eq.38, the energy loss can be stated as a simple MSE with the aleatoric error included:

$$L_E(E | \mathbf{x}) = \frac{1}{2} \log \hat{\sigma}_E^2 + \frac{1}{2\hat{\sigma}_E} \|E - \hat{E}\|^2, \quad (39)$$

and like the absorption points, the epistemic error can be calculated over the deep ensemble as in §3.4.1.

Predicting the energy, while simple at first sight, is more tricky than absorption points because of tail tracks. The presence of tail tracks, §2.1.4, can cause the DNNs to produce biased energy estimates \hat{E} with this loss function. Tail tracks look like lower energy tracks than their true energy label suggests, and this can lead to two possibilities:

- If tail tracks are difficult to distinguish from peak tracks this will cause the DNN to skew its overall predictions upward in energy to minimize eq.39.
- If low energy tail tracks are difficult to distinguish from high energy tail tracks, then most tail tracks will be assigned the mean energy of the training distribution as this minimizes the expected value of eq.39. This is usually undesirable for astrophysical spectra.

In principle these two effects could be remedied in a number of ways, for example by adjusting the loss function eq.39 [12] or by a careful look at the predicted energy uncertainty (tail tracks should have much higher uncertainties). In practice, since tail tracks also affect emission angle reconstruction and by extension polarization estimation, the recommended approach is to remove tail tracks from the analysis to the greatest extent possible [40]. This works well only if tail tracks are distinguishable from peak tracks. For IXPE GPD events this is fortunately the case, but a different approach may be better for other imaging X-ray polarimeters. In §4.4 we describe a DNN approach to identify and remove tail tracks.

The full DNN loss function to be minimized is now

$$L(\theta, x, y, E | \mathbf{x}) = L_\theta + L_{\text{abs}} + L_E + \alpha \|\mathbf{w}\|^2, \quad (40)$$

where the final term is an L2 regularization on the DNN weights controlled by hyperparameter α , to prevent overfitting the training data. The predicted aleatoric uncertainties act as automatically tuned hyperparameters for the remaining loss terms [39]. Given a track image input \mathbf{x} , an individual DNN outputs an 8 dimensional vector: $(\hat{\nu}, \hat{\kappa}^a, \hat{x}, \hat{y}, \hat{\sigma}_{\text{abs}}, \hat{E}, \hat{\sigma}_E)$.

4.4 Removing tail tracks

Photoelectron tracks from events converting outside the main detector gas are known as tail tracks, §2.1.4. For both DNNs and classical methods they cause problems

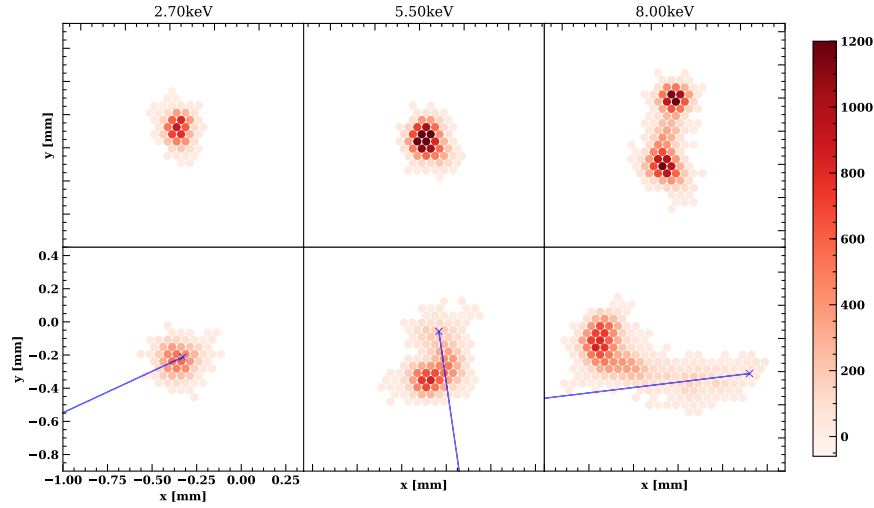


Fig. 11 Example tail events (top row) and peak events (bottom row) for three different recovered energies. All plots follow the same spatial and color scale. These tail events convert in the GEM. Color denotes charge deposited in a given detector pixel. Note that tail events are more compact, short and with high charge density at maximum.

with photon energy prediction, discussed in §4.3. Furthermore, tail tracks have comparatively high emission angle prediction error (low polarization sensitivity §2.1.4) and §4.6 will show that this error is not properly captured by deep ensemble predictive uncertainties. One possible approach to remedy the tail track issue is to attempt to excise tail tracks from the data analysis entirely.

For IXPE’s GPDs, tail tracks differ in morphology from peak events converting in the gas. Fig. 11 compares events with the same linearly recovered energy (largely determined through the summed pixel counts, the summed energy deposition in the gas). Since the GEM and window material have a lower mean free path for photo-electrons, detected tail events are typically due to photo-electrons ejected close to the GEM/window normal. Their tracks are thus more compact, with higher counts/pixel for the same recovered energy. Window events have larger drift diffusion than GEM conversions. Distinguishing peak and tail tracks can be formed as a computer vision classification problem, so a DNN would be a good model choice and could be trained on simulated events to recognize the differences.

A good approach for peak vs. tail track classification would be to include the classification task into the existing track reconstruction deep ensemble, §4.3. This would likely improve performance over training a separate model and at first glance would only require an additional loss term in eq.40. However, this approach would not fix the problem of improper energy predictions and emission angle uncertainties unless more complex cross terms to eq.40 were also added. The cross terms would

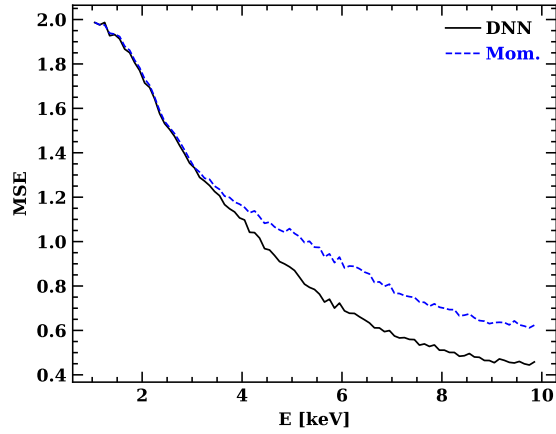


Fig. 12 Mean squared error on emission angle prediction as a function of photon energy, measured using eq. 41, for the classical moment analysis and deep ensemble.

ensure, for example, that tracks likely to be tail events have higher energy uncertainty. Opting for a simpler approach, [40] trained a separate peak vs. tail classifier DNN and used only peak tracks to train the track reconstruction deep ensemble. The [40] separation approach is described in this section, but a unified deep ensemble approach to tail vs peak classification would likely work better and is discussed as a future direction in §6.

Setting up a peak vs. tail DNN classification model is the same as setting up a multitask deep ensemble §4.3, only now the loss function to be minimized is different. In binary classification tasks, DNNs minimize the cross-entropy loss, eq.22. Given a single input, a DNN outputs a single scalar between 0 and 1 that represents the probability an input is the first class. [40] use exactly the same ResNet-18 architecture and data preprocessing, §4.1–4.2, when training the peak vs. tail classifier DNN; the only notable differences are the loss function and data labels.

4.5 Training and ensemble selection

Individual DNN training procedures should follow the guidelines in §3.1.1, §3.1.2. Specifics will depend on the particular DNN architecture chosen. CNNs for computer vision problems train particularly well using stochastic gradient descent with momentum [46] as the optimization algorithm. Normalizing the inputs and outputs before training typically helps improve convergence. From [12]:

The ResNet-18 architecture trains in a reasonable amount of time (~ 15 hours for 150 epochs on 4 Nvidia Titan GPUs, using a batch size of 2048). Before training, we normalize the track images, subtracting the pixel-wise mean from each track image and dividing by the pixel-wise standard deviation (where the mean and standard deviation are calculated over the full training set). The track energy and absorption point labels are similarly processed. Normalizing the training data helps prevent vanishing and exploding gradients during the NN training procedure and lead to faster convergence. We use stochastic gradient descent

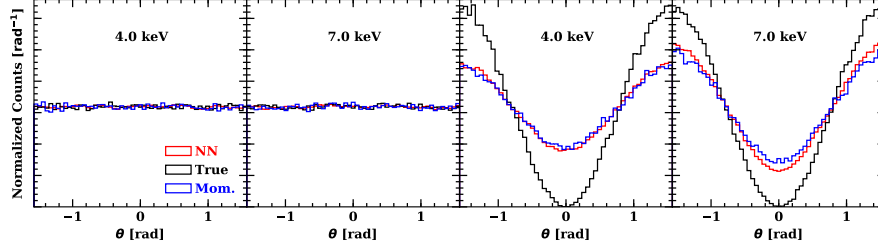


Fig. 13 Emission angle recovery for unpolarized, $p_0 = 0$, (left two panels) and 100% polarized, $p_0 = 1$, (right two panels) simulated IXPE data for 4.0 and 7.0 keV. The true photoelectron angle distribution is shown in black; standard moment analysis reconstruction is in blue and deep ensemble in red.

with momentum as our optimizing algorithm, typical in computer vision tasks [46], with a stepped decaying learning rate starting at 0.01. We choose batch sizes of 512, 1024, 2048 tracks ... We tune the hyperparameters to minimize the MSE for the validation set.

Once multiple DNNs are trained with optimized hyperparameters, individual DNNs should be chosen at random to compose the ensemble. This ensures accurate epistemic uncertainty estimates.

4.6 Performance

The performance of deep ensembles in track reconstruction is compared to the classical moment analysis §2 on simulated IXPE data. The results shown are for peak events only unless otherwise specified.

Emission angles. Two important measures of performance should be evaluated for emission angle recovery: the accuracy of recovery and the quality of uncertainty estimates. The accuracy of emission angle recovery can be measured, for example, by the MSE on a test dataset

$$\frac{1}{N} \sum_{i=1}^N \|\mathbf{v}_i - \hat{\mathbf{v}}_i\|^2. \quad (41)$$

Fig. 12 gives the above MSE as a function of the true track energy on an unpolarized IXPE test dataset for both classical moment analysis and deep ensemble approaches. At low energies, deep ensembles fail to improve significantly over the moment analysis; when resolution is low, track image moments contain all the relevant information.

It is also essential to confirm that the estimates $\hat{\theta}$ are unbiased. Fig.13 gives the distribution of emission angles $\hat{\theta}$ recovered by the deep ensemble and moment analysis for an unpolarized and polarized source at various energies. For both methods, emission angle estimates show no systematic biases, and at higher energies for the

polarized source the improved signal recovery of the deep ensemble is visible. As an additional check, the bottom row of fig. 14 displays the distribution of $\hat{\theta} - \theta$ for the moment analysis and deep ensemble at three different true energies. Both the moment analysis and deep ensemble have symmetric error distributions centered at zero for all energies. At higher energies, the lower dispersion and thus higher accuracy of the deep ensemble is again clear.

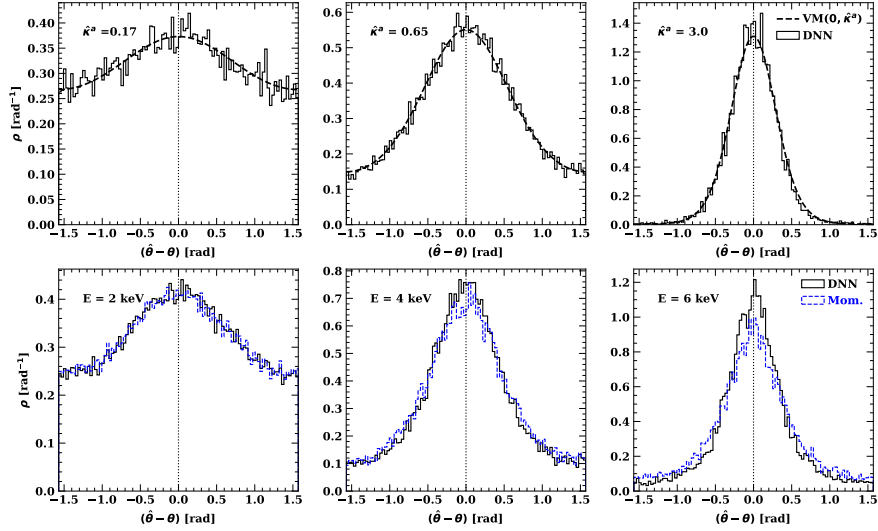


Fig. 14 *Top*: Histograms of the DNN emission angle residual for three different DNN predicted aleatoric VM concentration parameters $\hat{\kappa}^a$. The DNN predicted VM error distributions are overlotted. The DNN is able to correctly predict its emission angle aleatoric error distribution. *Bottom*: Histograms of emission angle residuals at three different energies for DNNs and the moment analysis. All methods are unbiased (centered at zero) and at higher energies the DNNs have better recovery of the true emission angle θ , as evidenced by the larger peak heights.

A simple way to assess the deep ensemble predicted uncertainty estimates $\hat{\kappa}$ is to consider the distribution of $\hat{\theta} - \theta$ for fixed $\hat{\kappa}$. According to the assumptions in §4.3, $\hat{\theta} - \theta$ should follow a $VM(0, \hat{\kappa})$ if the DNNs have been properly trained. In the top row of fig. 14, the distributions of $\hat{\theta} - \theta$ for three different fixed $\hat{\kappa}$ are displayed with $VM(0, \hat{\kappa})$ overlaid. In all cases, the distributions match very closely. Epistemic errors are negligible for the vast majority of peak tracks, suggesting good model choice and training convergence.

It is also instructive to visualize the distribution of deep ensemble predicted uncertainties as a function of energy, fig. 15. For higher energies a wider range of concentration parameters are predicted and the $\hat{\kappa}$ peak moves progressively higher. Below 2keV, barely any polarization signal is recoverable with IXPE. Fig. 15 highlights the strong heteroskedasticity in imaging X-ray polarimetry.

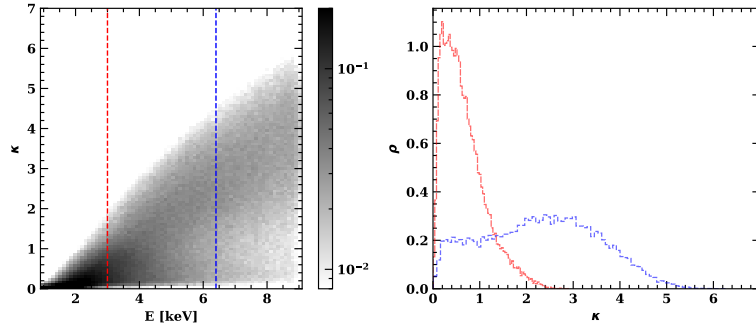


Fig. 15 Distribution of deep ensemble predicted concentration parameters $\hat{\kappa}$ across the IXPE energy spectrum (left). The right-hand plot shows the $\hat{\kappa}$ distribution for two specific energies 3.0 keV (red), 6.4 keV (blue). High concentration parameter κ means a low predictive uncertainty.

For a deep ensemble trained on both peak and tail tracks, the predicted uncertainties for tail tracks are untrustworthy. Comparing the error distributions in fig. 16 to 14, tail track error is not captured by deep ensemble uncertainty predictions, even with the epistemic uncertainty included. Furthermore, at low energies, tail track emission angle estimates are potentially biased.

Tail vs. peak classification. For tail vs. peak classification, there is no equivalent classical method to compare the DNN performance, so this section lists the classification results achieved by [40]. They use the predicted tail probability as an event threshold cut. Fig. 17 shows the effectiveness of different tail probability cuts. If all events with a DNN predicted tail probability higher than 70% are removed, a loss of only 3% of true peak events is incurred while 66% of true tail events are successfully removed. Using a DNN classifier, only a small loss to signal events need be incurred

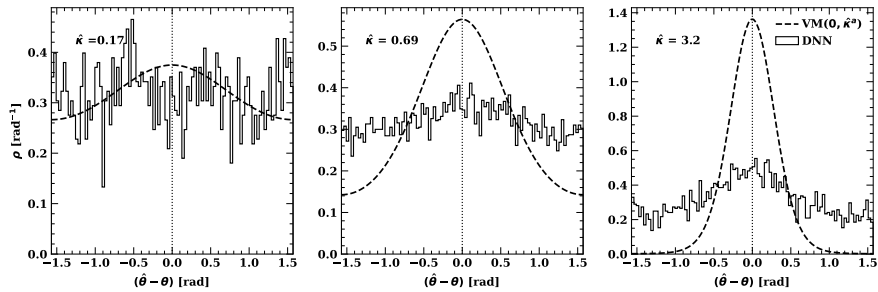


Fig. 16 Histograms of the emission angle residual at three different DNN predicted concentration parameters $\hat{\kappa}$ for tail events only. The predicted DNN error distributions are overplotted. When tail events are treated as if they were normal events, the DNNs do not learn to predict the appropriate uncertainties.

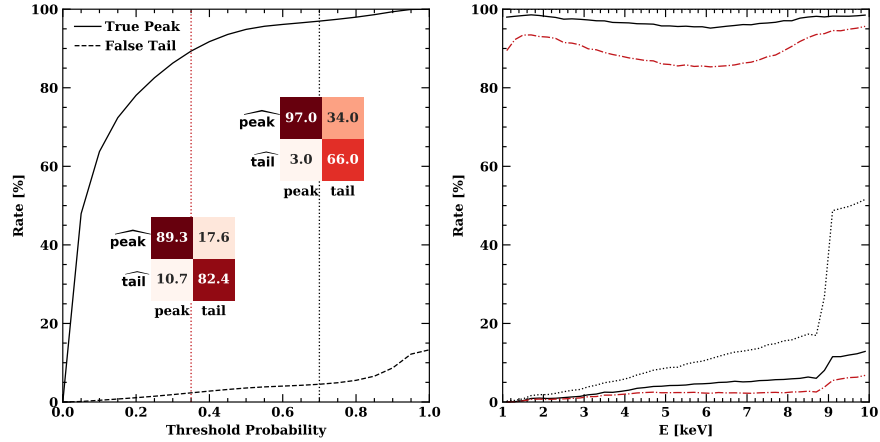


Fig. 17 *Left:* The solid curve shows the fraction of peak events retained as a function of the tail probability cut, while the dashed curve shows the fraction of the cut sample from the remaining tail events. Insets show the confusion matrices, normalized by column, for our adopted 70% cut and a 35% cut. *Right:* The top curves show the peak retention as a function of energy (black solid – 70% cut, red dot-dash – 35% cut). Below, the dotted curve shows the uncut fraction of the sample due to tail events, while the lower black and red curves show the residual tail pollution (70% and 35% cut, respectively). Depending on how harmful tail events are to the desired measurements, different cut levels could be appropriate.

to remove the majority of harmful tail events. Fig. 17 also shows how remaining tail and peak events are distributed in terms of true energy after a tail probability threshold cut. Remaining tail events come from all energies in proportion to their original population, while misidentified peak events come from middle energies where peak and tail events overlap most.

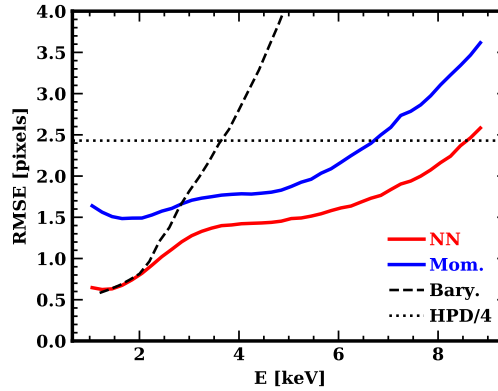
At least for IXPE GPDs, it is certainly possible to meaningfully distinguish between tail and peak events. The upcoming section on photon energy recovery and §6 demonstrate how tail cuts based on DNN probabilities can be used to improve energy and polarization resolution.

Absorption points. Absorption point accuracy can be evaluated using the MSE. In fig. 18 the classical moment analysis and deep ensemble predictions are compared at different photon energies. For IXPE, the track barycenter provides a better prediction of the absorption point for very low energies – the deep ensemble recovers this transition while the moment analysis does not.

For current generation X-ray polarimeters like IXPE, the telescope PSF is much larger than any absorption point discrepancy, so absorption point accuracy is less essential than emission angle or photon energy accuracy. However, future missions with higher spatial resolution may require very accurate absorption point estimates for detailed imaging of extended sources.

Photon energy. Energy recovery can be adversely affected by tail events. While the MSE in predicted energy is lower for deep ensemble methods compared to classical

Fig. 18 Photon absorption point localization accuracy using the root mean squared error. The deep ensemble predictions (red) do appreciably better than the moment analysis and matches a barycenter estimate at the lowest energies. For IXPE, all methods are adequate, as localization is much better than the size of the point source image produced by IXPE's mirrors. One quarter of IXPE's half power diameter is illustrated (black dotted line).



linear charge recovery, these results should be treated with caution. Much of this improvement is specious, coming from a mistreatment of tail events. Fig.19 plots the IXPE recovered energy histograms of the classical linear method (blue), a deep ensemble trained including tail events (black) and a deep ensemble trained on only peak events with tail events removed at a 70% peak vs. tail classifier probability (red). Around the true energy peak all methods perform similarly, with a slight advantage for the DNN based methods. As the energy increases, all methods show a low energy tail in reconstructed energy which is produced by tail events. The standard deep ensemble that does not differentiate peak and tail tracks also shows a high energy tail in reconstructed energy. This tail arises from low energy peak events 'looking' like higher energy tail events, so pushing some events up to higher energy tends to minimize the MSE. As discussed in §4.3, high energy tails in reconstructed energy are undesirable for astrophysical spectra. Training a deep ensemble with only peak events prevents this high-energy tail behavior.

Cutting events based on the predictions of the peak vs. tail classifier successfully reduces tails in the recovered energy histograms. A tail event cut improves the energy resolution at the cost of potentially losing some peak events and thus some polarization signal. §5.3 will discuss and quantify the polarization signal loss due to tail event cuts.

The energy prediction accuracies (mean absolute error and FWHM) for all three methods as a function of photon energy are given in fig. 20. Removing tail events provides a significant improvement to the energy prediction accuracy.

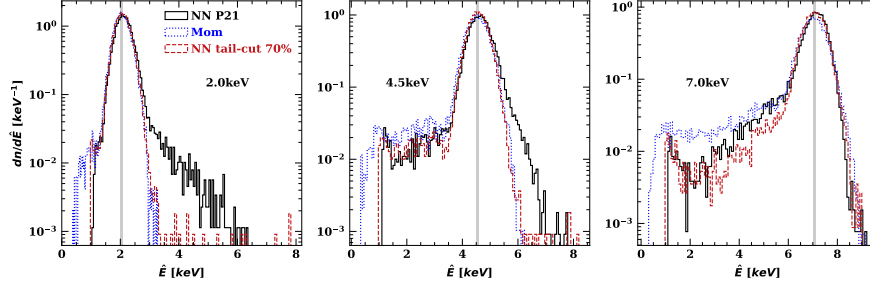


Fig. 19 Response for three true energies. Note that while the P21 analysis suppressed the large tail rate seen in the Moments processing, events leaked to a high energy tail for medium to low true energies. Our morphological tail cut further suppresses the GEM/window tail, avoids the high energy leakage and achieves comparable or better peak width.

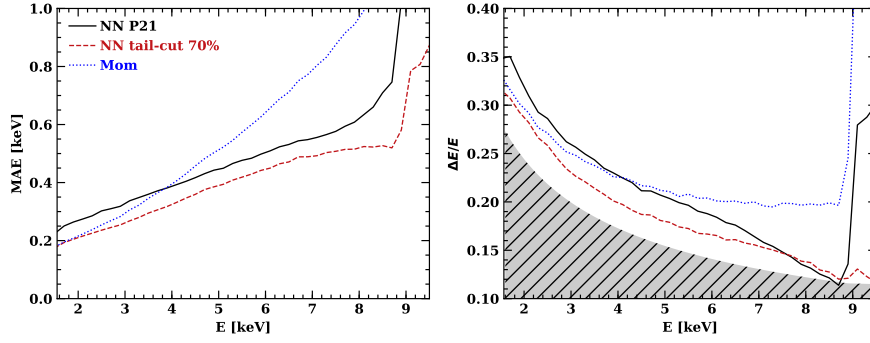


Fig. 20 *Left*: Mean absolute error in predicted energy as a function of true energy. *Right*: FWHM ($3.46 \times$ Median Absolute Deviation) of predicted energy distribution for a source at energy E . The grey band marks the limiting energy resolution in the purely-exponential multiplication regime. NNs with 70% tail cut performs better on every metric, more accurate per track and tighter resolution at all energies. All methods suffer from the large increase of tail events above 9 keV.

5 Neural networks for polarization estimation

§2.2 explained how to best estimate polarization parameters (p_0, θ_0) given a group of reconstructed photoelectron emission angles $\{\hat{\theta}_i\}_{i=1}^N$. This section explains how to best estimate polarization parameters given a group of reconstructed photoelectron emission angles *and their associated uncertainties* $\{\hat{\theta}_i, \hat{\kappa}_i\}_{i=1}^N$. These uncertainties can vary for different individual emission angles; the measurements are heteroskedastic. Emission angles with larger uncertainties should contribute less to the final polarization parameter estimates. In theory, this section need not use DNNs and the methods described can be used for any track reconstruction method that provides (trustworthy) emission angle uncertainties; the uncertainties can be parameterized by any appropriate distribution. In practice, the predicted emission angle uncertain-

ties come from deep ensembles and the uncertainties are assumed to follow a von Mises family of distributions.

5.1 Modulation factor

To develop a polarization estimation method that incorporates emission angle uncertainties, it is essential to specify how these uncertainties affect the recovered emission angle distribution eq. 2. For a source with polarization parameters (p_0, θ_0) , the true photoelectron emission angles follow the distribution eq. 2, while the recovered or measured emission angles follow eq. 3. The difference between the two is captured by the modulation factor μ . The modulation factor summarizes the effect of uncertainty in individual emission angle measurements.

Consider a single emission angle measurement $\hat{\theta}_i$. Since track reconstruction methods are imperfect and contain sources of error (even DNN based ones as was shown in §4), $\hat{\theta}_i$ can be considered a random variable:

$$\hat{\theta}_i = \theta_i + \epsilon_i \quad (42)$$

where θ is the true emission angle, which follows the distribution eq. 2, and the measurement error ϵ_i is a random variable with

$$\mathbb{E}[\epsilon_i] = 0, \text{Var}[\epsilon_i] = \sigma_i^2, p(\epsilon_i = 0) = p(\epsilon_i = 2\pi) \quad (43)$$

This assumes the $\hat{\theta}_i$ estimate is unbiased; reasonable given both the deep ensemble and moment analysis in §4.6 are for the most part unbiased estimators. The specific distribution for ϵ_i will depend on the track reconstruction method, but all distributions should follow the properties in eq. 43: be unbiased, have a finite variance, and be periodic, since $\hat{\theta}$ itself is periodic.

For any ϵ_i distribution with the above properties, it is possible to find the distribution for $\hat{\theta}_i$ as the convolution of the θ_i, ϵ_i distributions:

$$p(\hat{\theta}_i | p_0, \theta_0, \sigma_i^2) = \frac{1}{2\pi} (1 + \mu_i p_0 \cos[2(\hat{\theta}_i - \theta_0)]), \quad (44)$$

where $0 \leq \mu_i \leq 1$, $\mu_i(\sigma_i^2)$ and $\mu_i(\sigma_i^2 = 0) = 1$. In other words, the distribution of estimators $\hat{\theta}_i$ are the same as the distribution of the true values θ_i but with a reduced individual modulation factor $\mu_i(\sigma_i^2)$ that depends on the measurement error σ_i^2 . The measurement noise will blur the sinusoidal modulation signal by a factor μ_i for the specific event i .

The final distribution for a set of recovered emission angles $\{\hat{\theta}_i\}_{i=1}^N$ can now be simply formulated as an equal mixture model of the distributions of all individual recovered emission angles, eq.44:

$$p(\hat{\theta} | p_0, \theta_0) = \frac{1}{N} \sum_{i=1}^N \frac{1}{2\pi} (1 + \mu_i p_0 \cos[2(\hat{\theta} - \theta_0)]), \quad (45)$$

$$= \frac{1}{2\pi} (1 + (\frac{1}{N} \sum_{i=1}^N \mu_i) p_0 \cos[2(\hat{\theta} - \theta_0)]), \quad (46)$$

$$= \frac{1}{2\pi} (1 + \mu p_0 \cos[2(\hat{\theta} - \theta_0)]). \quad (47)$$

Thus, the final modulation factor of a set of emission angles with varying individual measurement errors $\{\hat{\theta}_i, \sigma_i^2\}_{i=1}^N$ is an average over all individual modulation factors

$$\mu = \frac{1}{N} \sum_{i=1}^N \mu_i(\sigma_i^2), \quad (48)$$

where individual modulation factors each depend on the individual measurement errors σ_i^2 .

For a specific example of an individual modulation factor, consider the case where ϵ_i is VM($0, \kappa_i$) between $[0, \pi]$. This is the uncertainty distribution learned by the deep ensembles in §4.

$$p(\epsilon_i) = \frac{1}{2\pi I_0(\kappa_i)} \exp(\kappa_i \cos 2\epsilon_i). \quad (49)$$

This distribution meets all the assumptions stipulated in eq. 43. Evaluating the convolution of the θ (eq.2) and ϵ (eq.49) distributions, one finds

$$p(\hat{\theta}_i | p_0, \theta_0, \kappa_i) = \frac{1}{2\pi} (1 + \frac{I_1(\kappa_i)}{I_0(\kappa_i)} p_0 \cos[2(\hat{\theta}_i - \theta_0)]). \quad (50)$$

Comparing to eq. 44

$$\mu_i = \frac{I_1(\kappa_i)}{I_0(\kappa_i)}, \quad (51)$$

and the modulation factor for set of tracks $\{\hat{\theta}_i, \kappa_i\}_{i=1}^N$ is

$$\mu = \frac{1}{N} \sum_{i=1}^N \frac{I_1(\kappa_i)}{I_0(\kappa_i)}, \quad (52)$$

5.2 Weighted maximum likelihood estimator

In §2.2, it was assumed when estimating polarization parameters that the modulation factor is a constant, μ . The previous section showed how the modulation factor arises from the measurement errors of many individual emission angle estimates. If individual uncertainties are known, rather than assuming these lead to one constant μ , individual modulation factors can be directly incorporated into the likelihood function for the polarization parameters.

With individual emission angle measurement errors known, the likelihood eq. 11 can be expressed as a more informative likelihood:

$$L(\{\hat{\theta}_i\}_{i=1}^N | \mathcal{Q}, \mathcal{U}) = \prod_{i=1}^N \frac{1}{2\pi} (1 + \mathcal{Q}\mu_i \cos 2\hat{\theta}_i + \mathcal{U}\mu_i \sin 2\hat{\theta}_i). \quad (53)$$

Now each estimated emission angle $\hat{\theta}_i$ comes with its own individual modulation factor μ_i , as opposed to a global μ . This likelihood can be maximized in exactly the same way as eq. 11. Following §2.2, assuming $|\mathcal{Q}\mu_i| \ll 1$, $|\mathcal{U}\mu_i| \ll 1$, the maximum likelihood estimators for Stokes' parameters (\mathcal{Q}, \mathcal{U}) are now

$$\hat{\mathcal{Q}} = \frac{2}{\sum_{i=1}^N \mu_i^2} \sum_{i=1}^N \mu_i \cos 2\hat{\theta}_i, \quad (54)$$

$$\hat{\mathcal{U}} = \frac{2}{\sum_{i=1}^N \mu_i^2} \sum_{i=1}^N \mu_i \sin 2\hat{\theta}_i. \quad (55)$$

Following the notation in [14] and [40], one can define the individual event weights

$$w_i = \mu_i \quad (56)$$

and the total modulation factor (now a weighted average)

$$\mu = \frac{\sum_{i=1}^N w_i \mu_i}{\sum_{i=1}^N w_i} = \frac{\sum_{i=1}^N w_i^2}{\sum_{i=1}^N w_i}. \quad (57)$$

Then eqs.54,55 are

$$\hat{\mathcal{Q}} = \frac{2}{\mu \sum_{i=1}^N w_i} \sum_{i=1}^N w_i \cos 2\hat{\theta}_i, \quad (58)$$

$$\hat{\mathcal{U}} = \frac{2}{\mu \sum_{i=1}^N w_i} \sum_{i=1}^N w_i \sin 2\hat{\theta}_i. \quad (59)$$

Now when estimating the polarization parameters, each event is weighted by its expected signal $w_i = \mu_i$. As in §2.2, [14] derive the posterior distribution for estimators eqs. 58,59. The posterior distribution in the weighted case is the same as eq.17 but now N is replaced with

$$N_{\text{eff}} = \frac{(\sum_{i=1}^N w_i)^2}{\sum_{i=1}^N w_i^2}, \quad (60)$$

the effective number of events. Because the posterior distribution has changed, now

$$\text{MDP}_{99} = \frac{4.29}{\mu \sqrt{N_{\text{eff}}}}. \quad (61)$$

Weighting events reduces the total number of events contributing to a measurement of the polarization parameters, making $\sqrt{N_{\text{eff}}}$ smaller, but it increases the modulation factor μ by more. Overall the MDP_{99} is reduced, improving the polarization measure-

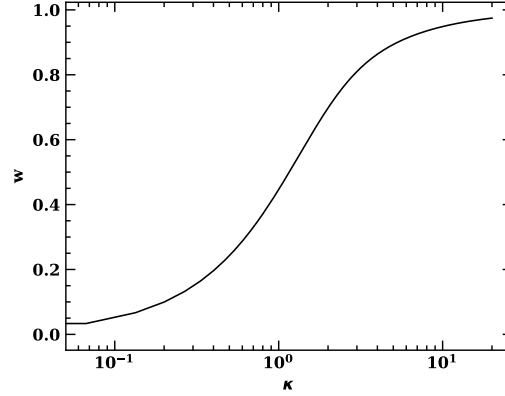


Fig. 21 Modulation factor or optimal event weight as a function of von Mises concentration parameter κ . The function is monotonic, higher concentration parameters are always weighted more.

ment. Since $w_i = \mu_i$ is the maximum likelihood estimator, this is the optimal event weighting scheme. It is also possible to arrive at this conclusion in reverse: starting with weighted estimators eq. 58,59 and MDP₉₉, eq.61, [40] prove that $w_i = \mu_i$ minimizes the MDP₉₉, or equivalently maximizes the signal-to-noise ratio.

5.2.1 Deep ensembles

A DNN approach to track reconstruction using deep ensembles gives accurate estimates of individual VM uncertainties κ_i , §4.6. If measurement uncertainties are VM, individual modulation factors are given by eq. 51. Then, for a set of deep ensemble predicted emission angles and uncertainties, $\{\hat{\theta}_i, \hat{\kappa}_i\}_{i=1}^N$, weighted maximum likelihood estimators for the polarization parameters eq. 58,59 should use

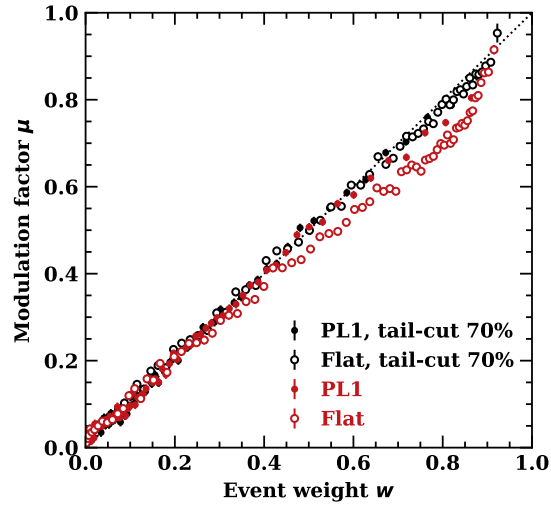
$$w_i = \mu_i = \frac{I_1(\hat{\kappa}_i)}{I_0(\hat{\kappa}_i)}. \quad (62)$$

Fig. 21 plots the optimal weighting (or equivalently, the modulation factor) as a function of κ . As expected, $0 \leq \mu_i \leq 1$ and an increasing concentration parameter κ (decreasing uncertainty) always increases the event weight.

5.3 Performance

It is now possible to compare the full polarimetry data analysis pipeline for a DNN approach with the classical moment analysis. Track reconstruction, step 1 of the pipeline, has already been compared in §4.6; this section will reveal how both track reconstruction improvements and uncertainty estimation affect polarization recovery. The quality of polarization recovery is measured by the MDP₉₉, eq.61.

Fig. 22 Measured μ as a function of deep ensemble event weight w for large test data sets of 1 – 10keV simulated events. Each μ bin contains 20,000 individual track events. Closed circles represent a PL1 source ($dN/dE = E^{-1}$) convolved with IXPE’s effective area, open circles a flat spectrum. Spectra with tail event cuts applied are in black, and full spectra in red.



5.3.1 Weights

Before comparing methods, it is important to confirm that individual modulation factors μ_i (or weights w_i) predicted by the deep ensemble for each event, eq.62, are accurate. This can be checked by observing a 100% polarized source and grouping events with same predicted μ_i , then empirically measuring the modulation factor μ of the group with eq.8. The measured and predicted μ should always match.

Fig. 22 gives the measured modulation factor μ as a function of deep ensemble predicted weights $w_i = \mu_i$ for two polarized IXPE datasets with 1.5×10^6 events. Any deviation from $y = x$ means the deep ensemble predicted uncertainties are imperfect. When most tail events are removed using a DNN tail cut, §4.4, the deep ensemble predicted weights are close to perfect and show minimal spectral dependence. If tail events are allowed to remain in the spectrum, deep ensemble weights stray from the optimal MLE for highly weighted events. This is because it is difficult for the deep ensemble to learn the appropriate uncertainties $\hat{\kappa}$ for tail events, §4.6. The effect is reduced for the PL1 spectrum where there are fewer high energy events and thus fewer tail events.

5.3.2 Comparison

To expose which parts of the DNN approach are providing the biggest MDP₉₉ improvements, table 1 gives the performance breakdown of the different components of the deep ensemble approach. The test datasets are inspired by realistic astrophysical spectra convolved with IXPE’s effective area, to give an idea of realized instrument improvements when using DNNs. Since the absolute MDP₉₉ values are specific to

Spectrum Method		MDP ₉₉ (%)
PL2	Mom.	5.36 ± 0.03
	Mom. w/ Ellip. weights	4.89 ± 0.03
	DNN	5.10 ± 0.03
	DNN w/ wts.	3.99 ± 0.02
	DNN w/ wts. 95%	3.98 ± 0.02 ←
	DNN w/ wts. 70%	4.08 ± 0.02
PL1	Mom.	4.80 ± 0.02
	Mom. w/ Ellip. weights	4.37 ± 0.02
	DNN	4.50 ± 0.02
	DNN w/ wts.	3.58 ± 0.01
	DNN w/ wts. 95%	3.57 ± 0.01 ←
	DNN w/ wts. 70%	3.83 ± 0.01

Table 1 Sensitivity analysis for two power law spectra ($dN/dE \sim E^{-N}$; PL2 for $N = 2$, PL1 for $N = 1$) each normalized to produce 10^5 2-8 keV photons when folded through IXPE's energy response. MDP₉₉ gives the sensitivities for the various reconstruction methods; smaller MDP₉₉ is better. Mom. denotes moments analysis. DNN denotes deep ensemble. Percentages denote tail cut thresholds, 95% means events with DNN predicted tail probability greater than 95% are removed.

simulated IXPE events, the relative improvements between methods are the important takeaways as these are likely to generalize to all imaging X-ray polarimeters.

Two baseline classical methods are provided in table 1, the standard moment analysis described in §2 and a weighted version. The weighted version uses event weights derived from moment analysis predicted track ellipticities. Events with higher ellipticities are weighted more in polarization estimation, eq.58,59. The specific function that transforms ellipticities to weights is optimized numerically. The ellipticities provide a rough proxy for emission angle uncertainty. Even this approximate uncertainty calculation can yield significant MDP₉₉ improvements.

The base DNN approach predicts emission angles using a deep ensemble but does not use the predicted uncertainty estimates. The improved emission angle estimates yield a marginal improvement over the moment analysis and are worse than a moment analysis with ellipticity based weights. Improvements are small since most astrophysical spectra are soft, containing many low energy photons, where DNN and moment analysis emission angle accuracies are comparable. Clearly, the main improvement to be had in X-ray polarimetry is properly accounting for emission angle heteroskedasticity by appropriately weighting events. This is born out in the weighted deep ensemble MDP₉₉ results, where a < 0.75 reduction in MDP₉₉ from the standard moment analysis is achieved for both spectra. Compared to the classical ellipticity weight approach, the deep ensemble achieves a ~ 0.82 MDP₉₉.

A 0.75 reduction in MDP₉₉ means a $0.75^2 = 0.563$ reduction in the number of counts (and observing time) required to reach the same signal-to-noise ratio. This makes a lot more science possible for an X-ray polarimetry mission, with no changes to the existing hardware.

Removing tail events using the tail vs peak classifier, §4.4, can provide a small additional sensitivity boost. However, improvements in energy resolution, §4.6, require substantially stronger tail exclusion with a threshold of 70% or less. There is a trade-off between achieving good spectral performance and minimizing the MDP_{99} because the DNN tail vs peak classifier is not perfect. When tail event cuts are applied, some peak events are lost. What tail cut threshold should be chosen will depend on the specific telescope and science goals.

It is important to note again that all of the results presented in this chapter are for *simulated* IXPE events. Fortunately, tests of the DNN techniques presented here on real IXPE GPD data have yielded similar relative improvements over classical techniques, once real detector systematics are taken into account. If the simulated detector events are close enough to real events, DNN performance will generalize.

6 Conclusion and future directions

X-ray polarimetry and its data analysis is a recently developed and rapidly moving field. Uncertainty aware DNN based approaches yield state-of-the-art signal recovery for imaging X-ray polarimeters and can reduce the required exposure time for telescopes by up to 45%. Nevertheless, the techniques and discussion presented here should not be considered the final say, but should form a solid base from which to develop better data analysis for ever-improving X-ray polarimeters. Indeed, this chapter mentions multiple promising areas for improvement. These include:

- *A unified approach to tail vs peak event classification.* Training a separate DNN to classify tail events and then cutting events based on its predictions is workable, but clearly not the best solution. A unified DNN approach minimizing a likelihood function that includes the effect of tail probability on emission angle and energy reconstruction should be the ultimate aim. In the case of IXPE, tail events originate from conversions in both the GEM and Beryllium window. These two types of tail event have different morphological properties, so it may also be useful to treat these as separate classes.
- *Geometric systematics.* Another shortcut in the DNN approach presented is using square ResNets on multiple rotations of non-square track images, §4.2. Again, this works well but is computationally inefficient during evaluation (the DNNs see six images of a single track event). A DNN model that works natively with hexagonal grids should be used for the best results. This DNN model will have the appropriate inductive biases for hexagonal geometries. A promising potential candidate would be the new state-of-the-art visual transformers [47]. These DNNs work with coordinates directly, so can be used for any kind of pixel grid.

Deep learning is also a rapidly moving field, so the specific models presented in this chapter may quickly become obsolete. However, any future DNN based data analysis will likely retain the essential ideas: track image processing, individual emission angle uncertainty estimates, full likelihood maximization, and a reduction of events external to the main detector. The author hopes that the next generation of imaging X-ray polarimeters will learn from and improve on the approaches outlined here.

Acknowledgements The author would like to thank Roger Romani for a careful reading of the text and the section editors for the opportunity to write this chapter.

References

1. H. Krawczynski, G. Matt, A.R. Ingram, R. Taverna, R. Turolla, F. Kislak, C.C.T. Cheung, A. Bykov, K. Sinha, H. Zhang, J. Heyl, N. Bucciantini, G. Madejski, T. Kallman, K.M. Jahoda, Q. Abarr, M.G. Baring, L. Baldini, M. Begelman, M. Böttcher, E. Cackett, I. Caiazzo, P. Coppi, E. Costa, J. Dexter, J.F. Fortin, C. Gammie, J.A. Gaskin, D. Giannios, P. Ghosh, A.K. Harding, D.H. Hartmann, K. Hayashida, S.G. Jorstad, P. Kaaret, T. Kitaguchi, L. Latronico, T. Maccarone, A. Marscher, H. Marshall, M. McConnell, J.M. Miller, S.L. O’Dell, F. Özel,

- T. Okajima, M. Pearce, J. Perkins, B.D. Ramsey, R.W. Romani, J.D. Schnittman, C. Sgro, P. Soffitta, H. Takahashi, T. Tamagawa, J. Tomsick, M.C. Weisskopf, G. Younes, **51**, 150 (2019). URL <http://adsabs.harvard.edu/abs/2019BAAS...51c.150K>. Conference Name: Bulletin of the American Astronomical Society
2. M. Weisskopf, *Galaxies* **6**, 33 (2018). DOI 10.3390/galaxies6010033. URL <http://adsabs.harvard.edu/abs/2018Galax...6...33W>
 3. R. Bellazzini, F. Angelini, L. Baldini, A. Brez, E. Costa, G.D. Persio, L. Latronico, M.M. Massai, N. Omodei, L. Pacciani, P. Soffitta, G. Spandre, in *Polarimetry in Astronomy*, vol. 4843 (International Society for Optics and Photonics, 2003), vol. 4843, pp. 383–393. DOI 10.1117/12.459381. URL <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/4843/0000/Novel-gaseous-x-ray-polarimeter-data-analysis-and-simulation/10.1117/12.459381.short>
 4. R. Bellazzini, G. Spandre, M. Minuti, L. Baldini, A. Brez, L. Latronico, N. Omodei, M. Razzano, M.M. Massai, M. Pesce-Rollins, C. Sgró, E. Costa, P. Soffitta, H. Sipila, E. Lempinen, *Nuclear Instruments and Methods in Physics Research A* **579**, 853 (2007). DOI 10.1016/j.nima.2007.05.304. URL <http://adsabs.harvard.edu/abs/2007NIMPA.579..853B>
 5. H. Feng, R. Bellazzini, *Nature Astronomy* **4**(5), 547 (2020). DOI 10.1038/s41550-020-1103-6. URL <https://www.nature.com/articles/s41550-020-1103-6>. Number: 5 Publisher: Nature Publishing Group
 6. C. Sgró, IXPE Team, *Nuclear Instruments and Methods in Physics Research A* **936**, 212 (2019). DOI 10.1016/j.nima.2018.10.111. URL <http://adsabs.harvard.edu/abs/2019NIMPA.936..212S>
 7. S.N. Zhang, M. Feroci, A. Santangelo, Y.W. Dong, H. Feng, F.J. Lu, K. Nandra, Z.S. Wang, S. Zhang, E. Bozzo, S. Brandt, A.D. Rosa, L.J. Gou, M. Hernanz, M.v.d. Klis, X.D. Li, Y. Liu, P. Orleanski, G. Pareschi, M. Pohl, J. Poutanen, J.L. Qu, S. Schanne, L. Stella, P. Uttley, A. Watts, R.X. Xu, W.F. Yu, J.J.M.i. Zand, S. Zane, L. Alvarez, L. Amati, L. Baldini, C. Bambi, S. Basso, S. Bhattacharyya, R. Bellazzini, T. Belloni, P. Bellutti, S. Bianchi, A. Brez, M. Bursa, V. Burwitz, C. Budtz-Jørgensen, I. Caiazzo, R. Campana, X.L. Cao, P. Casella, C.Y. Chen, L. Chen, T. Chen, Y. Chen, Y.P. Chen, M. Civitani, F.C. Zelati, W. Cui, W.W. Cui, Z.G. Dai, E.D. Monte, D.d. Martino, S.D. Cosimo, S. Diebold, M. Dovciak, I. Donnarumma, V. Doroshenko, P. Esposito, Y. Evangelista, Y. Favre, P. Friedrich, F. Fuschino, J.L. Galvez, Z.L. Gao, M.Y. Ge, O. Gevin, D. Goetz, D.W. Han, J. Heyl, J. Horak, W. Hu, F. Huang, Q.S. Huang, R. Hudec, D. Huppenkothen, G.L. Israel, A. Ingram, V. Karas, D. Karelín, P.A. Jenke, L. Ji, S. Korpela, D. Kunneriath, C. Labanti, G. Li, X. Li, Z.S. Li, E.W. Liang, O. Limousin, L. Lin, Z.X. Ling, H.B. Liu, H.W. Liu, Z. Liu, B. Lu, N. Lund, D. Lai, B. Luo, T. Luo, B. Ma, S. Mahmoodifar, M. Marisaldi, A. Martindale, N. Meidinger, Y.P. Men, M. Michalska, R. Mignani, M. Minuti, S. Motta, F. Muleri, J. Neilsen, M. Orlandini, A.T. Pan, A. Patruno, E. Perinati, A. Picciotto, C. Piemonte, M. Pinchera, A. Rachevski, M. Rapisarda, N. Rea, E.M.R. Rossi, A. Rubini, G. Sala, X.W. Shu, C. Sgro, Z.X. Shen, P. Soffitta, L.M. Song, G. Spandre, G. Stratta, T.E. Strohmayer, L. Sun, J. Svoboda, G. Tagliaferri, C. Tenzer, T. Hong, R. Taverna, G. Torok, R. Turolla, S. Vacchi, J. Wang, D. Walton, K. Wang, J.F. Wang, R.J. Wang, Y.F. Wang, S.S. Weng, J. Wilms, B. Winter, X. Wu, X.F. Wu, S.L. Xiong, Y.P. Xu, Y.Q. Xue, Z. Yan, S. Yang, X. Yang, Y.J. Yang, F. Yuan, W.M. Yuan, Y.F. Yuan, G. Zampa, N. Zampa, A. Zdziarski, C. Zhang, C.L. Zhang, L. Zhang, X. Zhang, Z. Zhang, W.D. Zhang, S.J. Zheng, P. Zhou, X.L. Zhou, in *Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray*, vol. 9905 (International Society for Optics and Photonics, 2017), vol. 9905, p. 99051Q. DOI 10.1117/12.2232034. URL <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/9905/99051Q/eXTP-Enhanced-X-ray-Timing-and-Polarization-mission/10.1117/12.2232034.short>
 8. L. Baldini, M. Barbanera, R. Bellazzini, R. Bonino, F. Borotto, A. Brez, C. Caporale, C. Cardelli, S. Castellano, M. Ceccanti, S. Citraro, N. Di Lalla, L. Latronico, L. Lucchesi, C. Magazzù, G. Magazzù, S. Maldera, A. Manfreda, M. Marengo, A. Marrocchesi,

- P. Mereu, M. Minuti, F. Mosti, H. Nasimi, A. Nuti, C. Oppedisano, L. Orsini, M. Pesce-Rollins, M. Pinchera, A. Profeti, C. Sgrò, G. Spandre, M. Tardiola, D. Zanetti, F. Amici, H. Andersson, P. Attinà, M. Bachetti, W. Baumgartner, D. Brienza, R. Carpentiero, M. Castroonuovo, L. Cavalli, E. Cavazzuti, M. Centrone, E. Costa, E. D'Alba, F. D'Amico, E. Del Monte, S. Di Cosimo, A. Di Marco, G. Di Persio, I. Donnarumma, Y. Evangelista, S. Fabiani, R. Ferrazzoli, T. Kitaguchi, F. La Monaca, C. Lefevre, P. Loffredo, P. Lorenzi, E. Mangraviti, G. Matt, T. Meilahti, A. Morbidini, F. Muleri, T. Nakano, B. Negri, S. Nenonen, S.L. O'Dell, M. Perri, R. Piazzolla, S. Pieraccini, M. Pilia, S. Puccetti, B.D. Ramsey, J. Rankin, A. Ratheesh, A. Rubini, F. Santoli, P. Sarra, E. Scalise, A. Sciortino, P. Soffitta, T. Tamagawa, A.F. Tennant, A. Tobia, A. Trois, K. Uchiyama, M. Vimercati, M.C. Weisskopf, F. Xie, F. Zanetti, Y. Zhou, *Astroparticle Physics* **133**, 102628 (2021). DOI 10.1016/j.astropartphys.2021.102628. URL <https://www.sciencedirect.com/science/article/pii/S0927650521000670>
9. N.O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. Velasco-Hernandez, L. Krpalkova, D. Riordan, J. Walsh, arXiv:1910.13796 [cs] **943** (2020). DOI 10.1007/978-3-030-17795-9. URL <http://arxiv.org/abs/1910.13796>. ArXiv: 1910.13796
 10. M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U.R. Acharya, V. Makarenkov, S. Nahavandi, *Information Fusion* **76**, 243 (2021). DOI 10.1016/j.inffus.2021.05.008. URL <http://arxiv.org/abs/2011.06225>. ArXiv: 2011.06225
 11. T. Kitaguchi, K. Black, T. Enoto, A. Hayato, J.E. Hill, W.B. Iwakiri, P. Kaaret, T. Mizuno, T. Tamagawa, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **942**, 162389 (2019). DOI 10.1016/j.nima.2019.162389. URL <http://arxiv.org/abs/1907.06442>. ArXiv: 1907.06442
 12. A.L. Peirson, R.W. Romani, H.L. Marshall, J.F. Steiner, L. Baldini, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **986**, 164740 (2021). DOI 10.1016/j.nima.2020.164740. URL <http://www.sciencedirect.com/science/article/pii/S0168900220311372>
 13. W.B. Iwakiri, J.K. Black, R. Cole, T. Enoto, A. Hayato, J.E. Hill, K. Jahoda, P. Kaaret, T. Kitaguchi, M. Kubota, H. Marlowe, R. McCurdy, Y. Takeuchi, T. Tamagawa, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **838**, 89 (2016). DOI 10.1016/j.nima.2016.09.024. URL <http://arxiv.org/abs/1610.06677>. ArXiv: 1610.06677
 14. F. Kislat, B. Clark, M. Beilicke, H. Krawczynski, *Astroparticle Physics* **68**, 45 (2015). DOI 10.1016/j.astropartphys.2015.02.007. URL <http://www.sciencedirect.com/science/article/pii/S092765051500033X>
 15. M.C. Weisskopf, R.F. Elsner, S.L. O'Dell, arXiv:1006.3711 [astro-ph] p. 77320E (2010). DOI 10.1117/12.857357. URL <http://arxiv.org/abs/1006.3711>. ArXiv: 1006.3711
 16. A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, in *Proceedings of the 23rd international conference on Machine learning* (Association for Computing Machinery, Pittsburgh, Pennsylvania, USA, 2006), ICML '06, pp. 369–376. DOI 10.1145/1143844.1143891. URL <https://doi.org/10.1145/1143844.1143891>
 17. T. Young, D. Hazarika, S. Poria, E. Cambria, arXiv:1708.02709 [cs] (2018). URL <http://arxiv.org/abs/1708.02709>. ArXiv: 1708.02709
 18. B. Tang, Z. Pan, K. Yin, A. Khateeb, *Frontiers in Genetics* **10** (2019). DOI 10.3389/fgene.2019.00214. URL <https://www.frontiersin.org/articles/10.3389/fgene.2019.00214/full>. Publisher: Frontiers
 19. A. Krizhevsky, I. Sutskever, G.E. Hinton, in *Advances in Neural Information Processing Systems* 25, ed. by F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Curran Associates, Inc., 2012), pp. 1097–1105. URL <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
 20. A. Brill, Q. Feng, T.B. Humensky, B. Kim, D. Nieto, T. Miener, 2019 New York Scientific Data Summit (NYSDS) pp. 1–4 (2019). DOI 10.1109/NYSDS.2019.8909697. URL <http://arxiv.org/abs/2001.03602>. ArXiv: 2001.03602

21. N. Choma, F. Monti, L. Gerhardt, T. Palczewski, Z. Ronaghi, Prabhat, W. Bhimji, M.M. Bronstein, S.R. Klein, J. Bruna, arXiv:1809.06166 [astro-ph, stat] (2018). URL <http://arxiv.org/abs/1809.06166>. ArXiv: 1809.06166
22. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (MIT Press, 2016). URL <http://www.deeplearningbook.org>
23. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning* (Springer, 2009). URL <https://link.springer.com/book/10.1007/978-0-387-84858-7>
24. K. Hornik, M. Stinchcombe, H. White, *Neural Networks* 2(5), 359 (1989). DOI 10.1016/0893-6080(89)90020-8. URL <https://www.sciencedirect.com/science/article/pii/0893608089900208>
25. J. Bridle, in *Advances in Neural Information Processing Systems*, vol. 2, ed. by D. Touretzky (Morgan-Kaufmann, 1990), vol. 2. URL <https://proceedings.neurips.cc/paper/1989/file/0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf>
26. T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, in *Advances in Neural Information Processing Systems*, vol. 33, ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Curran Associates, Inc., 2020), vol. 33, pp. 1877–1901. URL <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf>
27. Y. LeCun, Y. Bengio, in *The handbook of brain theory and neural networks* (MIT Press, Cambridge, MA, USA, 1998), pp. 255–258
28. K. He, X. Zhang, S. Ren, J. Sun, arXiv:1512.03385 [cs] (2015). URL <http://arxiv.org/abs/1512.03385>. ArXiv: 1512.03385
29. Y. Zhang, Q. Yang, arXiv:1707.08114 [cs] (2021). URL <http://arxiv.org/abs/1707.08114>. ArXiv: 1707.08114
30. D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, D. Mané, arXiv:1606.06565 [cs] (2016). URL <http://arxiv.org/abs/1606.06565>. ArXiv: 1606.06565
31. A. Kendall, Y. Gal, in *Advances in Neural Information Processing Systems 30*, ed. by I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Curran Associates, Inc., 2017), pp. 5574–5584. URL <http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf>
32. S. Choi, K. Lee, S. Lim, S. Oh, arXiv:1709.02249 [cs] (2017). URL <http://arxiv.org/abs/1709.02249>. ArXiv: 1709.02249
33. K. Jakkala, arXiv:2106.12135 [cs, stat] (2021). URL <http://arxiv.org/abs/2106.12135>. ArXiv: 2106.12135
34. B. Lakshminarayanan, A. Pritzel, C. Blundell, in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc., Long Beach, California, USA, 2017), NIPS'17, pp. 6405–6416
35. F.K. Gustafsson, M. Danelljan, T.B. Schon, (Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020), pp. 318–319. URL <https://arxiv.org/abs/1906.01620>
36. J. Antorán, J.U. Allingham, J.M. Hernández-Lobato, arXiv:2006.08437 [cs, stat] (2020). URL <http://arxiv.org/abs/2006.08437>. ArXiv: 2006.08437
37. S. Fort, H. Hu, B. Lakshminarayanan, arXiv:1912.02757 [cs, stat] (2019). URL <http://arxiv.org/abs/1912.02757>. ArXiv: 1912.02757
38. Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J.V. Dillon, B. Lakshminarayanan, J. Snoek, arXiv:1906.02530 [cs, stat] (2019). URL <http://arxiv.org/abs/1906.02530>. ArXiv: 1906.02530
39. A. Kendall, Y. Gal, R. Cipolla, arXiv:1705.07115 [cs] (2018). URL <http://arxiv.org/abs/1705.07115>. ArXiv: 1705.07115
40. A.L. Peirson, R.W. Romani, arXiv:2107.08289 [astro-ph] (2021). URL <http://arxiv.org/abs/2107.08289>. ArXiv: 2107.08289

41. S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytrcek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell'Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J.J. Gómez Cadenas, I. González, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F.W. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampén, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M.G. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J.P. Wellisch, T. Wenaus, D.C. Williams, D. Wright, T. Yamada, H. Yoshida, D. Zschesche, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **506**(3), 250 (2003). DOI 10.1016/S0168-9002(03)01368-8. URL <http://www.sciencedirect.com/science/article/pii/S0168900203013688>
42. C. Steppa, T.L. Holch, SoftwareX **9**, 193 (2019). DOI 10.1016/j.softx.2019.02.010. URL <http://arxiv.org/abs/1903.01814>. ArXiv: 1903.01814
43. R. Doon, T. Kumar Rawat, S. Gautam, in *2018 IEEE Punecon* (2018), pp. 1–5. DOI 10.1109/PUNECON.2018.8745428
44. G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, arXiv:1608.06993 [cs] (2018). URL <http://arxiv.org/abs/1608.06993>. ArXiv: 1608.06993
45. Attention Is All You Need (2017). DOI 10.48550/arXiv.1706.03762. URL <http://arxiv.org/abs/1706.03762>. Number: arXiv:1706.03762 arXiv:1706.03762 [cs]
46. I. Sutskever, J. Martens, G. Dahl, G. Hinton, in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28 (JMLR.org, Atlanta, GA, USA, 2013)*, ICML'13, pp. III–1139–III–1147
47. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, arXiv:2010.11929 [cs] (2021). URL <http://arxiv.org/abs/2010.11929>. ArXiv: 2010.11929