Vera C. Rubin Observatory
Systems Engineering

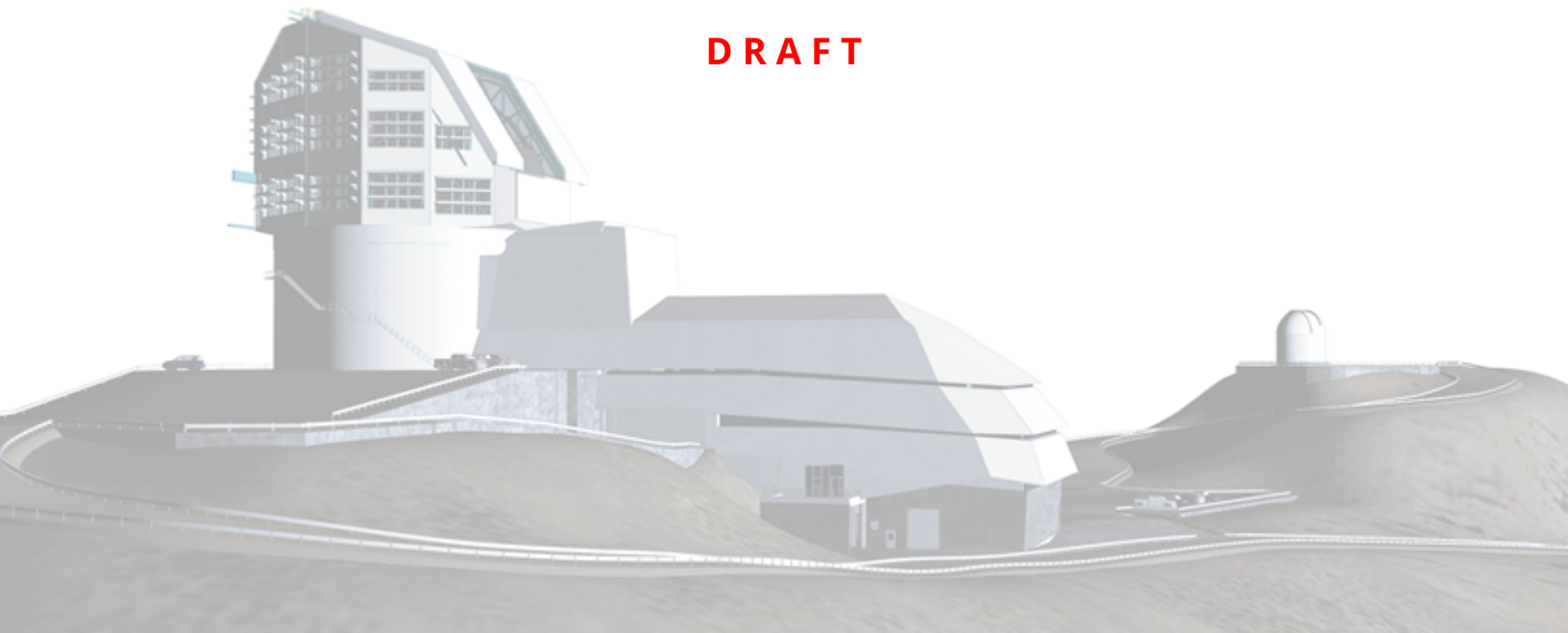
Initial studies of photometric redshifts with ComCam from DP1

Eric Charles, John Franklin Crenshaw, Tianqing Zhang, Sam Schmidt, Prakruth Adari, Melissa Graham

SITCOMTN-154

Latest Revision: 2025-06-30

DRAFT



Draft

Abstract

This technote holds reports based on the first analyses of the Data Preview 1 (DP1) ComCam data by the Science Unit for photometric redshifts. Although photometric redshifts are not a official DP1 data product, the "Photo-z Science Unit" generated photo-z estimates for every galaxy in DP1 using the available multi-band imaging on a best-effort basis. This work included developing training and test datasets by matching DP1 data to high-quality reference redshifts obtained with spectroscopy, Grism data, and multi-band photometry. The Science Unit used the RAIL software package to make photometric redshift estimates using eight different algorithms, developed simple scientific performance metrics, used those metrics to explore how the performance of the algorithms varied with configuration changes, derived more optimized configurations of the algorithms and tested the performance of those configurations. This work, the resulting data products and expected data distribution mechanism are all described there.

Draft

Change Record

Version	Date	Description	Owner name
1	2025-06-30	Initial release, coincident with release of DP1	Melissa Graham

Document source location: `https://github.com/lsst-sitcom/sitcomtn-154`

Draft

Contents

Draft

Initial studies of photometric redshifts with ComCam from DP1

A Data products

In the course of this work, we generated several data products, including redshift estimates and a variety of others that can be used to reproduce those estimates and to facilitate thorough evaluation of the algorithm performance. These products include: configuration files (Sec. A.1), ancillary inputs (Sec. A.2), trained models (Sec. A.3), redshift estimates (Sec. A.4), summary statistics (Sec. A.5, and performance monitoring plots (Sec. A.6), all of which are essential for understanding the quality of the photometric redshift estimates and for refining the algorithms. Together, these data products provide a comprehensive framework for conducting, managing, and evaluating photometric redshift estimation workflows in Rubin DP1. They ensure that the process is not only scientifically rigorous but also organized and reproducible, enabling effective collaboration and ongoing refinement of photometric redshift techniques.

A.1 Configuration files

The configuration files, collected in the GitHub `rail_project_config` repository, provide the necessary parameters and settings to control the various stages of the redshift estimation process, are a key element of `rail_projects` workflow. These files typically include specifications for the photometric bands used (e.g., g, r, i, z), the algorithm choices (e.g., template fitting or machine learning methods), details about data pre-processing, such as feature normalization and handling of missing data. These configuration files also specify the training and validation dataset splits, hyper-parameters for machine learning models, and paths for input/output data. These files are essential for ensuring reproducibility and for sharing the exact settings used in different redshift estimation runs, enabling other researchers to replicate or extend the analysis.

A.2 Ancillary input files

In addition to the configuration files, we require ancillary inputs such as the galaxy spectral energy distribution (SED) templates and the filter throughputs. Galaxy SED templates are collections of theoretical or observed spectra for galaxies at different redshifts and with dif-

ferent properties (e. g. , galaxy type, age, star formation history). These templates are used by template-fitting algorithms to model the expected galaxy colors as a function of redshift, allowing for the estimation of photometric redshifts by comparing observed colors to those predicted by the templates. Two SED sets are employed in this note, they are each described in Section **??**.

The filter throughputs specify the characteristics of the fiducial observational filter transmission curves, and are used in Rubin DP1, including their corresponding central wavelengths. These filter curves are employed in calculating the synthetic fluxes or magnitudes expected from each of the SED templates used in template-fitting algorithms, and for matching observational data to predicted theoretical values, e. g. the predicted colors shown in Figure **??**.

A.3 Estimator data models

After training, the trained models for each photometric redshift estimation algorithm are stored as serialized files, either in Pickle (for Python-based models) or YAML (for model configurations) format. These models encapsulate the learned relationships between photometric features (such as magnitudes and colors) and redshift values, allowing them to be applied to new data for redshift estimation. These files store the final state of the model, including the weights, biases, and other learned parameters for machine learning models. In the case of template-fitting methods, the corresponding model files may include the template sets and the fitting parameters. The Pickle or YAML format ensures that the models can be easily loaded, applied to new datasets, and evaluated in future studies.

A.4 Redshift estimates stored as QP ensembles

The per-object redshift estimates generated by the photometric redshift algorithms are stored in qp files. These files serve as containers for storing the redshift predictions for each object in the dataset before any detailed statistical analysis or final reporting. In the qp files, each galaxy's redshift estimate is stored in a format that is compatible with the algorithm providing the estimate. Specifically, for each object we store distribution $p(z)$, which, depending on the algorithm, maybe represent a posterior probability, a likelihood, a conditional likelihood, or just a hunch as to redshift of the object in question. These qp files are designed to be lightweight and easy to query, allowing users to quickly retrieve the redshift estimates for individual objects. The structure of qp files is optimized for efficient access and data retrieval,

making it easier for users to process and analyze large numbers of photometric redshift estimates across large datasets like Rubin DP1. Additional information, including usage examples, about qp and qp files is available at https://qp.readthedocs.io/en/main/.

A.5 Per-Object Point Estimates

In addition to the raw redshift estimates, the qp files also store per-object point estimates in the form of an ancillary table. These estimates include crucial information about the quality and reliability of each photometric redshift estimate, such as the uncertainty in the redshift prediction (e.g., the confidence interval or standard deviation or, the likelihood score (in the case of probabilistic models). This table will eventually includes flags for identifying objects with low-confidence estimates or those that may be outliers. These summary statistics are important for evaluating the overall performance of the redshift estimation algorithms on an object-by-object basis and are often used to filter out low-quality or problematic redshift estimates before conducting larger statistical analyses.

We also generate meta catalog file that collects the Object ID, magnitude information, and point estimates. The point estimates are named by "{algorithm}_z_{point estimate name}". Here is a list of the point estimates and summary statistics that are stored for each algorithm and the corresponding column names:

1. **mean**: is the per-object expectation of redshift given by the PDF.
2. **median**: is the 50% percentile of the PDF.
3. **mode**: is the redshift that yields maximum PDF evaluated on 301 grid points between $z = 0\text{-}3$.
4. **err68_lower(upper)**: is the 16th (84th) percentile of the per-galaxy PDF, which correspond to the 1σ confidence interval.
5. **err95_lower(upper)**: is the 2.5th (97.5th) percentile of the per-galaxy PDF, which correspond to the 2σ confidence interval.

Fig. 1 shows an example of the $p(z)$ distribution, point-estimates and summary statistics for a single object.

Yes.

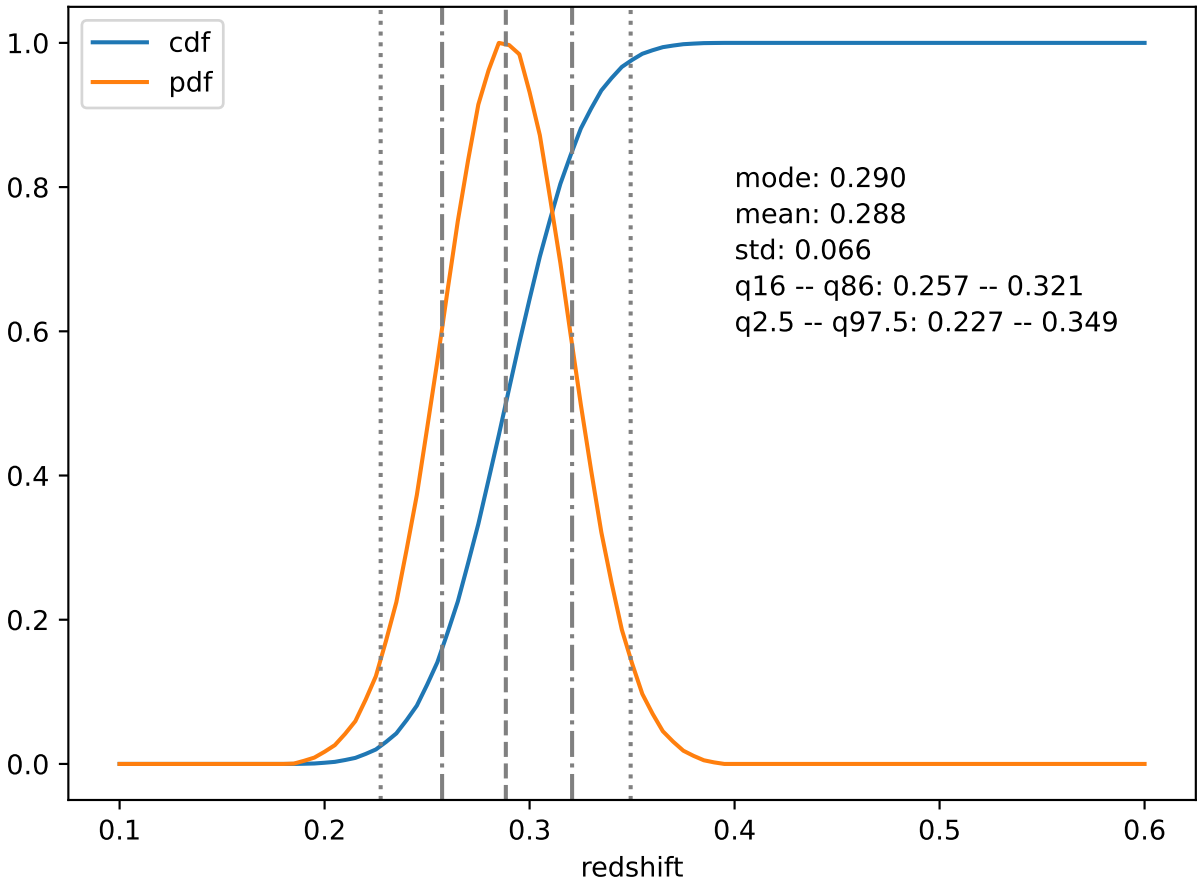


FIGURE 1: Single object $p(z)$ estimate, showing both the PDF, and the CDF, as well as the summary statistics described in the text.

A.6 Performance Monitoring Plots

Finally, we produced standardized performance monitoring plots as part of the photometric redshift workflow. These plots provide visual representations of the model's performance, allowing users to assess how well the redshift estimates align with the true redshifts from the spectroscopic training data. Common plots include:

- Input dataset characterization plots, as shown in Sec. **??**.

- Scatter plots to visualize the relationship between the predicted and true redshifts, highlighting any systematic biases or non-linearities.
- Redshift comparison plots, e.g., the bias or width of the redshift residuals, $\frac{z_{\rm phot}-z_{\rm spec}}{1+z_{\rm spec}}$, as a function of redshift or object magnitude.

To robustly summarize the distribution of residuals while minimizing the influence of outliers, we compute biweighted statistics following a two-step procedure. First, we apply an iterative 3σ clipping to the input array using `scipy.stats.sigmaclip`, repeating the clipping process `nclip` times (default is 3). This removes extreme outliers before computing robust estimators. We then calculate the *biweight location* and *biweight scale* of the clipped subset using the `astropy.stats` functions, which yield robust analogs to the mean and standard deviation, respectively.

In addition, we compute two outlier rates: the *relative outlier rate*, defined as the fraction of values in the original sample that deviate from zero by more than $3\times$ the biweight scale, and the *absolute outlier rate*, defined as the fraction of values exceeding a fixed threshold set by `self.config.abs_out_thresh`. This framework provides both robust central moments and a diagnostic of extreme deviations in the full sample.

Examples of the two latter types of plots for the BPZ and KNN algorithms are shown in Sec. **??**

B Data Distribution

To support both the Rubin community and the DESC, we will distribute these data products in several different ways:

1. Via the Rubin Data Butler (see Sec. B.1).
2. Via the Photo-z Server (see Sec. B.2).
3. Directly as files at the USDF and NERSC (see Sec. B.3).
4. Via LSDB (see Sec. B.4)

Draft

Dataset type	Description
Collection: `pretrained_models/pz/DP1/{selection}/{flavor}`	
`pzModel_{algo}`	Estimator models (see A.3)
Collection: `LSSTComCam/runs/DRP/DP1/pz/DM-51523/{selection}/{flavor}/{version}`	
`pz_estimate_{algo}`	QP ensembles (see A.4)
`pz_{algo}_config`	Configuration parameters (see A.1)
`pz_{algo}_log`	Log files
`pz_{algo}_metadata`	Processing metadata

TABLE 1: Photo-z related objects stored in the Rubin Data Butler. Note that `{selection}` describes the selection applied to the trained data set.

In each of these distribution mechanisms there are a number of metadata that are used as fields in defining specific data products. In particular:

- **algo**: specifies a particular photo-z estimation algorithm (see Tab. **??**),
- **selection**: specifies a particular data selection (see Sec. **??**),
- **flavor**: specifies a particular set of configuration parameters,
- **dataset**: specifies a particular dataset (see Sec. **??**).
- **version**: specifies a processing version (e.g., v1, v2).

B.1 Distribution via the Rubin Data Butler

Creation of photometric redshift estimates using RAIL in the Rubin DM framework and distribution via the Rubin Data Butler is supported the `meas_pz` software package for DM supported algorithms and by the `meas_pz_extensions` software package for the community-supported algorithms described in this note.

For DP1, the following objects stored in the data butler at USDF and NERSC are listed in Tab. 1.

B.2 Distribution via the Photo-z Server

The Photo-z Server (or PZ Server) is a web-based service available for the LSST community to create and host PZ-related lightweight data products. It relies on the infrastructure of the

Brazilian Independent Data Access Center and is developed and maintained by LIneA as part of the Brazilian in-kind contribution program.

The PZ server is open to any LSST member with a valid RSP account without the need for an extra local registry. Users can log into the system simply using the RSP credentials.

All data products described in this document will be hosted on the PZ Server, along with their respective metadata and documentation. There are two ways to access these data products:

B.2.1 From the PZ Server website

Data products are listed both on the 'Rubin PZ Data Products' page (for official data products released or recommended by LSST DM) and 'User-generated data products' for data products produced or uploaded by the LSST community members. The DP1 PZ data products described in this document will be available in the first one.

B.2.2 Via the `pzserver` Python library

Similar to the RSP, the PZ Server provides an API interface that enables users to access data through Python scripts from any location, provided they know the product name as registered on the server.

If it is the first time using the library, it must be installed via `pip` in the terminal or in a notebook cell: `pip install pzserver`

Then, the `PzServer` class opens the remote connection to the PZ Server database. An access token is required for authentication. The token can be generated by users on the PZ Server website (top right corner menu on the home page).

```
from pzserver import PzServer
pz_server = PzServer(token="<paste your access token here>")
```

To display the product metadata and download it to the local working directory (if not in a Jupyter notebook, replace `display` for `get`):

Object type	Relative Path
Training data sets	`data/train/*.hdf5`
Test data sets	`data/test/*.hdf5`
In : `pz/projects/dp1/pipelines`	
Pipeline configurations	`{pipeline}_{flavor}.yaml`
In : `pz/projects/dp1/data`	
Estimator models	`{selection}_{flavor}/model_inform_{algo}.pkl`
QP ensembles (for test data)	`{selection}_{flavor}/output_estimate_{algo}.hdf5`
QP ensembles (for other data)	`{selection}_{flavor}/{dataset}/output_estimate_{algo}.hdf5`

TABLE 2: Photo-z related files in the `rail_projects`-managed shared project areas.

```
pz_server.display_product_metadata(<product_id>)
pz_server.download_product(product_id, save_in=".")
```

Alternatively, it is possible to load a table directly into memory as a Pandas DataFrame or Astropy Table. For instance, to load a training set:

```
training_set = pz_server.get_product(<training_set_id>)
training_set.display_metadata()
```

A tutorial notebook with examples for all `pzserver` methods is available on the `pzserver` library's repository on GitHub.

B.3 Distribution as files at USDF and NERSC

All of the test and training files, the outputs from runs used to optimize the model hyperparamters, as well as from the optimized models, the photo-z estimates for the entire DP1 dataset, are all available in `rail_projects`-managed shared project area at both NERSC and USDF. These data products are listed in Tab. 2.

B.4 Distribution via LSDB

The Large Survey DataBase (LSDB) will host the DP1 data on the USDF and the Canadian IDAC RSP. The DP1 data will be turned into Hierarchical Adaptive Tiling Scheme (HATS) format. LSDB

enables researchers to load large datasets with limited memory, and fast cross match to other surveys like DESI DR1 and GAIA DR3.

The photo-z point estimates for DP1 in the **ECDFS**,**EDFS**, **Rubin_SV_95_-25**, and **Rubin_SV_38_7** will be served as a single tabular catalog in the LSDB. The location on the USDF for this catalog is `/sdf/data/rubin/shared/lsdb_commissioning/dp1_pz_hats`.

Draft

C References

D Acronyms

Acronym	Description
1D	One-dimensional
3D	Three-dimensional
AGN	Active Galactic Nuclei
API	Application Programming Interface
CDF	Cumulative Distribution Function
COSMOS	Cosmic Evolution Survey
DESC	Dark Energy Science Collaboration
DESI	Dark Energy Spectroscopic Instrument
DM	Data Management
DMTN	DM Technical Note
DP1	Data Preview 1
DP2	Data Preview 2
DR1	Data Release 1
DR3	Data Release 3
DRP	Data Release Processing
ECDFS	Extended Chandra Deep Field-South Survey
EDFS	Euclid Deep Field South
ELG	Emission-Line Galaxies
GOODS	The Great Observatories Origins Deep Survey
HST	Hubble Space Telescope
IDAC	Independent Data Access Center
JWST	James Webb Space Telescope (formerly known as NGST)
LRG	Luminous Red Galaxies
LSS	Large Scale Structure
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LSSTCam	LSST Science Camera

Draft

LSSTComCam	Rubin Commissioning Camera
MB	MegaByte
NED	NASA/IPAC Extragalactic Database
NERSC	National Energy Research Scientific Computing Center
PDF	Portable Document Format
PZ	photo-z
RMS	Root-Mean-Square
RSP	Rubin Science Platform
RTN	Rubin Technical Note
SE	System Engineering
SED	Spectral Energy Distribution
SNR	Signal to Noise Ratio
SV	Science Validation
USDF	United States Data Facility
YAML	Yet Another Markup Language
photo-z	photometric redshift

Draft