



Research Progress

2022.02.04
Guan Yunyi



Summary of MVTN

- Dataset:
standard ModelNet40 (not aligned)
- Training:
 - Learn the viewpoints for **each sample**
 - All samples have the same initial viewpoint by `views_config`
(circular, learned_circular, spherical, learned_spherical, ...)
- Validation:
 - Using 2468 samples from **test dataset** in ModelNet40
 - There is no validation data

```
dset_val = ModelNet40(setup["data_dir"], "test",
```

Summary of MVTN

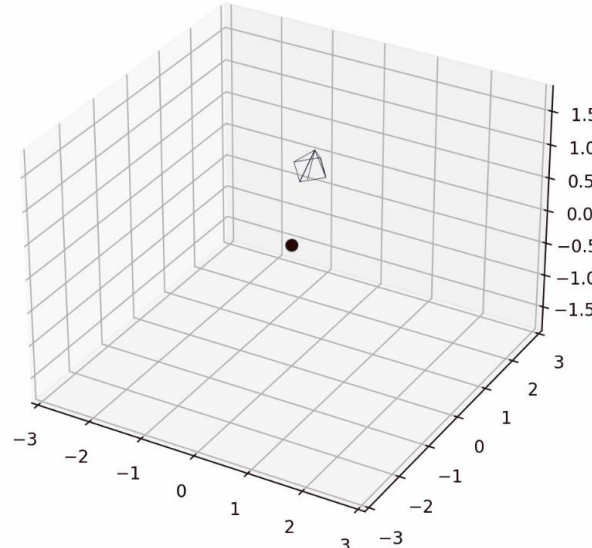
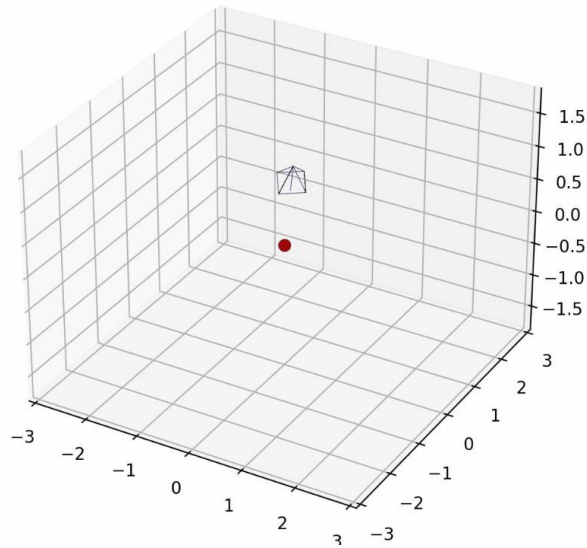
- Test:
 - Instead of predicting viewpoints and categories for new test samples, **continues the validation**
 - Is this right?

```
if setup["run_mode"] == "test_cls":
    print('\nEvaluation:')
    models_bag["mvnetwork"].eval()
    models_bag["mvtm"].eval()
    models_bag["mvrenderer"].eval()

    avg_test_acc, avg_test_loss, _ = evaluate(val_loader, models_bag, setup)
    print('\tVal Acc: %.2f - val Loss: %.4f' %
          (avg_test_acc.item(), avg_test_loss))
    print('\tCurrent best val acc: %.2f' % setup["best_acc"])
```

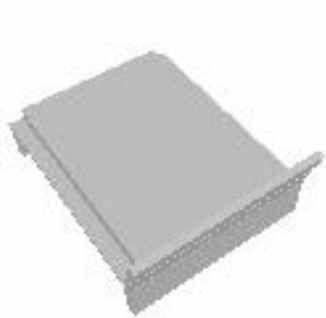
What I did- Results of `learned_circular, nb_views=1`

- Only moves within a small range of the initial viewpoint



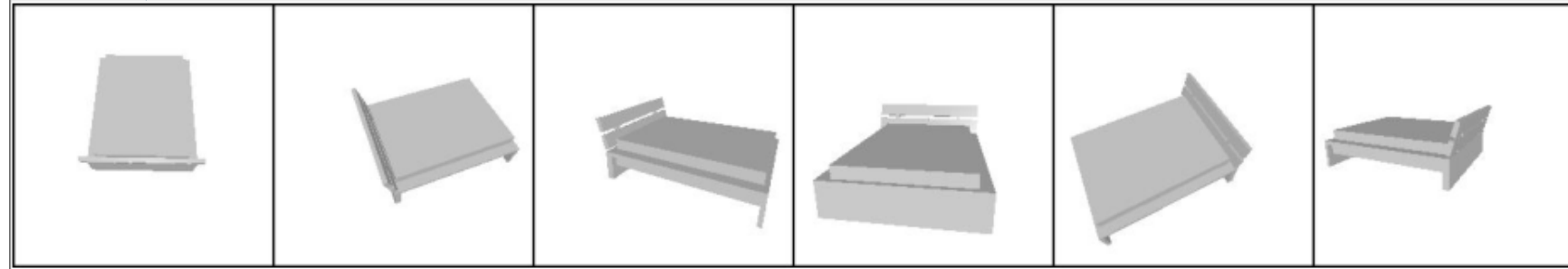
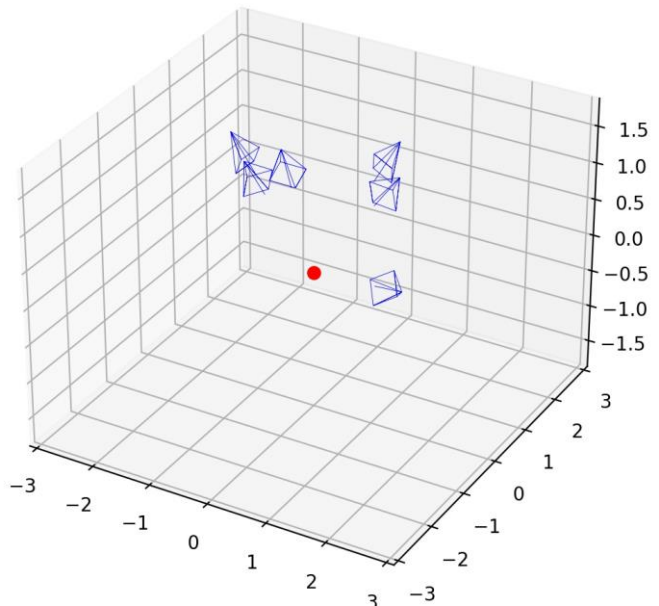
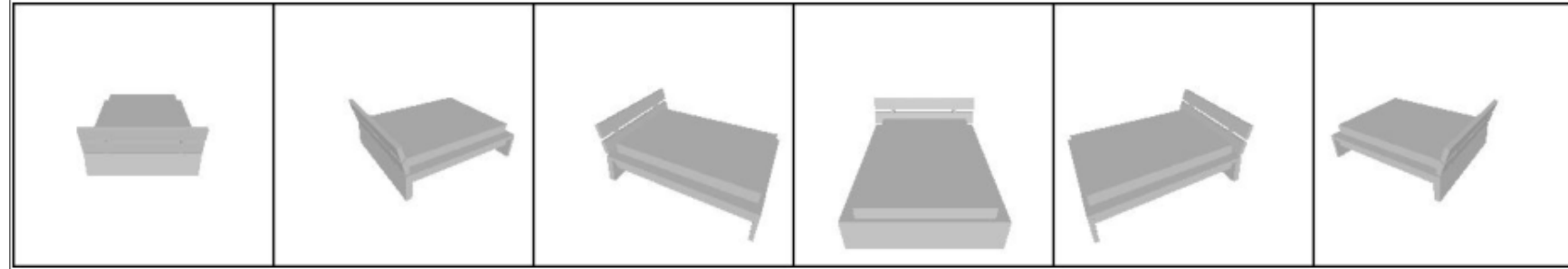
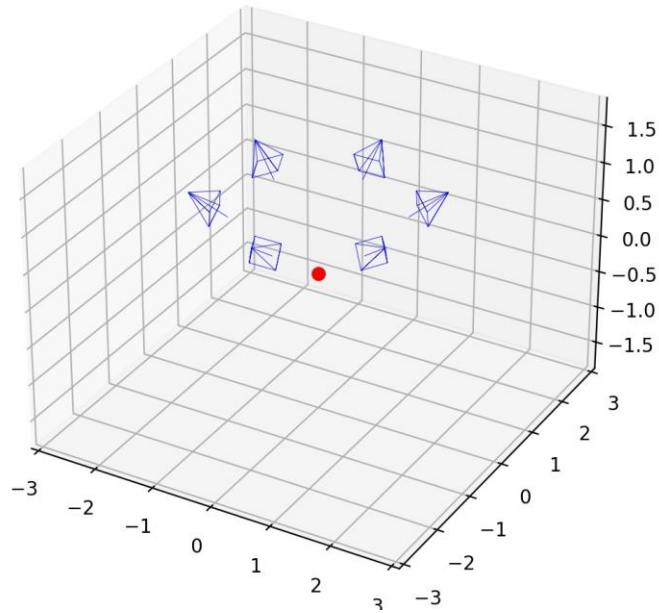
Epoch=100:

- train acc: 80.29, train loss: 0.6921
- val acc: 72.37, val Loss: 1.0615
- Current best val acc: **73.01**



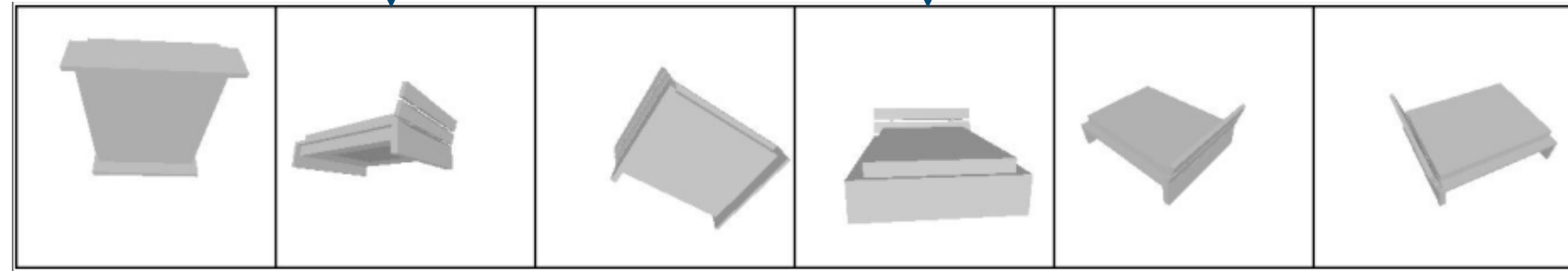
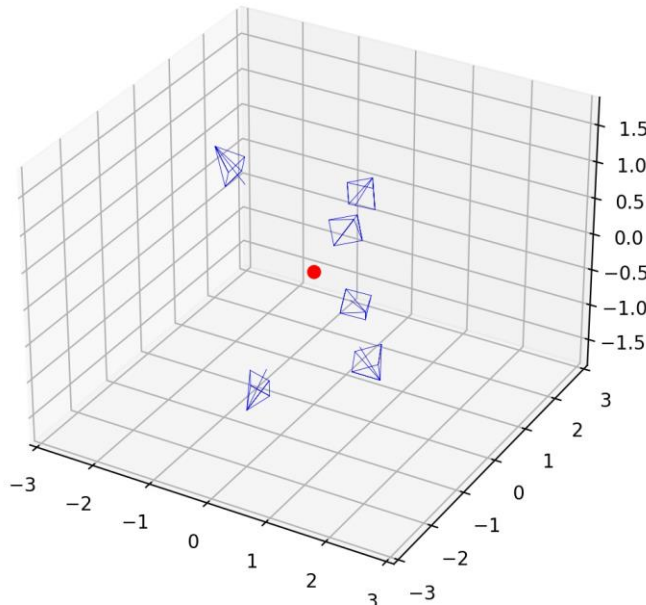
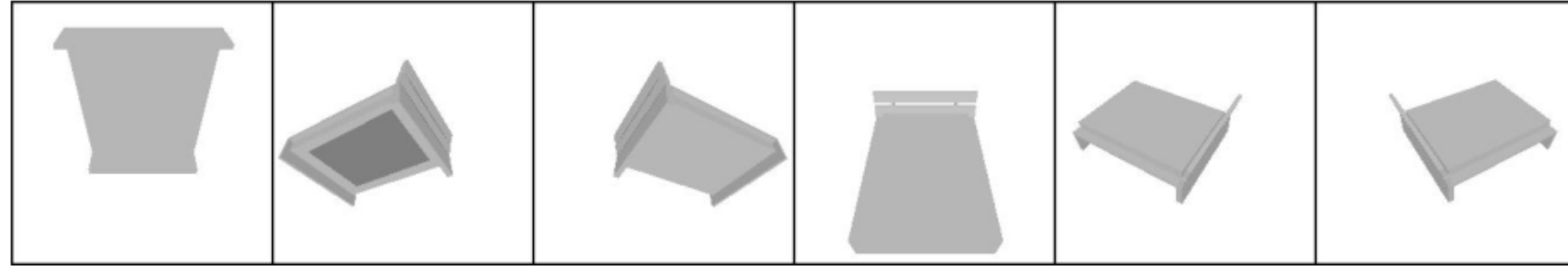
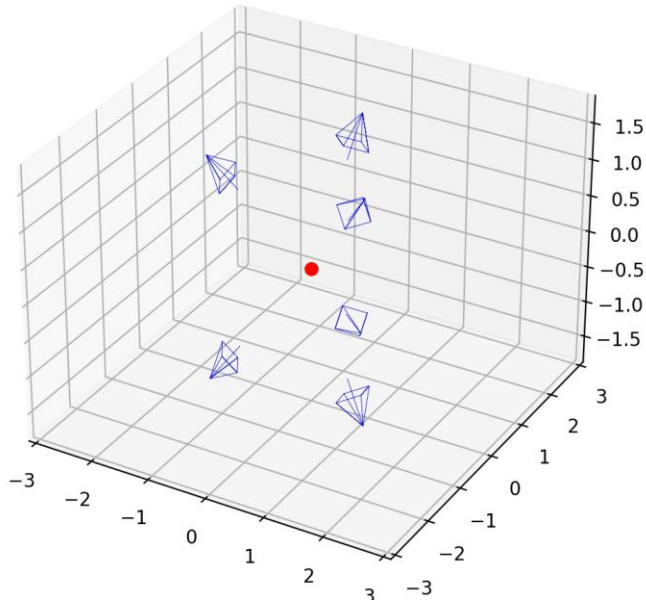
Results from the paper – learned_circular, nb_views=6

- When there are multiple viewpoints, the range of movement is larger



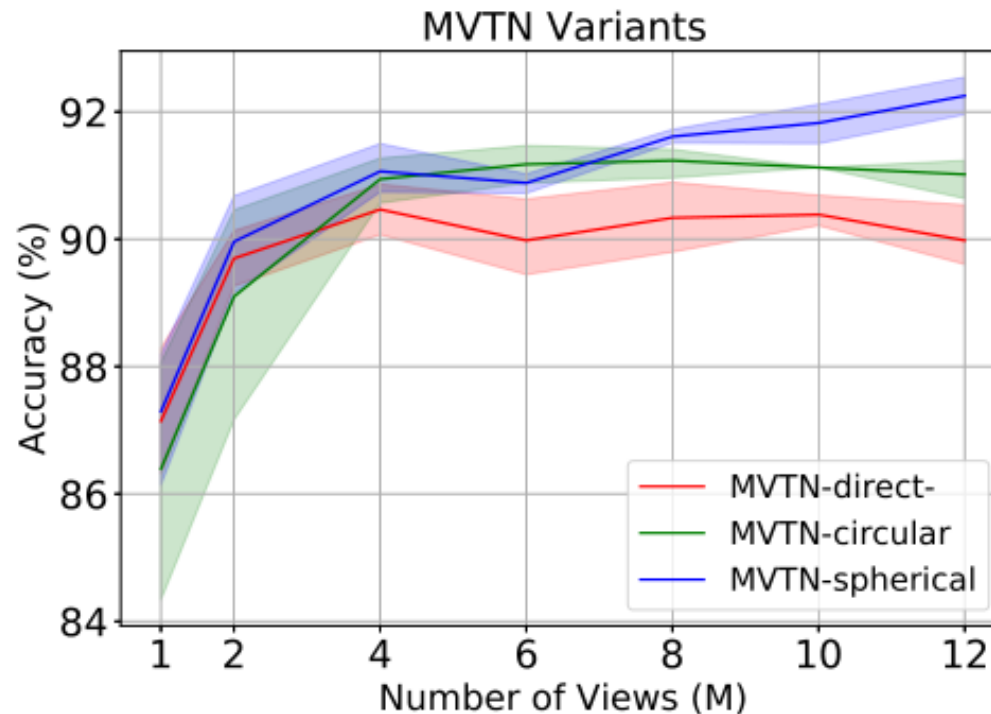
Results from the paper – learned_spherical, nb_views=6

- When there are multiple viewpoints, the range of movement is larger



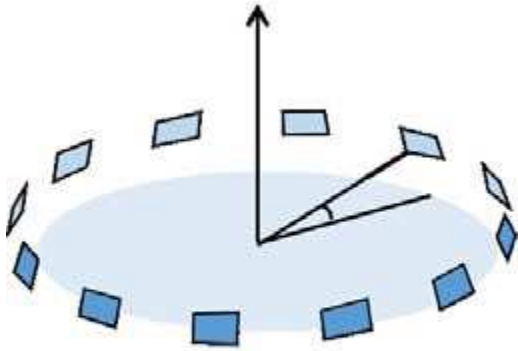
Results from the paper – with different views_config and nb_views

- `nb_views=1`, test accuracy should be around 84-88.
 - Epoch = 100, best val acc: **73.01**
 - > may be improved by increasing epoch
- (But there is not any training & test details in the paper...)



What I did- Problem with modifying codes for RotationNet

- For fixed view, view rotation candidates are used for prediction.
- For example, in case 1, the number of viewpoints = 12.



```
vcand = array([ [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],  
                [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0],  
                [ 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1],  
                [ 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, 2],  
                [ 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, 2, 3],  
                [ 5, 6, 7, 8, 9, 10, 11, 0, 1, 2, 3, 4],  
                [ 6, 7, 8, 9, 10, 11, 0, 1, 2, 3, 4, 5],  
                [ 7, 8, 9, 10, 11, 0, 1, 2, 3, 4, 5, 6],  
                [ 8, 9, 10, 11, 0, 1, 2, 3, 4, 5, 6, 7],  
                [ 9, 10, 11, 0, 1, 2, 3, 4, 5, 6, 7, 8],  
                [10, 11, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9],  
                [11, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]])
```

- How to create view rotation candidate for 1 viewpoint?

What I did- Problem in MVTN

- What should be: `nb_views == vcand.shape[1]`
 - > `nb_views` can be specified arbitrarily by command,
`vcand` has a fixed number of view candidates (`vcand.shape[1]==12`)
 - > Error of "Out of range"

```
output_ = output_.transpose(1, 2, 0) # (1, 40, 20)

scores = np.zeros((vcand.shape[0], num_classes, c_batch_size)) # (12, 40, 20)

# compute scores for all the candidate poses (see Eq.(5))
for j in range(vcand.shape[0]):
    for k in range(vcand.shape[1]):
        scores[j] = scores[j] + output_[vcand[j][k] * setup["nb_views"] + k]
# for each sample determine the best pose that maximizes the score for the target class (see Eq.(2))
for n in range(c_batch_size):
    j_max = np.argmax(scores[:, targets[n * setup["nb_views"]], n])
    # assign target labels
    for k in range(vcand.shape[1]):
        targets_[n * setup["nb_views"] * setup["nb_views"] + vcand[j_max][k] * setup["nb_views"] + k] = \
        targets[n * setup["nb_views"]]
```

Next to do

- Coding part:
 - Find the reason of small movement with 1 viewpoint
 - Modify MVTN for RotationNet with 1 viewpoint