MNRAS 000, 1–?? (2019) Preprint 15 September 2019 Compiled using MNRAS LATEX style file v3.0

Anomaly Detection in Hyper Suprime-Cam Images with Generative Adversarial Networks

Kate Storey-Fisher[1]★, Alexie Leauthaud[2], Marc Huertas-Company[3,4]
[1] Center for Cosmology and Particle Physics, Department of Physics, New York University, NY, USA
[2] Department of Astronomy and Astrophysics, University of California, Santa Cruz, Santa Cruz, CA, USA
[3] Instituto de Astrofísica de Canarias (IAC); Departamento de Astrofísica, Universidad de La Laguna (ULL), E-38200, La Laguna, Spain
[4] LERMA, Observatoire de Paris, CNRS, PSL, Université Paris Diderot, France

Last updated 2019 September 15

ABSTRACT

The problem of anomaly detection in astronomical surveys is becoming increasingly important as data sets grow in size. Unsupervised learning presents an approach for identifying outliers in data sets without prior labelling of data or specification of the outliers. We present the results of an anomaly detection method using generative adversarial networks (GANs) on optical galaxy images in the Hyper Suprime-Cam (HSC) survey. We find that the GAN is able to generate realistic HSC-like galaxies and learn the distribution of the training data. We identify images which are less well-represented in the GAN's latent space, and thus are more anomalous with respect to the rest of the data. This approach identifies ∼0.4% of the objects as anomalies at the 3σ level. We further characterize these anomalies by type using dimensionality reduction and clustering with mixture models on the residuals of the real and GAN-reconstructed images. Our initial results include the identification of multiple galaxy mergers and tidal features, and objects indicating AGN activity or central starbursts. We also identify a significant amount of "bad images", due to pipeline errors or bad detections, which can be used to improve the HSC pipeline. We conclude that GANs are a productive method for identifying anomalies in galaxy surveys. We will release a catalog of identified anomalous objects in the HSC survey in upcoming work. The code and catalog are available at `github.com/kstoreyf/anomalies-GAN-HSC`.

1 INTRODUCTION

Many discoveries in astronomy have been made by identifying unexpected outliers in collected data (e.g. Cardamone et al. 2009, Massey et al. 2019). These outliers, also referred to as anomalies or novelties, are data points that lie outside of the normal distribution of data. In the astronomy context, we are interested in finding unknown classes of objects, objects belonging to rare classes, and individual objects of known type with anomalous properties. As data sets increase in size, automated methods for detecting these outliers are becoming necessary. The Sloan Digital Sky Survey (SDSS) surveyed one third of the sky and observed over 1 billion cataloged objects (York et al. 2000). In the near future, the Large Synoptic Survey Telescope (LSST) will observe 40 billion objects (Ivezic et al. 2018). These present opportunities for discovery in their massive data sets, as well as the need for new, automated methods to filter the data and identify anomalies.

Outlier identification has been an area of study since as early as the 19th century (Edgeworth 1887). More recent work in anomaly detection for astronomy has applied a range of statistical and computational techniques. A nearest neighbors approach, often combined with a dimensionality reduction step, has been used for outlier detection in and cross-matched astronomical data sets more generally (Henrion et al. 2013). Applications often target specific objects, such as using Bayesian model selection to select rare high-redshift quasars from a star-dominated population (Mortlock et al. 2012). Another approach is Principal Component Analysis (PCA) to identify distinguishing features, which has been applied to SDSS and 2MASS flux and surface brightness data (Dutta et al. 2007).

Machine learning methods are being rapidly developed as approaches to anomaly detection in astronomy and other fields. A review of anomaly detection methods and applications using deep learning is presented in Chalapathy & Chawla (2019). Unsupervised learning lends itself to this problem, as it allows for outlier identification without expert labelling of training data or introducing biases based on expected outliers. Baron & Poznanski (2017) use random forests to find outliers in Sloan Digital Sky Survey (SDSS) spectroscopic data. Solarz et al. (2017) apply support vector machines to find anomalies in the Wide-field Infrared Survey Explorer (WISE) survey. Beyond galaxy surveys, deep learning has been applied to anomaly detection problems in supernovae data (Pruzhinskaya et al. 2019).

Deep generative models present another class of approaches to anomaly detection. These have a natural application to identifying outliers, as they are able to model complex distributions of high-dimensional data. One model class is generative adversarial networks (GANs), proposed by Goodfellow et al. (2014). They consist of a convolutional network known as a "discriminator" and a

deconvolutional network known as a "generator." The role of the generator is to learn a model of the training data set, and generate realistic images follow the same distribution. The discriminator is tasked with determining whether an image is real, i.e. from the training set, or fake, i.e. from the generator. The generator will be able to better model images that are more common in the training set, and will perform worse on images that are more anomalous relative to the rest of the data. Similarly, the discriminator learns to identify real data, and thus contains information about whether an object is realistic or anomalous.

GANs were first applied to anomaly detection by Schlegl et al. (2017), in the context of medical imaging. They demonstrate that a GAN trained on normal images can then be used to identify abnormal images. Zenati et al. (2018) show that training an encoder simultaneously with the GAN improves testing efficiency, and they demonstrate their performance on outlier detection tasks on a range of high-dimensional data. GANs have also been used to detect outliers in time-series data (Li et al. 2018). Di Mattia et al. (2019) present a survey of the application of GANs to anomaly detection and perform empirical validation of the models.

The Hyper Suprime-Cam Subaru Strategic Program (HSC-SSP) is a natural data set for anomaly detection applications Miyazaki et al. (2018). It is a wide-field optical survey with very good seeing and a deep magnitude limit, containing nearly half a billion primary objects. Many interesting objects have already been identified in HSC, including interacting galaxies (Goulding et al. 2017) and gravitationally lensed objects (Wong et al. 2018). We are interested in finding more of these types of objects, as well as potentially extreme color galaxies, galaxies with extreme activity, rare quasars, and other scientifically interesting objects. In addition to these, anomaly detection will be useful for filtering out instrumentation and pipeline errors in HSC.

In this work, we train a generative adversarial network to identify anomalous objects in a subsample of HSC images, of just under one million objects. We then characterize the anomalous images, to distinguish bad detections from interesting objects, and further classify these objects of interest. Our final work will present a complete catalog of anomalies in our HSC sample. To our knowledge, this is the first application of GANs to anomaly detection in astronomical data.

2 DATA

2.1 Hyper Suprime-Cam Survey

We use data from the Hyper Suprime-Cam Subaru Strategic Program. The wide-field optical survey is imaged with the Subaru Telescope and has been ongoing since March 2014. The second public data release (PDR2, Aihara et al. 2014) contains over 430 million primary objects in the wide field covering 1114 deg^2. The area observed in full-depth full-color covers 305 deg^2. These objects are observed in 5 broad-band filters, *grizy*, down to a magnitude limit of ~26.

2.2 Selection of Sample

We choose a magnitude slice for our analysis, with $20.0 < i < 20.5$. This allows for a more consistent sample in object size. We exclude objects flagged as having significant issues by the pipeline. These are, in any band: cosmic rays crossing the center pixel, saturated center pixel, interpolated center pixel, source at edge of survey volume, failed flux fit. The full query, including these cuts and the information we retain about each sample, is reproduced in appendix A.

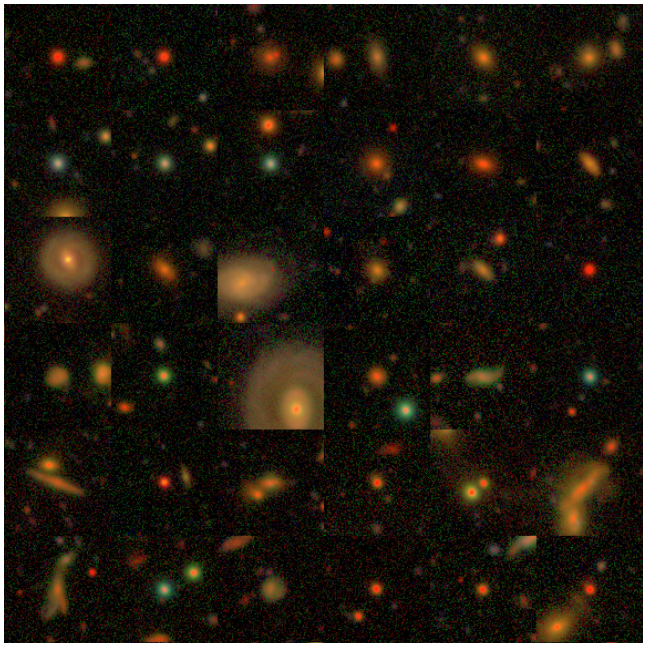

Figure 1. A subsample of the data sample used for training the GAN and identifying anomalies.

We generate cutouts of $96x96$ pixels around each image, with a pixel scale of 0.167 arcseconds; the cutouts are about $15x15$ arcseconds or 3 times the average effective radius in side length. This captures the entirety of most objects while still being a reasonable size for training the network, without the need for downsampling. We use the *gri*-bands to get 3-color images. This results in a sample of 942,782 objects, consisting of ~70% extended objects and ~30% compact objects. A subsample from our final training data selection is shown in Figure 1.

We perform some preprocessing on the images before feeding them to the neural network. We normalize the pixel values to avoid issues due to the raw data range spanning multiple orders of magnitude. We convert the flux values to RGB values using the method of Lupton (Lupton et al. 2003), and then convert these to between 0 and 1.

3 METHODS

3.1 GAN Architecture and Training

We construct a generative adversarial network based on the implementation by Gulrajani et al. (2017). The basic setup is a generator and a discriminator with separate loss functions, which compete against each other in a minimax game. The discriminator learns to distinguish real images from those generated by the generator. The generator, in turn, learns how realistic its generated images are based on feedback from the discriminator. The loss function to optimize is then

$$\min_G \max_D L(D,G) = \mathbb{E}_{x\sim p_{\text{real}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_{\text{latent}}(z)}[\log(1 - D(G(z)))] \quad (1)$$

GANs are notorious for instability in training; balancing the

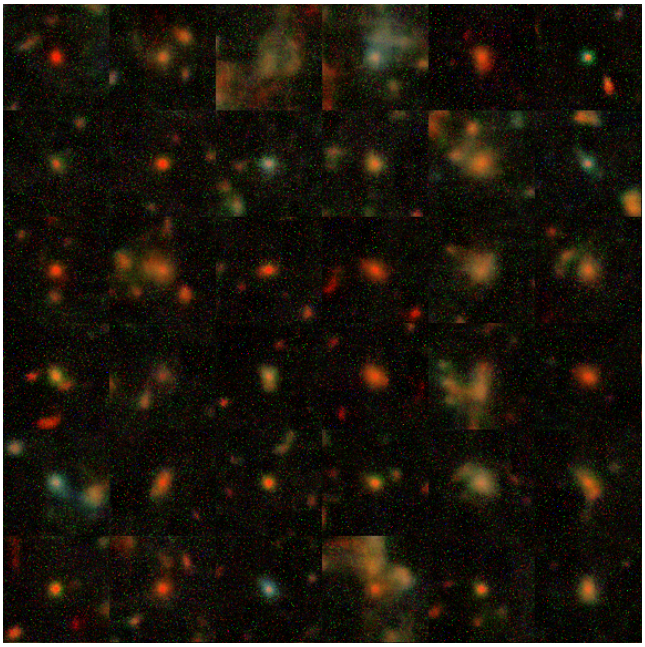

Figure 2. A random sample of images generated by the GAN, each starting from random noise.

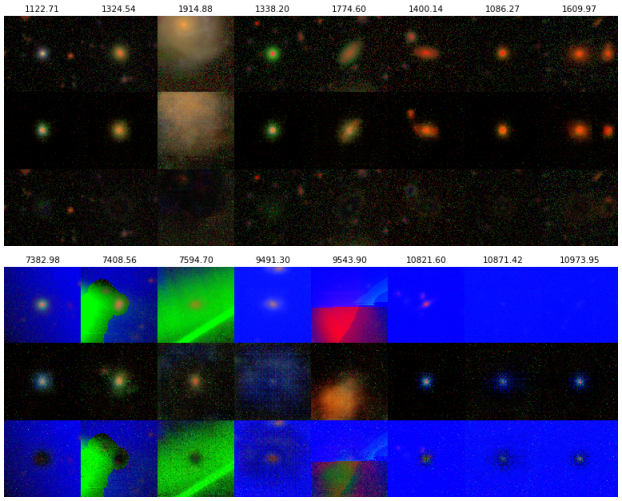

Figure 3. Objects and their GAN reconstructions. Top row is the real image, middle row is the reconstruction, and bottom row is the residual; each column is labelled by anomaly score. Left panel: A random sample of objects. Right panel: The most anomalous objects in the sample.

generator and discriminator losses is nontrivial. One improvement to this loss function to use the Wasserstein distance to compute the distance between probability distributions (Arjovsky et al. 2017). This is a more meaningful distance measure and is smooth even when the distributions are disjoint. Another issue is caused by the gradient vanishing when the discriminator is perfect, so the loss function cannot continue to be updated. One approach to solve this is by applying a "gradient penalty" (GP) to penalize the loss. These two improvements are known as a WGAN-GP, and we use this construction in our GAN. The implementation is in `tensorflow` and `python`.

We construct our generator to have a depth of 4 with a latent space of dimension 128, and a sigmoid activation function. The discriminator also has a depth of 4. We train the GAN in batches of 32 images, with 5 discriminator updates per generator update. After around 10,000 training iterations, the generator and discriminator losses stabilize and no longer improve. We thus select the generator and discriminator models at 10,000 training iterations. A random sample of images generated with this GAN, starting from random noise, is shown in Figure 2. We can see that the GAN is able to generate realistic images for less extended objects. However, it also generates diffuse-looking sources that are not realistic models of extended sources.

3.2 Anomaly Scores

We apply to generator to generate its best reconstruction of each image in the sample. To do this, we assign each attempted reconstruction a generator score and a discriminator score. The generator score $s_{\rm gen}$ is the mean square error of the difference between the pixels of the two images, summed over all bands. The discriminator also provides a way of measuring how anomalous an image is, as more anomalous images will more likely be marked as "fake" by the discriminator. To capture this, we output the image representation after the penultimate layer of the discriminator. The mean square error between this representation for the real and generated image is the discriminator score, $s_{\rm disc}$. The total anomaly score $s_{\rm total}$ is a weighted average of these,

$$s_{\rm total} = (1-\lambda)\cdot s_{\rm gen} + \lambda \cdot s_{\rm disc} \quad (2)$$

where λ is a weighting hyperparameter which we tune.

The generator attempts to generate a reconstruction with the minimal anomaly score. To speed up this process, we first train an encoder for the whole training sample. This convolutional network makes a first approximation of the 128-dimensional latent-space vector of the generator. We then perform a basic optimization for each image individually to reach a lower anomaly score, optimizing for 10 iterations. We note that the score converges before 10 for most images. The score after this process for each image is assigned as its final anomaly score. The scores are entirely relative and are meaningful only with respect to the rest of the sample. The result of this process is shown in Figure 3.

3.3 Classification of Anomalies

We expect the residual images, the absolute difference between the real and reconstructed images, to contain information about why the GAN marked an object as anomalous. We can project the residual images into a 2-dimensional representation using a Uniform Manifold Approximation and Projection (UMAP, McInnes et al. 2018). This organizes data by similarity for visualization and clustering purposes.

For the mapped quantities, we next perform clustering using a Gaussian mixture model (GMM). The GMM, which is similar to k-means clustering but can account for the variance of the data and performs soft classification, assigns the vectors to clusters based on their distance in the space of the UMAP. We can then visually inspect the clusters and determine which contain interesting objects.

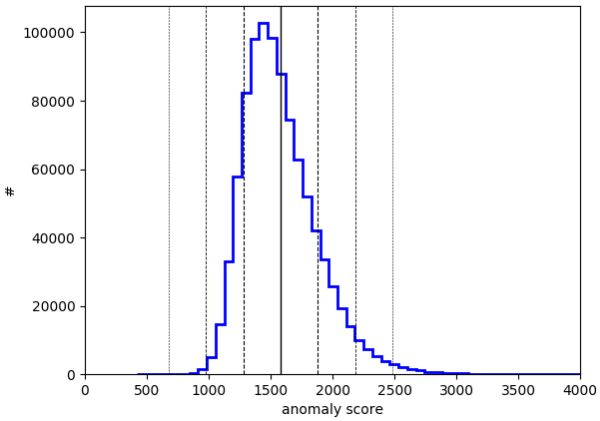

Figure 4. The distribution of anomaly scores for the ~940,000 objects in our sample. The solid line shows the sample mean; the dashed lines show 1σ , 2σ , and 3σ away from the mean. There are 80 objects with scores greater than 4000 not shown on the figure for clarity.

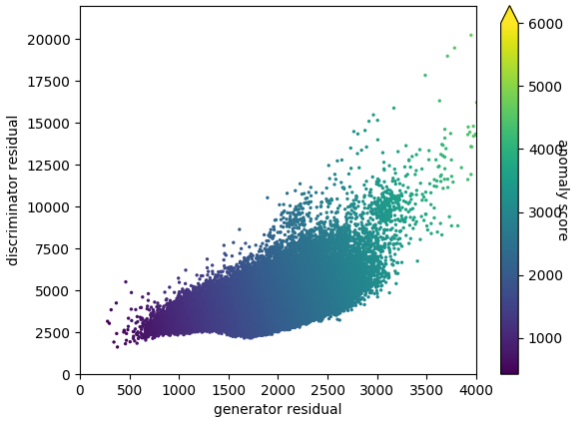

Figure 5. Discriminator vs. generator scores, based on the residuals from each network, for our sample. Objects with a generator score over 4000 are not shown.

4 RESULTS & DISCUSSION

4.1 Anomaly Score Distribution

We compute anomaly scores for each of the ~940,000 objects in our sample; their distribution is shown in Figure 4. Higher anomaly scores indicate more anomalous objects, while lower scores indicate objects that are more well-modeled by the GAN. The distribution is skewed towards higher scores, which is expected: most typical objects are reconstructed well by the GAN so have similar scores, while there are more ways to be anomalous than to be typical, resulting in a wider range of scores. There are 18 objects with extremely high anomaly scores, between 4,000 and 11,000, not shown on the figure for clarity. There are 9,648 objects with scores greater than 3σ above the mean, just over 1% of the sample; we take these as our "anomalies" and perform further classification on this sample.

We can understand the anomaly scores by comparing the assigned generator and discriminator scores. This is shown in Figure 5. As expected, objects with a higher generator score tend to have a higher discriminator score. We also see that there are populations of objects with one score relatively higher than the other, as well as clusters in this space. This may indicate particular classes of object, for example objects that the generator reproduces well in pixel space but contains anomalous features or patterns in the latent space of the discriminator. We plan to explore these regions in upcoming work.

4.2 UMAP Clustering

We apply the UMAP algorithm to the data to visualize it in a 2-dimensional space. We perform the embedding on the residual pixels of each image, as these contain information about the anomalous features of an image. Figure 6 shows the map for a random 100,000-object subsample (due to memory limitations, to be expanded in upcoming work), as well as the map for just the 3σ anomalies. We see significant structure in both embeddings. The higher anomaly score objects cluster in the center of the 100,000-object sample, showing that the residual pixels contain information about how anomalous the images are. The UMAP of 3σ anomalies shows significant clustering, with the very highest-scoring anomalies located in a few regions. There are also clusters offset from the majority of the objects, which contain a mix of anomaly scores but may contain objects of particular interest.

We can perform an initial exploration of this space using a GMM. We apply the GMM to the 2-dimensional mapping of each object in the 3σ anomaly sample, with 10 clusters. This choice of 10 is arbitrary, chosen to explore the data space in some level of detail; future work will refine this approach. The objects are color-coded by cluster assignment in Figure 7. We can then look at the images, as well as their GAN reconstruction and residual, in each cluster; a random subsample of objects from each cluster is shown in Figure 8. The clusters are visually distinct, most obviously by color. There are also patterns in the residuals with companion objects and extended objects that are not well-captured by the GAN. This shows that clustering of embedded-space residuals is providing useful information on anomaly type; we plan on performing more fine-grained clustering and classification with this technique.

We note that residual image pixel space on which we applied the UMAP is very high-dimensional, and may contain information less relevant to the anomalies. To reduce the dimensionality of the data and isolate the relevant information, we trained a convolutional autoencoder to map each residual image to a 64-dimensional vector. We then applied a UMAP to this vector. However, this approach did not, on initial inspection, produce better results than mapping directly on the residual pixels. We plan to return to this autoencoder approach in future work.

4.3 Correlation with HSC Catalog Information

We can compare the results of our anomaly detection process with the HSC catalog data. This acts as both a validation step, and a way to further cluster the data. We first look at the extendedness of the object, measured by its effective radius, which is computed from the moments of the flux fit. The radius as a function of anomaly score is shown in Figure 9 (left panel). We note that some objects were computed to have unrealistically high radii, so these values may not be entirely trustworthy. We see that more extended objects tend to be more anomalous. However, this effect isn't very strong;

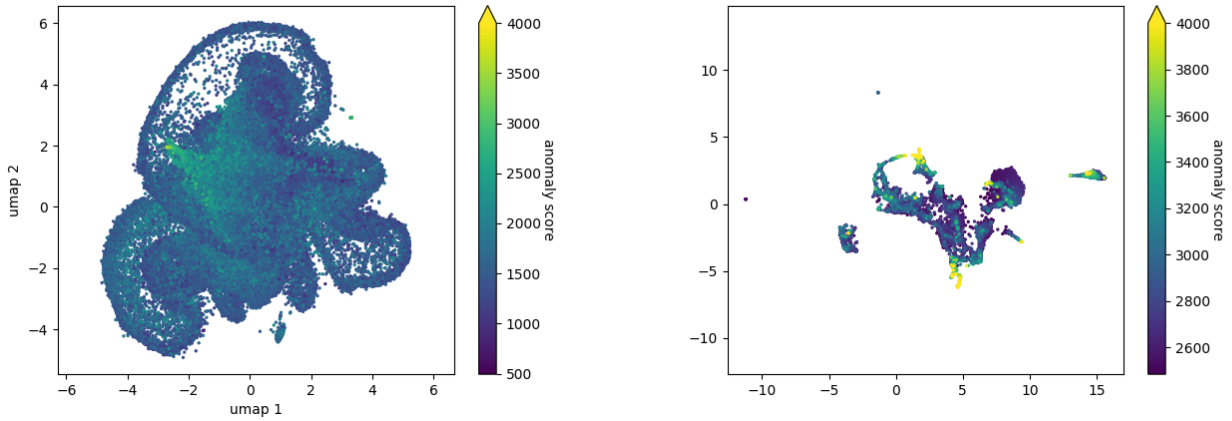

Figure 6. UMAP embeddings of the residual image pixels. Left panel shows the UMAP for a random 100,000 object subsample. Right panel shows only the 3σ anomalies. These are color-coded by anomaly score.

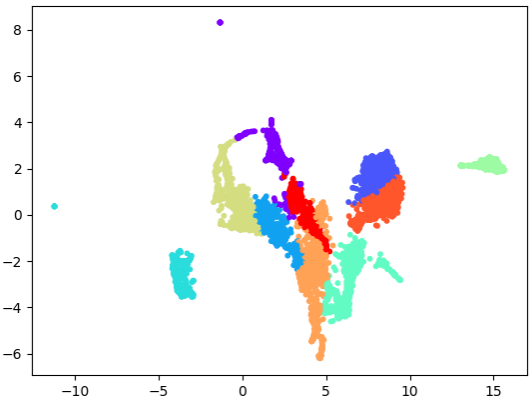

Figure 7. The same UMAP embedding of the residual image pixels for the 3σ anomalies as in the right panel of Figure reffig:umap, color-coded by GMM-assigned cluster.

most extended objects fall below the 2σ anomaly cut, including most of the largest objects. This means that the anomaly detector is not simply flagging every extended object as anomalous; it is able to recognize normal extended objects, as well as anomalous compact objects.

We also look at the blendedness of the object, which describes the contamination of one object by the light from other close objects (Figure 9, right panel). A blendedness of 0 indicates an isolated object, while a value near 1 indicates a very blended object. The red line at $10^{-0.375}$ at shows the cutoff suggested for eliminating extreme blends; 4.6% of the objects in our sample are above this threshold, while 9.9% of 3σ anomalies are above it. (Mandelbaum et al. 2018). We see a trend in anomaly score with blendedness: more blended objects are more likely to be anomalous. As in the extended case, even the objects with high blending are mostly less than 2σ anomalies. There is still a significant number of 3σ anomalies below the cutoff; we plan to investigate this sample in upcoming work.

4.4 Identified Anomalies

As a result of this process, we have identified several interesting anomalous objects. Some of these are shown in Figure 10. The images on the left are galaxies that have tidal features or are currently undergoing mergers. The images on the right are objects that contain anomalously blue cores. These could potentially indicate active galactic nucleus (AGN) activity, or a central starburst (e.g. Menanteau et al. 2005). These anomalies were found with a combination of the dimensionality reduction and clustering approaches detailed above, along with a cursory visual inspection. We expect to find interesting anomalies in a less supervised way in upcoming work, with improved clustering and incorporation of metadata, though visual inspection will likely still be required as a final step. For these and other anomalies found, we plan to see if these objects have been observed in other surveys, and potentially perform follow-up on particularly interesting objects.

5 SUMMARY & CONCLUSIONS

We searched for anomalous objects in a sample of ~940,000 Hyper Suprime-Cam images. We used a generative adversarial network (GAN) to build a generative model of our data space. This allowed us to identify images that are poorly represented by the model and thus more anomalous with respect to the rest of the data.

Our conclusions are as follows:

- Our trained GAN is able to generate mostly realistic images that represent the majority of the data.
- We find 1% of objects to be anomalous at the 3σ level, as determined by the residuals of the generator and discriminator for the closest image in the GAN's latent space.
- A dimensionality reduction process and clustering algorithm shows that the residual image pixels contain information about the type of anomaly, which are distinguished by both color and shape.
- There are weak correlations between an object's anomaly score and its extendedness and blendedness. There is still a significant number of anomalous objects that are compact and or isolated.
- We identify, through this process as well as initial visual inspection, a number of anomalies that fall into two categories: galaxies

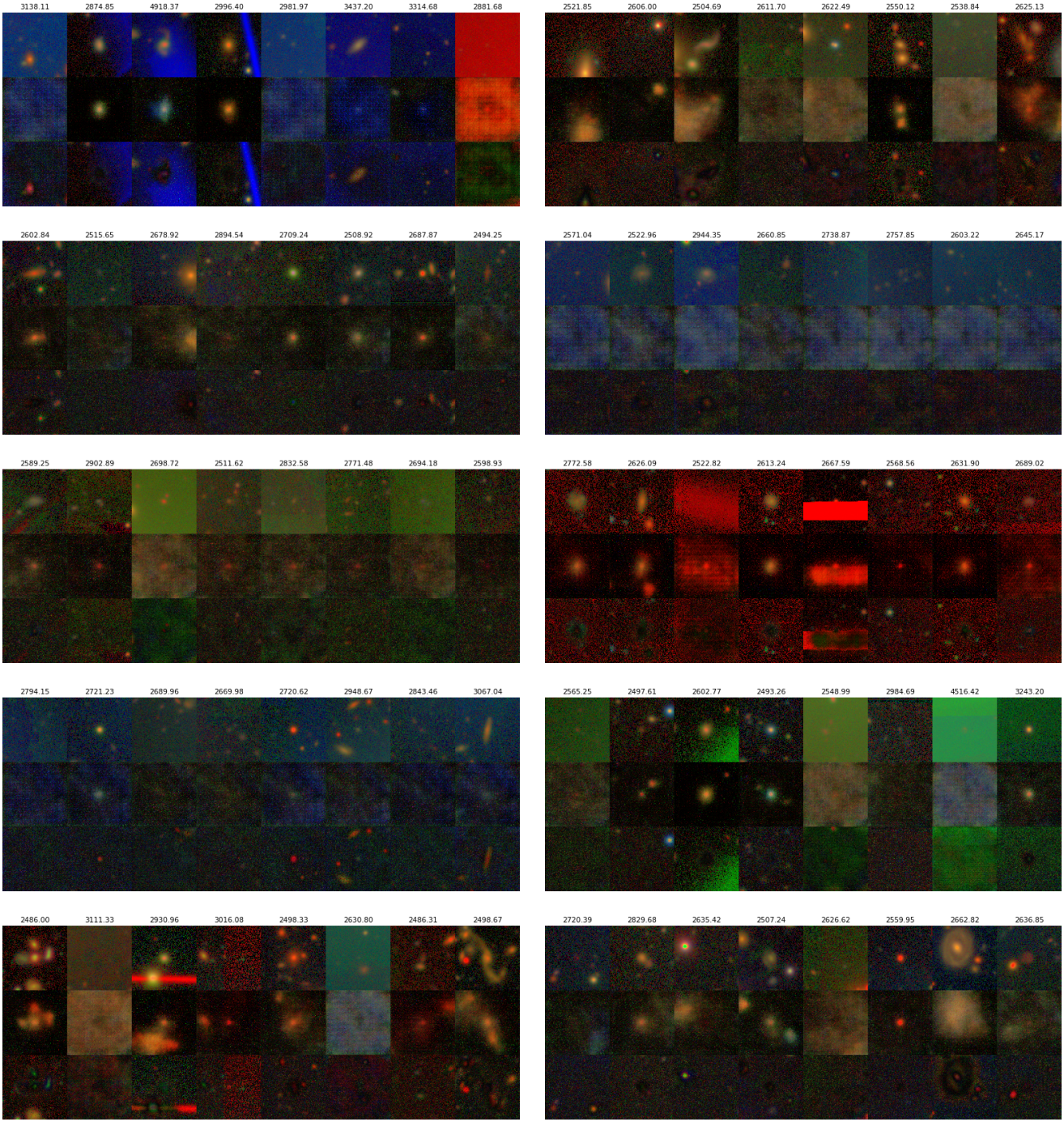

Figure 8. Clusters of the objects in the UMAP. Each of the 10 clusters corresponds to a different color cluster in Figure 7. Each column shows the real image at the top, the reconstruction in the middle, and the residual at the bottom, labelled with its anomaly score.

with tidal features and galaxy mergers, and objects with a blue core that indicate AGN or central starburst activity.

We plan to make improvements, address current issues, and extend this work as follows:

- Improve the construction and training of the GAN to better model the full data space and thus identify more accurate set of anomalies.
- Refine approach to clustering of anomalies. Revisit autoencoder to reduce dimensionality of residual images, and perform more principled clustering on the results.
- Use additional information to characterize anomalies, including generator and discriminator scores, catalog information such as blendedness and extendedness, and color information.
- Perform anomaly detection on other magnitude slices to get a complete sample of anomalies in the HSC catalog.
- Compare identified anomalies to previously published interesting objects in HSC, as well as to other anomaly detection methods.

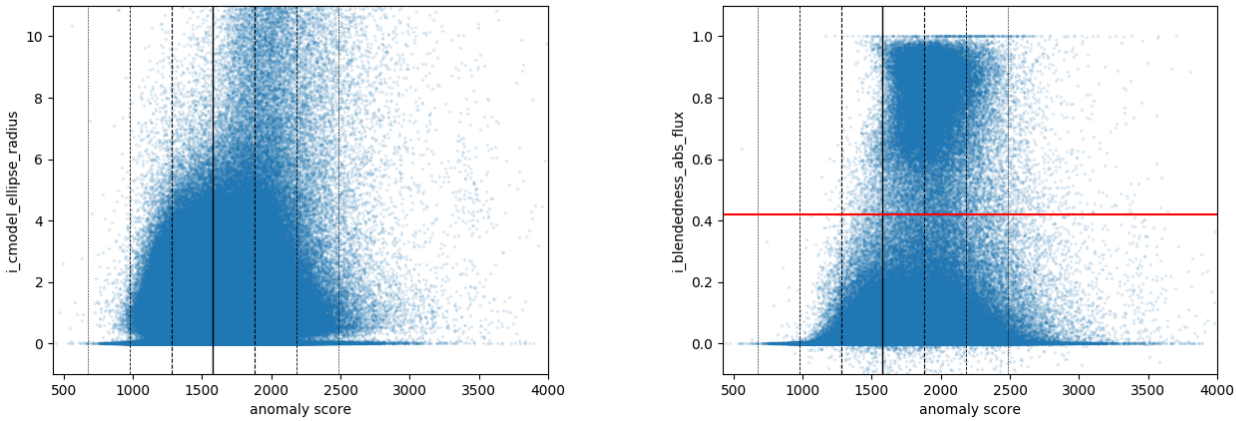

Figure 9. Left panel: Effective radius of the objects as a function of their anomaly score. Right panel: Blendedness of the objects as a function of their anomaly score. The red line is the cutoff value for most science use cases. The black solid lines show the sample mean; the dashed lines show 1σ , 2σ and 3σ away from the mean. There are 80 objects with scores greater than 4000 not shown on the figures for clarity.

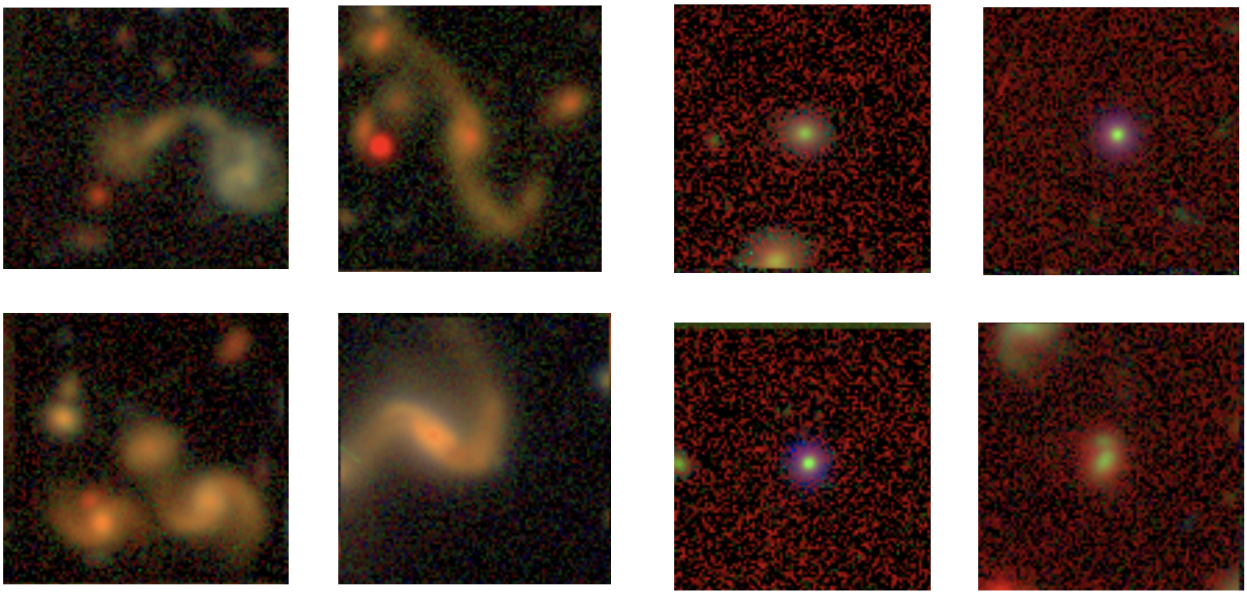

Figure 10. An initial selection of anomalies detected in our sample. Left panel: Galaxy mergers and tidal features. Right panel: Potential blue-core objects, which may indicate AGN or central starburst.

For an updated list of completed and ongoing tasks, see `github.com/kstoreyf/anomalies-GAN-HSC/issues`.

ACKNOWLEDGEMENTS

We gratefully acknowledge the Kavli Summer Program in Astrophysics for seeding this project; the initial work was completed at the 2019 program at the University of California, Santa Cruz. This work was funded by the Kavli Foundation, the National Science foundation, and UC Santa Cruz. KSF thanks Song Huang, Francois Lanusse, Nesar Ramachandra, Dezso Ribli, and Lorenzo Zanisi for helpful discussions.

REFERENCES

Aihara H., et al., 2014, Astronomical Society of Japan, 00, 1

Arjovsky M., Chintala S., Bottou L., 2017, Technical report, Wasserstein GAN, https://arxiv.org/pdf/1701.07875.pdf. (arXiv:1701.07875v3), https://arxiv.org/pdf/1701.07875.pdf

Baron D., Poznanski D., 2017, Monthly Notices of the Royal Astronomical Society, 465, 4530

Cardamone C., et al., 2009, Monthly Notices of the Royal Astronomical Society, 399, 1191

Chalapathy R., Chawla S., 2019, Technical report, Deep Learning for Anomaly Detection: A Survey, https://arxiv.org/pdf/1901.03407.pdfhttp://arxiv.org/abs/1901.03407. (arXiv:1901.03407), https://arxiv.org/pdf/1901.03407.pdfhttp://arxiv.org/abs/1901.03407

Di Mattia F., Galeone P., De Simoni M., Ghelfi E., 2019
Dutta H., Giannella C., Borne K., Kargupta H., 2007, in Proceedings of the 2007 SIAM International Conference on Data Mining. pp 473–478, http://www.siam.org/journals/ojsa.php
Edgeworth F. Y., 1887, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 23, 364
Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014
Goulding A. D., et al., 2017, Publ. Astron. Soc. Japan, pp 1–26
Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., Courville A., 2017, Technical report, Improved Training of Wasserstein GANs, https://github.com/igul222/improved{_}wgan{_}training.. (arXiv:1704.00028v3), https://github.com/igul222/improved{_}wgan{_}training.
Henrion M., Mortlock D. J., Hand D. J., Gandy A., 2013, Classification and Anomaly Detection for Astronomical Survey Data. Springer New York, New York, NY, pp 149–184, doi:10.1007/978-1-4614-3508-2_8, https://doi.org/10.1007/978-1-4614-3508-2{_}8
Ivezic Z., et al., 2018, Technical report, LSST: from Science Drivers to Reference Design and Anticipated Data Productš, https://arxiv.org/pdf/0805.2366.pdf. (arXiv:0805.2366v5), https://arxiv.org/pdf/0805.2366.pdf
Li D., Chen D., Goh J., Ng S.-K., 2018, in Workshop on Big Data, Streams and Heterogeneous Source Mining. (arXiv:1809.04758), https://github.com/LiDan456/GAN-ADhttp://arxiv.org/abs/1809.04758
Lupton R., Blanton M. R., Fekete G., Hogg D. W., O'mullane W., Szalay A., Wherry N., 2003, Technical report, Preparing Red-Green-Blue (RGB) Images from CCD Data, http://www.astro.princeton.edu/âĹijrhl/PrettyPictures.. http://www.astro.princeton.edu/âĹijrhl/PrettyPictures.
Mandelbaum R., et al., 2018, Astronomical Society of Japan, 70
Massey P., Neugent K. F., Levesque E. M., 2019, The Astronomical Journal
McInnes L., Healy J., Melville J., 2018, Technical report, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, https://arxiv.org/pdf/1802.03426.pdf. (arXiv:1802.03426v2), https://arxiv.org/pdf/1802.03426.pdf
Menanteau F., et al., 2005, The Astrophysical Journal, 620, 697
Miyazaki S., et al., 2018, Publications of the Astronomical Society of Japan, 70, 1
Mortlock D. J., Patel M., Warren S. J., Hewett P. C., Venemans B. P., Mcmahon R. G., Simpson C., 2012, Monthly Notices of the Royal Astronomical Society, 419, 390
Pruzhinskaya M. V., Malanchev K. L., Kornilov â. M. V., Ishida E. E. O., Mondon F., Volnova A. A., Korolev V. S., 2019, Technical report, Anomaly Detection in the Open Supernova Catalog, http://snad.space/osc/. (arXiv:1905.11516v2), http://snad.space/osc/
Schlegl T., Seeböck P., Waldstein S. M., Schmidt-Erfurth U., Langs G., 2017, in Information Processing in Medical Imaging. (arXiv:1703.05921v1), doi:10.1007/978-3-319-59050-9_12, https://arxiv.org/pdf/1703.05921.pdf
Solarz A., Bilicki M., Gromadzki M., Pollo A., Durkalec A., Wypych M., 2017, Astronomy and Astrophysics, 606
Wong K. C., et al., 2018, The Astrophysical Journal, 867
York D. G., et al., 2000, The Astronomical Journal, 120, 1579
Zenati H., Romain M., Foo C. S., Lecouat B., Chandrasekhar V., 2018, in Proceedings - IEEE International Conference on Data Mining, ICDM. pp 727–736 (arXiv:1812.02288v1), doi:10.1109/ICDM.2018.00088, https://arxiv.org/pdf/1812.02288.pdf

APPENDIX A: CATALOG QUERY

We reproduce the SQL query used to select our data sample in the HSC catalog. This includes the cuts, in magnitude and on flags, as well as information about the images and objects. The query can be run through the HSC data access site at https://hsc-release.mtk.nao.ac.jp/datasearch/.

```
SELECT
    -- Basic information
    f1.object_id, f1.ra, f1.dec, f1.tract,
        f1.patch, f1.parent_id,

    -- Galactic extinction
    f1.a_g, f1.a_r, f1.a_i, f1.a_z, f1.a_y,

    --- cmodel
    ---- Total
    f1.g_cmodel_mag, f1.r_cmodel_mag, f1.
        i_cmodel_mag, f1.z_cmodel_mag, f1.
        y_cmodel_mag,
    f1.g_cmodel_magsigma, f1.
        r_cmodel_magsigma, f1.
        i_cmodel_magsigma, f1.
        z_cmodel_magsigma, f1.
        y_cmodel_magsigma,

    ---- fracDev
    f1.g_cmodel_fracdev, f1.r_cmodel_fracdev
        , f1.i_cmodel_fracdev, f1.
        z_cmodel_fracdev, f1.
        y_cmodel_fracdev,

    ---- flag
    f1.g_cmodel_flag, f1.r_cmodel_flag, f1.
        i_cmodel_flag, f1.z_cmodel_flag, f1.
        y_cmodel_flag,

    --- PSF
    f2.g_psfflux_mag, f2.r_psfflux_mag, f2.
        i_psfflux_mag, f2.z_psfflux_mag, f2.
        y_psfflux_mag,
    f2.g_psfflux_magsigma, f2.
        r_psfflux_magsigma, f2.
        i_psfflux_magsigma, f2.
        z_psfflux_magsigma, f2.
        y_psfflux_magsigma,

    ---- flag
    f2.g_psfflux_flag, f2.r_psfflux_flag, f2
        .i_psfflux_flag, f2.z_psfflux_flag,
        f2.y_psfflux_flag,

    -- Flags
    ---- pixel edge
    f1.g_pixelflags_edge, f1.
        r_pixelflags_edge, f1.
        i_pixelflags_edge, f1.
        z_pixelflags_edge, f1.
        y_pixelflags_edge,

    ---- pixel interpolated
    f1.g_pixelflags_interpolated, f1.
        r_pixelflags_interpolated, f1.
        i_pixelflags_interpolated, f1.
        z_pixelflags_interpolated,
    f1.y_pixelflags_interpolated,
```

```
    ---- pixel saturated
    f1.g_pixelflags_saturated, f1.
        r_pixelflags_saturated, f1.
        i_pixelflags_saturated, f1.
        z_pixelflags_saturated,
    f1.y_pixelflags_saturated,

    ---- pixel cr
    f1.g_pixelflags_cr, f1.r_pixelflags_cr,
        f1.i_pixelflags_cr, f1.
        z_pixelflags_cr, f1.y_pixelflags_cr,

    ---- pixel clipped
    f1.g_pixelflags_clipped, f1.
        r_pixelflags_clipped, f1.
        i_pixelflags_clipped, f1.
        z_pixelflags_clipped,
    f1.y_pixelflags_clipped,

    ---- pixel reject
    f1.g_pixelflags_rejected, f1.
        r_pixelflags_rejected, f1.
        i_pixelflags_rejected, f1.
        z_pixelflags_rejected,
    f1.y_pixelflags_rejected,

    ---- pixel inexact psf
    f1.g_pixelflags_inexact_psf, f1.
        r_pixelflags_inexact_psf, f1.
        i_pixelflags_inexact_psf,
    f1.z_pixelflags_inexact_psf, f1.
        y_pixelflags_inexact_psf,

    ---- pixel interpolated center
    f1.g_pixelflags_interpolatedcenter, f1.
        r_pixelflags_interpolatedcenter, f1.
        i_pixelflags_interpolatedcenter,
    f1.z_pixelflags_interpolatedcenter, f1.
        y_pixelflags_interpolatedcenter,

    ---- pixel saturated center
    f1.g_pixelflags_saturatedcenter, f1.
        r_pixelflags_saturatedcenter, f1.
        i_pixelflags_saturatedcenter, f1.
        z_pixelflags_saturatedcenter,
    f1.y_pixelflags_saturatedcenter,

    ---- pixel cr center
    f1.g_pixelflags_crcenter, f1.
        r_pixelflags_crcenter, f1.
        i_pixelflags_crcenter, f1.
        z_pixelflags_crcenter, f1.
        y_pixelflags_crcenter,

    ---- pixel clipped center
    f1.g_pixelflags_clippedcenter, f1.
        r_pixelflags_clippedcenter, f1.
        i_pixelflags_clippedcenter, f1.
        z_pixelflags_clippedcenter,
    f1.y_pixelflags_clippedcenter,

    ---- pixel reject center
    f1.g_pixelflags_rejectedcenter, f1.
        r_pixelflags_rejectedcenter, f1.
        i_pixelflags_rejectedcenter, f1.
        z_pixelflags_rejectedcenter,
    f1.y_pixelflags_rejectedcenter,

    ---- pixel inexact psf center
    f1.g_pixelflags_inexact_psfcenter, f1.
        r_pixelflags_inexact_psfcenter, f1.
        i_pixelflags_inexact_psfcenter,
    f1.z_pixelflags_inexact_psfcenter, f1.
        y_pixelflags_inexact_psfcenter,

    ---- pixel bright object
    f1.g_pixelflags_bright_object, f1.
        r_pixelflags_bright_object, f1.
        i_pixelflags_bright_object,
    f1.z_pixelflags_bright_object, f1.
        y_pixelflags_bright_object,

    ---- pixel bright object center
    f1.g_pixelflags_bright_objectcenter, f1.
        r_pixelflags_bright_objectcenter, f1
        .i_pixelflags_bright_objectcenter,
    f1.z_pixelflags_bright_objectcenter, f1.
        y_pixelflags_bright_objectcenter,

        -- Measured information
        ---- blendedness
        m.g_blendedness_abs_flux, m.
            g_blendedness_flag,
        m.r_blendedness_abs_flux, m.
            r_blendedness_flag,
        m.i_blendedness_abs_flux, m.
            i_blendedness_flag,
        m.z_blendedness_abs_flux, m.
            z_blendedness_flag,
        m.y_blendedness_abs_flux, m.
            y_blendedness_flag,

        ---- Shape of the CModel model
        m.i_cmodel_exp_ellipse_11, m.
            i_cmodel_exp_ellipse_22, m.
            i_cmodel_exp_ellipse_12,
        m.i_cmodel_dev_ellipse_11, m.
            i_cmodel_dev_ellipse_22, m.
            i_cmodel_dev_ellipse_12,
        m.i_cmodel_ellipse_11, m.
            i_cmodel_ellipse_22, m.
            i_cmodel_ellipse_12,
        m.r_cmodel_exp_ellipse_11, m.
            r_cmodel_exp_ellipse_22, m.
            r_cmodel_exp_ellipse_12,
        m.r_cmodel_dev_ellipse_11, m.
            r_cmodel_dev_ellipse_22, m.
            r_cmodel_dev_ellipse_12,
        m.r_cmodel_ellipse_11, m.
            r_cmodel_ellipse_22, m.
            r_cmodel_ellipse_12

    -- Meta information
```

```
    ---- input count
    f1.g_inputcount_value, f1.
        r_inputcount_value, f1.
        i_inputcount_value, f1.
        z_inputcount_value, f1.
        y_inputcount_value,

    ---- extendedness
    f1.g_extendedness_value, f1.
        r_extendedness_value, f1.
        i_extendedness_value, f1.
        z_extendedness_value, f1.
        y_extendedness_value,
    f1.g_extendedness_flag, f1.
        r_extendedness_flag, f1.
        i_extendedness_flag, f1.
        z_extendedness_flag, f1.
        y_extendedness_flag,

    ---- background
    f1.g_localbackground_flux, f1.
        r_localbackground_flux, f1.
        i_localbackground_flux,
    f1.z_localbackground_flux, f1.
        y_localbackground_flux

FROM
        pdr2_wide.forced AS f1
        LEFT JOIN pdr2_wide.forced2 AS f2
            USING (object_id)
        LEFT JOIN pdr2_wide.meas AS m USING
            (object_id)

WHERE

-- Select only primary targets
f1.isprimary = True
AND f1.nchild = 0

-- Rough FDFC cuts
AND f1.g_inputcount_value >= 3
AND f1.r_inputcount_value >= 3
AND f1.i_inputcount_value >= 3
AND f1.z_inputcount_value >= 3
AND f1.y_inputcount_value >= 3

-- Cuts on bright objects
AND NOT f1.g_pixelflags_bright_objectcenter
AND NOT f1.r_pixelflags_bright_objectcenter
AND NOT f1.i_pixelflags_bright_objectcenter
AND NOT f1.z_pixelflags_bright_objectcenter
AND NOT f1.y_pixelflags_bright_objectcenter

AND NOT f1.g_pixelflags_bright_object
AND NOT f1.r_pixelflags_bright_object
AND NOT f1.i_pixelflags_bright_object
AND NOT f1.z_pixelflags_bright_object
AND NOT f1.y_pixelflags_bright_object

-- Cuts on bad flags
AND NOT f1.g_pixelflags_edge
AND NOT f1.r_pixelflags_edge
AND NOT f1.i_pixelflags_edge
AND NOT f1.z_pixelflags_edge
AND NOT f1.y_pixelflags_edge

AND NOT f1.g_pixelflags_saturatedcenter
AND NOT f1.r_pixelflags_saturatedcenter
AND NOT f1.i_pixelflags_saturatedcenter
AND NOT f1.z_pixelflags_saturatedcenter
AND NOT f1.y_pixelflags_saturatedcenter

AND NOT f1.g_cmodel_flag
AND NOT f1.r_cmodel_flag
AND NOT f1.i_cmodel_flag
AND NOT f1.z_cmodel_flag
AND NOT f1.y_cmodel_flag

AND NOT f1.g_pixelflags_interpolatedcenter
AND NOT f1.r_pixelflags_interpolatedcenter
AND NOT f1.i_pixelflags_interpolatedcenter
AND NOT f1.z_pixelflags_interpolatedcenter
AND NOT f1.y_pixelflags_interpolatedcenter

AND NOT f1.g_pixelflags_crcenter
AND NOT f1.r_pixelflags_crcenter
AND NOT f1.i_pixelflags_crcenter
AND NOT f1.z_pixelflags_crcenter
AND NOT f1.y_pixelflags_crcenter

-- CModel magnitude limited
AND f1.i_cmodel_mag < 20.5
AND f1.i_cmodel_mag >= 20.0
```