Astronomy & Astrophysics manuscript no. output
October 7, 2025

RGC: a radio AGN classifier based on deep learning

I. A semi-supervised model for the VLA images of bent radio AGNs

M. S. Hossain[1], M. S. H. Shahal[2,3], K. M. B. Asad[2,4], P. Saikia[2,5], A. Khan[1,2], F. Akter[6], A. A. Ali[1,3], M. A. Amin[1,3], A. Momen[1,2,4], M. Hasan[3], and A. K. M. M. Rahman[1,3]

[1] Center for Computational and Data Sciences, Independent University, Bangladesh, Dhaka 1229, Bangladesh
e-mail: sazzat@iub.edu.bd
[2] Center for Astronomy, Space Science and Astrophysics, Independent University, Bangladesh, Dhaka 1229, Bangladesh
e-mail: kasad@iub.edu.bd
[3] Department of Computer Science and Engineering, Independent University, Bangladesh, Dhaka 1229, Bangladesh
[4] Department of Physical Sciences, Independent University, Bangladesh, Dhaka 1229, Bangladesh
[5] Department of Astronomy and Physics, Yale University, New Haven, CT 06511, USA
[6] Department of Agricultural and Biosystems Engineering, North Dakota State University, Fargo, ND 58108, USA

Received ; accepted

ABSTRACT

Context. Bent radio active galactic nuclei (RAGNs), such as wide-angle tail (WAT) and narrow-angle tail (NAT) sources, are important tracers of dense environments in galaxy groups and clusters. However, to date no machine-learning classifier of WATs and NATs has been trained using both unlabeled data and purely visually inspected labeled samples.
Aims. We release the RGC Python package, which provides two newly pre-processed labeled datasets of 639 WATs and NATs derived from a publicly available catalog of visually inspected sources, together with a semi-supervised RGC model that leverages 20,000 unlabeled RAGNs.
Methods. The two labeled datasets in RGC were pre-processed with PyBDSF, which retains spurious sources, and with Photutils, which removes them. The RGC model combines the self-supervised framework BYOL with the group-equivariant supervised model E2CNN to construct a semi-supervised binary classifier.
Results. The RGC model trained and tested on the dataset with spurious sources removed achieves the best performance, with an accuracy of 88.9% and F1-scores of 0.90 for WATs and 0.85 for NATs.
Conclusions. The performance of RGC improves with the removal of spurious sources, despite the high class imbalance, as demonstrated in our discussions on model calibration, attention and transparency. This binary classifier can make a significant contribution to the development of future foundation models in radio astronomy.

Key words. Techniques: image processing – Methods: data analysis – Methods: statistical – Galaxies: active – Radio continuum: galaxies

1. Introduction

In this paper, we present Version 1.0 of RGC,[1] a radio AGN (active galactic nuclei)[2] classifier named after Radha Gobinda Chandra (1878–1975), a Bangladeshi-Indian amateur astronomer who contributed more than fifty thousand observations to the American Association of Variable Star Observers (Maitra 2021) and reported the observation of 1P/Halley in 1910 in Bangla (Kapoor 2023). RGC uses convolutional neural netowrks (CNN) to classify radio AGN (RAGN) with straight and bent tails created from synchrotron jets. Its performance in classifying these objects into Fanaroff–Riley (FR) types I and II was described by Hossain et al. (2023, hereafter H23). In this work, we present and analyze its performance in classifying bent RAGNs into wide-angle-tail (WAT) and narrow-angle-tail (NAT) sources. A background to the present work is given below.

The observable universe could contain almost two trillion galaxies (Conselice et al. 2016), but only a fraction of them have been observed at different wavelengths. The x-ray space telescope eROSITA is expected to find thousands of galaxy clusters and millions of AGNs (Predehl et al. 2021). Samples of 700 thousand AGNs (Merloni et al. 2024) and 12 thousand galaxy clusters and groups (Bulbul et al. 2024) have already been published. In visible light, the eleventh and twelfth data releases of SDSS-III[3] contained more than 1.3 million galaxies and more than a quarter of a million quasars (Alam et al. 2015). The mid-infrared space telescope WISE (Wide-field Infrared Survey Explorer) has identified almost 750 million radio sources which are part of its AllWISE data release (Kurcz et al. 2016). At mid radio frequencies, the Faint Images of the Radio Sky at Twenty-Centimeters (FIRST) taken using the Karl G. Jansky Very Large Array (VLA) contains more than 800 thousand sources in its April 2003 release (Proctor 2011); the initial catalog of Becker et al. (1995) contained 20 thousand sources. At a lower frequency, the Evolutionary Map of the Universe (EMU) conducted using the Australian Square Kilometre Array Pathfinder (ASKAP) contained more than 200 thousand sources

[1] https://github.com/cassaiub/rgc
[2] The radio AGNs we classify here are sometimes called radio galaxies, but we use the more general term 'radio AGN' (RAGN) throughout.
[3] Sloan Digital Sky Survey

in its pilot survey and it is expected to find many more (Norris et al. 2021). Moreover, at the lowest radio frequencies, LOw Frequency ARray (LOFAR) has detected almost 4.4 million sources as part of its Two-metre Sky Survey LOTSS (Shimwell et al. 2022).

Astronomers are also cross-matching the sources seen at different wavelengths, especially the AGNs with their host galaxies. For example, Best & Heckman (2012) cross-matched FIRST and SDSS with NVSS (NRAO[4] VLA Sky Survey) to produce a catalog of more than 18 thousand sources. Traditionally cross-matching has been achieved through visual inspection by astronomers which is not feasible for the latest generation of surveys. Therefore, Banfield et al. (2015) created the citizen science project Radio galaxy Zoo (RGZ) where users were asked to cross-identify radio images from FIRST and ATLAS (Australia Telescope Large Area Survey, a pathfinder of EMU) with their host galaxies imaged by WISE or Spitzer Space Telescope in infrared. Data Release 1 of RGZ contains classifications of 99,146 radio sources from FIRST, with an average reliability of 0.83 based on the weighted consensus levels of the citizen scientists (Wong et al. 2025).

If we focus on the history of 'radio' continuum surveys from the days of Grote Reber until the last decade, we see that the number of detected sources increased by an order of four from the 1940s to the 1970s, and since then we have seen an increase by four more orders of magnitude (Figure 2 of Norris 2017). EMU and LOFAR are expected to find close to 100 million radio sources. The number will increase more when the Square Kilometer Array (SKA) begins to operate in two phases, SKA1 and SKA2, and at two different frequency bands, SKA-Low (50–350 MHz) and SKA-Mid (350 MHz – 15 GHz). At mid-frequencies, SKA1 All Sky Survey (SASS1) will detect around 500 million galaxies and SASS2 over 3 billion galaxies spanning all redshifts (Norris et al. 2015). The first of the 131,072 two-meter-tall Christmas-tree-shaped antennas of SKA-Low was installed in a Wajarri country in Australia on 7 March 2024 and the first of the 197 fifteen-meter-wide dishes of SKA-Mid was lifted onto a pedestal on 4 July 2024 in the Karoo region of South Africa.[5]

In this context, artificial intelligence (AI) is necessary for localizing astronomical sources in large datasets and classifying them into scientifically meaningful classes. A recent review by Ndung'u et al. (2023) emphasizes on the new paradigm shift that has happened due to the application of machine learning (ML) and deep learning (DL) for the morphological classification of radio AGNs. They have reviewed 32 papers published between 2017 and 2023 that utilized both conventional ML and DL methods. The most frequently used methods were found to be based on shallow and deep convolutional neural networks (CNN). The methods were divided into model-centric and data-centric approaches. The model-centric approaches focus on applying novel architectures on existing well-curated datasets which are scarce. Most methods use low-resolution images taken by older telescopes such as, VLA. These are sometimes not suited to the new high-resolution images coming out of the latest surveys from the new generations of telescopes, for example, LOFAR and MeerKAT. The data-centric approaches leverage the availability of more numerous images and focus on transfer learning and semi-supervised learning.

Our data-centric approach is based on semi-supervised learning, where we leverage the availability of unlabeled data for classifying bent RAGNs. The model used here is the same as that of H23, but we use it for classifying bent RAGNs for the first time. The paper is organized as follows. Section 2 gives an overview of bent RAGNs and Section 3 gives a detailed description of the data we have prepared for an efficient use in DL. Section 4 presents our semi-supervised model in a more detailed manner than H23. Both Sections 3 and 4 refer to the installable Python package `RGC`[6] throughout, which is released along with this paper. Section 5 describes the performance of the model in classifying the bent RAGNs of our datasets, and Section 6 gives a critical discussion of the limitations and prospects of the model. We conclude the paper with Section 7.

2. Bent radio active galactic nuclei

AGNs are the centers of galaxies that produce extremely high luminosities from an extremely small region. They are made of a supermassive black hole (SMBH) surrounded by a thin accretion disk and a thick torus. The gravitational potential energy of the SMBH is the source of their luminosity. Outflows of high-energy particles are found along the poles of the disk or torus. These bipolar jets and their tails emit radio light through the synchrotron mechanism, creating the radio, or radio-loud, AGNs, the so-called RAGNs (Urry & Padovani 1995). If the brightness of the jets decreases from the center toward the edges, the RAGNs are called FRI sources, and, conversely, if the brightness increases toward the edges, they are called FRII sources (Fanaroff & Riley 1974). RAGNs of a hybrid FRI-FRII morphology are also found. There are also FR0 sources, which exhibit a bright core but lack clearly detectable extended emission.

Sometimes the jets and tails of the RAGNs (especially the FRI sources) are found to be bent, in which case they are called bent RAGNs, bent-tailed radio galaxies (BTRG; Lao et al. 2025), or simply tailed radio galaxies (Bhukta et al. 2022). Some examples are given in Appendix A. If the bending angle, or opening angle (BA / OA) of a bent RAGN is less than 90°, forming a shape similar to 'V', they are called narrow-angle-tail sources, NATs, or head-tail sources (HT; Rudnick & Owen 1976) because sometimes the two tails are so close that they create a single tail. And if the OA is more than 90°, forming a shape similar to 'C', they are called wide-angle-tail sources, WATs (Owen & Rudnick 1976; O'Dea & Baum 2023). An angle of exactly 90° would produce a shape similar to 'L' which could be considered a hybrid WAT-NAT source. Clear identification of a source as either WAT or NAT depends on various observational effects: projection effects related to the orientation of the object on the plane of our sky (Proctor 2011), and the sensitivity and resolution of our telescopes.

WATs show a sudden transition from jets to tails, possibly related to the transition from the interstellar medium (ISM) of a host galaxy to the intracluster medium (ICM) of a cluster (O'Dea & Baum 2023) or the intergalactic medium (IGM) of a dense environment. The reasons for this transition, the distortion of jets and the overall bending of tails provide motivations behind RAGN research: they can give insights into the nature of IGM in dense environments like groups and clusters of galaxies. Vardoulaki et al. (2025, Page 1) lists several reasons behind jet distortions discussed in the literature, including the path of jets in IGM (Miley et al. 1972), buoyancy forces, precession of jets, gravitational interaction with nearby galaxies, and the path of jets through a steep pressure gradient. Bending of tails due to ram pressure in IGM and ICM, have been studied particularly well. For example, the works of Garon et al. (2019), Mingo

[4] National Radio Astronomy Observatory

[5] According to `https://www.skao.int/news`.

[6] `https://pypi.org/project/rgc`.

Table 1: The surveys conducted by array radio telescopes (ART) that have found most of the bent RAGNs so far.

ART	Survey	Location	Sensitivity	Resolution	RAGNs	WATs	NATs	Bent RAGN reference
VLA	FIRST	USA	150	5″ (1400)	717	430	287	Sasmal et al. (2022a)
					4876	4424	652	Lao et al. (2025)
LOFAR	LOTSS	Netherlands	83	6″ (144)	35			Golden-Marx et al. (2021)
					55	45	10	Pal & Kumari (2023)
ATCA	ATLAS	Australia	20	12″ (1400)	45			Dehghan et al. (2014)
GMRT	TGSS	India	3500	25″ (150)	264	203	61	Bhukta et al. (2022)
MeerKAT	MIGHTEE	South Africa	6	5″ (1250)	359	-	-	Vardoulaki et al. (2025)

Notes. Sensitivity is given as 'median rms' in μJy beam^{-1}. Resolution is in arcsec, with the frequency given within brackets in MHz.

et al. (2019), and Golden-Marx et al. (2021) have shown that bent RAGNs are found predominantly in denser environments, if not in clusters, up to a redshift of 2.2. Moreover, more bent RAGNs are found in richer and more massive clusters; and the more bent the tails are the more they tend to be located near the centers of clusters (Vardoulaki et al. 2025, Page 2).

A substantial number of bent RAGNs have been found in the surveys conducted by different array radio telescopes (ART), for example, FIRST by VLA, LOTSS by LOFAR, ATLAS by Australia Telescope Compact Array (ATCA), the Tata Institute of Fundamental Research (TIFR) Giant Metrewave Radio Telescope (GMRT) Sky Survey (TGSS), and the MeerKAT International GHz Tiered Extragalactic Explorations (MIGHTEE). The number of bent RAGNs found in each case and the corresponding reference papers are given in Table 1, which clearly shows that FIRST has produced by far the highest number of these sources.

The bent RAGNs of FIRST have attracted a lot of investigations. An early example is Proctor (2011), where 7106 sources with four or more components were visually examined and manually annotated from more than 800,000 sources. The visual inspection was performed on 4′ cutouts, and around 400 sources were identified as either WAT or NAT, although some of them were uncertain. Garon et al. (2019) presented a sample of 4304 RAGNs from RGZ, and 87% of the sample were found to be within 50 Mly (mega-light-year) of an optically identified cluster. The brightest cluster galaxies (BCGs) were found to be more likely to harbor radio-emitting bent tails. However, they did not classify the sample into WATs and NATs.

Sasmal et al. (2022a) created the largest categorized sample of WATs and NATs up to that time. They used the Decemer 2014 data release of FIRST containing almost a million radio sources. In order to find extended RAGNs, they first filtered the sources with an angular size greater than 10″, at least twice the size of the synthesized beam of VLA. Around 95,000 automatically filtered sources were then visually examined, a remarkable achievement. During this inspection, 717 sources were categorized as either WAT or NAT based solely on the angle between the two tails. In order to determine the center of a RAGN from where the angle has to be calculated, they searched for the optical counterparts from SDSS data release 12. Optical counterparts were found for only half of the sources. For the other half, the center was guessed through 'eye estimation.' Both WAT and NAT have independent catalog numbering, each starting from 1. The scarcity of available information resulted in the inability to populate all data columns for a significant proportion of the sources.

An even bigger dataset (around 4800 sources) of WATs and NATs from the FRIST images was presented by Lao et al. (2025), and it is heavily skewed toward WATs: only 13% of the sources are NATs, as evident from our Table 1. They found optical counterparts for 86% of their sources using optical data from DESI (Dark Energy Spectroscopic Instrument) Legacy Surveys. Lao et al. (2025) cover a luminosity range of 8 orders of magnitude starting from 10^{20} W Hz^{-1}, up to a redshift of 3.43. Almost 37% of their RAGNs were found to be located in a cluster. Unlike the non-ML automated filtering method of Sasmal et al. (2022a), they have used a DL-based source finder called Radio Galaxies Classification with Mask Transfiner (RGCMT, their Section 2.2), that utilizes transformers. It was trained on a manually annotated dataset of around 3600 sources, including 400 bent RAGNs. Searching through almost a million FIRST sources, RGCMT could separate around 11,000 bent RAGN candidates, taking more than 17 hours with 3 GPUs (graphics processing units). These sources were visually inspected to produce the final catalog of 4876 bent RAGNs.

Unlike the supervised model of Lao et al. (2025) based on Mask R-CNN, our semi-supervised approach uses both unlabeled and labeled data. We perform the self-supervised pretraining on an unlabeled dataset of 20,000 sources taken from RGZ (described in the next section). And for supervised fine tuning, we have used the catalog of bent RAGNs made by Sasmal et al. (2022a). The number of labeled bent RAGNs in the training dataset of Lao et al. (2025) was around 400, less than the 717 labeled WATs and NATs of Sasmal et al. (2022a). Given the class imbalance due to scarcity of NAT sources, currently there are no semi-supervised binary classifier to categorize bent RAGNs into WATs and NATs. Our work fulfills this need. We also provide two batched datasets of bent RAGNs, labeled as WATs or NATs, ready for use in ML. Our processing and data preparation steps are described next.

3. Data and pre-processing

As mentioned before, both our labeled ($\mathbf{R}_L$) and unlabeled ($\mathbf{R}_U$) datasets are created using FIRST images.[7] Although the angular resolution of FIRST was 5 arcsec at 1400 MHz (Table 1), the final images have a pixel resolution of 1.8 arcsec. The images we use in training and testing have a dimension of 150 pixels on either side, giving an angular size of 4.5 arcmin.

3.1. Unlabeled data

We have used the batched dataset of 20,000 sources created for use in ML by Slijepcevic et al. (2022b) from RGZ. As described

[7] We use the symbol **R** because not only are these 'radio' images, but also the values of the pixels are 'real' numbers.

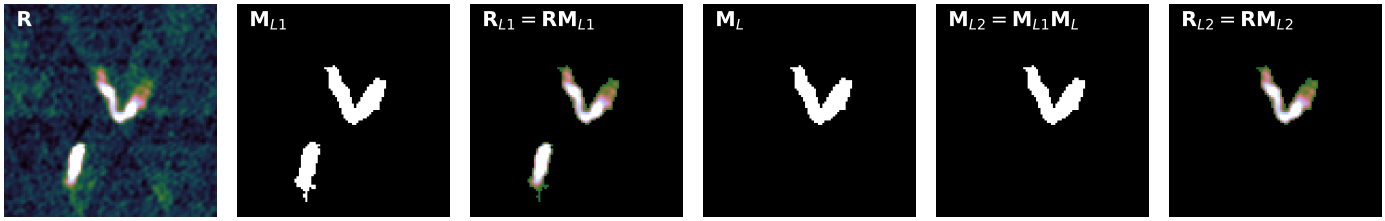


Fig. 1: An example of the process of masking an image as described in Section 3.2.2 and 3.2.3. $\mathbf{R}$ is the original image, $\mathbf{M}_{L1}$ its mask created using `PyBDSF` and $\mathbf{M}_L$ its mask produced trhough `Photutils`.

in their Section 3.2 and in Wong et al. (2025), RGZ data resease 1 (DR1) contains approximately 100,000 sources, 99.2% of which were taken from FIRST. For each FIRST object in this catalog with the largest angular size (θ_{LAS}) between 15 and 270 arcsec, Slijepcevic et al. (2022b) downloaded the corresponding image using the `Python API` of `SkyView`, a virtual observatory that provides access to a wide range of astronomical images from different surveys.[8] They cropped the inner 150^2 pixels from the original 300^2-pixel images, and put all pixels outside a radius of $0.6\theta_{LAS}$ to zero. After applying a $3-\sigma$ amplitude thresholding, each image $\mathbf{R}_o$ was normalized to

$$\mathbf{R}_n = 255 \times \frac{\mathbf{R}_o - min(\mathbf{R}_o)}{max(\mathbf{R}_o) - min(\mathbf{R}_o)} \quad (1)$$

in order to convert them to PNG (Portable Network Graphic) format, where $min(\mathbf{R}_o)$ is the minimum value of $\mathbf{R}_o$, and $max(\mathbf{R}_o)$ is the maximum. Thereafter all unresolved sources were removed, and the dataset was further reduced by discarding sources that were too dissimilar to their labeled dataset of FRI and FRII sources, the MiraBest dataset (Porter & Scaife 2023) which was also used in H23. The similarity between RGZ DR1 and MiraBest was analyzed using Fréchet inception distance (FID), as described in their Section 4.3. Most importantly, the labeled sources of MiraBest that were present in RGZ were also removed, so that there is no overlap between the labeled and unlabeled data. The final unlabeled dataset contained 20,000 sources organized in 10 batches of training data, and one batch of test data. We downloaded these from their GitHub repository[9] and used them without any further processing. Hereafter, this batched dataset will be referred to as $\mathbf{R}_U$.

Slijepcevic et al. (2024) uses a larger unlabeled dataset of 108,000 sources from RGZ for self-supervised learning, but pre-training on such a large dataset is beyond the scope of our present work.

3.2. Labeled data

Unlike $\mathbf{R}_U$, our labeled dataset $\mathbf{R}_L$ has been pre-processed for efficient use in ML in this work for the first time. As mentioned in Section 2, this dataset is created using the catalog of more than 700 WATs and NATs published by Sasmal et al. (2022b) (hereafter, the Sasmal catalog). However, it was created through visual inspection and, hence, there was no batched dataset ready for use in ML based on the catalog. We describe below how we refined the Sasmal catalog and prepared an ML-friendly batched dataset.

3.2.1. Downloading

We accessed the Sasmal catalog from `VizieR`[10] using the `astropy` module `astroquery`. It contains two tables: Table 1 lists 430 WATs, while Table 2 lists 287 NATs. The different columns of the table have been described above in Section 2. The catalog was downloaded in `VOTable` format and converted to a `Pandas DataFrame` for further processing. To facilitate automated catalog retrieval, we developed the `catalog_quest` function within our `RGC` package, which directly queries VizieR and returns the data in a structured `DataFrame` format.

The coordinates of the sources were used to download the corresponding FIRST images through the `Python API` of `SkyView` within `astroquery`.[11] The images were retrieved in `FITS` (Flexible Image Transport System) format. To facilitate the automated download of images, we have developed the function `celestial_capture` within `RGC`. This function takes the survey name (e. g., `FIRST`), source coordinates and output file path as input, and downloads the images of the sources to the specified file path. Additionally, we provide the function `celestial_capture_bulk`, which enables batch downloading of all sources in a given catalog. This function takes the catalog `DataFrame`, survey name, and directory path as input and downloads the sources in the catalog to the specified directory.

Among the 717 bent RAGNs of the Sasmal catalog, we could successfully retrieve 703. The remaining 14 sources could not be retrieved due to either server errors during the download, or the downloaded image being blank. Among the 703 images, 281 were NATs and 422 were WATs. We pre-processed these images and created a refined annotated catalog for efficient use in ML which will be called Sasmal-ML hereafter. The pre-processing were performed using two separate software: `PyBDSF` (Mohan & Rafferty 2015) and `Photutils` (Bradley et al. 2025). The labeled dataset $\mathbf{R}_L$ created through `PyBDSF` processing will be called $\mathbf{R}_{L1}$ and the one created through `Photutils` processing will be called $\mathbf{R}_{L2}$.

Because of the way we downloaded the images, the target is always located at the center of a 150^2-pixel image. During the first step of pre-processing, we created a mask that can be applied to the image to remove everything except the central target. This was performed in two steps: first, a mask was generated using `PyBDSF` for $\mathbf{R}_{L1}$ and, then, the mask was refined using `Photutils` for $\mathbf{R}_{L2}$. The two steps are described in the subsequent two subsections.

3.2.2. Generating masks using `PyBDSF`

We start by estimating the background noise level for each image. `PyBDSF` achieves this by segmenting the image into smaller

[8] Developed at the auspices of the High Energy Astrophysics Science Archive Research Center (HEASARC) at the Goddard Space Flight Center (GSFC) of NASA; `https://skyview.gsfc.nasa.gov`.
[9] `https://github.com/inigoval/fixmatch`
[10] `https://vizier.cds.unistra.fr/viz-bin/VizieR`
[11] `astroquery.skyview.SkyView`

subregions and computing the local mean ($\mu_{\rm local}$) and standard deviation ($\sigma_{\rm local}$) of the pixel intensities within each subregion. As given in the documentation of PyBDSF,[12] the background intensity of an original image $\mathbf{R}$ at a pixel (x, y)

$$B(x, y) = \mu_{\rm local} + k \cdot \sigma_{\rm local} \tag{2}$$

where k is a scaling factor accounting for typical noise fluctuations. After background estimation, sources are detected by identifying contiguous regions called 'islands' where the intensity exceeds the local background by a defined threshold. Two types of thresholds are applied. The 'island threshold' $T_{\rm isl}$ is the minimum signal level for considering a group of contiguous pixels as part of an island. A pixel is included in an island if its intensity

$$R(x, y) > B(x, y) + T_{\rm isl} \cdot \sigma_{\rm local} \tag{3}$$

where we used $T_{\rm isl} = 3$ for all cases. The 'peak threshold' $T_{\rm pix}$ is the peak intensity of an island, and only islands with peaks exceeding

$$R(x, y) > B(x, y) + T_{\rm pix} \cdot \sigma_{\rm local} \tag{4}$$

are considered valid detections. We used $T_{\rm pix} = 5$ for all the cases. Once the thresholds are applied, contiguous pixels meeting the criteria are grouped into islands. For each island, a two-dimensional Gaussian model is fitted to extract key parameters, including the centroid (x_0, y_0), peak intensity R_0, and morphological descriptors, for example, major axis $\sigma_{\rm maj}$, minor axis $\sigma_{\rm min}$, and position angle θ. The Gaussian model is

$$R(x, y) = R_0 \exp\left(-\frac{(x' - x_0)^2}{2\sigma_{\rm maj}^2} - \frac{(y' - y_0)^2}{2\sigma_{\rm min}^2}\right), \tag{5}$$

where (x', y') are the coordinates rotated by the angle θ. Once sources are identified, a binary mask is generated to delineate the spatial extent of each source. The mask $\mathbf{M}_{L1}$ is created by thresholding the fitted Gaussian model as

$$M_{L1}(x, y) = \begin{cases} 1, & \text{if } R(x, y) > B(x, y) + T_{\rm pix} \cdot \sigma_{\rm local}, \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

as shown in the Fig. 1. Because our sources were extended, the masks underwent a dilation by d pixels to ensure complete coverage of the source. The final binary masks are stored in FITS format for further analysis. These steps were performed using the `process_image` function of PyBDSF that takes the beamshape, frequency, and the aforementioned threshold parameters as inputs, and provides a PyBDSF object as output containing the detected sources. Using the `export_fits` method, we export the mask of the sources in FITS format, applying a dilation of $d = 0$ for most sources, with different values used for a few cases where adjustments were needed. The images for which different dilations had to be applied are identified in Sasmal-ML, the catalog published with this paper.

For ease of use, we provide the function `generate_mask` in RGC, which automates this process. It takes the image file name, beamshape, frequency, threshold values, dilation, and output file name as inputs, and generates the corresponding mask as output. Additionally, we offer `generate_mask_bulk` to process an entire catalog of images.

[12] https://pybdsf.readthedocs.io/en/latest/

Table 2: Distribution of WATs and NATs in our labeled dataset $\mathbf{R}_{L1}$ and $\mathbf{R}_{L2}$, created following the Sasmal-ML catalog demonstrated in Table A.1.

Type	Batch	WATs	NATs	Batch total	Type total
Train	0	39	25	64	576
	1	39	25	64	
	2	39	25	64	
	3	39	25	64	
	4	39	25	64	
	5	38	26	64	
	6	38	26	64	
	7	38	26	64	
	8	38	26	64	
Test	9	38	25	63	63
Total	10	385	254	639	639

3.2.3. Refining Masks Using `Photutils`

In order to improve the isolation of the sources, we applied another, independent masking based on `Photutils` pruning spurious sources. This refinement proceeds in two steps: first, generating a mask with `Photutils` from each FITS image and, second, taking the product of the PyBDSF and `Photutils` masks to make a conservative final mask.

For creating the `Photutils` mask $\mathbf{M}_L$, we first replace the Nan values in the original image with zeros. Next, the global background properties of the images are estimated using sigma-clipped statistics. The median m, mean μ, and standard deviation σ of an image are calculated by excluding all pixels that deviate more than 3σ from the mean. A global detection threshold is then defined as

$$T = m + k'\sigma \tag{7}$$

where $k' = 2.5$ for most images. Source detection is performed using the `detect_sources` function of `Photutils` which identifies sources in an image by identifying groups of connected pixels above a specified threshold. For each pixel $R'(x, y)$ in the image, we compare its value to the detection threshold T to get

$$S(x, y) = \begin{cases} 1, & R'(x, y) > T \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Pixels with $S(x, y) = 1$ are grouped into contiguous regions (called groups) using a connectivity criterion. If a group contains more than 10 pixels, it is given an integer label l_i, and all other pixels are again set to zero. The new image created by the indexed sources is named $\mathbf{L}$. In order to identify the target object, the center of the image is calculated and the group l_i at that point is examined. If no group is present, the search is widened to a square aperture of width $2r_{\rm search}$ centered at (x_c, y_c), with $r_{\rm search} = 25$ pixels, and the group with the highest number of nonzero pixels is selected. In order to capture the surrounding regions, we then search a radius $r_{\rm extend} = 20$ pixels and take all the unique groups therein.

The mask $\mathbf{M}_L$ is finally constructed as a binary image where all pixels that match any of these collected labeled groups are included as

$$M_L(x, y) = \begin{cases} 1 & \mathbf{L}(x, y) \in \{\ell_1, \ldots, \ell_M\} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

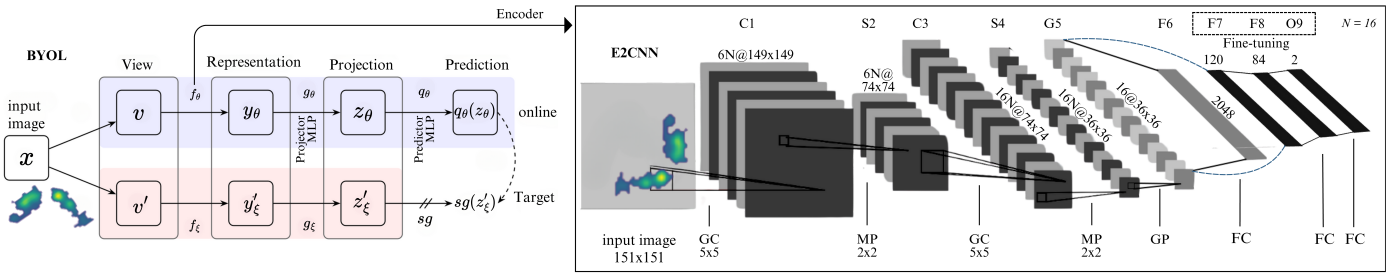


Fig. 2: *Left*: The architecture of BYOL used in our self-supervised pre-training, adapted from Grill et al. (2020). The encoder within BYOL in our case is E2CNN which has been used for our supervised downstream fine-tuning. *Right*: The architecture of E2CNN as used by Scaife & Porter (2021), adapted from the LeNet-5 architecture of Lecun et al. (1998). The operations GC, MP, GP anf FC stand for group convolution, max-pooling, group-pooling and fully-connected, respectively.

where $\{\ell_1, \ldots, \ell_M\}$ is the set of extended group labels. In order to ensure that the mask completely covers the source and marginally overestimates its extent, a morphological dilation d' is applied. The dilated binary mask $\mathbf{M}_L$ is saved as a FITS file. This is multiplied with the PyBDSF mask $\mathbf{M}_{L1}$ to create the final combined mask $\mathbf{M}_{L2}$ as shown in Fig. 1.

3.2.4. Conversion to PNG

The downloaded images and the generated masks were in FITS format, which is not suitable for efficient use in ML. Therefore, we converted the images and masks to PNG, which is widely used for image processing. For this conversion, the images were first normalized following Equation 1. The normalized images and masks were then converted to PNG using `Pillow`[13]. In Fig. 1, the leftmost panel shows the downloaded raw image in PNG format, and the masks are also shown as PNGs in the next three panels.

`RGC` has the function `fits_to_png` to convert the FITS files to PNG format automatically. It takes the FITS file name as input and converts the image to PNG format and returns an image in the form of a `PIL.Image` object. We also provide a function `fits_to_png_bulk` to convert all FITS files in a given directory to PNG format. The function takes the directory path and output directory path as input and converts all FITS files in the directory to PNG format and saves them in the output directory.

3.2.5. Applying the masks

Background removal from the images is achieved by applying the binary masks (values either 0 or 1) generated in the previous steps. By multiplying the original image $\mathbf{R}$ with its corresponding binary mask $\mathbf{M}_{L2}$, the background is suppressed, retaining only the source emissions, i. e. the masked image

$$\mathbf{R}_{L2} = \mathbf{R}\mathbf{M}_{L2} \tag{10}$$

which is shown in the rightmost panel of Fig. 1 in PNG format. `RGC` provides two functions named `mask_image` and `mask_image_bulk` in order to automate this process. The `mask_image` function takes `PIL.Image` objects of the image and the mask as inputs, applies the mask on the image, and returns the masked image as a `PIL.Image` object. The `mask_image_bulk` function processes an entire dataset by taking the directory paths of images and masks as input, applying the masks to all images, and saving the masked images in the specified output directory.

[13] `https://python-pillow.org/`

3.2.6. Sasmal-ML: a bent RAGN catalog and dataset for ML

The 703 images, masked, processed and converted following the aforementioned steps, were visually examined by three authors. The complete annotations are provided in our Sasmal-ML catalog provided in the Github repository of RGC as a `TSV` (tab separated values) file. Sixty-three rows of these catalog are shown in Appendix A.

Sasmal-ML contains 703 rows and 16 columns, as described in Table A.1 of the appendix. The original Sasmal catalog contained 717 sources, but 14 of its sources could not be downloaded. Among the 703 sources of Sasmal-ML, 639 were finally selected for our datasets $\mathbf{R}_{L1}$ and $\mathbf{R}_{L2}$. The reasons behind discarding 64 sources can be found in Sasmal-ML. From the Hossain column of Sasmal-ML, one can see that 31 sources were discarded because of their small (compact) size, 9 were discarded because of the confusion about their class (WAT or NAT), and 24 sources were discarded because of a combination of reasons, for example, low signal-to-noise ratio, low angular extent, and faint extended emission. The remaining 639 sources were given the label 'SMD' in this column. The last two columns are related to the attention of our DL model trained and tested on the data, to be discussed in Section 6.

These 639 sources were used to create our batched dataset for ML. It contains 385 WATs and 254 NATs distributed in 9 training batches, and 1 test batch, as shown in Table 2. Each training batch contains around 39 WATs and 25 NATs, and the test batch contains 38 WATs and 25 NATs. The overall class imbalance is approximately 66%. The two datasets $\mathbf{R}_{L1}$ and $\mathbf{R}_{L2}$ are given as a single `TAR.GZ` (tape archive, GNU zip) file in the Github repository of RGC. The file contains all the PNG images of our labeled dataset $\mathbf{R}_L$ preprocessed in two different ways.

4. The semi-supervised model of RGC

RGC combines different existing DL architectures to create a new framework for classifying RAGNs. It was first described by Hossain et al. (2023), or H23, where semi-supervised learning (SSL) methods were used to classify RAGNs into FRI and FRII types (see their Sections 3 and 4 for a detailed background). Here we use a refined version of RGC for classifying RAGNs into WATs and NATs and publish its Version 1.0 as a python package for the first time.

Slijepcevic et al. (2022a,b) showed early examples of using SSL for FR classification. They used the contrastive learning method SimCLR (Chen et al. 2020), the self-supervised method BYOL (Grill et al. 2020), and the semi-supervised method FixMatch (Sohn et al. 2020). H23 used SimCLR and BYOL in com-

Table 3: Comparison of the hyperparameters used in our self-supervised pre-training and supervised fine-tuning.

Hyperparameters	BYOL Pre-training	E2CNN Fine-tuning
Optimizer	Adam	Adam
Learning Rate	3×10^{-4} (constant)	3×10^{-4} (initial)
Batch size	16	16
Weight decay		10^{-6}
Scheduler		ReduceLROnPlateau
Epochs	500 (no early stopping)	500 (early stopping)

bination with an $E(2)$-steerable CNN (E2CNN; Weiler & Cesa 2019; Scaife & Porter 2021), and found BYOL to outperform SimCLR. Therefore, in this work, we include only BYOL with E2CNN in RGC 1.0. Both BYOL and E2CNN have been described sufficiently in the relevant papers, and in H23. Hence, below we focus mainly on the experimental setup of RGC 1.0 relevant for this work.

As described in Section 5 of H23, RGC has two stages: an unsupervised (or, equivalently, self-supervised) feature extraction, and a supervised fine-tuning. In the first stage, we pre-train RGC using BYOL on the unlabeled dataset $\mathbf{R}_U$. Here BYOL contains a G-CNN as an encoder, which preserves invariance under different transformations. In the second stage, we fine-tune the pre-trained model on the labeled dataset $\mathbf{R}_{L1}$ and $\mathbf{R}_{L2}$ using supervised learning. So the first stage is task-agnostic, and the second stage task-specific where classification is performed as a downstream task. In Fig. 2, we show the first stage on the left, and the second stage on the right.

4.1. Self-supervised pre-training

The first stage of RGC is depicted on the left panel of Fig. 2. It shows how BYOL learns a representation y_θ that can be used for downstream tasks. Its online (top row) and target (bottom row) networks are similar, but asymmetric, and they have different weights θ and ξ, respectively. The online network has three stages: an encoder f_θ, a projector g_θ and a predictor q_θ. In our case E2CNN was used as the encoder f_θ. BYOL uses its target network as regression 'targets' for training the online network. The process is shown from the left to the right of this panel. Given an image x, BYOL produces its two views v and v' by applying different augmentations t and t'. From v, the online network creates a representation y_θ and a projection z_θ; the target network produces the corresponding y'_ξ and z'_ξ from v'. Then the online network produces a prediction $q_\theta(z_\theta)$. Subsequently l_2-normalization is applied to the online prediction and the target projection. The difference between these two l_2-normalized vectors is the loss function of BYOL, defined as (Equation 2 of Grill et al. 2020)

$$\mathcal{L}_{\mathrm{BYOL}} = \|\overline{q_\theta}(z_\theta) - \overline{z}'_\xi\|_2^2 \tag{11}$$

which is a mean-squared error (MSE). At each training step, a stochastic optimization is performed to minimize the loss with respect to θ, but not ξ, which is shown by the stop-gradient (`sg`) in the left panel of Fig. 2.

The architecture of the encoder f_θ is based on E2CNN, which is designed to preserve invariance under transformations such as rotation and reflection, the Euclidean group $E(2)$. As shown on the right panel of Fig. 2, the encoder has 6 layers: the convolution layers C1 and C3, the subsampling layers S2 and S4, and the group-pooling layer G5. It follows the architectures of LeNet-5 (Lecun et al. 1998) and AlexNet (Krizhevsky et al. 2012), modified by Scaife & Porter (2021). The input image has dimensions of 151×151 because of zero-padding. After a group convolution on the images with kernel size 5×5, we get $6N$ channels with dimensions 149×149, where $N = 16$ is the number of equivariant groups. After a max-ppoling operation with a 2×2 kernel, the dimensions reduce to 74×74. Another group convolution and max-pooling reduce the dimensions to 36×36 and increase the number of channels to $16N$. Finally, a group-pooling operation combines the equivariant groups to create only 16 channels, which are flattened and passed through a fully-connected layer to produce a vector of dimension 2048.

4.2. Supervised fine-tuning

The second stage of the RGC model involves fine-tuning the pre-trained encoder on a labeled dataset of bent radio AGNs to classify them into WATs and NATs. The model is trained to minimize the cross-entropy loss between the predicted and true labels of the input data, defined as

$$\mathcal{L}_{\mathrm{E2CNN}} = -\sum_{i=1}^{N_c} y_i \log(p_i), \tag{12}$$

where y_i is the true label of the input data, p_i is the predicted probability of the input data belonging to the true class, and N_c is the number of classes. The model is trained to minimize this loss by updating the parameters of the encoder using backpropagation. As shown on the right panel of Fig. 2, the fine-tuning stage takes the feature maps of G5 as input and passes them through two FC layers to produce two vectors of dimensions 120 and 84, consecutively. The output layer O9, fully connected with F8, finally gives us two logits which are used to calculate the class probabilities through the softmax function.

4.3. Training details

In Table 3, the hyperparameters used in our pre-training and fine-tuning are given. Both were performed in `Timaeus`,[14] a high-performance workstation containing a NVIDIA Leadtek QuadroRTX A4000 GPU with 16-GB memory. During pre-training, we have used the Adam optimizer with no weight decay. The learning rate was kept constant throughout the 500 epochs. Because there were no labeled data in this stage, we continued the epochs without any early stopping until convergence of the loss.

During the supervised fine-tuning, we again used Adam optimizer, but this time with a weight decay, given in Table 3.

[14] `Timaeus` belongs to the Center for Astronomy, Space Science and Astrophysics (CASSA) and it is located in the Data Center of Independent University, Bangladesh.

Model	Accuracy [%]	WAT			NAT		
		Precision	Recall	F1-score	Precision	Recall	F1-score
RGC ($\mathbf{R}_{L2}$)	88.9	0.8974	0.9211	0.9091	0.8750	0.8400	0.8571
RGC ($\mathbf{R}_{L1}$)	87.3	0.8750	0.9211	0.8974	0.8696	0.8000	0.8333
ConvNeXT ($\mathbf{R}_{L1}$)	84.12	0.85	0.89	0.87	0.83	0.76	0.79
SWIN-B ($\mathbf{R}_{L1}$)	80.95	0.80	0.92	0.85	0.84	0.64	0.73
ResNet-50 ($\mathbf{R}_{L1}$)	77.77	0.79	0.87	0.82	0.76	0.64	0.70
ViT-B-16 ($\mathbf{R}_{L1}$)	76.19	0.76	0.89	0.82	0.77	0.73	0.74
VGG-16 ($\mathbf{R}_{L1}$)	74.60	0.73	0.92	0.81	0.80	0.48	0.60

Table 4: Performance metrics of RGC on two datasets, with ($\mathbf{R}_{L1}$) and without ($\mathbf{R}_{L2}$) spurious sources, and the performance metrics of five more fully supervised models trained and tested on $\mathbf{R}_{L1}$ (Hossain et al. 2025). Details can be found in Section 5.2 and 5.3.

The learning rate was the same as pre-training initially, but it was changed through a 'ReduceLROnPlateau' scheduler with a 'factor' of 0.9 and a 'patience' of 2. The scheduler waits for 2 epochs to evaluate the changes and then reduce the learning rate by 10%. Because we have labeled data in this stage, we implemented 'early stopping' based on the validation loss. TensorBoard tracks extensive measures such as precision, recall, and F1 score to monitor performance and reduce overfitting. Model checkpoints are saved on the basis of validation correctness.

4.4. Class activation mapping

In order to determine the significance of each spatial position on an image for a given class prediction, we used Grad-CAM (Selvaraju et al. 2017) where the gradients of any target flowing into the final convolutional layer are utilized to produce a coarse localization map L^c highlighting the important regions in the image. For an input image x and target class c, the process starts with a forward pass through the network to obtain the logit y^c and then calculate the gradient $\partial y^c / \partial A^k_{ij}$ where A^k_{ij} is the activation of a specific position (i, j) in a feature map k. We calculated the 'importance weight' for each feature map as

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}} \tag{13}$$

where Z is the total number of elements in the feature map. This parameter shows how much the feature map k contributes to a class c. The feature maps are then multiplied with their corresponding importance weights and summed to produce

$$L^c = \mathrm{ReLU}\left(\sum_k \alpha^c_k A^k \right) \tag{14}$$

which is used to produce a single 2D heatmap $\mathbf{R}_A$ that highlights the regions of an image where the model paid the most attention. The heatmaps created from the dataset with spurious sources ($\mathbf{R}_{L1}$) is named $\mathbf{L}_{L1}$ and from the one without spurious sources ($\mathbf{R}_{L2}$ is named $\mathbf{L}_{L2}$, as shown in Appendix B.

5. Performance of the RGC model

In this section, we evaluate the performance of the RGC model on the classification of bent RAGNs into WATs and NATs using two different datasets $\mathbf{R}_{L1}$ and $\mathbf{R}_{L2}$. A total of 63 images were used for this testing as shown in Table 2. The metrics used are confusion matrix, accuracy, precision, recall and F1-score.

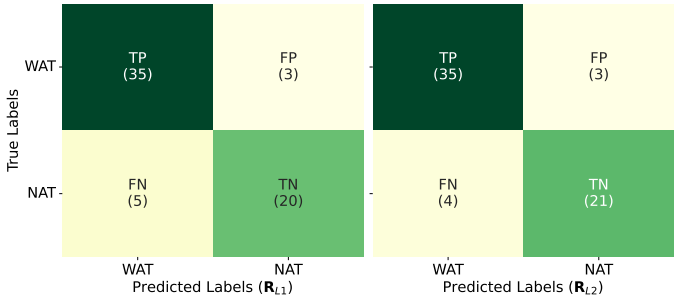


Fig. 3: Confusion matrices for the classification of bent RAGNs into WATs and NATs, showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for the RGC model trained on the datasets $\mathbf{R}_{L1}$ (left) and $\mathbf{R}_{L2}$ (right).

In addition, we have also used the Receiver Operating Characteristic (ROC) curve, Area Under the Curve (AUC) score, and Expected Calibration Error (ECE) to evaluate the discriminative ability and reliability of our model, as described below.

5.1. Confusion matrix

The confusion matrix shows four values: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The TP indicates the number of positive samples (WATs) that were correctly classified as positive, while TN indicates the number of negative samples (NATs) that were correctly classified as negative. FP indicates the number of negative samples that were incorrectly classified as positive, while FN indicates the number of positive samples that were incorrectly classified as negative. The confusion matrices created by testing RGC on the aforementioned datasets $\mathbf{R}_{L1}$ (left) and $\mathbf{R}_{L2}$ (right) shown in Fig. 3.

The confusion matrix shows that, without spurious sources ($\mathbf{R}_{L2}$), the model correctly classified 35 out of 38 WATs as WAT and 21 out of 25 NATs as NAT. Misclassifications include 3 WATs labeled NAT and 4 NATs labeled WAT. When spurious sources ($\mathbf{R}_{L1}$) are included, the model maintains the same performance for WATs, while identifying 20 of the 25 NATs correctly. In this case, 3 WATs are misclassified as NAT and 5 NATs as WAT. RGC performs slightly better if trained and tested on the dataset without spurious sources, but the difference is negligible in the confusion matrix.

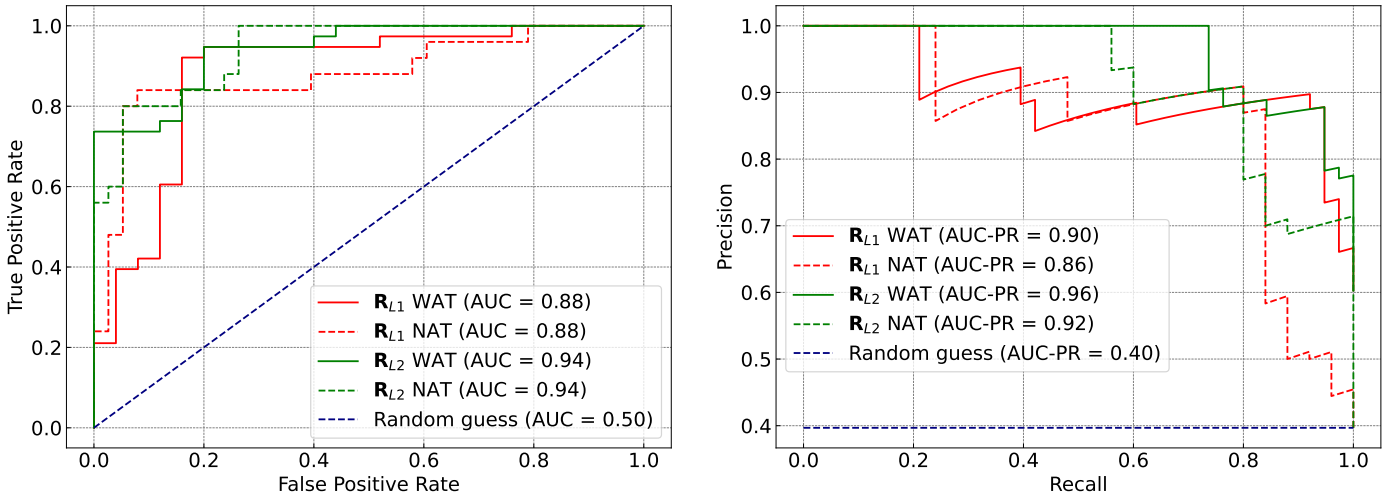


Fig. 4: *Left*: ROC curves and AUC scores for the RGC model trained and tested on two datasets $\mathbf{R}_{L1}$ (red) and $\mathbf{R}_{L2}$ (green), for both WATs (solid) and NATs (dashed), in comparison to the random guess (blue). *Right*: Precision-Recall (PR) curves and the corresponding AUC-PR scores for the same datasets and sources.

5.2. Classification metrics

Based on the output shown in the confusion matrix, we calculated the accuracy, precision, recall, and F1-score. The accuracy of a model is the proportion of correctly classified samples to the total number of samples. The precision of the model is the proportion of TPs to the sum of TPs and FPs: $TP/(TP + FP)$. It measures how many of the samples classified as positive are actually positive. The recall of the model is the proportion of TPs to the sum of TPs and FNs: $TP/(TP + FN)$. It measures how many of the actual positive samples are correctly classified as positive. The F1-score is the harmonic mean of precision and recall; it is a measure of the balance between precision and recall.

The values of these metrics for RGC are given in Table 4 for both datasets. As evident from the first two rows of the table, our RGC model performs better on the dataset without spurious sources, reaching an accuracy of 88.9%. Comparing the precision, recall and F1-score of WATs and NATs for both datasets, we can see that the metrics are better for WATs because these sources were more numerous.

5.3. Comparison with supervised models

Table 4 presents a comprehensive comparison of the performance of different *fully*-supervised baseline models along with our *semi*-supervised RGC model. Among the supervised models, ConvNeXT demonstrates the best overall performance with an accuracy of 84.12%. It achieves good results in both WAT (F1-score of 0.87) and NAT (F1-score of 0.79), indicating relatively balanced performance. SWIN-B also performs competitively, achieving 80.95% accuracy with a relatively high WAT recall (0.92), although its NAT recall is lower (0.64), showing that it tends to favor WAT predictions.

ResNet-50 and ViT-B-16 provide moderate performance, with accuracy values of 77.77% and 76.19%, respectively. While ViT-B-16 achieves a balanced trade-off between WAT and NAT metrics (F1-scores of 0.82 and 0.74), ResNet-50 shows a slightly better WAT recall but lower NAT precision. In contrast, VGG-16 lags behind the other baselines, obtaining the lowest overall accuracy (74.60%) and particularly poor NAT recall (0.48), which limits its effectiveness in distinguishing NAT samples. More details on these baseline benchmarking can be found in Hossain et al. (2025).

The semi-supervised RGC model outperforms all fully-supervised baselines. RGC trained without spurious sources (on $\mathbf{R}_{L2}$) achieves the highest overall accuracy and superior F1-scores for both WAT. RGC trained with spurious sources also performs strongly (87.30% accuracy), but its NAT recall (0.80) and F1-score (0.83) are slightly lower than the spurious-free version, demonstrating that removing spurious features improves generalization and class balance.

5.4. Discriminative ability

We further evaluated the discriminative ability of our model using ROC curves and the corresponding AUC scores. The ROC curve is a graphical representation of the performance of a binary classifier system. It plots the True Positive Rate (TPR, same as recall) against the False Positive Rate, FPR or $FP/(FP + TN)$, at various threshold settings, i. e., different probability values for classifying a sample as positive. The ROC curve shows the trade-off between correctly catching positive cases and mistakenly identifying negative cases across all classification thresholds. While it is a useful tool for visualizing the performance of a classifier, the AUC score provides a single numeric value to represent the classifier's performance. It ranges from 0 to 1, with a score of 1 indicating a perfect classifier and a score of 0.5 indicating a classifier that performs no better than a random guess.

ROC and its AUC can sometimes be misleading, especially when the dataset is imbalanced which is the case here. To address this, we have used the Precision-Recall (PR) curve and the corresponding AUC-PR score. The PR curve plots the precision against the recall at various threshold settings. It is useful for imbalanced data because it focuses on the positive class (WATs). AUC-PR provides a single numeric value to represent the performance ranging from 0 to 1, with 1 indicating a perfect classifier. A higher AUC-PR indicates better overall performance.

In Fig. 4, we show the ROC curves with their AUC (left panel) and the PR curves with their AUC-PR (right panel) for WATs and NATs separately in case of the RGC model trained and tested on our two datasets. The ROC curves are way above

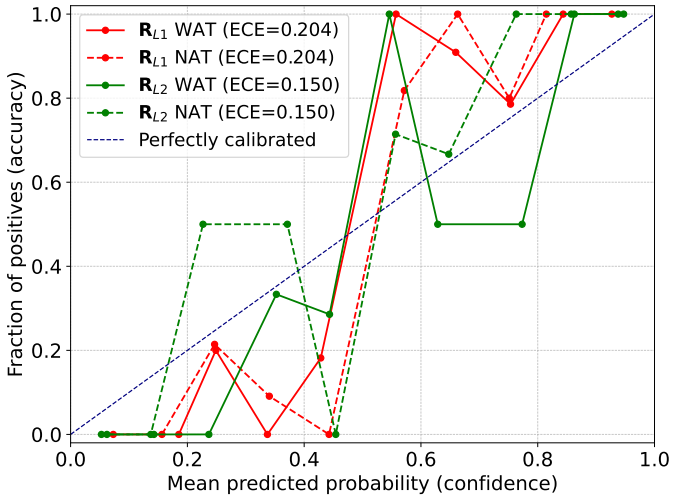


Fig. 5: ECE curves (described in Section 5.5) for our RGC model trained and tested on two datasets for classifying WATs and NATs. The colors and linestyles are the same as Fig. 4.

the random guess line, and close to the top-left corner of the plot indicating that our model has a high discriminative ability. The AUC score further confirms this: 0.88 with spurious sources, but 0.94 without ($\mathbf{R}_{L2}$). The PR curves for both WATs and NATs are high above the horizontal line at 0.40, which indicates that the model is performing much better than a random classifier. The AUC-PR score distinguishes the two classes better than AUC. While the AUC score was same for WATs and NATs, the AUC-PR for $\mathbf{R}_{L2}$ is 0.96 for WATs and 0.92 for NATs. This indicates that the model has good discriminative ability for both classes, despite the class imbalance in the data set.

5.5. Model calibration

Model calibration aims to align the predicted probabilities of a model with the true probabilities of the data to ensure reliability. We have used the Expected Calibration Error (ECE) metric to evaluate the calibration. It is calculated by dividing the samples into some bins of equal size as described in Guo et al. (2017). The ECE is then calculated as an weighted average of the differences between the accuracy and confidence of predictions within the bins (their Equation 3 and the definitions therein). The ECE curve is a plot of the accuracies (fraction of positives) as a function of the confidences (mean predicted probability) for all bins as shown in Figure 1 of Guo et al. (2017). For a perfectly calibrated model (defined in their Equation 1), the ECE curve would be a perfect diagonal.

The ECE curve of our RGC model is shown in Fig. 5 for both datasets and sources (WATs and NATs) using the same color scheme as in Fig. 4. They show that the model is well calibrated because the curves lie close to the diagonal. ECE is approximately 0.150 for both WATs and NATs without spurious sources ($\mathbf{R}_{L2}$), and increases to 0.204 for both classes with spurious sources.

6. Discussion

The key results from Section 5 will be summarized in Section 7. Here we discuss the transparency and limitations of the RGC model presented in Section 4.

6.1. Attentions to WATs and NATs

In order to make the RGC model transparent, we created localization maps for all 639 sources, using Grad-CAM following the procedure described in Section 4.4, for both our datasets. The localization maps produced from the dataset with spurious sources ($\mathbf{R}_{L1}$) is named $\mathbf{L}_{L1}$, and the maps prodcued from the dataset without spurious sources ($\mathbf{R}_{L2}$) is named $\mathbf{L}_{L2}$. These maps show the regions of an image where the model is paying the most 'attention' in order to make a prediction about a class. Sixty-three of these localization maps are given in Appendix B as contours overplotted on the masked images of RAGNs. A qualitative evaluation of the attentions of our model is given below using these plots.

For both our datasets, the attention is better for WATs than NATs because WATs were more numerous. For WATs, the attention contours are almost always centered on the source peaks and continue until the edges. The outermost contour usually lies outside the source and in many cases multiple components of a source are bound within a single such contour. NATs, however, exhibit more complicated attention behavior. For most of these sources, the attention contours do not surround the source but delimit only one side of the source. Even when attention is given on multiple sides of a source, the contours only touch the edges of the source far from the peaks. For a few NATs, the contours do not even touch the source, but trace the general shape of the source at a distance.

The part of the Sasmal-ML catalog given in Appendix A also gives specific comments about the quality of attention for the two datasets in the last two columns. Among the 63 test-batch images inspected visually in dataset $\mathbf{R}_{L1}$, 50% of the attentions in WATs were qualitatively good (marked 'g'), whereas 44% of the attentions in NATs were good. In the dataset without spurious sources, this increased substantially to 94% for WATs and 68% for NATs.

6.2. Spurious sources

The table in Appendix A and the figures in Appendix B give us a very good idea about the impact of removing spurious sources from an image. From the second-last column of the table we can see that 26 out of 63 (around 40%) of the images in $\mathbf{R}_{L1}$ had spurious sources aside from the central target source. All these spurious sources were successfully removed in $\mathbf{R}_{L2}$ (following the procedure given in Section 3.2.3) except just one image where the target source itself was partially cropped during the masking process. From an inspection of the $\mathbf{L}_{L1}$ contours plotted over $\mathbf{R}_{L1}$, it becomes obvious that the model was giving attention to the spurious sources, sometimes even more than the target sources depending on their relative brightness. However, the $\mathbf{L}_{L2}$ contours are clearly always near the target sources as there are no spurious sources in $\mathbf{R}_{L2}$.

Although this improvement in attention after removing the spurious sources is obvious in a visual inspection, it is not so obvious in Fig. 6 where the distance r_{il} between the peak of an image and the peak of a Grad-CAM localization map is plotted for all sources as a cumulative distribution function (CDF). The red and green lines represent the trends for the data with and without spurious sources, respectively, and their difference at almost all cumulative counts is no more than a few pixels, or arcseconds. However, r_{il} for the dataset $\mathbf{R}_{L2}$ is greater than that of $\mathbf{R}_{L1}$ at almost all cumulative counts. The reason behind this could be, again, the spurious sources: for almost half of the im-

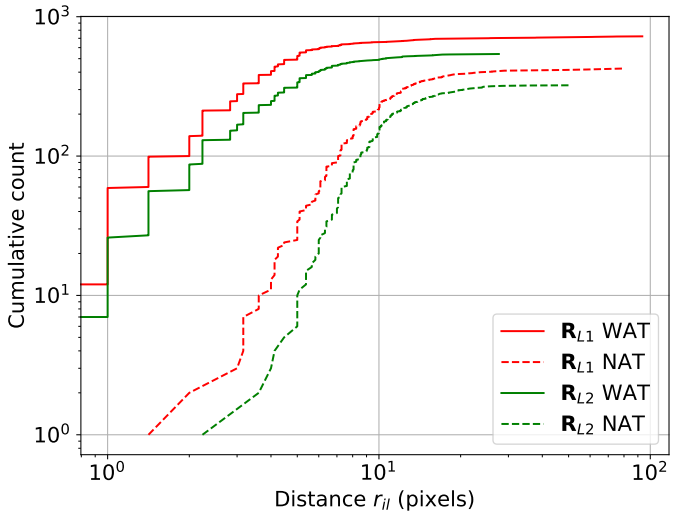


Fig. 6: The distance r_{il} between the peak of an image and the corresponding peak in its Grad-CAM localization map for both datasets and sources, following the same color and linestyle as in Fig. 4 and 5.

ages in the first dataset, the target peaks were being confused with the spurious peaks.

The effects of spurious sources are also apparent in the figures of the performance metrics. The confusion matrix did not show a substantial difference between the two datasets; $\mathbf{R}_{L1}$ had one more false negative compared to $\mathbf{R}_{L2}$. But the difference between the red and green curves is clear in both panels of Fig. 4. The ROC and PR curves for the dataset without spurious sources are clearly closer to the axes, giving higher values of AUC. The model also exhibits better calibration, lower ECE, in Fig. 5: the green curves are closer to the blue diagonal than the red ones.

6.3. Class imbalance

The biggest challenge for this work is the imbalance between the classes WAT and NAT. As mentioned before, the class imbalance is 66%: 385 sources out of 639 are WATs. The impact of this imbalance is apparent in all our performance metrics. WATs exhibit higher precision, recall and F1-score. The ROC curves do not show any bias toward any of the classes, but the PR curves clearly show a difference. In the right panel of Fig. 4, the solid lines (WAT) are closer to the axes than the dashed lines (NAT), giving higher values of AUC-PR.

The difference between the classes is most apparent in Fig. 6: the separation between the solid and dashed lines of the same color is much higher than the difference between the same type of lines of different colors. The distance r_{il} is much higher for NATs than WATs. We can see examples of this in the figures of Appendix B, as described in Section 6.1. Although class imbalance is a challenge for any ML model, it should be noted that WATs will always be more numerous than NATs because AGN jets are originally straight and bends only slowly. Given this limitation, it is remarkable that our model was able to reach good F1-scores for both classes.

7. Conclusions

With this paper, we publish two ML-friendly datasets ($\mathbf{R}_{L1}$ and $\mathbf{R}_{L2}$) of radio active galactic nuclei (RAGN) labeled as wide-angle-tail (WAT) and narrow-angle-tail (NAT) sources, and a semi-supervised model for the binary classification of WATs and NATs trained and tested on these labeled datasets and a publicly available unlabeled dataset ($\mathbf{R}_U$). Both datasets and models are part of our Github and pypi package called `RGC`. A previous version of $\mathbf{R}_{L1}$ was published in Hossain et al. (2025) with some initial benchmarking results and a previous version of the RGC model was briefly described in Hossain et al. (2023). All our datasets are created from the VLA images of RAGNs taken as part of the FIRST survey.

As described in Section 3 of this paper, $\mathbf{R}_U$ contains around 20,000 unlabeled sources, and $\mathbf{R}_{L1}$ and $\mathbf{R}_{L2}$ contain 639 labeled sources pre-processed by us through `PyBDSF` and `Photutils`, respectively. The second labeled dataset $\mathbf{R}_{L2}$ does not contain any spurious source, unlike the first labeled dataset. Among the labeled sources, 66% are WATs. A catalog of the labeled sources is included in `RGC` package, and a demonstration of the catalog is given in Appendix A. This catalog was created using a catalog of more than 700 WATs and NATs classified through visual inspection by Sasmal et al. (2022a). Although over 4000 WATs and NATs have been identified in the FIRST images (Table 1), we adopt the Sasmal catalog for its reliability as a fully visually inspected sample.

Our RGC model, described in Section 4, consists of a self-supervised pre-training stage with BYOL on the unlabeled data, followed by a supervised fine-tuning stage with E2CNN on the two labeled datasets. The performance and calibration of the model is described in Section 5, and its transparency and limitations are described in Section 6. Here we summarize the key results of the model trained and tested on the aforementioned datasets.

1. The semi-supervised RGC models perform better than the baseline supervised models with accuracies of 88.9% for the dataset $\mathbf{R}_{L2}$ and 87.3% for $\mathbf{R}_{L1}$ (Table 4).
2. The RGC model trained and tested on the dataset without spurious sources ($\mathbf{R}_{L2}$) performs better than the one with spurious sources in the performance metrics (Fig. 3 and Table 4), discriminative ability (ROC and PR curves of Fig. 4) and model calibration (ECE curves of Fig. 5).
3. Because of class imbalance, the precision, recall and F1-score for almost all models are better for WATs than NATs, but the RGC model trained and tested on $\mathbf{R}_{L2}$ achieves high F1-scores for both WATs (0.9091) and NATs (0.8571).
4. If there are spurious sources in an image, the RGC model gives attention to both the target and spurious sources as evident from the Grad-CAM localization maps overplotted as contours in the figures of Appendix B. The better performance of $\mathbf{R}_{L2}$ can be attributed to the absence of spurious sources, especially evident from the transparency metrics: ROC, PR and ECE curves.
5. The class imbalance between WATs and NATs has a significant effect on the attention of the RGC model. For WATs, the attention contours (Appendix B) are typically centered on the peak and enclose the source almost symmetrically. In contrast, for NATs, the contours tend to trace mostly the outer edges of the source (see Fig. 6).
6. The Sasmal catalog originally contained 717 sources, but we were able to include reliable FIRST images for only 639 of them. The remaining sources were discarded due to unavailability, compact morphology, or ambiguity in classification.

Group-equivariant convolutions have demonstrated strong performance in the supervised classification of RAGNs (Scaife &

Porter 2021), while the self-supervised BYOL framework has shown great potential for developing foundation models in radio astronomy (Slijepcevic et al. 2024). Our RGC model, trained and tested on two novel datasets, contributes to the global effort toward building foundation models for the forthcoming hyper-data era of radio astronomy.

Data and Code Availability

All our data and code are available in our Github package `cassaiub/rgc` linked in the footnotes of Page 1.

Acknowledgments

Asad, Amin, and Momen acknowledge support from the ICT Innovation Fund Grant No. 148 (2020) of the ICT Division, Government of Bangladesh, and from the Sponsored Research Grant No. 2020-SETS-13 of Independent University, Bangladesh (IUB). The authors used ChatGPT (OpenAI, 2025) to refine less than 10% of the manuscript text for better readability. The authors reviewed and edited all content refined by AI and take full responsibility for the final version of the manuscript.

Contributor Roles Taxonomy (CRediT)

The first five authors are listed in order of significance of their contribution, the last author is the supervisor of the first author, and the second-last author is the supervisor of the second author. All other authors are listed alphabetically. Here we describe the author contributions using the taxonomy given in Table 1 of Brand et al. 2015. *Hossain*: Methodology, Formal Analysis, Software, Data Curation, Investigation; *Shahal*: Methodology, Software, Validation, Data Curation, Visualization; *Asad*: Conceptualization, Writing - Original Draft; *Saikia*: Conceptualization, Writing - Review and Editing; *Khan*: Data Curation, Formal Analysis, Visualization; *Akter*: Data Curation; *Ali*: Supervision; *Amin*: Project Administration, Funding acquisition; *Momen*: Resources; *Hasan*: Supervision; *Rahman*: Supervision, Methodology, Project Administration.

References

Alam, S., Albareti, F. D., Allende Prieto, C., et al. 2015, ApJS, 219, 12
Banfield, J. K., Wong, O. I., Willett, K. W., et al. 2015, MNRAS, 453, 2326
Becker, R. H., White, R. L., & Helfand, D. J. 1995, ApJ, 450, 559
Best, P. N. & Heckman, T. M. 2012, MNRAS, 421, 1569
Bhukta, N., Mondal, S. K., & Pal, S. 2022, MNRAS, 516, 372
Bradley, L., Sipőcz, B., Robitaille, T., et al. 2025, astropy/photutils: 2.2.0
Brand, A., Allen, L., Altman, M., Hlava, M., & Scott, J. 2015, Learned Publishing, 28, 151
Bulbul, E., Liu, A., Kluge, M., et al. 2024, A&A, 685, A106
Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. 2020, in Proceedings of the 37th International Conference on Machine Learning, ICML'20 (JMLR.org)
Conselice, C. J., Wilkinson, A., Duncan, K., & Mortlock, A. 2016, ApJ, 830, 83
Dehghan, S., Johnston-Hollitt, M., Franzen, T. M. O., Norris, R. P., & Miller, N. A. 2014, AJ, 148, 75
Fanaroff, B. L. & Riley, J. M. 1974, MNRAS, 167, 31P
Garon, A. F., Rudnick, L., Wong, O. I., et al. 2019, AJ, 157, 126
Golden-Marx, E., Blanton, E. L., Paterno-Mahler, R., et al. 2021, ApJ, 907, 65
Grill, J.-B., Strub, F., Altché, F., et al. 2020, in Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20 (Red Hook, NY, USA: Curran Associates Inc.)
Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. 2017, in Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17 (JMLR.org), 1321–1330
Hossain, M. S., Asad, K. M. B., Saikia, P., et al. 2025, in 2025 IEEE International Conference on Image Processing (ICIP), 2868–2873
Hossain, M. S., Roy, S., Asad, K., et al. 2023, Procedia Computer Science, 222, 601, international Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023)
Kapoor, R. 2023, Journal of Astronomical History and Heritage, 26, 411
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems, ed. F. Pereira, C. Burges, L. Bottou, & K. Weinberger, Vol. 25 (Curran Associates, Inc.)
Kurcz, A., Bilicki, M., Solarz, A., et al. 2016, A&A, 592, A25
Lao, B., Andernach, H., Yang, X., et al. 2025, ApJS, 276, 46
Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, Proceedings of the IEEE, 86, 2278
Maitra, D. 2021, in 110th Annual Meeting of the American Association of Variable Star Observers (AAVSO)
Merloni, A., Lamer, G., Liu, T., et al. 2024, A&A, 682, A34
Miley, G. K., Perola, G. C., van der Kruit, P. C., & van der Laan, H. 1972, Nature, 237, 269
Mingo, B., Croston, J. H., Hardcastle, M. J., et al. 2019, MNRAS, 488, 2701
Mohan, N. & Rafferty, D. 2015, PyBDSF: Python Blob Detection and Source Finder, Astrophysics Source Code Library, record ascl:1502.007
Ndung'u, S., Grobler, T., Wijnholds, S. J., Karastoyanova, D., & Azzopardi, G. 2023, New A Rev., 97, 101685
Norris, R., Basu, K., Brown, M., et al. 2015, in Advancing Astrophysics with the Square Kilometre Array (AASKA14), 86
Norris, R. P. 2017, Nature Astronomy, 1, 671
Norris, R. P., Marvil, J., Collier, J. D., et al. 2021, PASA, 38, e046
O'Dea, C. P. & Baum, S. A. 2023, Galaxies, 11, 67
Owen, F. N. & Rudnick, L. 1976, ApJ, 205, L1
Pal, S. & Kumari, S. 2023, Journal of Astrophysics and Astronomy, 44, 17
Porter, F. A. M. & Scaife, A. M. M. 2023, RAS Techniques and Instruments, 2, 293
Predehl, P., Andritschke, R., Arefiev, V., et al. 2021, A&A, 647, A1
Proctor, D. D. 2011, ApJS, 194, 31
Rudnick, L. & Owen, F. N. 1976, ApJ, 203, L107
Sasmal, T. K., Bera, S., Pal, S., & Mondal, S. 2022a, ApJS, 259, 31
Sasmal, T. K., Bera, S., Pal, S., & Mondal, S. 2022b, VizieR Online Data Catalog: Head-tail radio galaxies from the VLA FIRST survey (Sasmal+, 2022), VizieR On-line Data Catalog: J/ApJS/259/31. Originally published in: 2022ApJS..259...31S
Scaife, A. M. M. & Porter, F. 2021, MNRAS, 503, 2369
Selvaraju, R. R., Cogswell, M., Das, A., et al. 2017, in 2017 IEEE International Conference on Computer Vision (ICCV), 618–626
Shimwell, T. W., Hardcastle, M. J., Tasse, C., et al. 2022, A&A, 659, A1
Slijepcevic, I. V., Scaife, A., Walmsley, M., & Bowles, M. R. 2022a, in Machine Learning for Astrophysics, 53
Slijepcevic, I. V., Scaife, A. M. M., Walmsley, M., et al. 2022b, MNRAS, 514, 2599
Slijepcevic, I. V., Scaife, A. M. M., Walmsley, M., et al. 2024, RAS Techniques and Instruments, 3, 19
Sohn, K., Berthelot, D., Li, C.-L., et al. 2020, in Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20 (Red Hook, NY, USA: Curran Associates Inc.)
Urry, C. M. & Padovani, P. 1995, 107, 803
Vardoulaki, E., Backöfer, V., Finoguenov, A., et al. 2025, A&A, 695, A178
Weiler, M. & Cesa, G. 2019, General E(2)-equivariant steerable CNNs (Red Hook, NY, USA: Curran Associates Inc.)
Wong, O. I., Garon, A. F., Alger, M. J., et al. 2025, MNRAS, 536, 3488

Appendix A: Sasmal-ML catalog

Table A.1: Information about the 63 sources from our Sasmal-ML catalog that are included in our Test Batch.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
RI	SI	MI	FCG	RA	Dec	Label	Batch	d	T_isl	T_pix	Saikia	Asad	Hossain	Attention RL1	Attention RL2
10	12	9	J0046-0805	00 46 36.48	-08 05 45.6	WAT	test batch	2	1.6	5	...	sig	SMD	g	g
12	14	10	J0057-0336	00 57 33.71	-03 36 09.6	WAT	test batch	0	3	5			SMD	s	g
18	21	15	J0229+0429	02 29 02.09	+04 29 03.3	WAT	test batch	0	3	5			SMD	s	g
24	27	21	J0312-0633	03 12 55.23	-06 33 43.1	WAT	test batch	0	3	5			SMD	g	g
33	37	29	J0715+4829	07 15 19.07	+48 29 32.0	WAT	test batch	0	3	5			SMD	g	g
34	38	30	J0717+4405	07 17 26.66	+44 05 02.3	WAT	test batch	0	3	5			SMD	g	g
43	48	37	J0744+4353	07 44 42.18	+43 53 07.8	WAT	test batch	0	3	5			SMD	g	g
44	47	38	J0744+1658	07 44 59.31	+16 58 59.7	WAT	test batch	0	3	5			SMD	g	g
48	52	42	J0750+5841	07 50 18.30	+58 41 36.3	WAT	test batch	0	3	5			SMD	g	g
49	51	43	J0750+1741	07 50 49.80	+17 41 41.9	WAT	test batch	0	3	5			SMD	s	g
54	58	47	J0811+1029	08 11 40.88	+10 29 27.3	WAT	test batch	0	3	5			SMD	c	g
91	95	81	J0902+1819	09 02 16.30	+18 19 04.3	WAT	test batch	0	3	5			SMD	g	g
93	97	83	J0904+5834	09 04 57.31	+58 34 49.9	WAT	test batch	0	3	5			SMD	s	g
122	127	110	J0943+0611	09 43 09.45	+06 11 31.8	WAT	test batch	0	3	5			SMD	s	g
151	156	137	J1031-0640	10 31 30.66	-06 40 28.1	WAT	test batch	0	3	5			SMD	s	g
168	174	152	J1105+5943	11 05 24.26	+59 43 41.9	WAT	test batch	0	3	5			SMD	s	g
171	177	155	J1114+3625	11 14 01.78	+34 25 03.2	WAT	test batch	0	3	5			SMD	g	g
184	190	168	J1130+2524	11 30 48.83	+25 24 35.6	WAT	test batch	0	3	5			SMD	c	c
194	200	177	J1145-0757	11 45 53.08	-07 57 53.6	WAT	test batch	0	3	5		bad	SMD	g	g
197	203	180	J1148+2332	11 48 33.14	+23 32 25.9	WAT	test batch	0	3	5			SMD	s2	g
207	213	189	J1201+3257	12 01 51.87	+32 57 01.3	WAT	test batch	0	3	5			SMD	g	g
215	220	197	J1213-0327	12 13 45.43	-03 27 09.5	WAT	test batch	0	3	5			SMD	s	g
217	223	199	J1219+0014	12 19 05.57	+00 14 16.4	WAT	test batch	1	2.6	4	...	sig	SMD	g	g
230	236	212	J1247+4042	12 47 01.00	+40 42 02.3	WAT	test batch	0	3	5			SMD	s	g
234	240	216	J1250+0839	12 50 10.59	+08 39 52.7	WAT	test batch	0	3	5			SMD	s	g
264	274	242	J1345+6110	13 45 03.06	+61 10 31.6	WAT	test batch	0	3	5			SMD	g	g
265	271	243	J1345+0855	13 45 04.67	+08 55 07.2	WAT	test batch	1	1.8	3.8		sig	SMD	s	g
270	276	248	J1351+0712	13 51 44.72	+07 12 17.0	WAT	test batch	0	3	5			SMD	s	g
276	284	253	J1401+5654	14 01 09.98	+56 54 20.6	WAT	test batch	0	3	5			SMD	s	g
291	298	268	J1416+2147	14 16 13.79	+21 47 42.8	WAT	test batch	1	2	5		sig	SMD	s3	g
352	360	324	J1601+3155	16 01 30.74	+31 55 32.4	WAT	test batch	0	3	5			SMD	s	g
355	363	325	J1608+0141	16 08 03.55	+01 41 54.34	WAT	test batch	0	3	5			SMD	g	g
357	365	326	J1616+0926	16 16 53.17	+09 26 35.7	WAT	test batch	0	3	5			SMD	g	g
390	399	356	J2139+1008	21 39 46.25	+10 08 28.0	WAT	test batch	0	3	5			SMD	s2	g
402	411	367	J2213-0854	22 13 12.43	-08 54 37.1	WAT	test batch	0	3	5			SMD	g	g
409	418	374	J2301+0037	23 01 35.06	+00 37 13.1	WAT	test batch	0	3	5			SMD	g	g
411	420	376	J2312-0919	23 12 12.16	-09 19 28.6	WAT	test batch	0	3	5			SMD	g	g
416	425	380	J2334-0759	23 34 55.59	-07 59 14.4	WAT	test batch	0	1	4.2	...	sig	SMD	g	cut
428	7	390	J0054+0302	00 54 43.83	+03 02 10.5	NAT	test batch	0	3	5			SMD	s	g
447	27	404	J0356+0013	03 56 50.87	+00 13 23.2	NAT	test batch	0	3	5			SMD	g	g
451	31	408	J0718+4820	07 18 25.52	+48 20 20.6	NAT	test batch	0	3	5			SMD	g	lim
454	34	411	J0731+3251	07 31 13.85	+32 51 15.1	NAT	test batch	0	3	5			SMD	g	g
474	56	426	J0829+6322	08 29 10.55	+63 22 28.5	NAT	test batch	0	3	5			SMD	g	g
476	58	428	J0834+3945	08 34 30.09	+39 45 09.5	NAT	test batch	0	3	5			SMD	s	n
478	60	430	J0846+2024	08 46 47.94	+20 24 48.9	NAT	test batch	0	3	5			SMD	g	lim
480	62	432	J0851+1615	08 51 10.34	+16 15 30.6	NAT	test batch	0	3	5			SMD	s	g
484	66	436	J0902+3505	09 02 30.30	+35 05 11.0	NAT	test batch	0	3	5			SMD	n	g
493	75	445	J0933+1341	09 33 28.30	+13 41 37.5	NAT	test batch	0	3	5			SMD	lim	n
509	92	459	J1016+4708	10 16 06.64	+47 08 06.6	NAT	test batch	0	3	5			SMD	s	lim
514	97	464	J1032+3152	10 32 15.87	+31 52 35.7	NAT	test batch	0	3	5			SMD	s	lim
536	119	481	J1132+1344	11 32 40.23	+13 44 26.2	NAT	test batch	0	3	5			SMD	g	g
541	124	485	J1145+1529	11 45 22.20	+15 29 43.	NAT	test batch	0	3	5			SMD	s	g
552	135	496	J1217+0336	12 17 31.4	+03 36 57.0	NAT	test batch	0	3	5			SMD	n	lim
555	138	499	J1222-0449	12 22 23.69	-04 49 35.4	NAT	test batch	0	3	5			SMD	s	g
600	183	542	J1403+0610	14 03 13.29	+06 10 08.3	NAT	test batch	0	3	5			SMD	g	g
602	185	544	J1407+4257	14 07 13.04	+42 57 35.7	NAT	test batch	0	3	5			SMD	n	g
625	208	565	J1509+3327	15 09 57.37	+33 27 15.0	NAT	test batch	0	3	5			SMD	s	g (s)
635	218	575	J1541+0301	15 41 41.70	+03 01 03.0	NAT	test batch	0	3	5			SMD	g	g
668	251	605	J1652+3055	16 52 05.38	+30 55 17.0	NAT	test batch	0	3	5			SMD	s	g
670	253	607	J1700+4442	17 00 47.39	+44 42 42.0	NAT	test batch	0	3	5			SMD	n	g
680	264	617	J1733+4229	17 33 43.25	+42 29 12.1	NAT	test batch	0	3	5			SMD	g	g
682	266	619	J2102-0145	21 02 14.61	-01 45 09.5	NAT	test batch	0	3	5			SMD	g	g
694	279	631	J2239-0932	22 39 02.43	-09 32 35.6	NAT	test batch	0	3	5			SMD	g	g

This table (Table A.1) contains the 63 sources used in our Test Batch. The complete annotated catalog can be found in our Github package (`https://github.com/cassaiub/rgc/`). The 16 numbered columns of this catalog are described below.

1. RI: RGC Index of 703 sources used throughout this paper for all bent radio active galactic nuclei (RAGN).
2. SI: original Sasmal Index of these RAGNs used in the catalog of Sasmal et al. (2022a).
3. MI: Masking Index of the 639 sources which were finally included in our labeled datasets; the pre-processing masks are stored using these indices.
4. FCG stands for FIRST Component Group or FIRST Catalog Galaxy, which is a common prefix for the IAU-compliant names of these objects.
5. RA: right ascension coordinate of a source.
6. Dec: declination coordinate of a source.
7. Label: the label given by Sasmal et al. (2022a) which is our ground truth.
8. Batch: the batch type in our batched labeled dataset $\mathbf{R}_{L1}$ and $\mathbf{R}_{L2}$.
9. d: the dilation used in `PyBDSF` preprocessing.
10. T_{isl}: the island threshold used in `PyBDSF` preprocessing.
11. T_{pix}: the pixel threshold used in `PyBDSF` preprocessing.
12. Saikia: the annotations by Saikia, hidden in this table with '...'. The complete annotations can be found in Github.
13. Asad: the annotations by Asad. Here 'sig' means more sigma clipping was needed during preprocessing, 'bad' means the source should be removed.
14. Hossain: the annotations by Hossain.
15. Attention RL1: comments about the quality of the attentions of our model on the dataset $\mathbf{R}_{L1}$. Here 'g' indicates good attention, 's' indicates the presence of a spurious source, 's2' and 's3' two and three spurious sources, respectively, 'c' indicates confusion about whether a component is spurious or part of the target source, 'n' indicates no attention, 'lim' stands for limited attention that did not cover or delimit the source wholly.
16. Attention RL2: comments about the quality of the attentions of our model on the dataset $\mathbf{R}_{L2}$. The letters have the same meaning as the previous column, except 'cut' which means part of the source was cropped while trying to remove spurious sources.

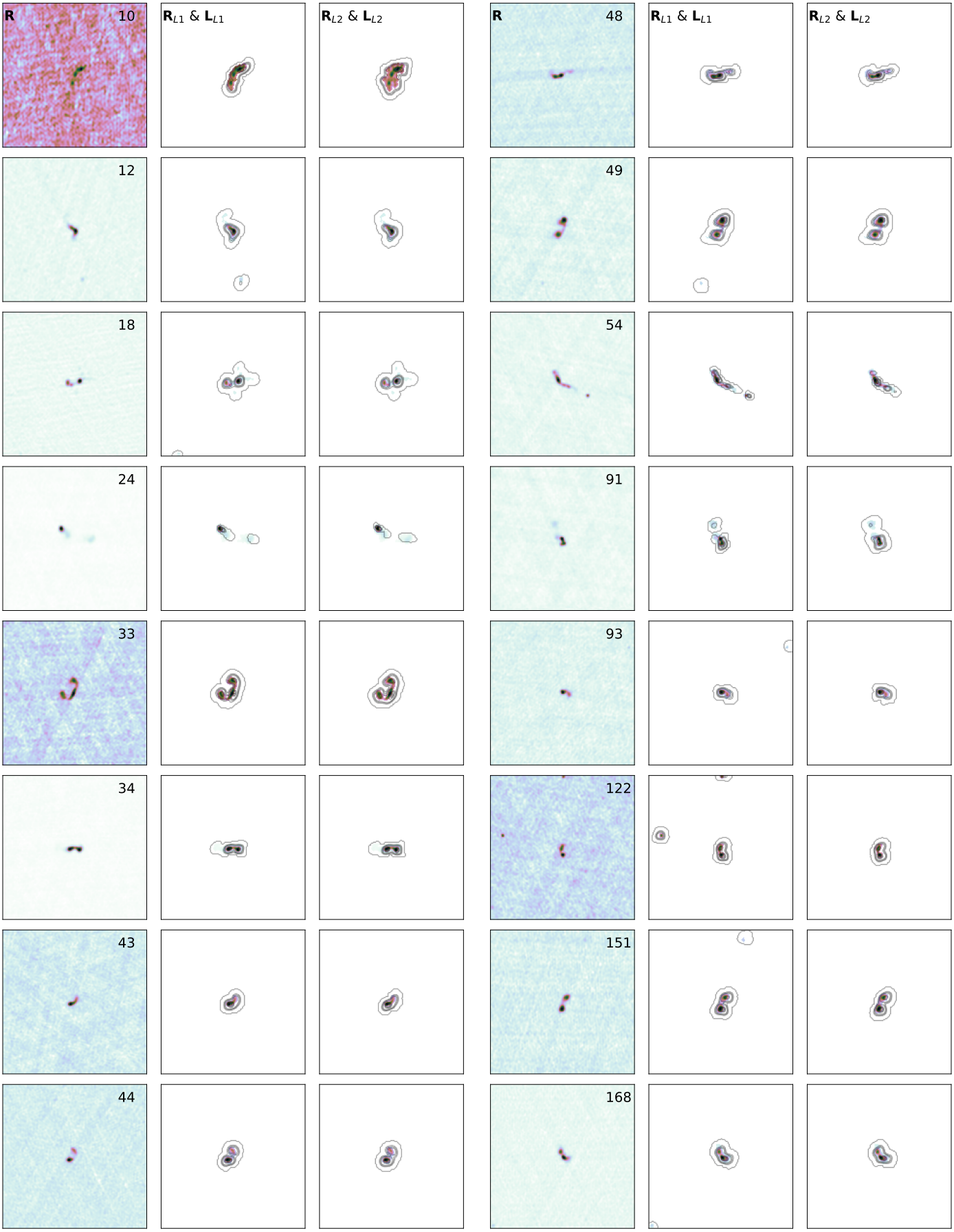
Appendix B: Grad-CAM localization maps

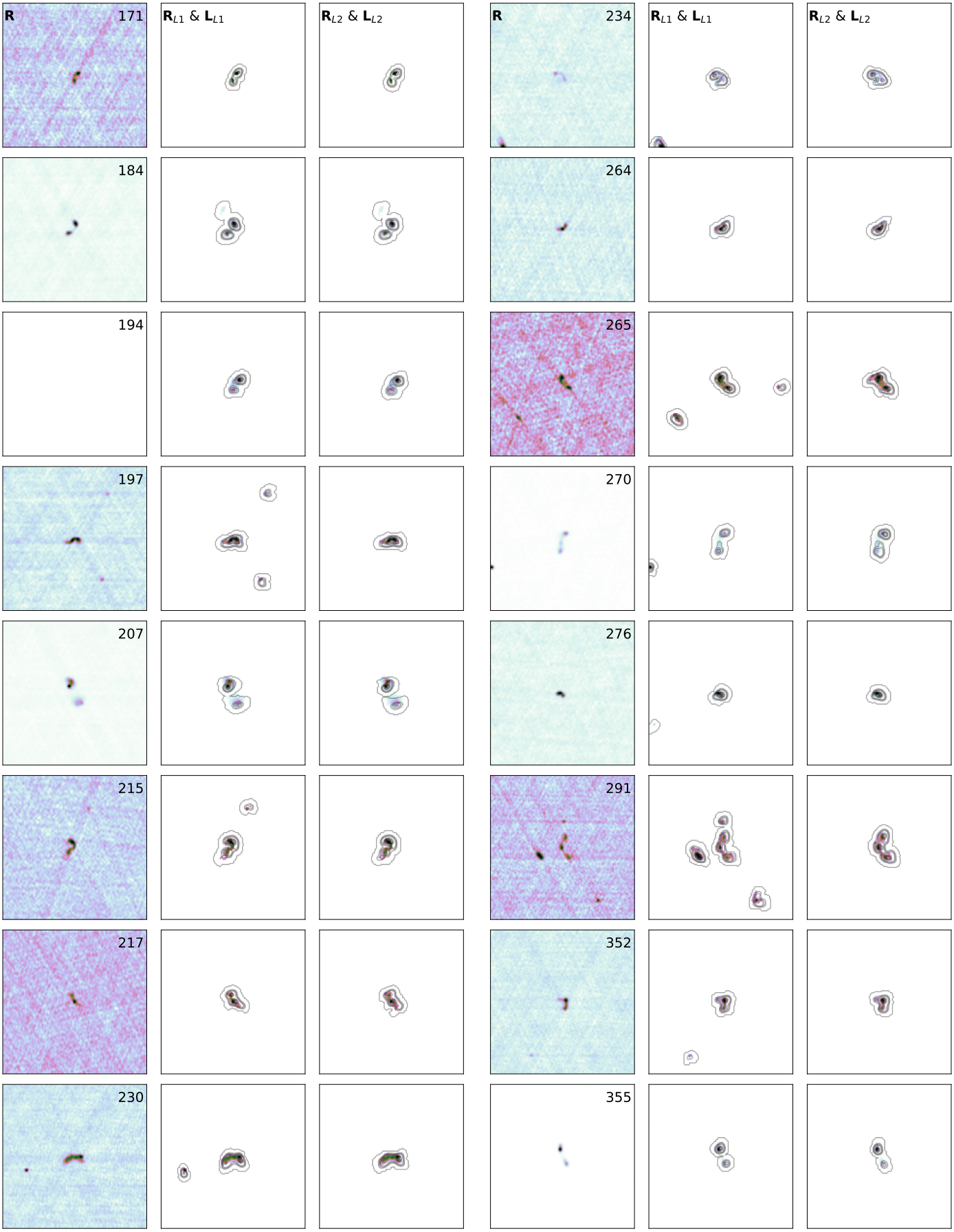
In the following pages we provide the original images $\mathbf{R}$, the masked images $\mathbf{R}_{L1}$ pre-processed using `PyBDSF`, and the masked images $\mathbf{R}_{L2}$ pre-processed using `Photutils` for the 63 RAGNs in the test batch of our batched datasets.

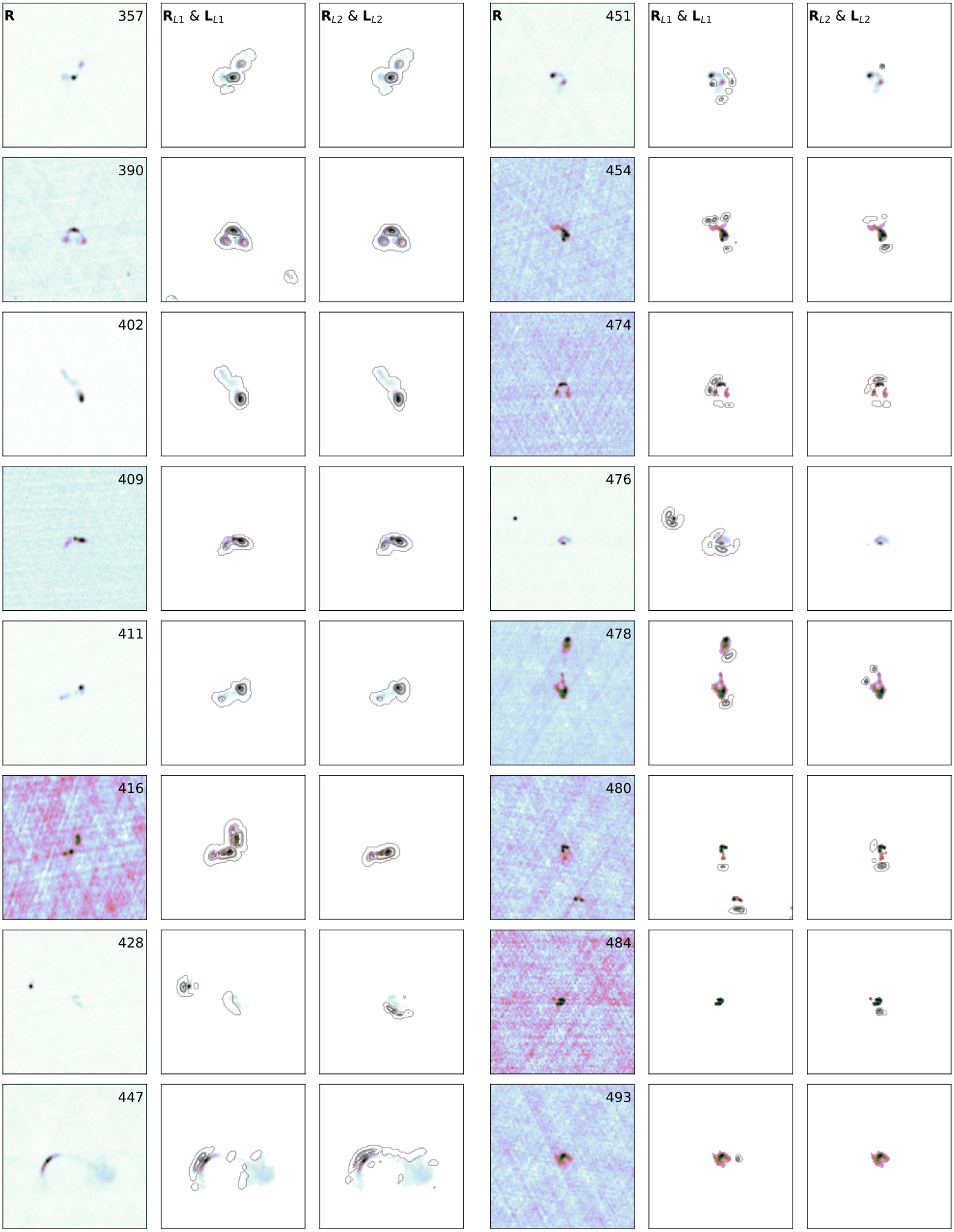
The corresponding localization maps produced by Grad-CAM ($\mathbf{L}_{L1}$ and $\mathbf{L}_{L2}$) are overplotted on the masked images as contours. The contours have been calculated from the heatmaps produced following the mechanism described in Section 4.4.

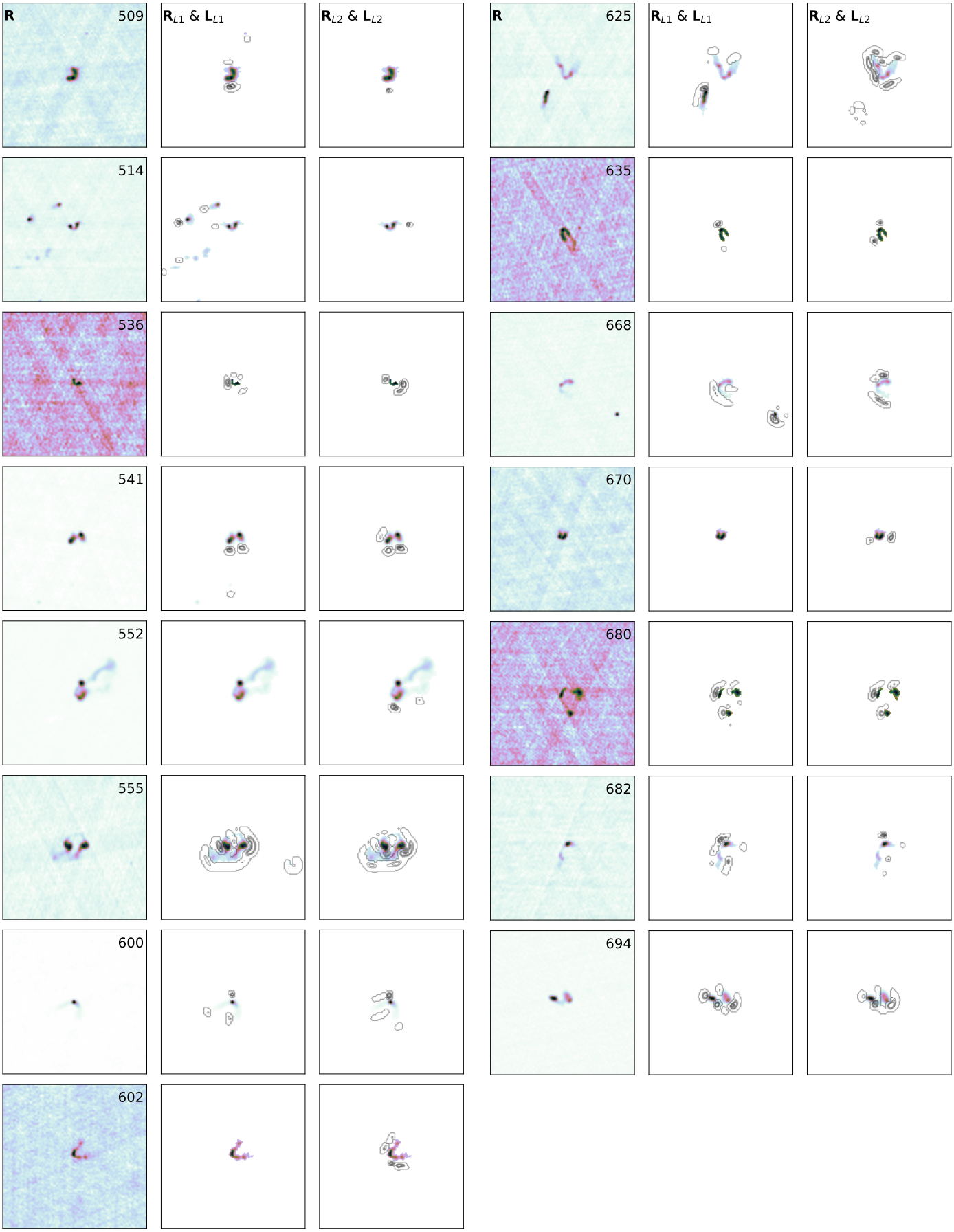
Each page has 6 columns and 8 rows. The first 3 columns show the original and masked images of 8 sources, and the last three columns show 8 more sources, resulting in 16 sources per page, except the last page where we show 15 sources.

The detailed labels are given only on the first row of each page, which are the same for the lower rows of the page. The integer on the upper right corner of every image on the first columns indicate the RGC index (RI), as given in the first column of Table A.1 in the previous appendix.


R
10
R_{L1} & L_{L1}
R_{L2} & L_{L2}
R
48
R_{L1} & L_{L1}
R_{L2} & L_{L2}
12
49
18
54
24
91
33
93
34
122
43
151
44
168


R
171
R$_{L1}$ & L$_{L1}$
R$_{L2}$ & L$_{L2}$
R
234
R$_{L1}$ & L$_{L1}$
R$_{L2}$ & L$_{L2}$
184
264
194
265
197
270
207
276
215
291
217
352
230
355


R
357
R_{L1} & L_{L1}
R_{L2} & L_{L2}
R
451
R_{L1} & L_{L1}
R_{L2} & L_{L2}
390
454
402
474
409
476
411
478
416
480
428
484
447
493


R
509
R_{L1} & L_{L1}
R_{L2} & L_{L2}
R
625
R_{L1} & L_{L1}
R_{L2} & L_{L2}
514
635
536
668
541
670
552
680
555
682
600
694
602