Counting Semantic Part Types of 3D Objects

Lucy Zhu, Panos Achlioptas, Kaichun Mo, Srinath Sridhar, Leonidas J. Guibas

Motivation

- Learn to **distinguish** the individual parts of 3D object point-clouds
- 3D models often have distinct parts **separable** in 3D modeling programs
- Same cannot be said for objects in the real world such as models obtained via 3D scanners
  - Manually model each piece
- Applications to computer vision and **3D Q&A**

swivel chair

?

1 back seat

2 back connectors

1 seat

5 legs

5 wheels

Challenges

- Ambiguity in structure of parts
- Addressing relatively rare parts
- Class imbalance
- Overfitting

Approach Overview

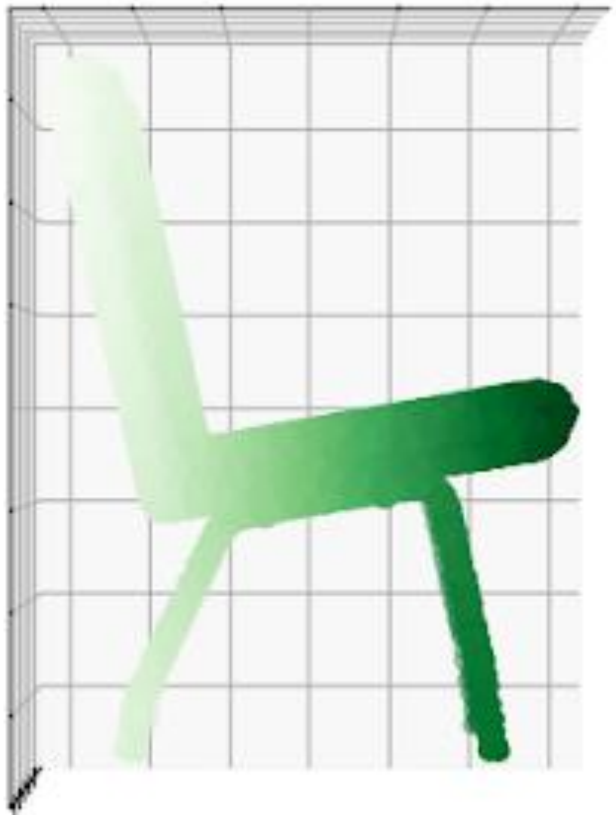
1. Prepare supervised dataset
2. Deploy 2 distinct Deep Neural Networks
   1. Specializing in *single* part type
   2. Generalizing across *multiple* parts
3. Train on different parts of 3D object point-clouds

Dataset

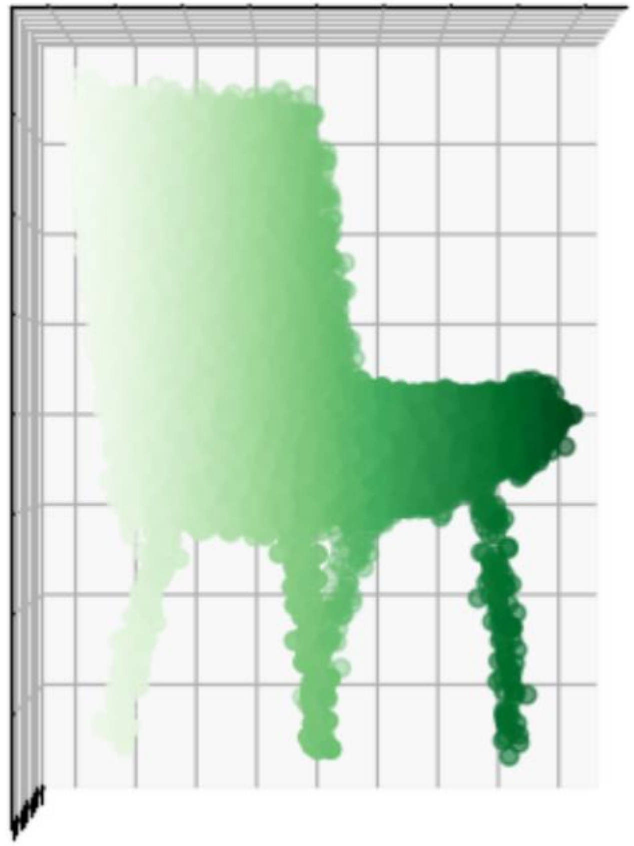
- 3D Object Point Clouds (**ShapeNet**) with labeled parts (**PartNet**)
- 2048 unique chair point clouds (split equally between testing and training dataset)
  - Chairs selected for how distinct their parts are compared to other objects
    - Avoid relatively small individual parts (windows in planes)
    - Avoid homogeneous appearance like mugs (number of parts must vary)

Point Cloud Model

- Randomly select 2,500/10,000 points for input
- Data augmentation
  - Invariance
  - Increase dataset size



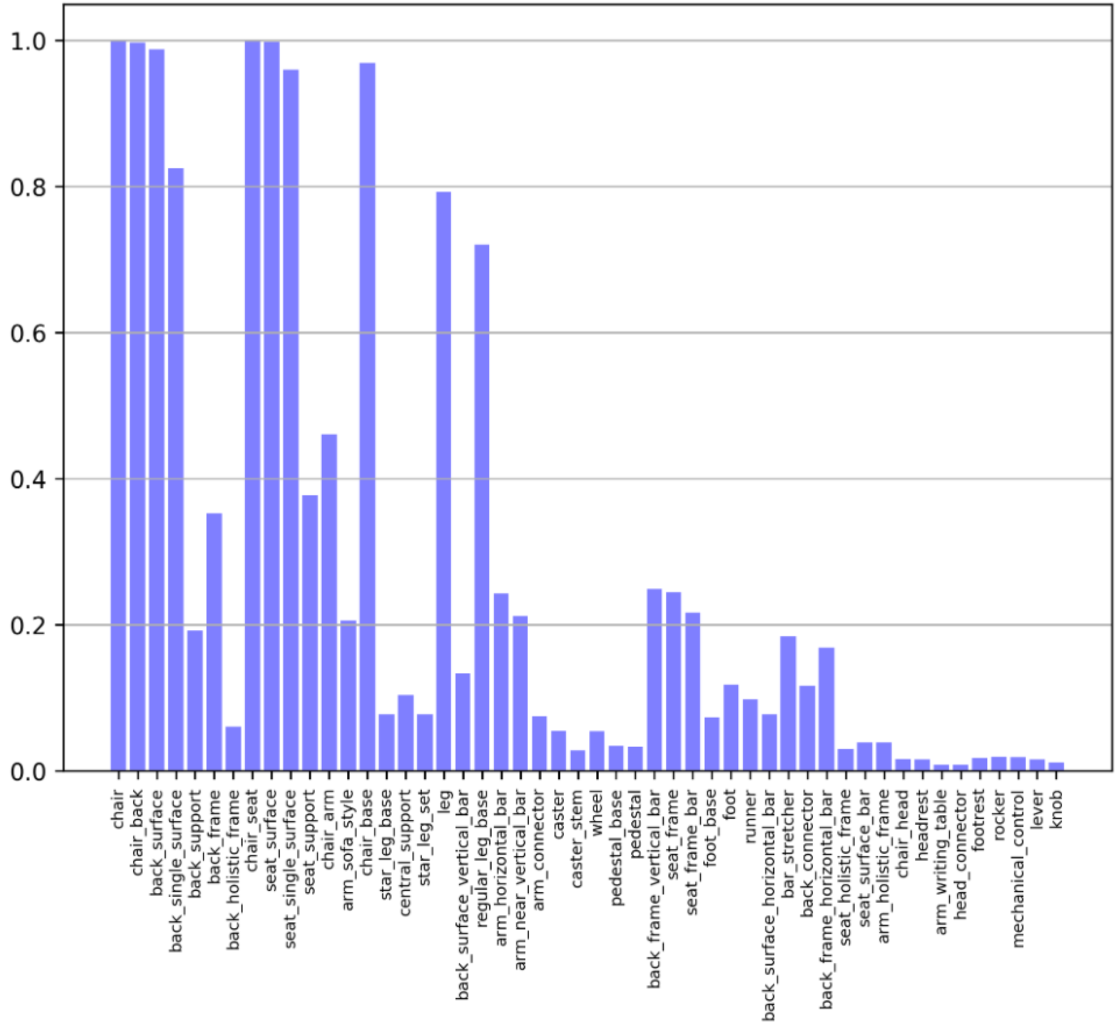
Before data augmentation.



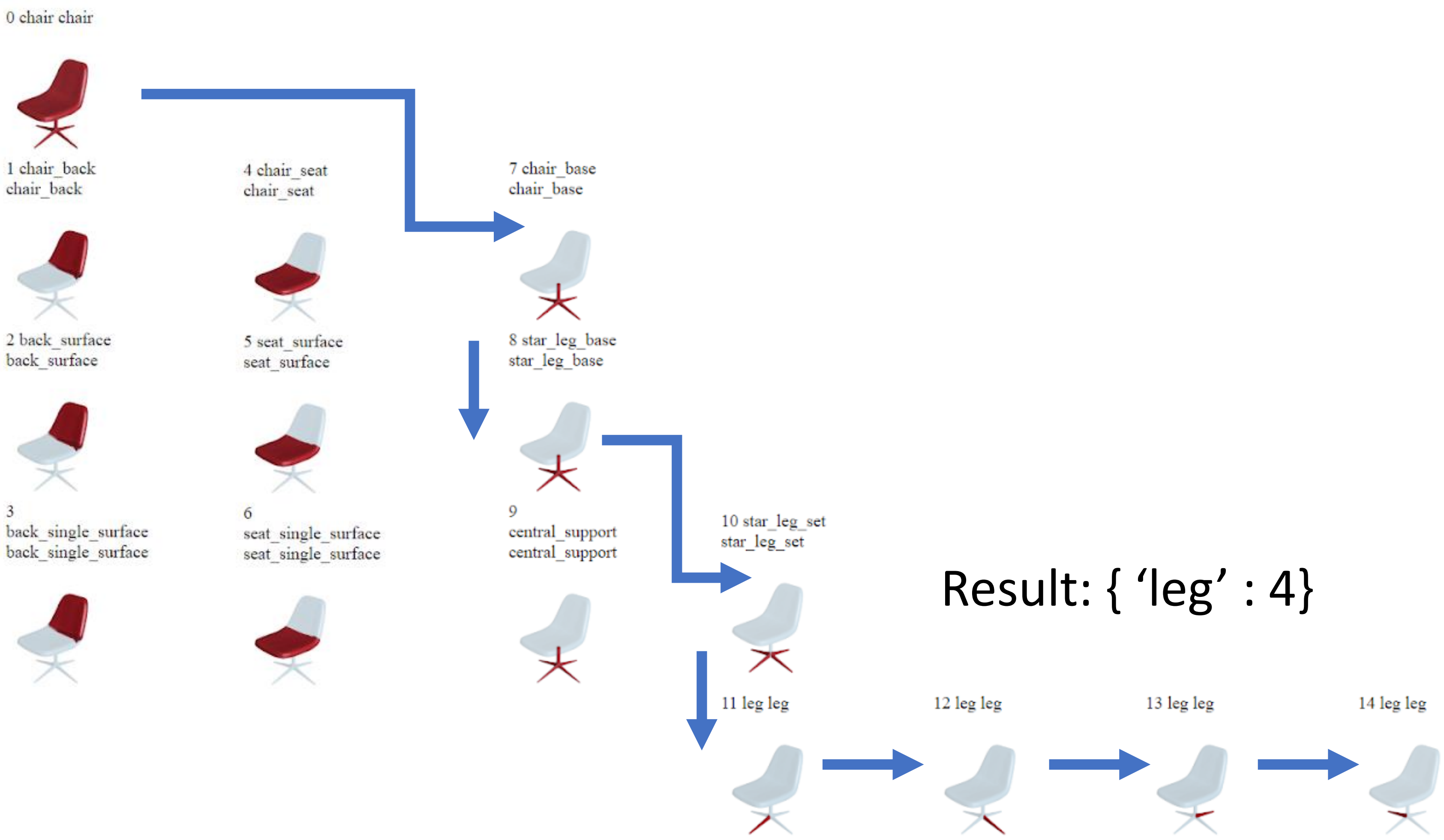
Jitter and rotation applied.

Selection of Parts to Analyze

- Avoid:
  - Rare parts (< 5% occurrences)
  - Parts occurring in ALL objects
  - Structurally similar/synonymous parts
    - Ex. Redundant to analyze *both* back_seat and back_seat_surface

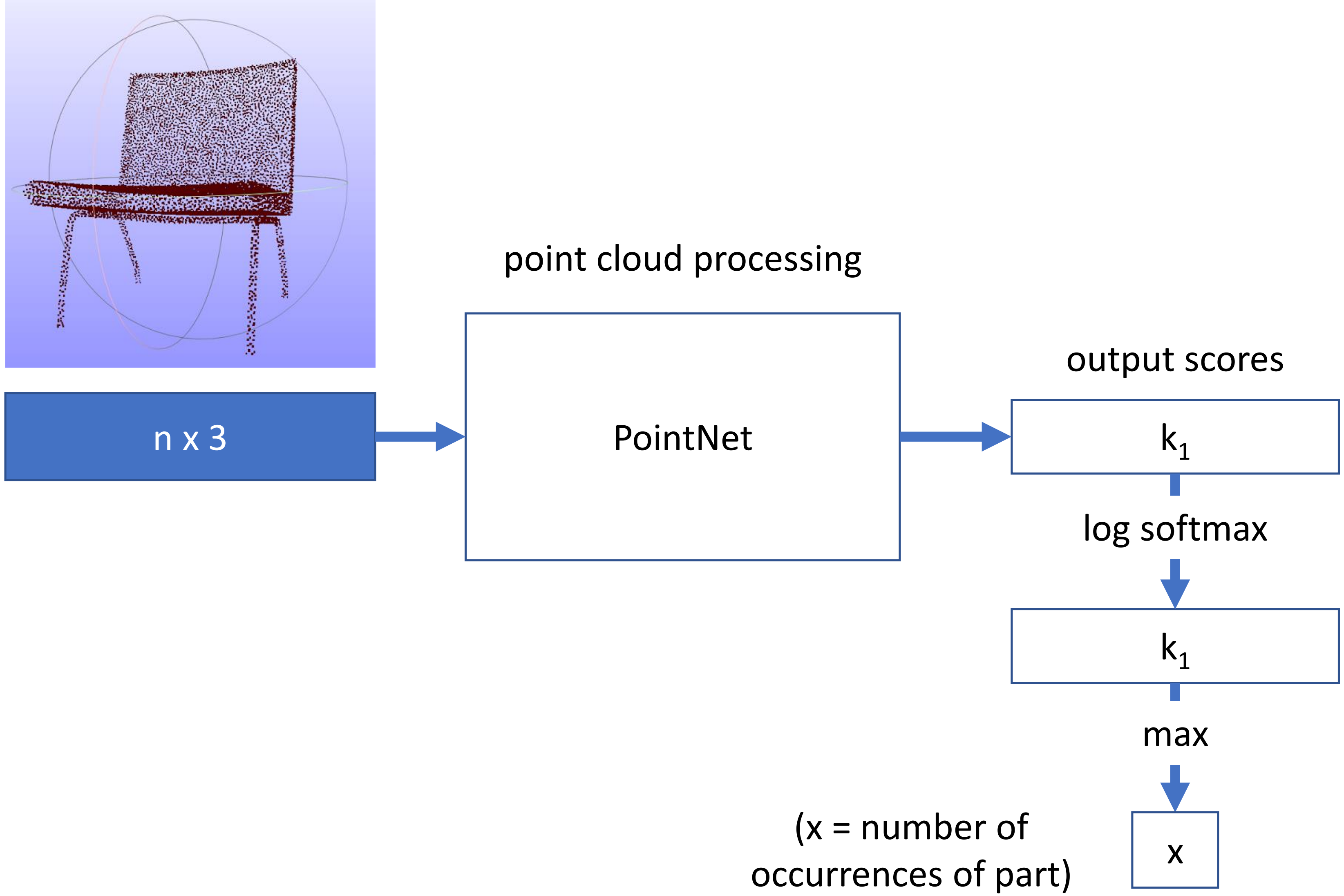


PartNet Label Hierarchy

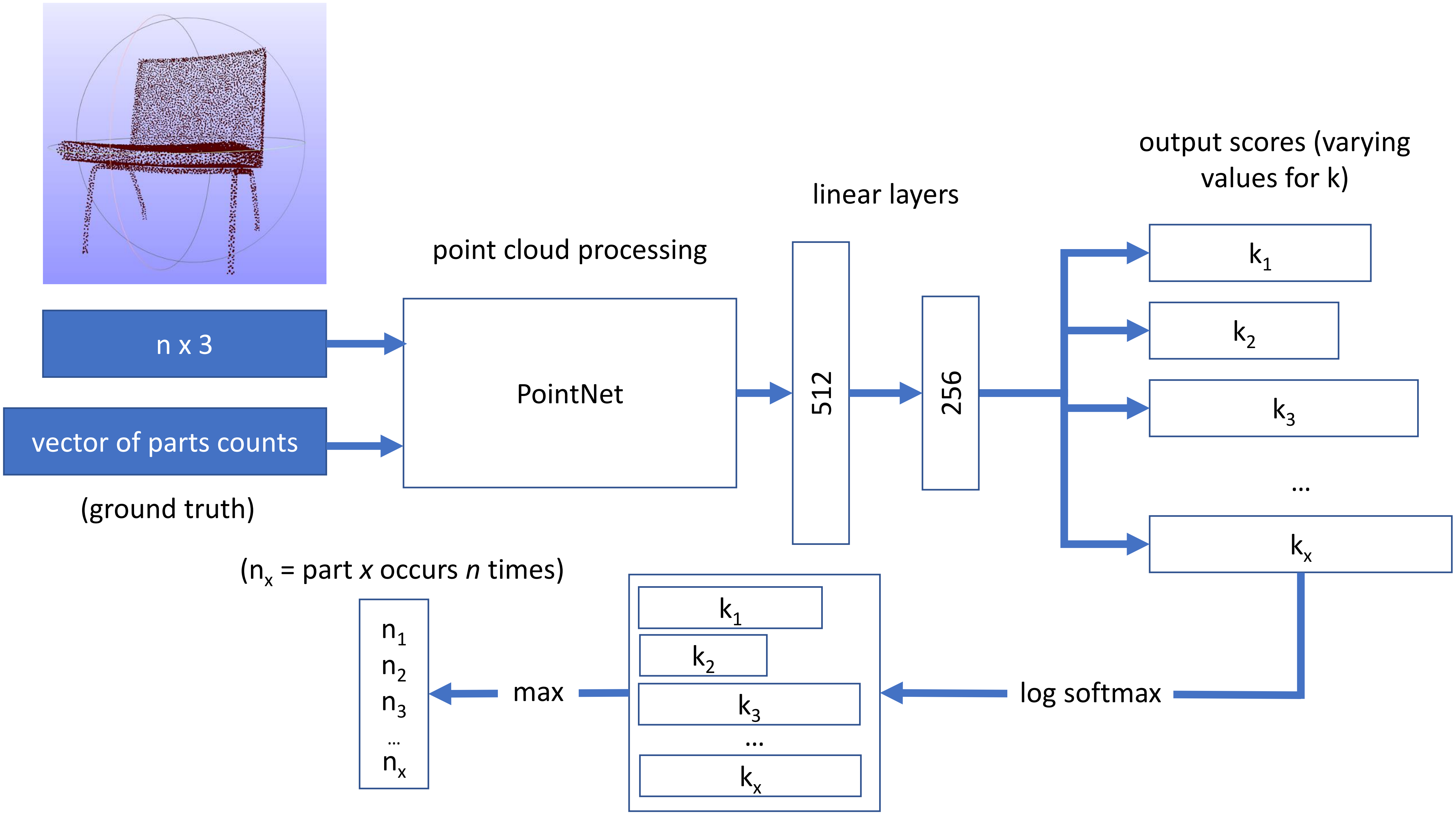


Result: { 'leg' : 4}

Specialized (specific part)



General (multiple parts)

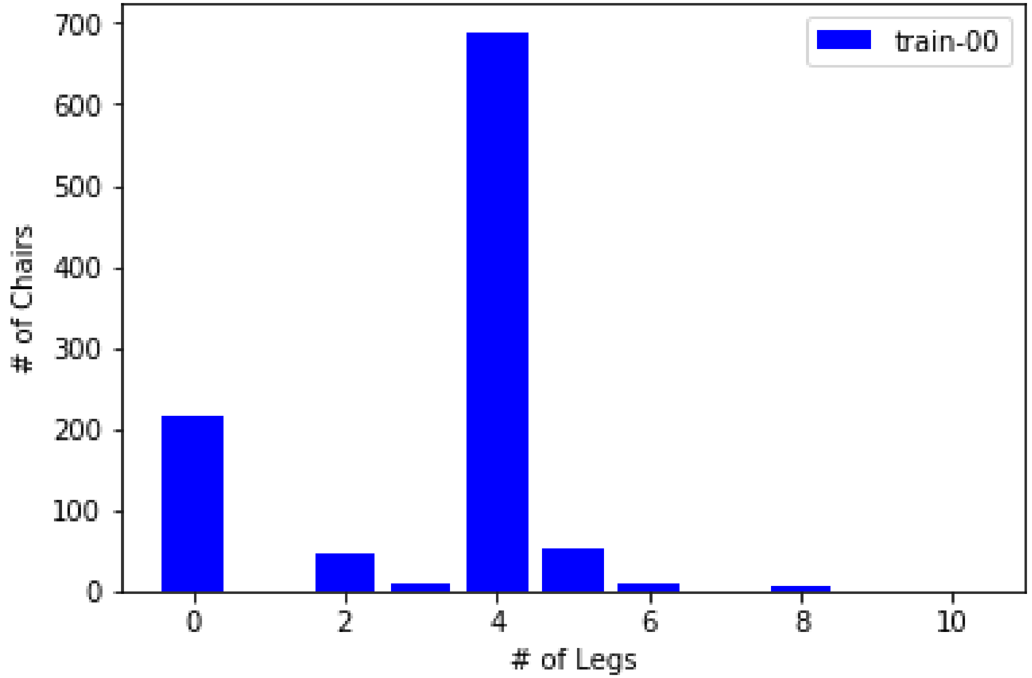


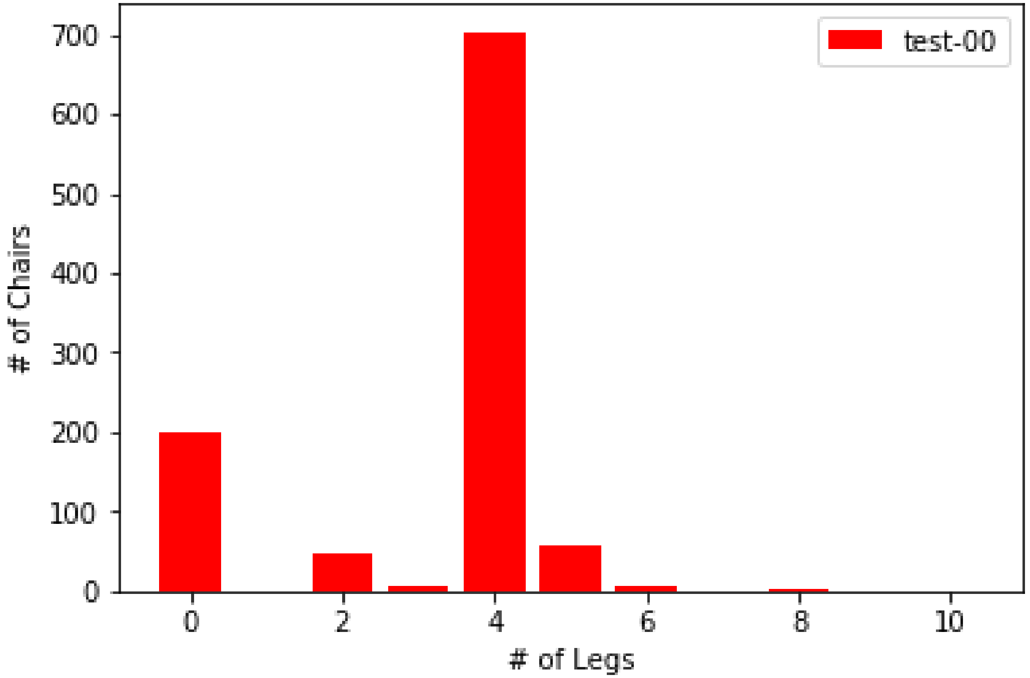
Experiment Results

- Trained and tested on multiple parts with varying resulting accuracies
- Specialized vs. General Deep Neural Networks

Experiment Results (Specialized – Legs)

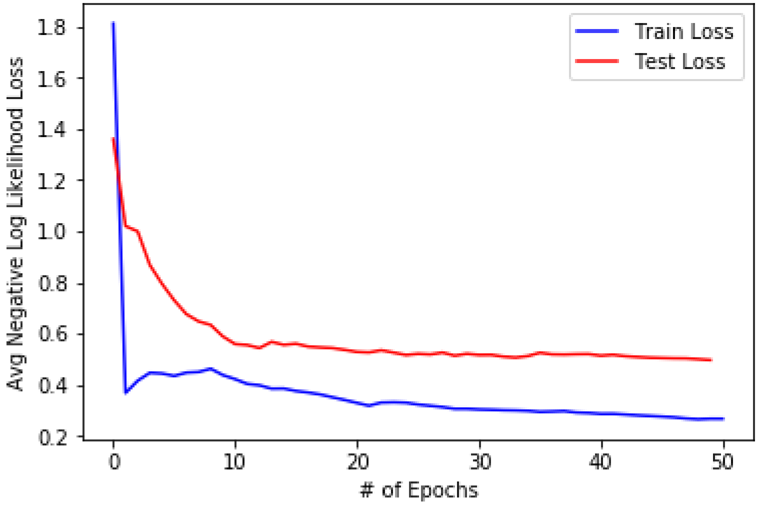
- Address class imbalance
  - 11 classes are reduced to 6 classes (0 to 5 legs) for the purpose of experiment
  - Skewed towards 4 legs (and somewhat towards 0 legs)

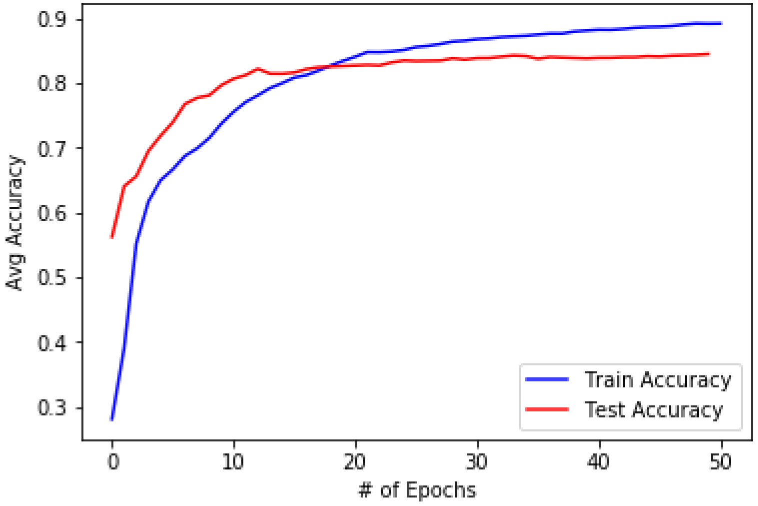




Experiment Results (Specialized – Legs)

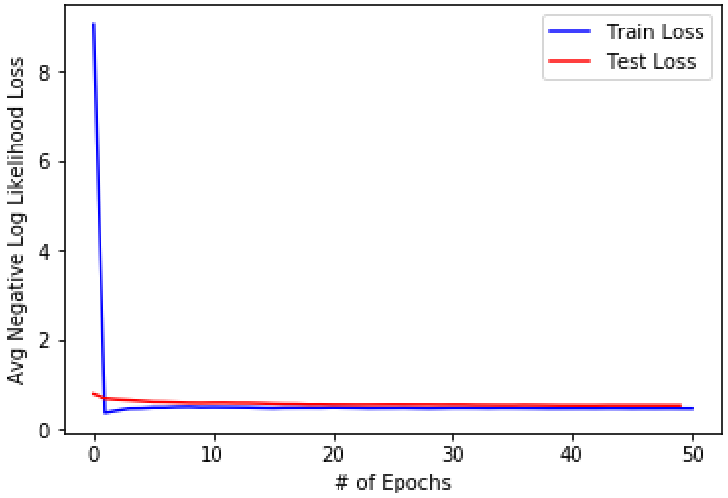
- # of Epochs: 50
- Learning Rate: 0.001
  - Annealing (learning rate * 0.9 for every 16 iterations)
- Batch Size: 32 (32 batches in each epoch)
- Final Accuracy: 0.859375

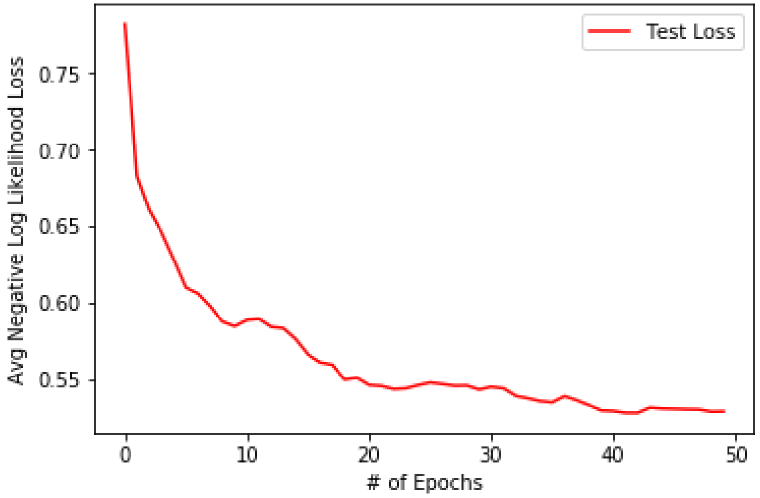


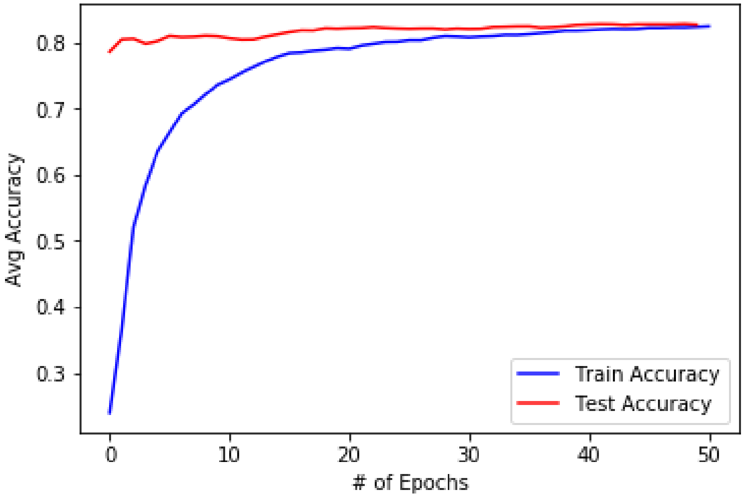


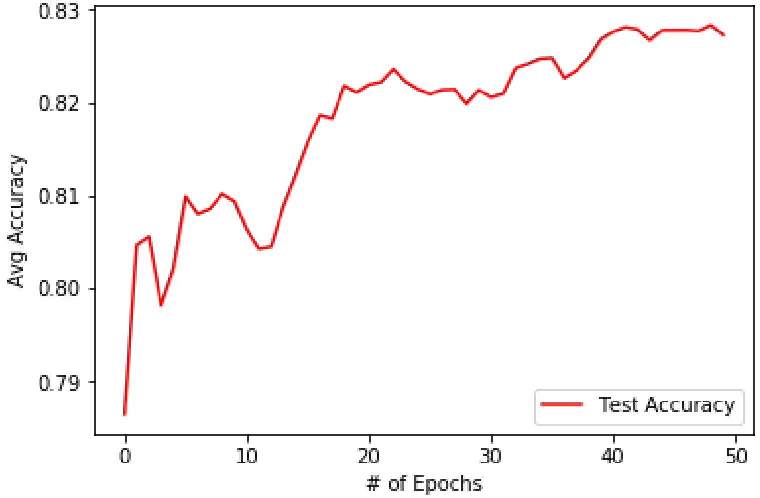
Experiment Results (General)

- # of Epochs: 50
- Learning Rate: 0.001
- batchSize: 32 (32 batches in each epoch)
- Final Accuracy: 0.8310546875



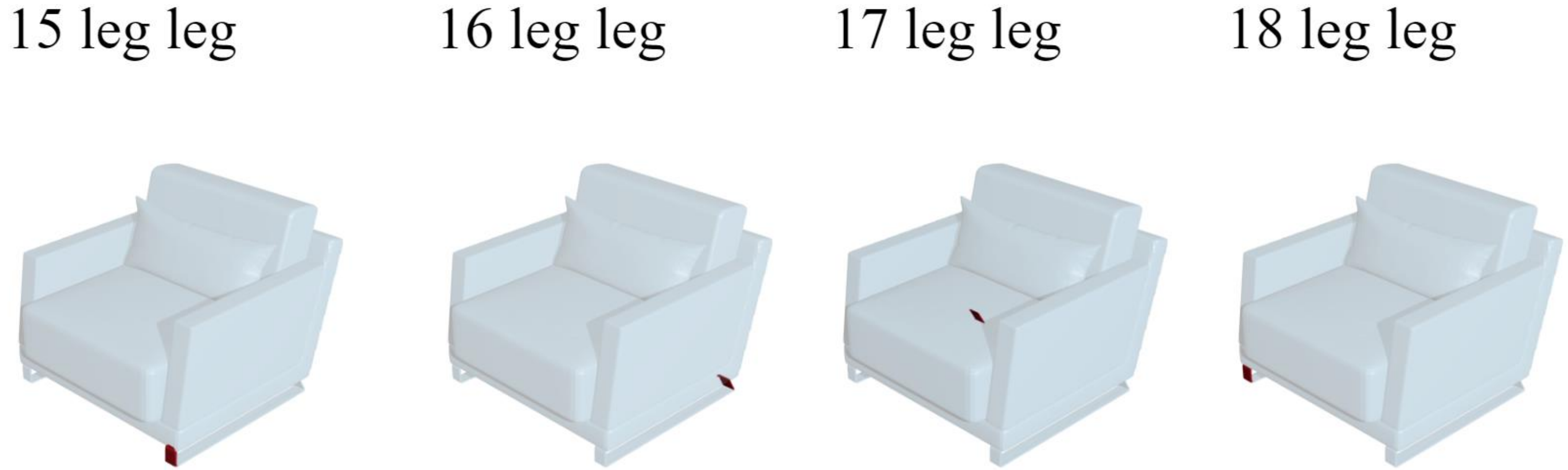






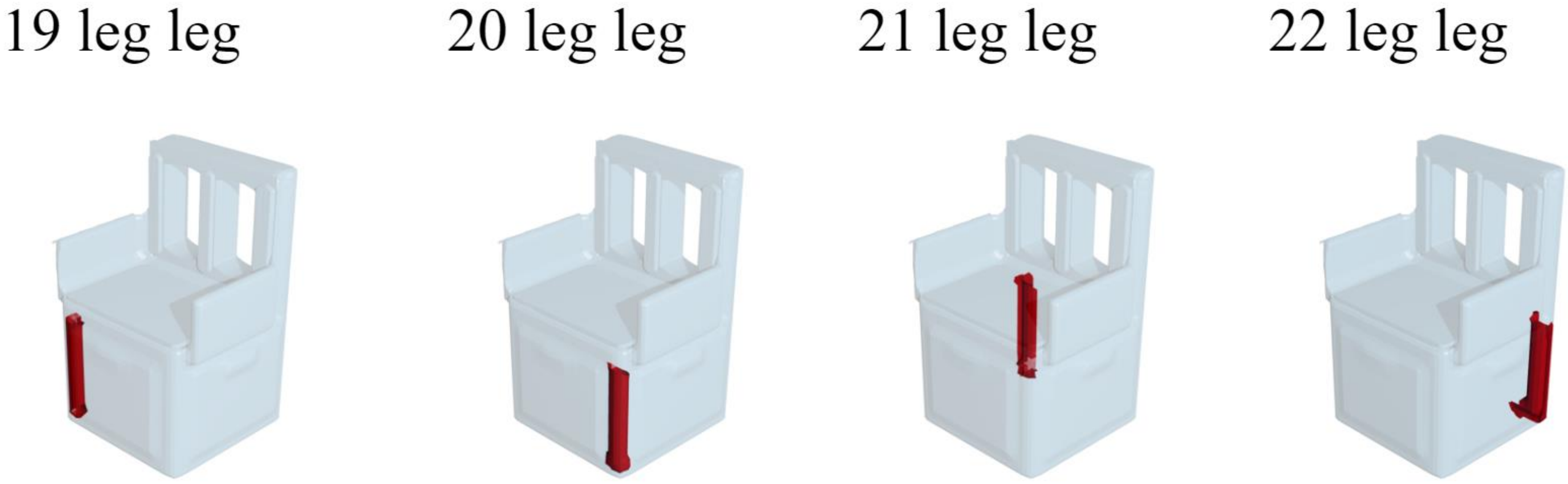
Experiment Results (Failure Cases)

- Legs small in comparison to overall shape
  - (pred: 0 target: 4)



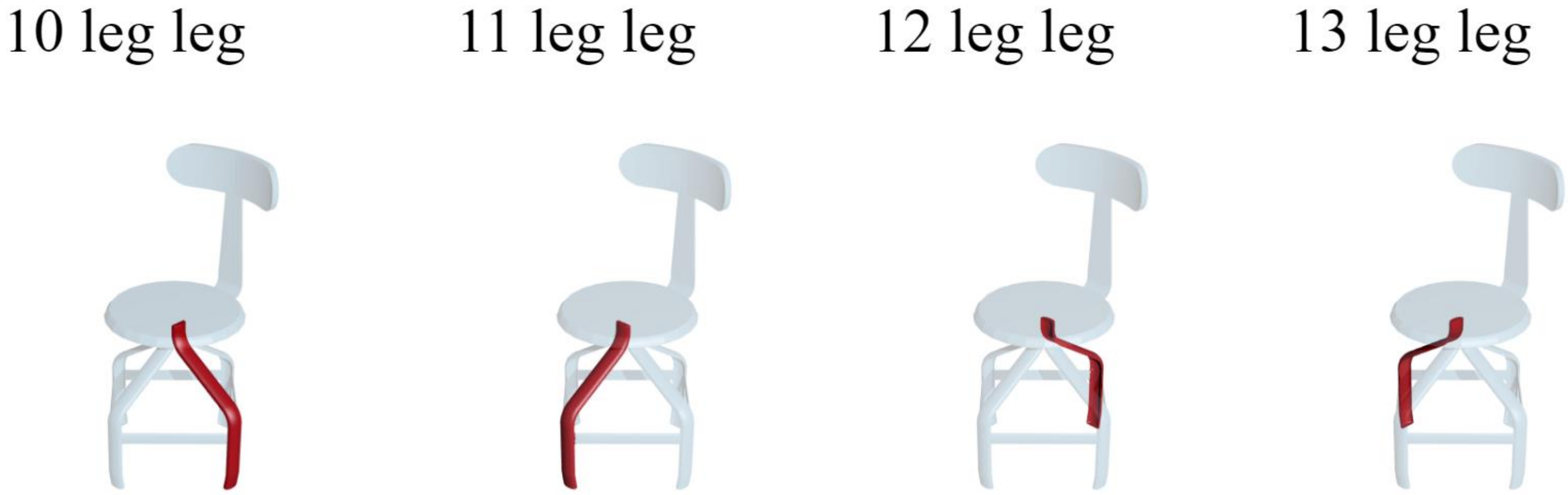
Experiment Results (Failure Cases)

- Legs are indistinct
  - (pred: 0 target: 4)



Experiment Results (Failure Cases)

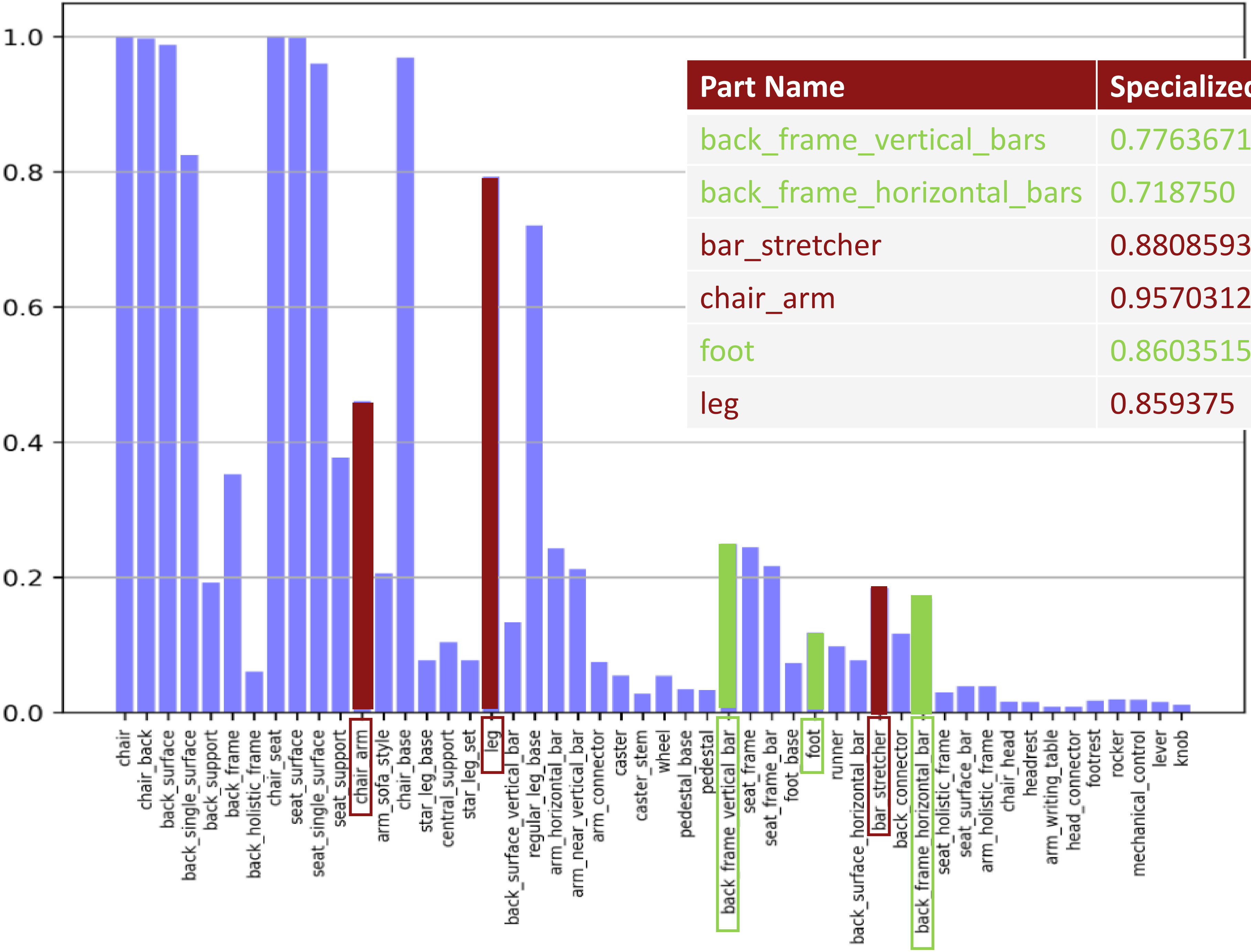
- Complex leg model
  - (pred: 5 target: 4)



Comparison of Models

Part Name	Specialized	General
back_frame_vertical_bars	0.7763671875	0.8173828125
back_frame_horizontal_bars	0.718750	0.7734375
bar_stretcher	0.880859375	0.8115234375
chair_arm	0.95703125	0.8994140625
foot	0.8603515625	0.8984375
leg	0.859375	0.7900390625

Varying changes in accuracies (some improved and some worsened despite having a similar model structure)



Part Name	Specialized	General
back_frame_vertical_bars	0.7763671875	0.8173828125
back_frame_horizontal_bars	0.718750	0.7734375
bar_stretcher	0.880859375	0.8115234375
chair_arm	0.95703125	0.8994140625
foot	0.8603515625	0.8984375
leg	0.859375	0.7900390625

Establishing correlations between different part structures potentially **enhances** the generalization capacity for *rarer* parts as evident in the general network.

Conclusion

- An approach that demonstrates the feasibility of solving the counting problem given proper part labels
- Specialized deep neural network not optimal when training dataset is sparse
  - Parts in objects vary in appearance
- Possible paths to develop further:
  - Point-cloud attention
  - Unsupervised methods that reduce the amount of training data needed

Any Questions?