Anacal on Clusters

20241028

Link to this deck

1. Join #desc-cl-shear-icl the link is pinned
2. Or scan the QRcode

Environment set up

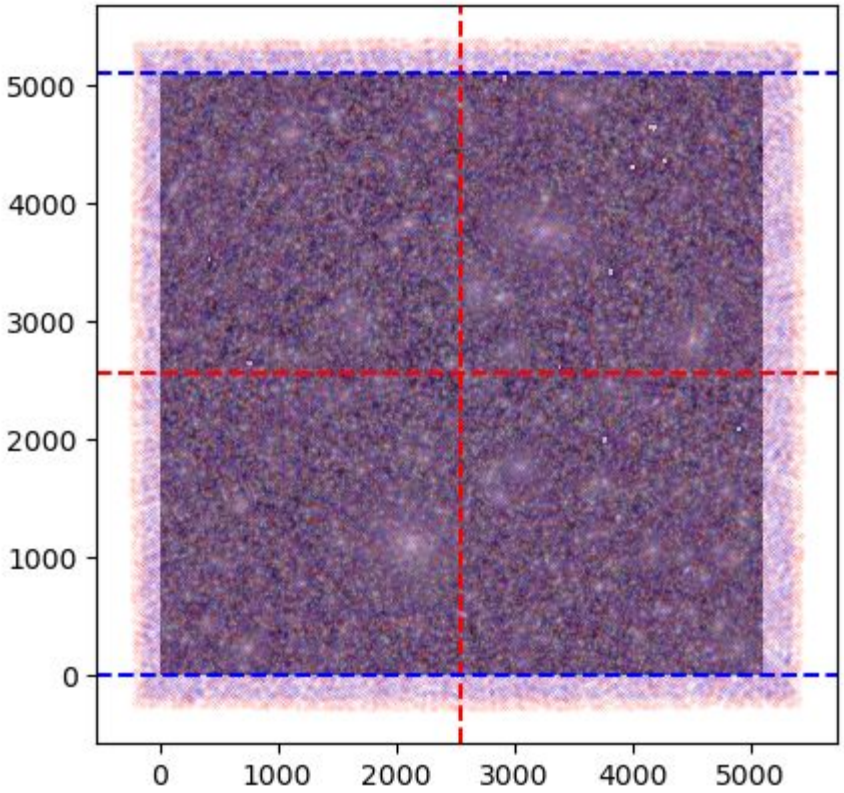
1. Currently the cluster branch of anacal resides in a fork in xlens that will be merged after merging some changes in dependencies
2. Currently everything is intended to work with sim images
3. Major dependencies
    a. stackvana for DM science pipeline
    b. lenstronomy for making sims
    c. lsstdesc-wl-shear-sim for making sims

installation and preparation

1. New environment
   a. conda create -n xlens
   b. conda install conda-forge::stackvana
   c. pip install git+https://github.com/zhouconghao/xlens.git@cluster_summary
   d. pip install git+https://github.com/zhouconghao/descwl-shear-sims.git@return_original_shift
   e. pip install lenstronomy
2. If you're on NERSC
   a. try the following command
   b. module load python
   c. conda activate /global/cfs/cdirs/desi/users/zhouc/conda_envs/xlens
   d. conda activate /pscratch/sd/z/zchusre/xlens/
3. git clone https://github.com/zhouconghao/xlens/tree/cluster_summary

Anacal in clusters

1. Run cluster sims
2. Run anacal in cluster fields
3. To extract truth shear information, we match anacal detection and input galaxy in image coordinates
4. We bin galaxies in prelensed coordinates, currently binning is fixed, but it will become configurable
5. We calculate anacal response and rotated response and ellipticity into the tangential frame of the source
6. We estimate tangential shear with rotated response and ellipticity


5000
4000
3000
2000
1000
0
0
1000
2000
3000
4000
5000

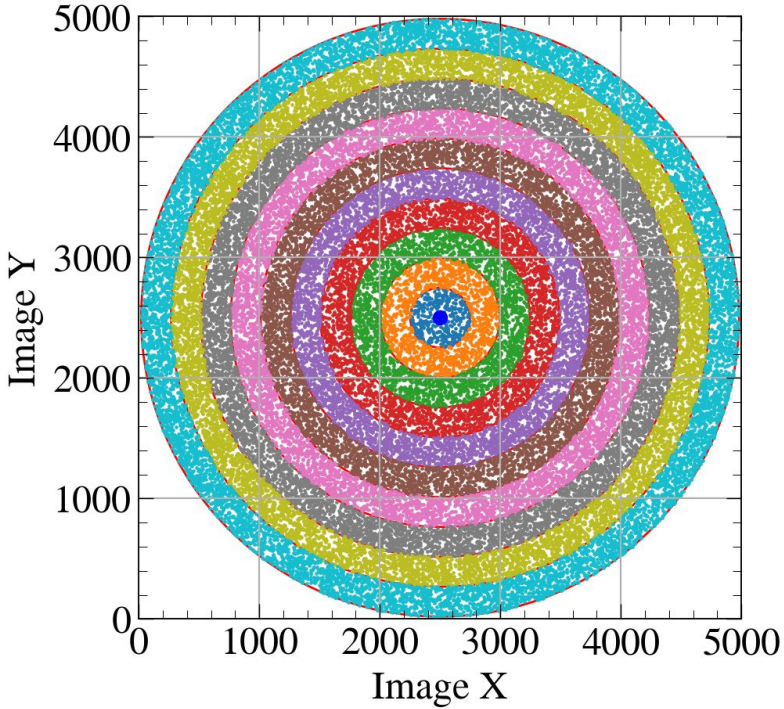
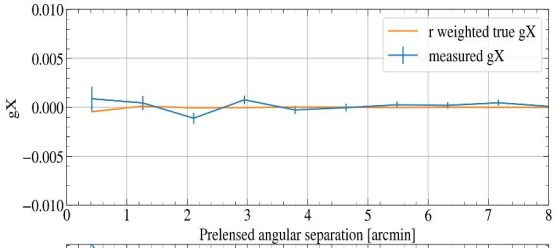
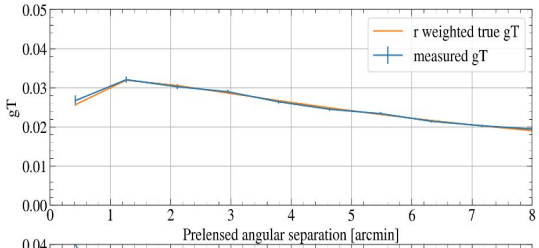
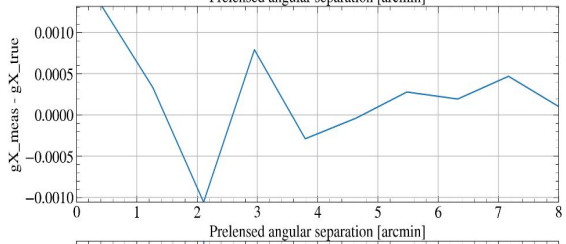


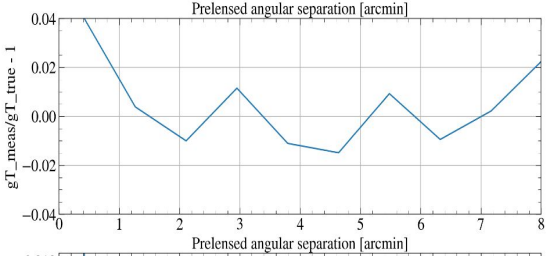
Image Y
Image X
0
1000
2000
3000
4000
5000

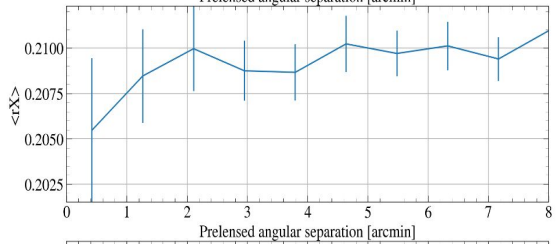
sults

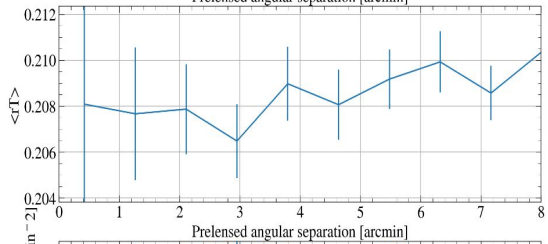

r weighted true gX
measured gX
gX
Prelensed angular separation [arcmin]

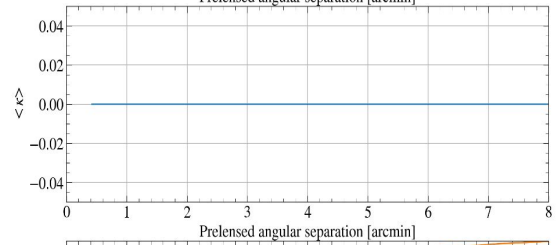

r weighted true gT
measured gT
gT
Prelensed angular separation [arcmin]

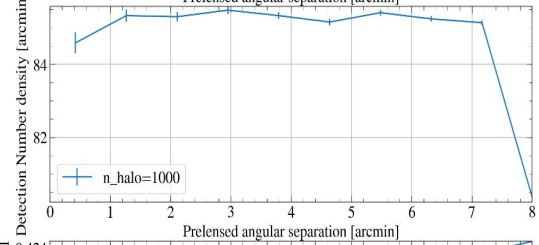

gX_meas - gX_true
Prelensed angular separation [arcmin]

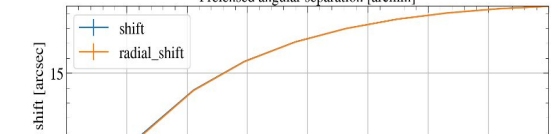

gT_meas/gT_true - 1
Prelensed angular separation [arcmin]


<rX>
Prelensed angular separation [arcmin]


<rT>
Prelensed angular separation [arcmin]


<κ>
Prelensed angular separation [arcmin]


Detection Number density [arcmin⁻²]
n_halo=1000
Prelensed angular separation [arcmin]


shift
radial_shift
shift [arcsec]


atch distance pixel

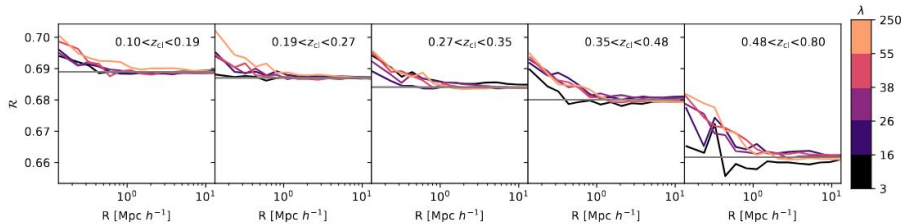
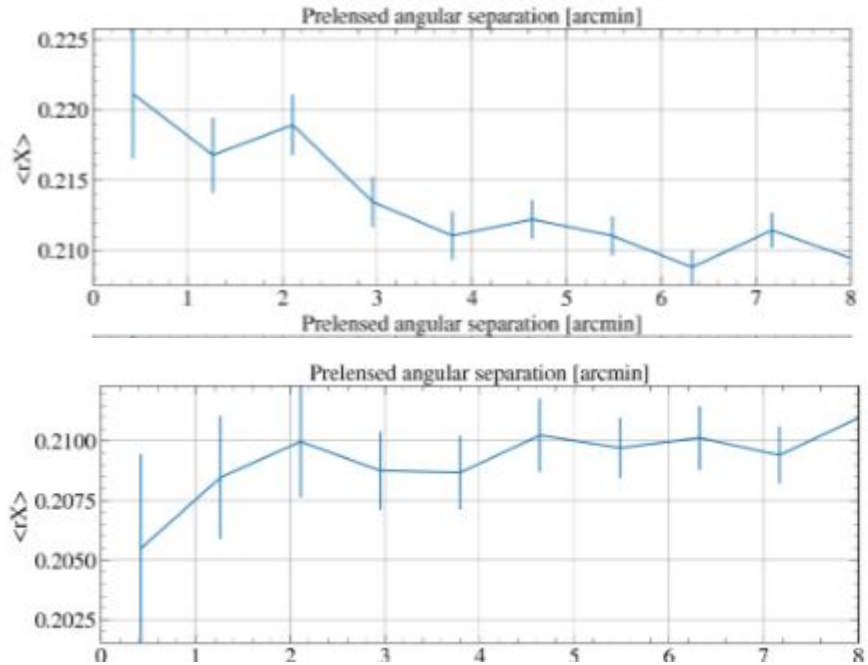
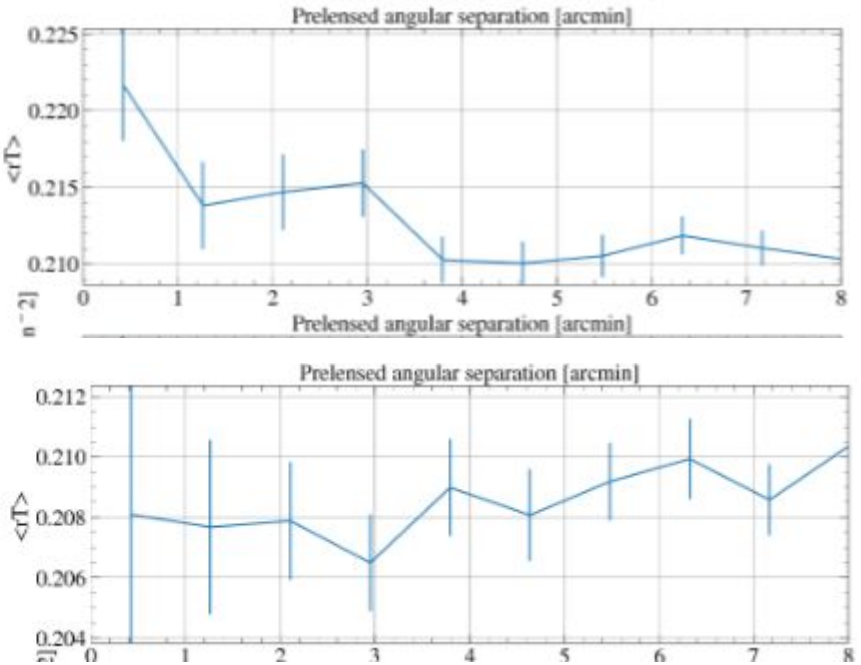


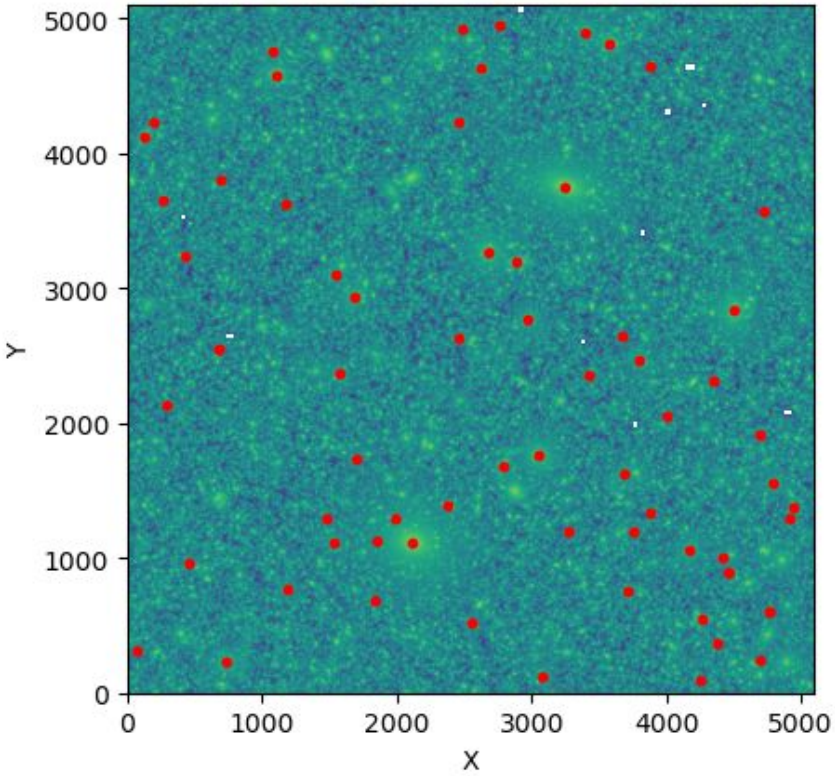
Fig. 10. Average shear response, as a function of cluster-centric distance, stacked in bins of cluster richness (color coded), and cluster redshift panels: lowest redshift left, higher redshift right. Noticeably, the shear response increases toward the cluster center. Interestingly, this signal seems to affect all lens redshift bins equally, independently of their cluster member contamination levels. We account for this effect in the cosmology-ready data products.





What we can do now

1. Specify lensz and sourcez, mass, concentration and generate simulated images
2. Run anacal on simulated images
3. Extract shear related statistics from the anacal measurement



Run the example

1. [your cloned path]/xlens/examples/anacal_dm/lsst_test
2. source butler_setup.sh
3. bps submit bps_slurm.yaml
4. plot the result with plot_summary.ipynb

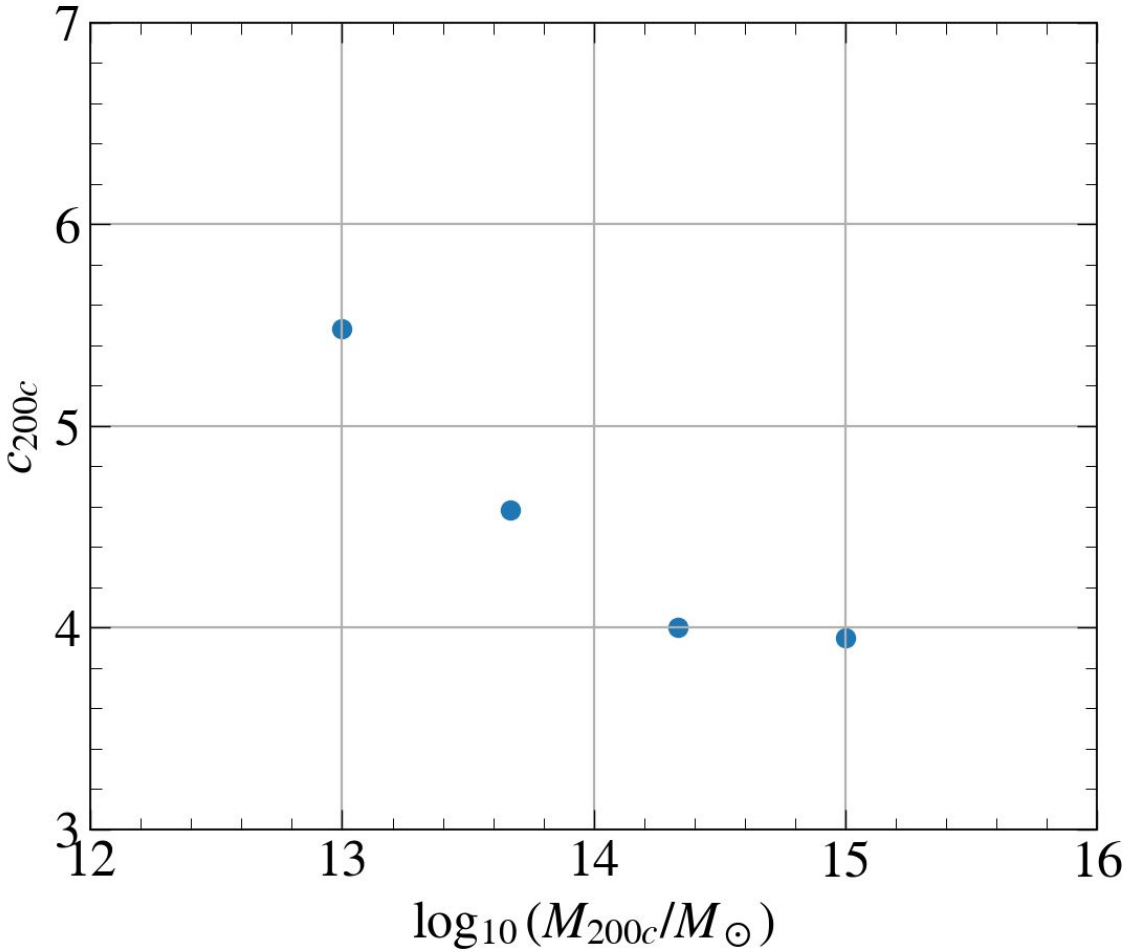
To run anacal in a cluster field, we need

1. A halo config that specify what kind of halo you want to simulate, parameters for anacal and summary statistics
2. A bps script that specify where you want to run the task
3. A script to extract the result

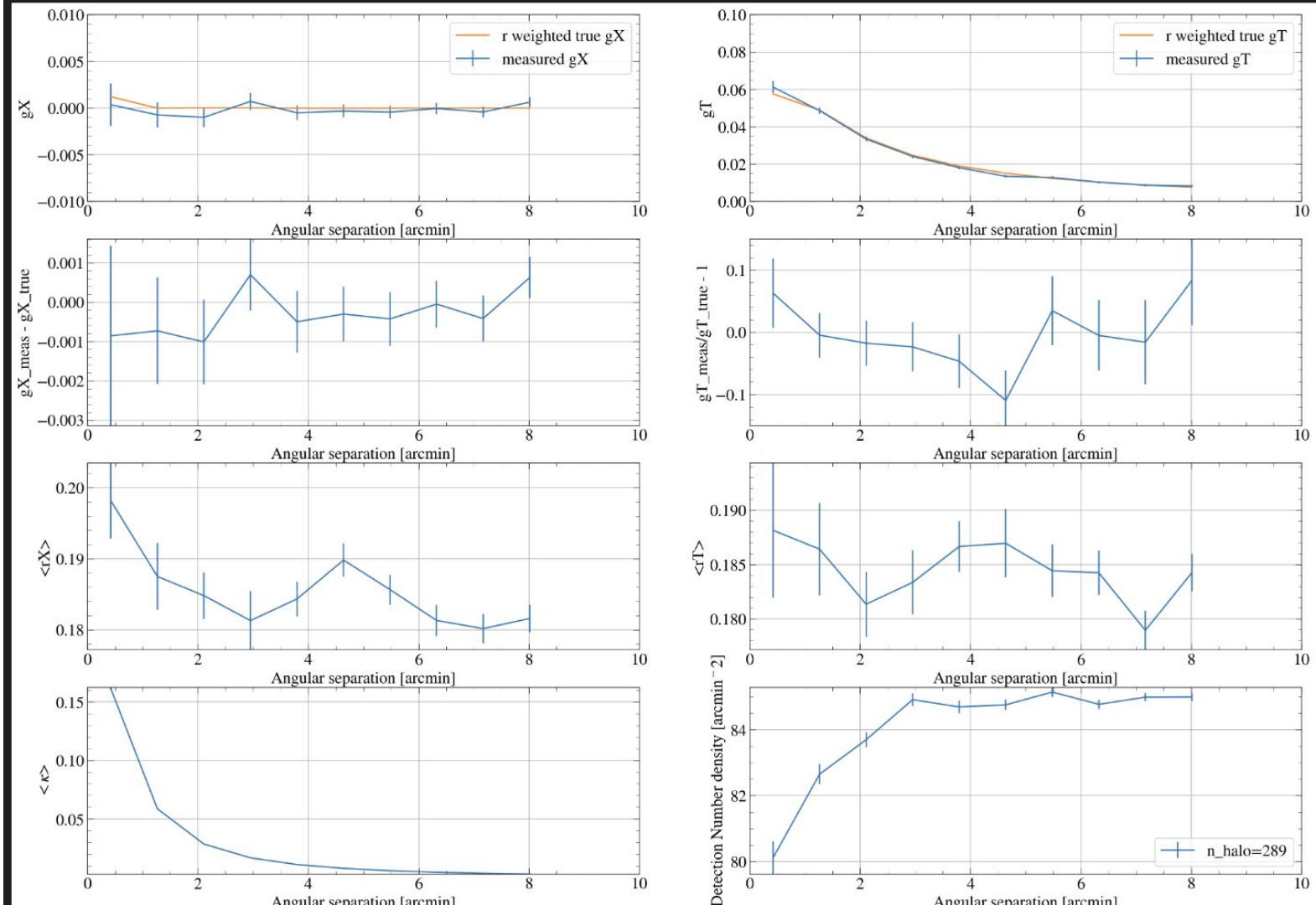
```
def get_summary_struct(n_halos, n_bins):
    dt = [
        ("angular_bin_left", f"({n_bins},)f8"),
        ("angular_bin_right", f"({n_bins},)f8"),
        ("ngal_in_bin", f"({n_bins},)i4"),
        ("eT", f"({n_bins},)f8"),
        ("eT_std", f"({n_bins},)f8"),
        ("eX", f"({n_bins},)f8"),
        ("eX_std", f"({n_bins},)f8"),
        ("rT", f"({n_bins},)f8"),
        ("rT_std", f"({n_bins},)f8"),
        ("rX", f"({n_bins},)f8"),
        ("rX_std", f"({n_bins},)f8"),
        ("gT_true", f"({n_bins},)f8"),
        ("gX_true", f"({n_bins},)f8"),
        ("kappa_true", f"({n_bins},)f8"),
        ("lensed_shift", f"({n_bins},)f8"),
        ("radial_lensed_shift", f"({n_bins},)f8"),
        ("r_weighted_gT", f"({n_bins},)f8"),
        ("r_weighted_gX", f"({n_bins},)f8"),
        ("median_match_dist", f"({n_bins},)f8"),
        ("match_failure_rate", f"({n_bins},)f8"),
    ]
    return np.zeros(n_halos, dtype=dt)
```

We sample 4 mass in mass range between 1e13 to 1e15, and calculate concentration with m-c relation.

For each halo we have ~300 realization of shear estimation based on NFW haloes. In sims there are no noise and no blending.



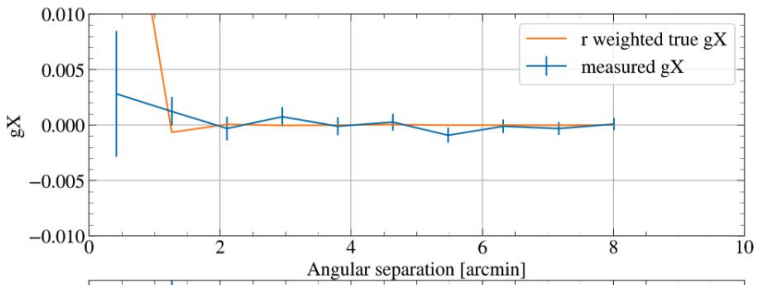
2e14

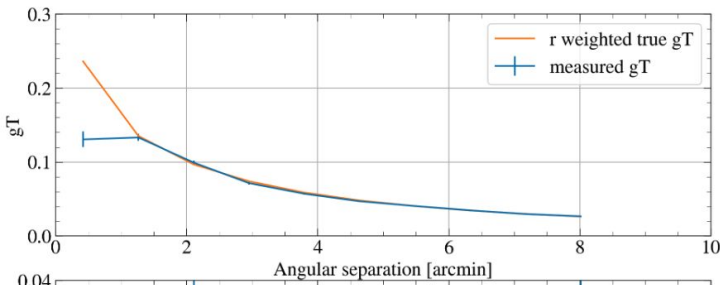

r weighted true gX
measured gX
gX
Angular separation [arcmin]
r weighted true gT
measured gT
gT
Angular separation [arcmin]
gX_meas - gX_true
Angular separation [arcmin]
gT_meas/gT_true - 1
Angular separation [arcmin]
<rX>
Angular separation [arcmin]
<rT>
Angular separation [arcmin]
<κ>
Angular separation [arcmin]
Detection Number density [arcmin⁻2]
n_halo=289
Angular separation [arcmin]

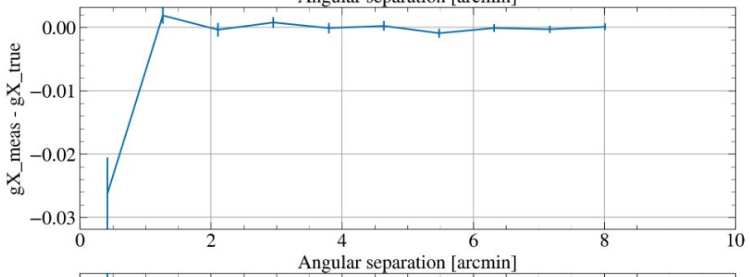
mass 1e15

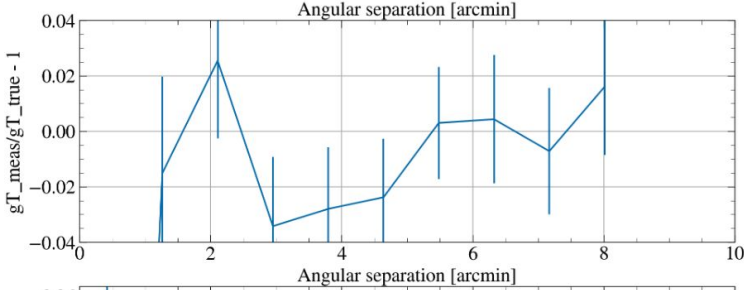
1 arcmin = 60 arcsec = 300 pixels = 300 kpc at z = 0.33

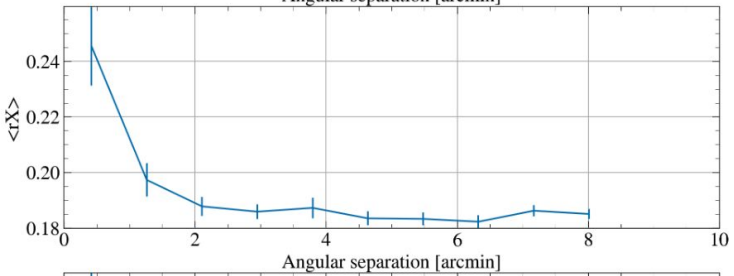
the inner most point is likely from a bug.

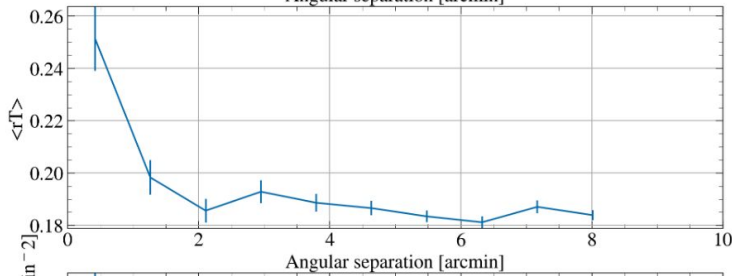


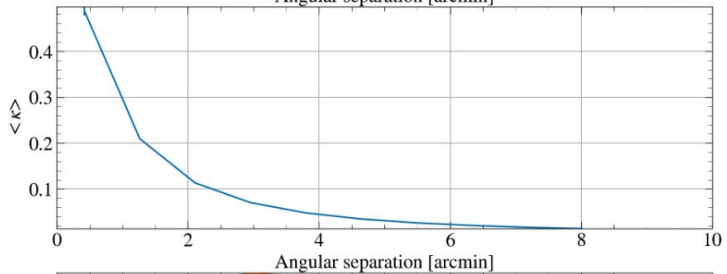


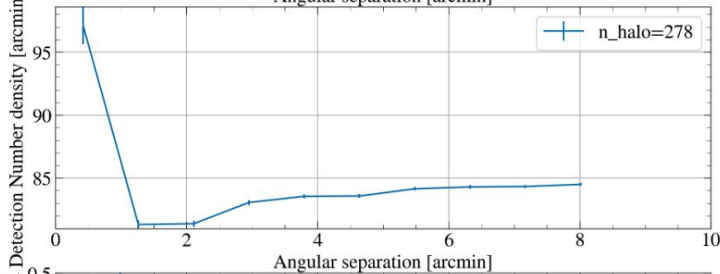


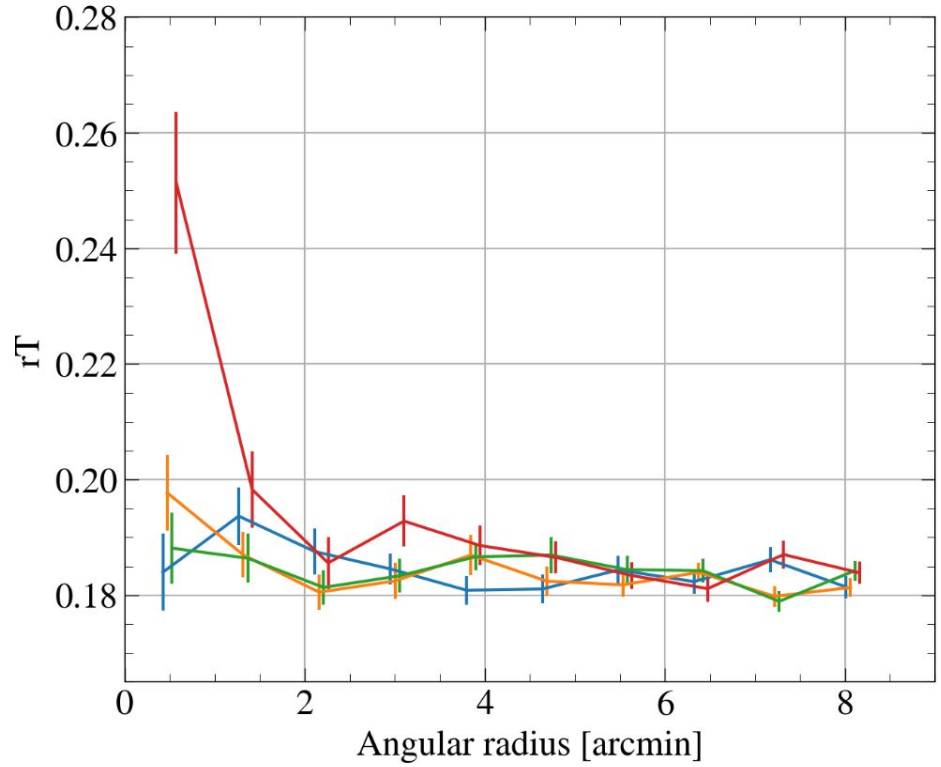


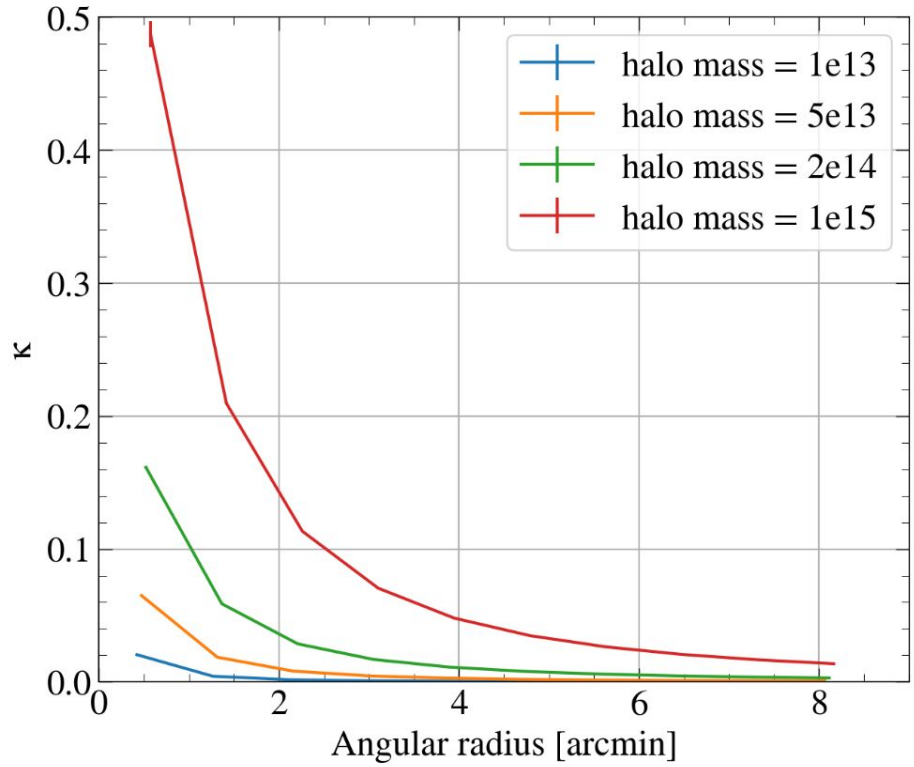


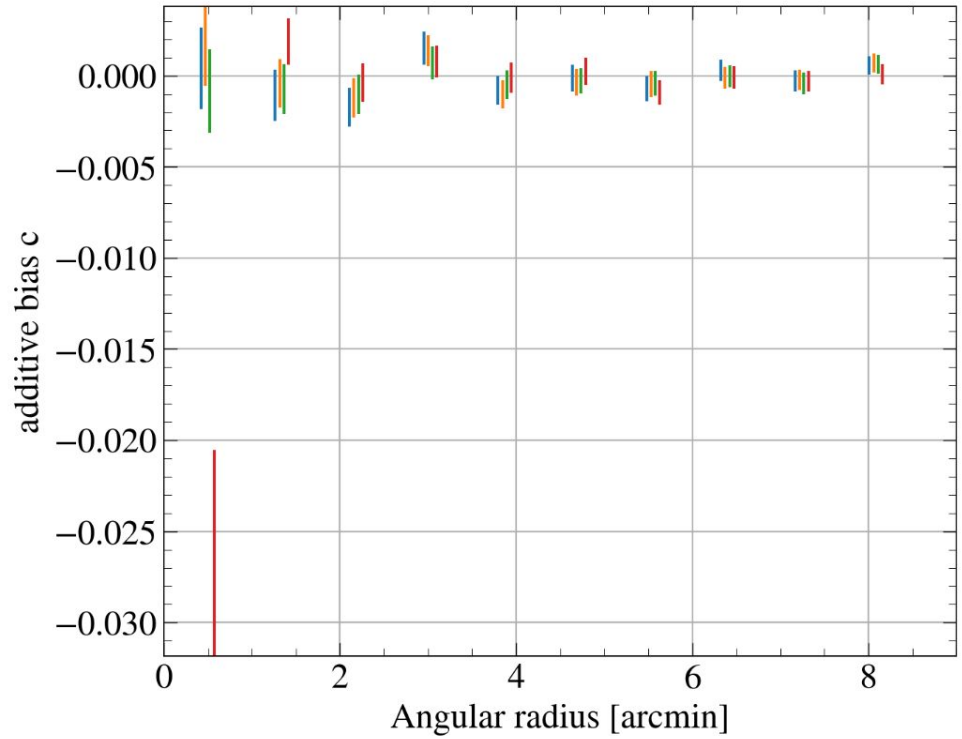


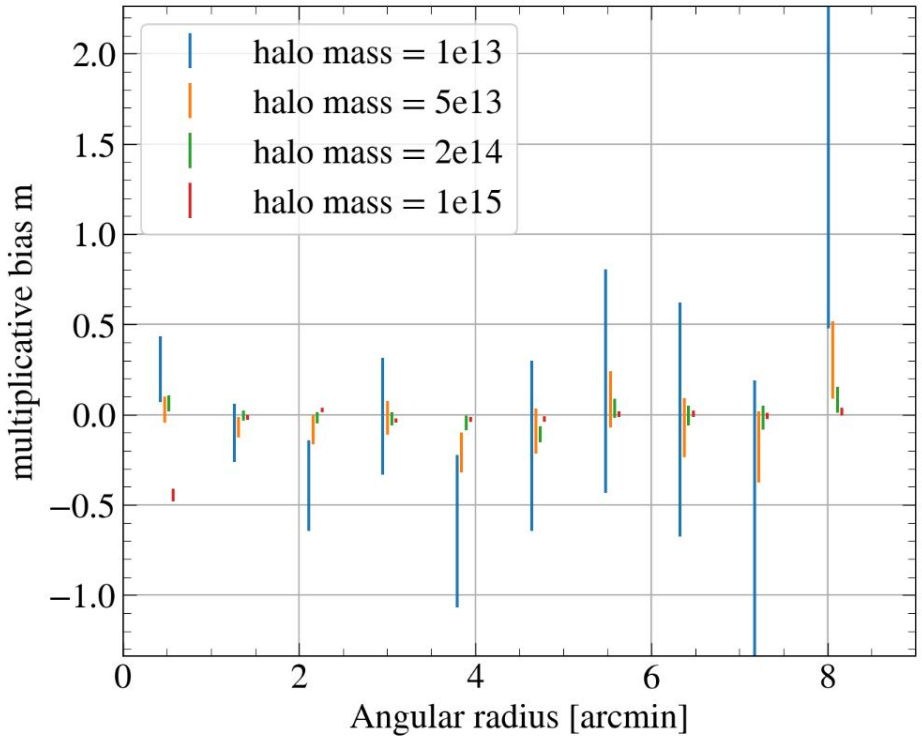





0.28
0.26
0.24
0.22
0.20
0.18
rT
0
2
4
6
8
Angular radius [arcmin]


0.5
0.4
0.3
0.2
0.1
0.0
κ
0
2
4
6
8
Angular radius [arcmin]
halo mass = 1e13
halo mass = 5e13
halo mass = 2e14
halo mass = 1e15


additive bias c
0.000
−0.005
−0.010
−0.015
−0.020
−0.025
−0.030
0
2
4
6
8
Angular radius [arcmin]


multiplicative bias m
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
halo mass = 1e13
halo mass = 5e13
halo mass = 2e14
halo mass = 1e15
0
2
4
6
8
Angular radius [arcmin]