

In [1]:

```
import pandas as pd
import numpy as np
np.random.seed(1)
from matplotlib import pyplot as plt
from tqdm import tqdm

import warnings
warnings.filterwarnings("ignore")

from multiprocessing import cpu_count, Pool
```

executed in 1.89s, finished 17:06:02 2018-11-05

In [9]:

```
tr = pd.read_pickle('../data/train.pkl')
tr_log = pd.read_pickle('../data/train_log.pkl')

y_true = pd.read_pickle('../data/y_true.pkl').values
y_pred = pd.read_pickle('../data/oof.pkl')
```

executed in 1.33s, finished 17:24:50 2018-11-05

In [10]:

```
idx_gal = tr[tr['hostgal_photoz'] == 0].index
idx_exgal = tr[tr['hostgal_photoz'] != 0].index

y_pred.iloc[idx_gal, [1, 3, 4, 6, 7, 9, 10, 11, 13]] = 0
y_pred.iloc[idx_exgal, [0, 2, 5, 8, 12]] = 0

y_pred = y_pred.values.astype(float)
y_pred /= y_pred.sum(1)[:,None]

tmp = y_true * y_pred

tr['loss'] = 1 - tmp.max(1)
```

executed in 26ms, finished 17:24:50 2018-11-05

In [11]:

```
labelname = [f'class_{i}' for i in [6, 15, 16, 42, 52, 53, 62, 64, 65, 67, 88, 90, 92, 95]]
tmp = pd.DataFrame(y_pred, columns=labelname)
tr = pd.concat([tr, tmp], axis=1)
```

executed in 39ms, finished 17:24:50 2018-11-05

In [34]:

```
tr['pred'] = tr[labelname].idxmax(1).map(lambda x: x.split('_')[1]).astype(int)
```

executed in 26ms, finished 17:40:34 2018-11-05

In [44]:

```
tr['tmp'] = tr['target'].astype(str) + ' -> ' + tr['pred'].astype(str)
```

executed in 67ms, finished 17:54:36 2018-11-05

...

In [41]:

```
tr.sort_values('loss', ascending=False, inplace=True)
tr.reset_index(drop=True, inplace=True)
tr
```

executed in 201ms, finished 17:47:32 2018-11-05

...

In [46]:

```
def plt_obj(oid=None, save=False, path=None, norm=False, return_df=False):
    if oid is None:
        oid = np.random.choice(tr.object_id)
        df = tr_log[tr_log.object_id==oid]
    if norm:
        df.flux /= df.flux.max()
    target = tr.loc[tr.object_id==oid, 'tmp'].values[0]
    loss = tr.loc[tr.object_id==oid, 'loss'].values[0]

    df['date'] = df.mjd.astype(int)

    df = pd.pivot_table(df, index=['date'], columns=['passband'], values=['flux'])
    if return_df:
        return df

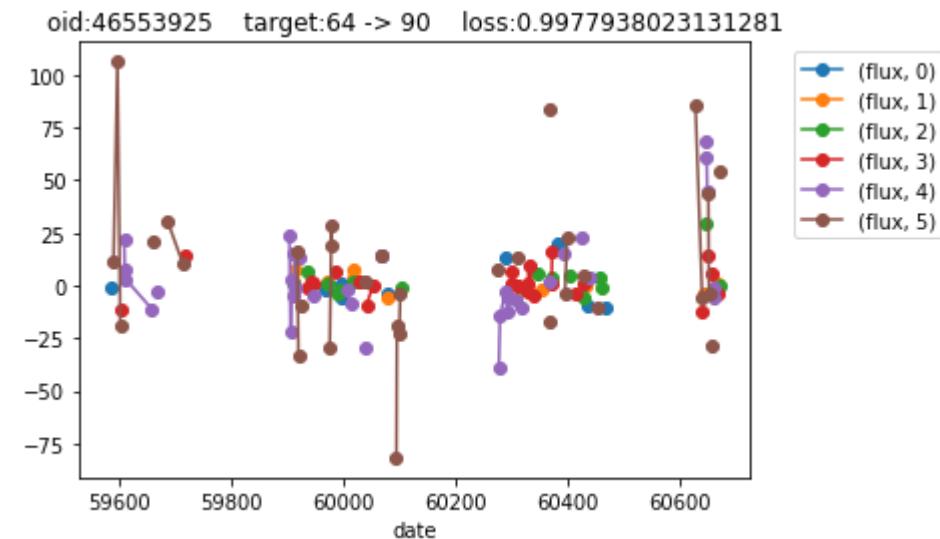
    df.plot(marker="o", legend=True)
    plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
    plt.title(f'oid:{oid} target:{target} loss:{loss}')
    if save==True and path is not None:
        plt.savefig(path)
return
```

executed in 99ms, finished 17:55:10 2018-11-05

In [47]:

plt\_obj(tr.object\_id[0])

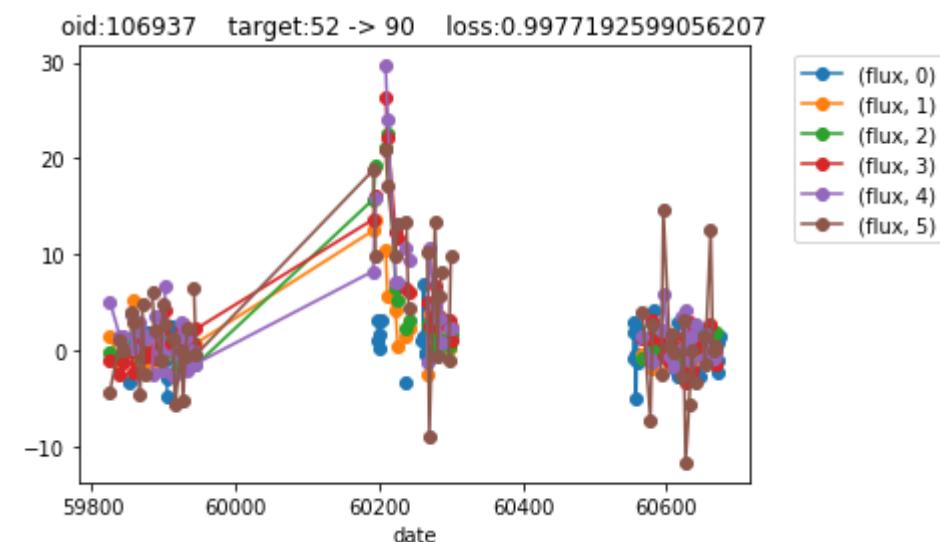
executed in 881ms, finished 17:55:12 2018-11-05



In [48]:

plt\_obj(tr.object\_id[1])

executed in 743ms, finished 17:55:20 2018-11-05



In [51]:

```

classes = [6, 15, 16, 42, 52, 53, 62, 64, 65, 67, 88, 90, 92, 95]
M, N = 5, 5
for c in tqdm(classes):
    li = tr[tr.target==c].head(M*N).object_id.tolist()
    fig, axes = plt.subplots(ncols=M, nrows=N, figsize=(28, 25), sharex=True)

    # プロット
    for i,(ax, oid) in enumerate(zip(axes.ravel(), li)):
        df=plt_obj(oid, return_df=True)

        target = tr.loc[tr.object_id==oid, 'tmp'].values[0]
        loss = tr.loc[tr.object_id==oid, 'loss'].values[0]
        loss = round(loss, 3)

        if i==24:
            legend=True
        else:
            legend = False
        df.plot(ax=ax, marker="o", legend=legend, title=f'oid:{oid} target:{target} loss:{loss}')

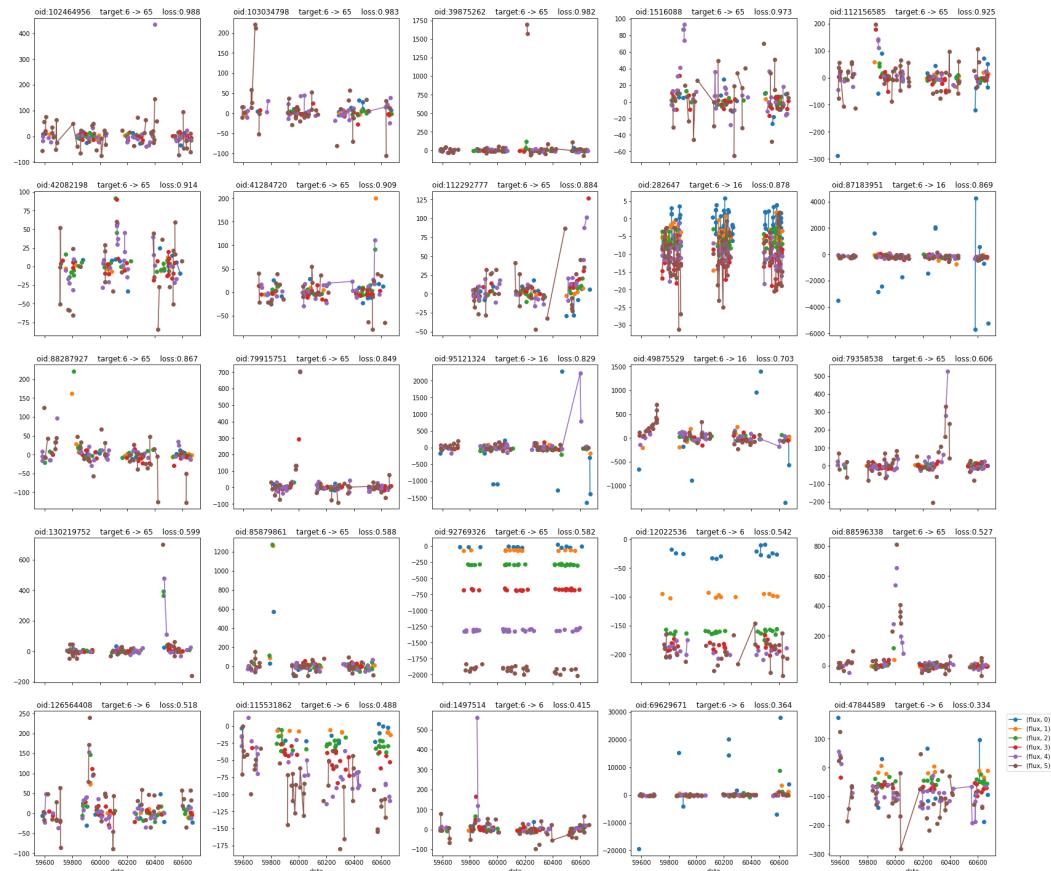
    fig.suptitle(f'target:{c}', fontsize=36)
    plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

```

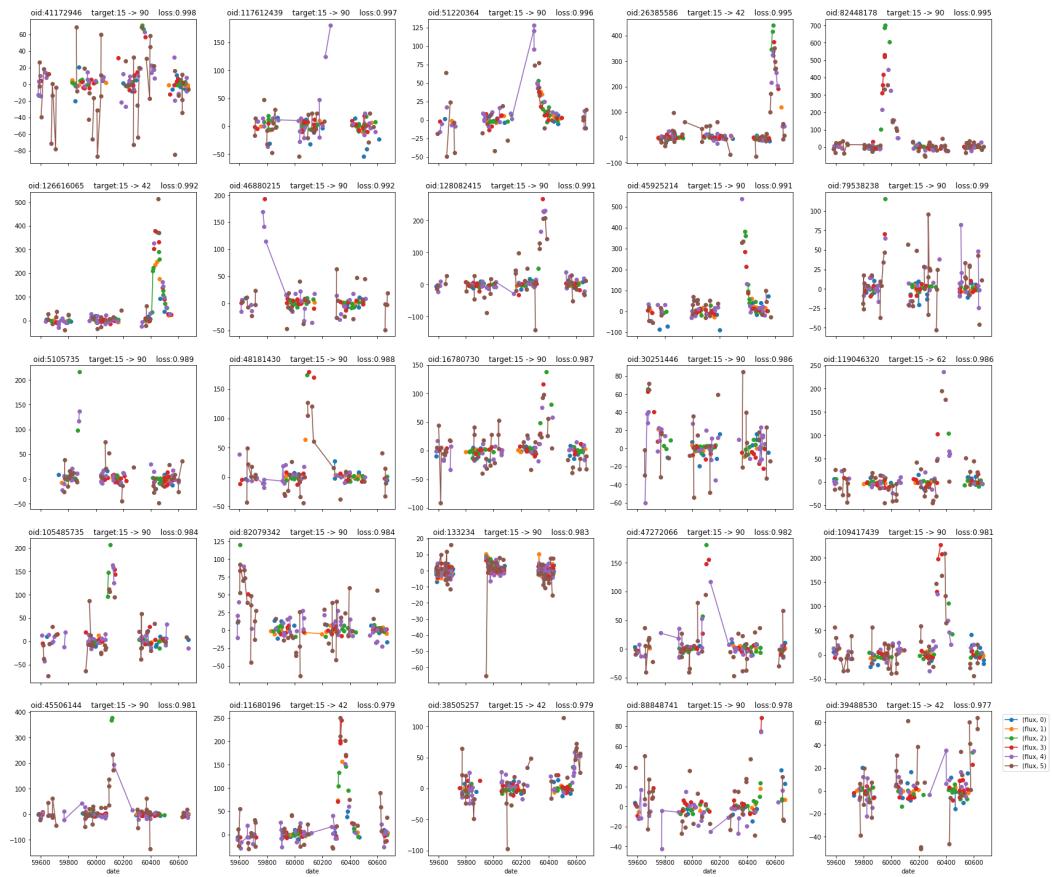
executed in 3m 26s, finished 18:09:14 2018-11-05

100% [██████████] 14/14 [03:01&lt;00:00, 12.95s/it]

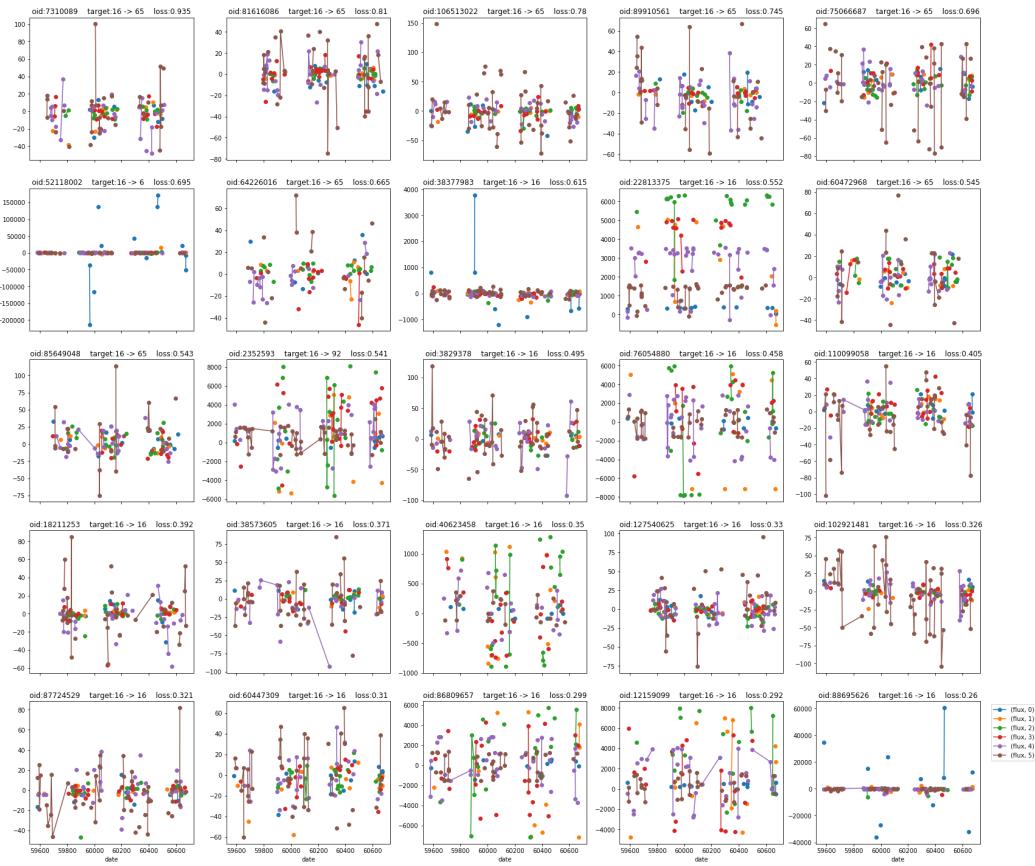
target:6



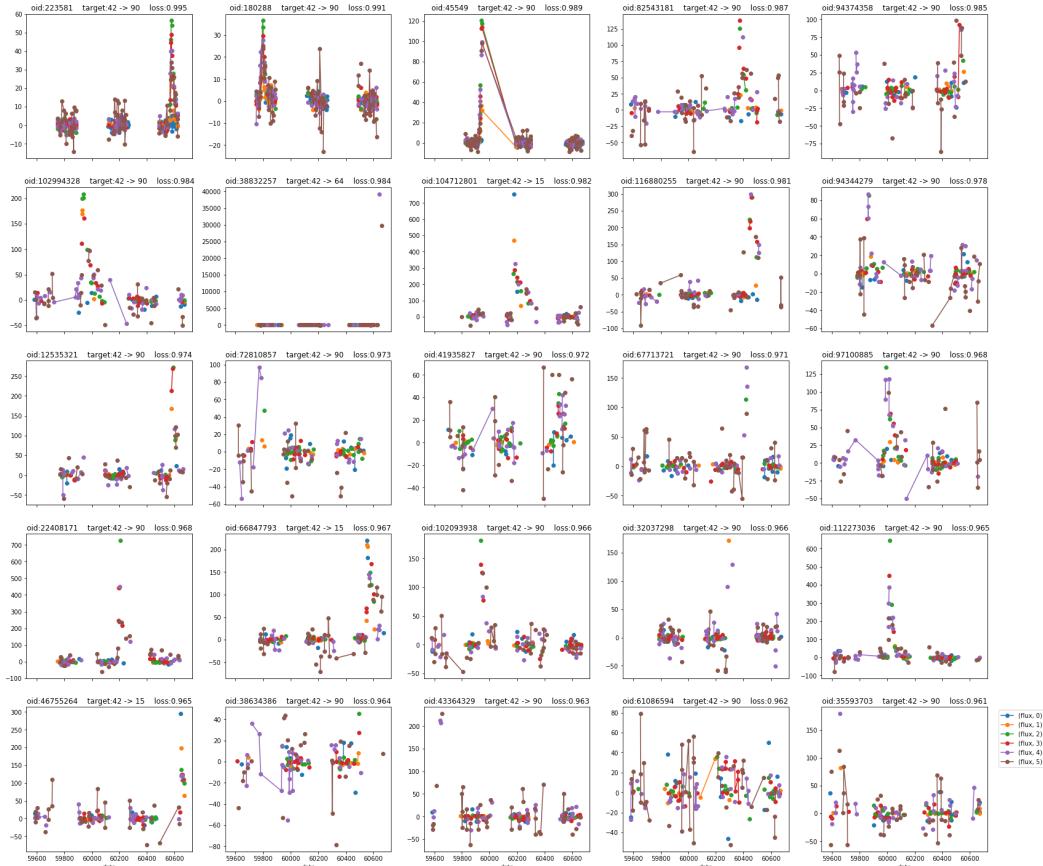
target:15



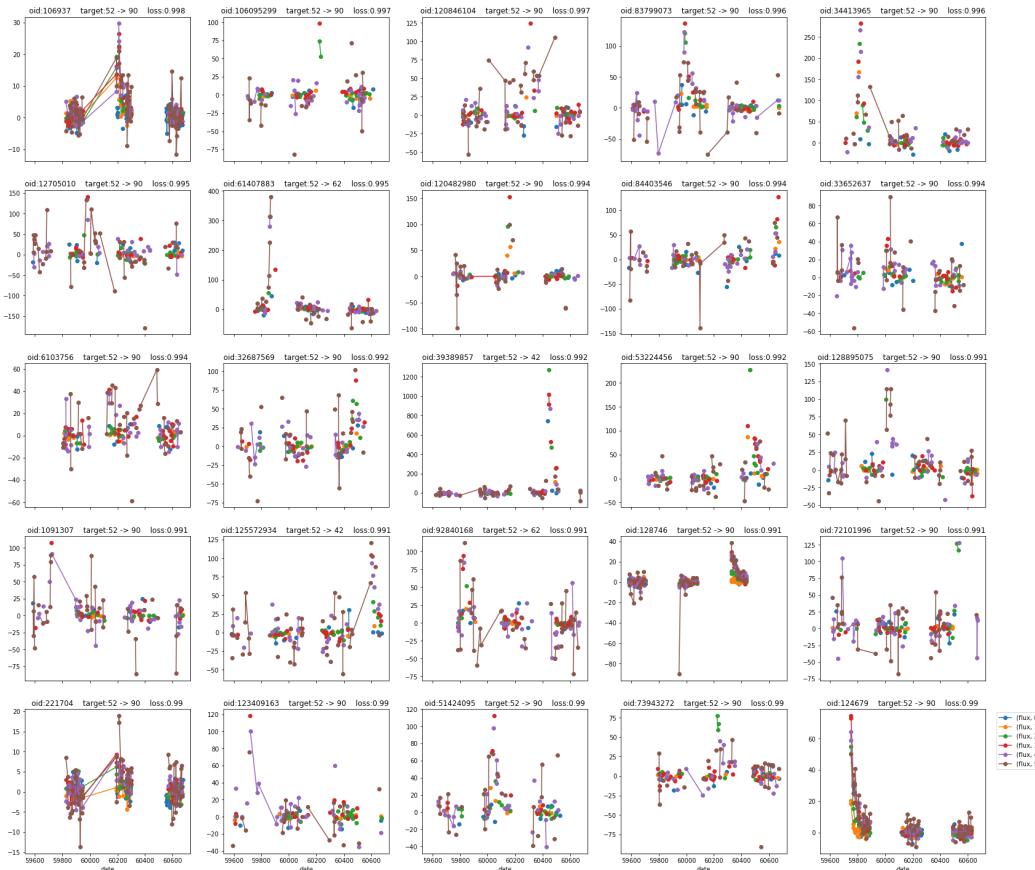
target:16



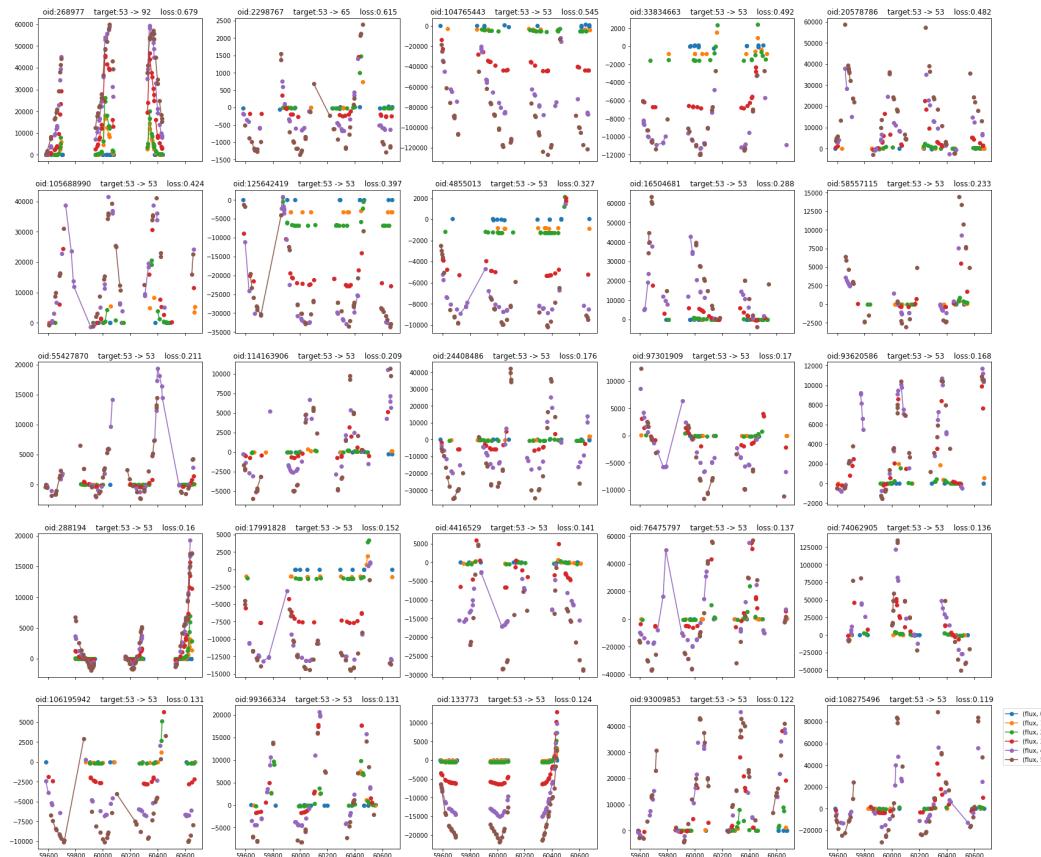
target:42



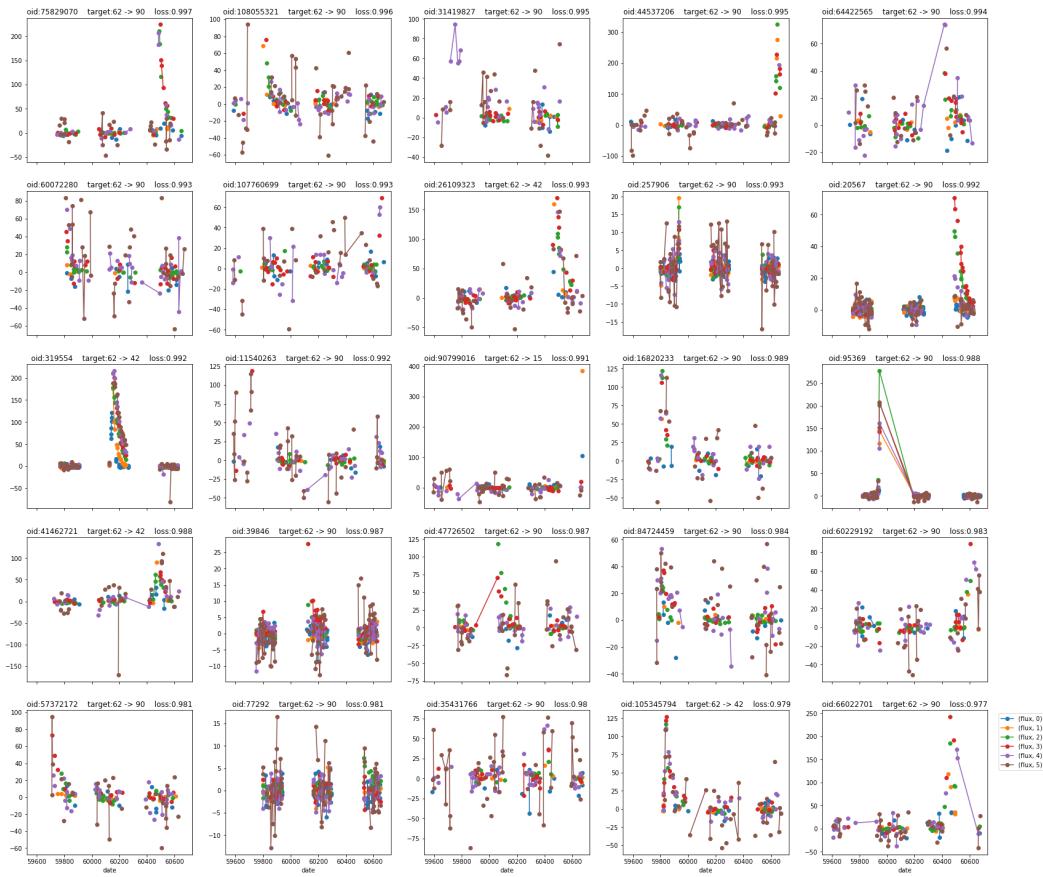
target:52



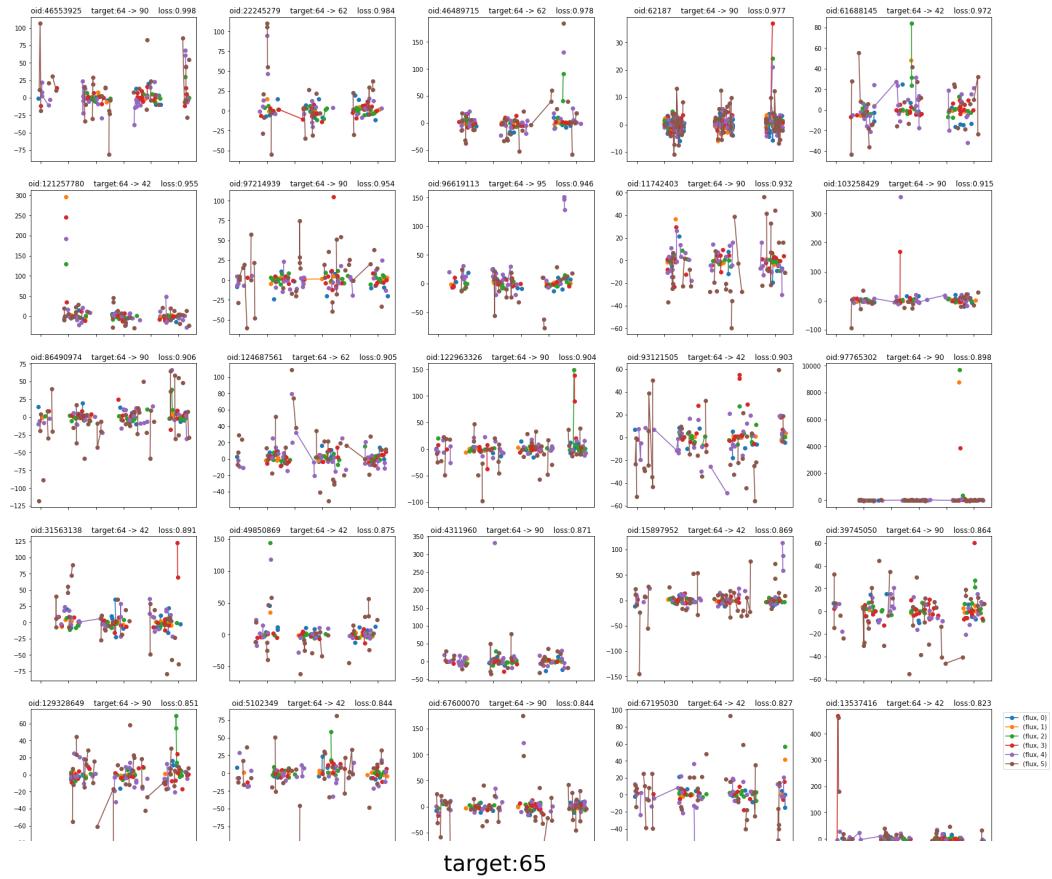
target:53



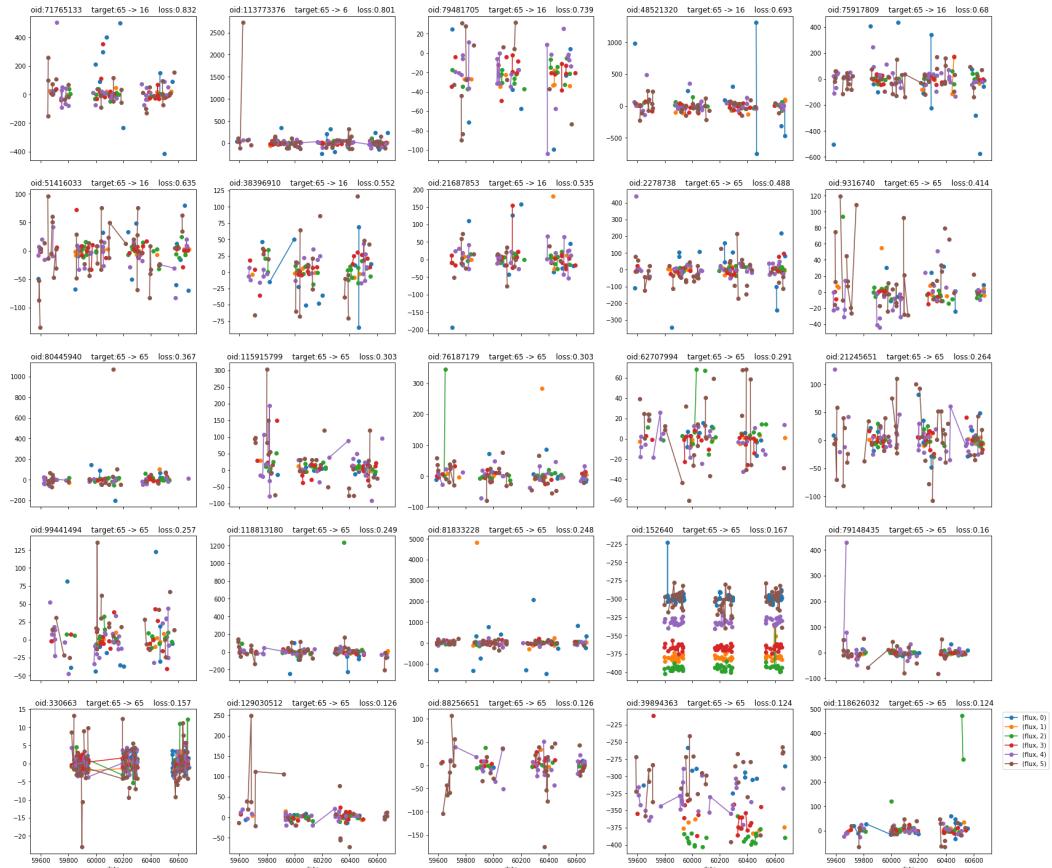
target:62



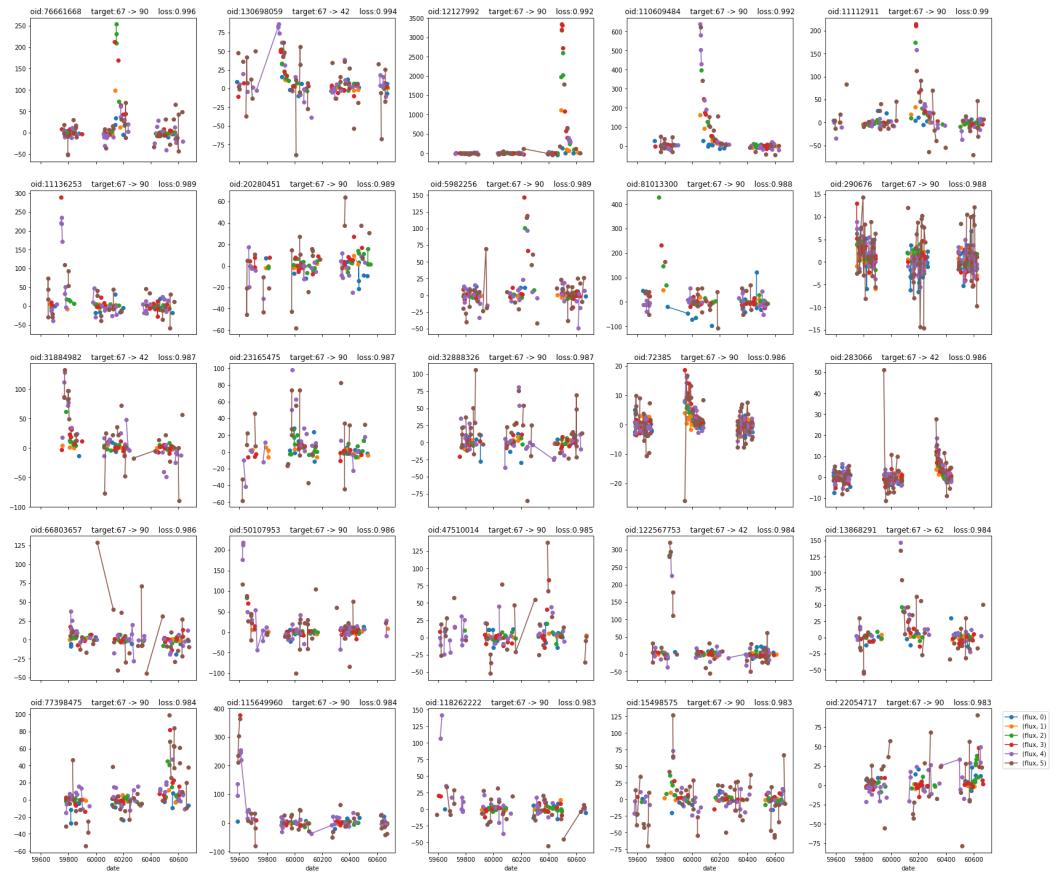
target:64



target:65

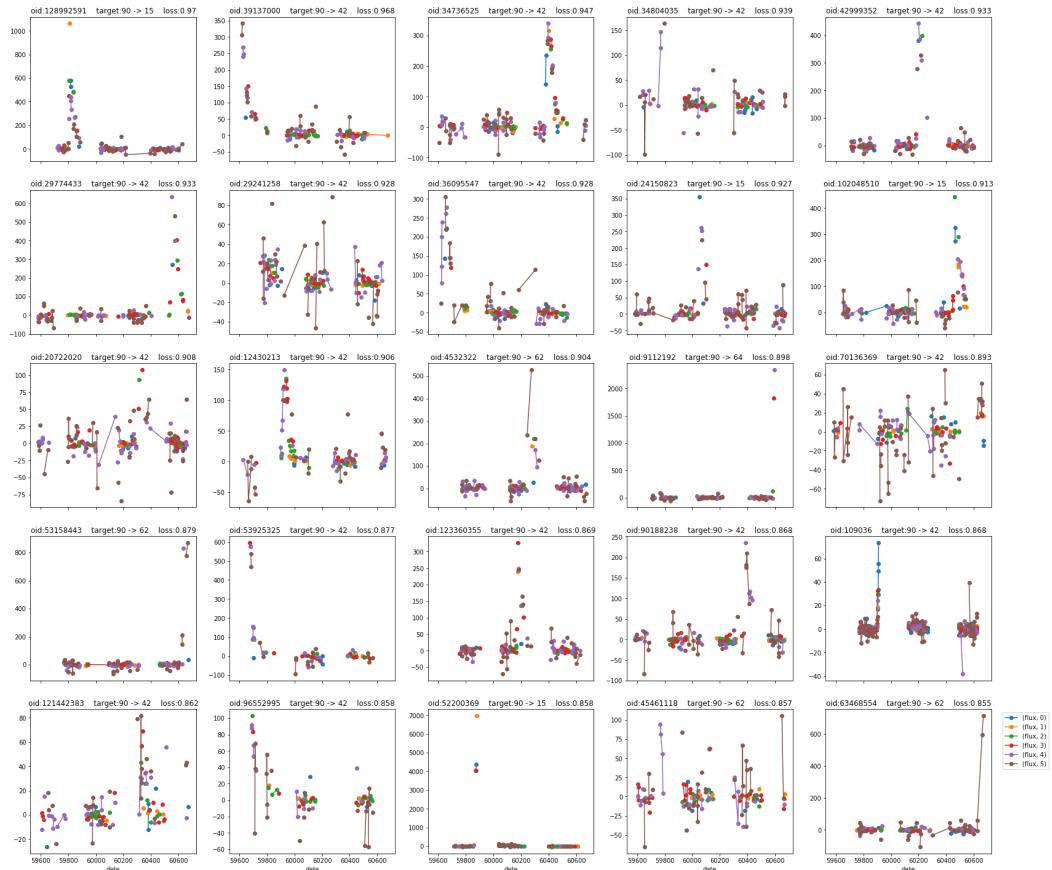


target:67

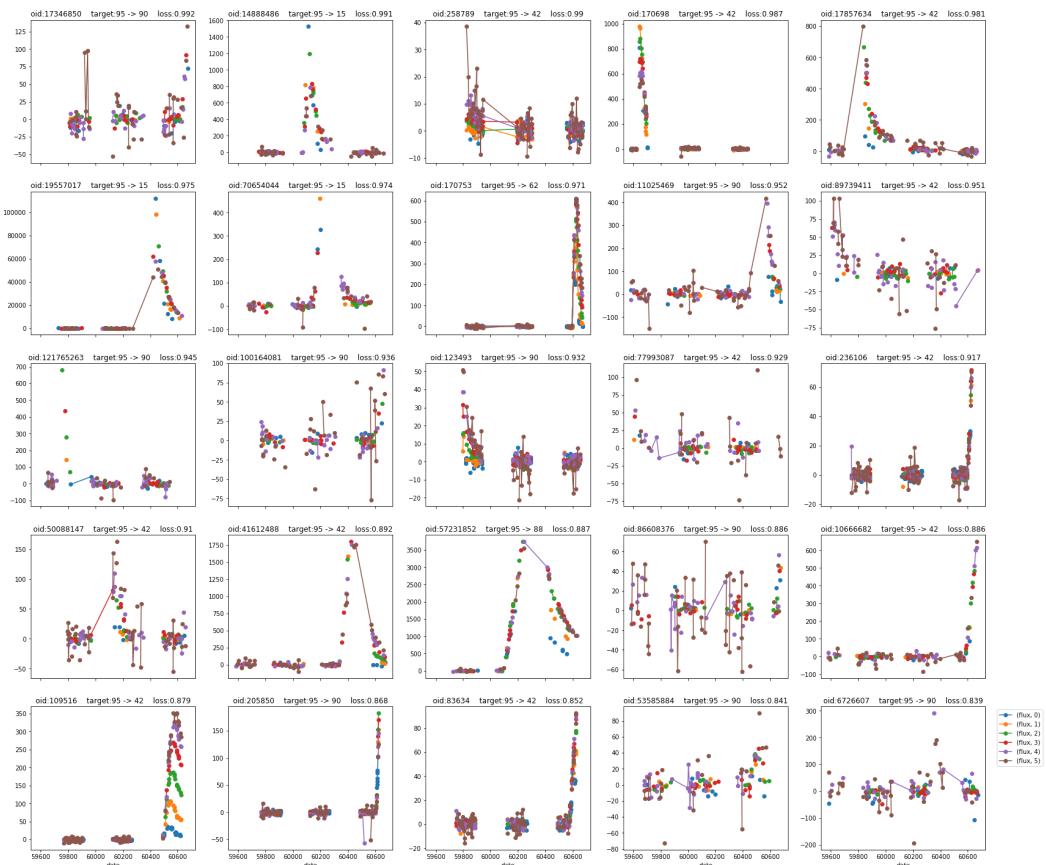
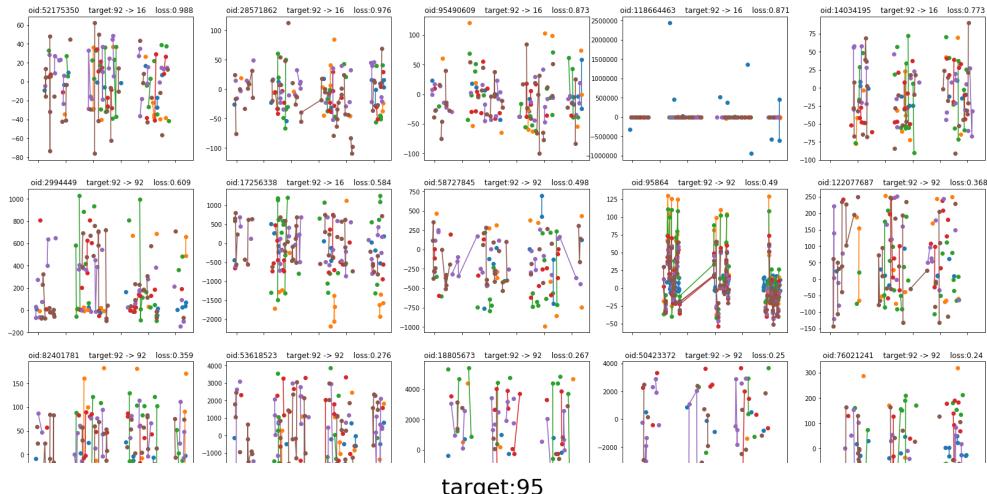


target:88

target:90



target:92



In [ ]: