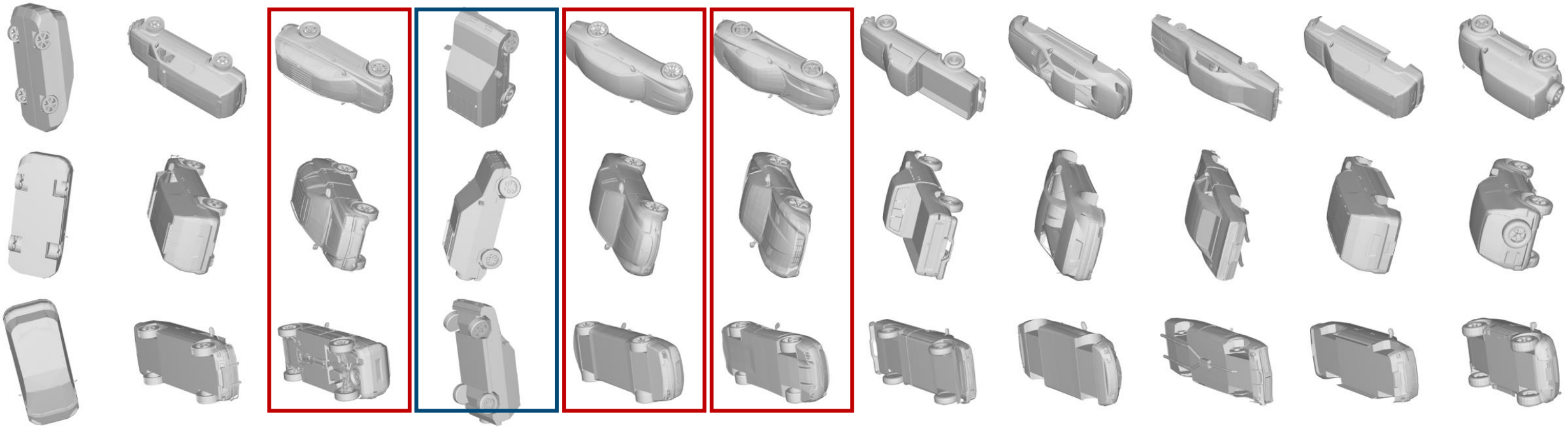
# Research Progress

2022.01.07

Guan Yunyi

# What I did – Last result

- Use test data in `ModelNet40_ori4` to do the pose estimation
- Sample: car_0198
- Completely misaligned



# What I did – Reason

- Training data path are not in the order of file-names in `train_c.txt`:

```
@@ -719,17 +696,9 @@
 ./ModelNet40_20_ori4/train/car/car_0035_016.png 7
 ./ModelNet40_20_ori4/train/car/car_0035_017.png 7
 ./ModelNet40_20_ori4/train/car/car_0035_018.png 7
-./ModelNet40_20_ori4/train/car/car_0049_014.png 7
-./ModelNet40_20_ori4/train/car/car_0049_015.png 7
-./ModelNet40_20_ori4/train/car/car_0049_016.png 7
-./ModelNet40_20_ori4/train/car/car_0049_017.png 7
-./ModelNet40_20_ori4/train/car/car_0049_018.png 7
-./ModelNet40_20_ori4/train/car/car_0049_019.png 7
-./ModelNet40_20_ori4/train/car/car_0049_020.png 7
-./ModelNet40_20_ori4/train/car/car_0059_010.png 7
-./ModelNet40_20_ori4/train/car/car_0059_011.png 7
 ./ModelNet40_20_ori4/train/car/car_0035_019.png 7
 ./ModelNet40_20_ori4/train/car/car_0035_020.png 7
+./ModelNet40_20_ori4/train/car/car_0036_001.png 7
 ./ModelNet40_20_ori4/train/car/car_0036_002.png 7
 ./ModelNet40_20_ori4/train/car/car_0036_003.png 7
 ./ModelNet40_20_ori4/train/car/car_0036_004.png 7
```

```
i=0
for j in `cat classes.txt`
do
    for k in `ls ./ModelNet40_20_ori4/train/$j/*.png`; do echo $k $i; done
> train_txt_ori/train_$j.txt
    i=`expr $i + 1`
done
```

**Recommend**:

Use `shell` instead of `python` when operating files under Linux

# What I did – Accident

- Error: No such file or directory
- Reason: training data in `ModelNet40_20_ori` is in the form of link:

  (`ModelNet40_20_ori -> modelnet40v2png_ori`)

```
car_0001_001.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_001.png*
car_0001_002.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_002.png*
car_0001_003.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_003.png*
car_0001_004.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_004.png*
car_0001_005.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_005.png*
car_0001_006.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_006.png*
car_0001_007.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_007.png*
car_0001_008.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_008.png*
car_0001_009.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_009.png*
car_0001_010.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_010.png*
car_0001_011.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_011.png*
car_0001_012.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_012.png*
car_0001_013.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_013.png*
car_0001_014.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_014.png*
car_0001_015.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_015.png*
car_0001_016.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_016.png*
car_0001_017.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_017.png*
car_0001_018.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_018.png*
car_0001_019.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_019.png*
car_0001_020.png -> ../../.././modelnet40v2png_ori4/car/train/car_0001_020.png*
car_0002_001.png -> ../../.././modelnet40v2png_ori4/car/train/car_0002_001.png*
car_0002_002.png -> ../../.././modelnet40v2png_ori4/car/train/car_0002_002.png*
car_0002_003.png -> ../../.././modelnet40v2png_ori4/car/train/car_0002_003.png*
car_0002_004.png -> ../../.././modelnet40v2png_ori4/car/train/car_0002_004.png*
```

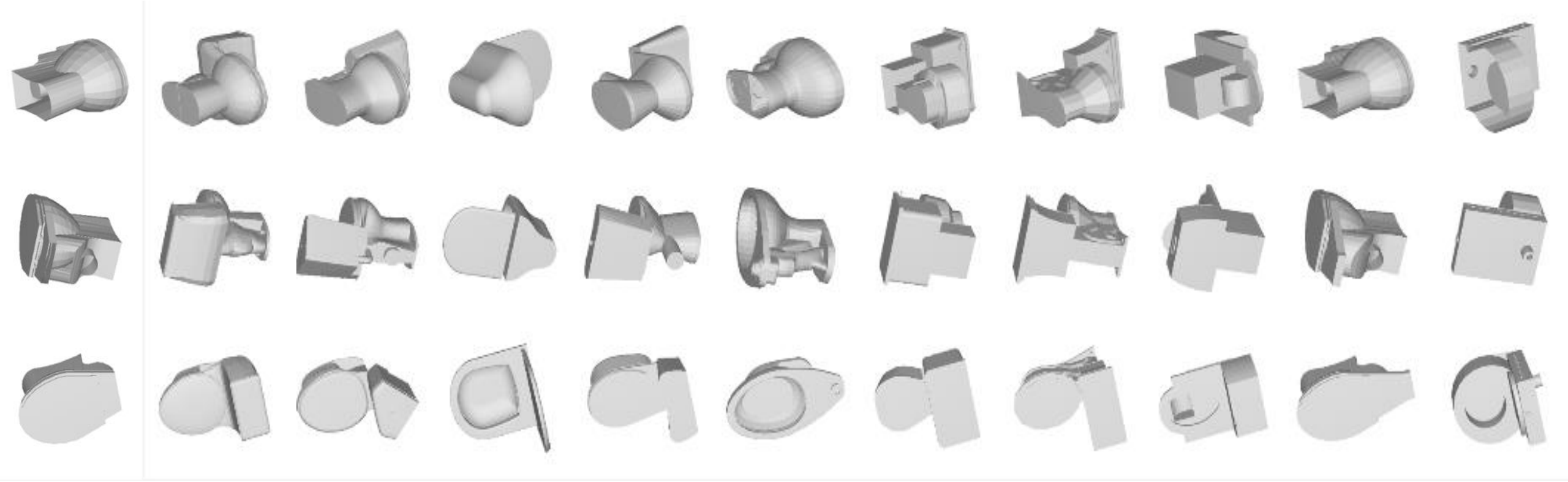
**Recommend**:

(1) Try not to move files that are not generated by yourself

(2) "No such file or directory" is usually path problem, use absolute path and `ll -h` to check the file information as a hint :)

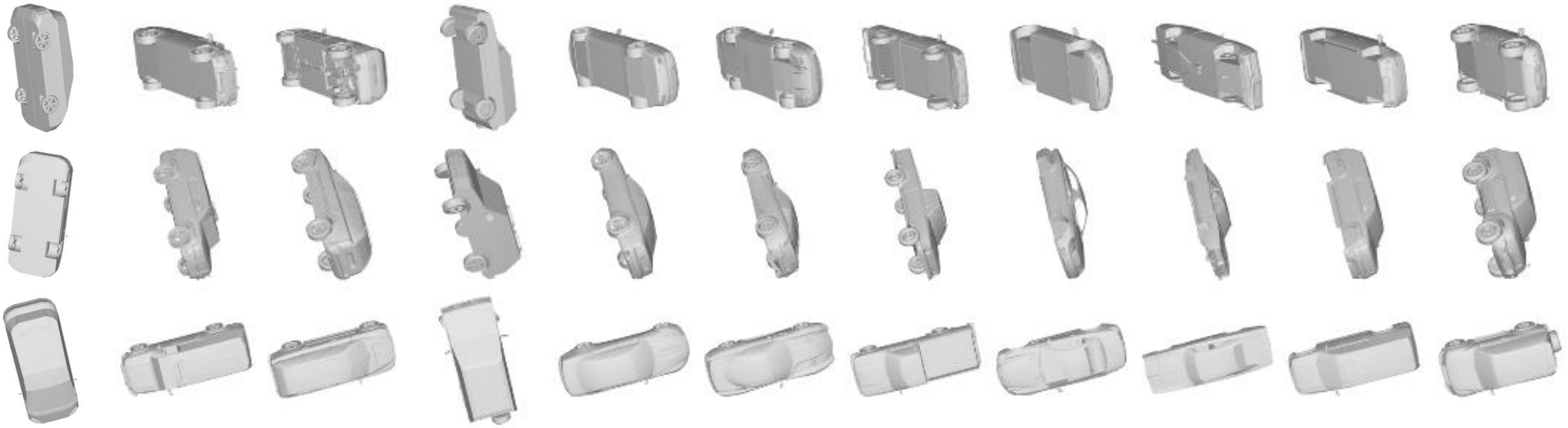
# What I did – Result of pose estimation

- Sample: toilet_0345
- Good alignment!



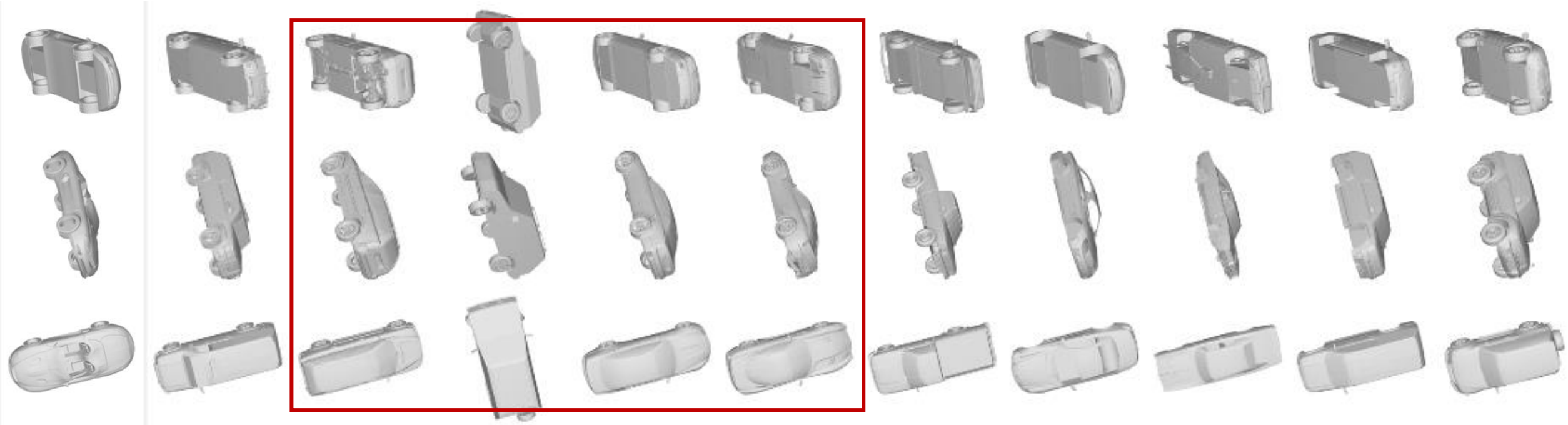
# What I did – Result of pose estimation

- Sample: car_0198, result is the same as car_0199
- Occasional failure example



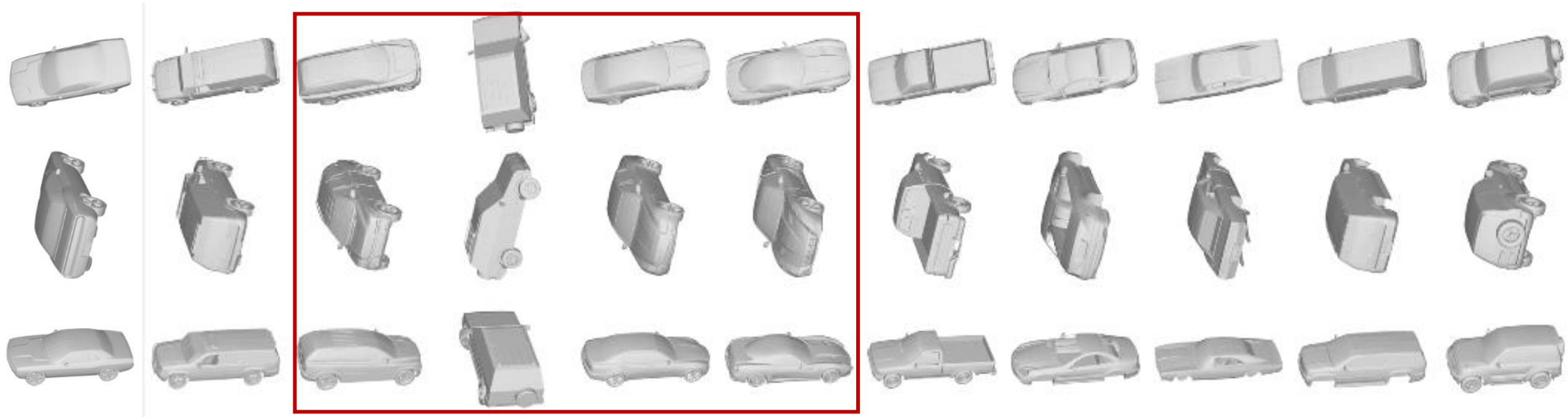
# What I did – Result of pose estimation

- Sample: car_0199
- Bad alignment(reversed) on training sample 002, 003, 004 and 005



# What I did – Result of pose estimation

- Sample: car_0200
- Bad alignment(reversed) on training sample 002, 003, 004 and 005

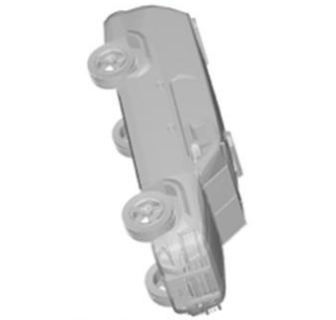


# What I did – Result of pose estimation

- Reason is <u>on alignment on training samples</u> but not test

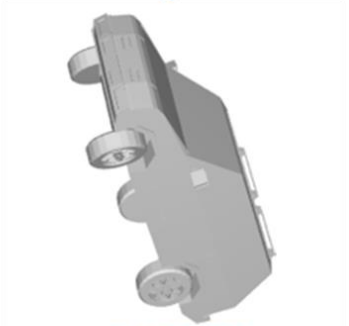
reference_pose_car.txt

```
./ModelNet40_20_ori4/train/car/car_0001_003.png
./ModelNet40_20_ori4/train/car/car_0001_009.png
./ModelNet40_20_ori4/train/car/car_0001_014.png
...
./ModelNet40_20_ori4/train/car/car_0002_003.png
./ModelNet40_20_ori4/train/car/car_0002_009.png
./ModelNet40_20_ori4/train/car/car_0002_014.png
...
./ModelNet40_20_ori4/train/car/car_0003_003.png
./ModelNet40_20_ori4/train/car/car_0003_009.png
./ModelNet40_20_ori4/train/car/car_0003_014.png
```



car_0001_003.png

car_0002_003.png



car_0003_003.png

shown files

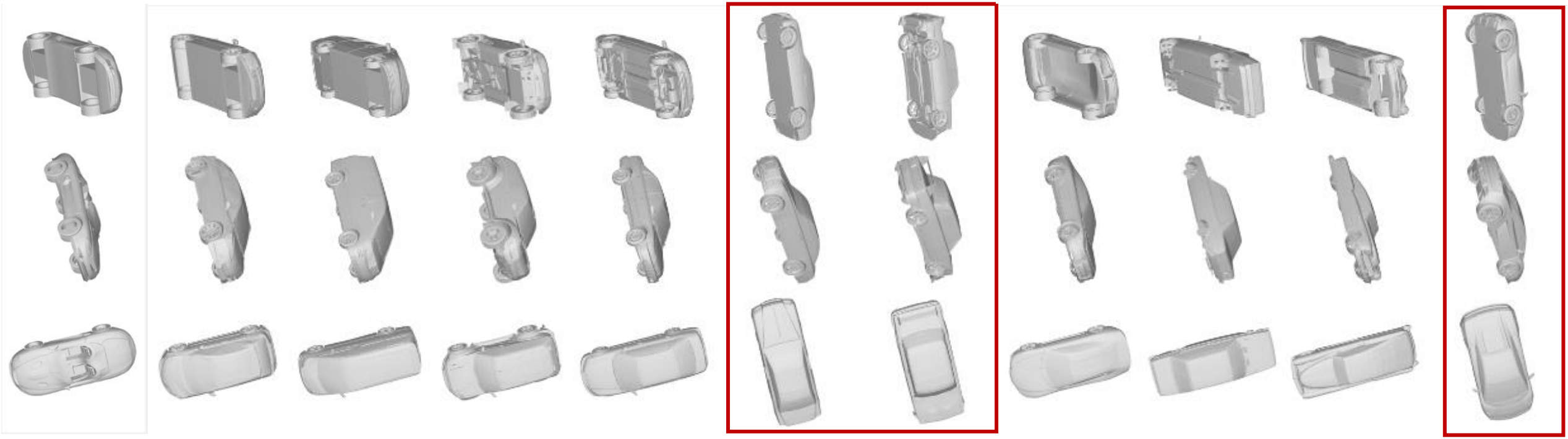
```
19
ModelNet40_20_ori4/train/car/car_0001_019.png
ModelNet40_20_ori4/train/car/car_0002_019.png
ModelNet40_20_ori4/train/car/car_0003_019.png
ModelNet40_20_ori4/train/car/car_0004_019.png
ModelNet40_20_ori4/train/car/car_0005_019.png
ModelNet40_20_ori4/train/car/car_0006_019.png
ModelNet40_20_ori4/train/car/car_0007_019.png
ModelNet40_20_ori4/train/car/car_0008_019.png
ModelNet40_20_ori4/train/car/car_0009_019.png
ModelNet40_20_ori4/train/car/car_0010_019.png
3
ModelNet40_20_ori4/train/car/car_0001_003.png
ModelNet40_20_ori4/train/car/car_0002_003.png
ModelNet40_20_ori4/train/car/car_0003_003.png
ModelNet40_20_ori4/train/car/car_0004_003.png
ModelNet40_20_ori4/train/car/car_0005_003.png
ModelNet40_20_ori4/train/car/car_0006_003.png
ModelNet40_20_ori4/train/car/car_0007_003.png
```

# What I did – Result of pose estimation (random training samples)

- Selecting training samples randomly instead of selecting first 10
- Sample: car_0199

-> bad in training but good in testing

```
19
ModelNet40_20_ori4/train/car/car_0095_019.png
ModelNet40_20_ori4/train/car/car_0064_019.png
ModelNet40_20_ori4/train/car/car_0066_019.png
ModelNet40_20_ori4/train/car/car_0118_019.png
ModelNet40_20_ori4/train/car/car_0181_019.png
ModelNet40_20_ori4/train/car/car_0050_019.png
ModelNet40_20_ori4/train/car/car_0113_019.png
ModelNet40_20_ori4/train/car/car_0164_019.png
ModelNet40_20_ori4/train/car/car_0136_019.png
ModelNet40_20_ori4/train/car/car_0043_019.png
```

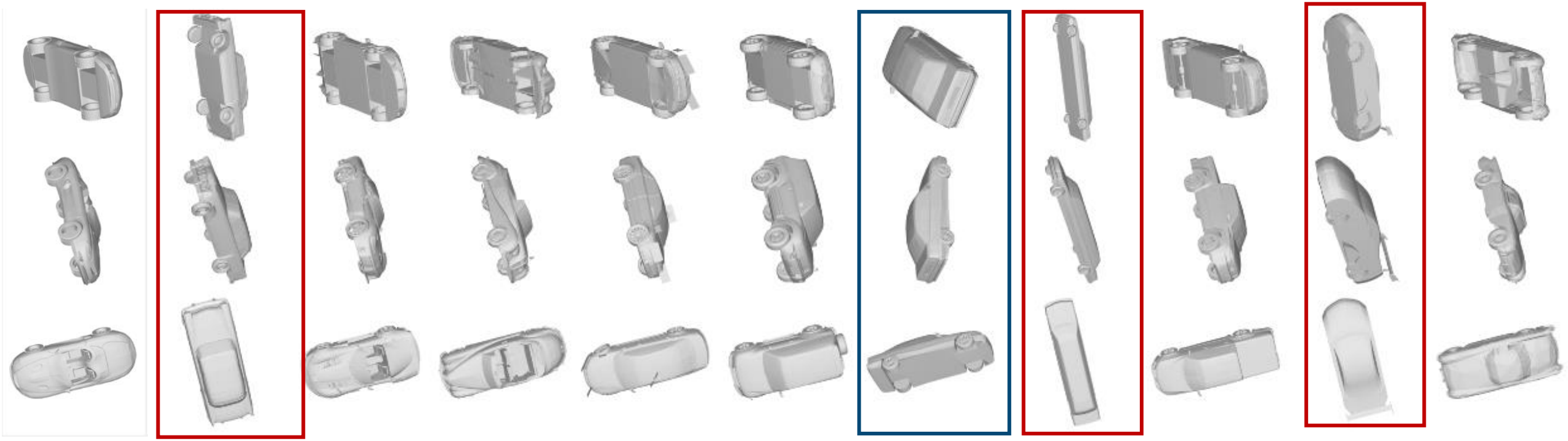


# What I did – Result of pose estimation (random training samples)

- Sample: car_0199 again

-> bad in training but good in testing

```
19
ModelNet40_20_ori4/train/car/car_0040_019.png
ModelNet40_20_ori4/train/car/car_0047_019.png
ModelNet40_20_ori4/train/car/car_0068_019.png
ModelNet40_20_ori4/train/car/car_0167_019.png
ModelNet40_20_ori4/train/car/car_0010_019.png
ModelNet40_20_ori4/train/car/car_0070_018.png
ModelNet40_20_ori4/train/car/car_0083_019.png
ModelNet40_20_ori4/train/car/car_0124_019.png
ModelNet40_20_ori4/train/car/car_0133_019.png
ModelNet40_20_ori4/train/car/car_0093_019.png
```

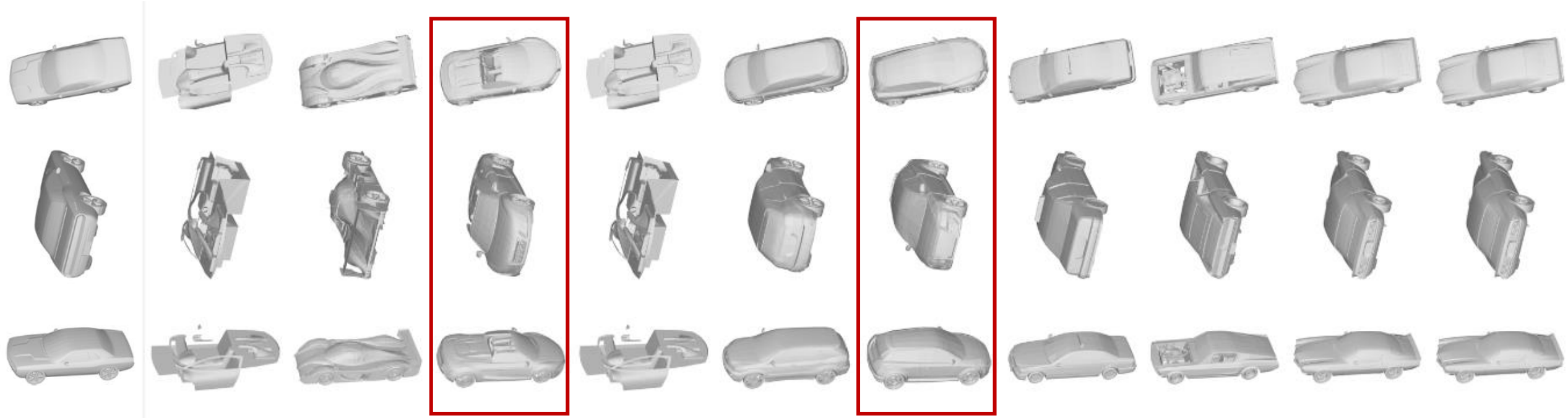


# What I did – Result of pose estimation (random training samples)

- Sample: car_0200

-> bad in training but good in testing

```
17
ModelNet40_20_ori4/train/car/car_0031_017.png
ModelNet40_20_ori4/train/car/car_0168_017.png
ModelNet40_20_ori4/train/car/car_0061_017.png
ModelNet40_20_ori4/train/car/car_0031_017.png
ModelNet40_20_ori4/train/car/car_0086_017.png
ModelNet40_20_ori4/train/car/car_0179_017.png
ModelNet40_20_ori4/train/car/car_0156_017.png
ModelNet40_20_ori4/train/car/car_0041_017.png
ModelNet40_20_ori4/train/car/car_0044_017.png
ModelNet40_20_ori4/train/car/car_0044_017.png
```



# What I did – Result of pose estimation (random training samples)

- Sample: airplane_0700

-> Relatively good alignment

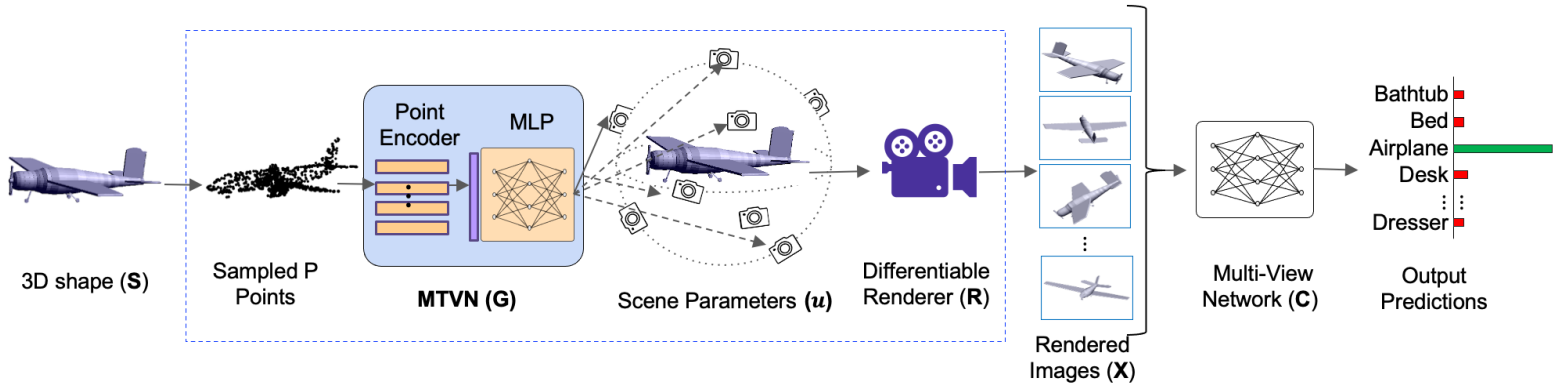
```
19
ModelNet40_20_ori4/train/airplane/airplane_0414_019.png
ModelNet40_20_ori4/train/airplane/airplane_0027_019.png
ModelNet40_20_ori4/train/airplane/airplane_0118_019.png
ModelNet40_20_ori4/train/airplane/airplane_0124_019.png
ModelNet40_20_ori4/train/airplane/airplane_0026_019.png
ModelNet40_20_ori4/train/airplane/airplane_0621_019.png
ModelNet40_20_ori4/train/airplane/airplane_0306_019.png
ModelNet40_20_ori4/train/airplane/airplane_0408_019.png
ModelNet40_20_ori4/train/airplane/airplane_0399_019.png
ModelNet40_20_ori4/train/airplane/airplane_0477_019.png
```

# What I did – Conclusion of pose estimation

- Pose estimation performed well.
- Code can be modified to determine whether the problem is in the training or test phase
- All the results can be improved by adjusting the trained model.

(Using ResNet instead of AlexNet, adjusting hyper-parameters, …)

# What I did- Error while running MVTN



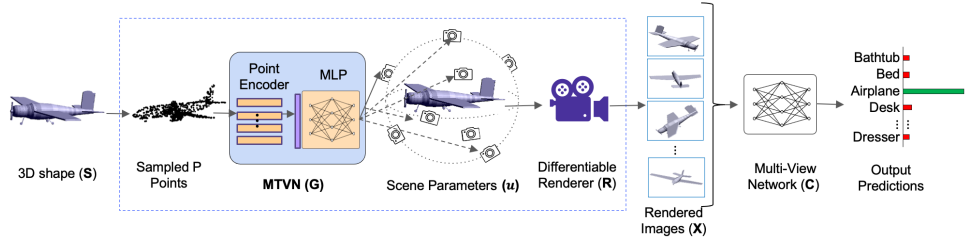

- RuntimeError: CUDA out of memory.
- Solutions for `Pytorch`:

(1) Reduce the batch size

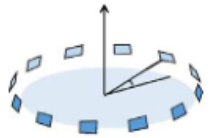
(2) Release the GPU by `torch.cuda.empty_cache()`

(3) Add `with torch.no_grad()` to the part that does not need to calculate the gradient in both training and testing code

# What I did- Result of MVTN




- Dataset: `ModelNet40` (input is mesh)
- RotationNet: case1(circular) -> `nb_view = 12`



- Running time: about 22 min for one epoch (default `epochs = 100`)

-> `epochs = 10, batch_size = 12`

- Result:

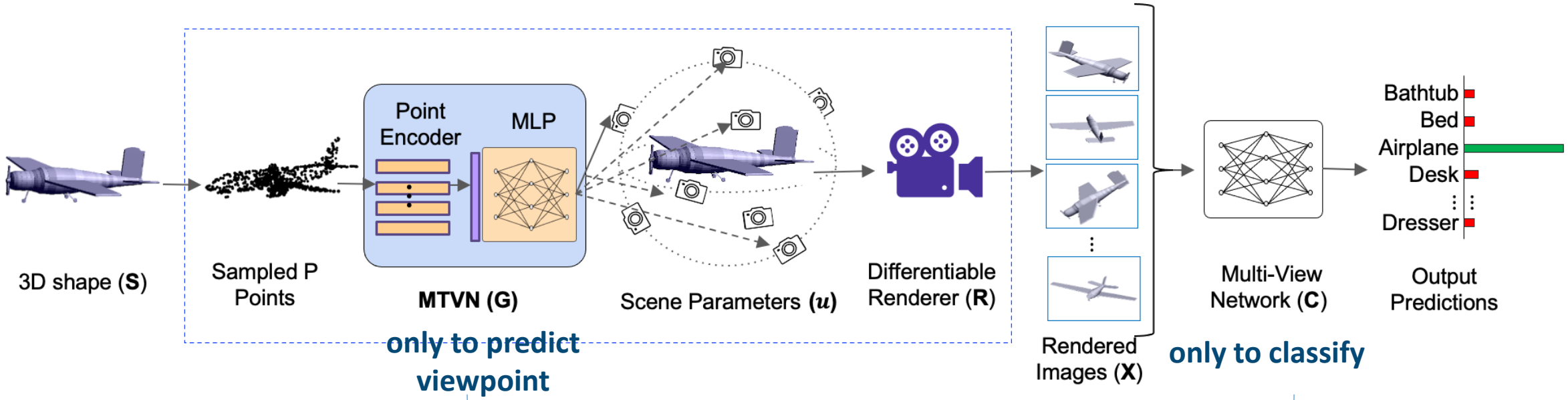
```
train acc: 77.98 - train Loss: 15.5652

Evaluation:
    Val Acc: 76.13 - val Loss: 13.0454
    Current best val acc: 75.45
    Saving checkpoint - Acc: 76.13
```

But there is no code for saving scene parameters with RotationNet…

-> Try running with MVCNN now

# Problem- How to modify MVTN to reduce training time?




$$\arg\min_{\theta_C,\theta_G} \sum_{n}^{N} L\left(\mathbf{C}(\mathbf{R}(\mathbf{S}_n, \mathbf{u}_n)),\ y_n\right),$$
$$\text{s. t.} \quad \mathbf{u}_n = \mathbf{u}_{\text{bound}}.\tanh(\mathbf{G}(\mathbf{S}_n))$$

**C and G are trained jointly on the same loss**

```
outputs = models_bag["mvnetwork"](rendered_images)[0]
loss = criterion(outputs, targets)
```

- Loss is calculated based on the output of classifier C and labels, and is also used to train G

-> Cannot only keep G and delete C directly…

# Next to do

- Coding part:
- Keep trying to run MVTN
- Test realistic dataset MIRO or ScanObjectNN? on RotationNet
- Consider how to determine the rank of scene parameters
- Paper reading