In [4]:
```
from latent_3d_points.src.ae_templates import mlp_architecture_ala_iclr_18, default_train_params
from latent_3d_points.src.autoencoder import Configuration as Conf
from latent_3d_points.src.point_net_ae import PointNetAutoEncoder
from latent_3d_points.src.in_out import snc_category_to_synth_id
```

In [ ]:
```
# from tf_lab.in_out.basics import Data_Splitter

# from tf_lab.point_clouds.in_out import load_point_clouds_from_filenames, PointCloudDataSet

# from tf_lab.data_sets.shape_net import pc_loader as snc_loader

# from tf_lab.iclr.helper import load_multiple_version_of_pcs, find_best_validation_epoch_from_t
rain_stats
```

In [5]:

In [ ]:
```
%load_ext autoreload
%autoreload 2
%matplotlib inline
```

In [ ]:
```
top_data_dir = '/orions4-zfs/projects/optas/DATA/'
experiment_tag = 'mlp_with_split_1pc_usampled_bnorm_on_encoder_only'
n_pc_points = 2048

class_name = raw_input('Give me the class type: ').lower()
bneck_size = 128
ae_loss = 'emd'
```

In [ ]:
```
top_lin_dir  = '/orions4-zfs/projects/lins2/Panos_Space/DATA/'
train_params = default_train_params()

for bneck in bneck_list:
    experiment_id = '_'.join(['ae', class_name, experiment_tag, str(n_pc_points), 'pts', str(bne
ck), 'bneck', loss])
    train_dir = osp.join(top_lin_dir, 'OUT/iclr/nn_models/', experiment_id)
    create_dir(train_dir)

    reset_tf_graph()
    encoder, decoder, enc_args, dec_args = mlp_architecture_ala_iclr_18(n_pc_points, bneck)
    conf = Conf(n_input = [n_pc_points, 3],
                loss = loss,
                training_epochs = 500,
                batch_size = train_params['batch_size'],
                denoising = False,
                learning_rate = train_params['learning_rate'],
                train_dir = train_dir,
                loss_display_step = 1,
                saver_step = train_params['saver_step'],
                z_rotate = False,
                encoder = encoder,
                decoder = decoder,
                encoder_args = enc_args,
                decoder_args = dec_args
               )
    print conf
    conf.experiment_name = 'experiment_' + str(experiment_id)
    conf.held_out_step = 5
    conf.save(osp.join(train_dir, 'configuration'))
    ae = PointNetAutoEncoder(conf.experiment_name, conf)

    buf_size = 1 # flush each line
    fout = open(osp.join(conf.train_dir, 'train_stats.txt'), 'a', buf_size)
#     train_stats = ae.train(in_data['train'], conf, log_file=fout, held_out_data=in_data['val']
)
    train_stats = ae.train(in_data['train'], conf, log_file=fout, held_out_data=in_data['test'])
    fout.close()
```