The Astronomical Journal, 168:84 (9pp), 2024 August https://doi.org/10.3847/1538-3881/ad5a80

IOP4, the Interactive Optical Photo-Polarimetric Python Pipeline

Juan Escudero Pedrosa[1], Iván Agudo[1], Daniel Morcuende[1], Jorge Otero-Santos[1], Giacomo Bonnoli[2], Vilppu Piirola[3], César Husillos[1,4], Mabel Bernardos[1], Rubén López-Coto[1], Alfredo Sota[1], Victor Casanova[1], Fran J. Aceituno[1], and Pablo Santos-Sanz[1]

[1] Instituto de Astrofísica de Andalucía, Glorieta de la Astronomía, s/n. Granada, 18008, Spain; jescudero@iaa.es
[2] INAF Osservatorio Astronomico di Brera, Via E. Bianchi 46. Merate (LC), 23807, Italy
[3] Department of Physics and Astronomy, University of Turku, Vesilinnantie 5. Turku, FI-20014, Finland
[4] Geological and Mining Institute of Spain (IGME-CSIC), Calle Ríos Rosas 23, E-28003, Madrid, Spain
Received 2024 April 23; revised 2024 June 11; accepted 2024 June 18; published 2024 July 22

Abstract

IOP4 is a pipeline to perform photometry and polarimetry analysis of optical data from Calar Alto (CAHA) and Sierra Nevada (OSN) observatories. IOP4 implements Object Relational Mapping to seamlessly integrate all information about the reduction and results in a database that can be used to query and plot results, flag data, and inspect the reduction process in an integrated fashion with the whole pipeline. It also ships with an already built-in web interface that can be used out of the box to browse the database and supervise all pipeline processes. It is built to ease debugging and inspection of data. Reduction from five different instruments are already implemented: RoperT90, AndorT90, DIPOL (at OSN 0.9 m telescope), AndorT150 (OSN 1.5 m telescope), and CAFOS (CAHA 2.2 m telescope). IOP4's modular design allows for easy integration of new observatories and instruments, and its results have already featured in several high-impact refereed publications. In this paper we describe the implementation and characteristics of IOP4.

Unified Astronomy Thesaurus concepts: Astronomy data analysis (1858); Photometry (1234); Polarimetry (1278); Astronomy databases (83); Open source software (1866)

1. Introduction

Optical photo-polarimetric observational programs, especially those dedicated to monitoring, can regularly produce large quantities of data that can take considerable time and effort to be managed and reduced. The effort necessary to produce good quality results extends beyond the use of automatic tools and can include a human-supervised iterative process of debugging the reduction and comparing the employed methods and results with those of different programs. The use of automatic tools is therefore necessary; however, in many instances they obscure the process of reduction and intermediate results, making the debugging of any problem in the results a hard task.

IOP4 implements object-relational mapping using Django's ORM system. This allows to transparently keep the database schema up to date with the models used by the pipeline without any need to mess with the underlying SQL queries. The choice of Django's as ORM backend allows to seamlessly use the rest of Django Framework to serve the results, including but not limited to its admin interface to inspect the database, and Django's debug web server. These tools are all written in Python and packaged for pip and conda, a programming language and distributions that many astronomers are already familiarized with, easing its installation and usage. This also makes IOP4 a multi-platform software, compatible both with macOS and Linux.

The main goal of being a multiinstrument pipeline differentiates IOP4 from preexisting software, which is usually instrument-specific. Five different instruments (RoperT90, AndorT90, AndorT150, DIPOL, and CAFOS) from three different telescopes (Sierra Nevada Observatory (OSN) 0.9 m, OSN 1.5 m, and Calar Alto Observatory (CAHA) 2.2m) in two different observatories (Sierra Nevada and Calar Alto) are already implemented (see Section 6). The choice of Python as the main programming language and IOP4's modular design facilitate the integration of new instruments and the implementation of new reduction procedures to the wider astronomical community. IOP4 intends to be not only a pipeline for data reduction, but to provide a fully equipped portal and web interface. This is specially useful for teams where the tasks of performing observations, reducing, inspecting, and debugging data, and publication or sharing of results are divided among several people, as large monitoring and observational programs often require.

IOP4 builds on top of existing technologies, some of them already cited: Django (ORM and web application framework), SQLite (default database backend), astrometry.net (blind astrometric calibration; Lang et al. 2010), Bokeh (high quality and interactive plots in the web interface), Vue.js (single-page web application), Quasar (user interface components), and JS9 (a ds9 web port for interactive FITS visualization). Many other open-source packages are used, the complete list can be found in the `pyproject.toml` file. All of them are automatically installed together with IOP4.

Hardware requirements of IOP4 are low, even compared today to modern day consumer-end laptops. The blind astrometric calibration using the astrometry.net solver and its index files can take as much as 35 GB with the default configuration, although this requirement can be lowered. SSD storage is also recommended, although not necessary, since IOP4 needs to read significant amounts of data and will benefit from the increased speed, especially during the astrometric calibration. Although it is able to run on a single-core,

significant speed improvements can be gained from using up to 20 cores, the recommended maximum with the default configuration and database (DB) backend. Higher levels of concurrency are possible by tuning the DB configuration or specifying a different backend. A good internet connection can speed up the first execution of IOP4, including tests, since it will need to download astrometry index files.

IOP4 facilitates both automatic reduction and manual inspection of the procedure. The `iop4` command provides several options to select different epochs and files and automatically process them. The IOP4LIB allows any user to write its own custom reduction scripts, and provides the tools to invoke any part of the reduction procedure, and inspect and manipulate any object in the database from an interactive terminal, a Python script or a Jupyter Notebook. The IOP4API implements several Application Public Interface (API) end points and web applications that allow end users to easily query results, produce plots, flag bad data, and inspect any object. The portal can be used standalone as in the ready for use IOP4SITE project, or integrated into other sites to be served to the general public.

2. General Reduction Procedure

The automatic reduction procedure generally starts with a simple invocation of the `iop4` command. The main script can be requested to select epochs in the local archive or to discover and download epochs in the remote telescope repositories. For each epoch, the script goes through the usual steps of photo-polarimetric reduction:

1. Classification of raw science images: this includes their type (bias, darks, flats, and science images), discovering the instrument and type of observation (photometry or polarimetry), and translation of standard and nonstandard keywords (exposure time, band, datetime, rotator angles, and objects).
2. Creation of master calibration frames: images of each type (bias, darks, and flats) are grouped and merged together to create the master calibration frames available for each night.
3. Reduction and calibration of science images: raw images are applied the corresponding master calibration frames. The reduced images are also given a correct World Coordinate System (WCS) in their header after astrometric calibration.
4. Computation of photo-polarimetric results: at the moment, the procedure implements relative photometry using known calibrators in the field, and polarimetry both for half-wave ($\lambda/2$) retarder based polarizers and polarized filter wheels.
5. Post-processing of results: this includes correction of magnitude and degree of polarization to account for the host contribution (Nilsson et al. 2007), and the possibly needed transformation to a standard photometric system, if the instrument used for the observations does not have one.

The same methods that the main script uses for bulk processing of epochs, can be invoked from the command line or custom scripts using the IOP4LIB directly. The IOP4LIB can used to query objects in the DB, debug the reduction process, or create custom procedures, as the examples in the documentation show.

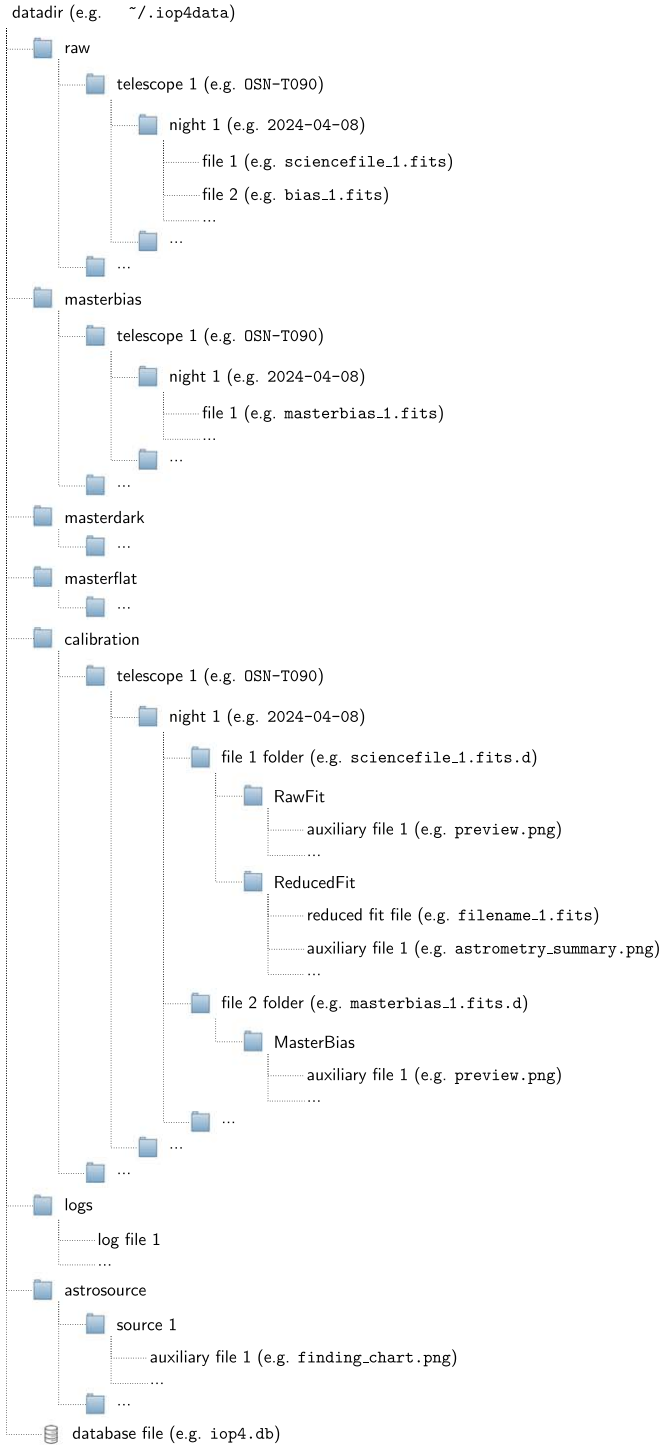

Figure 1. Directory structure of the IOP4 local archive.

3. Data Organization

IOP4 data directory structure follows the typical hierarchical schema shown in Figure 1. In this schema, all raw data is stored and isolated under a single folder (`raw/`), which allows to set up a local archive of the original data without any modifications for long-term conservation, and to set up the necessary permissions to protect and share it with other system users (e.g., creating a link) independently of the rest of IOP4-created files. Under the raw directory, data is organized first by telescope and then by night of observation. Other files such as built master calibration frames and

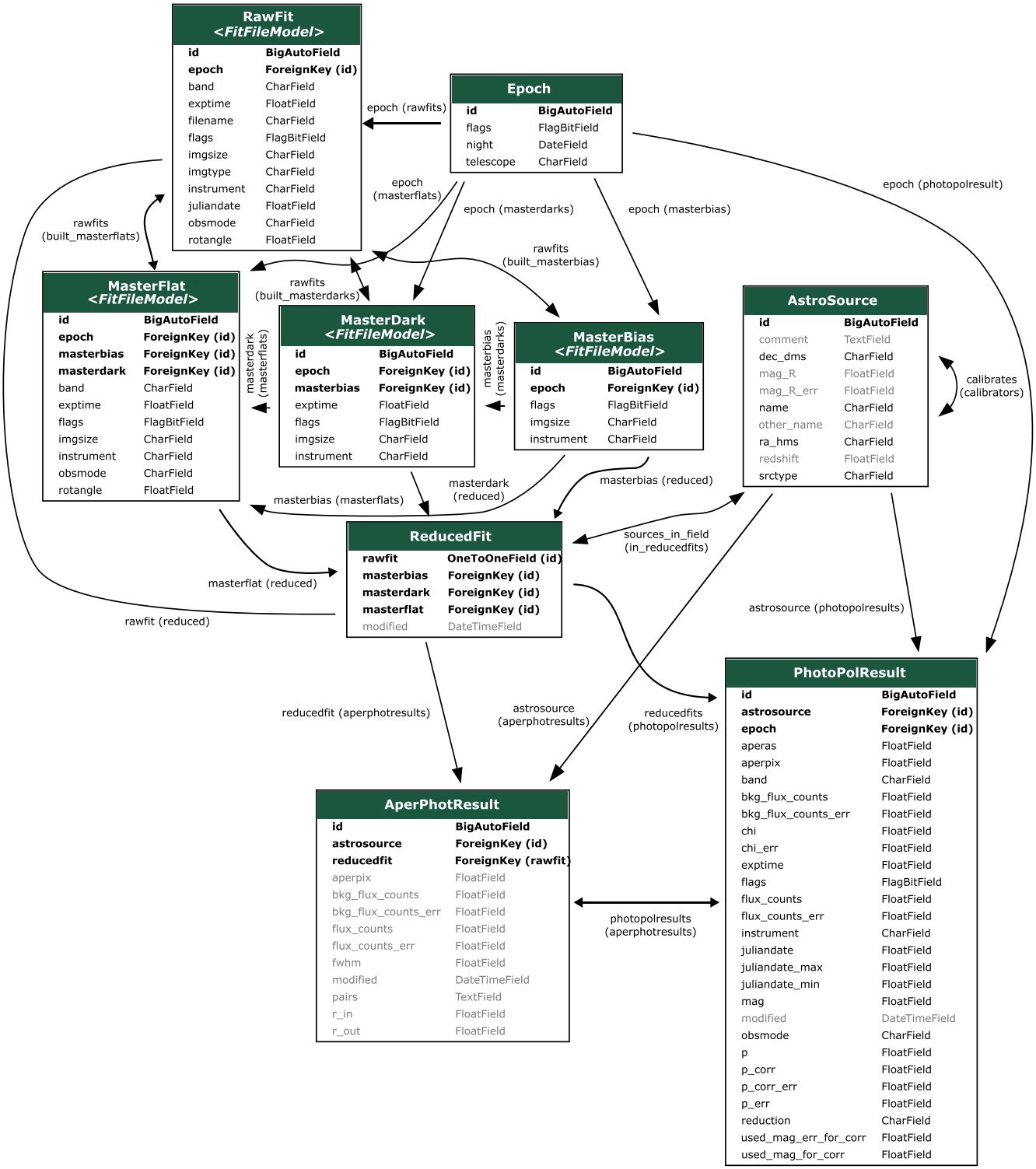

Figure 2. Database schema of IOP4. *ForeignKey* relationships relate one or several instances of model A to a single instance of model B. In *ManyToMany* relationships, multiple instances of model A and B are linked together. The latter does not show a field in the tables of this diagram, since the relationship is established through a hidden table omitted here. Some fields of the `AstroSource` model have been also omitted to save space (corresponding to literature magnitudes in other bands, e.g., `mag_B`, etc). An arrow in any direction signify multiple instances being linked to the origin (e.g., several RawFit(s) are linked to one Epoch).

reduced images are stored separately. Also, auxiliary images, such as automatically built previews, finding charts, summary plots, etc, which are too heavy to be stored in a database, are stored under different folders.

The database schema is shown in Figure 2. IOP4 implements object-relational mapping (ORM). In ORM, objects in a object-oriented programming language (Python classes) are mapped to tables in a relational database, while instances of these objects correspond to rows in each of the tables. This enormously simplifies the interaction with the database, removing the need for writing SQL queries and manipulating the database to keep its structure and content updated.

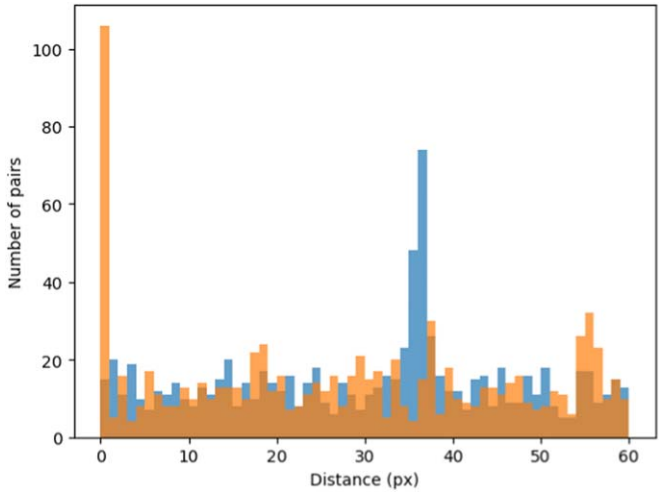

Figure 3. Distribution of the distances between all possible pairs in a imaging polarimetric frame. The two peaks correspond to the distance between the ordinary and extraordinary sources in the image. The image and the corresponding found pairs can be found in Figure 4. The orange and blue distributions correspond to the distances in the *X*- and *Y*-axis, respectively. The image corresponds to a CAFOS photo-polarimetric observation of the BL Lacertae field.

The default database backend is SQLite,[5] an in-process library that implements a self-contained, serverless, zero-configuration, and transactional SQL database; and uses the Django ORM[6] to interact with it. Customizations and changes to the DB schema can be written in the code (e.g., adding an attribute to the photo-polarimetric result model) and they will be automatically propagated to the DB schema by the Django migration system. IOP4 fine tunes SQLite configuration to improve its behavior and speed for high (write) concurrency, as it is needed for parallelization. This setup has been tested using up to 20 cores, although it is probable that a significantly higher number of cores is usable in the default configuration. For even higher workloads, such as those in cluster environments, a different DB backend can be specified (e.g., PostgreSQL).

4. Astrometric Calibration

Astronomical images are usually distributed in the FITS format (Pence et al. 2010). The format allows for a WCS (Greisen & Calabretta 2002) to be incorporated in the metadata or header section of the FITS file. However, most raw science images from telescopes do not include this precise information and need to be calibrated. For most images, which have a wide field of view ($>7'$) this is done by a local solver[7] based on the astrometry.net library solver.[8]

For observations with Ordinary (O) and Extraordinary (E) images (e.g., CAFOS and DIPOL imaging polarimetric observations), astrometric calibration involves a previous step of separating the detected sources in pairs. The source pairing is done by finding the most common distance between all pairs of sources in the image, which results in two distributions like those in Figure 3. Then, the separation can proceed by looking at which pairs are at the right shift (Figure 4). Without any constraint, this process is not completely error free, specially for images with few detected sources. However, since the distance between pairs is usually a stable property of the instrument, the known distance between pairs (obtained by the unconstrained pairing) for an instrument can be used as initial input, which improves the success rate up to 100%.

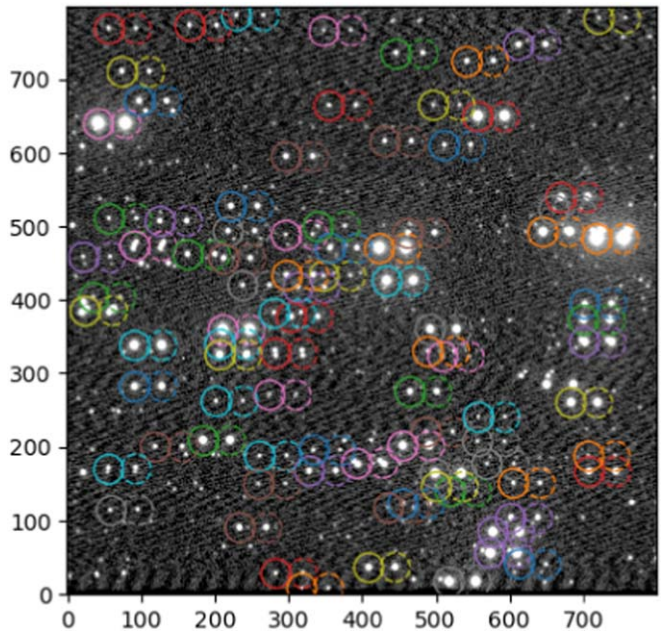

Figure 4. Paired sources in an example of CAFOS imaging polarimetric observation of the BL Lacertae field. The distance used for pairing was obtained from Figure 3. The resulting calibrated image can be found in Figure 5. Ordinary and Extraordinary sources are circled with the same color.

The detected sources in the image, or the ordinary set of pairs for images with pairs, can then be used as input for the local astrometry solver (Lang et al. 2010). The solver compares invariant hashes computed from the detected source positions to precomputed hashes from astronomical catalogs. Although it does not require an initial guess of the position nor the pixel size (blind solving), the speed and accuracy of the solution is greatly improved by providing the known pixel size for the instrument and a hint position, obtained from the header of the images. It returns a list of matches and their corresponding log-odds. Its success is dependent on the source detection step, and therefore several attempts are made with different detection parameters (such as signal threshold) until a good match is found. The resulting WCS is written to the header of the reduced FITS file. The WCS for the second or extraordinary set of pairs is directly built from the first one by translation, and written next to it in the same header. An example of the result can be found in Figure 5, which corresponds to the pairs in Figures 3 and 4.

4.1. A Quad Hash for DIPOL Polarimetry Images

To reduce the disk capacity requirements of DIPOL polarimetry images, only a subframe of the full field of DIPOL camera is saved (Section 6.3). The reduced field of view does not usually contain enough stars to perform the astrometric calibration using the default solver. In fact, in many cases, only the O and E images of the target source are visible in the image.

[5] https://www.sqlite.org/
[6] https://docs.djangoproject.com/
[7] The local solver is integrated as an external Python module dependency in the `pyproject.toml` file and is installed next to IOP4 automatically. The module is a wrapper around the native C functions of the astrometry.net library: https://github.com/neuromorphicsystems/astrometry.
[8] https://github.com/dstndstn/astrometry.net

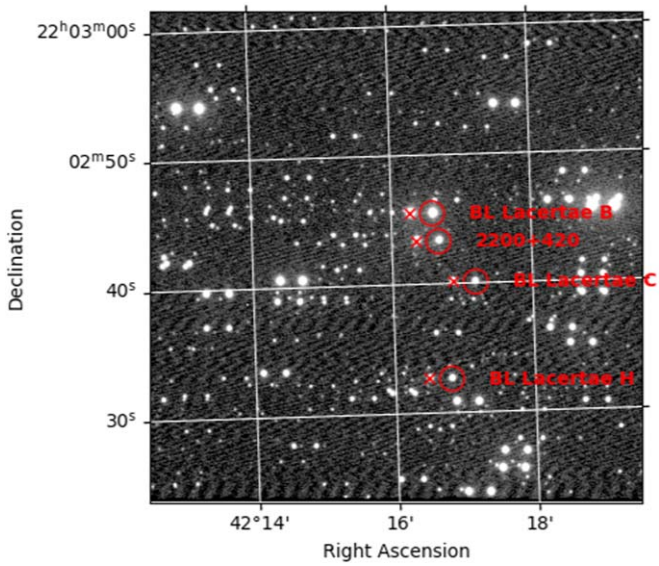

Figure 5. Calibrated CAFOS photo-polarimetric image of the BL Lacertae field. The positions of the ordinary image of BL Lacertae (labeled as 2200 +420) and its calibrators (B, C, H) are indicated with a circle. The position of the extraordinary images are represented with an "x."

For the case of images in which only one or two pairs of sources appear, the astrometric calibration can be as simple as using the central pair as a reference for the creation of the WCS with the known angle of the RA-DEC grid. However, this process is error-prone when more than two pairs of sources appear in the image, as the likelihood of choosing the wrong pair increases. Even for the case of high number of pairs, the size of the subframe ($2\farcm5 \times 2\farcm0$) is too small even for the smallest skymarks of the default astrometry.net index files.[9] The problem of choosing the right pair of sources as the reference can be solved by comparing the subframe of the polarimetry observation with the central part of a calibrated photometry (full frame) field. To this end, we have implemented a hash algorithm loosely based on Lang et al. (2010). The proposed hash is invariant under rotation and reflection, although not under scaling. An example of this algorithm at work can be found in Figure 6. It takes the ten brightest sources in each image (including E and O images) and compares all the quads between them. The best matching quads are used for identifying the target star.

5. High-level Web Interface

As one of its main goals, IOP4 also provides an API to interact with the data and a web interface to act as a client. Both are provided by the IOP4API Django application. It defines API end points to query results, produce interactive web-based plots, and flag data. It also integrates the IOP4ADMIN site, a customized Django-admin. The admin site allows us to inspect any models in the database (Figure 2).

The IOP4 catalog is part of the database. IOP4 uses a single, unified catalog that gathers all information about the sources of interest, including the target sources, calibrators, relationships between them, comments for observers, etc. Editing of the catalog can be done through IOP4 as with any other model, or through the web interface, and the changes take immediate effect on the reduction process of the pipeline.

The situation is illustrated in Figure 7. The exposed API end points are used by the single-page application (SPA) to authenticate the client, query, plot and flag data, and explore logs and catalog. The SPA provides a viewer for the colored logs and allows filtering by logging level (debug, info, warning, and error) and string searching. The interactive plot is built and serialized in the server using the Bokeh (Bokeh Development Team 2023) Python library, sent to the client and rendered by the BokehJS library. The interactive plot can be used to directly flag the data. TabulatorJS[10] is used to display the results, and allows filtering, selecting columns and exporting to several data formats such as CSV. The SPA itself is built in html, css and javascript using Vue.js framework as a standalone script to avoid the build step and allow IOP4 to be installed and used by the astrophysics community in a familiar way (through pip or conda).

The web application provides an auxiliary tool to facilitate the addition to the catalog of calibrators for sources with no known previously documented calibrators. A search for standard stars with constant brightness is performed within the PanSTARRS[11] catalog. For this, we filter stars within the FoV of all instruments that have a relatively large amount of observations available (typically $N \geqslant 10$), with a standard deviation of their aperture SDSS gri magnitudes <0.01. In order to use as calibrators stars that do not have the risk of saturating the images, we also restrict the search to targets with magnitudes between 13 and 18, typically. With these considerations, we retrieve the gri aperture magnitudes of nonvariable stars that will be used as calibrators. Due to the different photometric system filter used by the PanSTARRS database (based on *grizy* filters) and that from the instruments implemented in IOP4 (generally equipped with standard Johnson–Cousins filters) a conversion between photometric systems is needed. Three transformations are currently implemented in IOP4: Jester et al. (2005), Jordi et al. (2006), and Lupton (2005). As explained by these authors, these transformations are suitable for stars, with the caveat of Jester et al. (2005) being only suitable for stars with $R_C - I_C < 1.15$. All three transformations have been found to be compatible for the calibrators added following this procedure. IOP4 implements by default the transformations from Lupton (2005).

6. Current Instruments

Five instruments from three different telescopes are already implemented in IOP4: RoperT90, AndorT90 and DIPOL (at OSN 0.9 m telescope), AndorT150 (OSN 1.5 m telescope), and CAFOS (CAHA 2.2 m telescope). The modular design of IOP4 allows easily integrating new instruments and observatories. The `Telescope` base class provides the skeleton over which to implement new telescopes. It implements the necessary methods to query, download (e.g., through the `FTPArchiveMixin` class), and perform the initial classification of observing epochs.

Integrating a new instrument is as easy as subclassing the `Instrument` subclass. The new subclass must provide the necessary information to identify the instrument, and possibly implement methods to translate nonstandard keywords (if any), extract position and size hints for astrometry, and override existing reduction procedures or implement new ones.

[9] http://astrometry.net/doc/readme.html

[10] https://tabulator.info/

[11] https://catalogs.mast.stsci.edu/panstarrs/

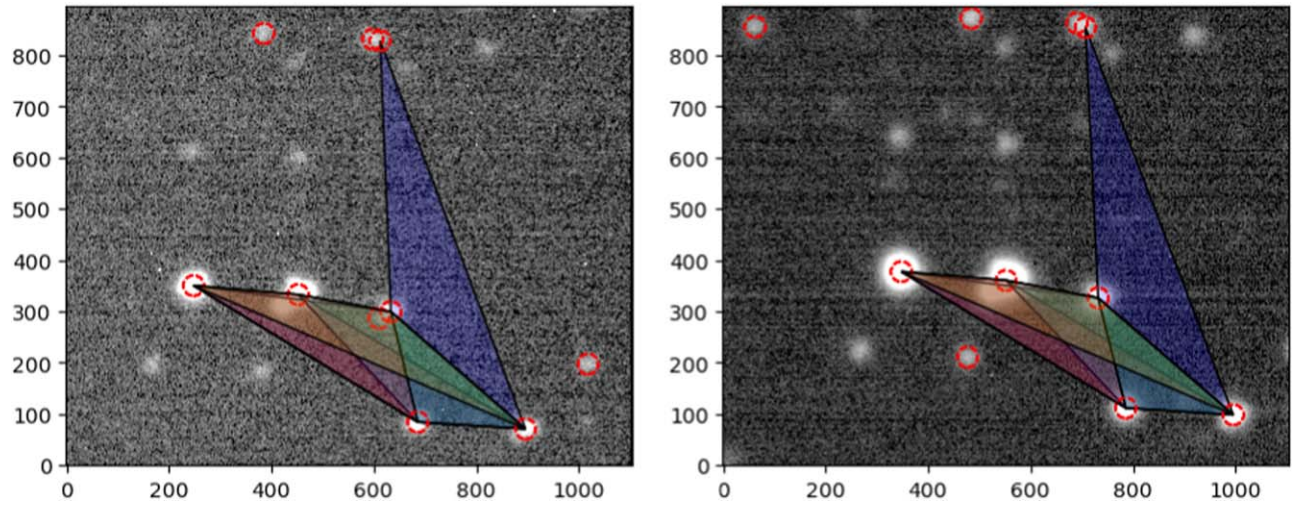

Figure 6. DIPOL polarimetry (left) and photometry (right) frames of BL Lacertae observations. Only the shown subframe ($2\farcm5 \times 2\farcm0$) is usually saved to disk during polarimetric observations with DIPOL, which prevents blind solving of the image. The full photometry field ($9\farcm2 \times 6\farcm3$) is saved, however, and the correspondence between both images can be found by comparing quads of sources through a hashing algorithm. This allows us to automatically distinguish the O and E images of our target source in the field.

6.1. RoperT90, AndorT90, AndorT150

IOP4 implements data reduction from the current CCD (Andor ikon-L) cameras at the 0.9 and 1.5 m telescopes in OSN. It also implements the old Roper (VersArray) cameras (which were installed until 2021 October and 2018 July, respectively). The AndorT90 instrument mounted at the Nasmyth east focus of the T090 telescope has a $13.2' \times 13.2'$ field of view and pixel size of $0\farcs387$ px^{-1}. The AndorT150 instrument at the Nasmyth west focus of the T150 has similar characteristics, with a $7\farcm92 \times 7\farcm92$ field of view and pixel size of $0\farcs232$ px^{-1}. They are cooled down down to -80 °C without need of liquid nitrogen, and can be further cooled down to -100 °C using liquid refrigerant. The low temperatures make the use of dark current calibration frames unnecessary. Apart from the usual band filter wheels, a polarized filter wheel is available at the T090 and the T150 that allows polarimetry measurements by taking series of four images at varying polarized angles in 45 deg steps. From these, the total flux and raw Stokes parameters are computed as

$$F = \langle f \rangle = \frac{1}{4}\sum_i f_i \tag{1}$$

and

$$q_{\rm raw} = \frac{f_0 - f_{90}}{f_0 + f_{90}} \tag{2}$$

$$u_{\rm raw} = \frac{f_{45} - f_{-45}}{f_{45} + f_{-45}}, \tag{3}$$

where f_i are the fluxes at each rotator angle. The instrumental polarization is corrected by applying an offset

$$q_c = q_{\rm raw} - q_{\rm inst} \tag{4}$$

$$u_c = u_{\rm raw} - u_{\rm inst} \tag{5}$$

and a rotation

$$q = q_c \cos(2\chi_{\rm inst}) - u_c \sin(2\chi_{\rm inst}) \tag{6}$$

$$u = u_c \sin(2\chi_{\rm inst}) + u_c \cos(2\chi_{\rm inst}), \tag{7}$$

from which the linear polarization degree and polarization angle are computed as

$$p = \sqrt{q^2 + u^2} \tag{8}$$

$$\chi = \frac{1}{2}\arctan(u, q). \tag{9}$$

6.2. CAFOS

The Calar Alto Faint Object Spectrograph (CAFOS) instrument, mounted on the 2.2 m telescope at Calar Alto Observatory, provides imaging polarimetry capabilities. It is equipped with a Wollaston prism and a rotable $\lambda/2$ retarder plate that provides two polarized images separated $18''$. Only a 800×800 subframe of the full 2048×2048 CCD chip is typically used, with a field of view of $34' \times 34'$ and a pixel size of $0\farcs530$ px^{-1}. The CCD chip is cooled to temperatures lower than -100 °C, making the dark current negligible. Several filters are available, while polarimetry observations are usually taken in Johnson R. The ordinary (O) and extraordinary (E) images at each angle are used to compute the total flux and the reduced Stokes parameters as (Zapatero Osorio et al. 2005)

$$F = \frac{1}{N}\sum_i \frac{f_{E,i} + f_{O,i}}{2} \tag{10}$$

and

$$R_Q = \sqrt{\frac{f_{O,0}/f_{E,0}}{f_{O,45}/f_{E,45}}} \tag{11}$$

$$R_U = \sqrt{\frac{f_{O,22}/f_{E,22}}{f_{O,67}/f_{E,67}}}, \tag{12}$$

where $f_{O,i}$ and $f_{E,i}$ are the fluxes of the ordinary and extraordinary images of the source at each angle i. From these, the Stokes parameters are reconstructed as

$$Q_I = \frac{R_Q - 1}{R_Q + 1} \tag{13}$$

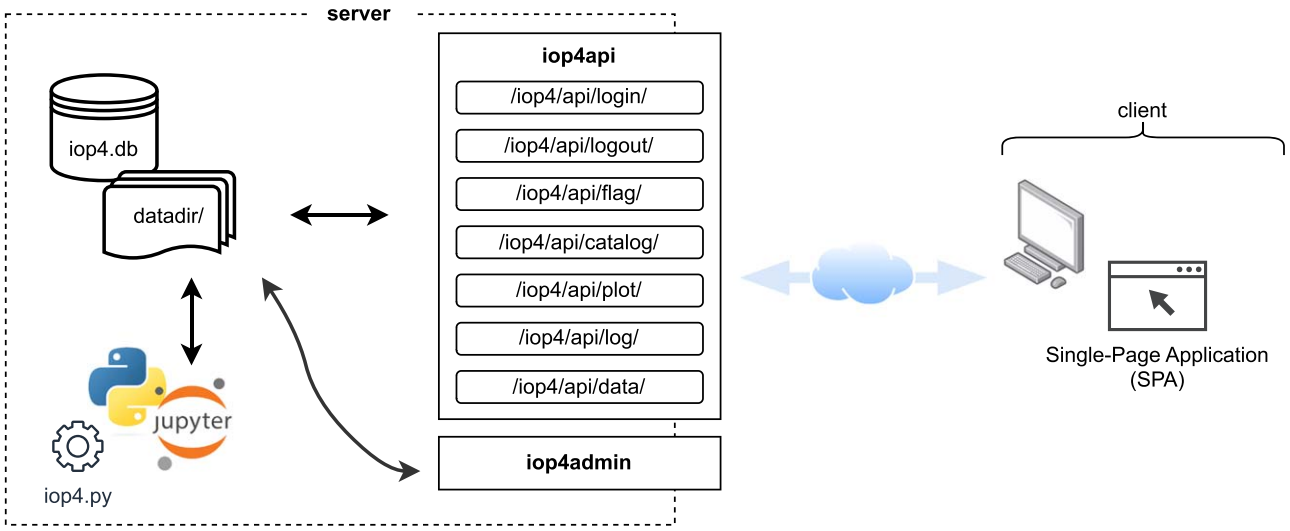

Figure 7. Different ways of interacting with IOP4. The pipeline (through the IOP4 command, Python script or Jupyter notebooks) can directly interact with the data aided by IOP4LIB. The IOP4API application exposes a series of API end points that allow the client to authenticate itself, query, plot and flag data, and explore catalog and logs from a web browser through a single-page application (SPA). The IOP4ADMIN site independently provides a way to interact and edit all models in the database.

$$U_I = \frac{R_U - 1}{R_U + 1}, \tag{14}$$

and the polarization degree and polarization angles from Equations (8) and (9). While instrumental polarization is negligible for CAFOS, the polarization angle still needs to be corrected according to

$$\chi = \chi - \mathrm{CPA}, \tag{15}$$

where CPA is the zero polarization angle.

6.3. DIPOL

The DIPOL-1 polarimeter is thoroughly described in Piirola et al. (2020) and Otero-Santos et al. (2024). It is based on a $\lambda/2$ retarder plate attached to a rotator and a high readout speed CMOS camera. Installed at the OSN-T090 the instrument has a field of view of $9\farcm2 \times 6\farcm3$ and a pixel scale of $0.134''$ px^{-1}. Cycles of 16 images are typically taken with varying rotator angles at $22\fdg5$ steps. For polarimetry observations, usually only a subframe of size $2\farcm5 \times 2\farcm0$ is saved, greatly reducing used disk space. Dark current calibration frames are necessary. High throughput sharp cutoff R, G, and B filters are available. The computation of the polarimetric results is done following Patat & Romaniello (2006). The Stokes parameters are computed as

$$Q_{\mathrm{raw}} = \frac{2}{N}\sum_i F_i \cos\left(\frac{\pi}{2}i\right) \tag{16}$$

$$U_{\mathrm{raw}} = \frac{2}{N}\sum_i F_i \sin\left(\frac{\pi}{2}i\right), \tag{17}$$

where the coefficients F_i are computed as

$$F_i = (F_{O,i} - F_{E,i})/(F_{O,i} + F_{E,i}), \tag{18}$$

$F_{O,i}$ and $F_{E,i}$ being the fluxes of the ordinary and extraordinary images of the source for each rotator angle i. The instrumental polarization is corrected with an offset

$$Q = Q_{\mathrm{raw}} - Q_{\mathrm{inst}} \tag{19}$$

$$U = U_{\mathrm{raw}} - U_{\mathrm{inst}}. \tag{20}$$

Then, p and χ are obtained per Equations (8), (9) and (15). Photometric (full frame) observations are also performed. The conversion of the magnitudes in the Baader RGB filters to standard Johnson–Cousins $UBVR_cI_c$ will be treated in a separate paper (in preparation).

7. Development and CI

IOP4 is open-source, its code hosted at https://github.com/juanep97/iop4. The repository contains all the necessary code to run IOP4 both as a program (`iop4` script) and as a library (IOP4LIB), plus the IOP4API Django application and the customized IOP4ADMIN site. Both can be readily used through the Django debug server and the provided IOP4SITE, or they can be integrated into another site and deployed for public use (see Serving IOP4 in production in IOP4 documentation). To ease the development process and the maintainability of the code, we implemented Continuous Integration and Continuous Deployment (CI/CD) workflows to perform automatic testing, build the documentation and facilitate the delivery of IOP4. The source code includes a test suit, the test data set is freely available at https://vhega.iaa.es/iop4/. The test procedure includes checking the reliability of the reduction and calibration procedures from raw data, and ensuring the quality of the photo-polarimetric results by comparison against tabulated values. The size of the astrometry.net solver index files makes inviable using GitHub-hosted ordinary runners for CI. Instead, a self-hosted runner provided by the VHEGA[12] group with self-provisioning capabilities using GARM[13] automatically runs tests on pull requests and merge commits to the main branch on isolated containers that already provide the test data set and astrometry index files. Anyone can run the test locally in their computers using PYTEST.[14] As of version `v1.2.0` (Escudero Pedrosa et al. 2024), test coverage for IOP4LIB is over 65%.

[12] https://vhega.iaa.es/
[13] https://github.com/cloudbase/garm
[14] https://docs.pytest.org/

The documentation can be built using SPHINX,[15] and contains several notebook examples. MYST-NB[16] and JUPYTEXT[17] allow using the percent format for the notebooks, an human-readable format easily integrable with version control software. The example notebooks are automatically run when building the documentation and use the test data set. The documentation is also automatically built and deployed to GitHub Pages as part of the CI. New releases of IOP4 are automatically deployed to the public PyPi software repository.[18]

8. Conclusions

IOP4 is an open-source, interactive photo-polarimetric pipeline written in Python. Its results have already featured in a number of refereed and high impact publications (e.g., Middei et al. 2023a, 2023b; Di Gesu et al. 2023; Ehlert et al. 2023; Marshall et al. 2023; Peirson et al. 2023; Kim et al. 2024; Otero-Santos et al. 2024; among others), following the footsteps of its predecessor IOP3[19] (e.g., Di Gesu et al. 2022; Liodakis et al. 2022, etc), and has provided data to many ongoing studies. Most of this data comes from the MAPCAT (Agudo et al. 2012) and TOP-MAPCAT programs, having reduced more than 600 GB of data from the these programs as of 2024 June. The former was running at Calar Alto from 2007 to 2018, the latter has been running from 2018 to the present day both at OSN and CAHA. Both programs are focused on the photo-polarimetric monitoring of blazars, combining regular observations with targets of opportunity. Aided by its parallel processing capabilities, it routinely downloads, reduces and serves results from a full night of observation of these programs in less than half an hour. Its speed makes it suitable to be used as a real-time analysis tool. Moreover, IOP4 has contributed to the publication of several Astronomer Telegrams (such as Otero-Santos et al. 2023a, 2023b) thanks to the promptness of its results and the ease of use.

Development of IOP4 is ongoing. Future releases of IOP4 might include several other instruments and reduction methods. As of version `v1.2.0`, IOP4 provides a prefabricated night summary script that sends the results of observations to subscribed users. Future versions might include a more advanced alert and trigger system. In any case, the IOP4LIB already allows the users to create their own scripts, for alerts or any other purpose.

The project welcomes any interested user to participate in IOP4 development and request or contribute to the implementation of new instruments and features.

Acknowledgments

The IAA-CSIC team acknowledges financial support from the Spanish "Ministerio de Ciencia e Innovación" (MCIN/AEI/ 10.13039/501100011033) through the Center of Excellence Severo Ochoa award for the Instituto de Astrofísica de Andalucía-CSIC (CEX2021-001131-S), and through grants PID2019-107847RB-C44 and PID2022-139117NB-C44. P.S-S. acknowledges financial support from the Spanish I+D+i project PID2022-139555NB-I00 (TNO-JWST) funded by MCIN/AEI/10.13039/501100011033. Based on observations made at the Sierra Nevada Observatory (OSN), operated by the Instituto de Astrofísica de Andalucía (IAA-CSIC), and at the Centro Astronómico Hispano-Alemán (CAHA), operated jointly by Junta de Andalucía and the IAA-CSIC. Development of this software would not have been possible without the invaluable and selfless contributions of the open-source community.

Facilities: Instituto de Astrofísica de Andalucía (IAA-CSIC), Observatorio de Sierra Nevada (OSN), Observatorio de Calar Alto (CAHA).

Software: Django (Django Software Foundation 2023), SQLite (SQLite Developers 2023), Vue.js (Vue Developers 2023), numpy (Harris et al. 2020), astropy (Astropy Collaboration et al. 2013, 2018, 2022), photutils (Bradley et al. 2023), astrometry.net (Lang et al. 2010).

ORCID iDs

Juan Escudero Pedrosa https://orcid.org/0000-0002-4131-655X
Iván Agudo https://orcid.org/0000-0002-3777-6182
Daniel Morcuende https://orcid.org/0000-0001-9400-0922
Jorge Otero-Santos https://orcid.org/0000-0002-4241-5875
Giacomo Bonnoli https://orcid.org/0000-0003-2464-9077
Vilppu Piirola https://orcid.org/0000-0003-0186-206X
César Husillos https://orcid.org/0000-0001-8286-5443
Rubén López-Coto https://orcid.org/0000-0002-3882-9477
Pablo Santos-Sanz https://orcid.org/0000-0002-1123-983X

References

Agudo, I., Molina, S. N., Gómez, J. L., et al. 2012, IJMPS, 8, 299
Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., et al. 2022, ApJ, 935, 167
Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 123
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Bokeh Development Team 2023, Bokeh: Python Library for Interactive Visualization, https://bokeh.org/
Bradley, L., Sipőcz, B., Robitaille, T., et al. 2023, astropy/photutils: v1.8.0, Zenodo, doi:10.5281/zenodo.7946442
Di Gesu, L., Donnarumma, I., Tavecchio, F., et al. 2022, ApJL, 938, L7
Di Gesu, L., Marshall, H. L., Ehlert, S. R., et al. 2023, NatAs, 7, 1245
Django Software Foundation 2023, Django, 4.2., https://www.djangoproject.com
Ehlert, S. R., Liodakis, I., Middei, R., et al. 2023, ApJ, 959, 61
Escudero Pedrosa, J., Morcuende Parrilla, D., & Otero-Santos, J. 2024, IOP4, v1.2.0, Zenodo, doi:10.5281/zenodo.11548299
Greisen, E. W., & Calabretta, M. R. 2002, A&A, 395, 1061
Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Natur, 585, 357
Husillos, C., Bernardos, M., Agudo, I., Bonnoli, G., & Escudero Pedrosa, J. 2021, cesarhusrod/iop3: v1.0.0, Zenodo, doi:10.5281/zenodo.10995575
Jester, S., Schneider, D. P., Richards, G. T., et al. 2005, AJ, 130, 873
Jordi, K., Grebel, E. K., & Ammon, K. 2006, A&A, 460, 339
Kim, D. E., Di Gesu, L., Liodakis, I., et al. 2024, A&A, 681, A12
Lang, D., Hogg, D. W., Mierle, K., Blanton, M., & Roweis, S. 2010, AJ, 139, 1782
Liodakis, I., Marscher, A. P., Agudo, I., et al. 2022, Natur, 611, 677
Lupton, R. 2005, Transformations between SDSS magnitudes and other systems—SDSS-III, https://www.sdss3.org/dr8/algorithms/sdssUBVRITransform.php#Lupton2005
Marshall, H. L., Liodakis, I., Marscher, A. P., et al. 2023, arXiv:2310.11510
Middei, R., Liodakis, I., Perri, M., et al. 2023a, ApJL, 942, L10
Middei, R., Perri, M., Puccetti, S., et al. 2023b, ApJL, 953, L28
Nilsson, K., Pasanen, M., Takalo, L. O., et al. 2007, A&A, 475, 199
Otero-Santos, J., Piirola, V., Escudero, J., et al. 2023a, ATel, 16305, 1
Otero-Santos, J., Piirola, V., Escudero Pedrosa, J., et al. 2024, AJ, 167, 137
Otero-Santos, J., Piirola, V., Pacciani, L., et al. 2023b, ATel, 16360, 1
Patat, F., & Romaniello, M. 2006, PASP, 118, 146

[15] https://www.sphinx-doc.org/
[16] https://myst-nb.readthedocs.io/
[17] https://jupytext.readthedocs.io/
[18] https://pypi.org/project/iop4/
[19] Husillos et al. (2021).

Peirson, A. L., Negro, M., Liodakis, I., et al. 2023, ApJL, 948, L25
Pence, W. D., Chiappetti, L., Page, C. G., Shaw, R. A., & Stobie, E. 2010, A&A, 524, A42
Piirola, V., Berdyugin, A., Frisch, P. C., et al. 2020, A&A, 635, A46
SQLite Developers 2023, SQLitehttps://www.sqlite.org/
Vue Developers 2023, Vue, 3.4.9., https://vuejs.org/
Zapatero Osorio, M. R., Caballero, J. A., & Béjar, V. J. S. 2005, ApJ, 621, 445