



Testing the LSST Difference Image Analysis Pipeline Using Synthetic Source Injection Analysis

S. Liu¹ , W. M. Wood-Vasey¹ , R. Armstrong² , G. Narayan^{3,4} , and B. O. Sánchez⁵
The Dark Energy Science Collaboration

¹ Pittsburgh Particle Physics, Astrophysics, and Cosmology Center (PITTPACC), Physics and Astronomy Department, University of Pittsburgh, 3941 O'Hara St., Pittsburgh, PA 15260, USA

² Lawrence Livermore National Laboratory, 7000 East Ave., Livermore, CA 94550-9234, USA

³ Department of Astronomy, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

⁴ Center for AstroPhysical Surveys (CAPS), National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

⁵ Aix Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France

Received 2023 November 27; revised 2024 March 17; accepted 2024 March 19; published 2024 May 13

Abstract

We evaluate the performance of the Legacy Survey of Space and Time Science Pipelines Difference Image Analysis (DIA) on simulated images. By adding synthetic sources to galaxies on images, we trace the recovery of injected synthetic sources to evaluate the pipeline on images from the Dark Energy Science Collaboration Data Challenge 2. The pipeline performs well, with efficiency and flux accuracy consistent with the signal-to-noise ratio of the input images. We explore different spatial degrees of freedom for the Alard–Lupton polynomial-Gaussian image subtraction kernel and analyze for trade-offs in efficiency versus artifact rate. Increasing the kernel spatial degrees of freedom reduces the artifact rate without loss of efficiency. The flux measurements with different kernel spatial degrees of freedom are consistent. We also here provide a set of DIA flags that substantially filter out artifacts from the DIA source table. We explore the morphology and possible origins of the observed remaining subtraction artifacts and suggest that given the complexity of these artifact origins, a convolution kernel with a set of flexible bases with spatial variation may be needed to yield further improvements.

Unified Astronomy Thesaurus concepts: [Transient detection \(1957\)](#); [Supernovae \(1668\)](#); [Type Ia supernovae \(1728\)](#)

1. Introduction

The Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST; Ivezić et al. 2019) will survey the southern sky from 2025 to 2035. This wide-field ground-based system uses an 8.4 m (6.5 m effective) telescope with a 3.2 gigapixel camera and a 9.6 deg² field of view. The LSST survey will repeatedly observe ∼18,000 deg² of the sky in the *ugrizy* filters during its 10 yr of operations.

The LSST data will be useful for a wide variety of science (LSST Science Collaboration et al. 2009), with guiding areas of probing dark energy and dark matter, studying the transient optical sky, producing an inventory of the solar system, and mapping the Milky Way. Achieving these scientific goals requires the discovery of transient sources and accurate photometric measurements. The LSST is expected to discover about 10⁶ transient sources per night (Ridgway et al. 2014; Graham et al. 2019; Ivezić et al. 2019), which requires a detection pipeline with high efficiency and a low rate of nonastrophysical artifacts.

The Rubin Observatory LSST Dark Energy Science Collaboration (DESC) developed the `dia_pipe`⁶ package to run

the LSST Science Pipelines⁷ Difference Image Analysis (DIA) algorithms. The `dia_pipe` combines different parts of the LSST Science Pipelines difference imaging code into a single pipeline, which enables the user to perform image difference, source association, and forced photometry. The DESC `dia_pipe` has been integrated with the LSST Data Release Production pipeline.

The LSST DESC Data Challenge 2 (DC2; LSST Dark Energy Science Collaboration (LSST DESC) et al. 2021) was a modeling and simulation effort that generated simulated LSST images. From a simulated Universe with galaxies, stars, and Type Ia supernovae (SNe Ia), DC2 used GalSim (Rowe et al. 2015) to render astronomical objects into a set of images for 5 yr at 300 deg² of the sky. These images were processed through the LSST Science Pipelines into data products modeled on those to be delivered by the Rubin Observatory. Sánchez et al. (2022) compared the DIA performance metrics to the Dark Energy Survey DIA performance (Kessler et al. 2015) using a 15 deg² subset of DC2 data, based on the SNe Ia that had been included in the DC2 simulation, and found similar performance to the DESC survey. Here, we perform a more in-depth evaluation of DIA using additional synthetic source injection on a subset of the DC2 data set. Our goal is to connect and explore the DIA algorithmic concepts and code as implemented in the LSST Science Pipelines.

In this work, we inject a controlled set of synthetic sources onto DC2 images to (1) evaluate the pipeline at different source magnitude levels and host magnitude levels, (2) compare the

⁶ https://github.com/LSSTDESC/dia_pipe

⁷ <https://github.com/lsst/>



pipeline performance with different kernel spatial orders, and (3) determine a set of flag-based cuts to removing false-positive detections (“artifacts”).

The essential strategy of the synthetic source injection technique is to add synthetic sources onto astronomical images based on the measured point-spread function (PSF) and monitor the detection status and flux measurements of these synthetic sources.

Traditionally, the use of DIA algorithms in surveys has suffered from significant subtraction artifacts, leading to the need for human-based or machine-learning-based classification (Kaiser et al. 2002; Frieman et al. 2008; Sako et al. 2008; Bernstein et al. 2012; Kessler et al. 2015). We explore the possible origins of these artifacts using difference images produced by the DIA pipeline. We study both the statistical properties of artifact measurements and the morphology of artifacts. A few possible causes are discussed, and we give suggestions for future improvements.

The outline of this work is as follows. Section 2 introduces image difference algorithms, Section 3 introduces the analysis framework of the synthetic source injection technique, Section 4 presents the analysis results of the DIA pipeline, Section 5 discusses the morphology and potential origin of subtraction artifacts, Section 6 discusses the current status of the DIA pipeline and possible future improvements, and we provide conclusions and outline future directions in Section 7.

2. Difference Image Algorithms

To begin DIA, we prepare two images that cover the same region of the sky with a time separation longer than a typical light-curve duration. One image, which usually was taken in good seeing conditions and has a high signal-to-noise ratio (S/N) for the magnitude range of interest, is designated as the template image. The other image is called the science image. Typically, the template image was taken months to years before the science image and may be a coaddition of many images to reduce the photon noise. In this analysis, we use coadded template images. The science image is typically from the current night, where we search for new transient sources or variations in the brightness or position of existing sources.

The next step is to match both images to the same seeing condition and the same astrometric reference so that we can subtract one image from the other to obtain a difference image that ideally represents the astrophysical differences between them. Source detection and photometric measurements are then performed on the difference image to produce a detection catalog. The LSST Science Pipelines version used in this work provides both the Alard–Lupton (AL; Alard & Lupton 1998) and the Zackay–Ofek–Gal–Yam (ZOGY; Zackay et al. 2016) algorithms to perform image differencing. The AL algorithm assumes that the PSFs of the science image (S) and template image (T) can be matched using a kernel κ . This process is done empirically by matching common sources in both images and does not explicitly use calculated PSFs for the images. The kernel κ is solved by minimizing the square sum of pixels from the difference image. The ZOGY algorithm assumes explicit knowledge of the PSF in each image and uses the Fourier transform between them to achieve the same seeing conditions of T and S and produce a detection image highlighting the astrophysical sources. A decorrelation method is also introduced in the ZOGY algorithm to perform decorrelation of the difference image.

2.1. AL Algorithm

The AL algorithm solves for the matching kernel κ , which uses stars in both images to match the PSF of the template image to the PSF of the science image. Galaxies can be used in addition to stars, although a galaxy provides less information than a star of the same flux level. Thus, this process is not dependent on reliable star–galaxy separation. In the spatially varying (Alard & Lupton 1998; Alard 2000) situation, the matching kernel can be expressed as

$$\kappa(u, v, x, y) = \sum_{q=1}^{N_k} a_q(x, y) k_q(u, v). \quad (1)$$

Here N_k is the number of kernel bases, a_q is a set of fitted spatially varying coefficients, and k_q is a set of kernel bases. The most commonly used kernel bases are composed with a set of Gaussian functions modulated with a set of polynomials (Alard & Lupton 1998). Another type of kernel basis is composed with delta bases (Bramich 2008; Miller et al. 2008), which are “shape-free” delta functions defined for each kernel pixel coordinate.

Using the matching kernel κ , the difference image of the AL algorithm can be expressed as

$$D(x, y) = S(x, y) - T(x, y) \otimes \kappa(x, y) - B(x, y), \quad (2)$$

where $B(x, y)$ is the fitted sky background.

2.2. ZOGY Algorithm

The ZOGY algorithm was designed for the case where PSFs and flux scaling factors for T and S are properly estimated, and the sky backgrounds of both images have been properly subtracted.⁸ The difference image from the ZOGY algorithm in Fourier space can be expressed as

$$\hat{D} = \frac{F_t \hat{P}_t \hat{S} - F_s \hat{P}_s \hat{T}}{\sqrt{\sigma_s^2 F_t^2 |\hat{P}_t|^2 + \sigma_t^2 F_s^2 |\hat{P}_s|^2}}. \quad (3)$$

The equivalent PSF of the ZOGY difference image is

$$\hat{P}_D = \frac{F_t F_s \hat{P}_t \hat{P}_s}{F_D \sqrt{\sigma_s^2 F_t^2 |\hat{P}_t|^2 + \sigma_t^2 F_s^2 |\hat{P}_s|^2}}. \quad (4)$$

Here the hat symbol represents the Fourier transform of each quantity, P_t and P_s are the PSFs of T and S , F_t and F_s are the flux scaling factors of both images, σ_t and σ_s are background noise levels, and F_D is the normalization constant. The expression in the denominator of $\hat{D}$ plays the role of decorrelating the noise.

However, ZOGY provides no mechanism to include information about the spatial variation of PSFs. One approach to using a spatially varying ZOGY would be to split T and S into subregions within which the spatial variation of PSFs is much smaller (Zackay et al. 2016; Reiss 2017; Hu et al. 2022). Performing the ZOGY algorithm on each subregion is achievable at the expense of more CPU and more complicated bookkeeping for detections near and across the edges of subregions. Such a “cell-based” approach to templates and subtraction could fail if the PSFs are varying rapidly across the images.

⁸ The ZOGY paper uses N to denote the science image and R for the template image. We here use T and S consistently for both AL and ZOGY.

3. The DC2 Data Set and Analysis Framework

3.1. The DC2 Data Set

To prepare for analysis of LSST data, LSST DESC created the DC2 data set with an end-to-end approach (LSST Dark Energy Science Collaboration (LSST DESC) et al. 2021) to simulate a subset of the full LSST survey. This approach started with a large N -body simulation of galaxies in the Universe, applied LSST-like observations, simulated each exposure, ran the LSST Data Release Processing pipeline,⁹ and produced data products analogous to those that will be delivered by the Rubin Observatory. The DC2 observations include six optical bands (*ugrizy*) with coverage of 300 deg^2 in a wide-fast-deep area and 1 deg^2 in a deep drilling field. It simulates 5 yr of the LSST survey.

The DC2 data set provides two classes of imaging data. For single-frame processing, it provides calibrated exposures (calexps) with sky background subtraction. Each calexp image is a 4000×4072 pixel image array with $0''.2 \text{ pixel}^{-1}$ scale. It also has a source catalog (src) that contains measurements of sources in that visit. After producing calexps, multiple visits are coadded together to create deepCoadd exposures. The resampling grid of deepCoadds is defined with tracts and patches. Each tract contains 7×7 patches, and each patch contains 4100×4100 pixels with a scale of $0''.2 \text{ pixel}^{-1}$. A visual representation of the tract-and-patch structure can be found in Sánchez et al. (2022) or LSST Dark Energy Science Collaboration (LSST DESC) et al. (2021).

3.2. Template Exposure

We focus on *i*-band exposures of the DC2 data set for our analysis. We choose “deepCoadd” exposures with seven patches as template exposures. The total area analyzed in this paper is about 0.36 deg^2 .

3.3. Host Selection Criteria and Science Exposure

We undertook this work from the perspective of SN cosmology, and so we are interested in the detection and photometric accuracy of new point sources (SNe) on top of extended sources (galaxies). GCRCatalogs (Mao et al. 2018) is a DESC package that provides users with a consistent interface to access both the input truth catalogs and the output processed catalogs for the DC2 simulations. We use GCRCatalogs to extract a list of potential host galaxies from the input truth table with a range of apparent host magnitudes. Next, we measure the local surface brightness (SB) with $1''$ aperture photometry at the center of each galaxy in the simulated images. We sort the galaxy sample into $1 \text{ mag arcsec}^{-2}$ bins distributed from 20 to $25 \text{ mag arcsec}^{-2}$ with $1 \text{ mag arcsec}^{-2}$ bin widths. While in the real Universe, SNe occur throughout galaxies, what we really want to explore here is the dependence of SN detection on underlying SB. Thus, we place the SN at the centers of galaxies with different central SB.

Within each patch and each SB magnitude bin, we select a set of 20 hosts. Galaxies are chosen to satisfy two criteria: (1) each host center is at least 200 pixels away from the image boundaries in both the x - and y -directions, and (2) each host center is at least 100 pixels away from another host center. These criteria help us avoid subtractions near image edges and

injecting overlapping synthetic sources. Locations of hosts on a patch are shown in Figure 1.

After selecting the host set within each patch and each host SB magnitude bin, we match each host set to 10 calexps using the `tracts_mapping.sqlite3` database file of the DC2 data set. We inject synthetic sources onto each calexp and then use those as the science exposures for running the image difference algorithm. We define a calexp and its corresponding deepCoadd exposure as an “image pair.” In total, we have 70 image pairs. The LSST survey takes images at a range of spatial and rotational dithers; thus, a single calexp often does not fully cover all 20 hosts in each host set. We restrict our sample to calexps that cover at least 15 host galaxies.

3.4. Synthetic Source Injection Framework

Our synthetic source injection procedure is based on the `insertFakes` module of the `pipe_tasks`¹⁰ package of the LSST Science Pipelines. We convert the R.A., decl. of the host to the x , y coordinate on the image. We then simulate point sources based on the calexp PSF model at that location on the image. The synthetic source injection framework converts the magnitude to the instrumental flux and scales the normalized synthetic source model by the instrumental flux. In the single-frame processing, the instrumental flux to magnitude calibration of the images was calculated with respect to a 12 pixel radius aperture. While the PSF model is finite in size, it is always larger than the calibration radius. The PSF model is normalized to sum to 1. We calculate the aperture correction between the full PSF and the smaller calibration radius and scale the model of the PSF by this correction. We then multiply the scaled PSF by the instrumental flux from the calibration to inject a source that has the correct photoelectron contribution for its magnitude. The scaled model is added to the image plane of the calexp without making any changes to its variance plane, which means we ignore the Poisson noise of the synthetic source.

After finishing the injection stage, we run the DIA pipeline on the injected data set using the `imageDifferenceDriver.py` of the `dia_pipe` package with v23.0.0¹¹ of the LSST Science Pipelines. This stage produces difference images and then runs photometry measurement algorithms on that difference image to produce the DIA source (`diaSrc`) table, which contains source measurements of the detected source. We obtain positions of variable stars and active galactic nuclei (AGNs) from the truth tables in GCRCatalogs and match them to the `diaSrc` tables. All variables and AGNs that get matched are removed from the `diaSrc` tables. As a result, the detections from the `diaSrc` tables only have synthetic sources and artifacts.¹² Next, we match the known injection positions to the `diaSrc` tables to determine whether the injected synthetic sources are detected in the difference exposures. The `diaSrc` fluxes and the flag information of the detected sources are also extracted from the `diaSrc` tables for further analysis.

3.5. LSST Software

In this work, we use the v23.0.0 version of the LSST stack package to provide basic functionalities for synthetic source

⁹ <https://pipelines.lsst.io/>

¹⁰ https://github.com/lsst/pipe_tasks

¹¹ https://pipelines.lsst.io/releases/v23_0_0.html

¹² Strictly speaking, `diaSrc` tables also have simulated SNe. Given the rarity of the true SNe, classifying them as artifacts does not affect our statistics.

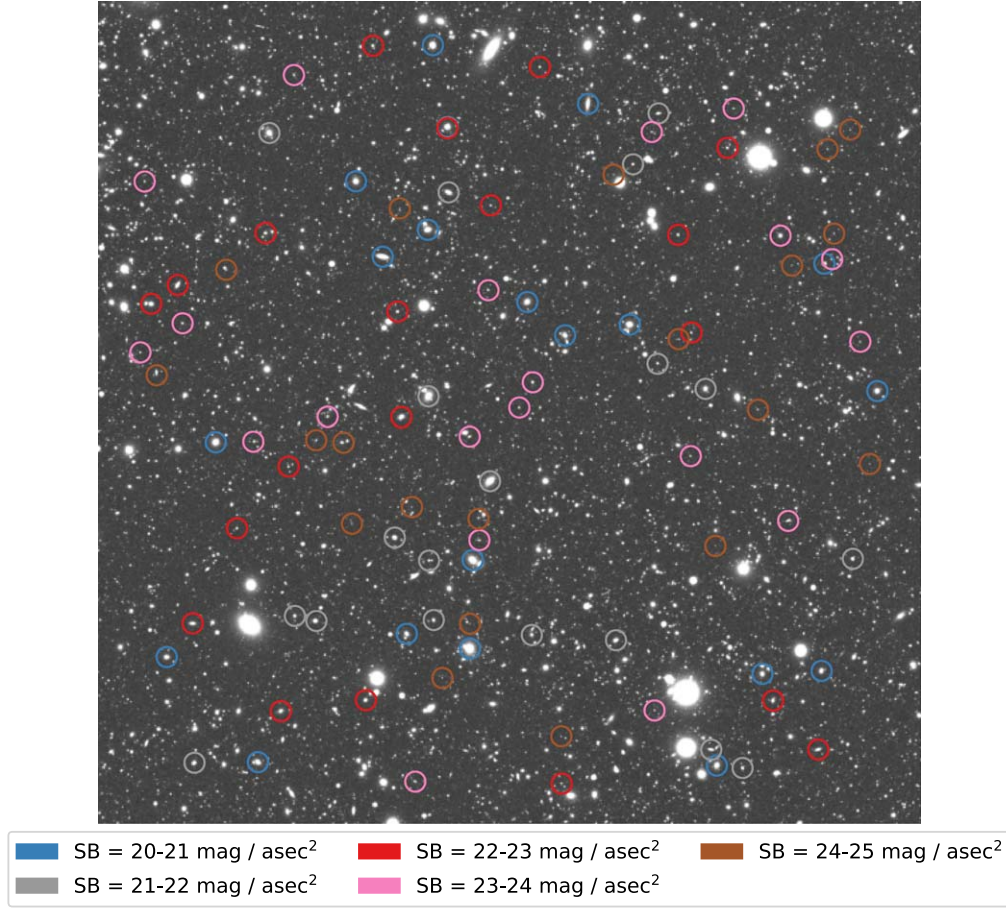


Figure 1. Locations of host galaxies on a patch (4100×4100 pixels). Hosts with different SB are indicated with different colors.

injection and image difference. We have documented the proper software versions in the GitHub repo LSSTDESC/dia_improvement,¹³ which includes scripts to reproduce our work.

4. DIA Pipeline Analysis

In this section, we describe performance metrics, injection strategy, and analysis results of the DIA pipeline. Different kernel spatial degree of freedoms have been implemented in the pipeline to reduce subtraction artifacts. We compare their performance here and identify the photometric flags that effectively select true positives with high efficiency and a low artifact rate.

4.1. Metrics

We evaluate the subtraction quality of the DIA pipeline using the following metrics. These metrics should reflect the detection ability and the photometric measurement accuracy of the pipeline.

1. *Efficiency*¹⁴ (*Eff*): the number of detected synthetic sources divided by the number of injected synthetic sources. We expect it to approach unity when synthetic sources are bright and decrease to 0 at the faint magnitude

end. The efficiency curve can be fitted using a sigmoid model. In this work, we use the function

$$\text{eff}(m) = \frac{1}{1 + e^{-a(m-b)}} \quad (5)$$

to fit our data, with a and b as fitting parameters and m as the synthetic source magnitude.¹⁵ Using this equation, we can solve the magnitude depth $m_{1/2}$, which corresponds to the synthetic source magnitude, where the efficiency drops to 50%. The magnitude depth $m_{1/2}$ is mathematically equivalent to the fitting parameter b .

2. *Artifact rate (AR)*: the number of subtraction artifacts per square degree.
3. *Flux pull* ($\Delta F/\sigma$):

$$\text{fluxpull} = (\text{measuredflux} - \text{injectedflux})/(\text{fluxerror}).$$

This flux pull should follow a normal distribution with a mean of 0 and a standard deviation of 1. In this analysis, we calculate the pull statistic of the diaSrc PSF flux.

4.2. Analysis Strategies

We apply two analysis strategies to sets of DC2 calexp. The first strategy is to focus on measuring efficiency as a function of source magnitude and background SB; we inject synthetic

¹³ https://github.com/LSSTDESC/dia_improvement

¹⁴ In this work, we estimate the detection efficiency using synthetic sources that are injected on host galaxies. A method that untangles the effects of Poisson noise versus subtraction artifacts can be found in Doctor et al. (2017).

¹⁵ This equation is correct for constant noise and PSF within an image. Our use across a set of images with some variation in noise and PSF is an approximation. A more robust relationship would be efficiency versus injected S/N.

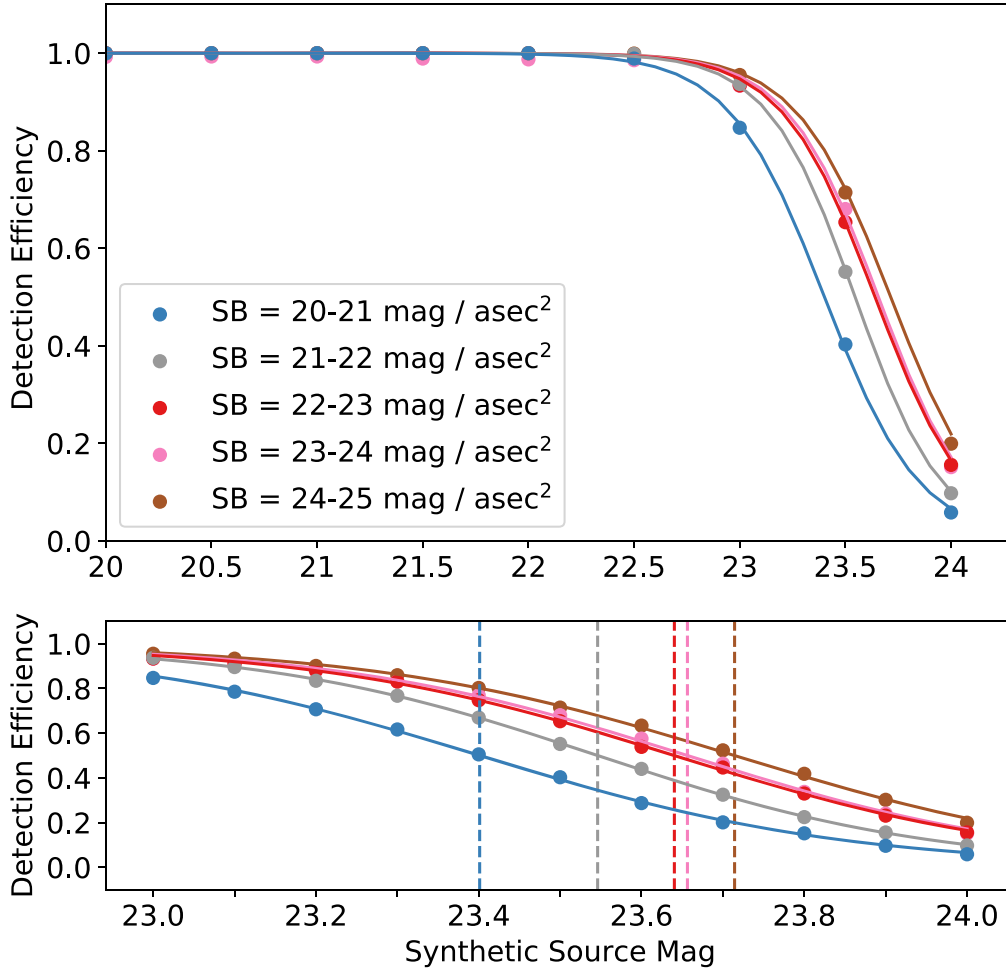


Figure 2. Detection efficiency vs. source mag (dots). Solid lines are sigmoid models fitted to the data. Different SB magnitudes are indicated using the colors shown in the legend. The upper plot corresponds to 20–24 synthetic source magnitudes with a 0.5 mag interval. The lower plot corresponds to 23–24 synthetic source magnitudes with a 0.1 mag interval. The $m_{1/2}$ for different host SBs are 23.40 (20–21 mag arcsec⁻²), 23.55 (21–22 mag arcsec⁻²), 23.64 (22–23 mag arcsec⁻²), 23.66 (23–24 mag arcsec⁻²), and 23.71 (24–25 mag arcsec⁻²). They are shown as vertical lines in the lower plot.

sources from 20–23 mag in 0.5 mag steps and 23–24 mag in 0.1 mag steps. These finer steps focus on the magnitude range where the efficiency drops rapidly. We use this strategy to determine the magnitude depth $m_{1/2}$ at which 50% of the sources are recovered.

For the second strategy, we undertook a related but separate campaign of studying efficiency and artifact rate for high signal-to-noise synthetic sources and low signal-to-noise synthetic sources with different subtraction configurations with two consistent sets of 20 mag (MAG20) sources and 23 mag (MAG23) sources.

4.3. Performance of the Default Configuration of the DIA Pipeline

To get a baseline evaluation of the performance of the DIA pipeline, we run `dia_pipe` using the default configuration of the pipeline. The default difference imaging algorithm is AL with a set of Gauss-polynomial bases. The AL kernel has three Gaussians with sigma values of 0.7, 1.5, and 3.0 pixels, with respective order 4, 2, and 2 polynomials in kernel (u, v) space. These kernel terms are allowed to spatially vary at first order in x, y across the image. The background level is allowed to vary spatially at second order in x, y .

The upper plot of Figure 2 shows our recovered detection efficiency as a function of synthetic source magnitude. The efficiency follows predicted smooth curve efficiency functions, with a decrease in efficiency as a function of both (1) decreasing injected flux and (2) increasing underlying SB (and thus background noise) at the location the synthetic source is placed. The effect of the underlying host galaxy light is most visible for the brightest hosts, SB = 20–21 mag arcsec⁻², which shows a substantial 0.25 mag $m_{1/2}$ shift in the efficiency curve. The DIA pipeline successfully recovers all¹⁶ synthetic sources brighter than 22 mag. The efficiency starts to drop below 1 around 22 mag, and it drops below 0.5 between 23 and 24 mag. In the lower plot of Figure 2, we focus on the detection efficiency at the faint magnitude (low S/N) limit. The predicted values of the $m_{1/2}$ of the fitted function are between 23.40 and 23.71 mag.

Figure 3 shows the S/N distribution of the detected synthetic sources when injected at 21, 22, 23, and 24 mag. At the synthetic source magnitude of 21, the S/N distribution is far above the detection limit, which means the pipeline should be able to recover all synthetic sources. At the injected magnitude of 23, the S/N distribution overlaps the detection threshold, and the efficiency begins to drop. When the synthetic source

¹⁶ Efficiency of 0.989.

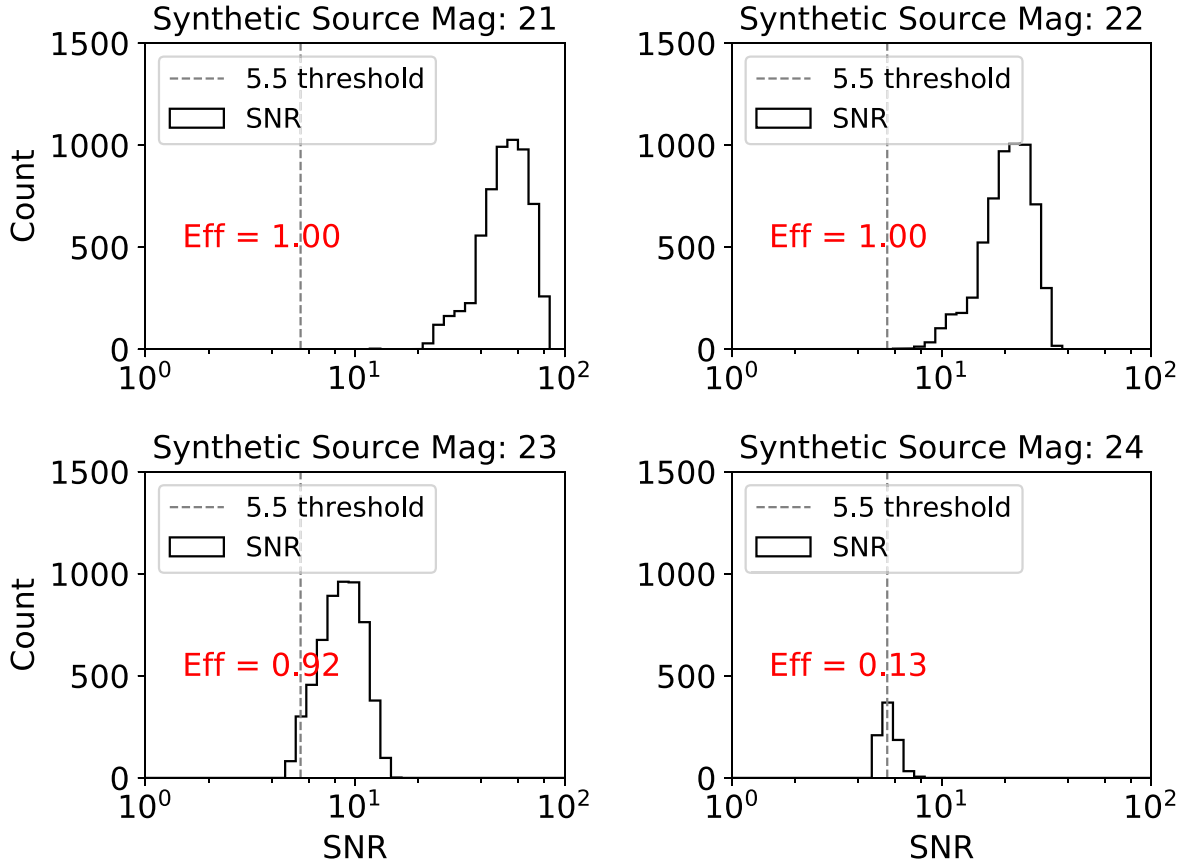


Figure 3. S/N distribution of detected sources with true mag listed above each panel. Injections from different host SB are binned together. The efficiency at each source magnitude is shown on each plot. The vertical line in each plot shows the $S/N = 5.5$ detection threshold. The S/N distribution is closer to the threshold for fainter synthetic sources.

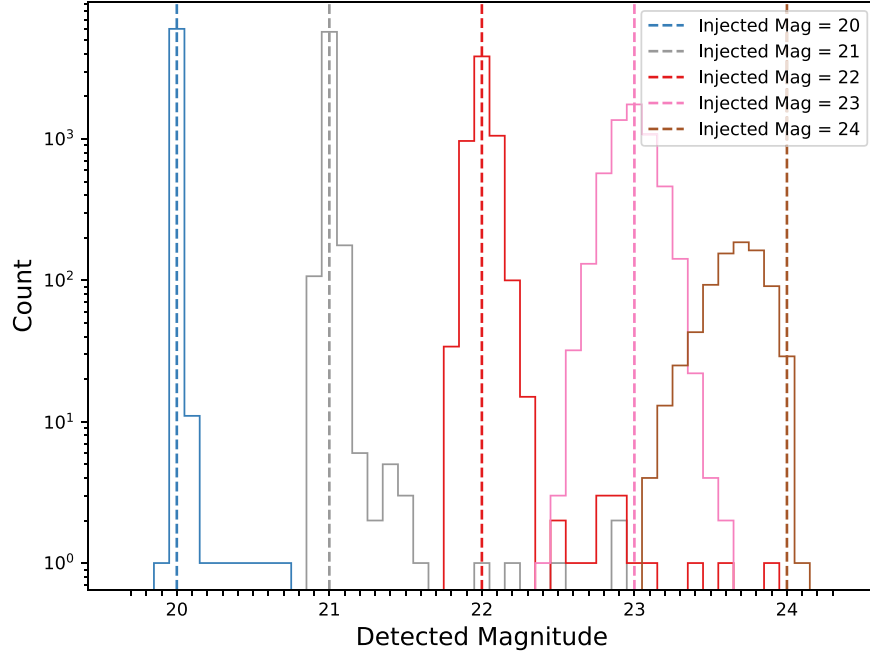


Figure 4. Distributions of recovered magnitudes (solid lines) of synthetic sources at different injected magnitudes (dashed lines). For injected magnitudes of (20, 21, 22, 23, 24), more than (99%, 99%, 90%, 57%, 8%) of the recovered flux measurements are within 0.1 mag of the injected magnitudes. At 24 mag, the distribution of recovered magnitudes is significantly biased toward the bright side due to Malmquist bias.

magnitude is 24, most injections are below the detection threshold, which explains why the efficiency is low at this magnitude.

Figure 4 shows the distributions of the detected magnitude and injected magnitude of synthetic sources. At the injected magnitude of (20, 21, 22, 23, 24), more than (99%, 99%, 90%, 57%, 90%,

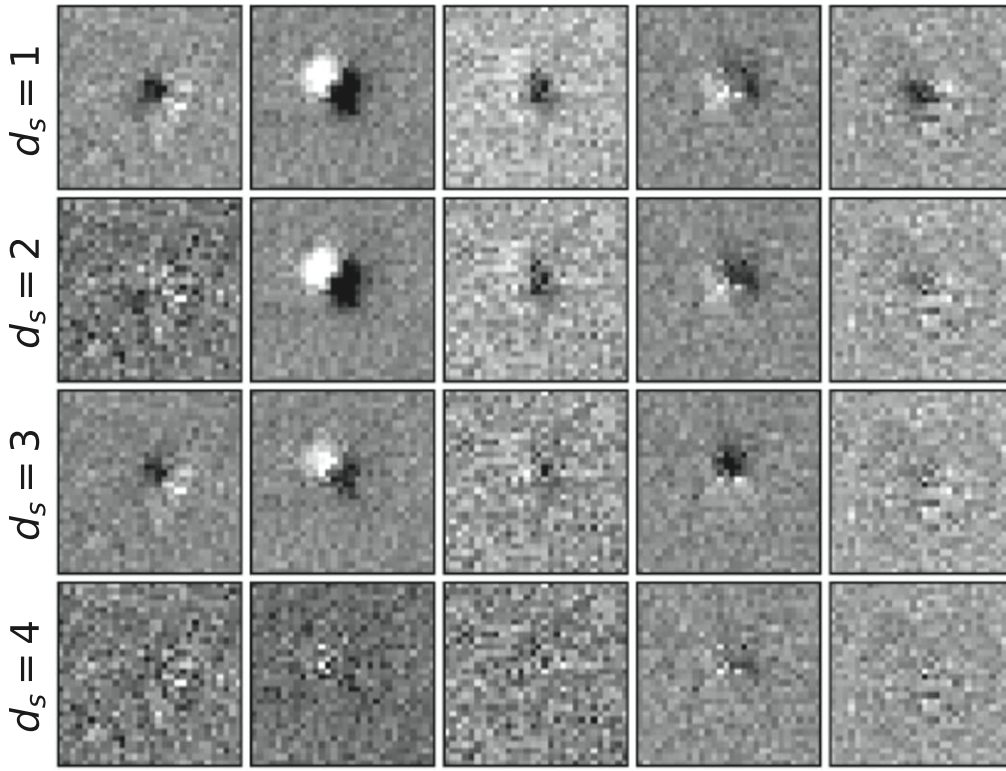


Figure 5. Postage stamps of difference images from different kernel spatial degrees of freedom (d_s). Each column shows one location on the subtraction image. Each row corresponds to a specific d_s .

57%, 8%) of the flux measurements are within 0.1 mag of the injected magnitudes. At 24 mag, the distribution is significantly biased toward the bright side due to the Malmquist bias (Malmquist 1922, 1925).

4.4. Comparison with Previous DESC DC2 Analysis

Here we give a comparison of the i -band $m_{1/2}$ to the previous DESC DC2 DIA (Sánchez et al. 2022). That work evaluated the $m_{1/2}$ of the LSST DIA pipeline using about 15 deg² of the DC2 data set with artificial SN Ia light curves overlaid onto exposures. The estimated $m_{1/2}$ in the i band is 23.50 mag for all sources.

In this work, we estimate the $m_{1/2}$ in the i band using about 0.36 deg² of the DC2 data set with carefully controlled synthetic point sources distributed across different host brightness. The fitted $m_{1/2}$ is between 23.40 mag and 23.71 mag, depending on the underlying SB.

4.5. Kernel Spatial Degree of Freedom

Alard & Lupton (1998) and Alard (2000) suggest expanding the AL kernel coefficients $a_q(x, y)$ (Section 2.1) as a function of a power series of positions. The highest-order d_s of the power series defines the kernel spatial degree of freedom. The implementation of the AL algorithm in the DIA pipeline can be configured to run with different values of d_s . In this section, we explore the pipeline performance with d_s varying from 1 to 4. The pipeline default value of d_s is 1. Our analysis was constrained to a maximum spatial order of $d_s = 4$ due to limitations in v23.0.0 of the LSST Science Pipelines, which have since been lifted. Raising d_s beyond 4 offers increased kernel flexibility, thereby potentially reducing the artifact rate.

However, this approach carries the risk of overfitting, since the number of spatial polynomials scales proportionally with d_s^2 (Alard 2000; Bramich et al. 2013). Additionally, it would result in an increase in computational time.

The primary goal of exploring the spatial degrees of freedom is to reduce artifacts while maintaining high efficiency and flux accuracy. We focus here on high-S/N sources at 20 mag (MAG20), which is far above the detection limit, and low-S/N sources at 23 mag (MAG23), which is close to the $m_{1/2}$. Since the comparison focuses on the region that is away from the image edge,¹⁷ it is possible that the high d_s s are performing worse near the image edge. More evaluations near the image boundary are still needed in future work.

In Table 1, we show the efficiency, artifact rate,¹⁸ mean, and standard deviation of the flux pull distribution for each spatial degree of freedom at MAG20 and MAG23. We find that the results from high S/N and low S/N are consistent. Up to $d_s = 4$, the artifact rate is a monotonically decreasing function of d_s . The fluctuation of the recovered mean pull of the flux of the injected sources is less than 0.020. The fluctuation of the standard deviation is less than 0.010. The fluctuation of the efficiency is less than 0.013.

In Figure 5, we show the postage stamps of difference images from different spatial degrees of freedom. Each column of stamps is made at the same position, while each row corresponds to a specific d_s . By comparing the stamps, we find that high d_s s, especially $d_s = 4$, can produce cleaner difference

¹⁷ Host galaxies are selected to be 200 pixels away from the image boundary.

¹⁸ To be consistent with Section 4.6, we use only detections from SB = 20–21 mag arcsec⁻² to calculate efficiency and artifact rate.

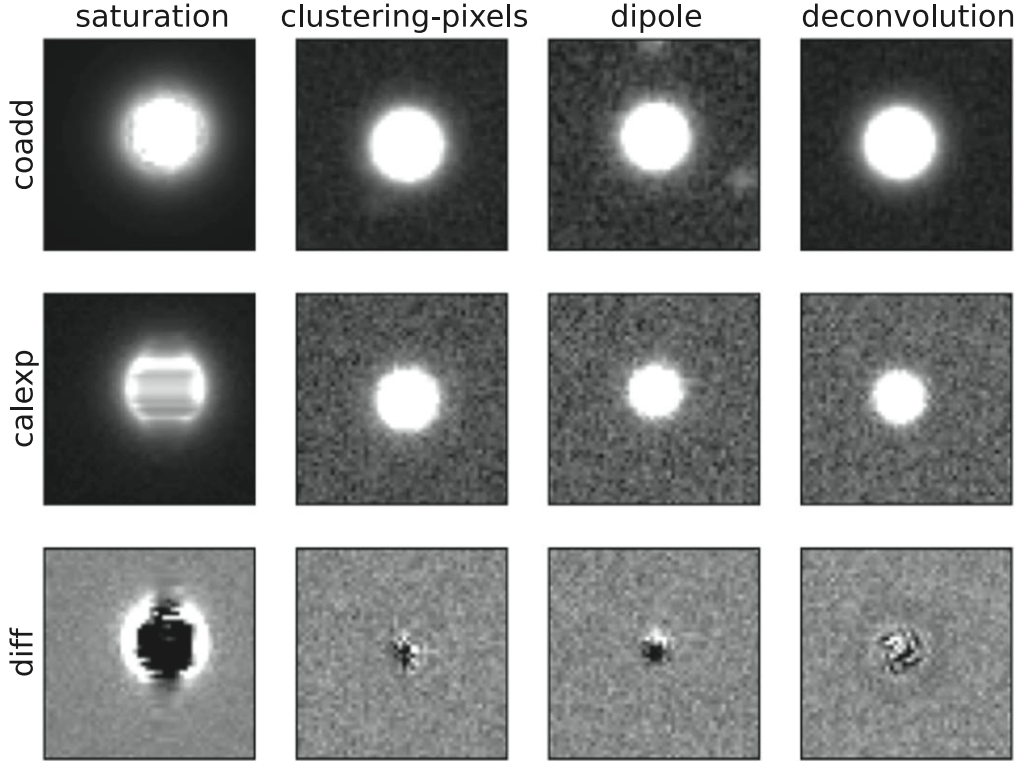


Figure 6. Artifacts commonly seen in difference images. Sources with saturated pixels (left) are easy to identify and marked with saturation flags in the diaSrc table. Clustering pixels (second from left) come from subtraction residuals of background sources or background noise. Dipoles (second from right) require measurements and fitting; many are successfully identified, but there is a continuum of less-offset and lower signal-to-noise dipoles that are hard to identify. The dipole artifact is a subclass of the clustering-pixels artifact. Deconvolution (right) leaves characteristic rings from trying to invert information that is not available in the larger PSF image.

Table 1
Flux Difference Pull Distribution, Efficiency, and Artifact Rate with Different Kernel Spatial Degree of Freedom

d_s	MAG20				MAG23			
	Mean	Std.	Eff.	Artifact Rate	Mean	Std.	Eff.	Artifact Rate
	$\Delta F/\sigma$			(deg ⁻²)	$\Delta F/\sigma$			(deg ⁻²)
1	0.090	1.232	1.000	695	0.096	1.000	0.847	700
2	0.087	1.228	1.000	689	0.090	0.999	0.840	693
3	0.092	1.237	1.000	650	0.088	1.011	0.843	651
4	0.106	1.232	1.000	612	0.092	1.006	0.852	608

Note. The mean and standard deviations are of the pull distribution of the difference between detected flux and injected flux ($\Delta F/\sigma$). We apply a 5σ clip to the pull distribution to remove outliers. At MAG20, the percentages of outliers are (0.249%, 0.232%, 0.182%, 0.265%) at ($d_s = 1, 2, 3, 4$). At MAG23, the percentages of outliers are (0.054%, 0.036%, 0.054%, 0.036%) at ($d_s = 1, 2, 3, 4$). The default spatial degree of freedom is $d_s = 1$.

images. As a result, the artifact rate decreases as the d_s increases.

4.6. Detection Flags

The LSST Science Pipelines are configured to detect any possible source and characterize it with flags to indicate detections that are not real astrophysical sources. These 109 flags identify pixel-level features such as saturation, bad pixels, and cosmic rays as well as characteristics of the source in the subtraction such as moment and dipole fits. We use these flags to reject artifacts, which we here define as detections that are not in circular regions with $0''.8$ radii of known synthetic sources. The data are from MAG20 and MAG23, with host SB between 20 and 21 mag arcsec⁻². In this synthetic source injection analysis, we split the detected sources from each diaSrc table into two tables: one for the detected synthetic

sources (MAG20: 1184 entries; MAG23: 1003 entries) and the other for detections that are artifacts (MAG20: 2446 entries; MAG23: 2464 entries).

We use the efficiency and artifact rate to examine the power of each flag to classify sources into real and artifact. We calculate these metrics for all flags and show the full table in Table 2. From this information in this table, we can select a small set of flags where (1) the efficiency of each flag is close to 1 at MAG20 and (2) each flag has a clear physical interpretation related to artifact classification. We document this “selected flags” set in Table 3. To evaluate the efficacy of this flag set, we compare the efficiency and artifact rate before and after applying the union of these flags to our detection. At MAG20, we find that the efficiency drops from 1.000 to 0.999, and the artifact rate drops from 695 deg⁻² to 69 deg⁻². At MAG23, the efficiency drops from 0.847 to 0.791, and the artifact rate drops from 700 deg⁻² to 68 deg⁻². These results

Table 2
Efficiency and Artifact Rate of All Flags

Flag	Description	MAG20		MAG23	
		Eff.	Artifact Rate (deg ⁻²)	Eff.	Artifact Rate (deg ⁻²)
No flag applied	...	1.000	695	0.847	700
flags_negative	Set if source was detected as significantly negative	1.000	695	0.847	700
base_NaiveCentroid_flag	General failure flag	1.000	685	0.847	689
base_NaiveCentroid_flag_noCounts	Object to be centroided has no counts	1.000	695	0.847	700
base_NaiveCentroid_flag_edge	Object too close to edge	1.000	695	0.847	700
base_NaiveCentroid_flag_resetToPeak	Set if CentroidChecker reset the centroid	1.000	685	0.847	689
base_PeakCentroid_flag	Centroiding failed	1.000	695	0.847	700
base_SdssCentroid_flag	General failure flag	1.000	230	0.846	220
base_SdssCentroid_flag_edge	Object too close to edge	1.000	683	0.847	689
base_SdssCentroid_flag_noSecondDerivative	Vanishing second derivative	1.000	695	0.847	700
base_SdssCentroid_flag_almostNoSecondDerivative	Almost vanishing second derivative	1.000	676	0.847	681
base_SdssCentroid_flag_notAtMaximum	Object is not at a maximum	1.000	275	0.846	266
base_SdssCentroid_flag_resetToPeak	Set if CentroidChecker reset the centroid	1.000	679	0.847	683
base_SdssCentroid_flag_badError	Error on x and/or y position is NaN	1.000	681	0.847	685
ip_diffim_NaiveDipoleCentroid_flag	General failure flag, set if anything went wrong	1.000	695	0.847	700
base_CircularApertureFlux_flag_badCentroid	General failure flag, set if anything went wrong	1.000	695	0.847	700
base_GaussianFlux_flag_badCentroid	General failure flag, set if anything went wrong	1.000	695	0.847	700
base_NaiveCentroid_flag_badInitialCentroid	General failure flag, set if anything went wrong	1.000	695	0.847	700
base_PeakLikelihoodFlux_flag_badCentroid	General failure flag, set if anything went wrong	1.000	695	0.847	700
base_PsfFlux_flag_badCentroid	General failure flag, set if anything went wrong	1.000	695	0.847	700
base_SdssCentroid_flag_badInitialCentroid	General failure flag, set if anything went wrong	1.000	695	0.847	700
base_SdssShape_flag_badCentroid	General failure flag, set if anything went wrong	1.000	695	0.847	700
slot_Centroid_flag	General failure flag, set if anything went wrong	1.000	695	0.847	700
ip_diffim_NaiveDipoleCentroid_pos_flag	Failure flag for positive, set if anything went wrong	1.000	695	0.847	700
slot_Centroid_pos_flag	Failure flag for positive, set if anything went wrong	1.000	695	0.847	700
ip_diffim_NaiveDipoleCentroid_neg_flag	Failure flag for negative, set if anything went wrong	1.000	695	0.847	700
slot_Centroid_neg_flag	Failure flag for negative, set if anything went wrong	1.000	695	0.847	700
base_SdssShape_flag	General failure flag	1.000	161	0.791	157
base_GaussianFlux_flag_badShape	General failure flag	1.000	161	0.791	157
slot_Shape_flag	General failure flag	1.000	161	0.791	157
base_SdssShape_flag_unweightedBad	Both weighted and unweighted moments were invalid	1.000	229	0.826	225
base_GaussianFlux_flag_badShape_unweightedBad	Both weighted and unweighted moments were invalid	1.000	229	0.826	225
slot_Shape_flag_unweightedBad	Both weighted and unweighted moments were invalid	1.000	229	0.826	225
base_SdssShape_flag_unweighted	Weighted moments converged to an invalid value; using unweighted moments	1.000	633	0.812	639
base_GaussianFlux_flag_badShape_unweighted	Weighted moments converged to an invalid value; using unweighted moments	1.000	633	0.812	639
slot_Shape_flag_unweighted	Weighted moments converged to an invalid value; using unweighted moments	1.000	633	0.812	639
base_SdssShape_flag_shift	Centroid shifted by more than the maximum allowed amount	1.000	665	0.847	673
base_GaussianFlux_flag_badShape_shift	Centroid shifted by more than the maximum allowed amount	1.000	665	0.847	673
slot_Shape_flag_shift	Centroid shifted by more than the maximum allowed amount	1.000	665	0.847	673
base_SdssShape_flag_maxIter	Too many iterations in adaptive moments	1.000	681	0.825	684
base_GaussianFlux_flag_badShape_maxIter	Too many iterations in adaptive moments	1.000	681	0.825	684
slot_Shape_flag_maxIter	Too many iterations in adaptive moments	1.000	681	0.825	684
base_SdssShape_flag_psf	Failure in measuring PSF model shape	1.000	695	0.847	700
base_GaussianFlux_flag_badShape_psf	Failure in measuring PSF model shape	1.000	695	0.847	700
slot_Shape_flag_psf	Failure in measuring PSF model shape	1.000	695	0.847	700
base_CircularApertureFlux_3_0_flag	General failure flag	1.000	695	0.847	700
base_CircularApertureFlux_3_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	695	0.847	700
base_CircularApertureFlux_3_0_flag_sincCoeffsTruncated	Full sinc coefficient image did not fit within measurement image	1.000	695	0.847	700
base_CircularApertureFlux_4_5_flag	General failure flag	1.000	695	0.847	700
base_CircularApertureFlux_4_5_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	695	0.847	700

Table 2
(Continued)

Flag	Description	MAG20		MAG23	
		Eff.	Artifact Rate (deg ⁻²)	Eff.	Artifact Rate (deg ⁻²)
base_CircularApertureFlux_4_5_flag_sincCoeffsTruncated	Full sinc coefficient image did not fit within measurement image	1.000	694	0.847	699
base_CircularApertureFlux_6_0_flag	General failure flag	1.000	695	0.847	700
base_CircularApertureFlux_6_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	695	0.847	700
base_CircularApertureFlux_6_0_flag_sincCoeffsTruncated	Full sinc coefficient image did not fit within measurement image	1.000	694	0.847	699
base_CircularApertureFlux_9_0_flag	General failure flag	1.000	695	0.847	700
base_CircularApertureFlux_9_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	695	0.847	700
base_CircularApertureFlux_9_0_flag_sincCoeffsTruncated	Full sinc coefficient image did not fit within measurement image	1.000	676	0.847	681
base_CircularApertureFlux_12_0_flag	General failure flag	1.000	695	0.847	700
slot_ApFlux_flag	General failure flag	1.000	695	0.847	700
base_CircularApertureFlux_12_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	695	0.847	700
slot_ApFlux_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	695	0.847	700
base_CircularApertureFlux_17_0_flag	General failure flag	1.000	691	0.847	696
base_CircularApertureFlux_17_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	691	0.847	696
base_CircularApertureFlux_25_0_flag	General failure flag	1.000	682	0.847	687
base_CircularApertureFlux_25_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	682	0.847	687
base_CircularApertureFlux_35_0_flag	General failure flag	1.000	668	0.847	673
base_CircularApertureFlux_35_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	668	0.847	673
base_CircularApertureFlux_50_0_flag	General failure flag	1.000	651	0.847	656
base_CircularApertureFlux_50_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	651	0.847	656
base_CircularApertureFlux_70_0_flag	General failure flag	1.000	637	0.847	641
base_CircularApertureFlux_70_0_flag_apertureTruncated	Aperture did not fit within measurement image	1.000	637	0.847	641
base_GaussianFlux_flag	General failure flag	1.000	267	0.837	260
base_LocalPhotoCalib_flag	Set for any fatal failure	1.000	695	0.847	700
base_LocalWcs_flag	Set for any fatal failure	1.000	695	0.847	700
base_PeakLikelihoodFlux_flag	General failure flag	1.000	695	0.847	700
base_PixelFlags_flag	General failure flag, set if anything went wrong	1.000	695	0.847	700
base_PixelFlags_flag_offimage	Source center is off image	1.000	695	0.847	700
base_PixelFlags_flag_edge	Source is outside usable exposure region (masked EDGE or NO_DATA)	1.000	556	0.847	561
base_PixelFlags_flag_interpolated	Interpolated pixel in the source footprint	0.796	322	0.747	328
base_PixelFlags_flag_saturated	Saturated pixel in the source footprint	1.000	337	0.847	343
base_PixelFlags_flag_cr	Cosmic ray in the source footprint	0.796	562	0.747	567
base_PixelFlags_flag_bad	Bad pixel in the source footprint	1.000	695	0.847	700
base_PixelFlags_flag_suspect	Source's footprint includes suspect pixels	1.000	695	0.847	700
base_PixelFlags_flag_interpolatedCenter	Interpolated pixel in the source center	0.935	362	0.795	369
base_PixelFlags_flag_saturatedCenter	Saturated pixel in the source center	1.000	340	0.847	345
base_PixelFlags_flag_crCenter	Cosmic ray in the source center	0.935	660	0.795	665
base_PixelFlags_flag_suspectCenter	Source's center is close to suspect pixels	1.000	695	0.847	700
base_PsfFlux_flag	General failure flag	1.000	673	0.847	678
slot_PsfFlux_flag	General failure flag	1.000	673	0.847	678
base_PsfFlux_flag_noGoodPixels	Not enough nonrejected pixels in data to attempt the fit	1.000	695	0.847	700
slot_PsfFlux_flag_noGoodPixels	Not enough nonrejected pixels in data to attempt the fit	1.000	695	0.847	700
base_PsfFlux_flag_edge	Object was too close to the edge of the image to use the full PSF model	1.000	689	0.847	694
slot_PsfFlux_flag_edge	Object was too close to the edge of the image to use the full PSF model	1.000	689	0.847	694
ip_diffim_NaiveDipoleFlux_flag	General failure flag, set if anything went wrong	1.000	695	0.847	700
ip_diffim_NaiveDipoleFlux_pos_flag	Failure flag for positive, set if anything went wrong	1.000	695	0.847	700
ip_diffim_NaiveDipoleFlux_neg_flag	Failure flag for negative, set if anything went wrong	1.000	695	0.847	700
ip_diffim_PsfDipoleFlux_flag	General failure flag, set if anything went wrong	1.000	647	0.847	650
ip_diffim_PsfDipoleFlux_pos_flag	Failure flag for positive, set if anything went wrong	1.000	695	0.847	700
ip_diffim_PsfDipoleFlux_neg_flag	Failure flag for negative, set if anything went wrong	1.000	695	0.847	700
ip_diffim_ClassificationDipole_flag	Set to 1 for any fatal failure	1.000	695	0.847	700
ip_diffim_DipoleFit_flag_classification	Flag indicating diaSrc is classified as a dipole	0.999	474	0.846	480
ip_diffim_DipoleFit_flag_classificationAttempted	Flag indicating attempt to classify diaSrc as a dipole	0.999	472	0.846	478

Table 2
(Continued)

Flag	Description	MAG20		MAG23	
		Eff.	Artifact Rate (deg ⁻²)	Eff.	Artifact Rate (deg ⁻²)
ip_diffim_DipoleFit_flag	General failure flag for dipole fit	0.001	222	0.001	221
ip_diffim_DipoleFit_flag_edge	Flag set when dipole is too close to edge of image	1.000	693	0.847	699
base_GaussianFlux_flag_apCorr	Set if unable to aperture correct base_GaussianFlux	1.000	695	0.847	700
base_PsfFlux_flag_apCorr	Set if unable to aperture correct base_PsfFlux	1.000	695	0.847	700
slot_PsfFlux_flag_apCorr	Set if unable to aperture correct base_PsfFlux	1.000	695	0.847	700
ip_diffim_forced_PsfFlux_flag	Forced PSF flux general failure flag	1.000	689	0.847	694
ip_diffim_forced_PsfFlux_flag_noGoodPixels	Forced PSF flux not enough nonrejected pixels in data to attempt the fit	1.000	695	0.847	700
ip_diffim_forced_PsfFlux_flag_edge	Forced PSF flux object was too close to the edge of the image	1.000	689	0.847	694
Union of flags	Apply all above flags for source selection	0.000	0	0.000	0

Note. Flags can be used for removing artifacts. For a specific flag, detected sources are classified as artifacts if the flag is set to True. The first row shows the results without applying flags. The last row shows the results of applying all of the above flags.

Table 3
Efficiency and Artifact Rate of Selected Flags

Flag	Description	MAG20		MAG23	
		Eff.	Artifact Rate (deg ⁻²)	Eff.	Artifact Rate (deg ⁻²)
No flag applied	...	1.000	695	0.847	700
base_PixelFlags_flag_saturated	Saturated pixel in the source footprint	1.000	337	0.847	343
base_PixelFlags_flag_saturatedCenter	Saturated pixel in the source center	1.000	340	0.847	345
base_PixelFlags_flag_suspect	Source's footprint includes suspect pixels	1.000	695	0.847	700
base_PixelFlags_flag_suspectCenter	Source's center is close to suspect pixels	1.000	695	0.847	700
base_PixelFlags_flag_offimage	Source center is off image	1.000	695	0.847	700
base_PixelFlags_flag_edge	Source is outside usable exposure region (masked EDGE or NO_DATA)	1.000	556	0.847	561
base_PixelFlags_flag_bad	Bad pixel in the source footprint	1.000	695	0.847	700
ip_diffim_DipoleFit_flag_classification	Flag indicating diaSrc is classified as a dipole	0.999	474	0.846	480
ip_diffim_DipoleFit_flag_classificationAttempted	Flag indicating classification of diaSrc as a dipole was attempted	0.999	472	0.846	478
base_SdssShape_flag	General failure flag	1.000	161	0.791	157
base_GaussianFlux_flag_badShape	General failure flag	1.000	161	0.791	157
slot_Shape_flag	General failure flag	1.000	161	0.791	157
Union of flags	Apply all above flags for source selection	0.999	69	0.791	68

Note. Flags can be used for removing artifacts. For a specific flag, detected sources are classified as artifacts if the flag is set to True. The first row shows the results without applying flags. The last row shows the results of applying all of the above flags.

suggest that using the “selected flags” to sift artifacts significantly improves the artifact rate with only a small decline in efficiency. We strongly recommend the use of these flags when looking to identify a set of real astrophysical detections in diaSrc tables.

5. Artifact Morphology

5.1. Artifact Definition

Now that we can identify artifacts using the “selected flags” discussed in Section 4.6, we examine their physical origins using artifacts from AL-based subtractions on the MAG20 data set. We define four types of artifacts that are commonly seen in difference images. The morphology of these artifacts is shown in Figure 6.

1. Saturation artifact: caused by saturated pixels, where the PSF shape is distorted. As a result, an optimal kernel solution does not exist.
2. Clustering-pixels artifact: a clustering-pixels artifact appears as clusters of positive pixels and negative pixels. It comes from subtraction residuals of background sources or background noise.
3. Dipole artifact: offset positive and negative lobes. Inaccurate image registration or nonoptimal image resampling can cause dipole artifacts. It is a subclass of the clustering-pixels artifact.
4. Deconvolution artifact: a ring pattern. It is found in AL difference images when the full width at half-maximum (FWHM) of the science image is sharper than the FWHM of the template image. In this scenario, the matching

Table 4
Definitions of Saturation Flags, Dipole Flags, and Shape Flags

Flag Set	Flags	Notes
Saturation flags	base_PixelFlags_flag_saturated, base_PixelFlags_flag_saturatedCenter, base_PixelFlags_flag_suspect, base_PixelFlags_flag_suspectCenter, base_PixelFlags_flag_offimage, base_PixelFlags_flag_edge, base_PixelFlags_flag_bad	Some of these flags are not actually for saturation but rather for other types of pixel-level issues. We include them here because they also indicate pixel issues. Visually inspecting detected sources shows that the majority of artifacts that have these flags set to True are saturation artifacts.
Dipole flags	ip_diffim_DipoleFit_flag_classification, ip_diffim_DipoleFit_flag_classificationAttempted	...
Shape flags	base_SdssShape_flag, base_GaussianFlux_flag_badShape	The algorithm is based on the elliptical shape (Lupton et al. 2001; Bosch et al. 2017) of the detected sources. The implementation (v23.0.0) used in our analysis flags negative pixel values, which is not what we want for time-variable objects that can be brighter and fainter than in the template image. This bug has been fixed in v25.0.0 of the LSST Science Pipelines.

kernel has to “deconvolve” the PSF of the template but is not able to do so because the necessary information is not available.

5.2. Artifact Morphology and Flags

To help us understand the relationship between flags and artifact morphology, we split the “selected flags” defined in Section 4.6 into three groups: saturation flags, dipole flags, and shape flags. As their names suggest, saturation flags are set to True if the detected source contains saturated pixels. We name detected artifacts with the saturation flag set to True saturation-flag artifacts and name the rest non-saturation-flag artifacts. Dipole flags are set to True for dipole artifacts. It is possible that a detected source visually appears as a dipole but has no dipole flags set to True. This could be related to the trigger mechanism and flux separation of dipole measurements. When the DIA pipeline is making a dipole classification for a detected source, it fits the fluxes of the negative and positive lobes and their corresponding centroids simultaneously. A detected source is classified as a dipole if the quadrature sum of the fitted fluxes is above the corresponding minimum S/N threshold and the fitted fluxes weighted by the total flux are both smaller than the maximum flux ratio threshold. Any detection that visually appears as a dipole but does not satisfy these criteria is not classified as a dipole artifact. Also, the fitting algorithm itself may have degeneracy in estimating dipole fluxes. Dipoles from faint and highly separated sources are almost indistinguishable from dipoles from bright and closely separated sources (Reiss 2016). The inaccuracy of dipole flux estimation could further affect dipole classification. Possible improvements such as using presubtraction images in dipole fitting are discussed in Reiss (2016). Shape flags are set to True if the detected source has second moments that are inconsistent with a Gaussian shape. More detailed definitions of these flag sets are given in Table 4.

5.3. Seeing Conditions

Artifact morphology can be affected by the relative seeing conditions of template images and science images. We split our image pairs defined in Section 3.3 into three classes based on

the FWHM of their coadd (template)/calexp (science) image PSFs. These classes are defined as

1. *broad*: $\text{calexp FWHM} \geq \text{coadd FWHM}$;
2. *near*: $\text{calexp FWHM} < \text{coadd FWHM}$ and $(\text{calexp FWHM} - \text{coadd FWHM}) / (\text{coadd FWHM}) > -0.05$; and
3. *sharp*: $(\text{calexp FWHM} - \text{coadd FWHM}) / (\text{coadd FWHM}) \leq -0.05$.

In the *broad* class, calexp have larger PSFs than coadd exposures, which is the condition that the AL algorithm is capable of matching the coadd PSF to the calexp PSF. In the *near* class, calexp PSFs are smaller than coadd PSFs. However, the relative difference of FWHMs is still close. In the *sharp* class, calexp PSFs are much smaller than coadd PSFs. The AL matching kernel has to “deconvolve” the coadd PSF to match the calexp PSF.

5.4. Artifact Morphology and Artifact Fraction

Using the above definitions, we can study artifact morphology in each FWHM class. We start with calculating the proportions of artifacts in each flag group. The results are shown in Table 5. The fraction of artifacts in each class with saturation flags set to True is nearly 50%. Since they are subtraction residuals of bright background sources (stars, galaxies, etc.), it is important to understand at what background source S/N this issue occurs most frequently. We match background sources from the calexp src (Section 3.1) to artifacts using a one-direction-matching¹⁹ method with a 2'' search radius. Figure 7 shows the artifact fraction with respect to the calexp source S/N.²⁰ There is an increasing trend between the artifact fraction and the source S/N. This comparison shows that using saturation flags to identify pixel issues can help us to remove saturation artifacts, especially when the source S/N is greater than 600.

Next, we study the artifacts that have no saturation flags set to True. These non-saturation-flag artifacts are detected under different seeing conditions with different artifact rates. For our

¹⁹ One background source may cause subtraction artifacts in many science images.

²⁰ We calculate the S/Ns of the original contributing sources by dividing $\text{base_PsfFlux_instFlux}$ by $\text{base_PsfFlux_instFluxErr}$ from the src.

Table 5
Artifact Morphology Fraction

	Broad	Near	Sharp
Saturation flags	51.90%	47.60%	53.31%
Dipole flags only	1.45%	2.92%	3.08%
Shape flags only	25.11%	24.84%	21.76%
Both dipole flags and shape flags	12.28%	12.74%	12.14%
Remaining	9.26%	11.90%	9.71%
Non-saturation-flag artifacts per square degree	286	312	414

Note. Percentage of artifacts in each flag set of the *broad* class, *near* class, and *sharp* class. The first row shows the fraction of saturation-flag artifacts. The second row shows the fraction of non-saturation-flag artifacts that have dipole flags set to True and shape flags set to False. The third row shows the fraction of non-saturation-flag artifacts that have dipole flags set to False and shape flags set to True. The fourth row shows the fraction of non-saturation-flag artifacts that have both dipole flags set to True and shape flags set to True. The fifth row shows the fraction of non-saturation-flag artifacts that have both dipole flags and shape flags set to False. The last row shows the artifact rate (deg^{-2}) of non-saturation-flag artifacts.

analysis, we find that the rates of non-saturation-flag artifacts per square degree are 286 (*broad*), 312 (*near*), and 414 (*sharp*), which suggests that as the calexp PSFs become sharper, the artifact rate increases (Table 5).

The morphologies of non-saturation-flag artifacts are shown in Figures 8, 9, and 10. By inspecting postage stamps of these artifacts, we find that the majority of artifacts in the *broad* class are dipole artifacts and clustering-pixels artifacts. In the *near* class, the artifact morphology is similar to the *broad* class. However, in the *sharp* class, there are more ring artifacts, which indicates the PSF matching kernel is deconvolving the seeing condition of the template image.

To understand the source properties of the non-saturation-flag artifacts, we match them to the src using the one-direction-matching method with a $2''$ matching radius.²¹ Of the artifacts, 80.2% have matches in the src, which means most of these artifacts are also subtraction residuals of background sources. In Figure 11, S/N distributions of matched sources in each FWHM class are plotted. Most sources have S/Ns close to or higher than 200, which means bright sources are the main contribution to detection artifacts. This is reasonable because we can only observe artifacts from bright sources where subtraction residuals can exceed sky background fluctuations. It is hard for faint sources to produce visible artifacts because their subtraction residuals are overwhelmed by sky background noise. Other than bright sources, we also find clusters of faint sources in the S/N distributions of each class. In Figure 12, we plot a few stamps of faint sources from coadd exposures, calexp, and their corresponding artifacts from difference exposures. These plots show that peak pixels are the main causes of these artifacts. As to the artifacts that have no matches from the src, we show their morphology in Figure 13. Clearly, background noise is the origin of these artifacts. The artifact rate of this type is 62 deg^{-2} in the *broad* class, 57 deg^{-2} in the *near* class, and 79 deg^{-2} in the *sharp* class.

²¹ We exclude the saturation artifacts because the origin of these artifacts can be inferred by the name of the flag, which is set to True.

5.5. Dipole Artifacts

Because dipole artifacts are prevalent in all classes with the AL algorithm, we investigate whether the ZOGY algorithm also produces dipole artifacts at the same locations. Figure 14 shows postage stamps of AL dipoles and postage stamps of the ZOGY algorithm at the same positions. Both the kernel matching method (AL) and the cross-filtering method (ZOGY) have dipole artifacts with the same orientations at the same locations.

Here, we give a brief discussion of several different possible causes of dipole artifacts.

(1) Image registration. It is possible that dipoles are caused by small errors in the relative astrometric registration of the images. Applying astrometric calibration may reduce the registration error. However, even subpixel errors could be magnified by bright sources, thereby producing visible dipole artifacts that have positive-negative separations of multiple pixels. (2) Sampling. Another possible origin of dipoles is the sampling process used for observing or warping exposures. Errors introduced in this process can be magnified by the brightness of background stars, which would result in dipole artifacts. (3) Moving object. An object with a small motion between the images can create a dipole that represents real astrophysical change. However, there are no stars with proper motion in the DC2 simulation, so this particular effect is not applicable to our testing here. (4) PSF matching. It is also possible that the AL algorithm finds an approximate kernel solution. An asymmetric kernel convolving a symmetric coadd source and subtracted by a symmetric calexp source could produce a dipole artifact. However, we also find dipole artifacts using the ZOGY cross-filtering method, which has no kernel fitting involved (Figure 14). We do not entirely exclude the kernel fitting explanation because the PSF models used for cross-filtering are measured from images. It is thus also possible that the approximate PSF used for cross-filtering caused the dipole artifact, which is mathematically similar to a nonoptimal AL matching kernel. (5) Poisson fluctuation. In the bright region, the Poisson fluctuation of background sources can exceed the background sky fluctuation, which leads to local bright/faint residuals. These residuals can spread to nearby pixels after convolution operation, which may appear as dipole artifacts in difference images. More accurate estimations of variance and covariance may increase the detection thresholds and thereby remove these artifacts from the diaSrc table. (6) Differential chromatic refraction (DCR). The atmosphere of the Earth refracts incident light, thereby deflecting the observed positions of sources (FILIP-PENKO 1982). This effect could result in dipole artifacts in difference images. Since the DC2 images do not include this effect, we exclude DCR as the cause of the artifacts in our analysis.

From this discussion, we conclude that the origin of the dipole artifacts is still undetermined. To thoroughly understand the cause will require control of the image resampling in more detail and carefully considering the resulting PSFs of the calexp and coadd templates.

6. Possible Improvements

We here summarize some potential improvements to the performance of the LSST Science Pipelines image subtraction algorithms.

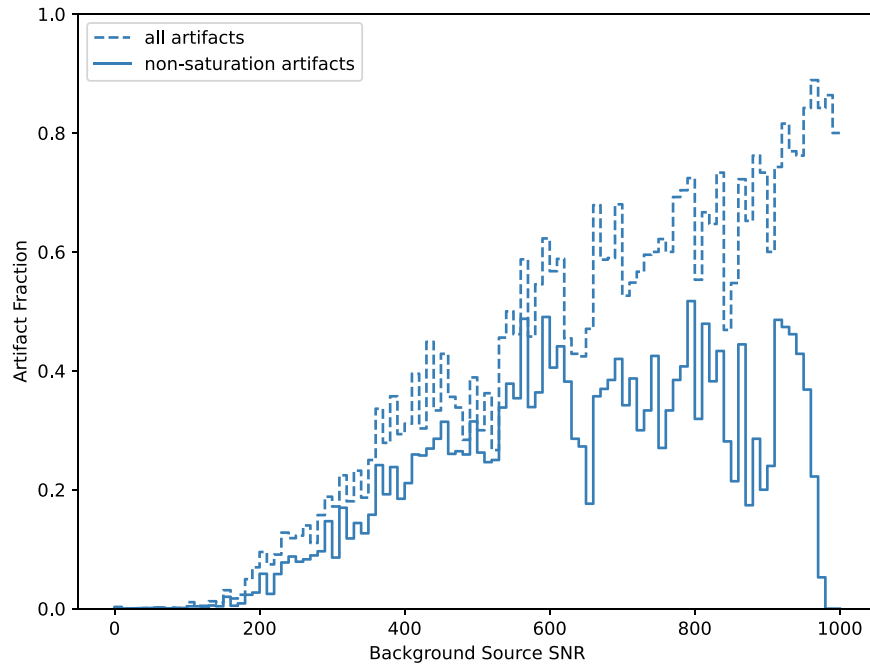


Figure 7. Artifacts come from the mis-subtraction of bright sources. Here we show the fraction of detections that are artifacts as a function of the S/N of the nearest background source within $2''$. The dashed line shows the fraction of all artifacts. The solid line shows the fraction of artifacts that have no saturation flags set to True.

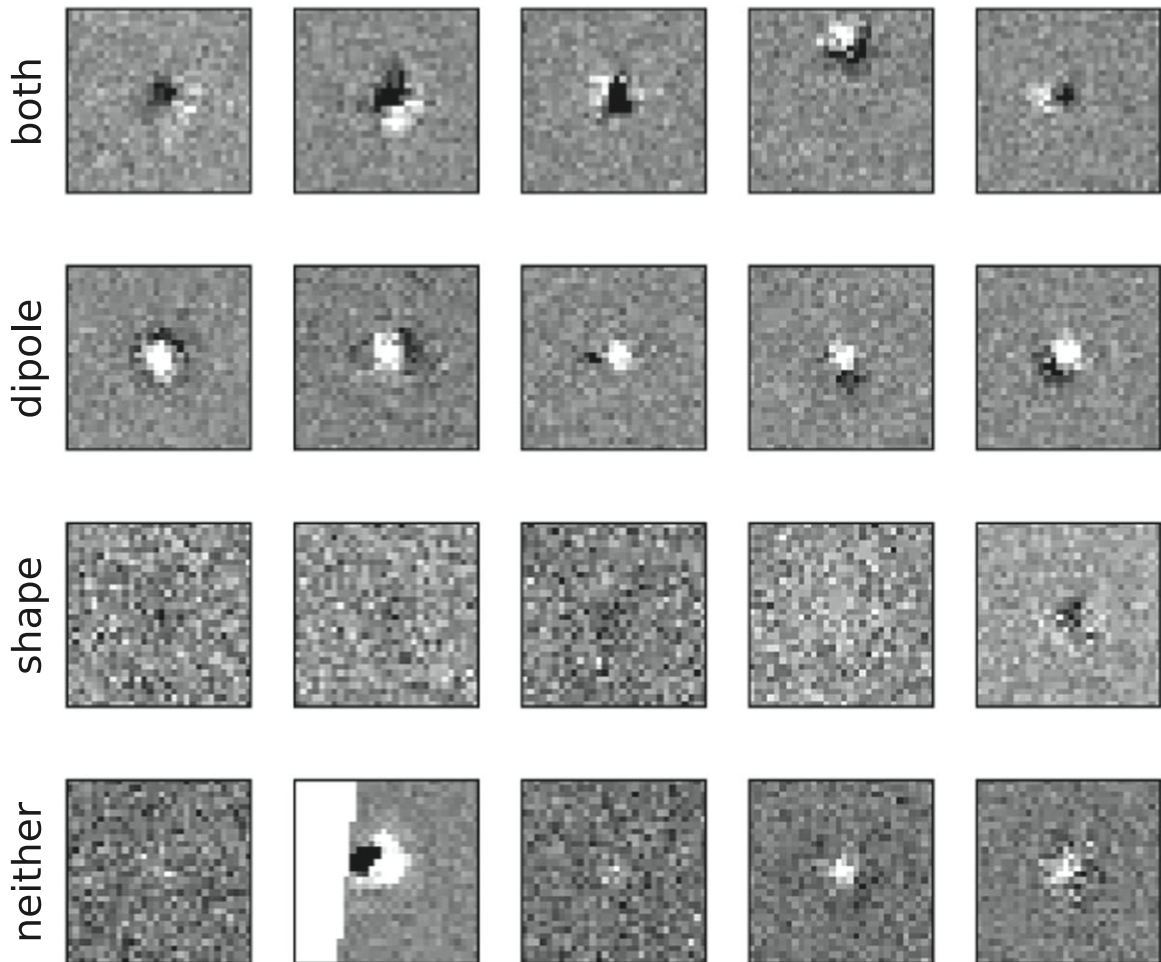


Figure 8. Morphology of non-saturation-flag artifacts from the *broad* class. The top row shows artifacts with both dipole flags and shape flags set to True. The second row from the top shows artifacts with dipole flags set to True and shape flags set to False. The second row from the bottom shows artifacts with dipole flags set to False and shape flags set to True. The bottom row shows artifacts with both dipole flags and shape flags set to False.

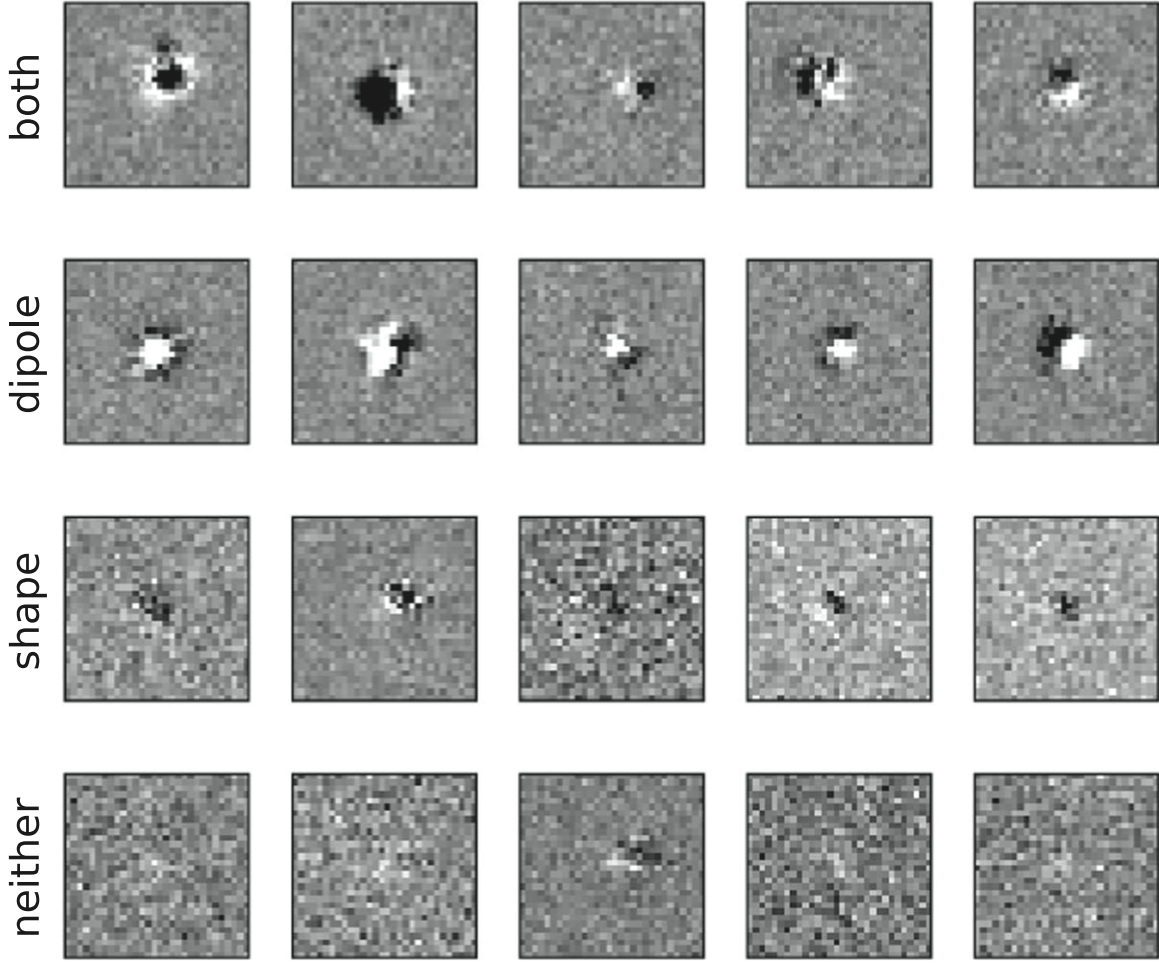


Figure 9. Morphology of non-saturation-flag artifacts from the *near* class. The top row shows artifacts with both dipole flags and shape flags set to True. The second row from the top shows artifacts with dipole flags set to True and shape flags set to False. The second row from the bottom shows artifacts with dipole flags set to False and shape flags set to True. The bottom row shows artifacts with both dipole flags and shape flags set to False.

(1) Implement spatially varying delta bases. In our work, we find that the default kernel bases used in the DIA pipeline do not fully correct sampling or registration issues. Instead of using Gauss-polynomial functions as the bases, we suggest using delta bases (Bramich 2008; Miller et al. 2008), which are the most flexible bases for image difference. We also recommend including spatial variation in the delta kernel because noise properties and PSF shapes may vary across a CCD image. To avoid overfitting the model to noise using these more flexible bases, different regularization techniques (Becker et al. 2012; Bramich et al. 2016) should also be tested.

(2) Correct the kernel solution when using preconvolution or choosing calexp as templates. When the PSF of a calexp is sharper than the PSF of a coadd exposure, the image subtraction kernel needs to perform the deconvolution to the coadd exposure, which usually results in ring artifacts in the difference image. To avoid ring artifacts, one approach is to “preconvolve” the calexp using either its own PSF or a more generic Gaussian kernel. The other approach is to use the calexp as the template image and match its PSF to the PSF of the coadd exposure. Both approaches involve convolving the calexp, which adds correlations between pixels. The default normal equation of the image difference solution assumes the template is noiseless. To account for the effect of correlations,

we suggest including the pixel covariance matrix Σ in the kernel solution of the AL algorithm. Due to the huge size of Σ , how to properly estimate the correlations of neighboring pixels and store Σ are worthy questions for consideration in implementation.

(3) Improve the numerical stability of the ZOGY decorrelation kernel and its AL equivalent. The ZOGY algorithm provides a source-detection optimal difference image with the advantage of noise decorrelation. However, the numerical instability caused by the division of small values at high frequencies can affect the quality of its difference image and equivalent PSF (Reiss 2017; Kovacs 2021). Kovacs (2021) suggests that if we express the ZOGY $\hat{D}$ image as $\hat{D} = \hat{k}_s \hat{S} - \hat{k}_t \hat{T}$, with $\hat{k}_s \equiv \frac{F_t \hat{P}_t}{\sqrt{\sigma_s^2 F_t^2 |\hat{P}_t|^2 + \sigma_t^2 F_s^2 |\hat{P}_s|^2}}$ and

$\hat{k}_t \equiv \frac{F_s \hat{P}_s}{\sqrt{\sigma_s^2 F_t^2 |\hat{P}_t|^2 + \sigma_t^2 F_s^2 |\hat{P}_s|^2}}$, the asymptotic behaviors of $\hat{k}_s$ and

$\hat{k}_t$ can be determined by examining the ratio of $\frac{|\hat{P}_t|}{|\hat{P}_s|}$, which is either 0 or ∞ . In practice, the fast Fourier transform applied to obtain frequency images cannot maintain this ratio as a smooth function of frequency due to numerical accuracy. To correct this issue, Kovacs (2021) proposes three possible work-arounds, which are (a) using Gaussian models to create PSFs in Fourier space directly, (b) replacing the high-frequency noise using Gaussian limit values, and (c) using the score

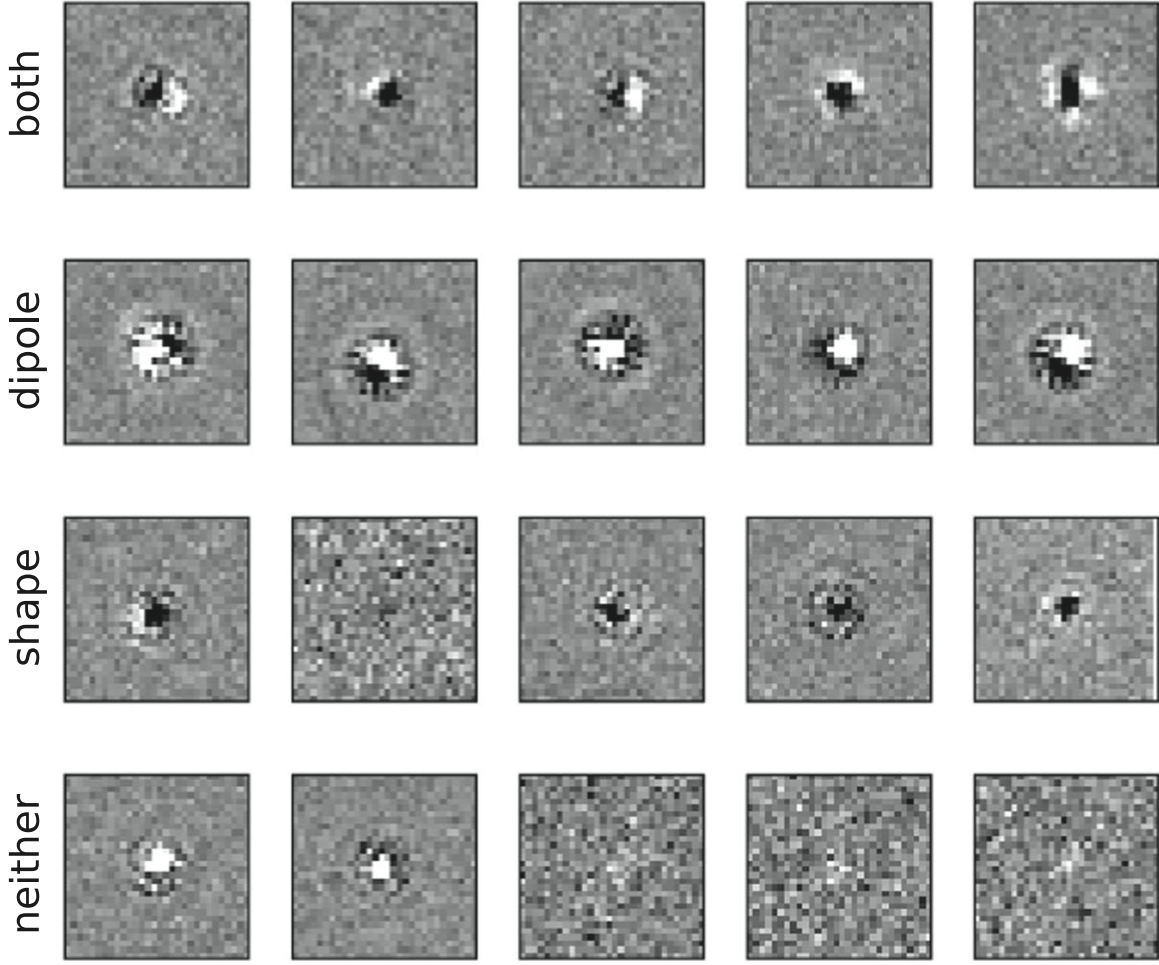


Figure 10. Morphology of non-saturation-flag artifacts from the *sharp* class. The top row shows artifacts with both dipole flags and shape flags set to True. The second row from the top shows artifacts with dipole flags set to True and shape flags set to False. The second row from the bottom shows artifacts with dipole flags set to False and shape flags set to True. The bottom row shows artifacts with both dipole flags and shape flags set to False.

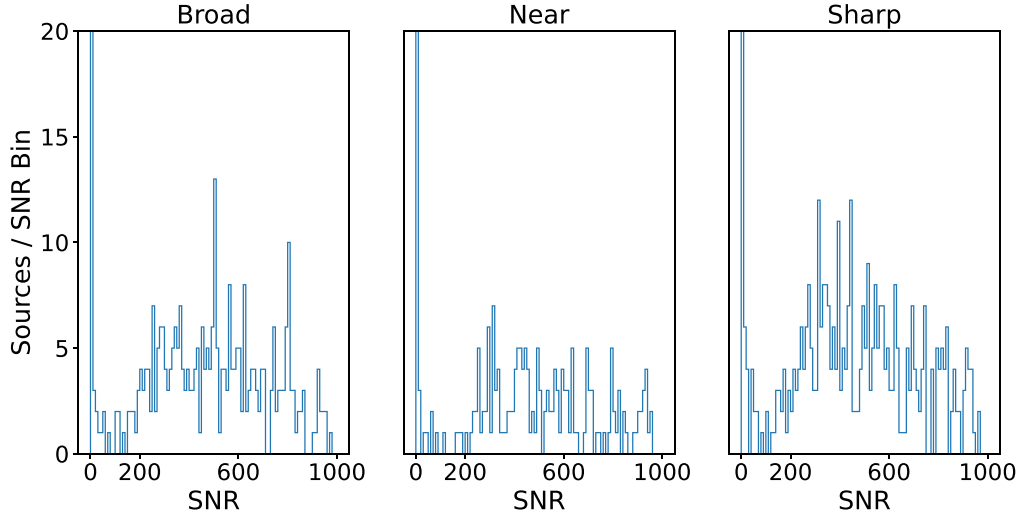


Figure 11. Signal-to-noise distribution of the sources in different FWHM classes.

image of the ZOGY algorithm for detection because the score image is further convolved by P_D , which can suppress the noisy part in D . The validity of these work-arounds has not been thoroughly studied and would benefit from careful thought and testing.

(4) Decorrelate the Poisson noise for bright sources. The ideal difference image is expected to have white noise, which means the covariance matrix of the pixels of the difference is diagonal. The ZOGY algorithm proposes dividing the cross-filtering difference image by $\frac{1}{\sqrt{\sigma_s^2 F_t^2 | \hat{P}_t |^2 + \sigma_t^2 F_s^2 | \hat{P}_s |^2}}$ for image

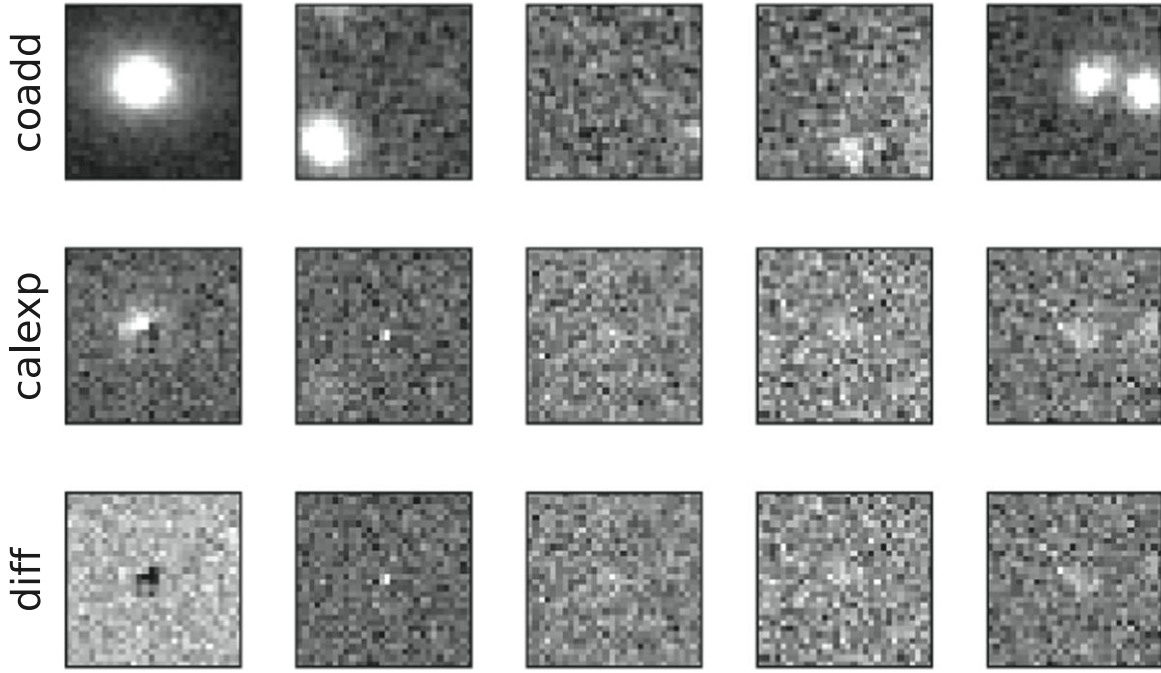


Figure 12. Morphology of artifacts from faint background sources. (Top row) Postage stamps of coadd images. (Second row from top) Postage stamps of calexp images. (Bottom row) Postage stamps of artifacts.

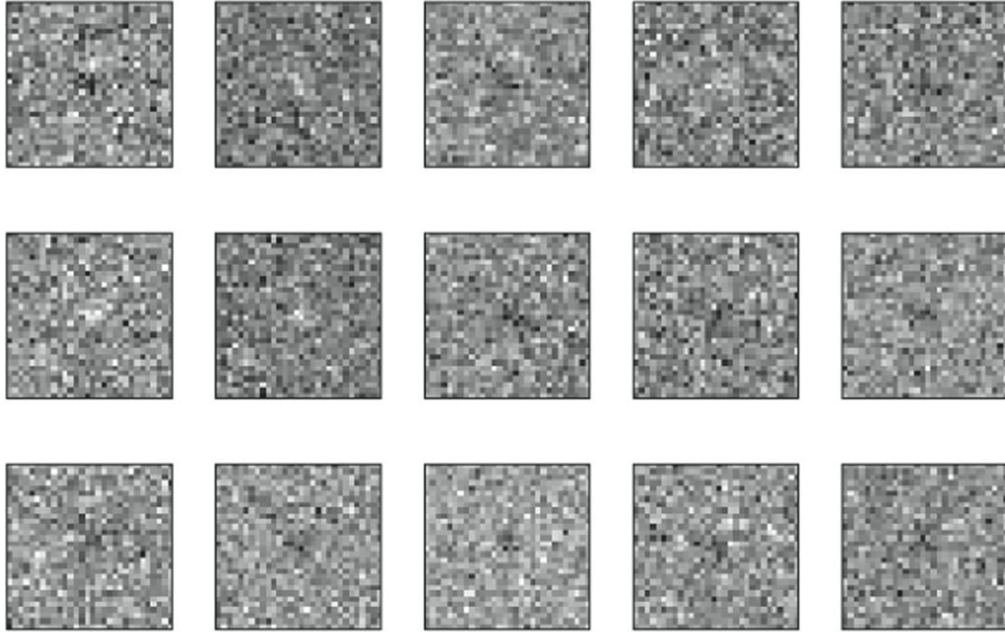


Figure 13. Morphology of artifacts that have no matches from the src. These artifacts are visually similar in each class. Inspecting their stamps shows that they are from correlated noise.

noise whitening. Mathematically, this is equivalent to the zero-phase component analysis (ZCA; Kessy et al. 2015) in complex space. The proof of this equivalence is given in the [Appendix](#). In reality, however, the white-noise assumption for the original template image and science image is not proper for bright sources, where the Poisson noise of bright pixels is dominant. To obtain white noise for bright sources, it would be possible to apply a standard whitening process for the region around each bright source as long as we have its covariance matrix.

However, given the large number of bright sources, calculating the inverse matrix of the covariance matrix may be time-consuming. A work-around method would be to treat each row or column of the stamp as a random vector and require the correlation between rows or columns to be white (Shi et al. 2016; Seghouane & Shokouhi 2018). This simplification significantly reduces the dimensions of the covariance matrix; however, the reliability of this method for image difference has not been tested.

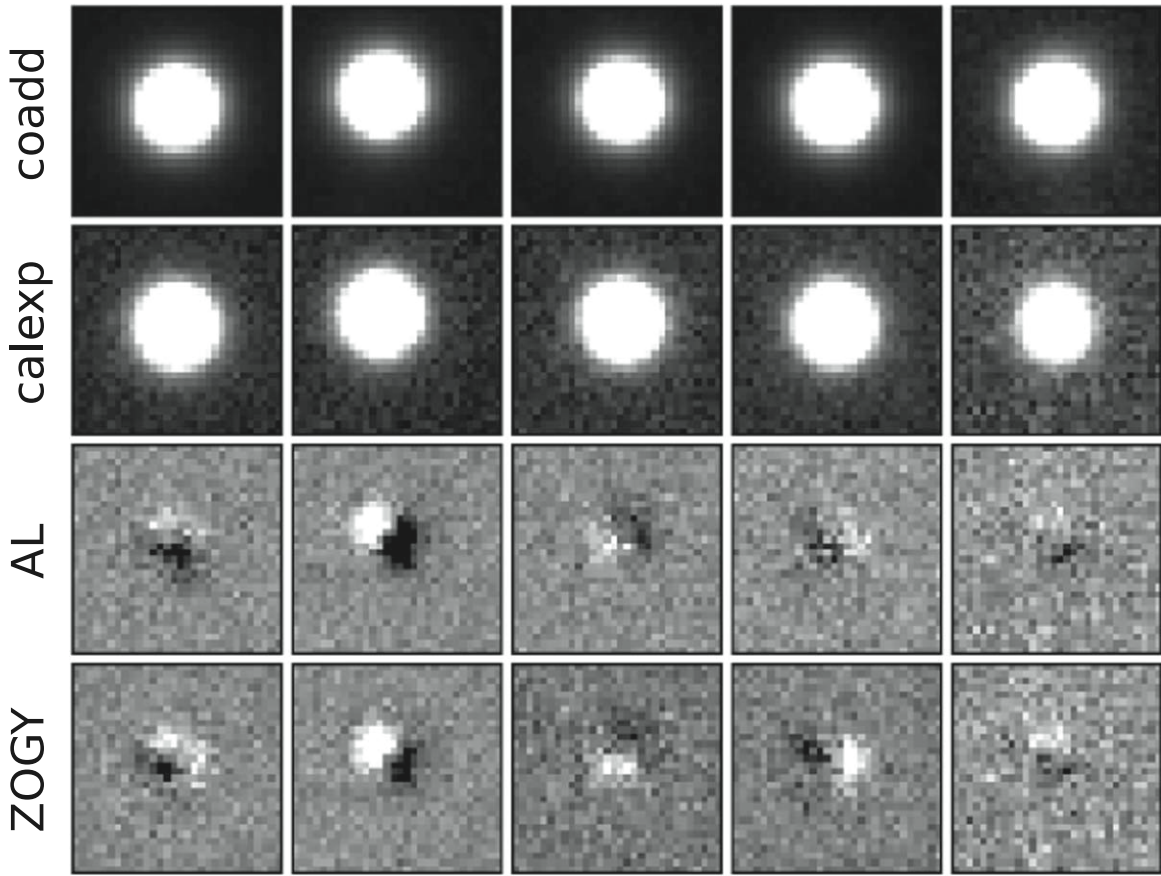


Figure 14. (Top row) Postage stamps of coadd images. (Second row from top) Postage stamps of calexp images. (Second row from bottom) Postage stamps of dipole artifacts in subtraction using the AL Gauss-polynomial algorithm. (Bottom row) Postage stamps of subtractions using the ZOGY algorithm made at the same positions as the AL dipoles. The artifacts are consistent between the two methods.

7. Conclusions

We have evaluated the LSST Science Pipelines DIA pipeline using synthetic source injection. We selected seven coadd template exposures along with 10 calexp exposures overlapping each coadd in the i band from the DC2 data set. By tracing the recovery of injected synthetic sources, we predict the range of the magnitude depth ($m_{1/2}$) of the pipeline for realistic simulated LSST images to be 23.40–23.71 mag when the host SB ranges between 20–21 mag arcsec⁻² and 24–25 mag arcsec⁻². We explored the effects of changing the AL kernel spatial degree of freedom on reducing artifacts with synthetic injections at MAG20 and MAG23.

We find that the default setup of the AL algorithm is good in terms of transient source detection and photometric measurements. At the injected magnitudes of 20 and 21, the efficiency is above 0.989, and more than 99% of the flux measurements are within 0.1 mag of the injected magnitudes. In the region away from the image edge, increasing the spatial degrees of freedom up to 4 can reduce the artifact rate without hurting efficiency and flux measurement. We calculate the efficiency and artifact rate of the flags from the diaSrc table. We find that by applying a proper combination of saturation flags, dipole flags, and shape flags to the detection from the diaSrc table, we can reduce the artifact rate from 695 deg⁻² to 69 deg⁻² at MAG20, and the efficiency only drops from 1.000 to 0.999. At MAG23, the artifact rate drops from 700 deg⁻² to 68 deg⁻², while the efficiency drops from 0.847 to 0.791. To

study the morphology of the artifacts and understand their origins, we split the artifacts into *broad*, *near*, and *sharp* classes, which correspond to the seeing condition where the calexp PSF is larger than the coadd PSF, the calexp PSF is smaller but less than 5% smaller than the coadd PSF, and the calexp PSF is more than 5% smaller than the coadd PSF. After removing saturation artifacts from the analysis, the artifact per square degree rate is 286 in the broad class, 312 in the near class, and 414 in the sharp class. We find dipole artifacts and correlated-pixel artifacts in subtractions from all classes. Ring artifacts are seen only in the *sharp* class, which is consistent with rings being decorrelation artifacts. By mapping these artifacts to the src of the original calexp, we find that most of the artifacts are subtraction residuals from regions around bright ($S/N \geq 200$) stars. Possible causes of these artifacts include inexact PSF matching, registration issues, sampling errors, and Poisson fluctuation of bright pixels. To remove these artifacts from difference images, we suggest possible improvements, such as applying a set of more flexible kernel bases with spatial variation and improving the treatment of correlated noise in the images.

Acknowledgments

This paper has undergone internal review in the LSST Dark Energy Science Collaboration. The primary authors would like to thank the internal reviewers Bob Armstrong, David Jones, and Ian Sullivan for their comments.

We also further thank Richard Kessler, who contributed to the discussion of the difference image analysis and made many suggestions that improved the paper.

Author contributions are listed below. S.L.: project lead, tested the DESC DIA pipeline using synthetic source injection method, wrote the paper. W.M.W.-V.: project lead, provided feedback to the difference image analysis, wrote the paper. R.A.: contributed to the development of the DESC DIA pipeline, provided technical advice for running image difference software. G.N.: contributed to the discussion of the difference image analysis. B.S.: contributed to the discussion of the difference image analysis, draft reviewer.

The DESC acknowledges ongoing support from the Institut National de Physique Nucléaire et de Physique des Particules in France; the Science & Technology Facilities Council in the United Kingdom; and the Department of Energy, the National Science Foundation, and the LSST Corporation in the United States. DESC uses resources of the IN2P3 Computing Center (CC-IN2P3–Lyon/Villeurbanne - France) funded by the Centre National de la Recherche Scientifique; the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under contract No. DE-AC02-05CH11231; STFC DiRAC HPC Facilities, funded by UK BEIS National E-infrastructure capital grants; and the UK particle physics grid, supported by the GridPP Collaboration. This work was performed in part under DOE contract DE-AC02-76SF00515.

W.M.W.-V. and S.L. were supported in part by DOE Office of Science HEP grant DE-SC0007914.

Software: Difference Image Analysis pipeline using LSST DM Science Pipelines developed for DESC Data Challenges (https://github.com/LSSTDESC/dia_pipe), LSST Science Pipelines (v23.0.0; https://pipelines.lsst.io/releases/v23_0_0.html), The obs model for the LSST cameras (w.2022.48; https://github.com/lsst/obs_lsst), Astropy (Astropy Collaboration et al. 2022), Matplotlib (Hunter 2007), Numpy (Harris et al. 2020), pandas (McKinney 2010; The pandas development team 2023), Scipy (Virtanen et al. 2020), SQLite (Hipp 2020).

Appendix Connections between the ZOGY Algorithm and ZCA Whitening

In signal processing, whitening is defined as a linear transformation that maps a random vector to a new vector with an identity covariance matrix. The ZCA is a whitening process that minimizes the squared distance between the original vector and the whitened vector (Kessy et al. 2015) in real space. In this section, we prove that the ZOGY decorrelation is equivalent to the ZCA whitening in complex space. For simplicity, we assume that the expectation of the original random vector is 0.

In ZCA whitening, a random vector X with covariance Σ_X can be transformed to a white vector Z via matrix W as $Z = WX$. The solution of W is

$$W = \Sigma_X^{-\frac{1}{2}} = U \Lambda^{-\frac{1}{2}} U^T, \quad (\text{A1})$$

with $\Sigma_X^{-\frac{1}{2}T} \Sigma_X^{-\frac{1}{2}} = \Sigma_X^{-1}$. U is an orthogonal transformation, and Λ is a diagonal matrix where each element is an eigenvalue of Σ_X .

The covariance matrix of Z is the identity matrix I :

$$\Sigma_Z = E(ZZ^T) = W \Sigma_X W^T = \Sigma_X^{-\frac{1}{2}} (U \Lambda U^T) \Sigma_X^{-\frac{1}{2}T} = I. \quad (\text{A2})$$

In complex space, we generalize the ZCA whitening as

$$W = \Sigma_X^{-\frac{1}{2}} = U \Lambda^{-\frac{1}{2}} U^\dagger, \quad (\text{A3})$$

with U as a unitary matrix and $U^\dagger$ as its conjugate transpose.

Suppose we want to decorrelate a 1D cross-filtering difference vector M , defined as

$$M = p_s \otimes T - p_t \otimes S. \quad (\text{A4})$$

The decorrelation matrix W_M should equal $\Sigma_M^{-\frac{1}{2}}$. Define P_s and P_t as circulant matrices of 1D vectors p_s and p_t . M can be expressed as $M = P_s T - P_t S$. P_s and P_t can be diagonalized by the Fourier transform matrix F and its conjugate transpose $F^\dagger$ as $P_s = F^\dagger \hat{P}_s F$ and $P_t = F^\dagger \hat{P}_t F$, with $\hat{P}_s$ and $\hat{P}_t$ being the diagonal eigenvalue matrices of P_s and P_t , which are also the Fourier transforms of p_s and p_t . The Fourier transform matrix F is defined as

$$(F_{mn}) = \left(\frac{1}{\sqrt{N}} e^{-\frac{2\pi i}{N} mn} \right). \quad (\text{A5})$$

Now we can express Σ_M as

$$\begin{aligned} \Sigma_M &= \Sigma_{P_s T} + \Sigma_{P_t S} = P_s \Sigma_T P_s^\dagger + P_t \Sigma_S P_t^\dagger \\ &= F^\dagger (\hat{P}_s F \Sigma_T F^\dagger \hat{P}_s^\dagger + \hat{P}_t F \Sigma_S F^\dagger \hat{P}_t^\dagger) F. \end{aligned} \quad (\text{A6})$$

If T and S are white with covariance matrices $\sigma_t^2 I$ and $\sigma_s^2 I$, Σ_M can be simplified as

$$\Sigma_M = F^\dagger (\sigma_t^2 |\hat{P}_s|^2 + \sigma_s^2 |\hat{P}_t|^2) F. \quad (\text{A7})$$

The inverse square root of Σ_M is

$$\Sigma_M^{-\frac{1}{2}} = F^\dagger (\sigma_t^2 |\hat{P}_s|^2 + \sigma_s^2 |\hat{P}_t|^2)^{-\frac{1}{2}} F. \quad (\text{A8})$$

We can interpret $W_M = \Sigma_M^{-\frac{1}{2}}$ as a rotation of M to $\hat{M}$ by the Fourier transform. It then divides each component of $\hat{M}$ by the corresponding component of $(\sigma_t^2 |\hat{P}_s|^2 + \sigma_s^2 |\hat{P}_t|^2)^{-\frac{1}{2}}$. Finally, it rotates the $(\sigma_t^2 |\hat{P}_s|^2 + \sigma_s^2 |\hat{P}_t|^2)^{-\frac{1}{2}} \hat{M}$ back to the original coordinate system. This is exactly the same whitening procedure as the ZOGY deconvolution solution in one dimension.

In 2D, images can be represented as 1D row-ordered vectors, and PSFs can be represented as doubly circulant matrices. The Fourier transform can be represented as a Kronecker product of 1D Fourier transform matrices. The inverse of the 2D Fourier transform matrix can diagonalize circulant matrices with their Fourier transforms as eigenvalues (Jain 1989). Thus, the proof in 1D can be generalized to 2D.

ORCID iDs

S. Liu  <https://orcid.org/0000-0002-4461-2143>
W. M. Wood-Vasey  <https://orcid.org/0000-0001-7113-1233>
R. Armstrong  <https://orcid.org/0000-0002-6911-1038>
G. Narayan  <https://orcid.org/0000-0001-6022-0484>
B. O. Sánchez  <https://orcid.org/0000-0002-8687-0669>

References

- Alard, C. 2000, *A&AS*, **144**, 363
- Alard, C., & Lupton, R. H. 1998, *ApJ*, **503**, 325
- Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., et al. 2022, *ApJ*, **935**, 167

- Becker, A. C., Homrighausen, D., Connolly, A. J., et al. 2012, *MNRAS*, **425**, 1341
- Bernstein, J. P., Kessler, R., Kuhlmann, S., et al. 2012, *ApJ*, **753**, 152
- Bosch, J., Armstrong, R., Bickerton, S., et al. 2017, *PASJ*, **70**, S5
- Bramich, D. M. 2008, *MNRAS*, **386**, L77
- Bramich, D. M., Horne, K., Albrow, M. D., et al. 2013, *MNRAS*, **428**, 2275
- Bramich, D. M., Horne, K., Alsubai, K. A., et al. 2016, *MNRAS*, **457**, 542
- Doctor, Z., Kessler, R., Chen, H. Y., et al. 2017, *ApJ*, **837**, 57
- Filippenko, A. V. 1982, *PASP*, **94**, 715
- Frieman, J. A., Bassett, B., Becker, A., et al. 2008, *AJ*, **135**, 338
- Graham, M. L., Bellm, E., Guy, L., et al. 2019, LSST Alerts: Key Numbers DMTR-102, Vera C. Rubin Observatory, <https://dmtr-102.lsst.io/>
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Natur*, **585**, 357
- Hipp, R. D., et al., 2020 SQLite, v3.31.1, <https://www.sqlite.org/index.html>
- Hu, L., Wang, L., Chen, X., & Yang, J. 2022, *ApJ*, **936**, 157
- Hunter, J. D. 2007, *CSE*, **9**, 90
- Jain, A. 1989, *Fundamentals of Digital Image Processing* (4th ed.; Englewood Cliffs, NJ: Prentice-Hall), 145
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, *ApJ*, **873**, 111
- Kaiser, N., Aussel, H., Burke, B. E., et al. 2002, *Proc. SPIE*, **4836**, 154
- Kessler, R., Marriner, J., Childress, M., et al. 2015, *AJ*, **150**, 172
- Kessy, A., Lewin, A., & Strimmer, K. 2015, *Am. Stat.*, **72**, 309
- Kovacs, G. 2021, The ZOGY Image Differencing Matching Kernel and PSF Solutions and Their Practical Implementation Issues DMTN-179, Vera C. Rubin Observatory, <https://dmtn-179.lsst.io/>
- LSST Dark Energy Science Collaboration (LSST DESC), Abolfathi, B., Alonso, D., et al. 2021, *ApJS*, **253**, 31
- LSST Science Collaboration, Abell, P. A., Allison, J., et al. 2009, arXiv:0912.0201
- Lupton, R., Gunn, J. E., Ivezić, Z., Knapp, G. R., & Kent, S. 2001, in ASP Conf. Ser. 238, *Astronomical Data Analysis Software and Systems X*, ed. J. Harnden et al. (San Francisco, CA: ASP), 269
- Malmquist, K. G. 1922, *MeLuF*, **100**, 1
- Malmquist, K. G. 1925, *MeLuF*, **106**, 1
- Mao, Y.-Y., Kovacs, E., Heitmann, K., et al. 2018, *ApJS*, **234**, 36
- McKinney, W. 2010, Proc. 9th Python in Science Conf. (SciPy 2010), ed. S. van der Walt & J. Millman, (Austin, TX: SciPy),
- Miller, J. P., Pennypacker, C. R., & White, G. L. 2008, *PASP*, **120**, 449
- Reiss, D. 2016, Dipole Characterization for Image Differencing DMTN-007 DMTN-007, Vera C. Rubin Observatory, <https://dmtn-007.lsst.io/>
- Reiss, D. J. 2017, State of Image Subtraction in the LSST Stack DMTN-061, Vera C. Rubin Observatory, <https://dmtn-061.lsst.io/>
- Ridgway, S. T., Matheson, T., Mighell, K. J., Olsen, K. A., & Howell, S. B. 2014, *ApJ*, **796**, 53
- Rowe, B. T. P., Jarvis, M., Mandelbaum, R., et al. 2015, *A&C*, **10**, 121
- Sako, M., Bassett, B., Becker, A., et al. 2008, *AJ*, **135**, 348
- Sánchez, B. O., Kessler, R., Scolnic, D., et al. 2022, *ApJ*, **934**, 96
- Seghouane, A.-K., & Shokouhi, N. 2018, *ISPL*, **25**, 521
- Shi, X., Guo, Z., Nie, F., et al. 2016, *ITPAM*, **38**, 2130
- The pandas development team, 2023 pandas-dev/pandas: v2.0.3, Zenodo, doi: 10.5281/zenodo.8092754
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *NatMe*, **17**, 261
- Zackay, B., Ofek, E. O., & Gal-Yam, A. 2016, *ApJ*, **830**, 27