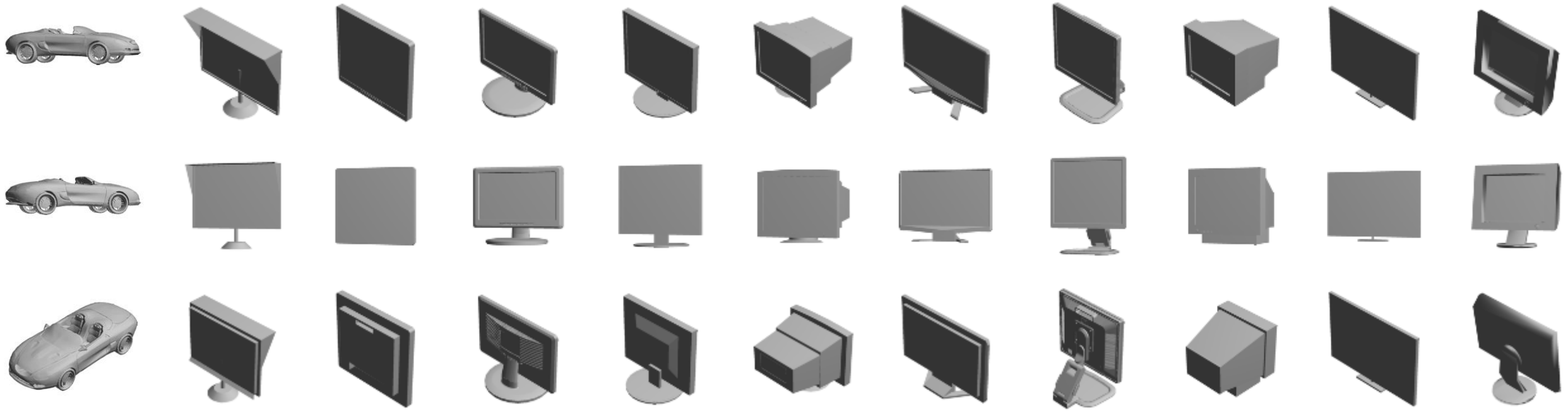# Research Progress

2021.12.17
Guan Yunyi

aklab

# What I did – Last result

- Input 3 images of one sample and test them with trained model
- Strange result: wrong predictions and subtle alignment of poses...

  ->the performance of aligning training data(making reference poses) is good, but there is bugs in prediction part

# What I did - Cause of classification error

- Incorrectly used model trained with the dataset of `ModelNet40_ori4`.

- Difference between `ModelNet40` and `ModelNet40_ori4`:
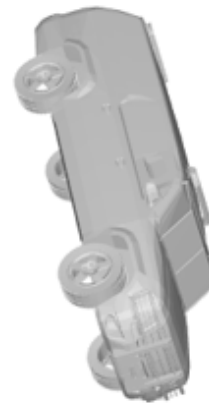
car_000000079_001.png

car_000000079_002.png

car_0001_001.png

car_0001_002.png



Contains images of the same viewpoint rotated 4 times

There is no rotation, one image per viewpoint
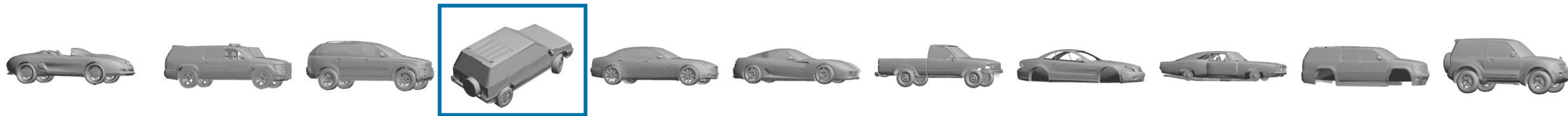
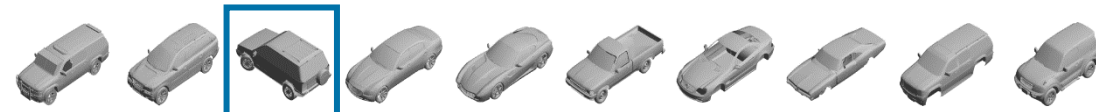car_000000079_003.png

car_000000079_004.png

car_0001_003.png

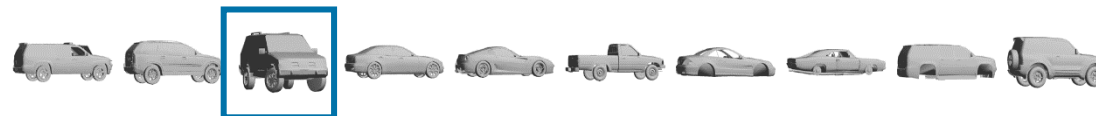# What I did - Result of pose estimation

- Retrained a new model using `ModelNet40`

- Follow the process: save scores ->make reference poses ->pose estimation



- Sample: car_000000079

`caffe`
version
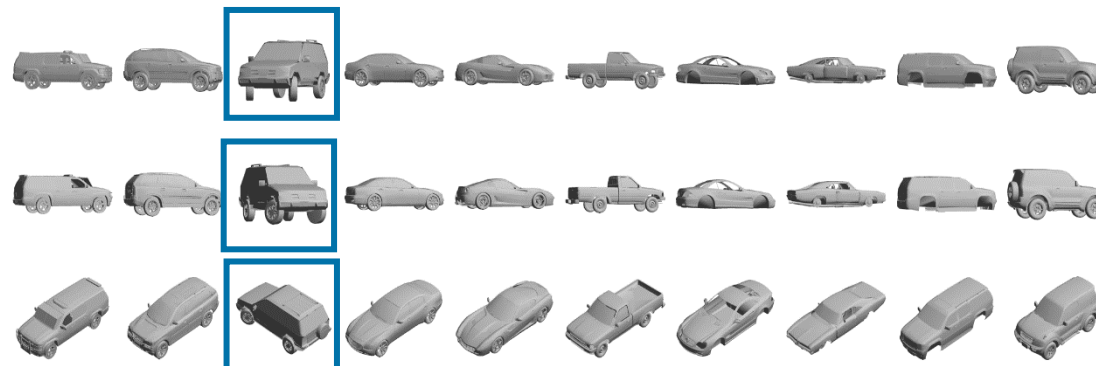
# What I did - Result of pose estimation

- Sample: car_000000073
- Completely misaligned.

# What I did - Result of pose estimation

- Sample: toilet_000000020
- Completely aligned

# What I did - Result of pose estimation

- Use test data in `ModelNet40_ori4` to do the pose estimation

- Sample: car_0198

- Completely misaligned

# What I did - Result of pose estimation
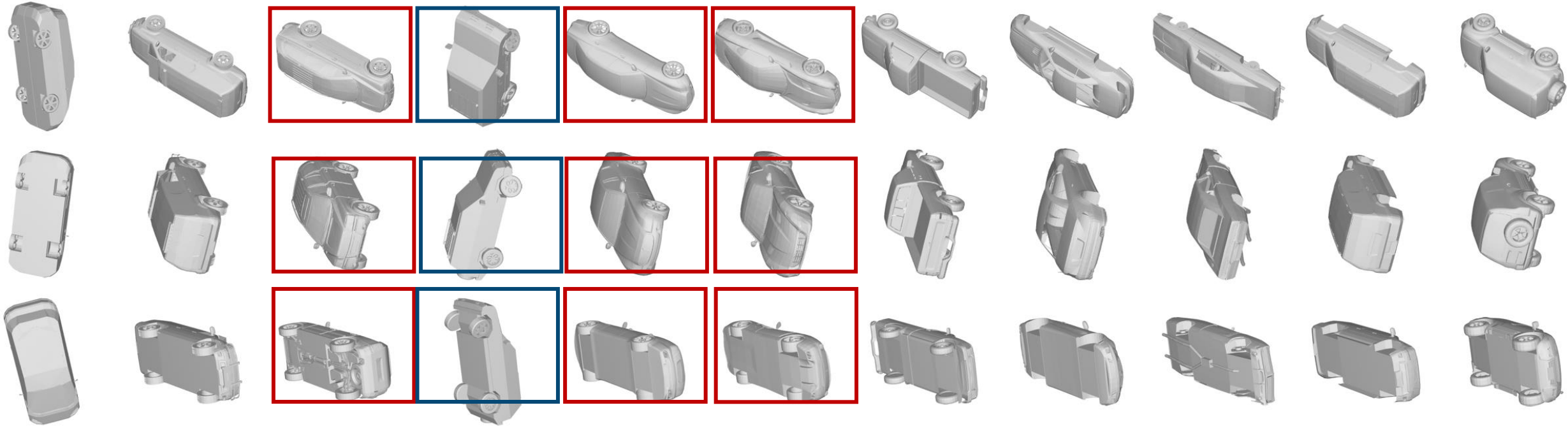
- Sample: toilet_0345

- Completely misaligned.
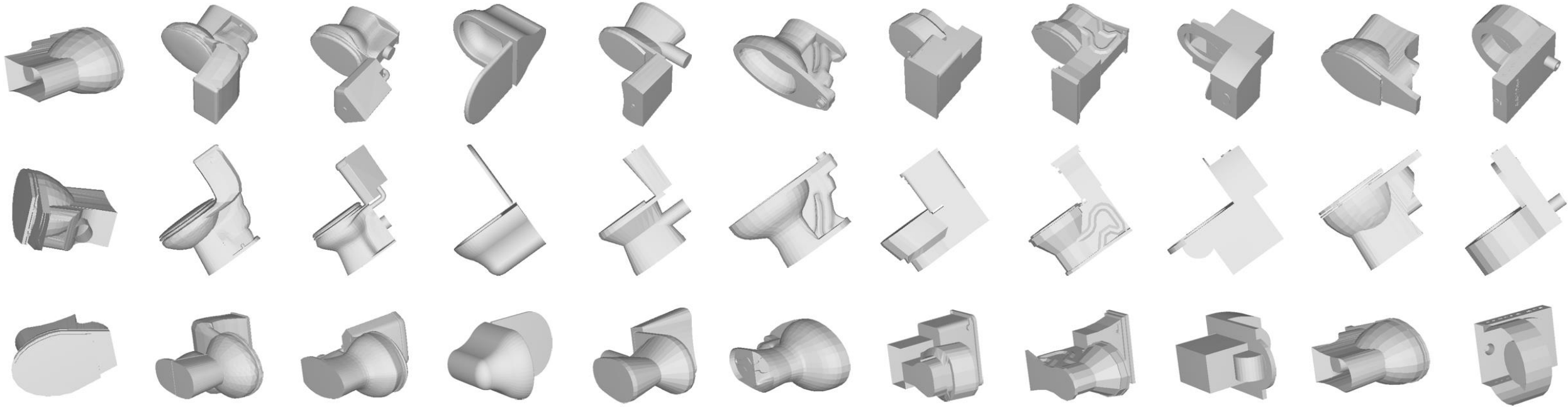
## Problems-Epoch in testing

- While pose estimation, if set `model.train()`, `epochs>1`, results of alignment seems to be better.

Epochs=1



Epochs=200
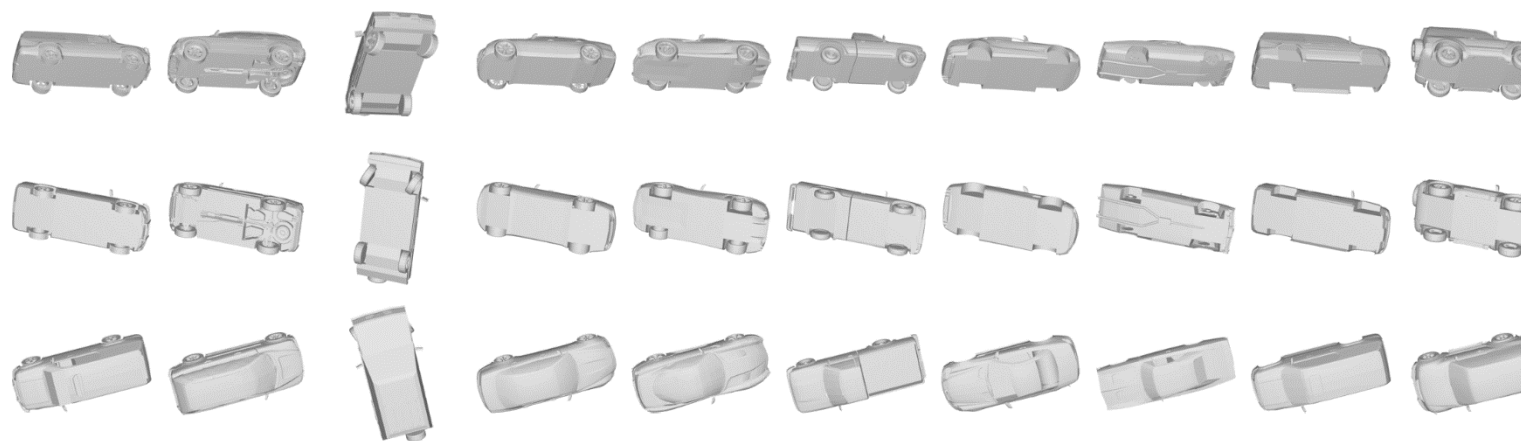


- But it is more of <u>going on training</u> than testing.

- If set `model.eval()`, `epochs>1`, results will be the same as `epochs==1`

- Is epoch in testing not meaningful?

## Problems-Batch size in testing

- While inputting 3 images(epoch==1):

- If `batch size=3`, there will be three elements in `DataLoader`.
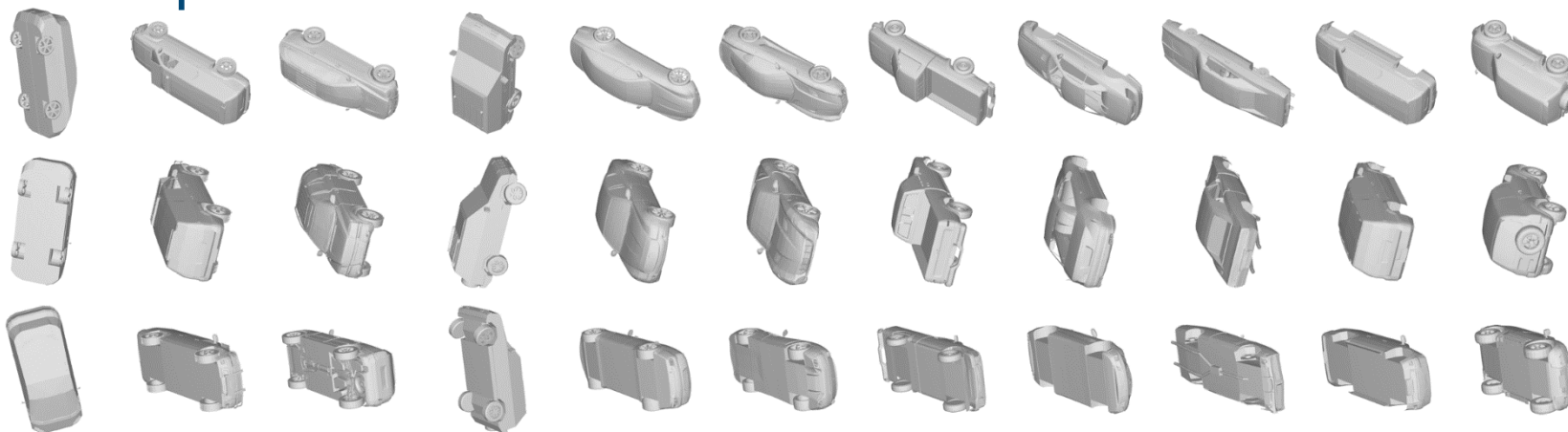
  -> All three images are sent into the network together

  -> Not input new view images <u>sequentially</u>.

- If `batch size=1`(also set `model.train()`), there is only one element in `DataLoader`.

  -> Predicts class and estimates pose for each input image at one time.

  -> There will be three outputs while `caffe` version only has one.

  -> Also the <u>going on training</u> or <u>testing</u> problem.

## Problems-View pictures on lab sever

- My method: use `Pycharm` to connect remotely and view pictures in it.

- Disadvantages: sometimes very slow, very time-consuming!
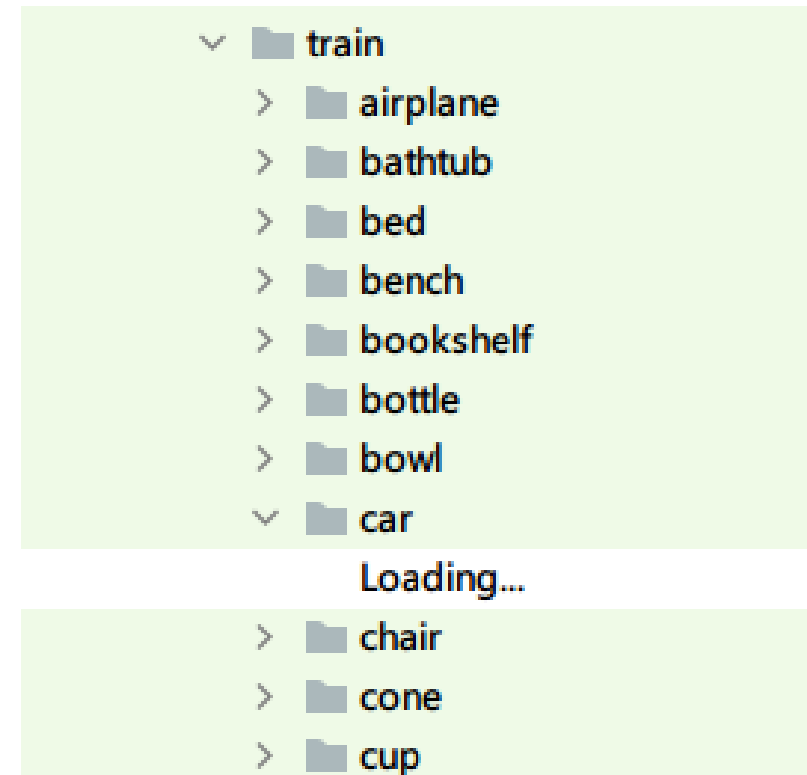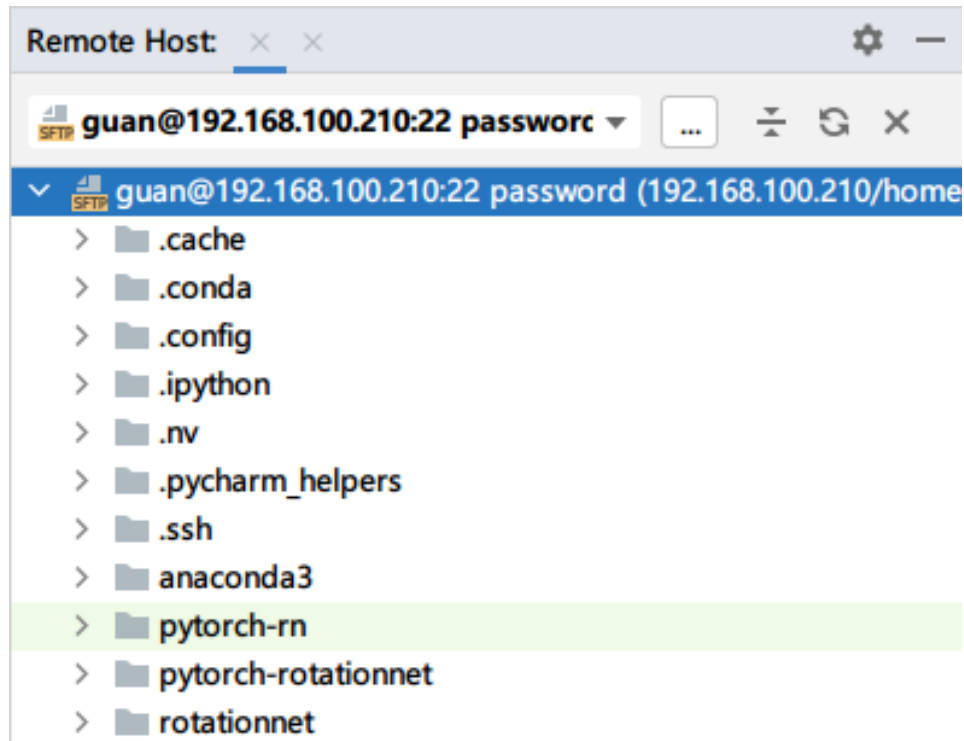
# What I did- Paper reading

Hamdi, Abdullah & Giancola, Silvio & Bing, Li & Thabet, Ali & Ghanem, Bernard. (2020). MVTN: Multi-View Transformation Network for 3D Shape Recognition.
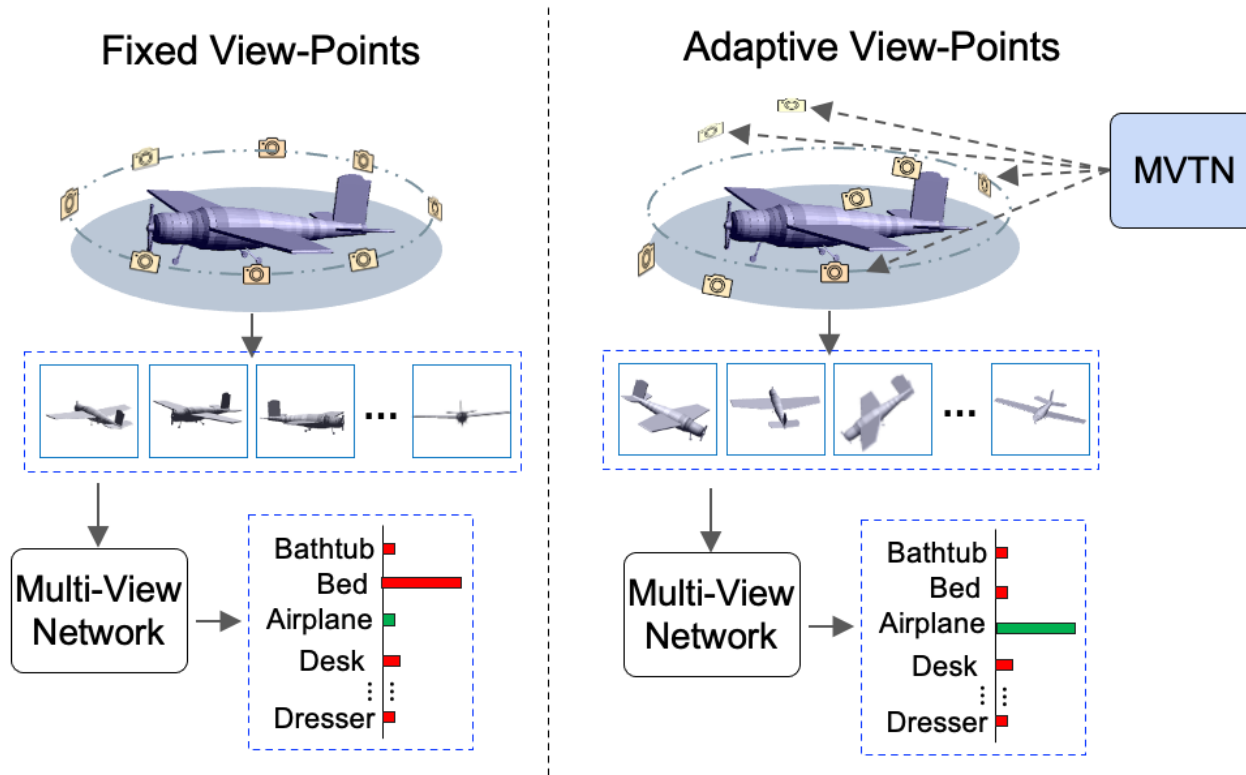


- Previous methods: **fixed rendered 3D datasets**.
- MVTN: regresses optimal view-points based on **differentiable rendering**.

The scene parameters **u**(e.g. azimuth and elevation angles of cameras) for each shape(sample) can be different.

# What I did- Paper reading



$$\arg\min_{\theta_C, \theta_G} \sum_n^N L\left(C(R(S_n, u_n)), y_n\right),$$

$$\text{s. t.} \quad u_n = u_{bound}.\tanh(G(S_n))$$

**C and G are trained jointly on the same loss**
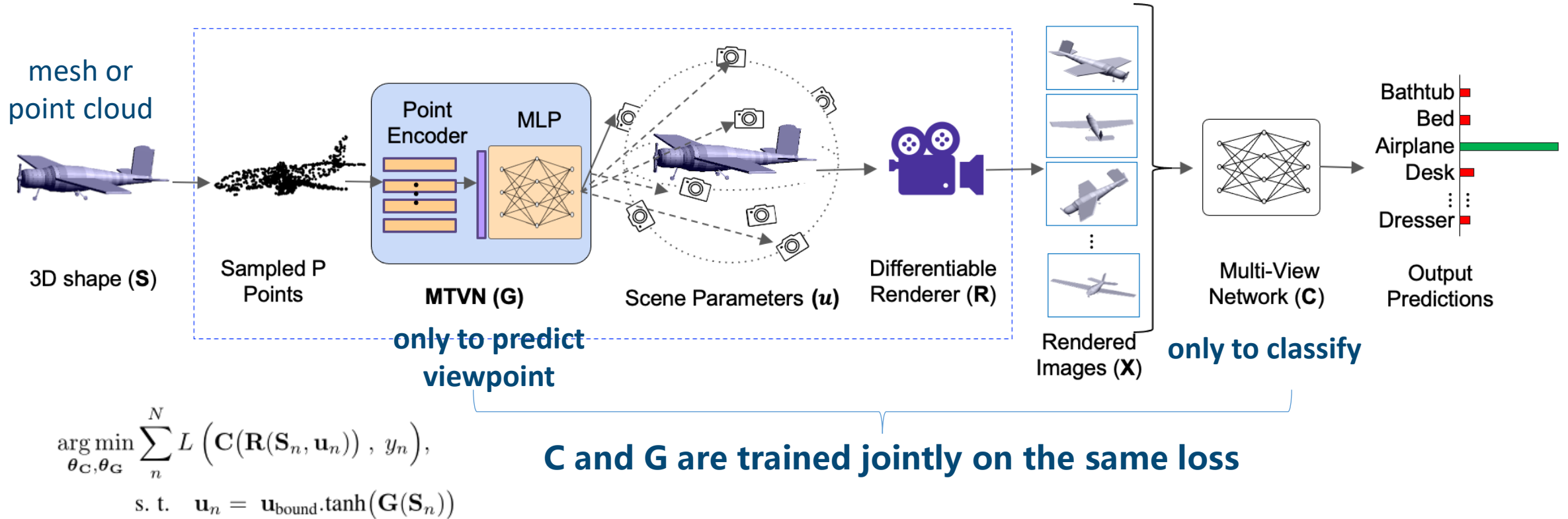
- **G**: (1) <u>Point encoder</u>(PointNet): extracts coarse features from **S**

  (2) <u>MLP</u>: predict optimal view-points (scene parameters **u**)

- **R**: renderer(Pytorch3D), inputs **S** + **u**, outputs multi-view images **X**

- **C**: classifier(using existing methods e.g. RotationNet)

# What I did- Paper reading

3D Shape Classification on ModelNet40

| Method | Data Type | Classification Accuracy (Per-Class) | (Overall) |
|---|---|---|---|
| VoxNet [52] | Voxels | 83.0 | 85.9 |
| PointNet [55] | Points | 86.2 | 89.2 |
| PointNet++ [57] | Points | - | 91.9 |
| PointCNN [46] | Points | 88.1 | 91.8 |
| DGCNN [66] | Points | 90.2 | 92.2 |
| SGAS [44] | Points | - | 93.2 |
| KPConv[63] | Points | - | 92.9 |
| PTransformer[78] | Points | **90.6** | **93.7** |
| MVCNN [61] | 12 Views | 90.1 | 90.1 |
| GVCNN [19] | 12 Views | 90.7 | 93.1 |
| ViewGCN [67] | 20 Views | **96.5** | **97.6** |
| ViewGCN [67]* | 12 views | 90.7 | 93.0 |
| ViewGCN [67]* | 20 views | 91.3 | 93.3 |
| MVTN (ours)* | 12 Views | 92.0 | **93.8** |
| MVTN (ours)* | 20 Views | **92.2** | 93.5 |

Integrating MVTN with Multi-View Networks
(on ModelNet40 with 12 views)

| View Selection | Multi-View Networks | | |
|---|---|---|---|
| | MVCNN[61] | RotNet[38] | ViewGCN[67] |
| fixed views | 90.4 | 91.6 | 93.0 |
| with MVTN | **92.6** | **93.2** | **93.8** |

# What I did- Paper reading

- Project deep feature of the last layer in **C** into a more expressive space using **LFDA reduction**, and consider the reduced feature as shape signature.
- At test time, shape signatures are used to retrieve the most similar shapes.

$$AP = \frac{1}{GTP} \sum_n^N \frac{\mathbb{1}(\mathbf{S}_n)}{n}$$

| Method | Data Type | Shape Retrieval (mAP) | |
| --- | --- | --- | --- |
| | | **ModelNet40** | **ShapeNet Core** |
| LFD [10] | Voxels | 40.9 | - |
| 3D ShapeNets [71] | Voxels | 49.2 | - |
| Densepoint[48] | Points | 88.5 | - |
| PVNet[75] | Points | 89.5 | - |
| MVCNN [61] | 12 Views | 80.2 | 73.5 |
| GIFT [2] | 20 Views | - | 64.0 |
| MVFusionNet [34] | 12 Views | - | 62.2 |
| ReVGG [60] | 20 Views | - | 74.9 |
| RotNet [38] | 20 Views | - | 77.2 |
| ViewGCN [67] | 20 Views | - | 78.4 |
| MLVCNN [35] | 24 Views | 92.2 | - |
| MVTN (ours) | 12 Views | **92.9** | **82.9** |