Deep Learning-Based Super-Resolution and De-Noising for XMM-Newton Images

Sam F. Sweere,[1,2] ⋆ Ivan Valtchanov,[3] Maggie Lieu,[4] Antonia Vojtekova,[2] Eva Verdugo,[2] Maria Santos-Lleo,[2] Florian Pacaud,[5] Alexia Briassouli[1] and Daniel Cámpora Pérez[1]

[1] *Faculty of Science and Engineering, Maastricht University, Maastricht, Netherlands*
[2] *European Space Agency, ESAC, Camino Bajo del Castillo, 28692, Villanueva de la Cañada, Madrid, Spain*
[3] *Telespazio UK for European Space Agency, ESAC, Camino Bajo del Castillo, 28692, Villanueva de la Cañada, Madrid, Spain*
[4] *School of Physics & Astronomy, University of Nottingham, University Park, Nottingham, NG7 2RD, UK*
[5] *University of Bonn, Argelander Institut für Astronomie (AIFA), Auf dem Huegel 71, D-53121, Bonn, Germany*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

The field of artificial intelligence based image enhancement has been rapidly evolving over the last few years and is able to produce impressive results on non-astronomical images. In this work we present the first application of Machine Learning based super-resolution (SR) and de-noising (DN) to enhance X-ray images from the European Space Agency's *XMM-Newton* telescope. Using *XMM-Newton* images in band [0.5,2] keV from the European Photon Imaging Camera pn detector (EPIC-pn), we develop *XMM-SuperRes* and *XMM-DeNoise* — deep learning-based models that can generate enhanced SR and DN images from real observations. The models are trained on realistic *XMM-Newton* simulations such that *XMM-SuperRes* will output images with two times smaller point-spread function and with improved noise characteristics. The *XMM-DeNoise* model is trained to produce images with 2.5× the input exposure time from 20 to 50 ks. When tested on real images, DN improves the image quality by 8.2%, as quantified by the global peak-signal-to-noise ratio. These enhanced images allow identification of features that are otherwise hard or impossible to perceive in the original or in filtered/smoothed images with traditional methods. We demonstrate the feasibility of using our deep learning models to enhance *XMM-Newton* X-ray images to increase their scientific value in a way that could benefit the legacy of the *XMM-Newton* archive.

Key words: techniques: image processing – techniques: high angular resolution – X-rays: general

1 INTRODUCTION

Over the last two decades, the European Space Agency *XMM-Newton* (Jansen et al. 2001) X-ray space observatory has been continuously advancing our understanding of the cosmos through detailed observations of black holes, the formation of galaxies and many other phenomena in our X-ray sky (Santos-Lleo et al. 2009; Wilkins et al. 2021). The 3 X-ray telescopes on-board are equipped with a set of imaging CCD detectors: European Photon Imaging Cameras (EPIC), with two MOS-CCD arrays Turner et al. (2001) and one pn-CCD Strüder et al. (2001). EPIC-pn has an effective area on average ∼2-3 times that of MOS, depending on the energy band. Concerning the characteristics of the cameras for imaging, they have comparable point-spread function (PSF) with Full Width at Half Maximum (FWHM) of ∼5-6″ on axis (and half-energy width HEW of ∼14-15″), comparable field of view of ∼14-15′ radius, and MOS detectors have pixel physical size of 1.1″, compared to 4.1″ for pn (see e.g. the *XMM-Newton* User Handbook). The NASA's *Chandra* X-ray telescope (Weisskopf et al. 2000) has a spatial resolution far superior to *XMM-Newton* , with PSF HEW of the ACIS detector of ∼ 0.5″, limited by the physical pixel size, with a drawback of having much smaller effective area. It is desirable to have both good sensitivity and spatial resolutions: longer exposures allow collecting more photons and hence pick up fainter sources. However, the noise will also increase, thus there is a need for better sensitivity in order to detect extended emission.

Most observations need to achieve a certain signal-to-noise (SNR) for the targets under study, in order to be able to draw scientific conclusions. This threshold dictates the exposure time that the observers request to the time allocation committees, and for observatories with high over-subscription rate asking too much time has its drawbacks, i.e. less chances for approval. Therefore, methods to enhance the SNR for a given observing time can be used to increase the science quality of the data. Image enhancement through noise level reduction is a popular way to improve the SNR (e.g. Vojtekova et al. 2020).

In X-ray observations, photon counts are subject to a Poissonian noise that is dependent on the count rate itself. Therefore, the SNR is smaller in low count rate areas, limiting the detection of faint sources. Binning of X-ray photons is one way to increase the total SNR, albeit at the cost of reducing spatial resolution. Sanders & Fabian (2001) use an adaptive binning method on Chandra observations of the Perseus cluster to reveal structure in the central region. Bourdin et al. (2001) introduced a multi-scale wavelet transform approach to denoise MOS1 and MOS2 images and were able to successfully recover

⋆ E-mail: samsweere@gmail.com

© 2022 The Authors

the total flux and signal shape of toy-model sources, demonstrating that de-noising methods like these can be used to provide more accurate brightness mapping.

In addition to the noise, the PSF can lead to blending and sources confusion, when sources closer than a certain fraction of the PSF's FWHM can no longer be separated. Resolving and deblending such sources can be achieved with super-resolution.

Super-resolution (SR) describes a class of methods that can upscale video or images from lower resolutions to higher ones. Such methods have been successfully demonstrated on astronomical imaging, e.g. (Starck et al. 2002; Puschmann & Kneer 2005; Li et al. 2018). Many methods for SR exist (see e.g. Zhou et al. 2012; Siu & Hung 2012). Alternatively, edge-detection methods (see e.g. Sanders et al. 2016) are being used for enhancing features and identifications of structures, although in general they do not increase the spatial resolution.

Traditionally, interpolation methods such as bilinear and nearest neighbour interpolation are used for upscaling. However, these methods often introduce side effects such as noise amplification and blurring. Furthermore, super-resolution on X-ray images imposes additional challenges since X-ray images are typically sparse, and the data are poisson distributed. Nevertheless, Feng et al. (2003) demonstrate using a direct demodulation (DD) method, the spatial resolution of *XMM-Newton* EPIC images can be improved by a factor of 5 whilst adhering to the requirements for spectral studies.

Super-resolution and de-noising is fundamentally an ill-posed problem since given a noisy/low-resolution input image, there are an infinite number of possible enhanced (high resolution) images that it could correspond to. The noisy, low resolution, input image inherently does not contain all the information of an enhanced image. However, in recent years significant progress has been made in the field of de-noising and super-resolution using machine learning methods (Yang et al. 2019; Zhang et al. 2019; Wang et al. 2021, 2018; Chen et al. 2018; Lugmayr et al. 2020; Dong et al. 2014; Jain & Seung 2008). In these learning-based approaches, a network is trained with data to learn the mapping between an image and an enhanced image, where the enhanced image in our case is a higher SNR image and/or a higher resolution image.

These models primarily make use of Fully Convolutional Networks (FCN), trained using a relevant quantitative metric used as a loss function. Similar to traditional convolutional neural networks (CNNs, LeCun et al. 1989, 1998), FCNs comprise of convolutional, pooling and layers, however they do not have dense layers (see e.g. Su et al. 2020, for an in depth introduction to these components) and their output size are typically the same or larger than the input. For this reason, FCNs are often used for computer vision tasks such as semantic segmentation, de-noising and super-resolution (Jain & Seung 2008; Dong et al. 2014; Chen et al. 2018). Images generated this way tend to lack clarity as they often minimise a *simple* loss function such as the mean absolute error (L1), that favors predicting the average over all plausible enhanced images. This leads to fewer finer details in the generated images. To address this, more recent approaches make use of more complex loss functions. The perceptual loss function (Johnson et al. 2016; Zhang et al. 2019) incorporates style transfer through pre-training on a target dataset with a particular style or content. Generative Adversarial Networks (GANs) use two competing models - a generator to produce enhanced images from a given input image and a discriminator to differentiate between the real and generated images. Such networks make use of an adversarial loss (Wang et al. 2018) to obtain photorealistic images. In astronomy, these methods have been used to improve observations. Schawinski et al. (2017) showed that using a GAN, they were able to recover features from artificially degraded optical observations. Vojtekova et al. (2020) used a FCN and perceptual loss to de-noise Hubble Space Telescope images, improving the signal-to-noise ratio by a factor of 1.3-1.5, and Lauritsen et al. (2021) use an auto-encoder to obtain super-resolution of *Herschel* observations in the sub-millimetre wavelength range.

This paper aims to apply these ideas and develop deep learning-based methods for super-resolution and de-noising of images from *XMM-Newton* to increase their scientific value. The *XMM-Newton* Science Archive contains observations spanning over 20 years and therefore there is ample data to satiate the training of a machine learning model. Improving the quality of this existing data is of great interest to the astronomical community and the lasting legacy of *XMM-Newton*. In Section 2 we introduce the real and simulated data that are used to train and validate our method, we describe the different components in the simulated data sets and the pre-processing techniques. In Section 3 we define the models and the model architecture and we detail the loss functions and the evaluation metrics. The optimisation process of the models is presented in Section 4 and the results on simulated and real observations are given in Section 5. We discuss the results and put forward some caveats and limitations in Section 6 and we summarise with our conclusions in Section 7. More technical details on particular aspects of the work are separated in the Appendices.

2 DATA

To train and validate our models we created a dataset consisting of real *XMM-Newton* observations (section 2.1) and a separate dataset of simulated *XMM-Newton* observations (section 2.2).

2.1 Real XMM-Newton Dataset

XMM-Newton observations are in the form of *eventlists* that record the time photons of a certain energy hit a specific CCD pixel. We need to transform these event lists into images, so in order to limit the scope of this research we focus on the EPIC-pn detector in the (extended) full-frame mode and build images using events with energy in band [0.5,2.0] keV.

We use the entire *XMM-Newton* Science Archive (XSA), filtering out observations with less than 20 ks exposure times, bad time intervals and events. We split the event lists in 10ks intervals for each observation, i.e. for a 40 ks observation, we generate 4x10 ks images, 2x20 ks images, 1x30 ks and 1x40 ks image. The images with multiple exposure times enables us to train super-resolution and de-noising models using the same exposure-time for different observations. It also enables us to train a de-noising model with pairs of low and high exposure images. The exact implementation details of generating the real *XMM-Newton* dataset are described in Appendix A.

The final dataset contains 5554 unique EPIC pn exposures giving rise to the same number of full exposure images and almost 24000 sub-images after splitting the eventfiles into sub-images with multiples of 10ks exposure times. We only use 20ks, 50ks and 100ks sub-images in our final datasets (Table 1).

2.2 Simulated XMM-Newton Dataset

The real *XMM-Newton* dataset cannot be used to train a super-resolution model. To train a super-resolution model we need a data set consisting of low resolution input images and their high resolution counterparts as our targets.

Table 1. The distribution of all the sub-images extracted from the total 5554 EPIC pn exposures as a function of their exposure time and their corresponding train, validate and test splits.

Exposure time (ks)	Number of images	Train	Validate	Test
20	5022	3489	775	758
50	834	583	123	128
100	109	81	9	19

The creation of such a dataset, consisting of high and low resolution pairs, is often achieved digitally, through down-sampling high resolution images (e.g. Wang et al. 2018; Lugmayr et al. 2020; Ledig et al. 2017; Chen et al. 2022) or optically, through aligning images taken with different optical zoom scales (e.g. Ledig et al. 2017; Chen et al. 2022; Zhang et al. 2019). In this study we want to achieve higher than *XMM-Newton* resolution images both spatially (smaller pixel size) and with smaller PSF size. Thus, the down-sampling approach is not an option as it would only decrease the pixel size and not the PSF.

Zooming is also not an option, since the *XMM-Newton* telescope (a glancing reflector) cannot zoom. It would be possible to combine low resolution *XMM-Newton* images with higher spatial resolution images taken with another X-ray telescope such as *Chandra*, however, we would be limited by the number of fields that have been observed by both *XMM-Newton* and *Chandra* and therefore there would not be enough data to train the model. Additionally, we note that most X-ray sources are variable, hence ideally we would need simultaneous observations, making the available data even more limited, and the different telescopes have different properties that need to be taken into consideration.

An appropriate training dataset can be achieved through the use of simulations, where we can artificially increase the resolution (both the angular resolution and the sensor resolution) whilst maintaining the required observational properties of real *XMM-Newton* images.

2.2.1 Simulating XMM-Newton Images

For our simulations we use the SIXTE X-Ray simulation software package (Dauser et al. 2019). This is a X-ray simulation software package provided by ECAP/Remeis observatory[1]. We create custom configuration files to resemble the *XMM-Newton* EPIC-pn detector and individual events in the [0.5,2] keV energy band. This configuration provides realistic images with all important instrumental properties: vignetting, position-dependent PSF, background noise. We use a recent calibration file to replicate the vignetting properties of the *XMM-Newton*[2]. For a detailed description of the simulation configuration see appendix B.

We create two sets of simulated images: one with the actual *XMM-Newton* PSF and another one with a rescaled PSF with twice the resolution (i.e. FWHM and HEW are two times smaller). As this study is a proof of concept we chose to be somewhat conservative and only increase the resolution twice. We keep the pixel size of the images in the second set at half the size of the original (i.e. keeping the PSF sampling the same), hence they contain 4 times more pixels for the same field-of-view.

[1] https://github.com/thdauser/sixte
[2] https://xmmweb.esac.esa.int/docs/documents/CAL-SRN-0321-1-2.pdf

We focus on simulated observations of complex fields, like observations of galaxies or clusters of galaxies which provide images with three main components: the extended emission from the galaxy or the cluster, "contaminant" point sources which are mainly Active Galactic Nuclei (AGN) and the background Figure 1). We will simulate these three components separately and describe them in the next sub-sections.

2.2.2 Extended Source Component

To simulate extended sources, we use the IllustrisTNG[3] suite of a large scale, cosmological magnetohydrodynamical simulations of galaxy formation. The three simulations we are using are IllustrisTNG 50-1 (Nelson et al. 2019; Pillepich et al. 2019), IllustrisTNG 100-1 and IllustrisTNG 300-1 (Springel et al. 2018; Marinacci et al. 2018; Nelson et al. 2018; Naiman et al. 2018; Pillepich et al. 2018) at a redshift of 0.01. These are simulated at different scales (cubic volumes of ~ 50, 100, and 300 Mpc side lengths) and mass resolutions that enable the study of different types of sources - supernova remnants at the smaller scale and galaxy clusters at the larger scale. The simulations include full baryonic physics. In each simulation, we select the top 400 subhalos based on the M_{gas}. We then project the subhalo from the x,y, and z axes on two different scales. We project at two different scales for a close-up of the source and a projection that is four times further away (tng50-1: 100 kpc and 400 kpc, tng100-1: 400 kpc and 1.6 Mpc, tng300-1: 1 Mpc and 4 Mpc) in order to capture different spatial information from the same source. From these projections, we calculate the X-ray photon intensity in the [0.5,2] keV energy range at redshift 0.01. X-ray dim subhalos are manually removed from our dataset. This results in:

- TNG50-1: 1632 images
- TNG100-1: 2165 images
- TNG300-1: 2374 images

To convert the intensity maps from the TNG simulations to photon flux maps in $[0.5, 2.0]$ keV, we need to assume a spectral model. Galaxy clusters exhibit thermal spectrum, however, the subhalos from different TNG scales are not necessarily galaxy clusters. Therefore, for simplicity, we decided to use an absorbed power law. We use `XSPEC` (Arnaud 1996) to do this, assuming $N_H = 0.04 \times 10^{22}$ cm^{-2}, and photon index $\Gamma = 2$. The flux of the extended sources are further modified to reflect real extended sources that *XMM-Newton* observed: we set the central part of the source (a box at 5% of the image width/height at the center of the image) to have a flux value randomly sampled uniformly to be between 5 and 50 times the standard deviation of background noise (σ_B) at the boresight.

Additionally, to increase our training sample size, we artificially augment the data by apply a random zoom of scale $[1, 2]$, and a random x, y perturbation offset using a standard distribution with a standard deviation of 5% of the image height/width.

After the various augmentations are applied we are left with a total of 30855 augmented inputs that are used as the extended source component of the simulated *XMM-Newton* observations.

2.2.3 Point-Source Component

The point source component of the observations are mostly AGNs. Based on measurements of AGN number counts $N(< S)$ as function of flux S (Gilli et al. 2007), we compute the expected number of

[3] https://www.tng-project.org/

AGNs in each observation corrected for the *XMM-Newton* FOV [4]. Gilli et al. (2007) do not publish their uncertainties, therefore we include an additional Poisson uncertainty $\pm\sqrt{N_{\rm AGN}}$.

We simulate AGN absorption at different Galactic latitudes by shifting the $\log(N)/\log(S)$ distribution down by a certain factor, i.e. dividing the flux S. We set the absorption factor to 100 at the Galactic plane ($b = 0\,\deg$) and to 1 at the Galactic poles $|b| = 90\,\deg$. We draw random absorption factors from log distribution from 1 to 100, such that we favour more extragalactic fields. For example, a field with absorption factor of 20 will have much less AGNs than a field with no absorption (absorption factor of 1).

2.2.4 Background Component

The background of the EPIC data has different components. The first component is due to the astrophysical sky background, from thermal low energy emission, unresolved cosmological sources and solar wind charge exchange. There is also particle induced background and finally electronic noise. The sky background is position dependent, while the quiescent particle background (QPB) is time-variable and correlates with the solar cycle. In general, in the $[0.5, 2]$ keV energy band, the sky background level is ~ 2 times higher than the QPB.

We decided not to simulate the individual background components but use the available Blank Sky event files[5] `pn_t_ff_g` (Carter & Read 2007) which contain all components as they are produced using real *XMM-Newton* observations. The background component is simulated using the spectrum extracted from these Blank Sky event files.

2.2.5 Co-Added Images

Simulating the extended, point source and background components separately enables us to create many combinations of images. For example, if we look at uniquely simulated sources at 100ks, we have 30855 simulations of extended sources, 25000 simulations of point sources and 25000 simulations of background noise. This amounts to $30855 \times 25000 \times 25000 \approx 2 \times 10^{13}$ possible unique simulated pn images. We generate noisy images to use as inputs to both of our networks, paired with noiseless counterpart images as our DN network targets, and high-resolution, noiseless counterpart images as our SR network targets. Although many of these will be visually similar, the different combinations help reduce overfitting of the model. This compensates for the lack of traditional image augmentations such as spatial and colour transforms that would change the properties of the observation.

2.3 Data Pre-Processing

To help accelerate the optimization of the model and more efficient convergence, the data need to be pre-processed.

We transform the data from counts to counts/s by dividing the image by the exposure time. This enables us to use training data with different exposure times whilst maintaining the input pixel intensity distribution.

Bright sources can have large pixel values that are orders of magnitudes higher than other pixels in a particular observation. This big difference can make training a deep learning model very unstable and therefore we clip pixel values to 200 times the mean background rate $\mu_B = 1.1168 \times 10^{-5}$ counts/s for the denoising data set and $50\mu_B$ for the (2x) super resolution dataset[6]. This can lead to the loss of detail in bright regions, however, the majority of the extended features have X-ray counts below 200 times the mean background.

The image is then normalized to $[0, 1]$. Even on normalised images, fainter features would not be visible to the human eye when visualizing them without a suitable data scaling (or stretch). Many interesting structures have pixel counts a few times above the background noise (σ_b), while bright parts of the image, such as centres of point-like sources, can have pixel counts in the hundreds of σ_b.

The pixel intensity distribution can also affect the training of the model. For example, an L1 loss, would put more weight on features with higher pixel values and bias the results. Our main focus is to enhance the visual clarity of faint details, and for this reason we explore several different data scaling functions. We compare linear, square root (sqrt), logarithmic (log) and hyperbolic arcsine (asinh) stretch functions. Each of these highlights different levels of the normalised pixel values, as shown in Figure 2, with asinh lying between the sqrt and log stretch.

Table 2. The train, validation and test splits of the simulated sub-components.

Component	Train	Validation	Test
Extended Sources	24678	3090	3087
AGNs	20000	2500	2500
Background	20000	2500	2500

2.4 Train, Validation and Test split

The final images are split into train, validation and test sets where only the training dataset is used to update the weights of the network, the validation data is used to monitor the performance of the network and the test data is reserved for final evaluation of network and is not seen by the network during training.

For the simulated dataset, the splits are made in a way that all the spatial augmentations done during the simulations are always in the same set. Note that a specific source can appear multiple times across the sets but with different projections and distances. The choice to not split based on the sub-halos themselves was made because rare source structures could then be over-represented in one of the splits. Since different projections and distances of the same source look very different, this should not be an issue.

For each component (extended sources, point sources, background) of the simulated dataset, we split the distribution to have 80%, 10% and 10%, train, validation and test subsets respectively. The number of sub-components in each subset is shown in Table 2.

Since the real dataset has a smaller number of images we need a larger percentage of the images to validate and test the results in comparison to the simulated dataset. The train, val, test split distribution chosen for real *XMM-Newton* images is thus 70%, 15% and 15% respectively.

[4] The expected number of AGNs was determined using the following tool: `http://www.bo.astro.it/~gilli/counts.html`

[5] `https://xmm-tools.cosmos.esa.int/external/xmm_calibration/background/bs_repository/blanksky_all.html`

[6] The clipping value for 2x is four times smaller because the pixel density is four times larger, meaning that the pixel counts on one pixel on the 1x resolution scale will be distributed over four pixels in the 4x resolution scale.

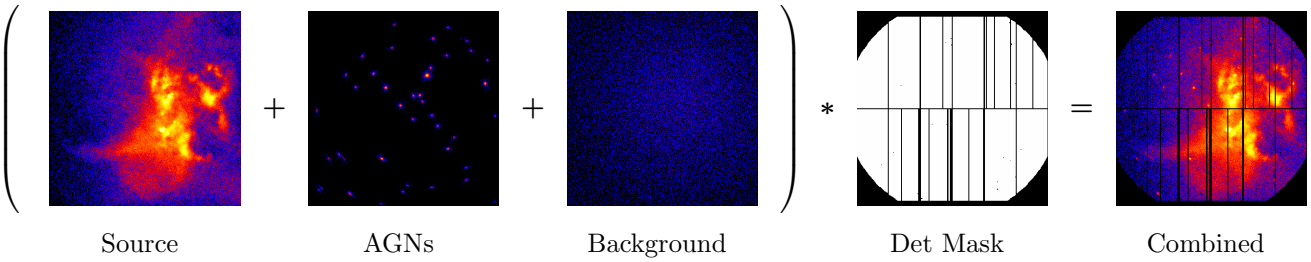

Figure 1. Different components that make up a simulated *XMM-Newton* observation. The simulated extended source, AGNs and background are added together and then multiplied by the detector mask to create a simulated *XMM-Newton* observation. The images are logarithmic scaled for visualisation.

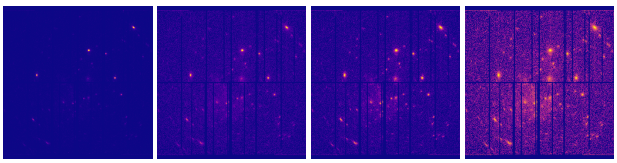

Figure 2. Examples of the different scaling applied to galaxy M101 (obs id: 0824450501). From *left* to *right*: linear, sqrt, asinh and log.

3 METHOD

3.1 De-Noising and Super-Resolution Model

For the de-noising model, the input image is at the default *XMM-Newton* resolution with 20ks exposure time and the target image is similarly at the default *XMM-Newton* resolution but with 50ks exposure time. We choose this combination of exposure times to replicate more realistic observations and to ensure our results are trustworthy. Having an exposure time of for example 100ks would force the model to make more uncertain predictions based on the input image. At 50ks exposure we also have more real world data to train and validate on.

The input to the SR model is the simulated *XMM-Newton* image at the normal resolution with an exposure time of 20ks and background noise. For the target image we use the simulation image with 2x resolution, an exposure time of 100ks and without background noise. Omitting the background noise from the label image allows the model to concentrate on the source.

Originally, for our super-resolution problem, we took an approach based on a GAN architecture (Goodfellow et al. 2014). GANs use a generator network to generate realistic images and a discriminator network to ensure that the generated images are visually indistinguishable from the high-resolution target images. We initially chose the ESR-GAN model (Wang et al. 2018) for its proven success as a super-resolution model. It consists of a stacked Residual-in-Residual Dense Block (RRDB, see subsection 3.2) generator and a deep CNN discriminator. However, like all GAN based algorithms, ESR-GAN suffers from hallucinations of non-existent features in the model output. These hallucinations are caused by the discriminatory network that forces the generator output to be visually similar to the images in the training dataset. However, when for example, a part of the input image does not have sufficient information to generate a high resolution counterpart the model starts to hallucinate detailed features in order to generate a visually similar output image. This can have catastrophic consequences in astronomy.

For more robust reconstructions, and at the expense of generating images that are less visually similar to the target, we choose to omit the adversarial component. Our main model is therefore based on only the RRDB generator used in ESR-GAN. As we only use a generator model, our architecture is no longer a GAN. This architecture generates more reliable outputs, however is only able to generate sharp reconstructions in areas where the model learned to generate the features with high confidence, such as high signal-to-noise point sources. Note that the model does not output the confidence level of the generated features. Aspects of low confidence such as the background noise will result in blurry reconstructions and the output image is unlikely to visually resemble the target. The areas of low and high confidence are different for each image since they are dependent on the features present in the input image.

3.2 Model Architectures

The main feature of our architectures is the use of the RRDB block (Figure 3). This block is inspired by the DenseNet architecture (Iandola et al. 2014) and connects all layers within the residual block with each other. The RRDB block consists out of three Dense Blocks, within which contain 4 consecutive convolution layers each followed by Leaky ReLU activations and an additional convolutional layer. The concatenated output of every previous layer is fed into the next convolution layer. Thus the number of input channels in every consecutive convolution layer increases linearly:

$$N_{i+1} = N_i + N_0 \tag{1}$$

Where N_{i+1} is the number of input channels in the next layer, N_i is the number input channels in the current layer and N_0 is the number of input channels in the first layer.

For our task of super-resolution, we base our architecture on the original ESR-GAN generator (Figure 4), however we replace the nearest-neighbour interpolation upsampling layer with pixel shuffle upsampling Shi et al. (2016). Pixel shuffle has more connections and does not interpolate the upsampled image. This should improve quantitative details on smaller scales.

For the denoising model, we use the same architecture however we remove the upsampling layer and the last convolutional layer (Figure 5). Additionally more skip connections are introduced as inspired by Zhang et al. (2020a). This helps to learn smaller features in the image and improves training speed since the model does not have to process all the small features. Instead, it only learns the features to suppress to create the de-noised output image.

3.2.1 *Weight Initialisation*

Since our output images have values between 0.0 and 1.0, we need to ensure that the first pass through the model results in values in this

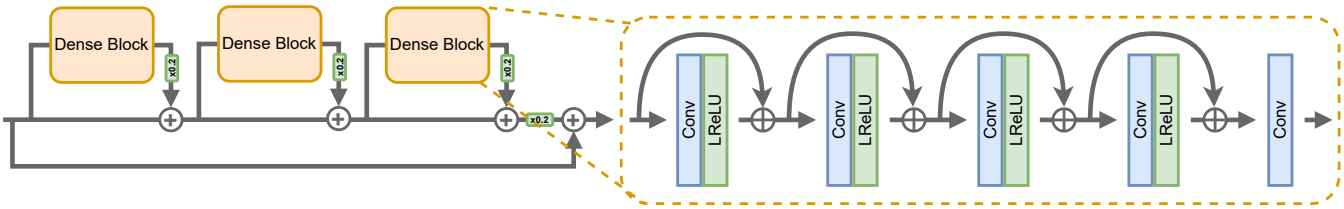

Figure 3. RRDB Basic Block, with ⊕ resembling concatenation. Adapted from (Wang et al. 2018).

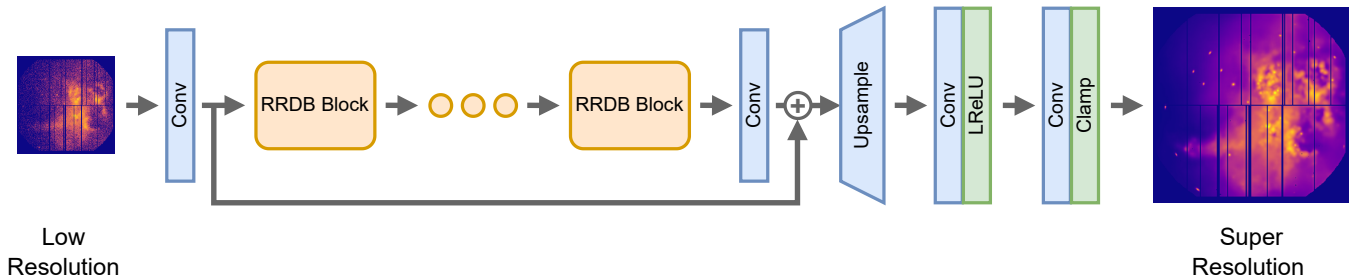

Figure 4. RRDB Super-Resolution Model Architecture. The network takes a low resolution image and undergoes a convolutional layer followed by a series of RRDB blocks, another convolutional layer with skip connections, an upsampling layer, and finally 2 more convolutional layers to return a higher resolution mapping.

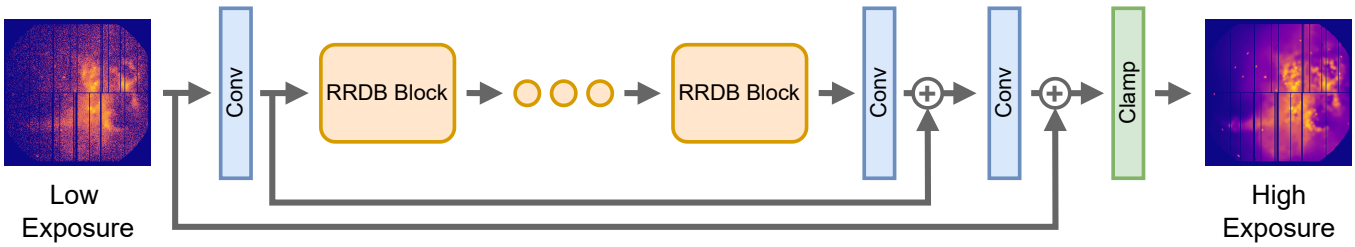

Figure 5. RRDB de-noise model architecture. The network takes a noisy low exposure image and undergoes a convolutional layer followed by a series of RRDB blocks, another 2 convolutional layer, with skip connections to output a higher SNR mapping.

interval. If this does not happen, when for example the values are all negative, everything will be clipped to 0. This will result in no usable gradients for back-propagation, i.e. the model will not train. Therefore we skew the initial weights in the last convolution layer to be slightly more positive.

In general, the weights are initialized using a random normal distribution where the standard deviation is based on the size of the convolution layer:

$$std = \frac{1}{\sqrt{\text{layer size}}} \tag{2}$$

In order to prevent the initial forward passes from being outside the image range, we initialize the weights in the last convolution layer to be uniformly distributed from $[-std, std + 0.01 \times std]$, this ensures the weights are slightly more positive. An alternative solution would be to explore different final activation layers but this is beyond the scope of this work.

3.3 Loss Functions

The loss function determines how good a prediction of the model is with respect to the reference image. In preliminary testing, we observe a substantial difference in the visual appearance of the generated images and the target images. However, we require a more quantitative measurement of the reconstruction than a simple visual comparison. Different loss functions optimise the model for different attributes of the output. We consider L1, Poisson, Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM, Zhou Wang et al. 2004), and Multi-scale Structural Similarity Index (MS-SSIM, Wang et al. 2003) loss functions.

The L1 loss minimizes the mean absolute difference between pixel values of the generated and target images. This is the simplest loss function. We do not include the mean square error loss (L2) as it is sensitive to outliers and extreme values which can lead to bad performance on e.g. observations of AGNs. We include the Poisson loss, which measures the likelihood of the generated pixel values assuming that the target comes from a Poisson distribution conditioned on the input. It's relevant here because our data is count data and follows a Poisson distribution. The PSNR is a measure of the ratio between the maximum signal and the distorting noise. It is one of the basic metrics in denoising models. A higher PSNR value equates to better denoising. Lastly SSIM and MS-SSIM are perceptual metrics that incorporate the idea that spatially close pixels have strong inter-dependencies. These losses therefore measure the similarity of structure in images on a single scale and a combination of different scales respectively. The SSIM and MS-SSIM have parameters that needed to be fine-tuned for our problem. For SSIM we empirically

found the following parameters to work well: *window size* = 13, $\sigma = 2.5$, $K_1 = 0.01$ and $K_2 = 0.05$. For MS-SSIM we used the same parameters as for SSIM with the weight for each scale being $[0.0448, 0.2856, 0.3001, 0.2363, 0.1333]$.

To meaningfully combine loss functions, we need to normalize them since the values of different loss functions can differ by orders of magnitude. We normalize the loss based on trial runs of the model trained with Poisson loss with the various data scaling functions and an untrained model. A model trained with a loss function different from the Poisson loss will generate different output images. These will have different loss values. However, the difference between a trained and untrained model with any of our loss functions will be huge. Therefore, the final loss metrics should be approximately on the same scale.

We aim to have the normalized loss value of the untrained model at 1 and the trained model at 0. We calculated the normalization with the following formula:

$$L_{normalized} = \alpha \, L_{unnormalized} + \beta \tag{3}$$

With:

$$\alpha = \frac{y_2 - y_1}{x_2 - x_1} \tag{4}$$

$$\beta = y_1 - a \, x_1 \tag{5}$$

Where y_1 is the target loss value for the untrained model (in our case $y_1 = 1$), y_2 is the target loss value for the trained model (in our case $y_2 = 0$), x_1 is the measured loss of the untrained model and x_2 is the measured loss of the trained model. We can now combine different loss functions by adding the normalized loss functions together since the loss functions are now on the same scale.

3.4 Evaluation Metrics

To evaluate our models, in addition to the loss metrics discussed in subsection 3.3, we make use of the Feature Similarity Index (FSIM, Zhang et al. 2011) and the Haar wavelet-based Perceptual Similarity Index (HaarPSI, Reisenhofer et al. 2018) metrics.

In a similar fashion to SSIM and MS-SSIM, FSIM is a metric that aims to mimic human vision. The human visual system perceives images through salient low-level features, and FSIM uses 2 kinds of these features to determine image quality - the phase congruency (PC) and the gradient magnitude (GM). Rather than the areas with sharp changes in contrast, PC highlights features as areas where the order in the phase component of the Fourier transform is high. Thus PC is an illumination and contrast invariant measure of feature significance. However since contrast is also an important aspect of human vision, FSIM also incorporates gradient magnitude to encode contrast information.

HaarPSI uses coefficients obtained from a discrete wavelet transform to construct local similarity maps between two images. The Haar wavelet is used, being the simplest and most efficient to compute. Next an non-linearity is applied in the form of a logistic function to highlight the relative importance of those areas.

4 MODEL OPTIMISATION

Hyper-parameters influence the training of a model and its performance. To tune for the optimal configuration we perform a parameters search where we trained many models with different hyper-parameters to gain insight into the influence of each hyper-parameter on the model performance. There are two categories of hyper-parameters: the model hyper-parameters (subsection 4.1) and the data hyper-parameters (subsection 4.2). The model hyper-parameters are tuned first and fixed before the data hyper-parameters are tuned.

Table 3. Final model hyper-parameters.

Hyper-parameter	Value
RRDB convolutional filters	32
RRDB blocks	4
Batch size	1
Learning rate	0.0001
Data scaling	square root
Loss function	PSNR and MS_SSIM

We use a grid-search approach to hyper parameter tuning, which can be computationally expensive and therefore we only train on a 25% subset of the simulated dataset where the inputs are further cropped to 128x128 pixels around the boresight. We train the models for 50 epochs on this reduced dataset. Although this is slightly different from the final model training, we argue that it gives enough insight into the model performance to make informed choices on the hyper-parameters used in the final model.

4.1 Model Hyper-Parameter Tuning

The model hyper-parameter-search aims to optimise parameters based on the model's learning ability. For this sweep, we use a Poisson loss with square root data-scaling since this resulted in desirable results in initial testing. We train models with a range of combination of hyper-parameters (180 models) and monitor the loss of the validation data. For exact details see Appendix C. The final model hyper-parameters are shown in Table 3.

4.2 Data Hyper-Parameter Tuning

Having determined the model hyper-parameters we tune the hyper-parameters that influence the visual properties of the generated images: the loss function and data scaling. Here, we fix the model hyper-parameters to the optimal values determined in subsection 4.1 however the batch size used is set to 4 to increase the training speed. We train a model for all possible combination of loss functions (subsection 3.3) and data scalings (subsection 2.3, 128 models).

To determine the optimal data hyper-parameters we visually compare the generated images and their image quality metrics. Since we cannot consistently inspect the thousands of generated images, we first select the best performing models based on the evaluation metrics and before deciding the final data hyper-parameters based on both a quantitative and qualitative visual inspection. For the exact details of this process see Appendix D.

We correlate each hyper-parameter with the combined metric score and find that the sqrt data scaling performs the best. Fixing the image scaling to sqrt, we then determine the optimal loss function.

Based the models performance on the image quality metrics and visual inspections of the generated images we chose to train the final model with the PSNR combined with MS_SSIM loss function. Since the PSNR (L_{psnr}) and MS_SSIM ($L_{\mathrm{ms_ssim}}$) losses are at different scales we needed to normalize them in order to meaningfully combine

Metric	Input	Simulated Data	Real Data	Fine-Tuned (*XMM-DeNoise*)
L1	0.006528	0.005202	0.004628	0.004408
PSNR	39.349	41.728	42.227	42.693
Poisson	0.07616	0.04782	0.04856	0.04778
SSIM	0.9484	0.9359	0.9512	0.9567
MS_SSIM	0.9922	0.9910	0.9930	0.9939
FSIM	0.9688	0.9577	0.9745	0.9783
HaarPsi	0.8879	0.9006	0.9139	0.9253

Table 4. Various de-noising models to test the influence of the training data used when applied on the real test set. The input column refers to the direct comparison between the real input data and target image. We show the results for models trained on simulated data, real data and simulated then fine-tuned to real data compared to the target. The rows correspond to the different metric scores when applied to the real data test set.

them, as described in subsection 3.3. The final normalized combined loss (L_c) is defined as:

$$L_c = 5.43 - 0.0609\ L_{\text{psnr}} - 1.51\ L_{\text{ms_ssim}}$$

The final SR model was trained on the full simulated training dataset. For the DN models we trained on the full simulated training dataset, the full real training dataset and a combination of the two using transfer learning. We selected the best preforming model out of these three as our final DN model. We train the final models for 50 epochs using an Adam optimiser (Kingma & Ba 2014). After the training is complete we select the model from the epoch which achieved the best validation loss as our final model.

4.3 Transfer Learning

For the de-noising model we have access to real data. Whilst the simulated dataset contains more images, the real data encodes the domain that we are interested in. With the smaller dataset of real images, the performance of training a model on real data alone could be limited. Instead we make use of transfer learning (Tan et al. 2018) by taking the model trained on the larger simulated dataset and fine-tuning the weights to optimise for the real data. Fine-tuning is done by further training the model using the real data for another 50 epochs. We again select the model from the epoch which performed best on the validation loss as our final model.

5 RESULTS

5.1 XMM-Denoise

In Table 4 we quantify the performance of the best de-noising models trained either on simulated data, real data or simulated and later fine-tuned to real data when applied on a real data based test sample. The models trained on the real dataset generally score better than those trained on the simulated data. This is expected as the test set was based on real data and certain features present in the real data will not be present in the simulated data. The model that performed the best overall is the model that was first trained on simulated data and then fine-tuned on real data. We, therefore, select this as our final DN model, named *XMM-DeNoise*.

5.1.1 *Wavelet Comparison*

We qualitatively compare our *XMM-Denoise* model to the non machine learning based wavelet transform. The use of wavelet based de-noising methods has been shown to optimize the detection of AGNs, galaxy clusters and other features in X-ray images of different telescopes (e.g. Valtchanov et al. 2001; Faccioli et al. 2018; Xu et al. 2018; Zhang et al. 2020b. Our implementation is based on Faccioli et al. (2018).
In Figure 6 de-noised examples generated by *XMM-DeNoise* are shown compared to wavelet transformed image and the target image. The images are cropped to highlight the details. We can see that our de-noised images are, compared to the more smoothed wavelet transformed images, visually much closer to the target images. Note that for the wavelet technique the goal is not to mimic the higher exposure time image but to de-noise the images substantially. Certain features, such at the shock waves in W49B (top row) are better defined because of this. However, this also comes at the risk of having more artifacts or filtering out too much information — the wavelet transform will filter out regions with constant gradient, e.g. flat background. For example, in the M51 images (bottom row) we can see that the wavelet transformations filtered out the extended features of the source in the center left of the image. And using radially symmetric wavelet function will predominantly produce spherical morphologies.

5.2 XMM-SuperRes

Figure 7 shows a select few examples of generated super-resolution images. The generated images tend to contain more defined structures and more AGN. The performance of the *XMM-SuperRes* model based on the simulated test set are shown in Table 5. The metrics are calculated compared to the target image using the unscaled (linear) data. To be able to do a comparison between the input and the target images, we need to match their resolutions. We use a naive method, namely nearest-neighbour upsampling. Our model improved the input image on all metrics.

5.2.1 *Brightness Analysis*

To analyze the performance SR models in detail, we take vertical segments through the boresight of the input and generated images (Figure 8) and plot the pixel value distribution summed along the

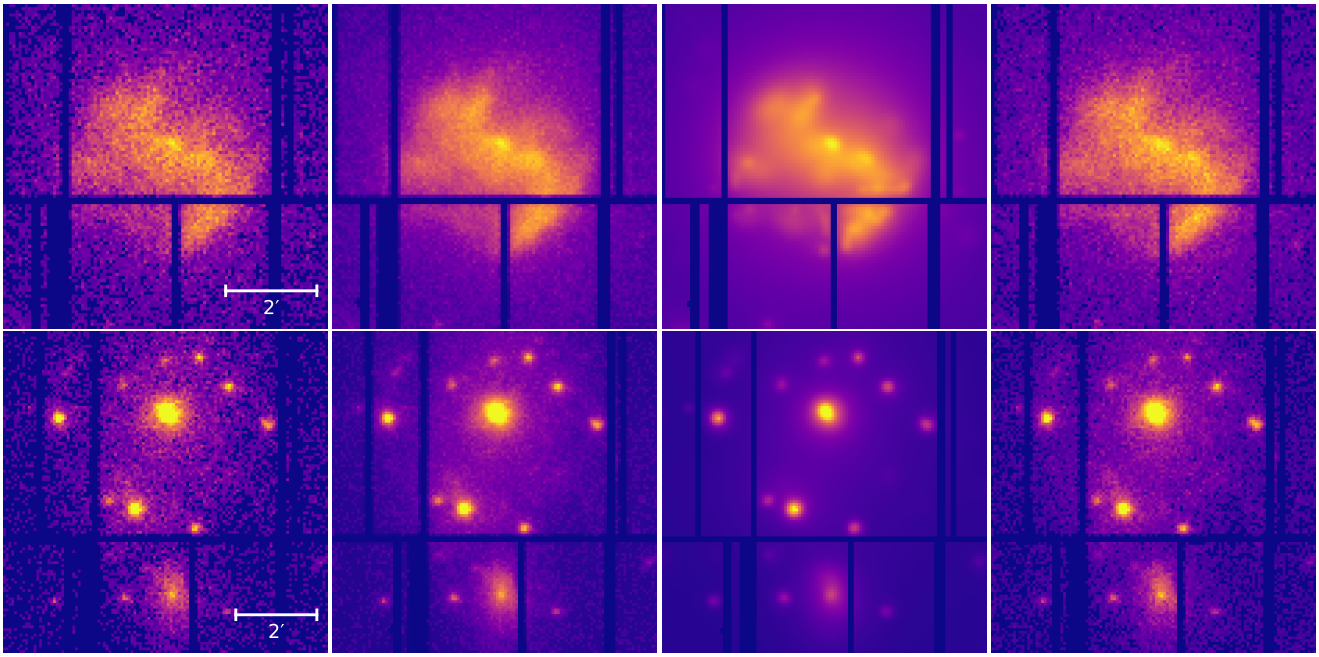

Figure 6. De-noised and wavelet transformed examples, W49B (top) and M51 (bottom), from the real *XMM-Newton* dataset. Cropped to the central source and scaled with the square root function. From *left* to *right*: Input image at 1x resolution with 20ks exposure, generated de-noised image for 50ks, wavelet transformed image and the target image at 50ks.

Metric	Input	Predicted (*XMM-SuperRes*)
L1	0.01096	0.006508
PSNR	33.525	38.034
Poisson	0.08285	0.04997
SSIM	0.8248	0.907
MS_SSIM	0.9499	0.9846
FSIM	0.8657	0.8688
HaarPsi	0.5312	0.697

Table 5. Super Resolution model metrics based on the simulated data test-set. The input column refers to the direct comparison between the simulated input and target image. The rows correspond to the different metric scores when applied to the simulated data test set compared to the target.

minor axis (Figure 9). The generated images are smoother than the input and target images; therefore, we smooth the input and target images using 1d convolution with a Gaussian kernel of size 5 and $\sigma = 1.0$ for a fairer comparison. Since the input image is at 20ks and the target image is at 100ks, the input image will have lower counts per region. The sudden drop in count values corresponds with the chip-gaps where the count is zero. The predicted *XMM-SuperRes* image (red) bares better resemblance to the target brightness in comparison to the input image.

5.2.2 *Chandra Comparison*

For the super-resolution model, it is not possible to learn the domain mapping for real *XMM-Newton* observations. However, we can probe its performance with real data by comparing SR-generated images with their Chandra counterparts.

As a qualitative measure of our results we compare our SR *XMM-Newton* generated images with Chandra observations of the same source. The *Chandra* images have a higher resolution of 0.5 arcsec HEW compared to the 17 arcsec of the *XMM-Newton* EPIC-pn. We use the full exposure time of the *Chandra* images. We do however stress that the properties of the two telescopes are not equivalent. The PSF of the two instruments are not the same and *XMM-Newton* is more sensitive than *Chandra*, so these images can not be considered as ground-truth. Nonetheless we present a few examples to cover a variety of fields and source morpholigies.

Our first case study is the Bullet cluster (Figure 10) a well know system of two interacting galaxy clusters. The cavity between the two X-ray components is enhanced in both the *Chandra* and the generated SR image in comparison to the input *XMM-Newton* image. Looking at the real *XMM-Newton* image and the SR-generated one, with white contours from *Chandra* overlayed, we can see that the cavity between the two clusters is much better defined in the SR and DN image compared to the original *XMM-Newton* image, the *Chandra* image also clearly contains this feature.

Our next case is supernova remnant W49B (Figure 11). Here, again, we see more pronounced features in the SR image in comparison to the input. The extended features on the top of the image seen in the *Chandra* contour lines are better defined in the generated SR image compared with the real *XMM-Newton* image.

Messier 51 (M51) is an interacting spiral galaxy with an active galactic nuclei, and it is another useful case study to see how the network performs (Figure 12). In this example we see that the generated image has point sources that are better defined compared to the real *XMM-Newton* image. For example, the faint source at the bottom left of the centre is clearly visible in the SR and *Chandra* image but are barely in the real *XMM-Newton* image. Looking at the real *XMM-Newton* image and generated SR *XMM-Newton* with in white

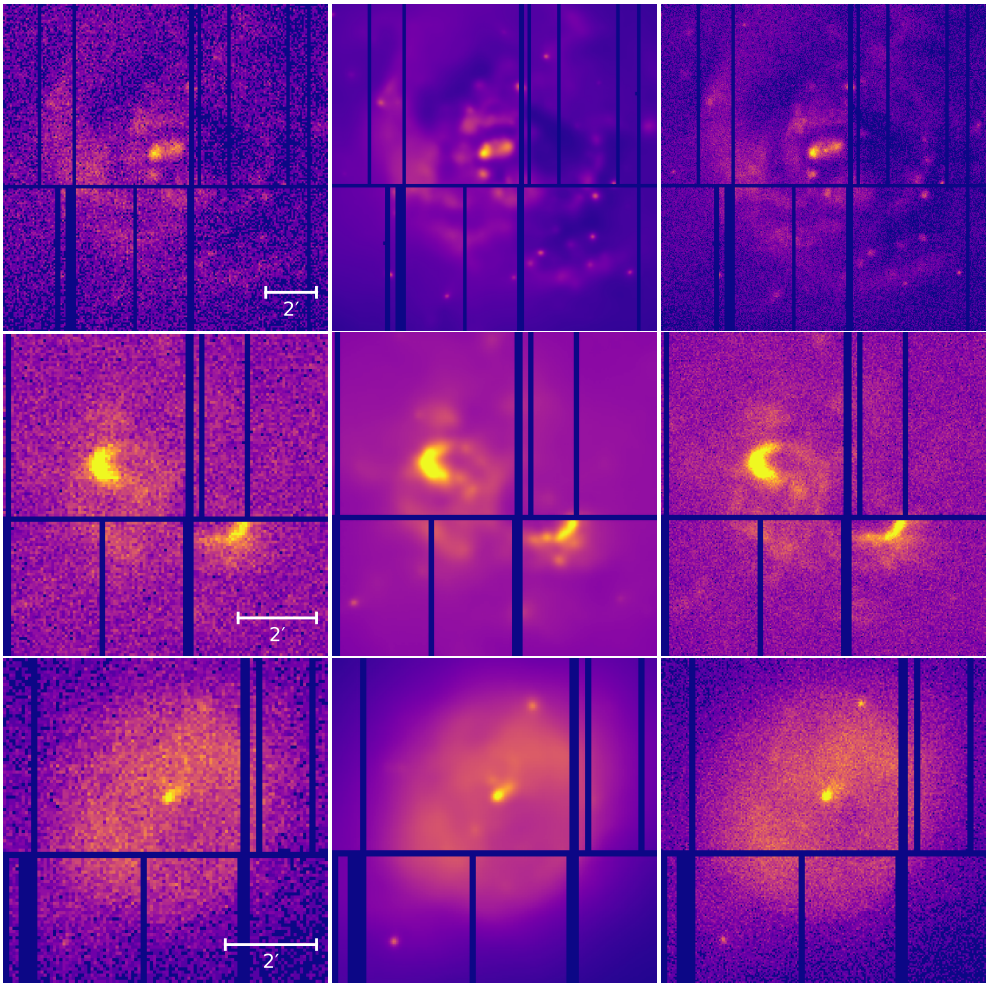

Figure 7. Super-resolution examples from the simulated *XMM-Newton* dataset. Cropped to the central source and scaled with the square root function. Each row from *top* to *bottom*: TNG50 Subhalo 382215, TNG100 Subhalo 41583 and TNG300 Subhalo 296363. From *left* to *right*: Input image at 1x resolution with 20ks exposure, generated super-resolution image for 100ks and the label image at 100ks without background noise.

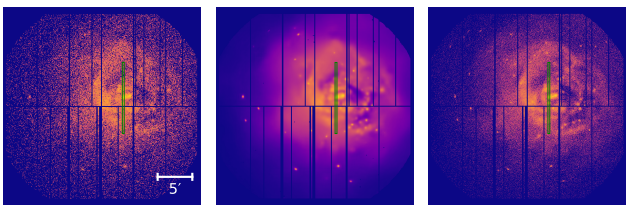

Figure 8. Strip plot regions TNG50 Subhalo 382215, scaled with a logaritmic funtion. From left to right: input image at 1x resolution at 20ks, *XMM-SuperRes* generated image with 2x resolution at 100ks and the target image with 2x resolution at 100ks without background noise. The green regions indicate the regions that we will analyse on brightness.

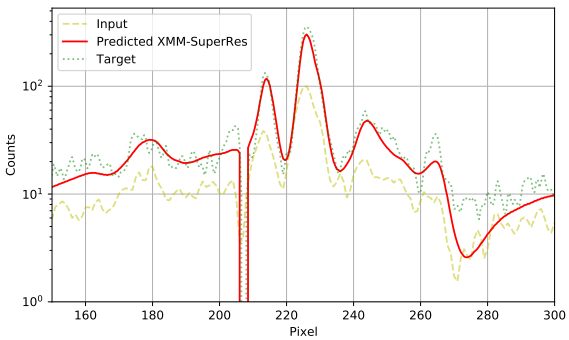

Figure 9. Count plots of the input image, *XMM-SuperRes* generated image and the target image of TNG50 Subhalo 382215 corresponding to the vertical cutout regions in Figure 8.

the contours of *Chandra* overlayed. We can also see that in the top left of the SR image an extended feature is visible that matches with the contours of *Chandra*, this extended feature is harder to be seen in the real *XMM-Newton* image. However, the SR image also does sometimes mis-predict features, for example on the right side the is a circular *blob* visible in the *Chandra* contours. In the real *XMM-Newton* image it is hard to tell if there is anything present. However, the SR model predicted almost no counts in that area.

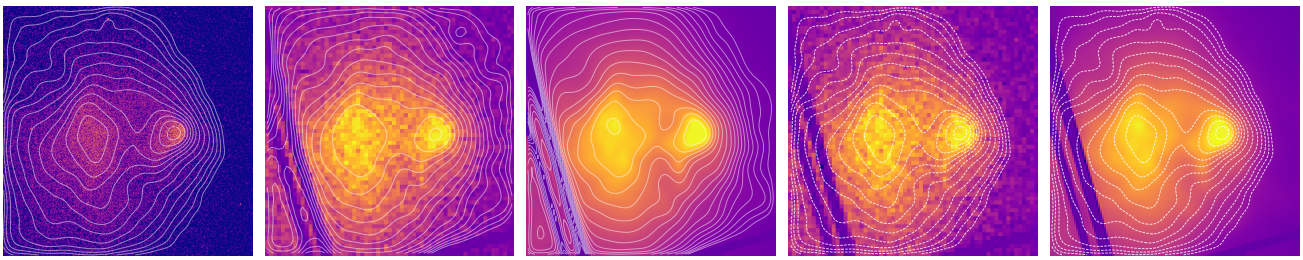

Figure 10. Images of the two colliding clusters of galaxies Bullet Cluster (1E 0657-56) with contours highlighted in white. Cropped to a frame size of 4.2‘. From *left* to *right*: *Chandra* at 88ks exposure, *XMM-Newton* at 20ks exposure, generated *XMM-Newton* SR, *XMM-Newton* at 20ks exposure overlayed with the *Chandra* contours and generated *XMM-Newton* SR overlayed with the *Chandra* contours.

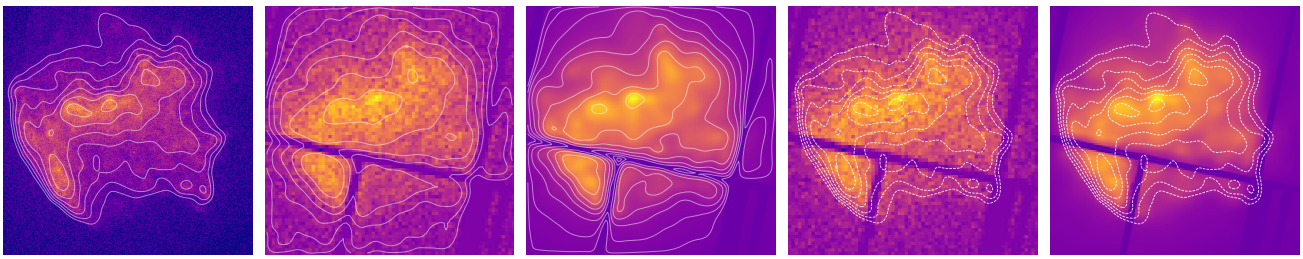

Figure 11. Images of the supernova remnant W49B (SNR G043.3-00.2), with contours highlighted in white. Cropped to a frame size of 5‘. From *left* to *right*: *Chandra* at 158ks exposure, *XMM-Newton* at 20ks exposure, generated *XMM-Newton* SR, *XMM-Newton* at 20ks exposure overlayed with the *Chandra* contours and generated *XMM-Newton* SR overlayed with the *Chandra* contours.

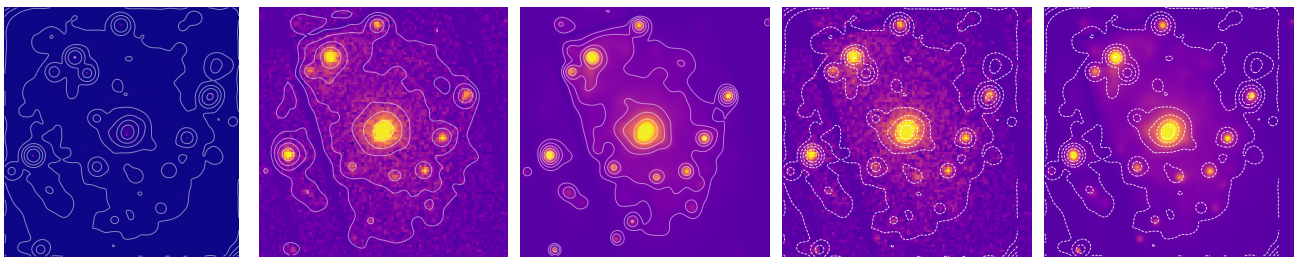

Figure 12. Images of the group of galaxies M51 with with contours highlighted in white. Cropped to a frame size of 6.7‘. From *left* to *right*: *Chandra* at 190ks exposure, *XMM-Newton* at 20ks exposure, generated *XMM-Newton* SR, *XMM-Newton* at 20ks exposure overlayed with the *Chandra* contours and generated *XMM-Newton* SR overlayed with the *Chandra* contours.

6 DISCUSSION

6.1 Detector Coordinates

Our models use the *XMM-Newton* images in detector coordinates instead of sky coordinates. This was done to make it easier for the models to learn the image imperfections such as the chip-gaps and bad pixels, which in detector coordinates are always at the same location. The exact pipeline we used for processing real observations is described in Appendix A.

6.2 Reliability

We have shown that our model is able to generate images with enhanced features that correspond well to features in the higher spatial resolution *Chandra* observations. However, due to the nature of deep-learning-based SR and DN, our reconstructed images are susceptible to "hallucinated" features. Multiple images can be consistent as the SR/DN counterpart to a given low-resolution/noisy image, however our model only predicts one possibility. Other approaches such as flow based models (Kobyzev et al. 2020) allow us to actively tune the generated image to coincide with different characteristics. We explored the use of the flow based super resolution model SR-Flow (Lugmayr et al. 2020) for SR/DN of *XMM-Newton* images, with promising initial results. With this approach we could tune the model to minimize artifacts, albeit with more blurry output images, or create perceptually realistic images with the compromise of an increased number of artefacts. To limit the scope of this paper, we did not continue further with this model, although it could be interesting for future research.

The goal of the metrics in Table 4 and Table 5 is to compare the models to each other and the input. We did not calculate the error on the metrics, e.g. by retraining with different initial random weights, since it would not add any constraints on the reliability of the individually generated outputs, because of the diversity of their contents. In addition, during the hyper-parameter tuning we observed similar performance of similar models within narrow margins of the validation loss (Figure C1) and therefore we expect that *XMM-SuperRes* and *XMM-DeNoise* will have similar performance after retraining.

6.3 AGN Deblending

Our research primarily focused on extended sources and less on AGNs. However, the models perform well at enhancing faint AGNs and deblending them. Deblending allows us to resolve two sources when the spatially separation is smaller than the telescope resolution. Future research could focus on using the *XMM-SuperRes* model for this purpose or even training a new SR and DN model specifically for deblending.

6.4 Limitations

6.4.1 Clipping

We clip extreme pixel values of the data used in this work to improve the training stability of our models, however this destroys information associated with bright sources. Since our main objective is to improve the resolution and de-noise to enhance the visibility of extended structures, losing detail in bright sources is an acceptable compromise. Some sources (not within our data sample) will have interesting features above our chosen clipping limit, and any user should take precaution when applying our model to sources of interest with count rates above our threshold. Whilst it is possible to retrain our models with a higher clipping threshold, we note that such action will affect other variables such as the data scaling function and loss.

6.4.2 Data Sample

The current models are limited to the energy range [0.5, 2] keV which corresponds to the energy range of the majority of extended source emission. We would advise against applying our model to images in different energy bands as whilst the PSF is only marginally dependent on the energy range, the vignetting and noise properties on the other hand are known to be energy-dependent. Also, our model inputs are 20 ks exposure time observations from the *XMM-Newton* EPIC-pn sensor since it has the largest effective area and, therefore, good spectral and spatial resolutions. We argue that the domain chosen for this research is sufficient to show the effectiveness of our proof-of-concept method. A future extension to this work could look into expanding the energy range, incorporating the MOS detectors, incorporating spectral information and increasing the flexibility of the input image exposure time.

6.4.3 Simulations

Our simulations do not contain telescope properties such as out-of-time events. These phenomena tend to be caused by extremely bright sources. These are not the sources that we are interested in this research and therefore it is more efficient for us to negate out-of-time events in our simulator.

7 CONCLUSIONS

We have developed deep-learning-based super-resolution (SR) and denoising (DN) models to enhance *XMM-Newton* X-ray EPIC images. As a proof of concept, we only considered the EPIC-pn detector and images with photon events with energies in [0.5,2] keV. We increase the resolution of the observations and de-noise to improve the SNR and enhance features that are challenging to locate in the original images.

To train the SR and DN models, we simulated EPIC-pn images with twice the nominal spatial resolution and images with larger exposure times. We explored the influence of the model architecture parameters, data pre-processing, and loss functions on the model's performance. To enhance the image quality, we proposed using a combined loss function consisting of both PSRN and MS_SSIM. To address the problem of the high dynamical range of pixel values present in X-ray images, we implemented data-scaling with different stretch functions. We showed that using suitable data-scaling, our models generated fewer artifacts in low surface brightness areas in extended sources while preserving the details we are interested in.

Our SR and DN model (*XMM-SuperRes*) generates enhanced SR and DN images with twice the spatial resolution and an improved image quality metrics as quantified by the PSNR. The network-produced images have the desired properties, such as a smaller PSF, and all the tested image quality metrics were improved when the model was applied on the test dataset. Specifically, with the simulated datasets, it improved the PSNR by 21.5% and reduced the L1 by 40.3%. We have validated the performance of the model by applying it on real data and visually comparing with observations taken by NASA's *Chandra* telescope, which has much higher spatial resolution.

We find that the model produce images that are able to enhance features with obvious counterparts in the *Chandra* observations. Nevertheless, due to the nature of the reconstruction, some of the generated SR features may be spurious; hence whilst this model may be able to find and uncover interesting details, further detailed analysis and ideally follow-up observations at higher spatial resolution will be needed for their confirmation.

Our denoising model (*XMM-DeNoise*) based on *XMM-SuperRes*, generates images with 2.5 times higher exposure without increasing the resolution. This enabled us to train and validate the model with real *XMM-Newton* observations. We found that training the denoising model on simulated data first and fine-tuning it on real data resulted in the best results for most image quality metrics. *XMM-DeNoise* similarly improves the quality of real *XMM-Newton* images on all measured global quality metrics. Specifically, it improved the PSNR by 8.15% and reduced the L1 by 38.4%.

In conclusion, we have demonstrated the feasibility of using deep-learning models to improve the spatial resolution and denoising of *XMM-Newton* EPIC-pn X-ray astronomy images to increase their scientific value. The *XMM-SuperRes* and *XMM-DeNoise* models developed in this paper could be used as a proof-of-concept to create more elaborated methods. Such as creating a model that can output a range of possible SR and DN images emphasising on different characteristics (e.g. shock fronts in supernova remnants or deblending of point sources) to more directly tackle the ill-posed nature of the problem. The next steps for future work after this pilot study are obvious: training the models with the other *XMM-Newton* instruments, incorporating a set of different energy ranges and exposure times and also extend it to other current and future X-ray telescopes.

ACKNOWLEDGEMENTS

We would like to thank the anonymous referee for the valuable and constructive comments that helped improve the paper. We also thank the *XMM-Newton* Project Scientist, Norbert Schartel, for many helpful discussions during the initiation and the development of the study. We appreciate the valuable assistance from the SIXTE team. Based on observations obtained with *XMM-Newton* , an ESA science mission with instruments and contributions directly funded by ESA Member States and NASA. The IllustrisTNG simulations were

undertaken with compute time awarded by the Gauss Centre for Supercomputing (GCS) under GCS Large-Scale Projects GCS-ILLU and GCS-DWAR on the GCS share of the supercomputer Hazel Hen at the High Performance Computing Center Stuttgart (HLRS), as well as on the machines of the Max Planck Computing and Data Facility (MPCDF) in Garching, Germany. SFS acknowledges support from the ESA traineeship program and further support from ESA through the Science Faculty of the European Space Astronomy Centre (ESAC) — Funding reference ESA-SCI-SC-LE-033. ML acknowledges a Machine Learning and Cosmology research fellowship from the University of Nottingham.

DATA AVAILABILITY

More detail about our data generation process is provided in Appendix A and Appendix B. Data will be available on request. Our code is publicly available. The EPIC-pn simulator and dataset generation code is available on `https://github.com/SamSweere/xmm-epicpn-simulator` and the code to train and run inference on the SR and DN models is available on `https://github.com/SamSweere/xmm-superres-denoise`, as well as more implementation details in the SFS's Master's thesis which is included in the GitHub folder.

REFERENCES

Arnaud K. A., 1996, in Jacoby G. H., Barnes J., eds, Astronomical Society of the Pacific Conference Series Vol. 101, Astronomical Data Analysis Software and Systems V. p. 17
Bourdin H., Slezak E., Bijaoui A., Arnaud M., 2001, Clusters of Galaxies and the High Redshift Universe Observed in X-rays, p. 26
Carter J., Read A., 2007, å, 464, 1155
Chen C., Chen Q., Xu J., Koltun V., 2018, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 3291–3300
Chen H., He X., Qing L., Wu Y., Ren C., Sheriff R. E., Zhu C., 2022, Information Fusion, 79, 124
Dauser T., et al., 2019, A&A, 630, A66
Dong C., Loy C. C., He K., Tang X., 2014, in Fleet D., Pajdla T., Schiele B., Tuytelaars T., eds, Computer Vision – ECCV 2014. Springer International Publishing, Cham, pp 184–199
Faccioli L., et al., 2018, Astronomy & Astrophysics, 620, A9
Feng H., Chen Y., Zhang S. N., Lu F. J., Li T. P., 2003, A&A, 402, 1151
Gilli R., Comastri A., Hasinger G., 2007, A&A, 463, 79
Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014, Advances in neural information processing systems, 27
Iandola F., Moskewicz M., Karayev S., Girshick R., Darrell T., Keutzer K., 2014, arXiv preprint arXiv:1404.1869
Jain V., Seung S., 2008, Advances in neural information processing systems, 21
Jansen F., et al., 2001, A&A, 365, L1
Johnson J., Alahi A., Fei-Fei L., 2016, in Leibe B., Matas J., Sebe N., Welling M., eds, Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp 694–711
Kingma D. P., Ba J., 2014, arXiv preprint arXiv:1412.6980
Kobyzev I., Prince S. J., Brubaker M. A., 2020, IEEE transactions on pattern analysis and machine intelligence, 43, 3964
Lauritsen L., Dickinson H., Bromley J., Serjeant S., Lim C.-F., Gao Z.-K., Wang W.-H., 2021, arXiv preprint arXiv:2102.06222
LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., 1989, Neural computation, 1, 541
LeCun Y., Bottou L., Bengio Y., Haffner P., 1998, Proceedings of the IEEE, 86, 2278
Ledig C., et al., 2017, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Li Z., Peng Q., Bhanu B., Zhang Q., He H., 2018, Ap&SS, 363, 1
Lugmayr A., Danelljan M., Van Gool L., Timofte R., 2020, in Vedaldi A., Bischof H., Brox T., Frahm J.-M., eds, Computer Vision – ECCV 2020. Springer International Publishing, Cham, pp 715–732
Marinacci F., et al., 2018, MNRAS, 480, 5113
Naiman J. P., et al., 2018, MNRAS, 477, 1206
Nelson D., et al., 2018, MNRAS, 475, 624
Nelson D., et al., 2019, MNRAS, 490, 3234
Pillepich A., et al., 2018, MNRAS, 475, 648
Pillepich A., et al., 2019, MNRAS, 490, 3196
Puschmann K. G., Kneer F., 2005, A&A, 436, 373
Reisenhofer R., Bosse S., Kutyniok G., Wiegand T., 2018, Signal Processing: Image Communication, 61, 33
Sanders J., Fabian A., 2001, MNRAS, 325, 178
Sanders J. S., Fabian A. C., Russell H. R., Walker S. A., Blundell K. M., 2016, MNRAS, 460, 1898
Santos-Lleo M., Schartel N., Tananbaum H., Tucker W., Weisskopf M. C., 2009, Nature, 462, 997
Schawinski K., Zhang C., Zhang H., Fowler L., Santhanam G. K., 2017, MNRAS, p. slx008
Shi W., Caballero J., Huszár F., Totz J., Aitken A. P., Bishop R., Rueckert D., Wang Z., 2016, in Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1874–1883
Siu W.-C., Hung K.-W., 2012, in Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference. pp 1–10
Springel V., et al., 2018, MNRAS, 475, 676
Starck J.-L., Pantin E., Murtagh F., 2002, PASP, 114, 1051
Strüder L., et al., 2001, A&A, 365, L18
Su Y., et al., 2020, Monthly Notices of the Royal Astronomical Society, 498, 5620
Tan C., Sun F., Kong T., Zhang W., Yang C., Liu C., 2018, in Kůrková V., Manolopoulos Y., Hammer B., Iliadis L., Maglogiannis I., eds, Artificial Neural Networks and Machine Learning – ICANN 2018. Springer International Publishing, Cham, pp 270–279
Turner M. J., et al., 2001, Astronomy & Astrophysics, 365, L27
Valtchanov I., Pierre M., Gastaud R., 2001, A&A, 370, 689
Vojtekova A., Lieu M., Valtchanov I., Altieri B., Old L., Chen Q., Hroch F., 2020, Monthly Notices of the Royal Astronomical Society, 503, 3204
Wang Z., Simoncelli E. P., Bovik A. C., 2003, in The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. pp 1398–1402
Wang X., Yu K., Wu S., Gu J., Liu Y., Dong C., Qiao Y., Change Loy C., 2018, in Proceedings of the European conference on computer vision (ECCV) workshops. pp 0–0
Wang Z., Chen J., Hoi S. C. H., 2021, IEEE Transactions on Pattern Analysis and Machine Intelligence, 43, 3365
Weisskopf M. C., Tananbaum H. D., Speybroeck L. P. V., O'Dell S. L., 2000, in Truemper J. E., Aschenbach B., eds, Vol. 4012, X-Ray Optics, Instruments, and Missions III. SPIE, pp 2 – 16, doi:10.1117/12.391545, `https://doi.org/10.1117/12.391545`
Wells D. C., Greisen E. W., 1979, in Sedmak G., Capaccioli M., Allen R. J., eds, Image Processing in Astronomy. p. 445
Wilkins D., Gallo L., Costantini E., Brandt W., Blandford R., 2021, Nature, 595, 657
Xu W., Ramos-Ceja M. E., Pacaud F., Reiprich T. H., Erben T., 2018, Astronomy & Astrophysics, 619, A162
Yang W., Zhang X., Tian Y., Wang W., Xue J.-H., Liao Q., 2019, IEEE Transactions on Multimedia, 21, 3106
Zhang L., Zhang L., Mou X., Zhang D., 2011, IEEE transactions on Image Processing, 20, 2378
Zhang X., Chen Q., Ng R., Koltun V., 2019, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 3762–3770
Zhang Y., Tian Y., Kong Y., Zhong B., Fu Y., 2020a, IEEE Transactions on Pattern Analysis and Machine Intelligence, 43, 2480
Zhang C., Ramos-Ceja M. E., Pacaud F., Reiprich T. H., 2020b, Astronomy & Astrophysics, 642, A17

Zhou Wang Bovik A. C., Sheikh H. R., Simoncelli E. P., 2004, IEEE Transactions on Image Processing, 13, 600
Zhou F., Yang W., Liao Q., 2012, IEEE Transactions on Image Processing, 21, 3312

APPENDIX A: REAL XMM-NEWTON DATASET GENERATION

The real *XMM-Newton* dataset was created using standard workflow.

- We start off with the full *XMM-Newton* archive, containing all the historical observations.
- First, we select all observations of at least $20ks$ observation time using either full-frame or extended full-frame mode. These modes use all the 12 EPIC-pn CCDs.
- We use the *XMM-Newton* pipeline produced products (PPS) for calibrated eventlists. Sometimes, an observation is split into different exposures and we select the one with the longest on-time.
- We filter the eventlist by removing intervals of high background. We use the PPS-generated light-curve and the PPS-derived threshold and we apply the following filtering expression to produce cleaned event lists:

```
PI > 300 && RAWY > 12 && PATTERN <= 4 &&
    ((FLAG == 0) || ((FLAG & 0x10000) != 0))
```

  We keep the out of field-of-view events (flagged with `0x10000`) in the corners to control the instrumental background and compare with SIXTE simulations.
- We convert this eventlist into smaller eventlists of different exposure times with increments of 10 ks. The biggest exposure time depends on the exposure time of the cleaned eventlist. I.e. if we have a clean exposure of 40 ks, we will generate 4x10 ks images, 2x20 ks images, 1x30 ks, and 1x40 ks images. The images with multiple exposure times make the dataset more flexible to use. It also enables us to train a de-noising model with low and high exposure image pairs.
- Finally, using the cleaned event lists we create images in detector coordinates. We use the default binsize of 80 (4"/pix). The final image is saved in the FITS format Wells & Greisen (1979).

APPENDIX B: SIMULATION SETUP DETAILS

B1 EPIC-pn Image Simulation

The EPIC-pn sensor consists of 12 CCDs. Initially, we simulated all these separately based on the physical properties, including out-of-time events. However, this significantly increased the computation time of the simulator by a factor of 12 since the whole telescope has to be simulated separately for each sensor. A single observation would require 12 separate simulations. The benefits of simulating every sensor separately are that specific properties such as out-of-time events are simulated as well. However, the sources we are interested in are usually not extremely bright, which is the main cause of out-of-time events. The impact of not having these properties is minimal on the final image.

Therefore we simulate all 12 sensors as one big sensor, without out-of-time events. Since we now do not have any chip gaps in-between the CCDs, we multiply the final image with the *XMM-Newton* detector mask. The detector mask filters out all the areas in the image where no recording of events is possible, including the chip gaps, known bad pixels, and areas outside the field of view. Additionally we filter out over-exposed images that can result in undesirable effects. These are typically associated solar flares. We use the exposure map as a detector mask. The resolution of this exposure map matches the resolution of the observed/simulated images at 1x scale. The detector mask can therefore be used both on the real image and the simulated one. For the higher resolution we increase the resolution of the detector mask without interpolation (by repeating every pixel).

B2 Boresight Determination

Note that the optical axis (also called boresight) is not exactly in the middle of the image but is slightly offset, in order to avoid a chip gap. The boresight position also changed over time. We used the information from the latest calibration file: XMM_MISCDATA_0022.CCF to determine the position of the optical axis for the simulations. This is important for the vignetting and the PSF, since these depend on the off-axis angle.

B3 PSF

The PSF (point spread function) of the *XMM-Newton* is not constant, this also needed to be simulated. In SIXTE, there are two PSF implementations: Using a single PSF for the whole image or setting separate PSFs for certain sections. These sections are radially distributed, centering around the boresight using a polar coordinate system. For every X-ray photon of specific energy entering the simulated telescope, SIXTE will then use the closest given PSF for that specific energy and location. The PSF distributions that SIXTE uses need to be provided as images. During development, this created a problem since providing many PSF images, which make the simulation more realistic, resulted in very high memory use, limiting the number of simulations we could run in parallel.

We decided to use three different energy levels: 0.5, 1.0 and 2.0 KeV to optimize this. Use a ϕ degree interval of 4 degrees, and $\theta = 0, 210, 420, 600, 720, 900, 1200$ arcsec. Resulting in 630 unique PSF images for every energy level. The PSF image resolution was set to 120x120; this is just big enough to cover the most stretched PSF at the edge of the sensor. The PSF images were created using the *psfgen* program, which is part of the official *XMM-Newton* Science Analysis System (XMM-SAS). To increase the simulation's spatial resolution, we have to decrease the PSF size. However, we do want to keep the same PSF distortion shape. Therefore, we decreased the size of the original PSF images by the resolution multiplier.

APPENDIX C: MODEL HYPER-PARAMETER TUNING

To determine the model hyper-parameters we run a hyper-parameter-search. Where we fix the loss to Poisson and use square root data-scaling since this resulted in desirable results in initial testing. The model hyper-parameters we try and their ranges are:

- Number of RRDB convolutional filters: $[8, 16, 32]$
- Number of RRDB blocks: $[2, 4, 8]$
- Learning rate: $[10^{-3}, 5\times10^{-3}, 10^{-4}, 5\times10^{-4}, 10^{-5}, 5\times10^{-5}]$
- Batch size: $[2, 4, 8]$

We train models for every possible combination of these hyper-parameters (180 models) and monitor the loss of the validation data. Several runs fail to converge. Some of these runs result in a validation loss greater or equal than 1.0, and only generate blank images.

Filtering out failed and poorly performing runs (val/loss ≥ 0.434) still leaves us with a huge number of viable model hyper-parameter

combinations. We therefore look at the correlation of the parameters with the loss.

The batch size has the largest positive correlation with the final loss. This positive correlation indicates that the bigger the $batch_size$, the worse the performance. After discarding runs that use a batch size of 8, we find that the next biggest correlation comes from the learning rate. It was clear that many of the failed runs were a result of high learning rates that make the training unstable but it's also known that small learning rates risk getting stuck at local minima. We therefore discard the extreme learning rates $lr > 0.0001$ and $lr < 0.00005$.

Figure C1 shows the remaining runs after filtering, all of which result in very similar validation loss values. Since these models are trained on a cropped image and reduced dataset size, we can assume that there is more to learn from the data and that a larger model would be more suitable for the real run. However, bigger models also take longer to train and use more GPU memory, which is a limiting factor when processing full-size images. For the final model (both SR and DN), we opt for 32 RRDB convolutional filters and 4 RRDB blocks to leave room to learn more complex data without hitting our GPU memory limitation. We opt for a batch size of 1 to similarly reduce the computational strain on the GPU and a learning rate of 0.0001.

APPENDIX D: DATA HYPER-PARAMETER TUNING

To tune the data hyper-parameters, the loss function and data scaling, we fix the model hyper-parameters to the values determined in subsection 4.1 but with a batch size of 4 to increase the training speed. Next, we train a model with every possible combination of loss functions (subsection 3.3) and data scalings (subsection 2.3), this results in 128 models.

The data hyper-parameters directly influence the visual properties of the generated images. Therefore, to determine their optimal values, we select the best-performing models based on the image quality metrics first and then do a qualitative visual inspection.

Each image quality metric emphasizes different visual elements in the generated image so we define our quantitative measure as a metric score that combines all the metrics into a single value. Some metrics are ascending, and others are descending so we invert ascending metrics such that all metrics are descending. Additionally, since the metrics map to different numerical scales, we apply a min-max normalization to their values before they are summed to create the combined metric score. Here, a lower combined metric score is better. Since images with different data scalings have different properties, the metrics' value is also differs.

We correlate each hyper-parameter with the combined metric score (Figure D1) and find that the logarithmic scale correlates heavily with bad performance on all metrics. The asinh scale also under-performs with respect to the sqrt and linear scale. The sup-par performance of asinh and log is likely due to the noise level getting pushed close to the structure level, making it difficult to distinguish between the unpredictable background noise and any real features. The linear and sqrt data scalings perform the best.

Visual inspection of the linear data-scaling models show the tendency of generating patchy images that are not present in the ground truth image. These artefacts are barely visible on linear scales, but they become problematic when the image is stretched. Models trained with a sqrt data scaling suffer less from this problem, see Figure D2, which motivates our choice of sqrt data scaling for our final model.

Fixing the image scaling to sqrt, we then determine the optimal loss function. Again we correlate the loss function with respect to the combined metric score (Figure D3), and find that L1 loss only performs well with respect to the L1 evaluation metric and performs poorly with respect to all other metrics. As one might expect, the best correlations occur where the chosen loss function is also used as the evaluation metric.

When visually inspecting the generated images we observed that models trained with the SSIM loss tend to contain overly defined structures and AGNs in comparison to the target image. While models trained with Poisson, PSNR and MS_SSIM where visually closer to the target images. Models trained with L1 loss seem to suffer from a quantization problem, where there are distinct regions visible in what should be continuous distributed area. Based on these observations and the models performance on the image quality metrics we chose to train the final model with the PSNR combined with MS_SSIM loss function.

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

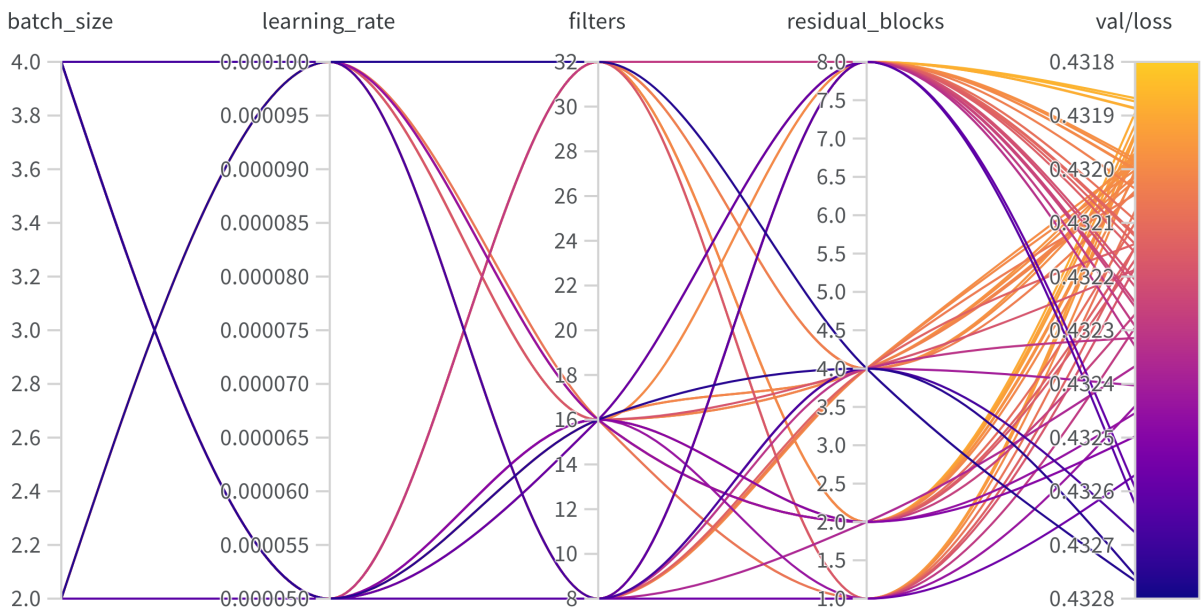

Figure C1. Results model sweep. With $val/loss >= 0.434$, $batch_size <= 4$ and $0.0001 >= learning_rate >= 0.00005$.

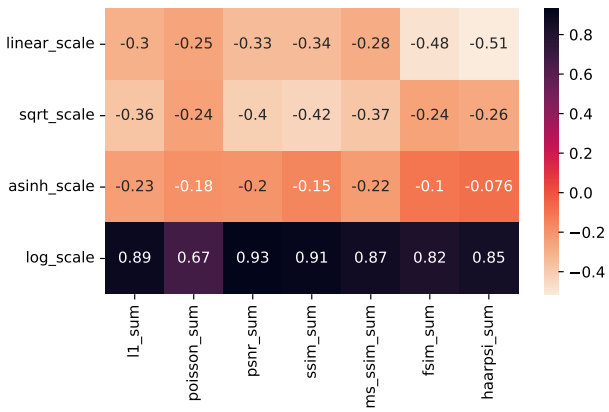

Figure D1. Correlation matrix data scaling with respect to image quality metrics. Lower correlation indicates better performance.

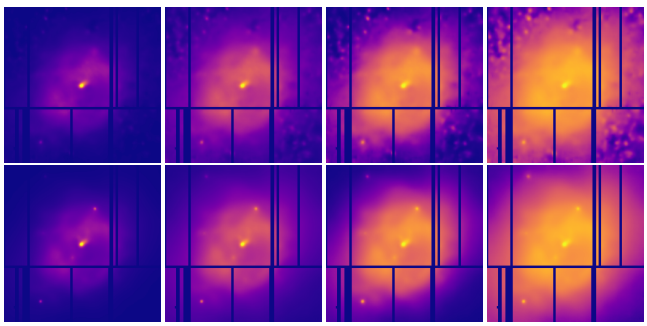

Figure D2. A model trained at linear data scale (top row) and a model trained at the sqrt data scale (bottom row). The display data scales from left to right: linear, sqrt, asinh, and log. At the asinh and log data scale, the *blobs* generated by the linear trained model are not present with the sqrt trained model.

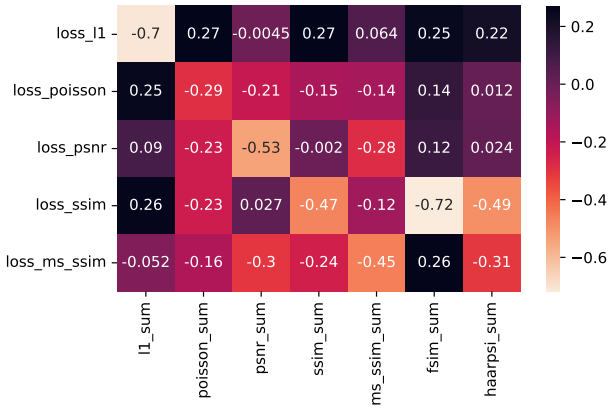

Figure D3. Correlation of the loss function elements with respect to the summed normalized metrics, given sqrt data scaling. Lower correlation indicates better performance.