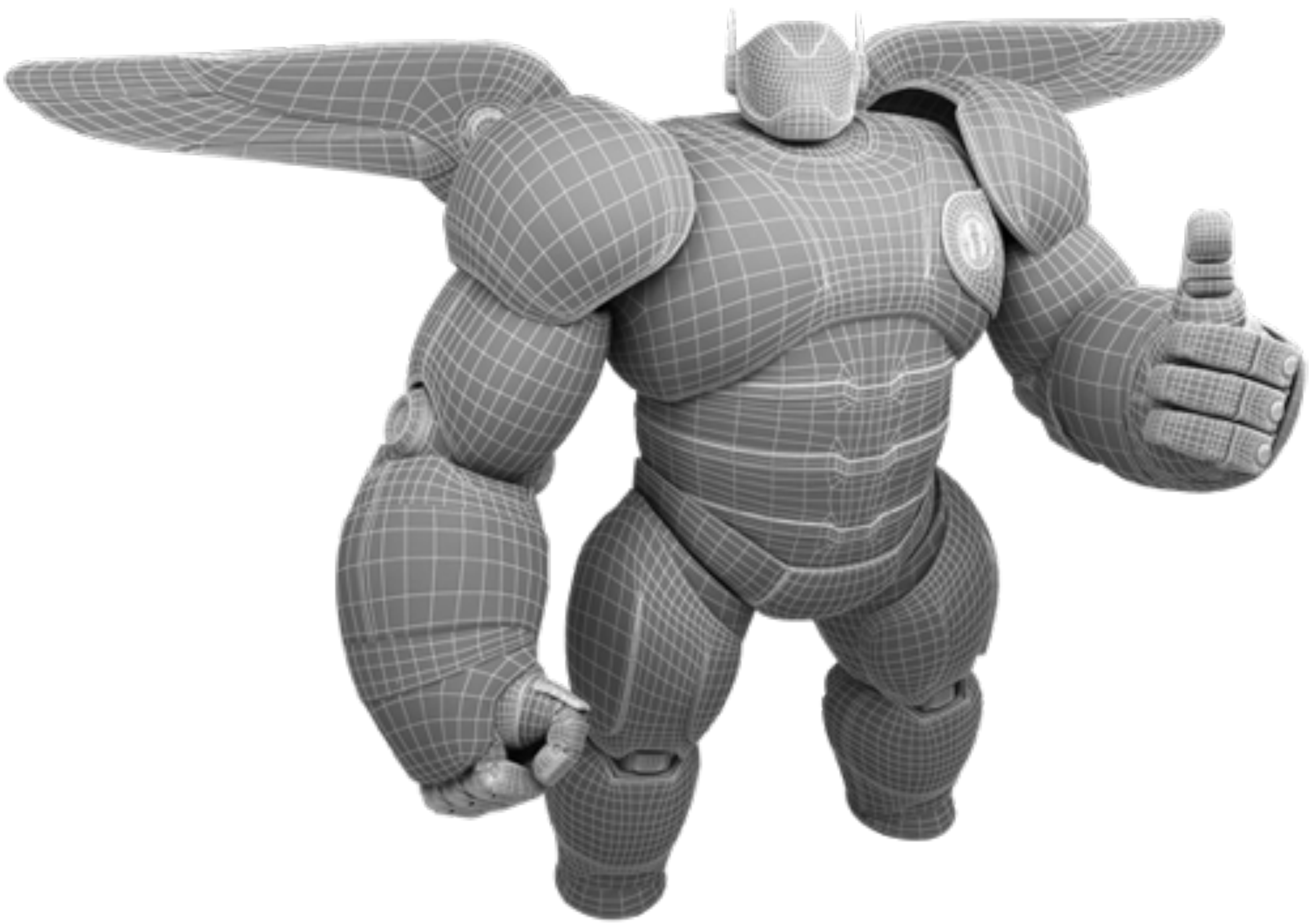
Data-driven Geometry Processing

Vladimir (Vova) Kim

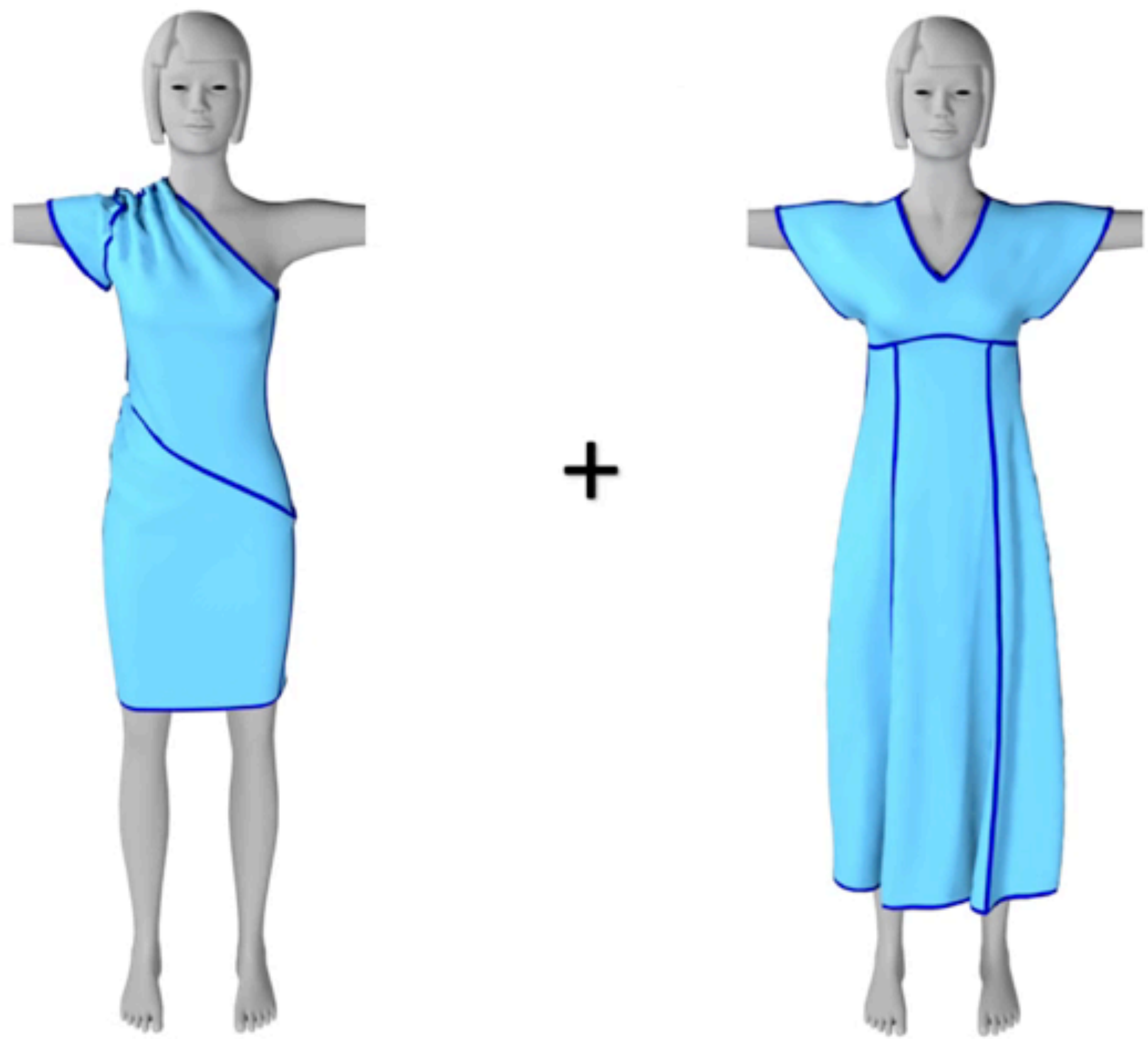
Adobe Research

Modeling 3D Assets



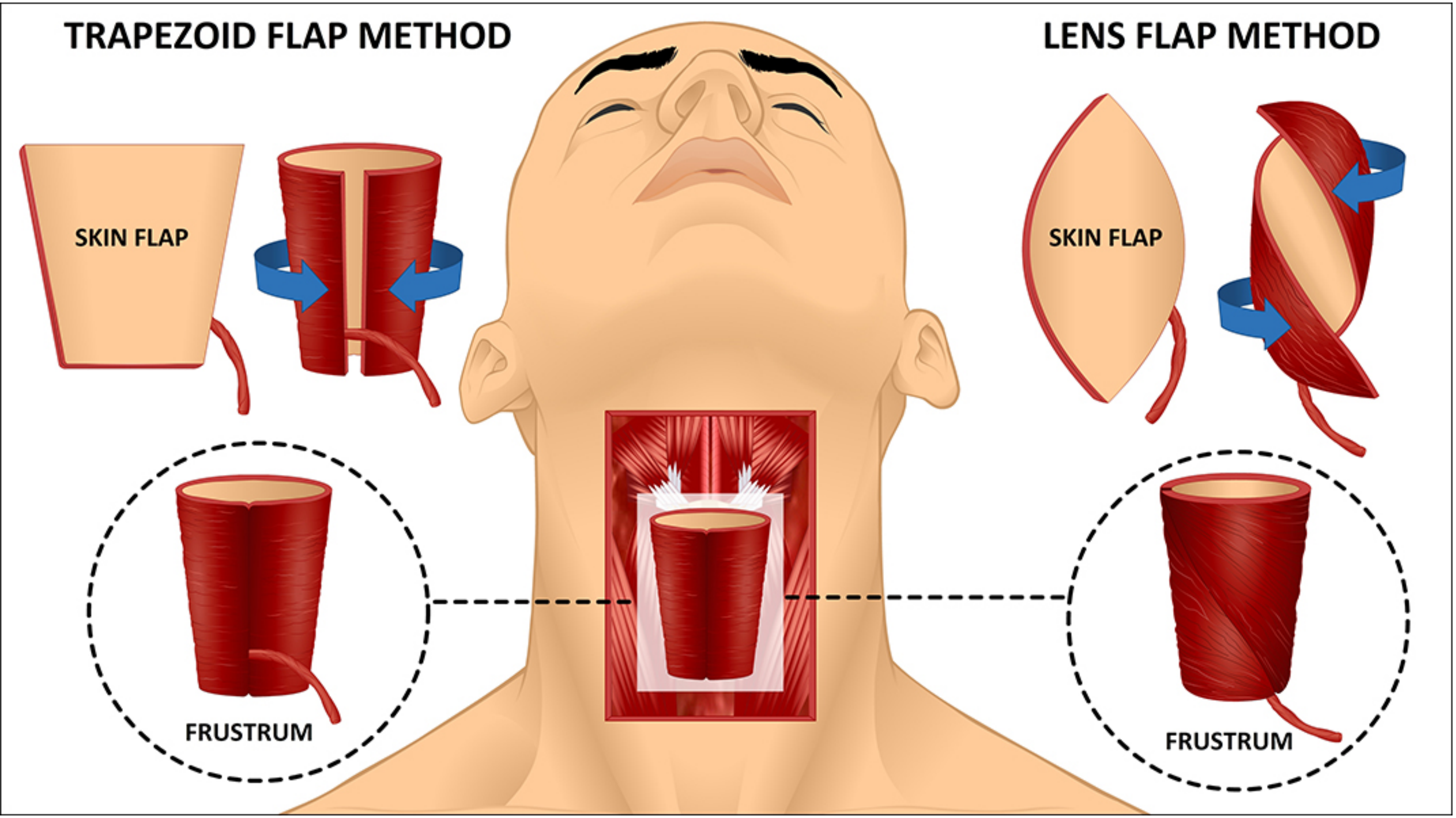
Modeling Garments

Direct
3D Edit

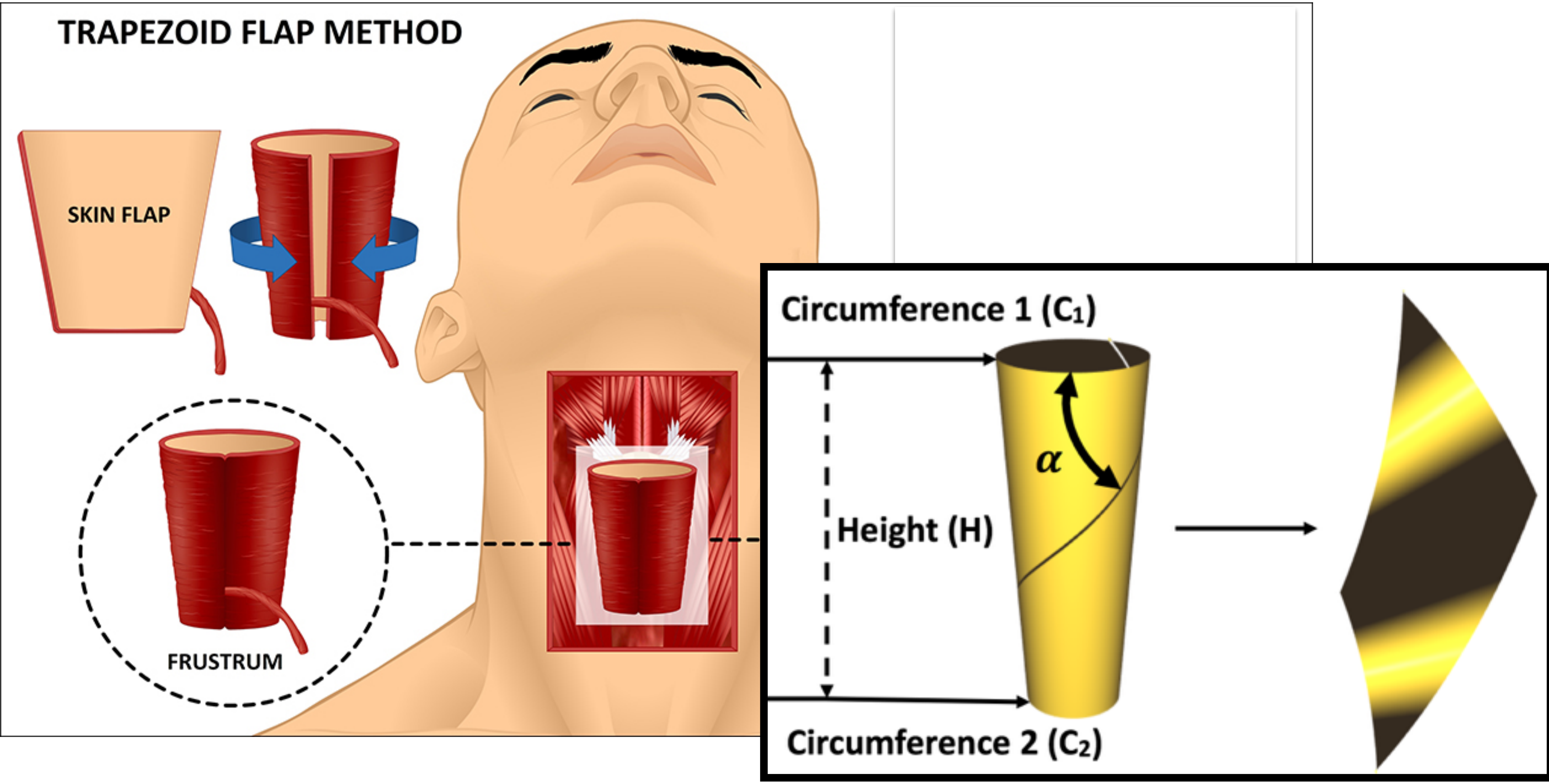


Physics-driven Pattern Adjustment for Direct 3D Garment Editing. Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny Kaufman, Nicholas Vining, Floraine Berthouzoz
SIGGRAPH 2016

Modeling Body Parts

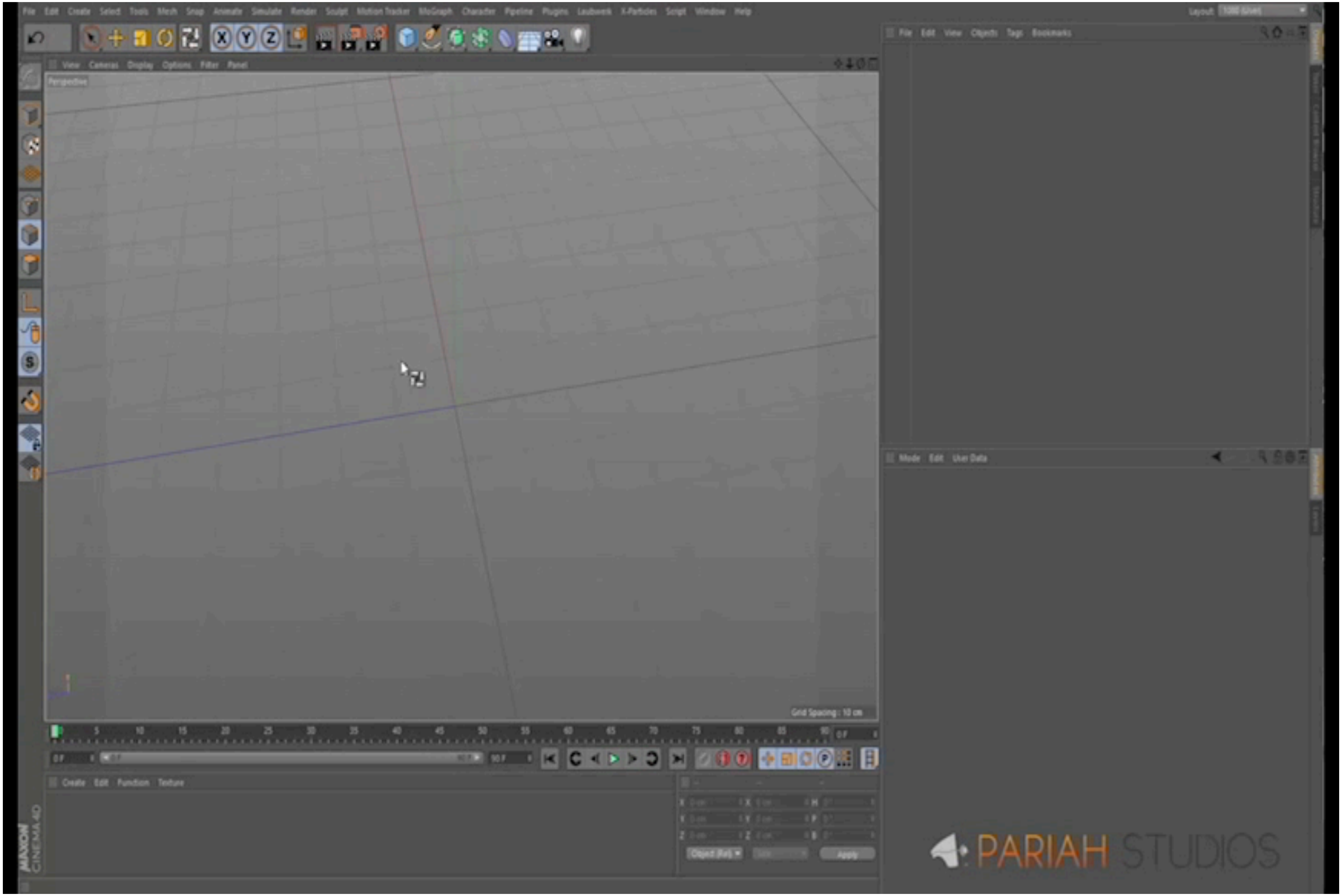


Modeling Body Parts



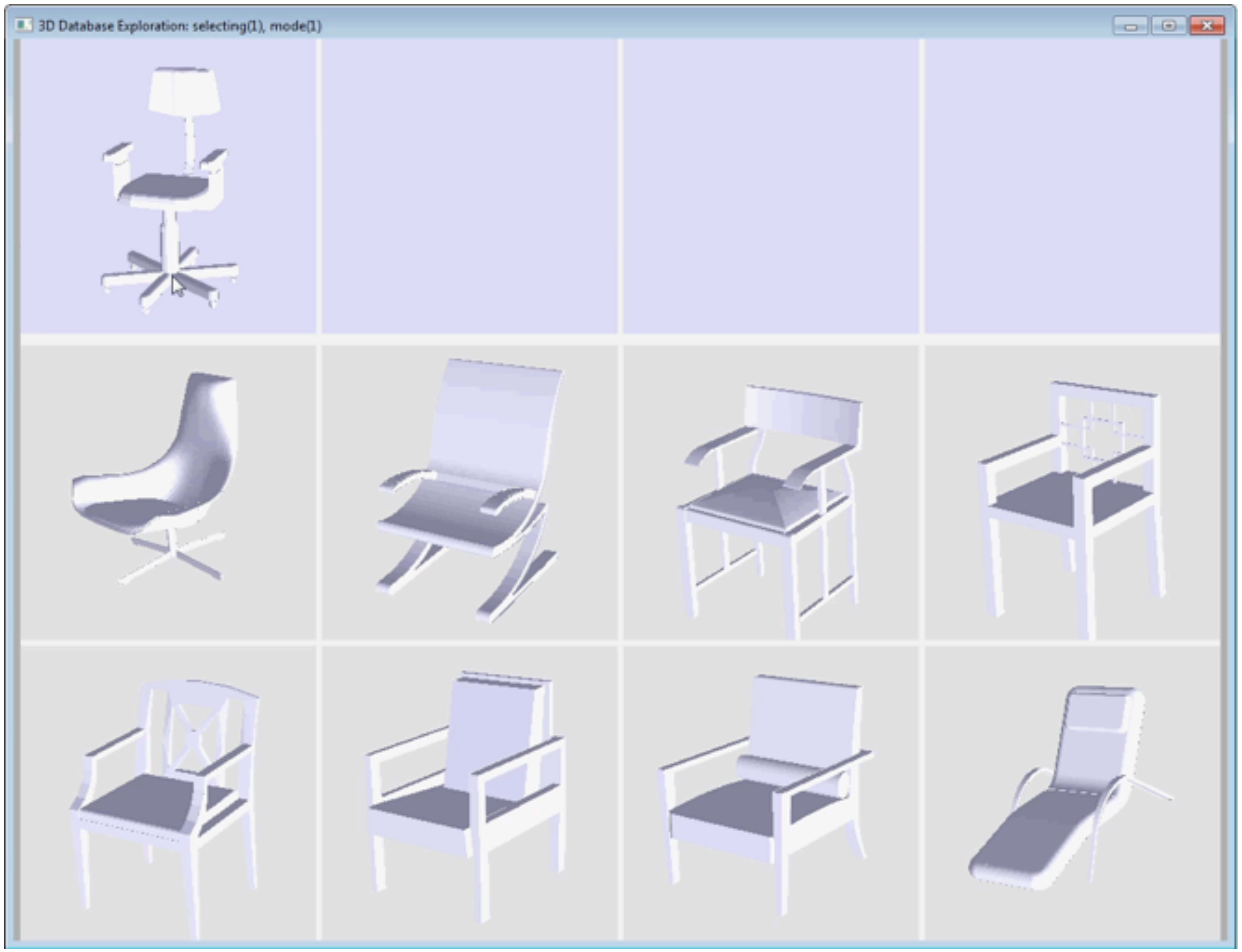
Customized Software to Optimize Circumferential Pharyngoesophageal Free Flap Reconstruction. Oleksandr Butskiy, Vladimir G. Kim, Brent Chang, Donald Anderson, and Eitan Prisman. Laryngoscope, 2017

Modeling is Difficult



Why Data-Driven?

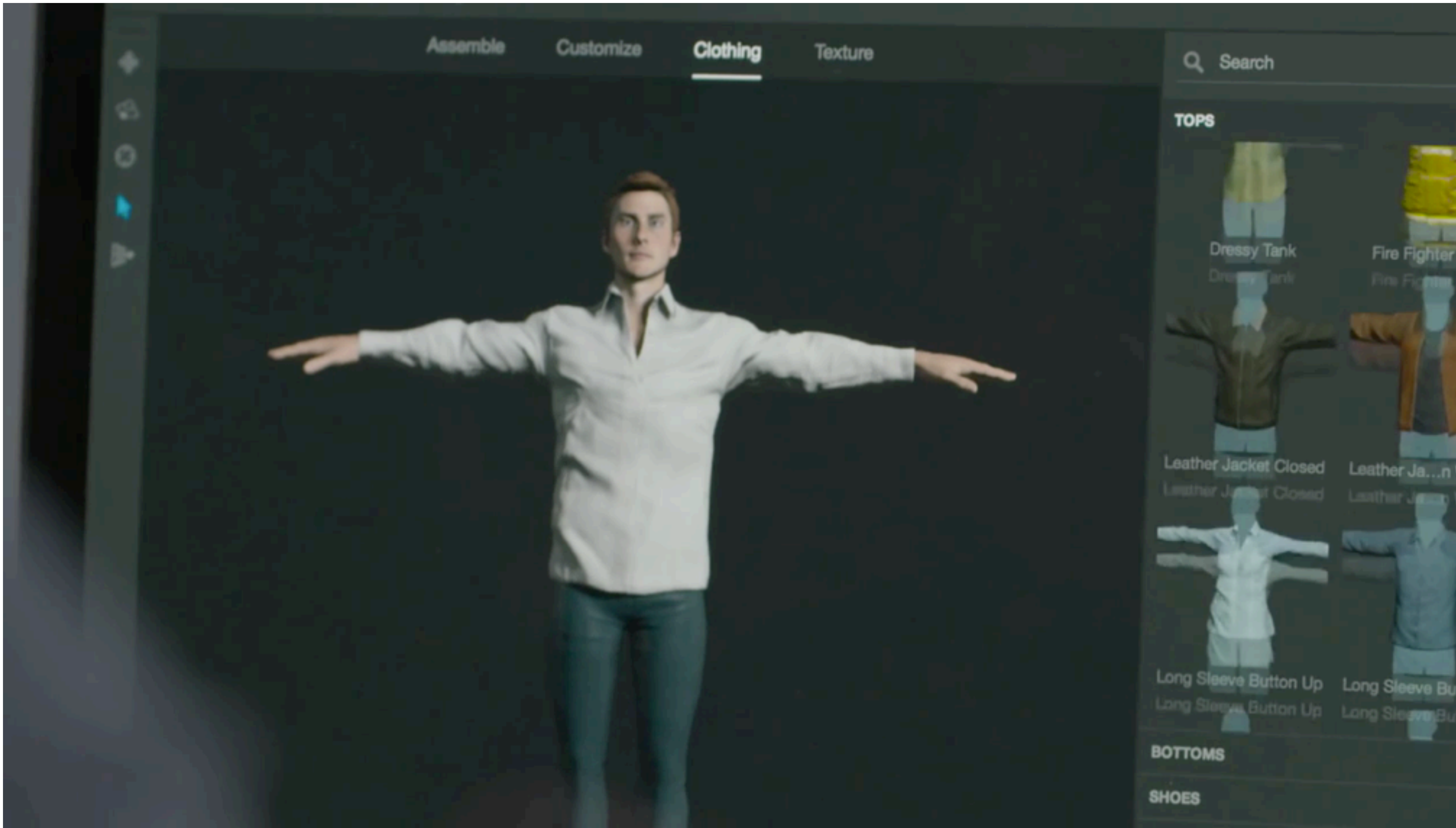
Find desired model in existing dataset



Exploring Collections of 3D Models using Fuzzy Correspondences.
V. Kim, W. Li, N. Mitra, S. Chaudhuri, S. DiVerdi, T. Funkhouser, SIGGRAPH 2012

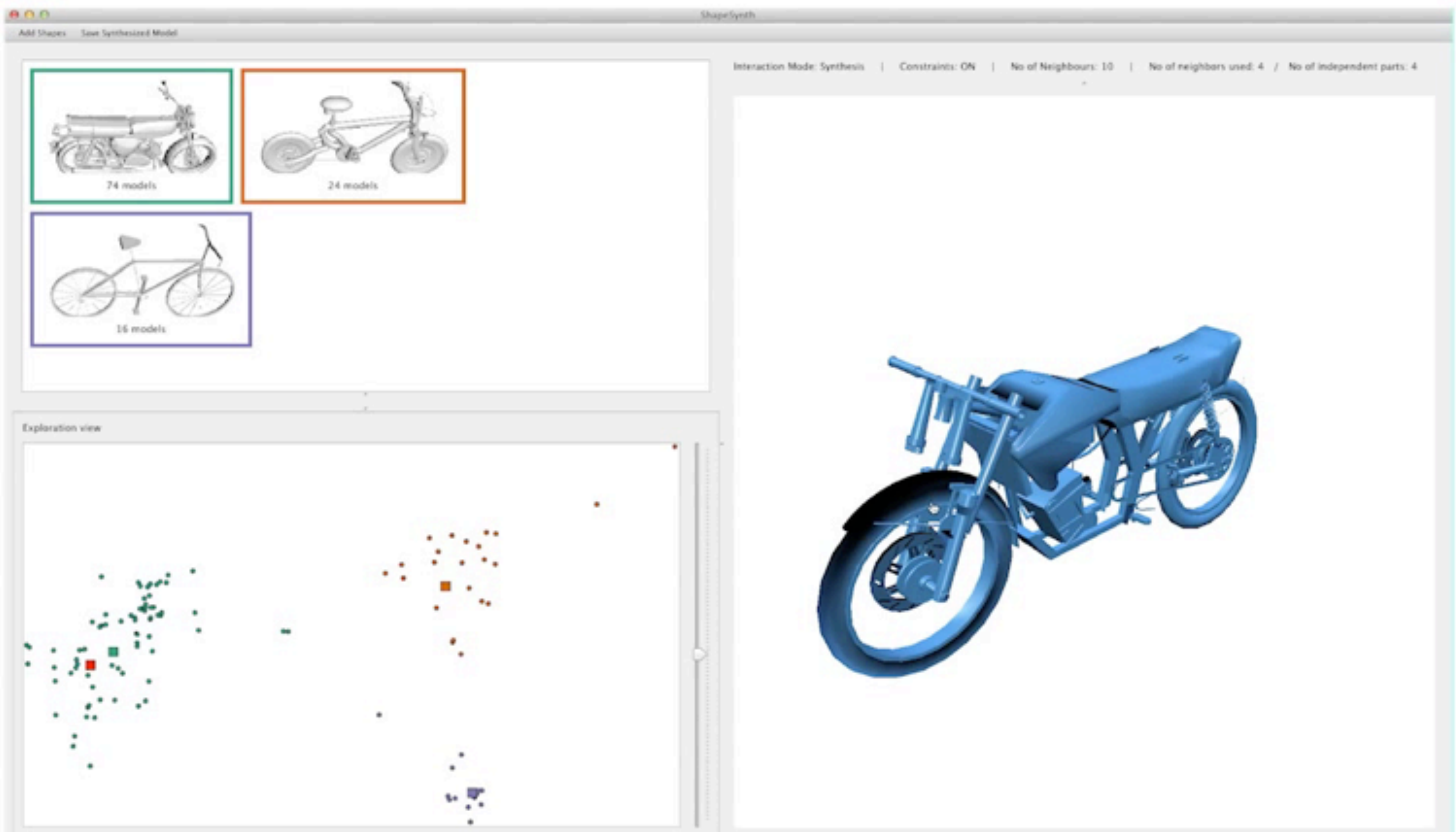
Why Data-Driven?

Create new model from existing parts



Why Data-Driven?

Understand context and variations



ShapeSynth: Parameterizing Model Collections for Coupled Shape Exploration and Synthesis. M. Averkiou, V. Kim, Y. Zheng, and N. Mitra, Eurographics 2014

What is the right "Data"

3D shapes

Categories

Parts

Part Labels

…

Machine Learning Pipeline

Training

Get Data

Label Data

Train Model

Testing

Apply Model

- Label input
- Sample from model
- …

Example: Shape Completion

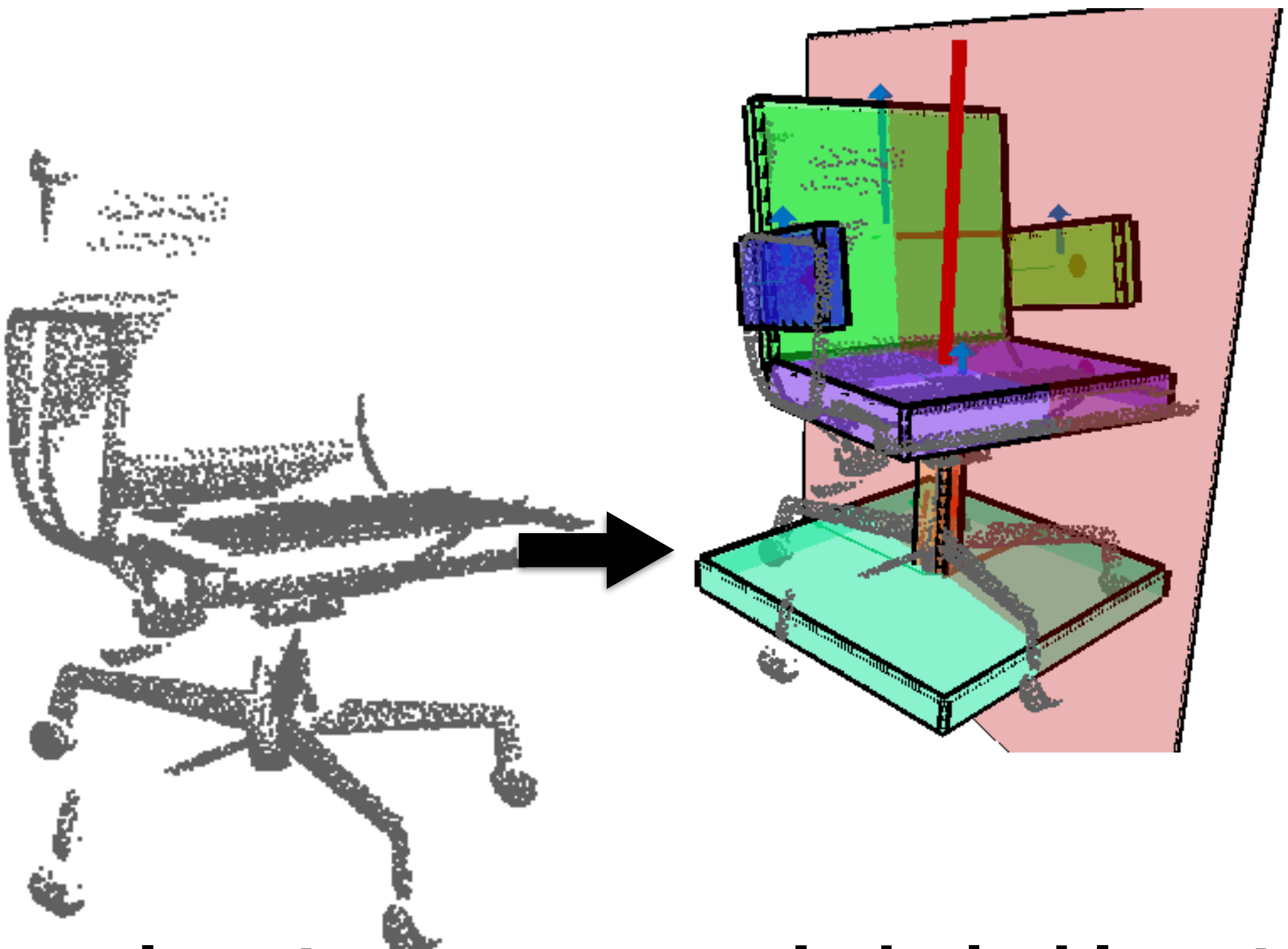
Use trained model to complete a partial scan

Input

Data-Driven Structural Priors for Shape Completion. Minhyuk Sung, Vladimir G. Kim, Roland Angst, and Leonidas Guibas SIGGRAPH Asia 2015

Example: Shape Completion

Use trained model to complete a partial scan

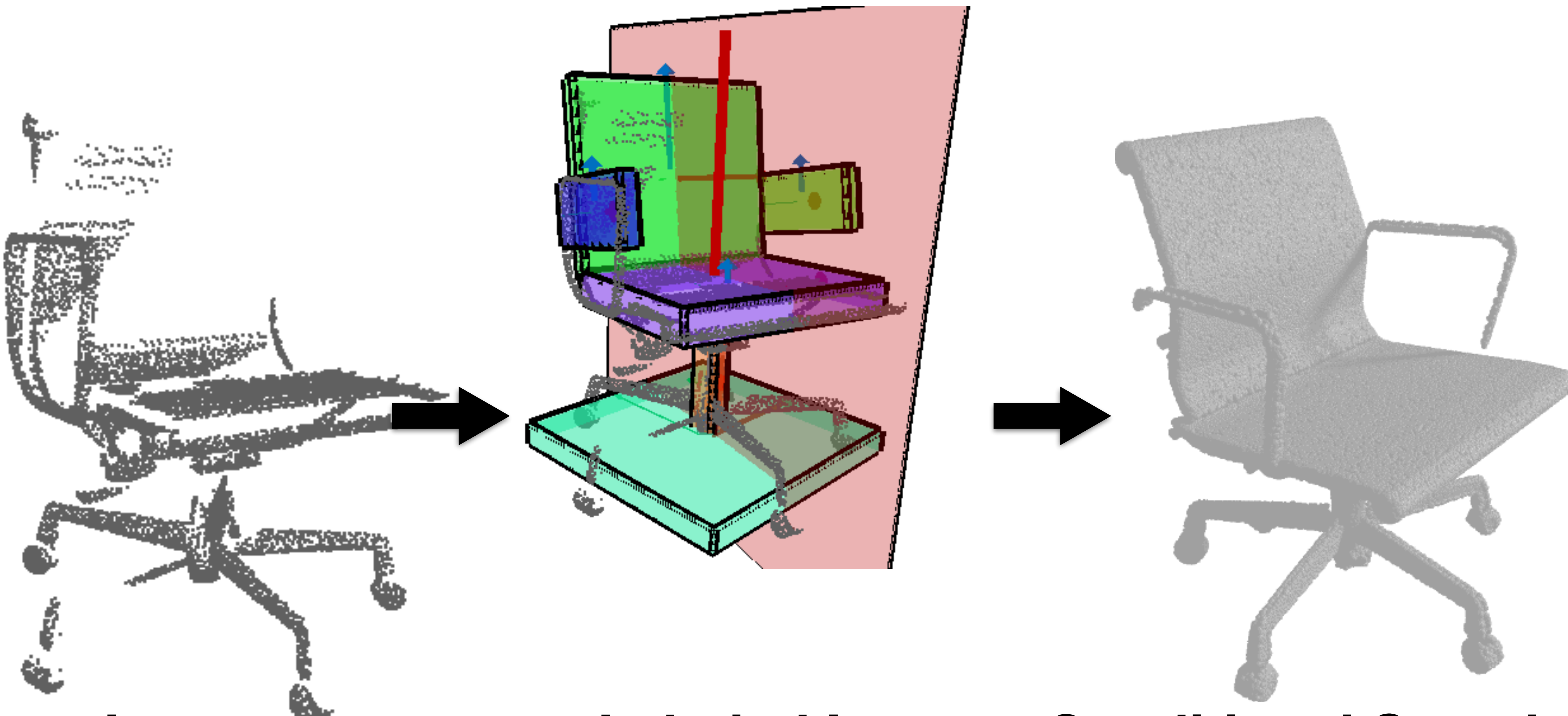


Input **Labeled Input**

Data-Driven Structural Priors for Shape Completion. Minhyuk Sung, Vladimir G. Kim, Roland Angst, and Leonidas Guibas SIGGRAPH Asia 2015

Example: Shape Completion

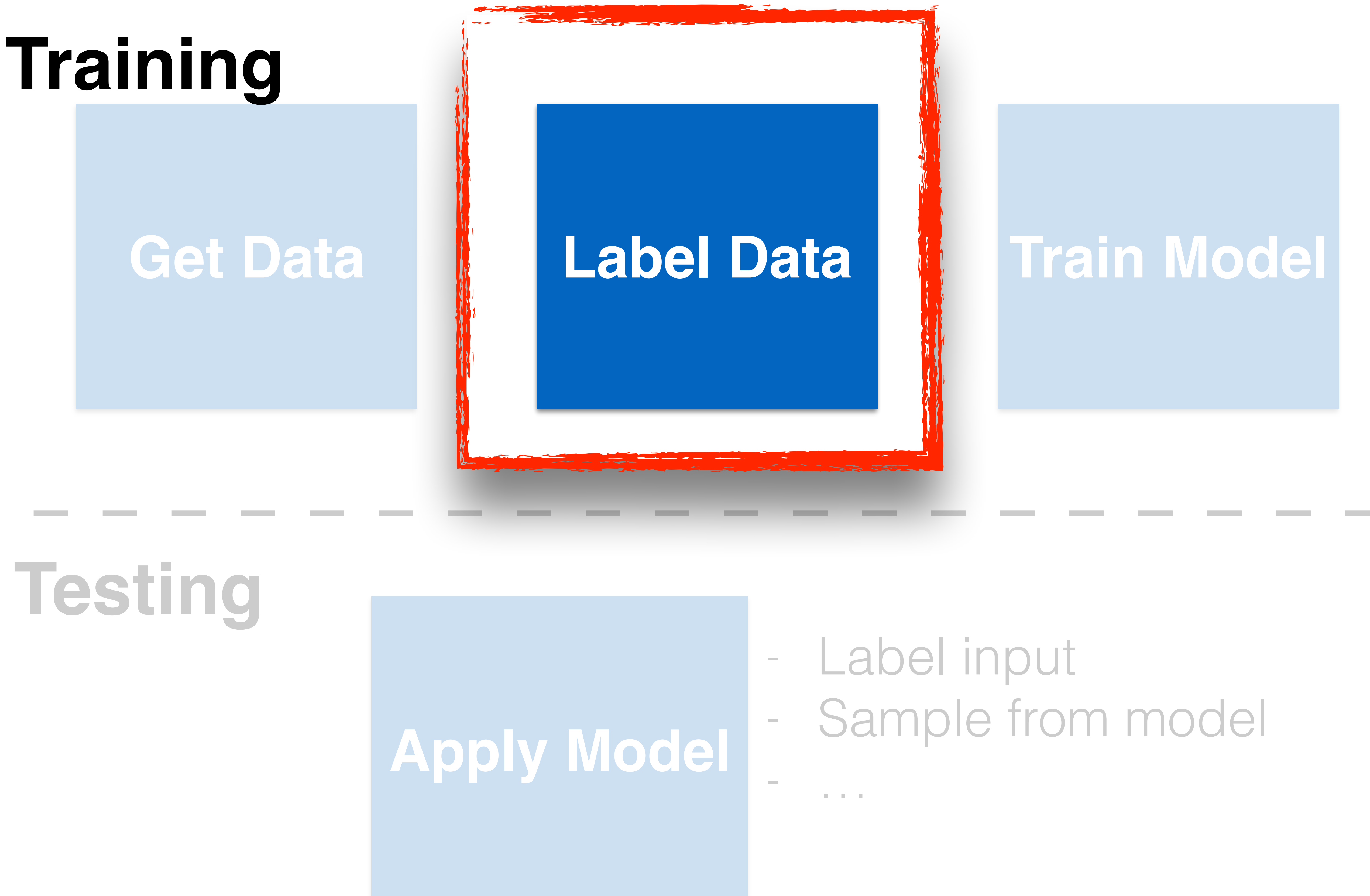
Use trained model to complete a partial scan



Input **Labeled Input** **Conditional Sample**

Data-Driven Structural Priors for Shape Completion. Minhyuk Sung, Vladimir G. Kim, Roland Angst, and Leonidas Guibas SIGGRAPH Asia 2015

Machine Learning Pipeline



Part Labeling

Somebody provided data for you

Part Labeling

Somebody provided data for you

Human labeling

- Experts (aka "graduate students")
- Crowd workers

Part Labeling

Somebody provided data for you

Human labeling

- Experts (aka “graduate students”)
- Crowd workers

ML algorithms

Part Labeling

Somebody provided data for you

Human labeling

- Experts (aka "graduate students")
- Crowd workers

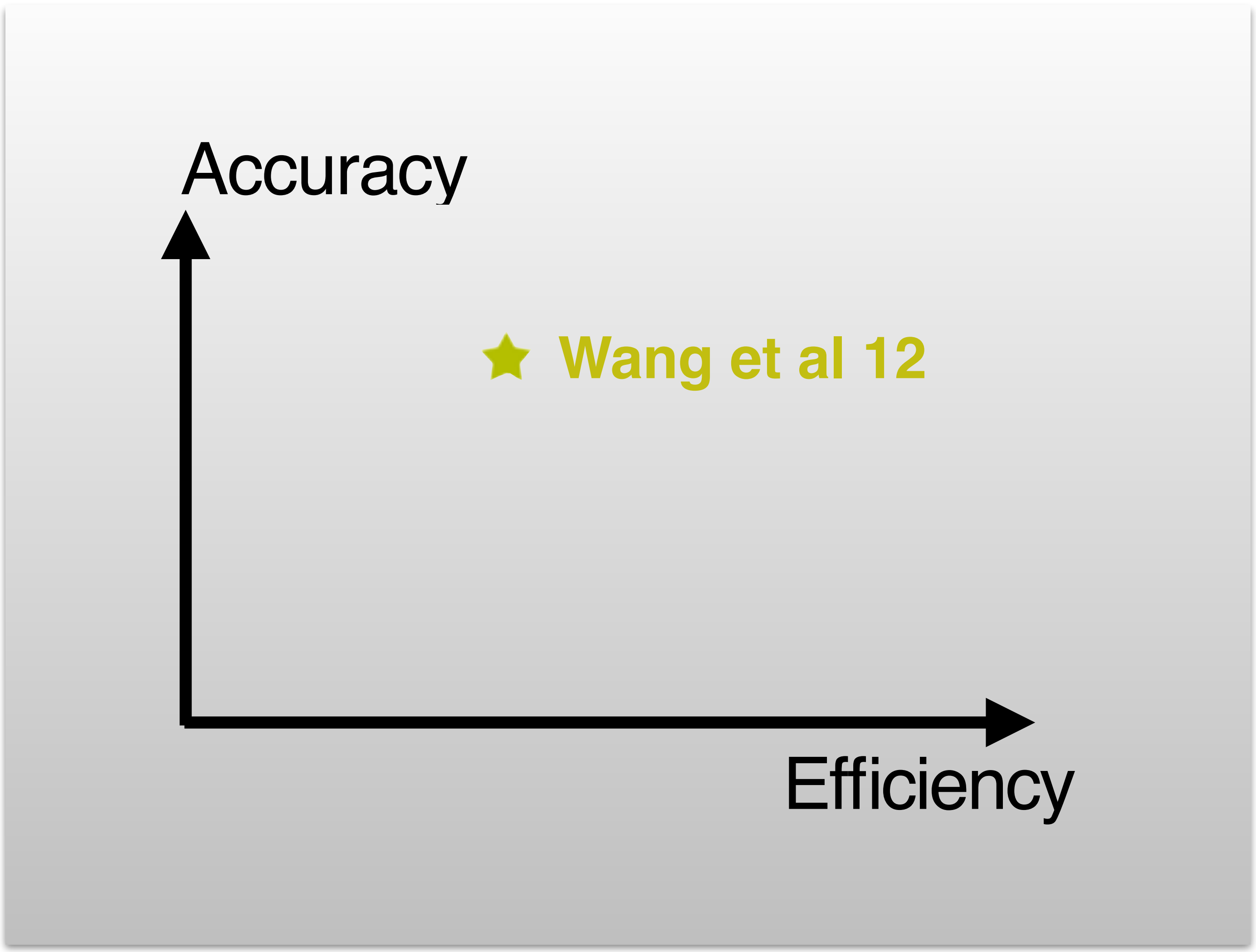
ML algorithms

Some combination of above

Training Data

Where do we get segmented shapes?

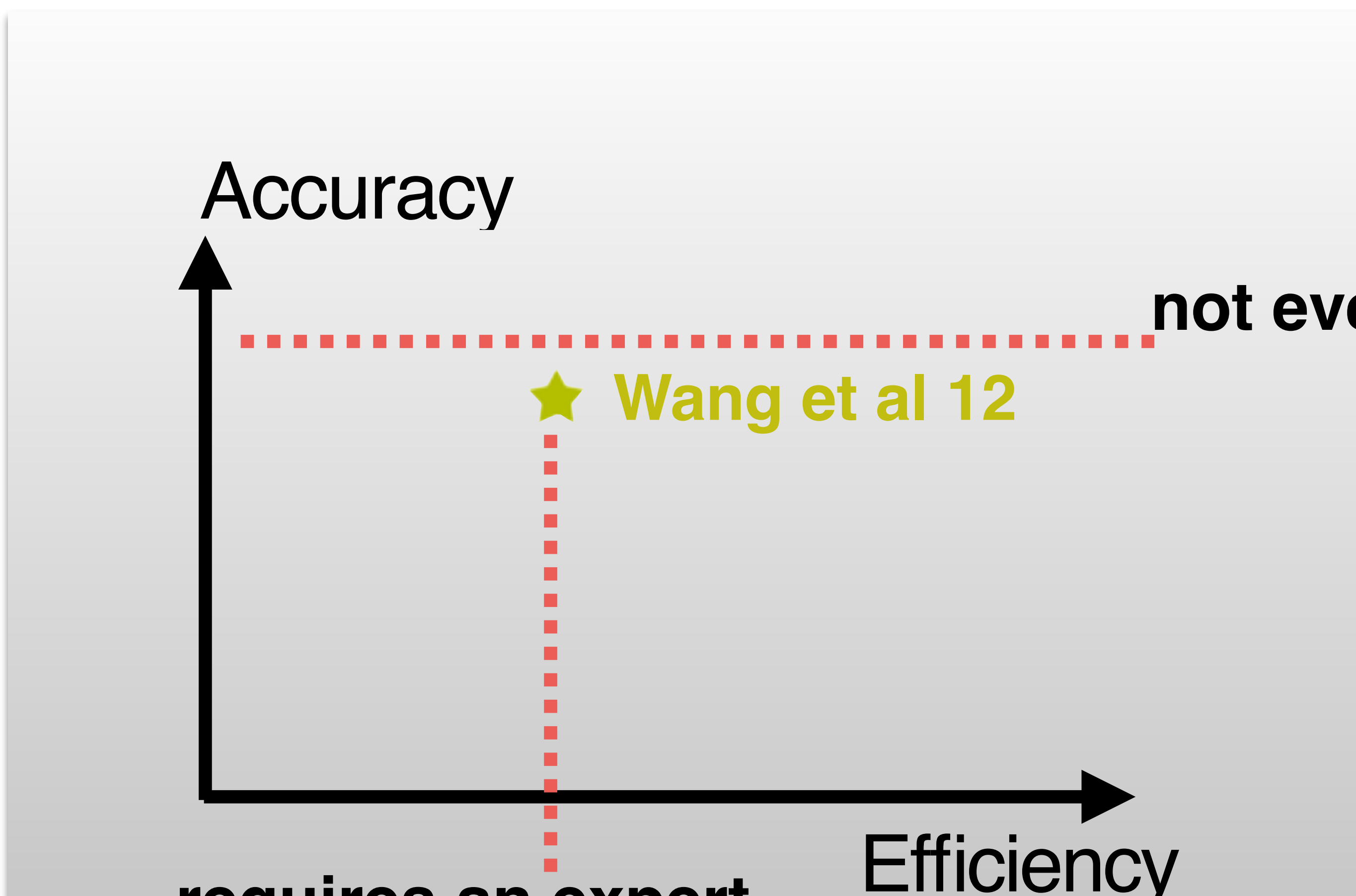
- Active learning for labeling [Wang'12]
  - Dataset: 1090 meshes, 2-5 parts per mesh, watertight manifold



Training Data

Where do we get segmented shapes?

- Active learning for labeling [Wang'12]
  - Dataset: 1090 meshes, 2-5 parts per mesh, watertight manifold



not every model is verified by a human

requires an expert

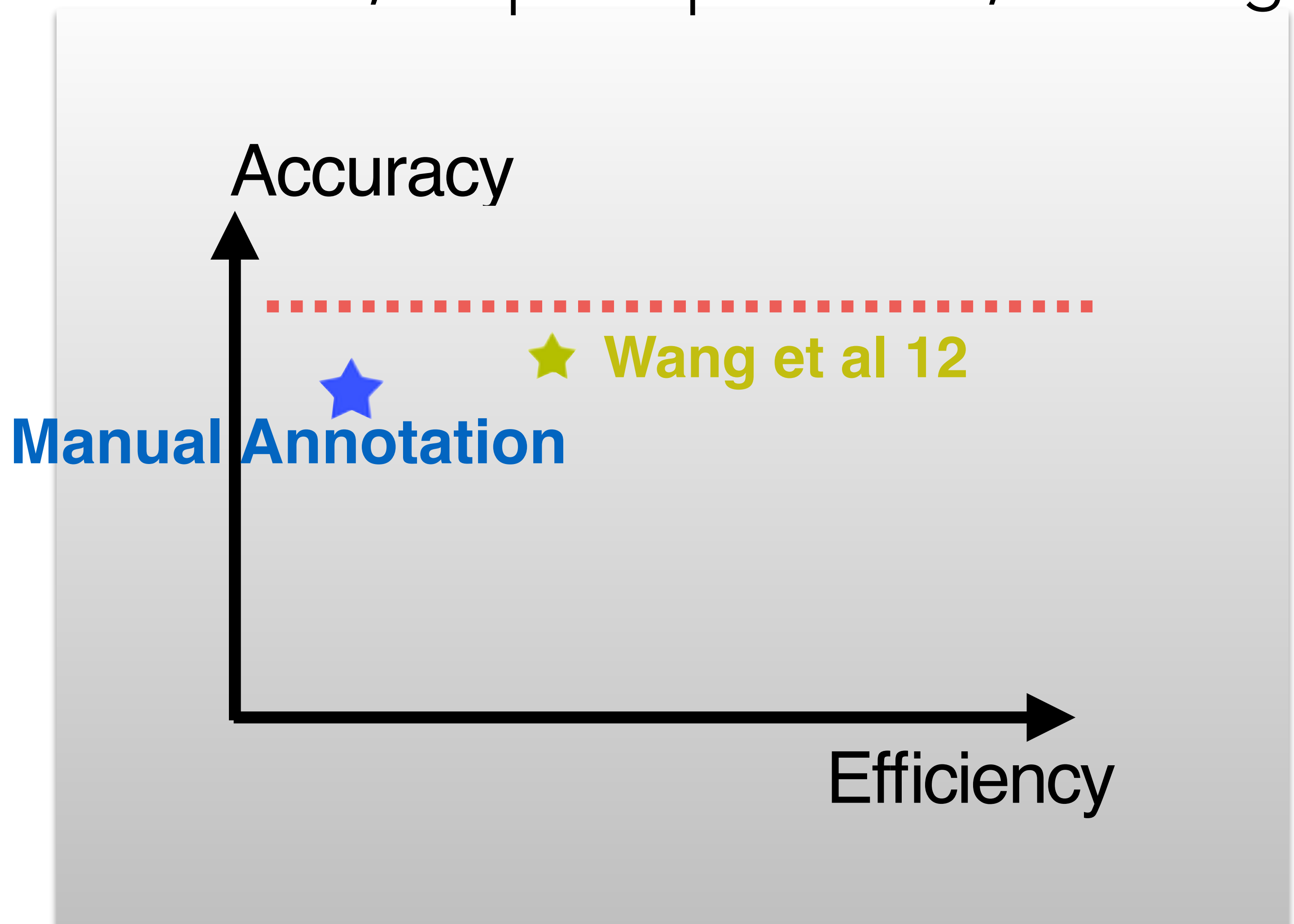
does not directly optimize efficiency

(active learning objective: train the best classifier)

Training Data

Where do we get segmented shapes?

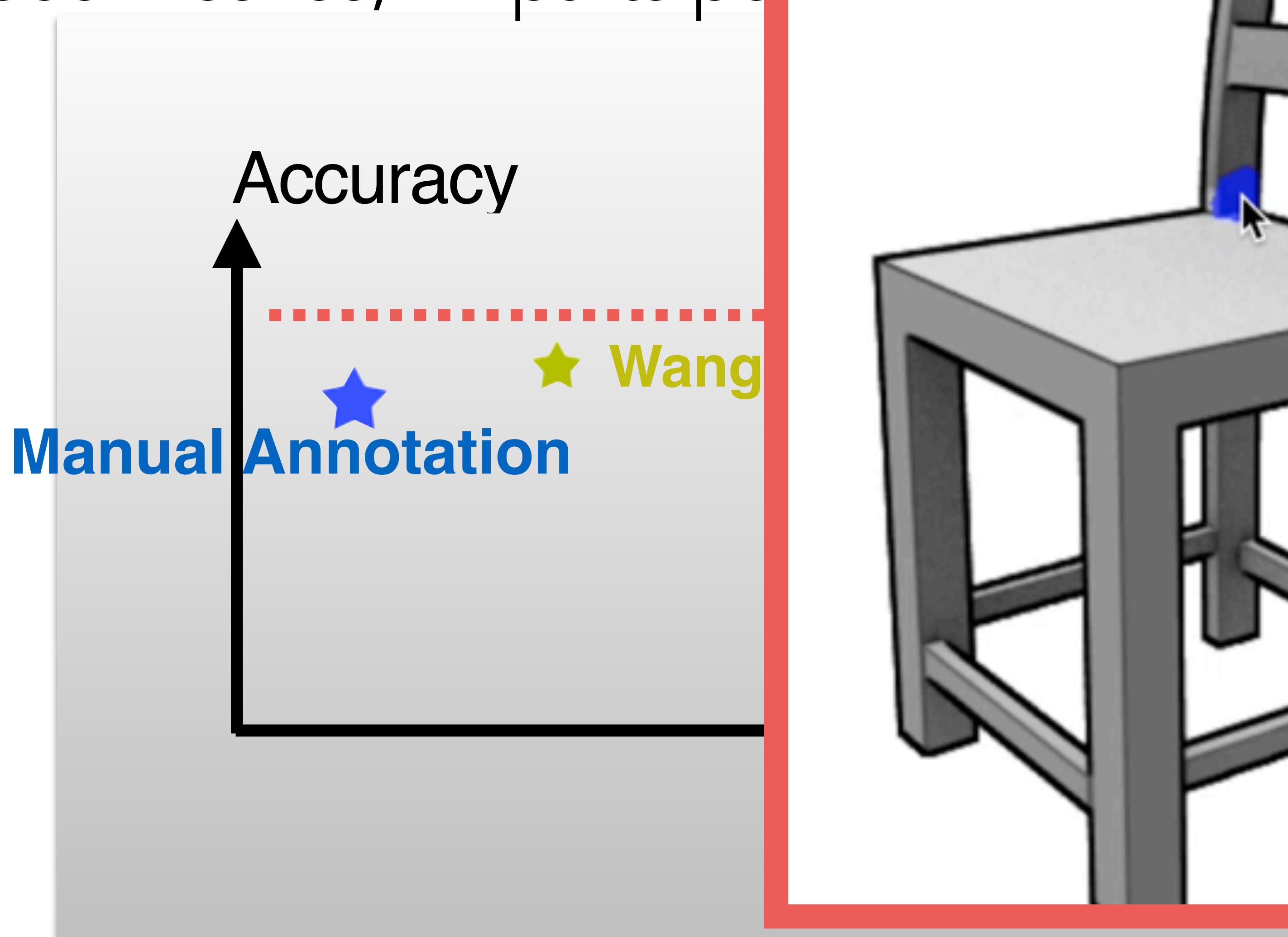
- Active learning for labeling [Wang'12]
  - Dataset: 1090 meshes, 2-5 parts per mesh, watertight manifold
- Manual annotation [Chen'09]
  - Dataset: 380 meshes, 19 parts per mesh, watertight manifold

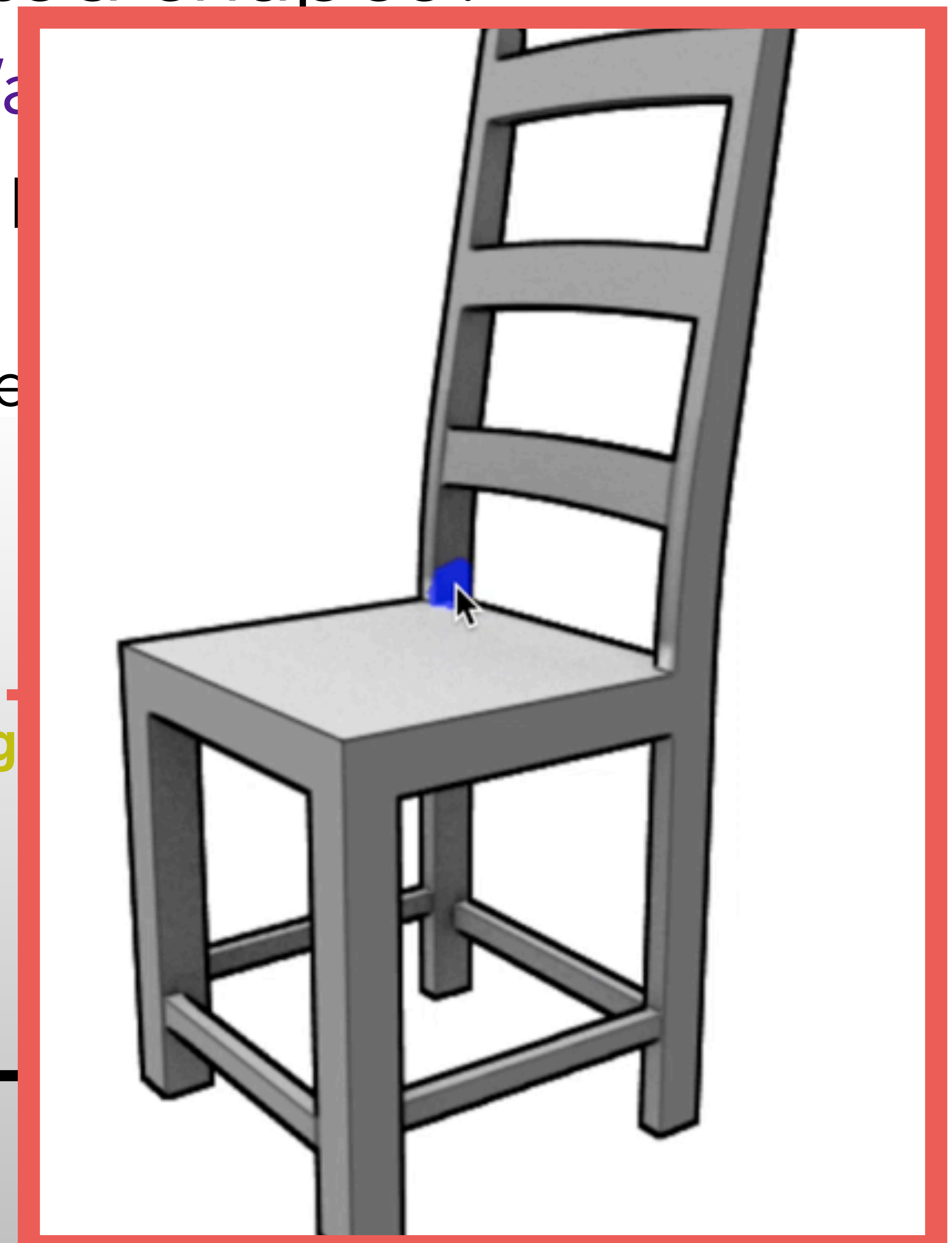


Training Data

Where do we get segmented shapes?

- Active learning for labeling [Wa
  - Dataset: 1090 meshes, 2-5 parts
- Manual annotation [Chen'09]
  - Dataset: 380 meshes, 19 parts pe



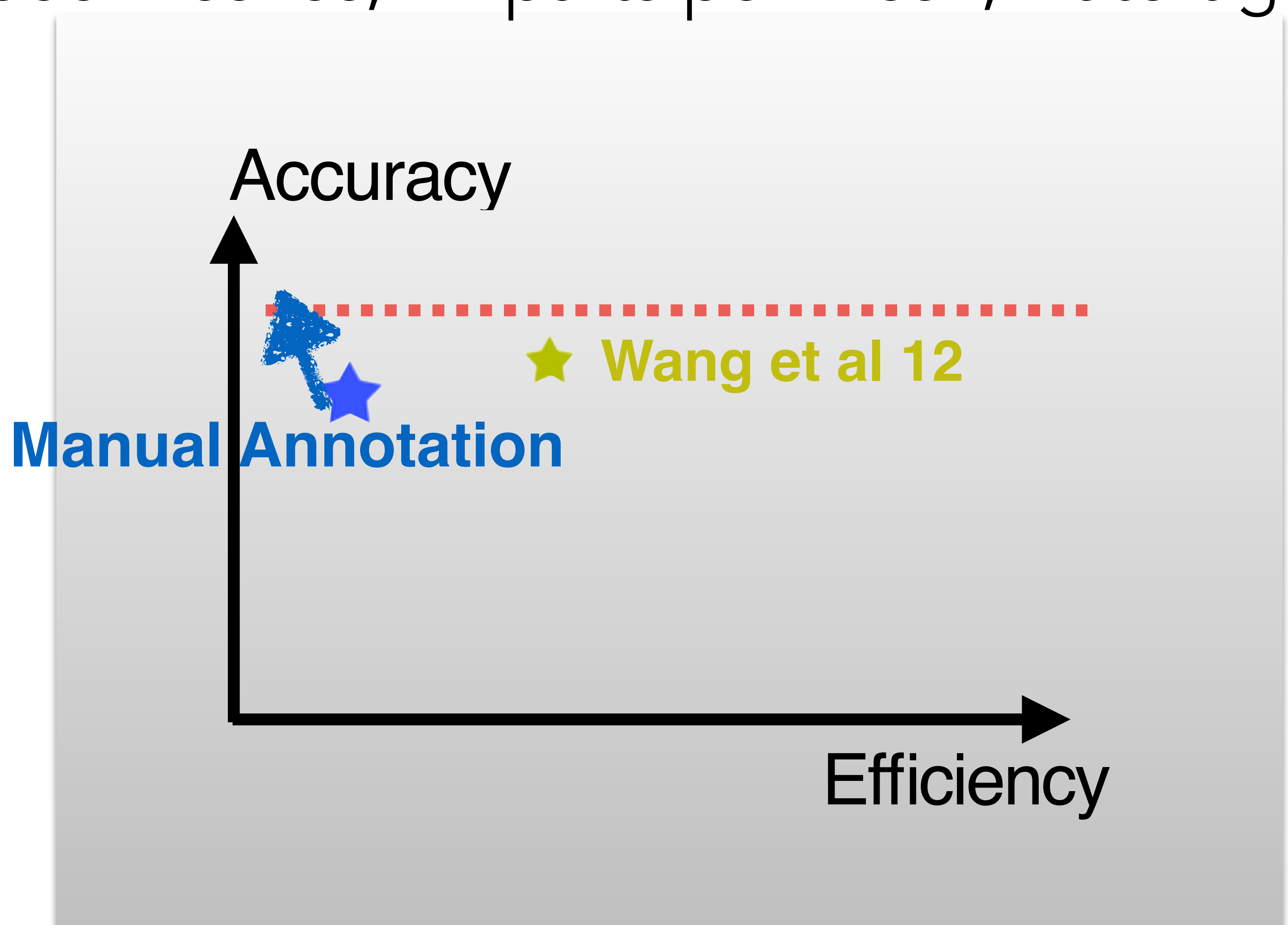


Training Data

Where do we get segmented shapes?

- Active learning for labeling [Wang'12]
  - Dataset: 1090 meshes, 2-5 parts per mesh, watertight manifold
- Manual annotation [Chen'09]
  - Dataset: 380 meshes, 19 parts per mesh, watertight manifold

Verify all results!

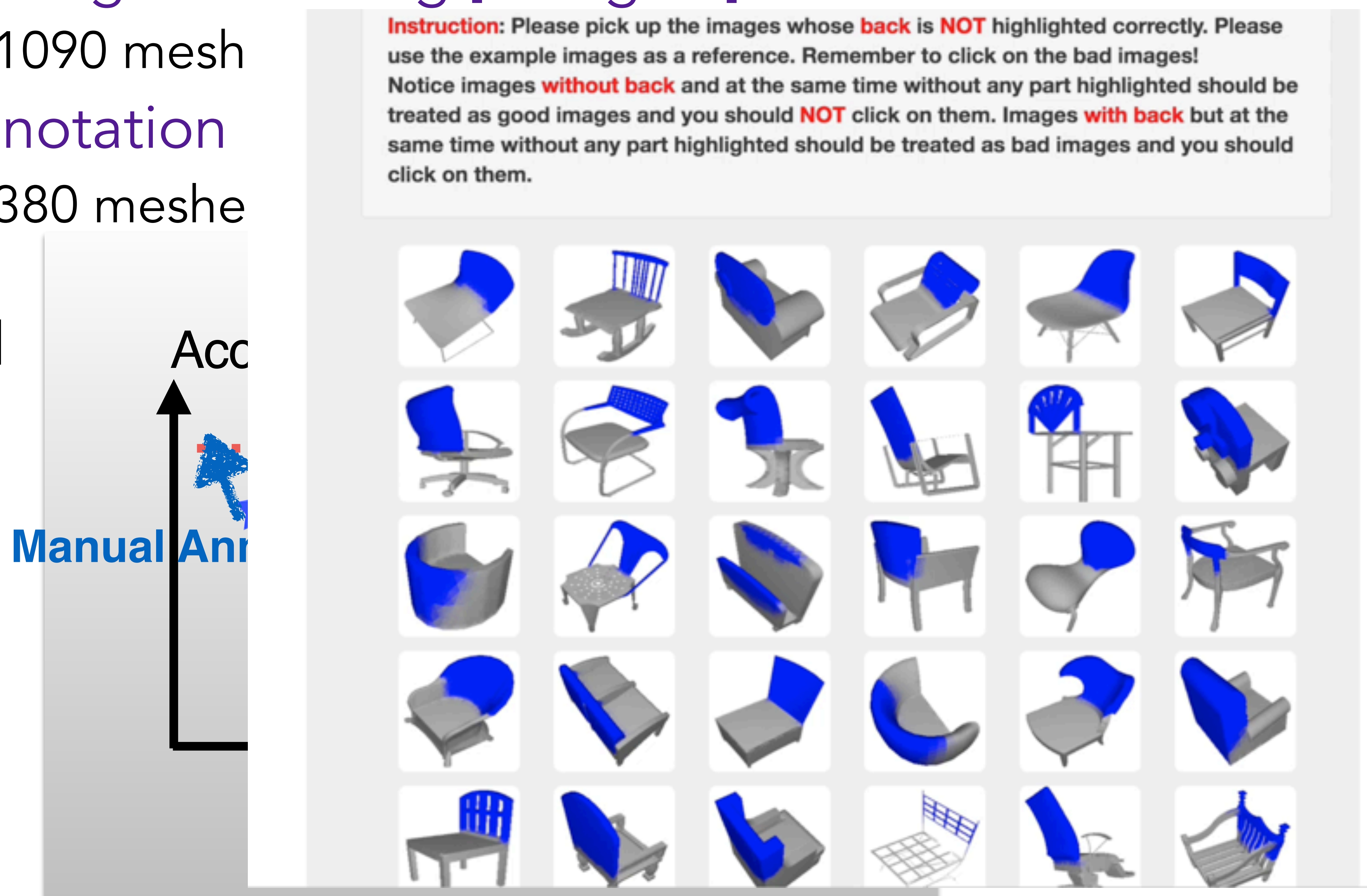


Training Data

Where do we get segmented shapes?

- Active learning for labeling [Wang'12]
  - Dataset: 1090 mesh
- Manual annotation
  - Dataset: 380 meshe

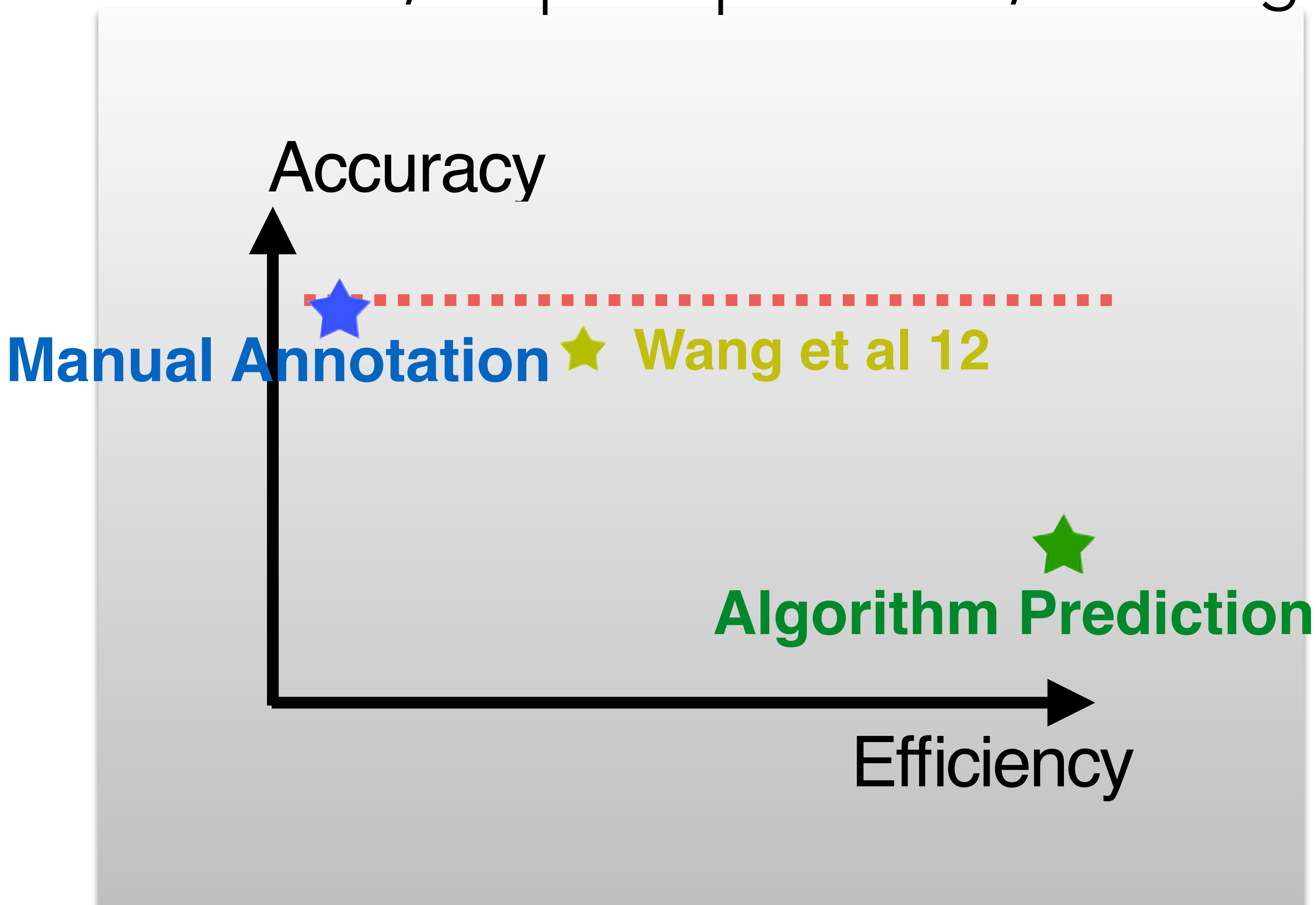
Verify all results!



Training Data

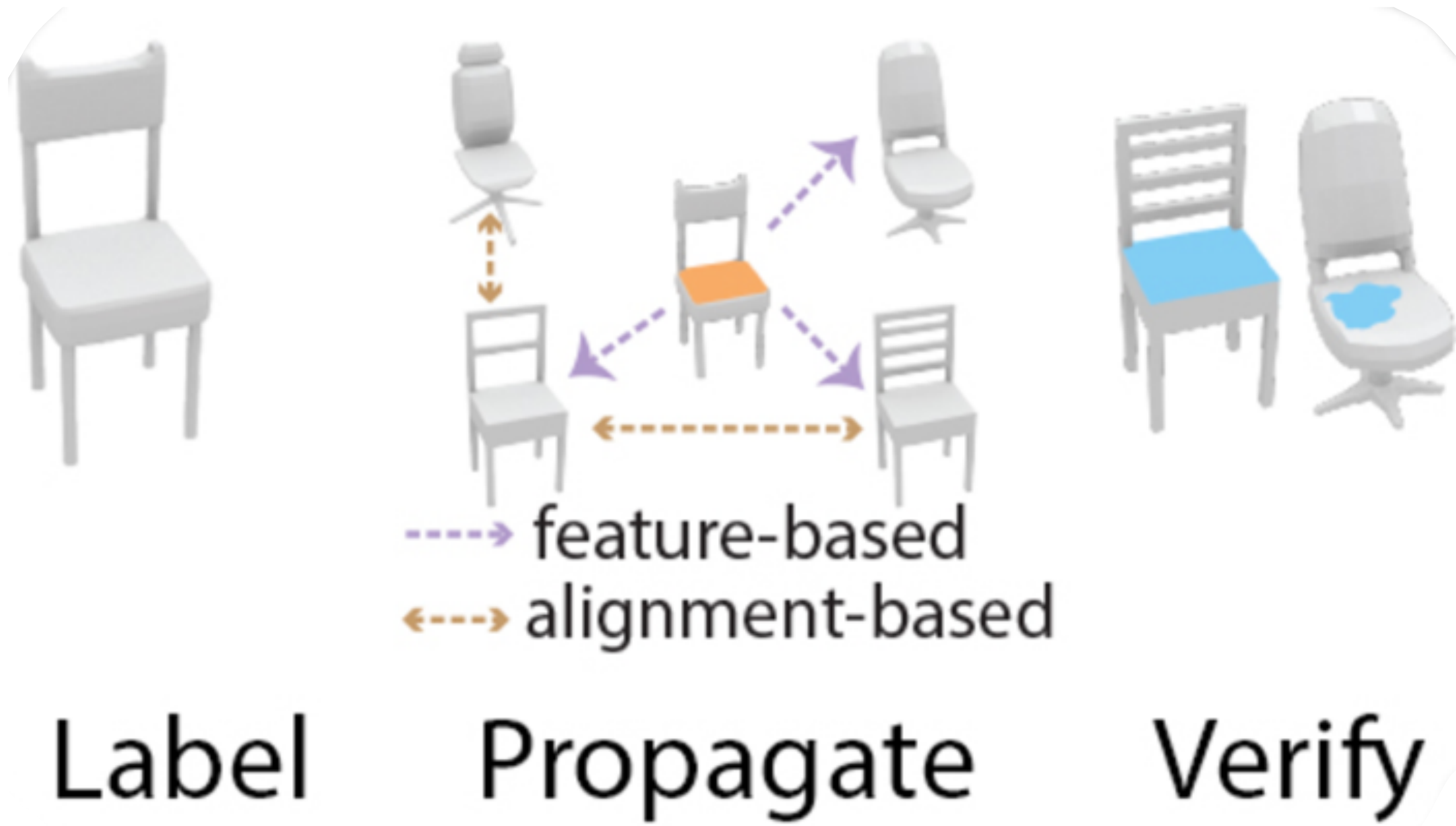
Where do we get segmented shapes?

- Active learning for labeling [Wang'12]
  - Dataset: 1090 meshes, 2-5 parts per mesh, watertight manifold
- Manual annotation [Chen'09]
  - Dataset: 380 meshes, 19 parts per mesh, watertight manifold



Our Approach

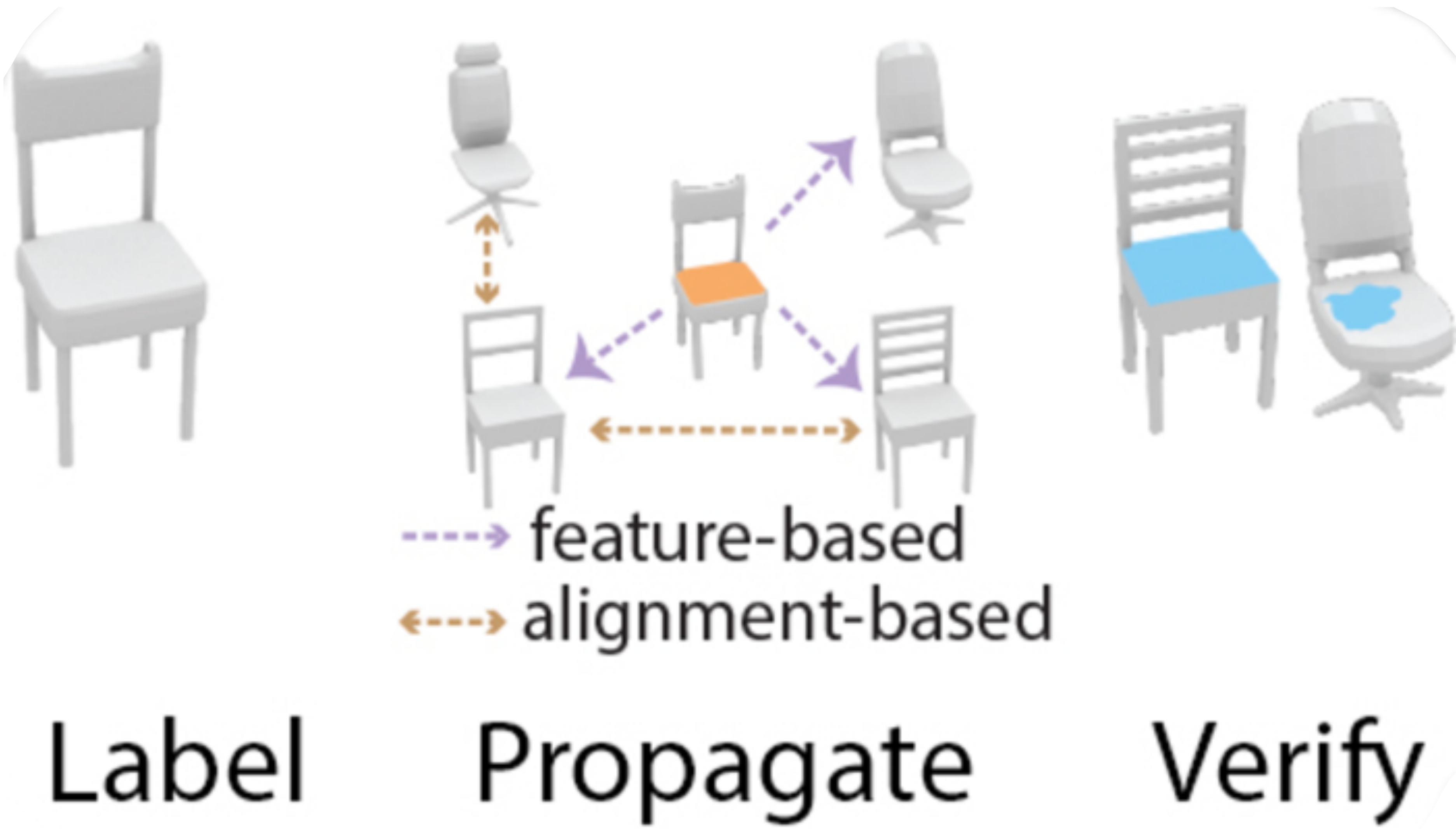
Verify algorithmic predictions



A Scalable Active Framework for Region Annotation in 3D Shape Collections Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas SIGGRAPH Asia 2016

Our Approach

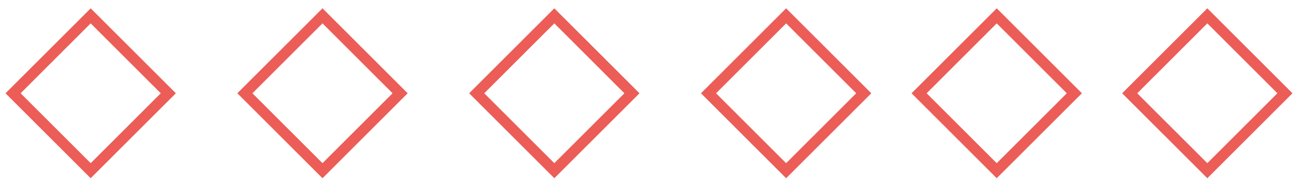
Verify algorithmic predictions



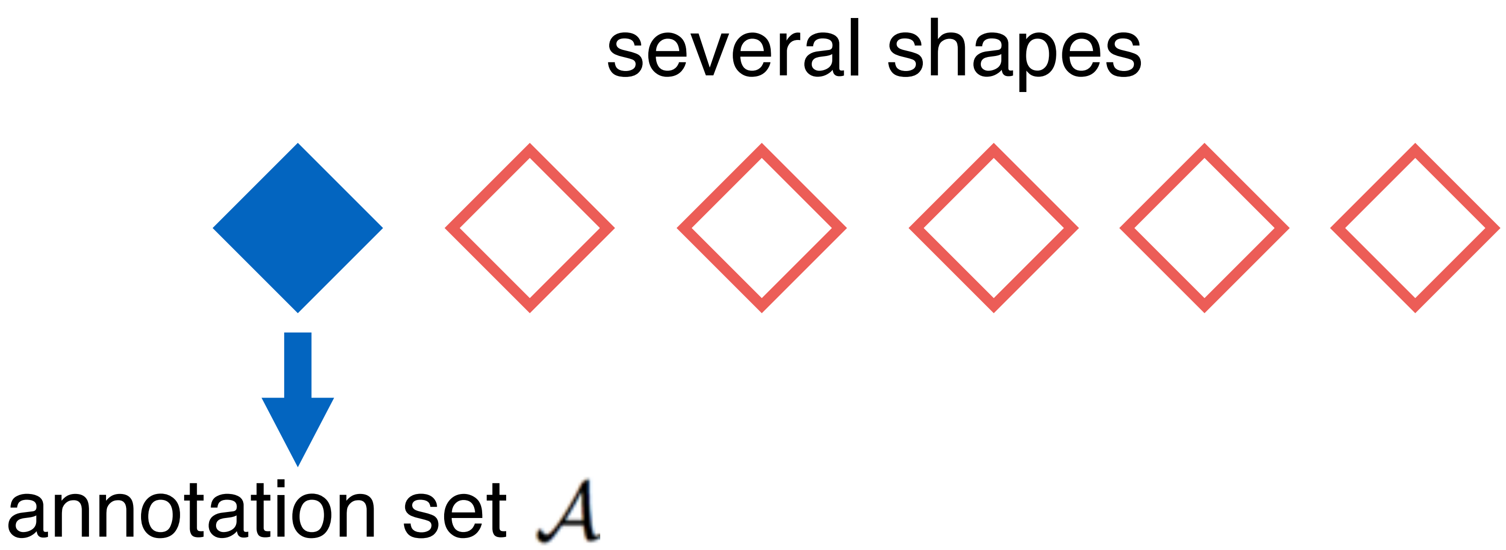
Which models to label & which labels to verify?

Utility Function

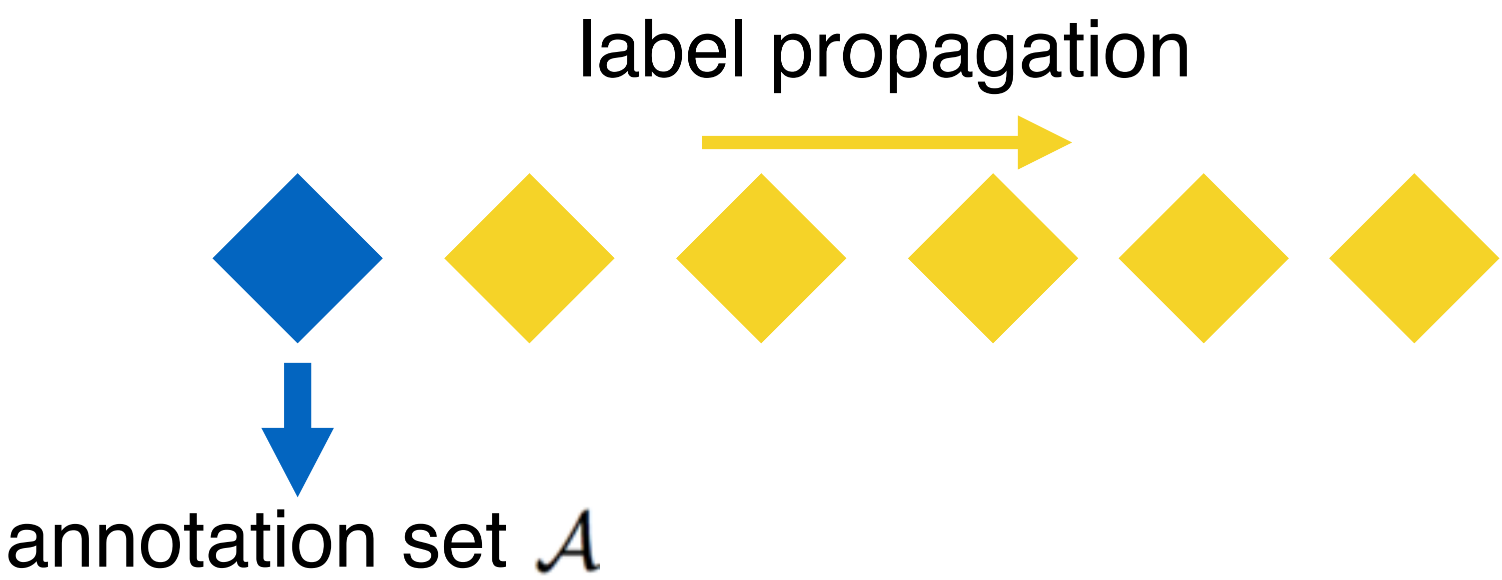
several shapes



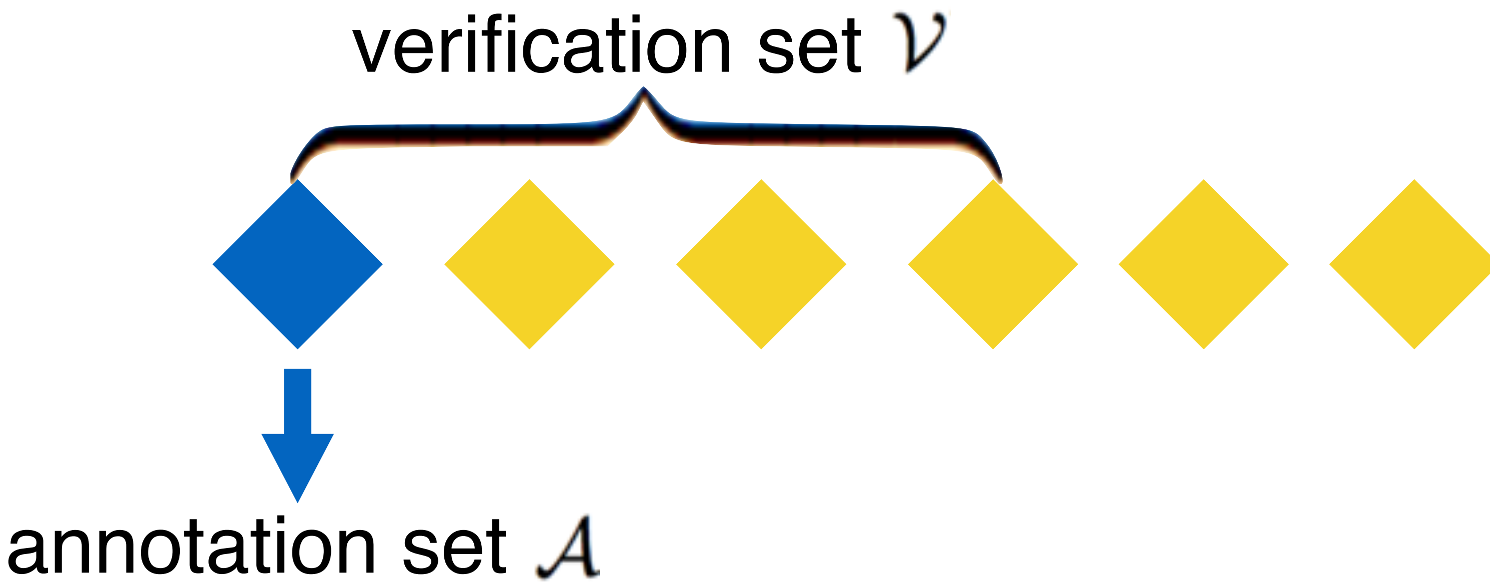
Utility Function



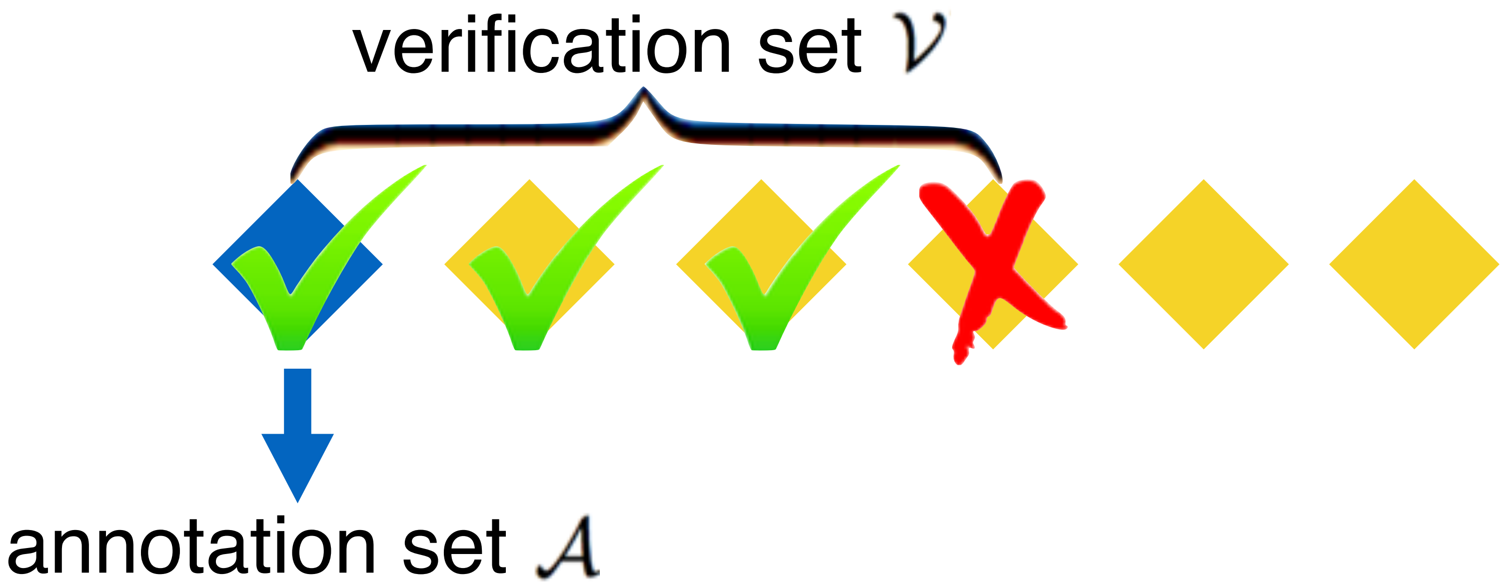
Utility Function



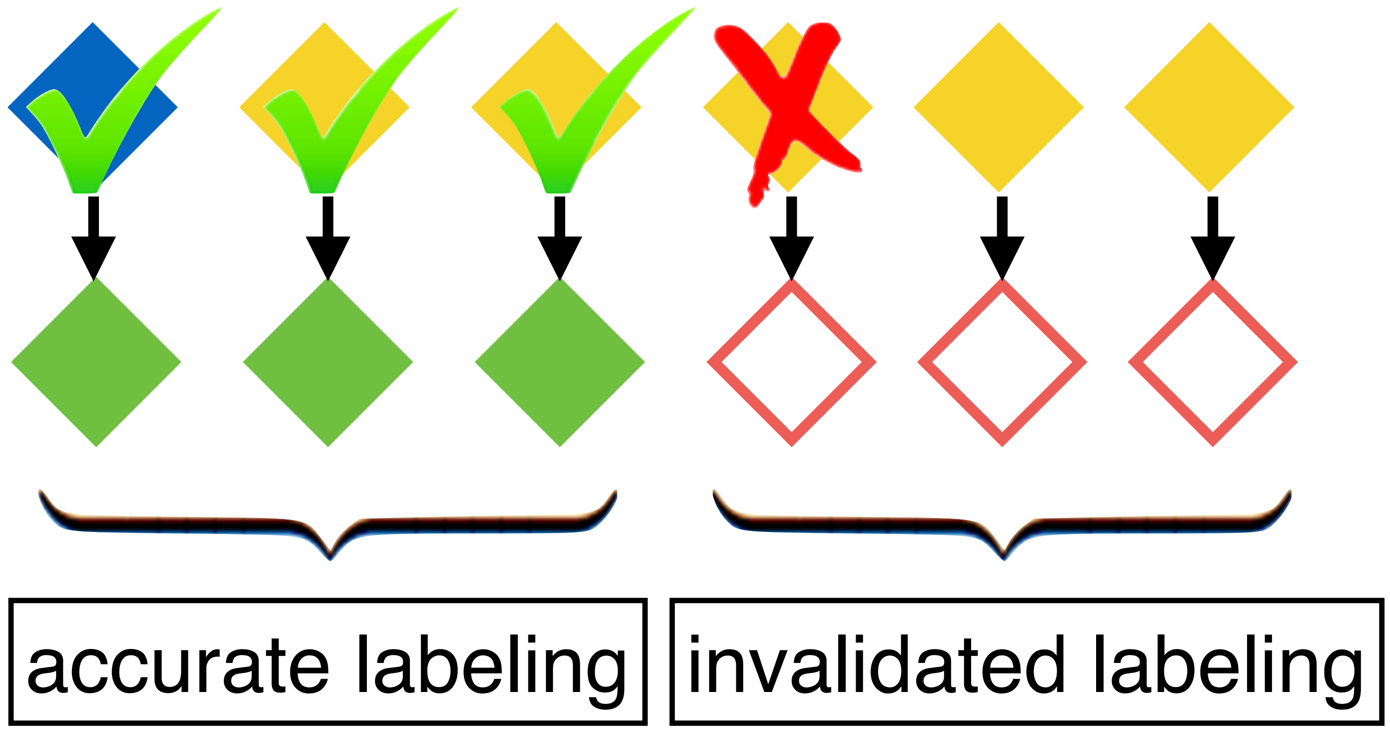
Utility Function



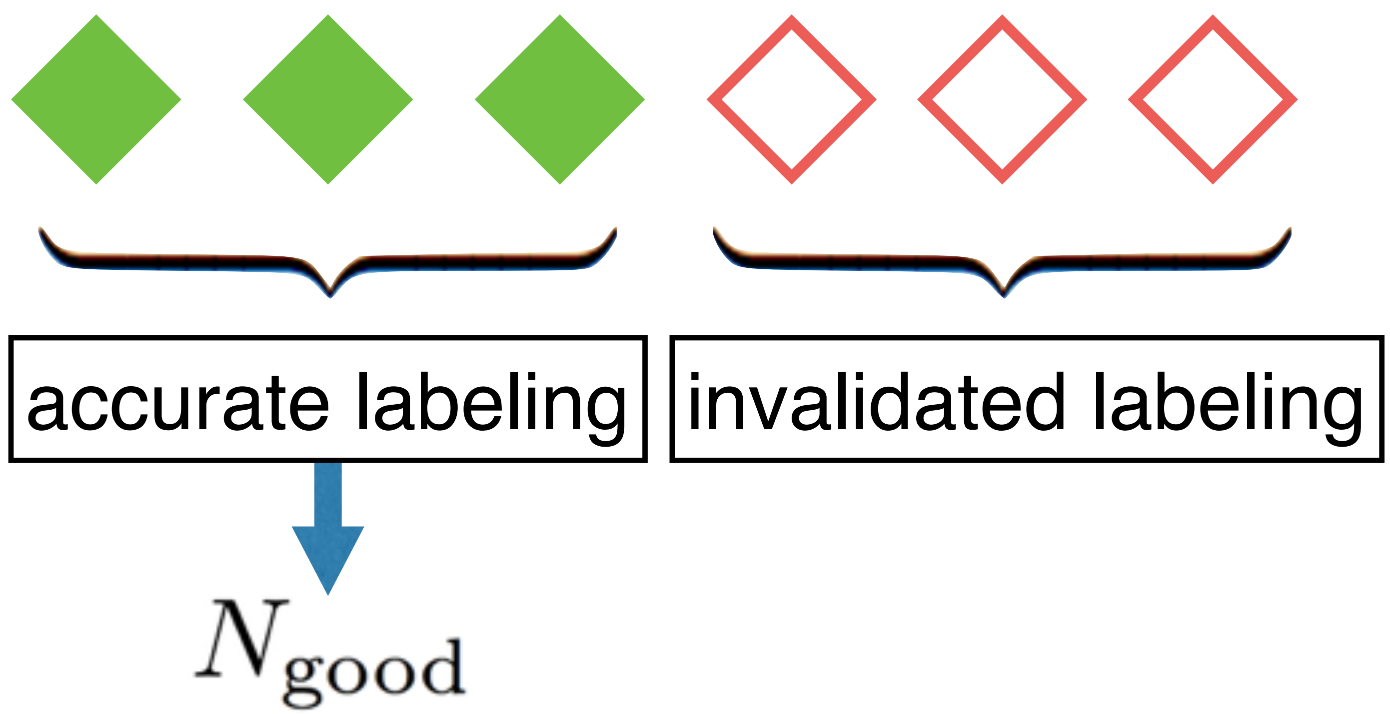
Utility Function



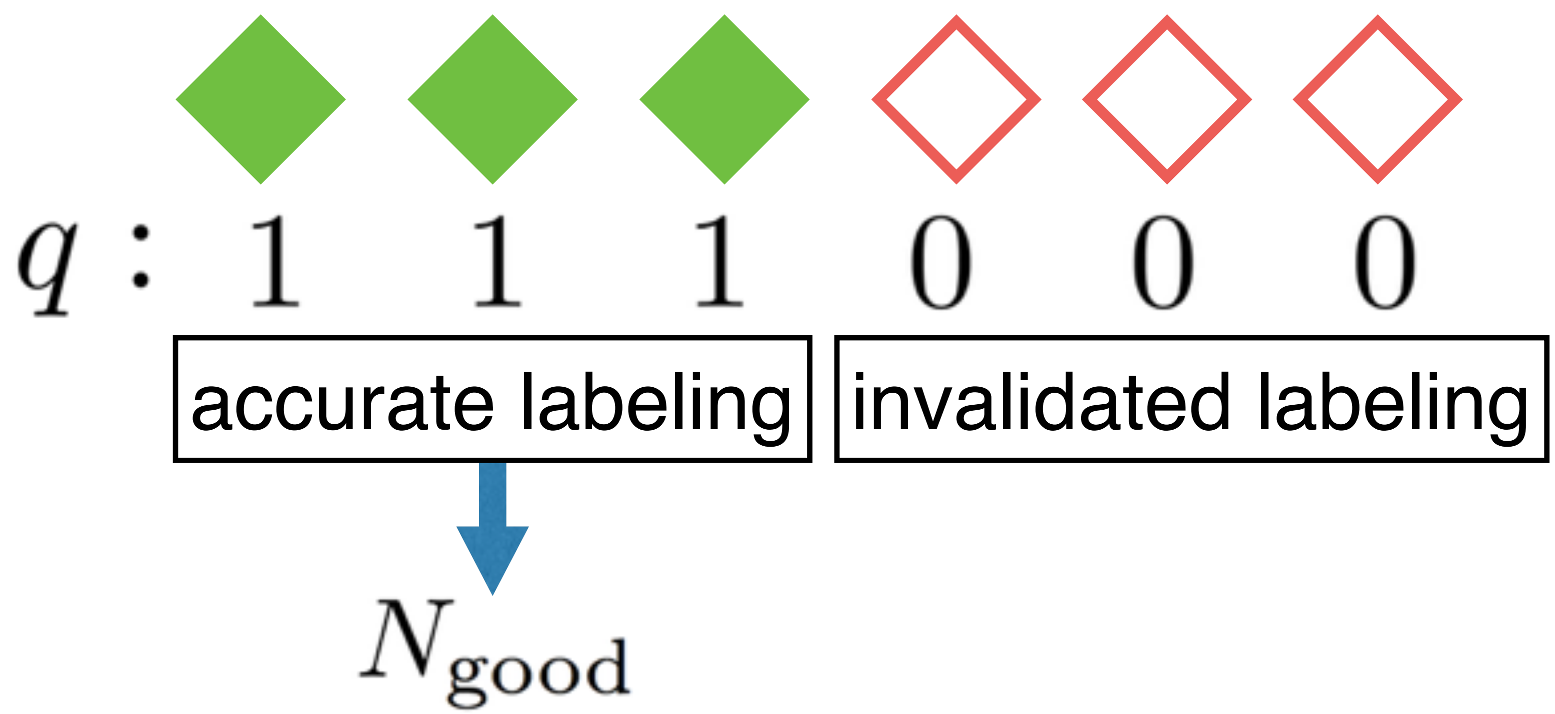
Utility Function



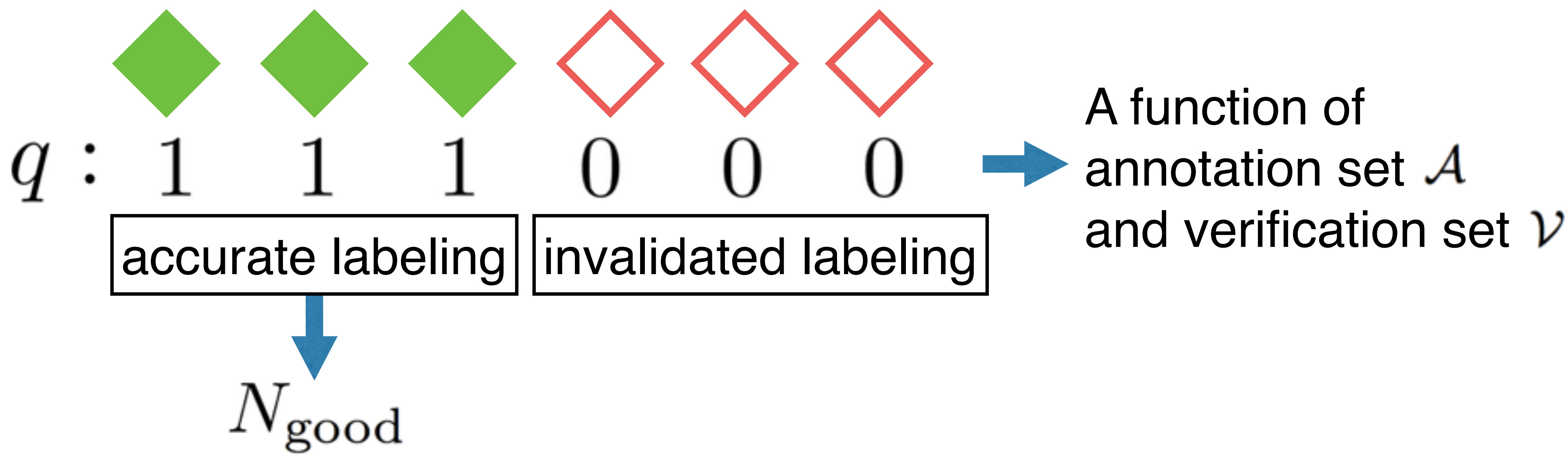
Utility Function



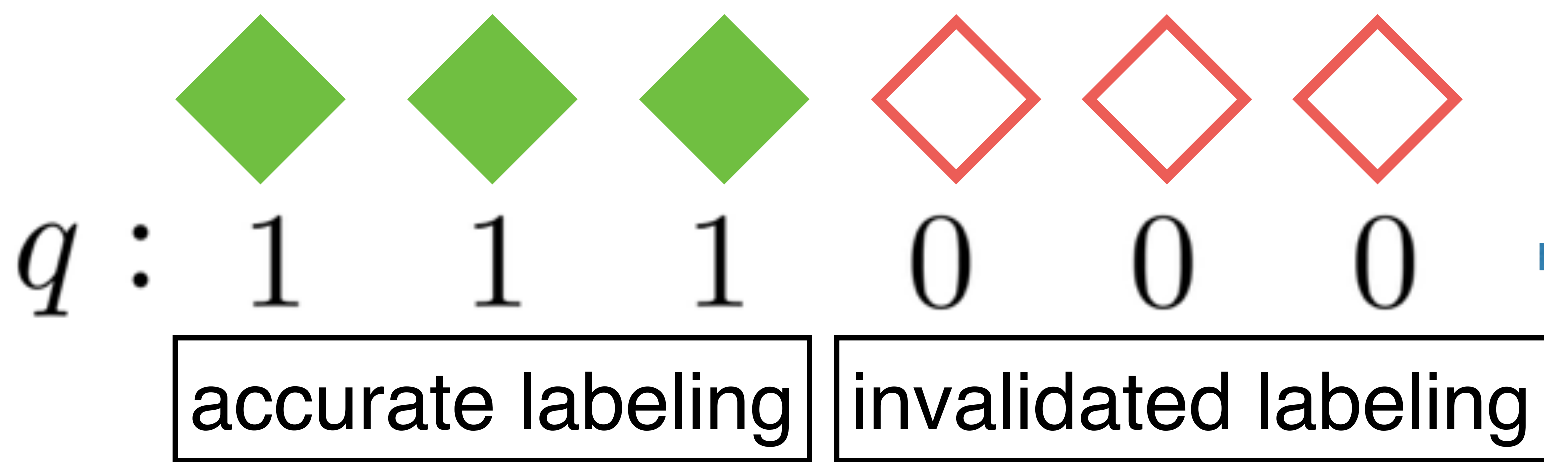
Utility Function



Utility Function



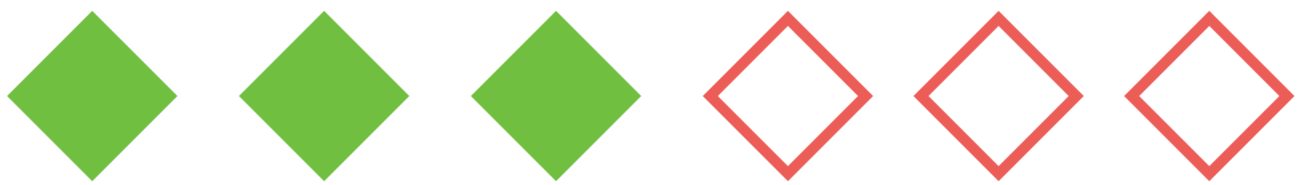
Utility Function



A function of annotation set $\mathcal{A}$ and verification set $\mathcal{V}$

$$N_{\text{good}} = \sum q(\mathcal{A}, \mathcal{V})$$

Utility Function



$q: 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0$

A function of annotation set $\mathcal{A}$ and verification set $\mathcal{V}$

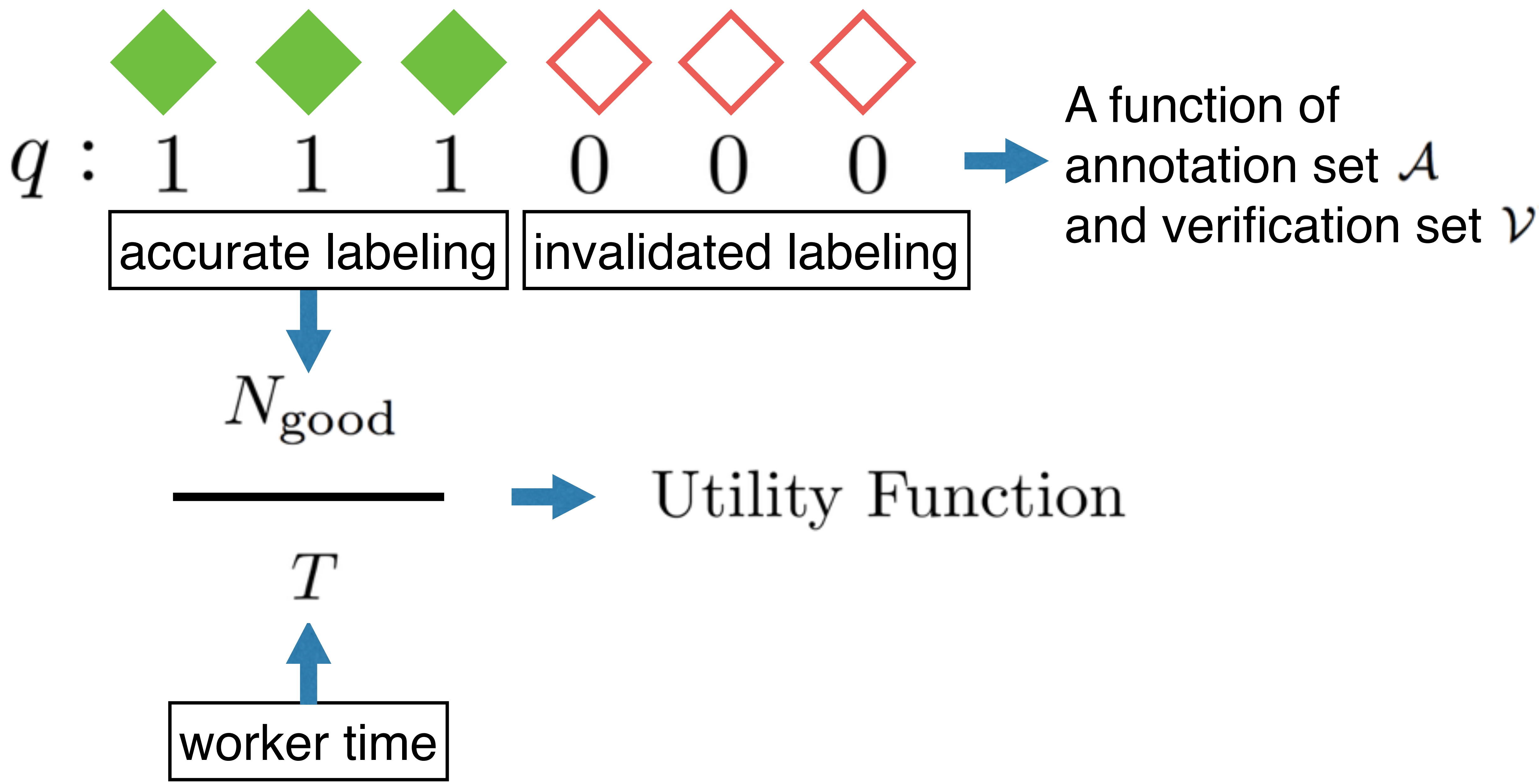
accurate labeling

invalidated labeling

$$\frac{N_{\text{good}}}{T}$$

worker time

Utility Function

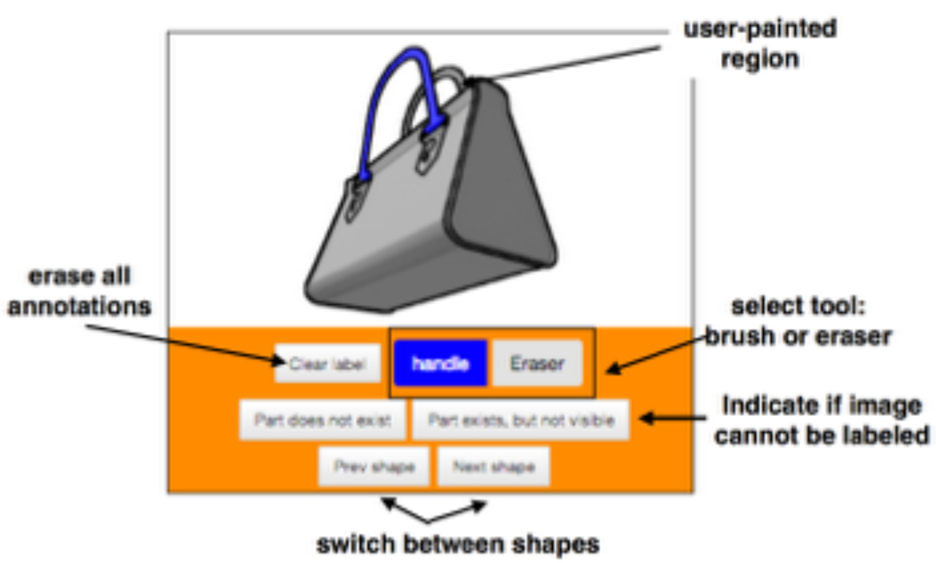


Utility Function

$$T = t_{\text{annotation}}(\mathcal{A}) + t_{\text{verification}}(\mathcal{V}, \{q\})$$

Utility Function

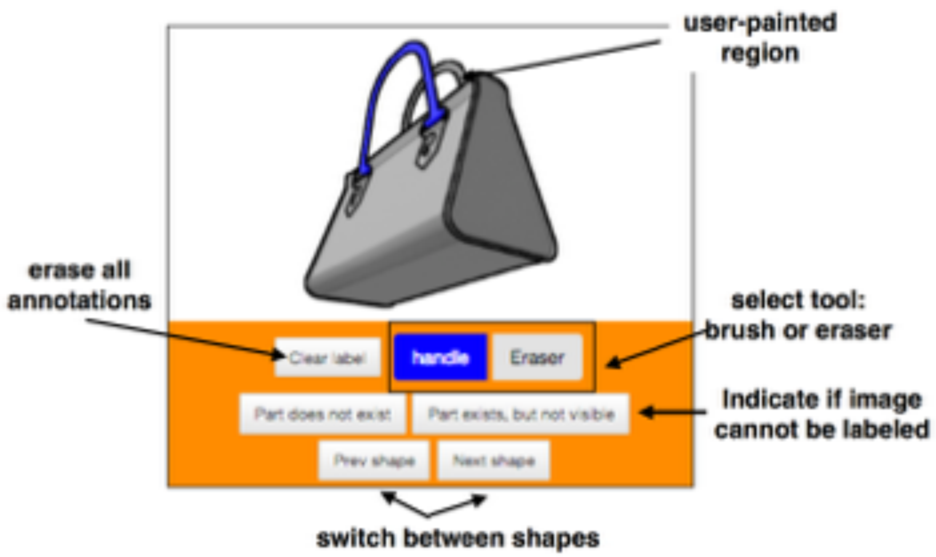
$$T = t_{\text{annotation}}(\mathcal{A}) + t_{\text{verification}}(\mathcal{V}, \{q\})$$



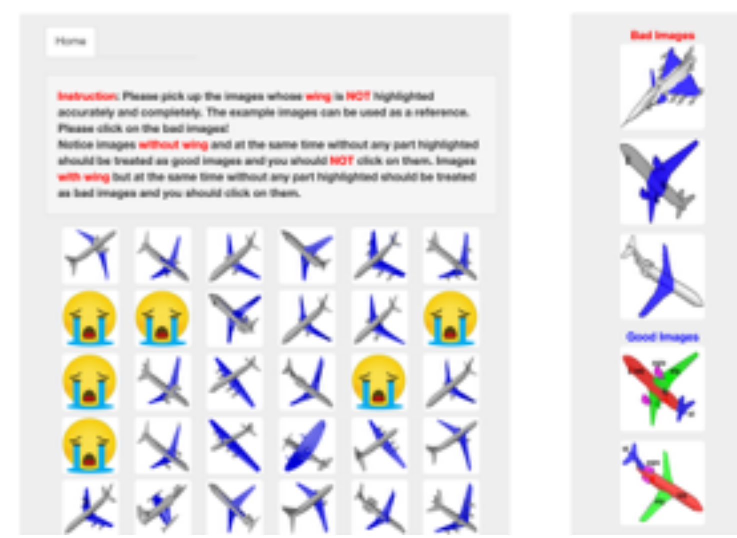
$$\tau_{\text{annotation}} \approx 30s/\text{label}$$

Utility Function

$$T = t_{\text{annotation}}(\mathcal{A}) + t_{\text{verification}}(\mathcal{V}, \{q\})$$



$$\tau_{\text{annotation}} \approx 30s/\text{label}$$



$$\tau_{\text{identification}} \approx 0.3s/\text{label}$$

$$\tau_{\text{click}} \approx 1.1s/\text{label}$$

Utility Function

$$\frac{N_{\text{good}}}{T}$$

Utility Function

$$\frac{N_{\text{good}}}{T}$$

$\{q\}$ is not known before selecting $\mathcal{A}$ and $\mathcal{V}$

Utility Function

$$\frac{N_{\text{good}}}{T}$$

$\{q\}$ is not known before selecting $\mathcal{A}$ and $\mathcal{V}$

Estimate probability and use expectation

Utility Function

$$\frac{N_{\text{good}}}{T}$$

$\{q\}$ is not known before selecting $\mathcal{A}$ and $\mathcal{V}$

Estimate probability and use expectation

$$\boxed{\frac{\mathbb{E}[N_{\text{good}}]}{\mathbb{E}[T]}}$$

Utility Function

$$\frac{N_{\text{good}}}{T}$$

$\{q\}$ is not known before selecting $\mathcal{A}$ and $\mathcal{V}$

Estimate probability and use expectation

$$\boxed{\frac{\mathbb{E}[N_{\text{good}}]}{\mathbb{E}[T]}}$$

Algorithm behavior

Utility Function

$$\operatorname*{argmin}_{\mathcal{A},\mathcal{V}} \frac{\mathbb{E}[N_{\mathbf{good}}]}{\mathbb{E}[T]}$$

Utility Function

$$\underset{\mathcal{A},\mathcal{V}}{\operatorname{argmin}} \frac{\mathbb{E}[N_{\mathrm{good}}]}{\mathbb{E}[T]}$$

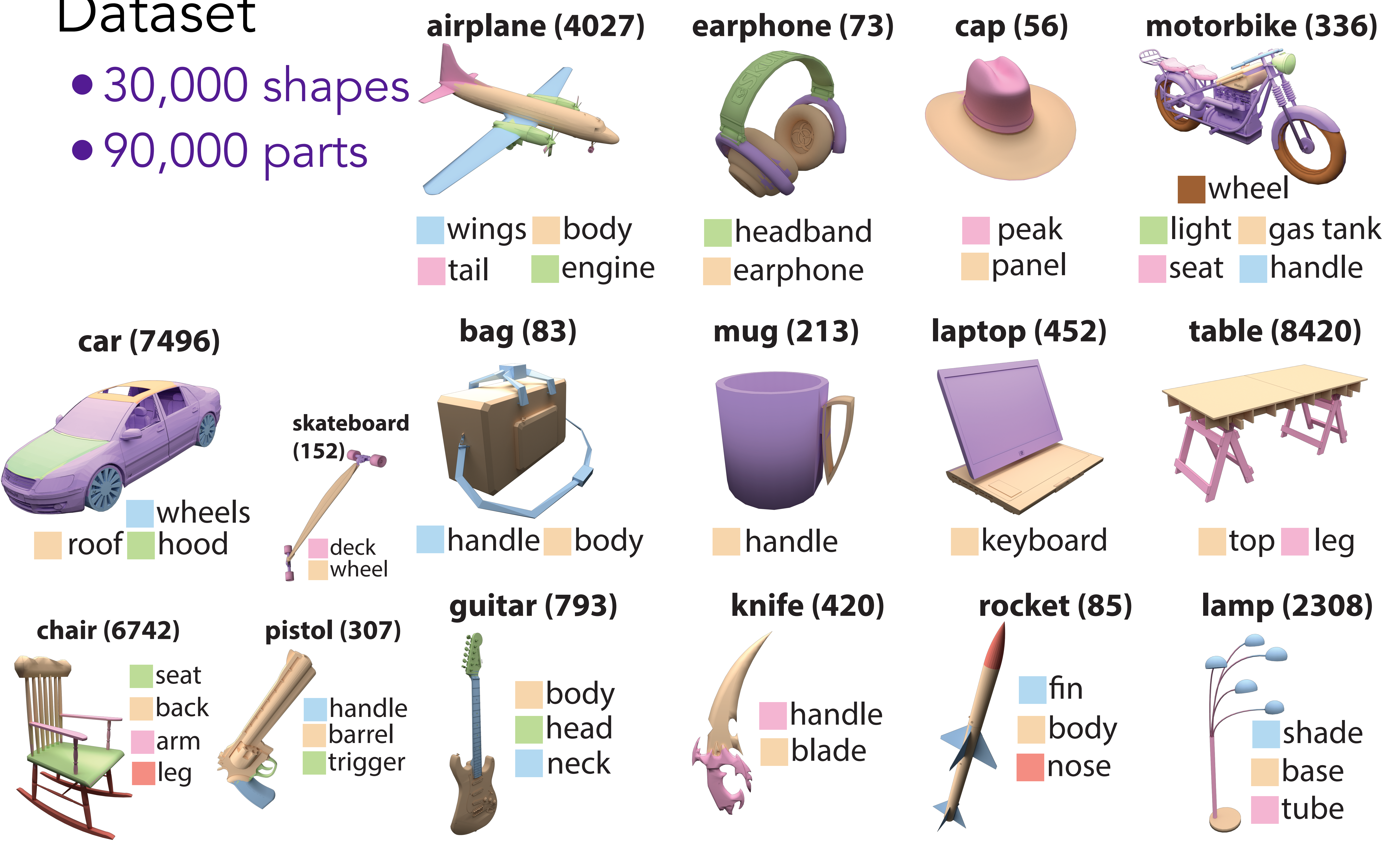
Select Annotation Set: ***Beam Search***

Select Verification Set: ***Greedy Search***

Results

Dataset

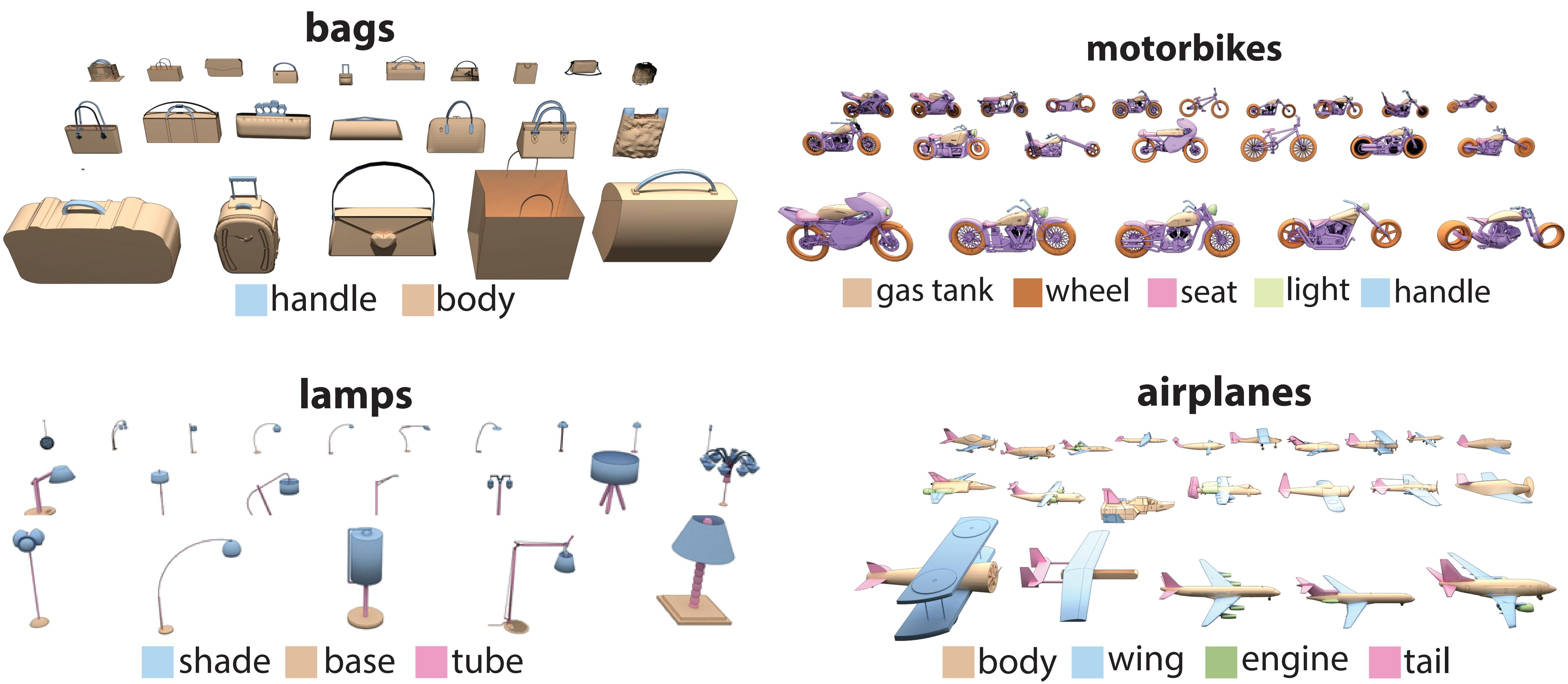
- 30,000 shapes
- 90,000 parts



Results

Dataset

- 30,000 shapes
- 90,000 parts



Time Saving

Comparison on manifold shapes

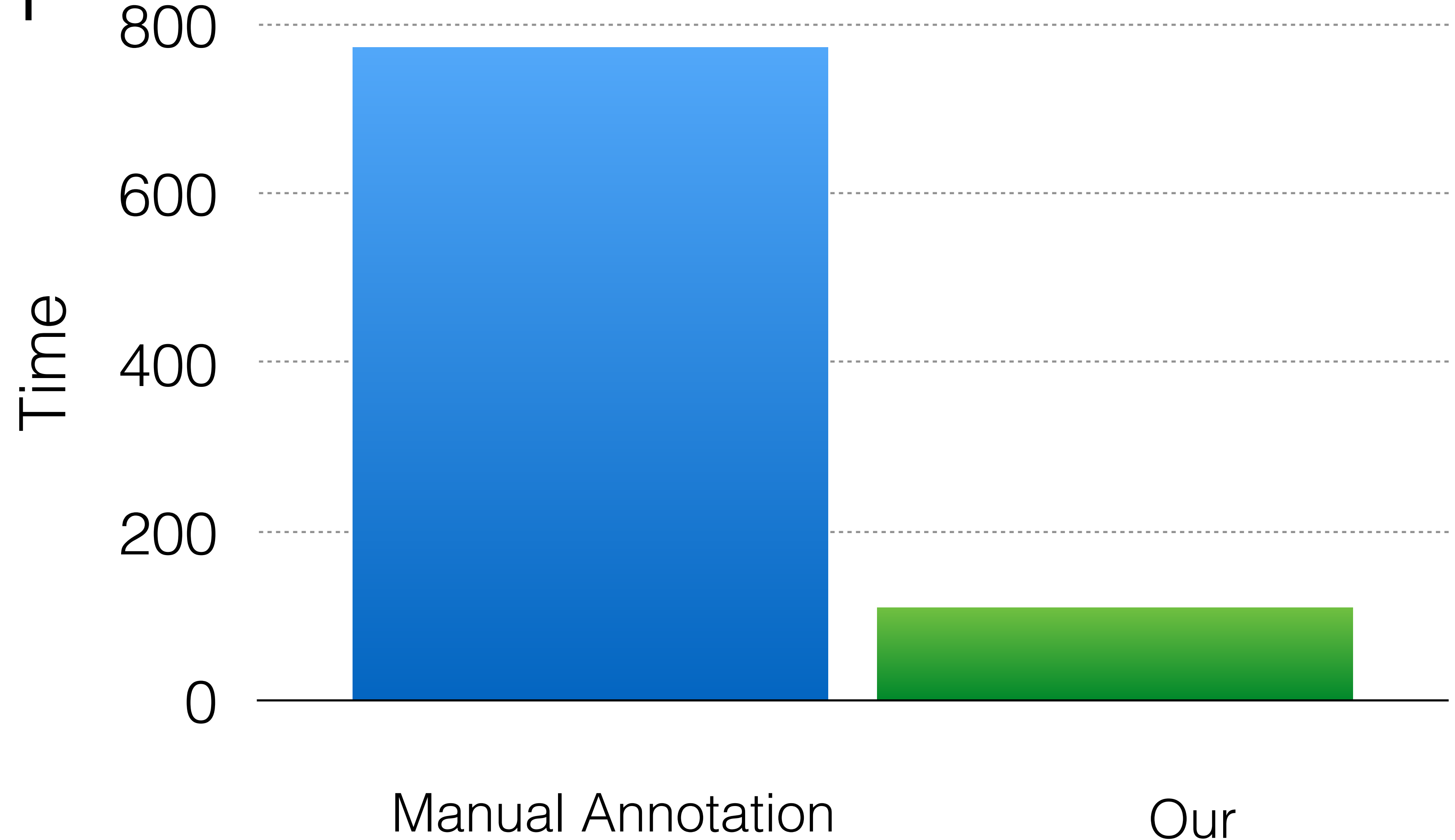
- Chen'09 (manual) : about x12 more expensive
- Wang'12 (active) : about x2 more expensive

Time Saving

Comparison on manifold shapes

- Chen'09 (manual) : about x12 more expensive
- Wang'12 (active) : about x2 more expensive

Comparison on data "in the wild"



Impact of Data

Training

Get Data

Label Data

Train Model

Testing

Apply Model

Impact of Data

[1] *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Hao Su, Charles Qi, Kaichun Mo, Leonidas Guibas. CVPR 2017*

[2] *SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation. Li Yi, Hao Su, Xingwen Guo, Leonidas Guibas. CVPR 2017*

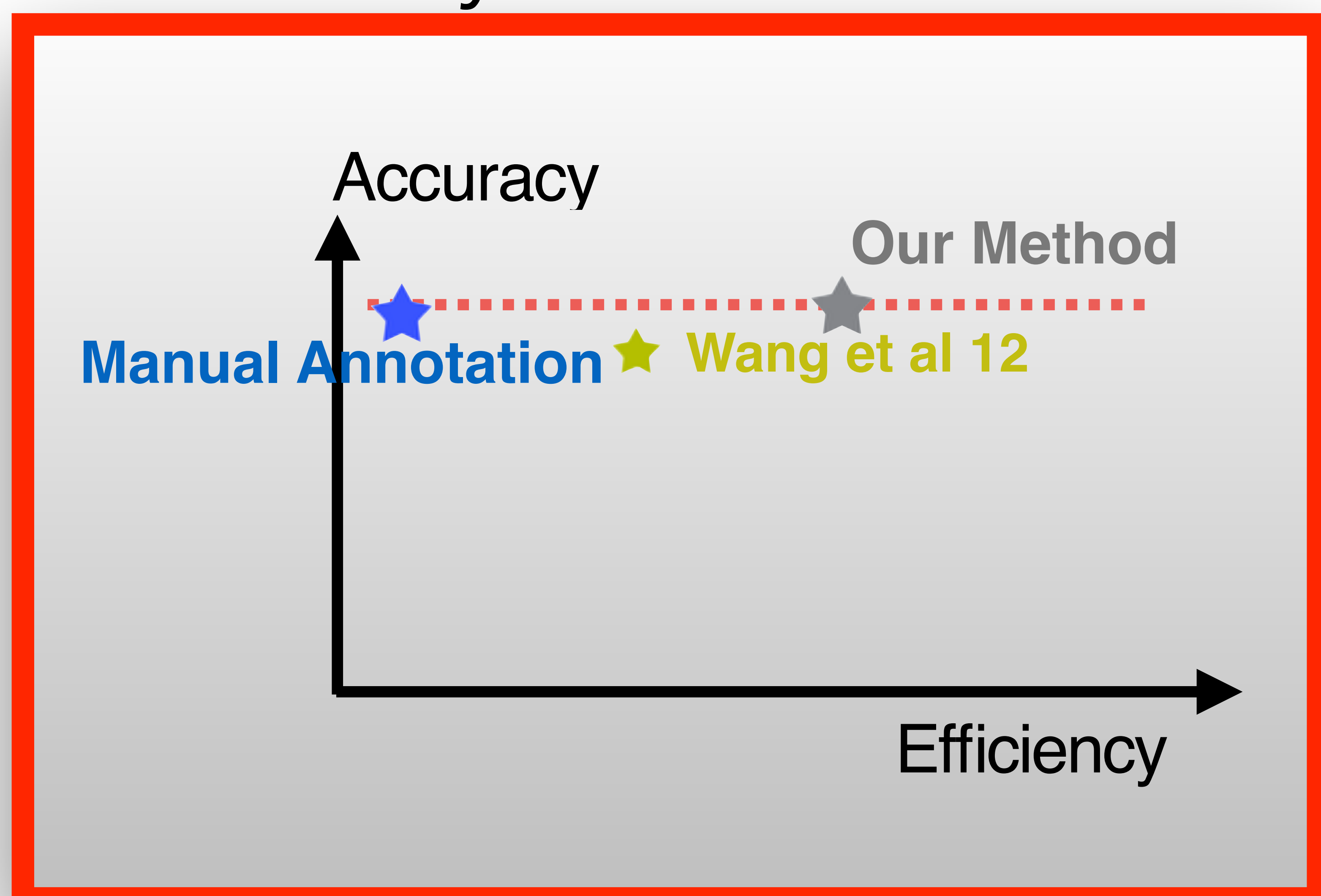
[3] *3D Shape Segmentation with Projective Convolutional Networks. Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji and Siddhartha Chaudhuri. CVPR 2017*

...

Are we done with labeling?

Smaller than analogues image datasets ≈30,000 vs ≈300,000 segmentations

Geometry varies in resolution (we only capture

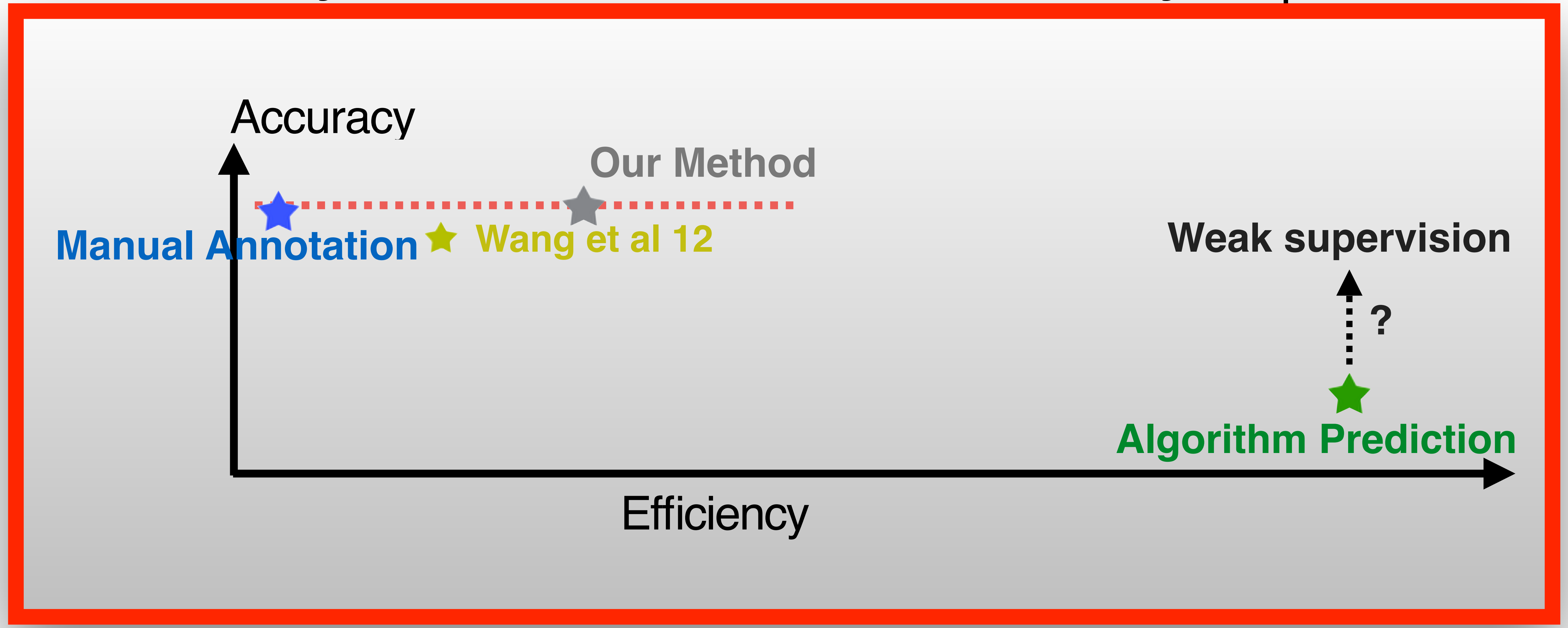


er dataset

Are we done with labeling?

Smaller than analogues image datasets ≈30,000 vs ≈300,000 segmentations

Geometry varies in resolution (we only capture



Part Labeling

Somebody provided data for you

Human labeling

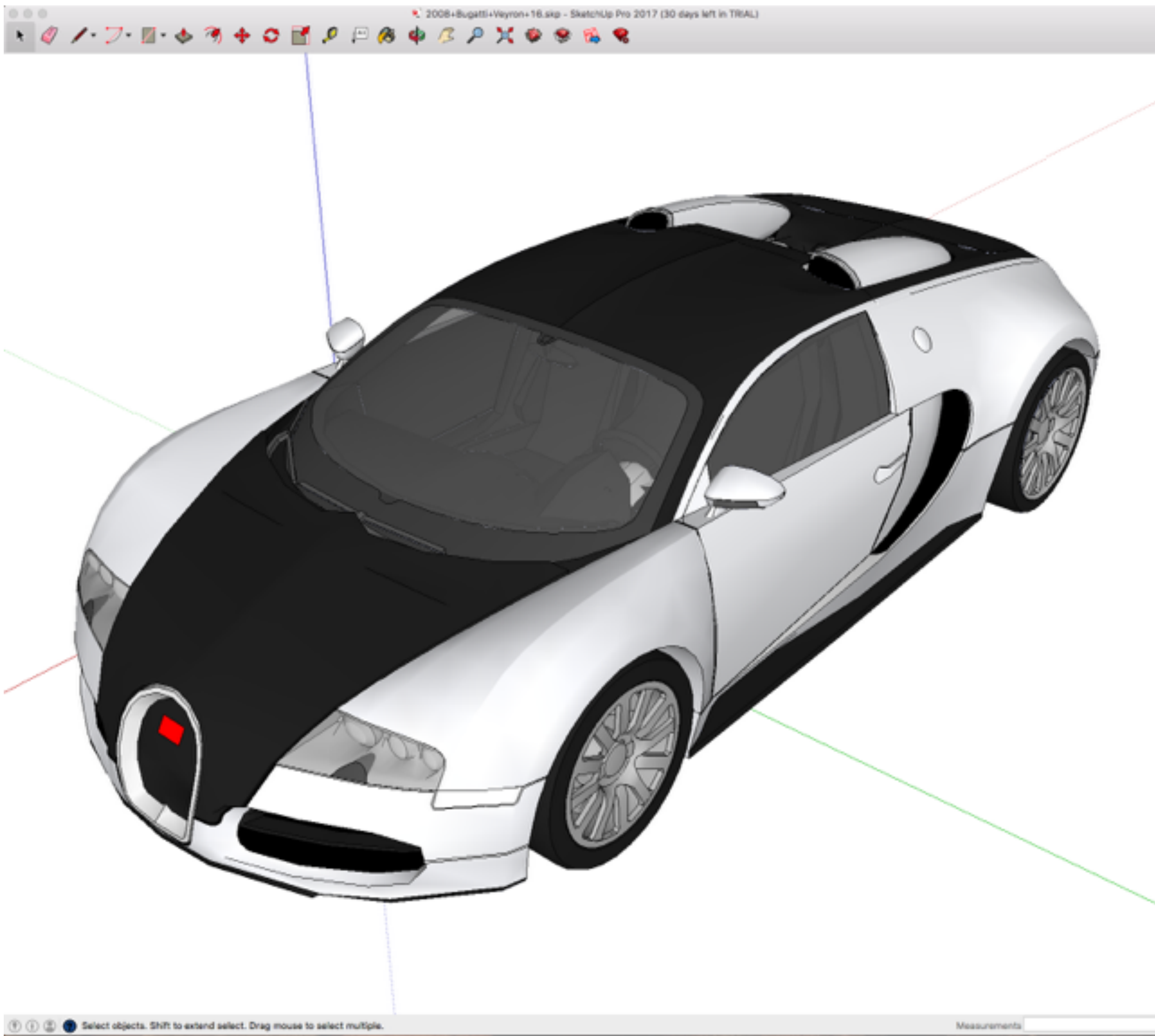
- Experts (aka “graduate students”)
- Crowd workers

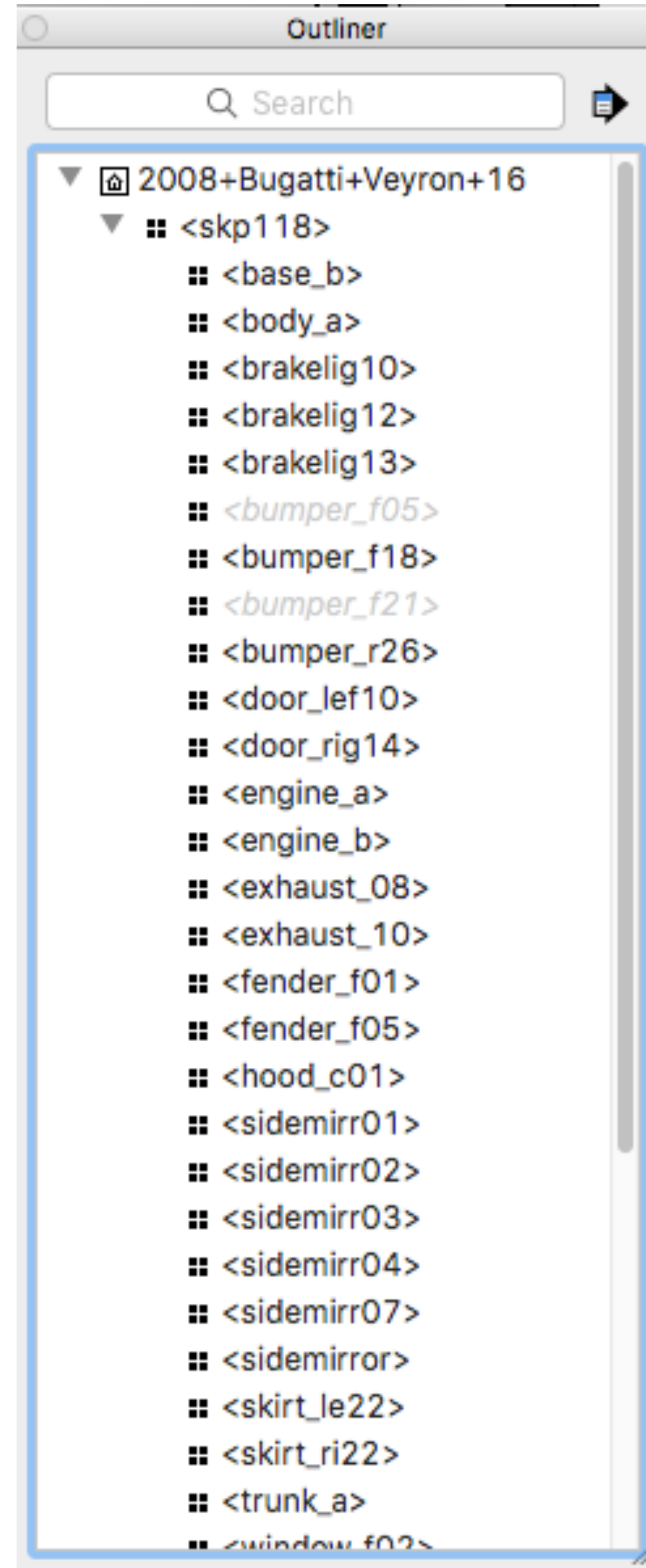
ML algorithms

Some combination of above

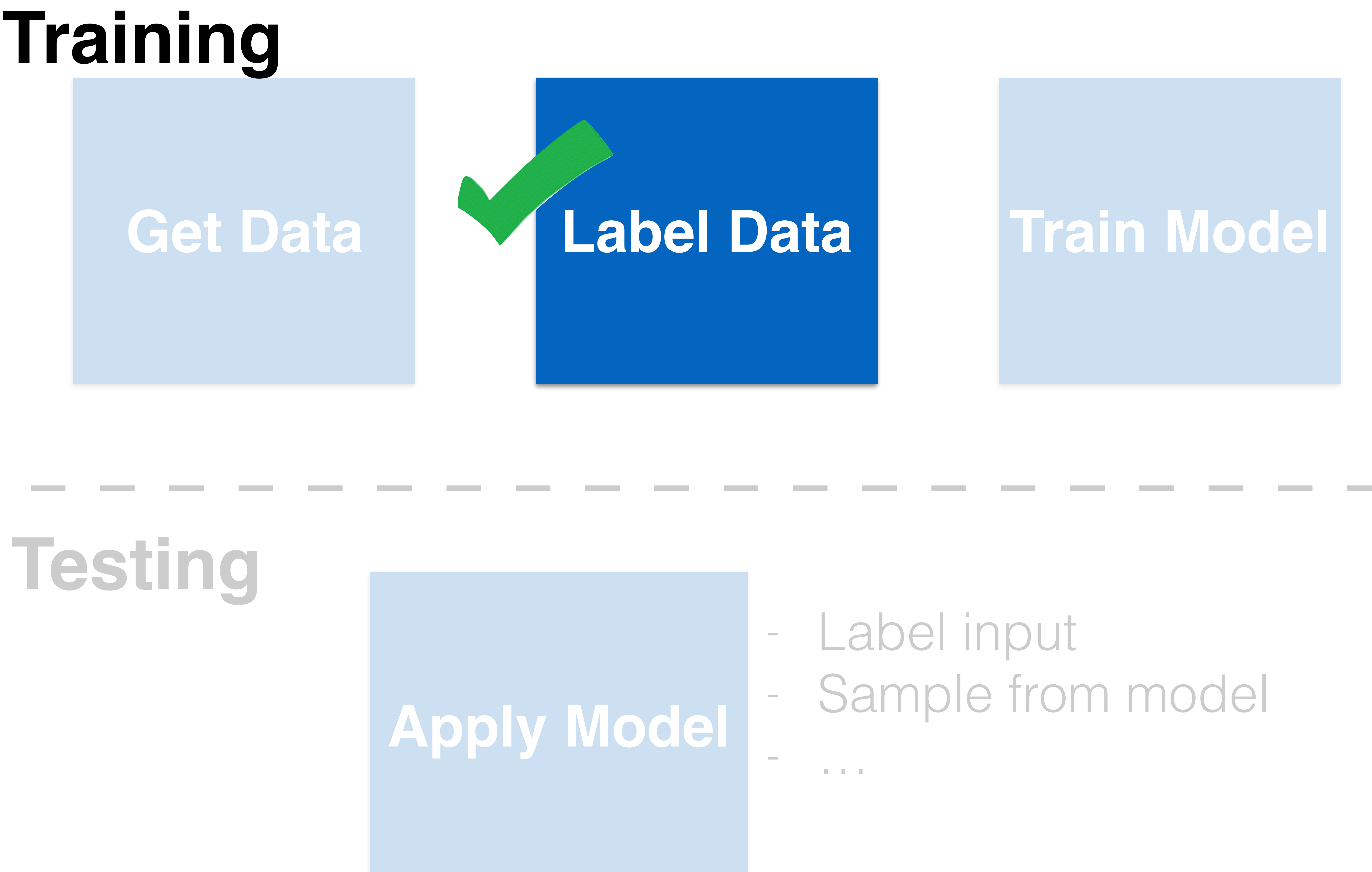
Part Labeling

Somebody provided data for you



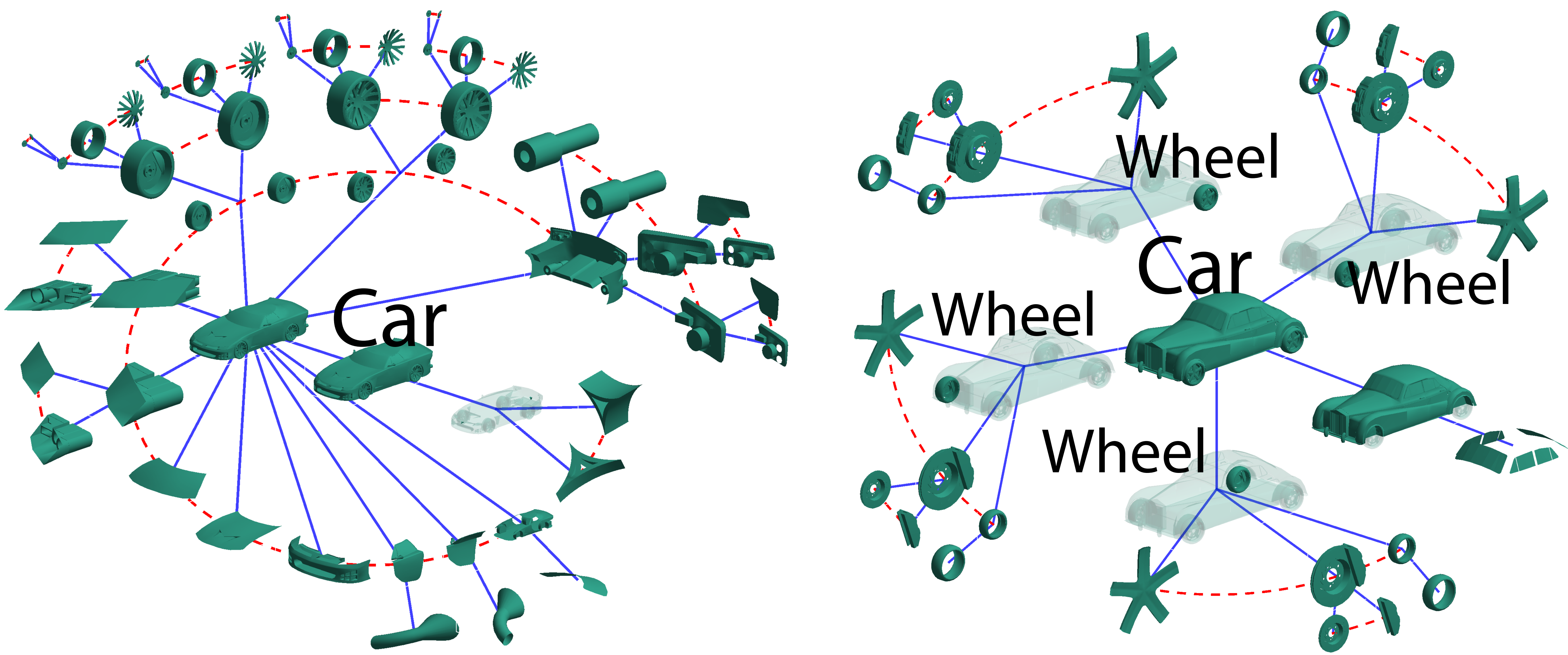


Machine Learning Pipeline



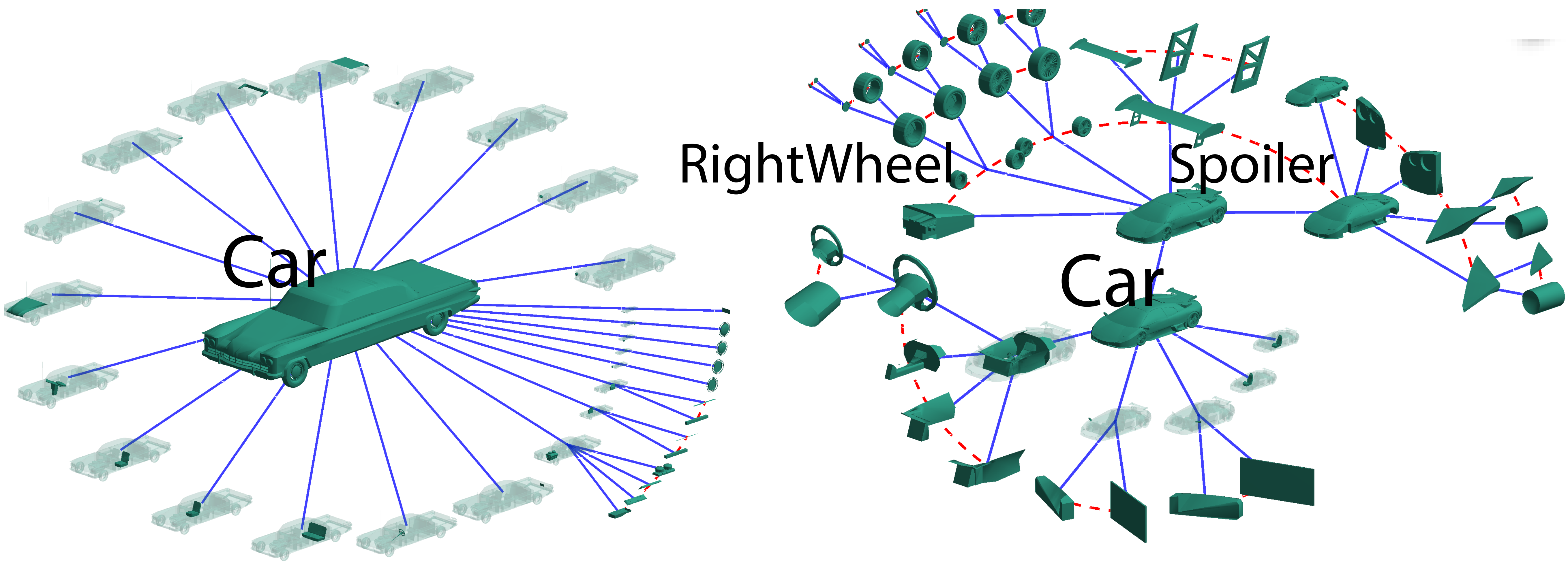
Challenge

Input scene graphs are messy and inconsistent



Scene Graphs

Input data is messy and inconsistent



Our Approach

Training **(robust to inconsistencies)**

Get Data

Label Data

Train Model
(Part-Based Analysis, Mesh Segmentaiton)

Testing

Apply Model
(Mesh Segmentation)

Our Approach

Training **(robust to inconsistencies)**

Get Data

Label Data

Train Model
(Part-Based Analysis, Mesh Segmentaiton)

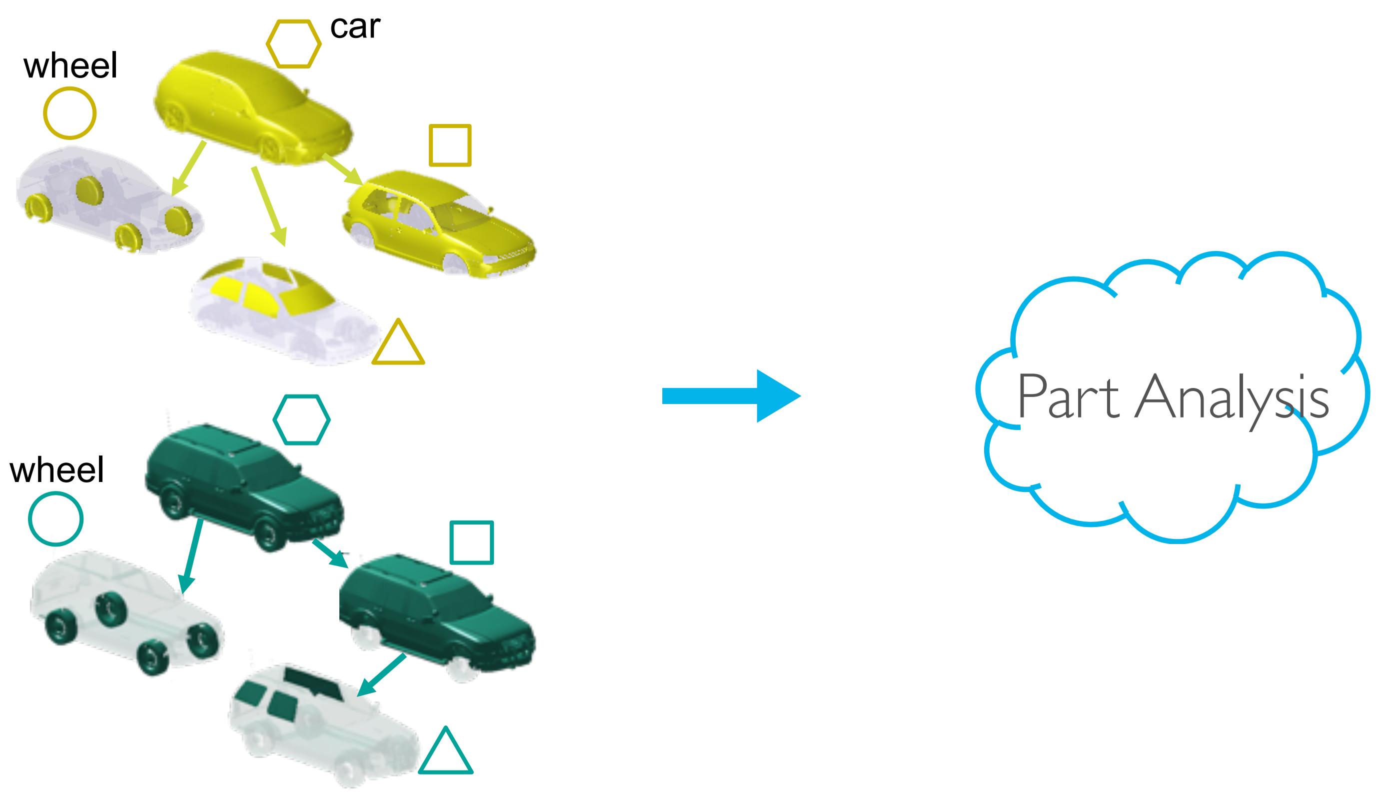
Testing

Apply Model
(Mesh Segmentation)

Learning Hierarchical Shape Segmentation and Labeling from Online Repositories.
Li Yi, Leonidas Guibas, Aaron Hertzmann, Vladimir G. Kim, Hao Su, and Ersin Yumer
SIGGRAPH 2017

Objective Function

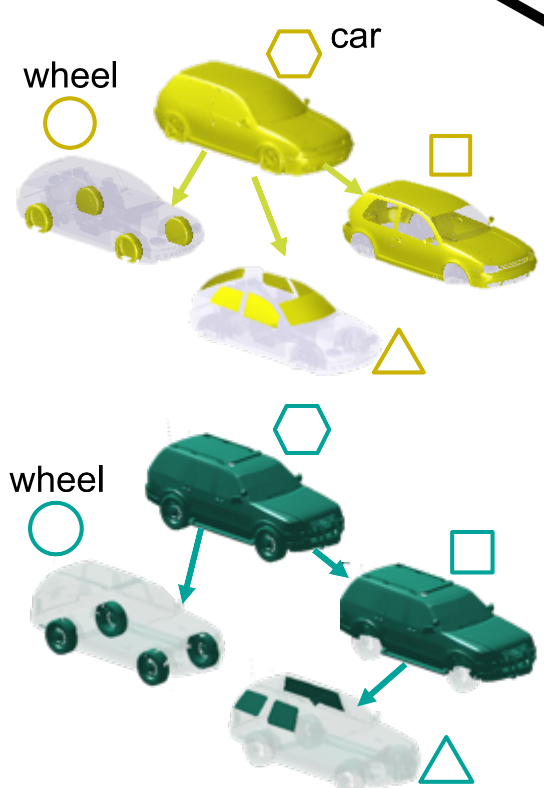
$$E(\theta, p, c, M) = \lambda_c E_c + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

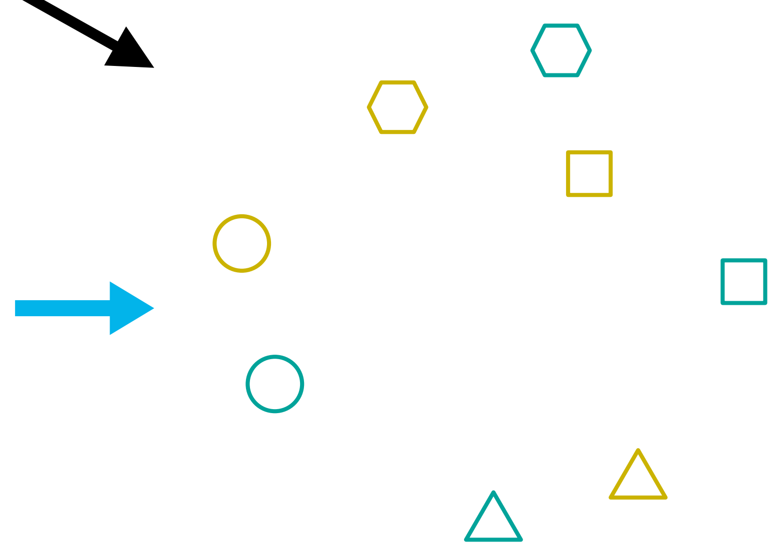


Objective Function

$$E(\theta, p, c, M) = \lambda_c E_c + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

Embedding of every node in a scene graph

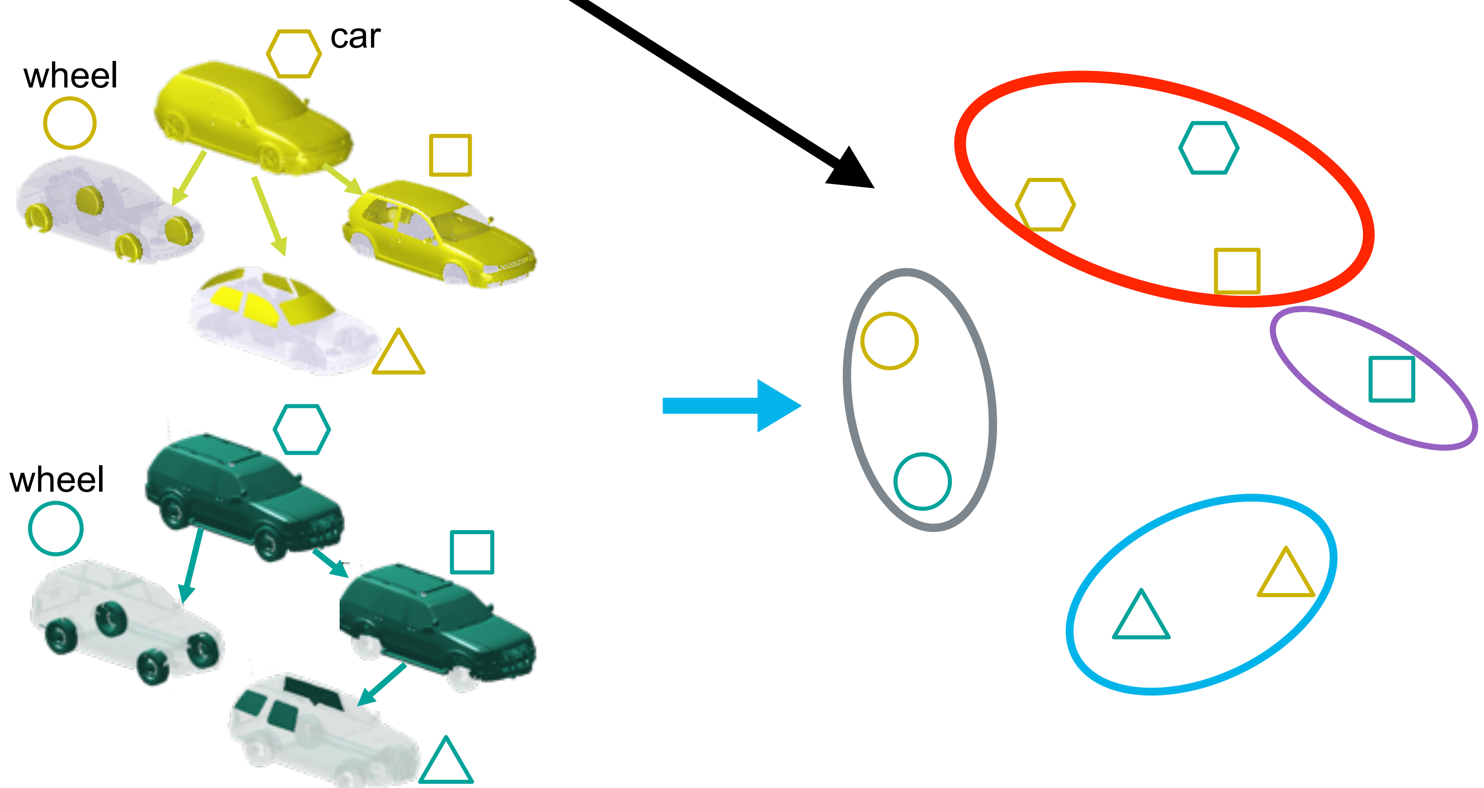




Objective Function

$$E(\theta, p, c, M) = \lambda_c E_c + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

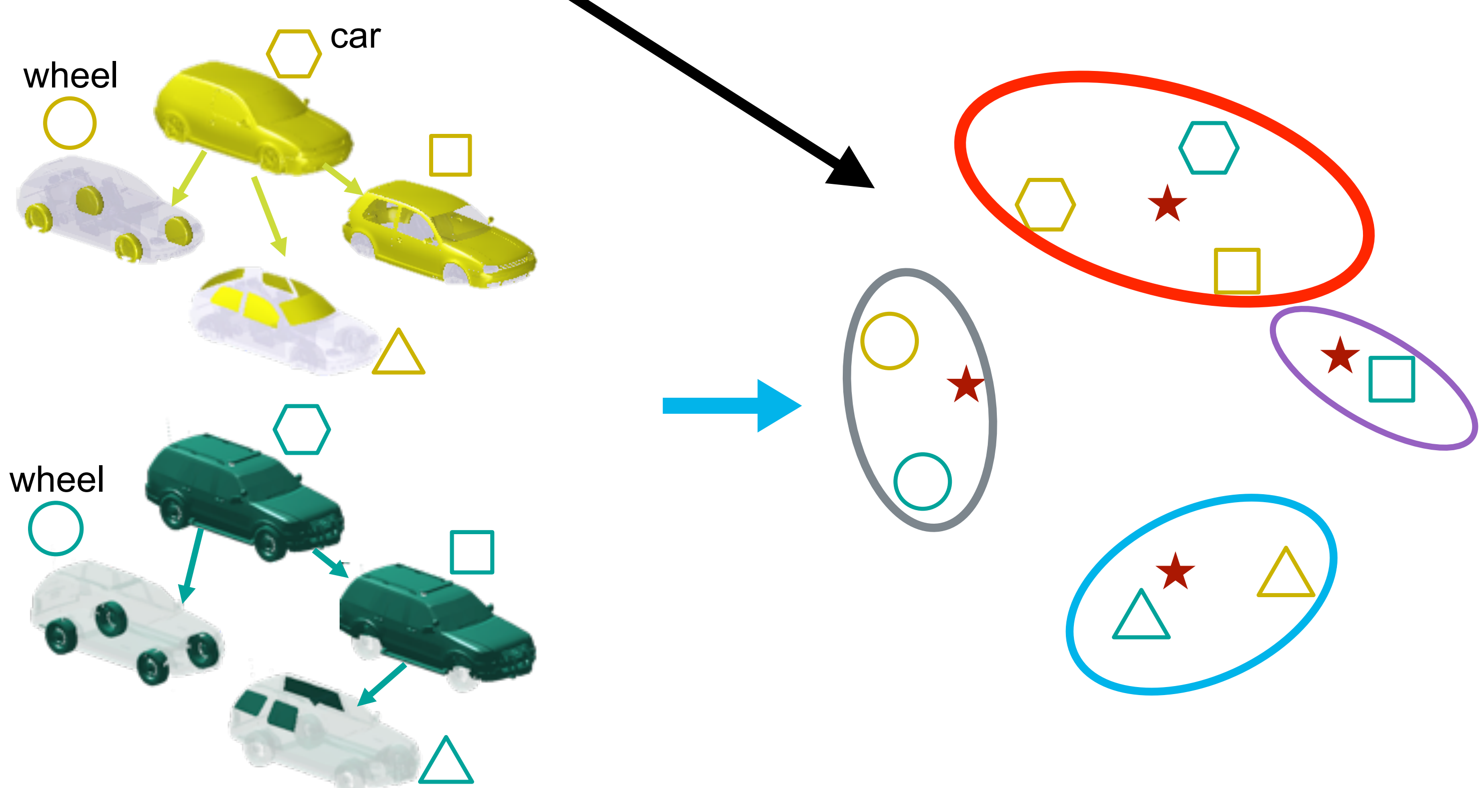
Clustering: labels



Objective Function

$$E(\theta, \underline{p, c}, M) = \lambda_c E_c + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

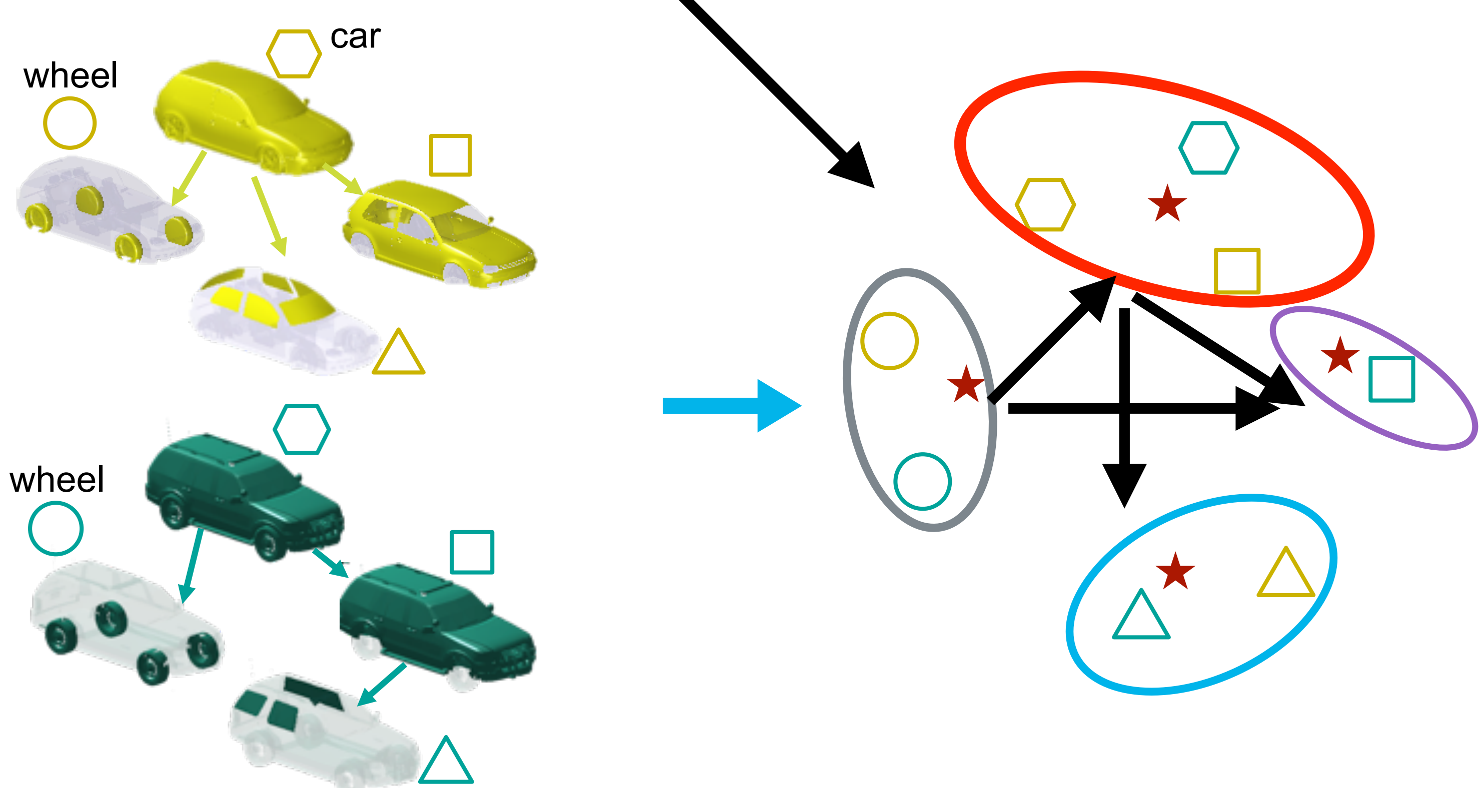
Clustering: labels, centroids



Objective Function

$$E(\theta, p, c, M) = \lambda_c E_c + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

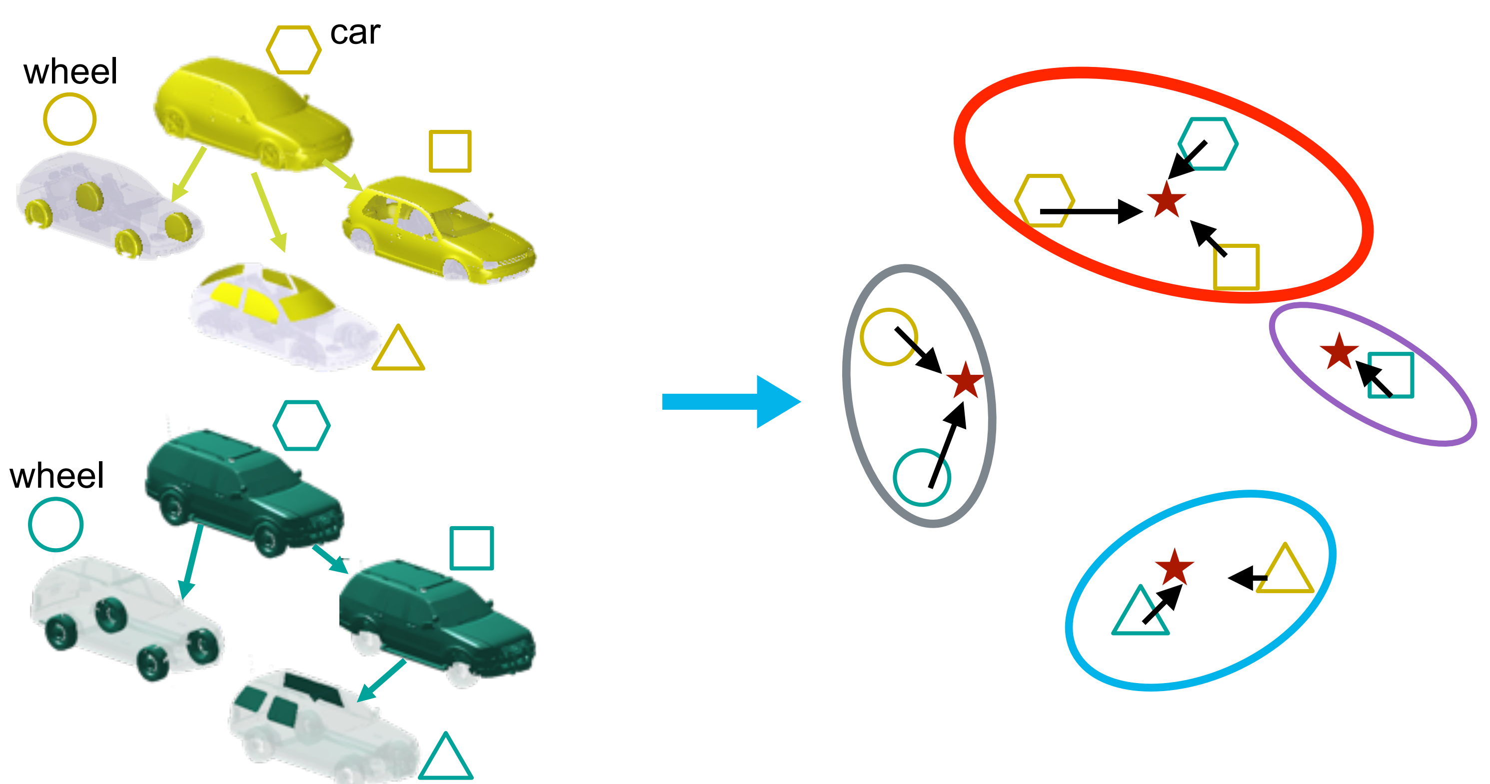
Soft hierarchy graph



Objective Function

$$E(\theta, p, c, M) = \boxed{\lambda_c E_c} + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

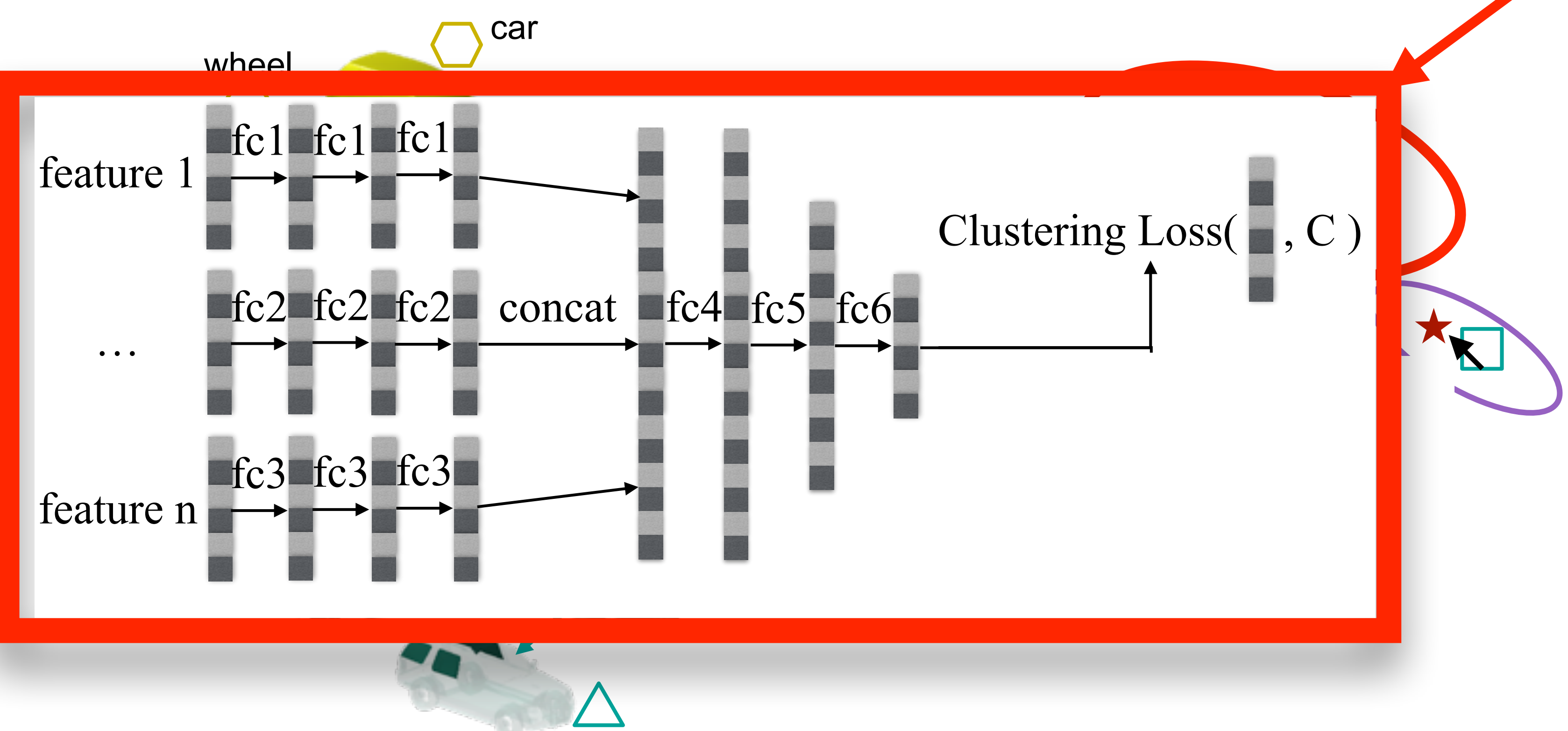
$$E_c = \sum_{\text{parts}} \sum_{\text{centroids}} p_{\text{part,centroid}} ||f(\text{part}) - \text{centroid}||$$



Objective Function

$$E(\theta, p, c, M) = \boxed{\lambda_c E_c} + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

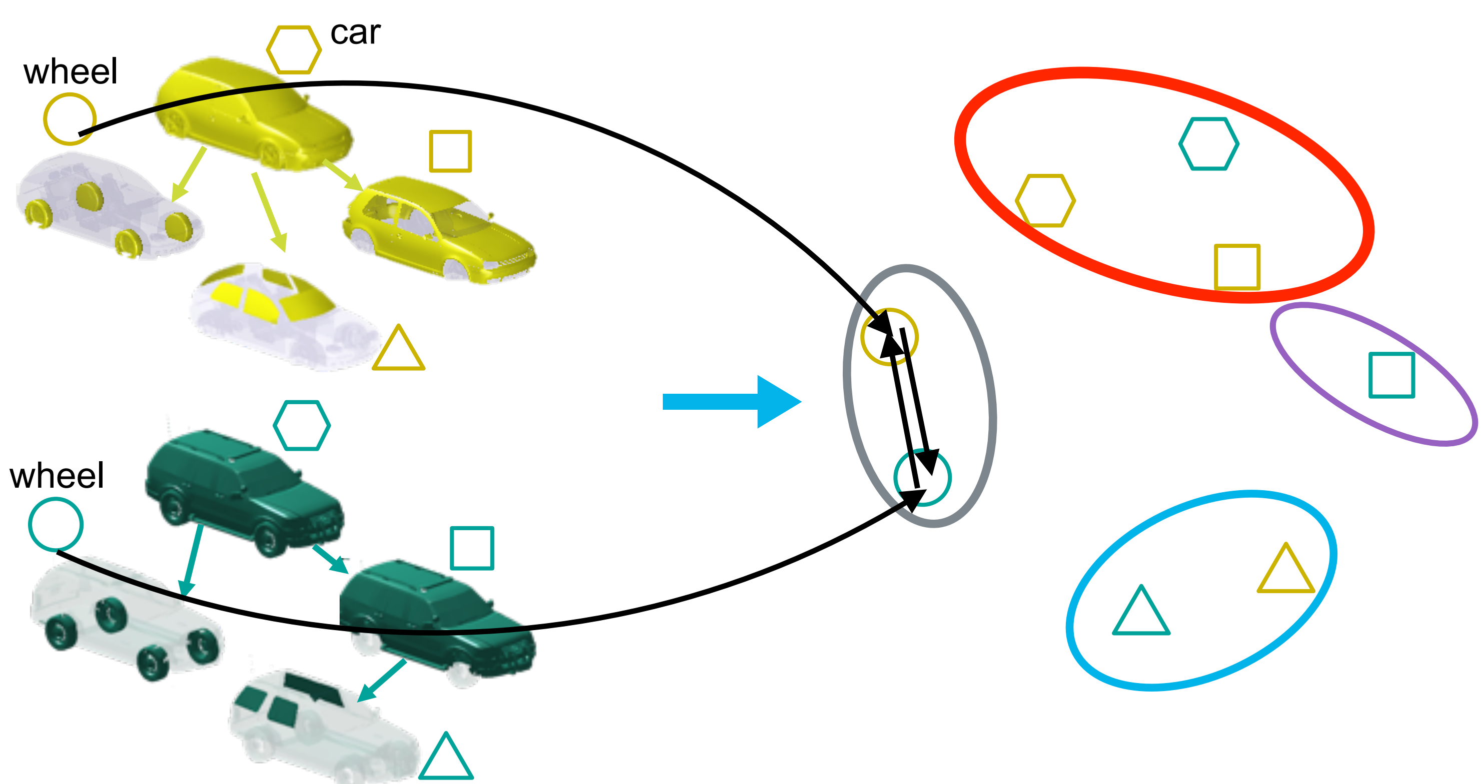
$$E_c = \sum_{\text{parts}} \sum_{\text{centroids}} p_{\text{part,centroid}} || \boxed{f(\text{part})} - \text{centroid} ||$$



Objective Function

$$E(\theta, p, c, M) = \lambda_c E_c + \boxed{\lambda_s E_s} + \lambda_d E_d + \lambda_m E_m - H$$

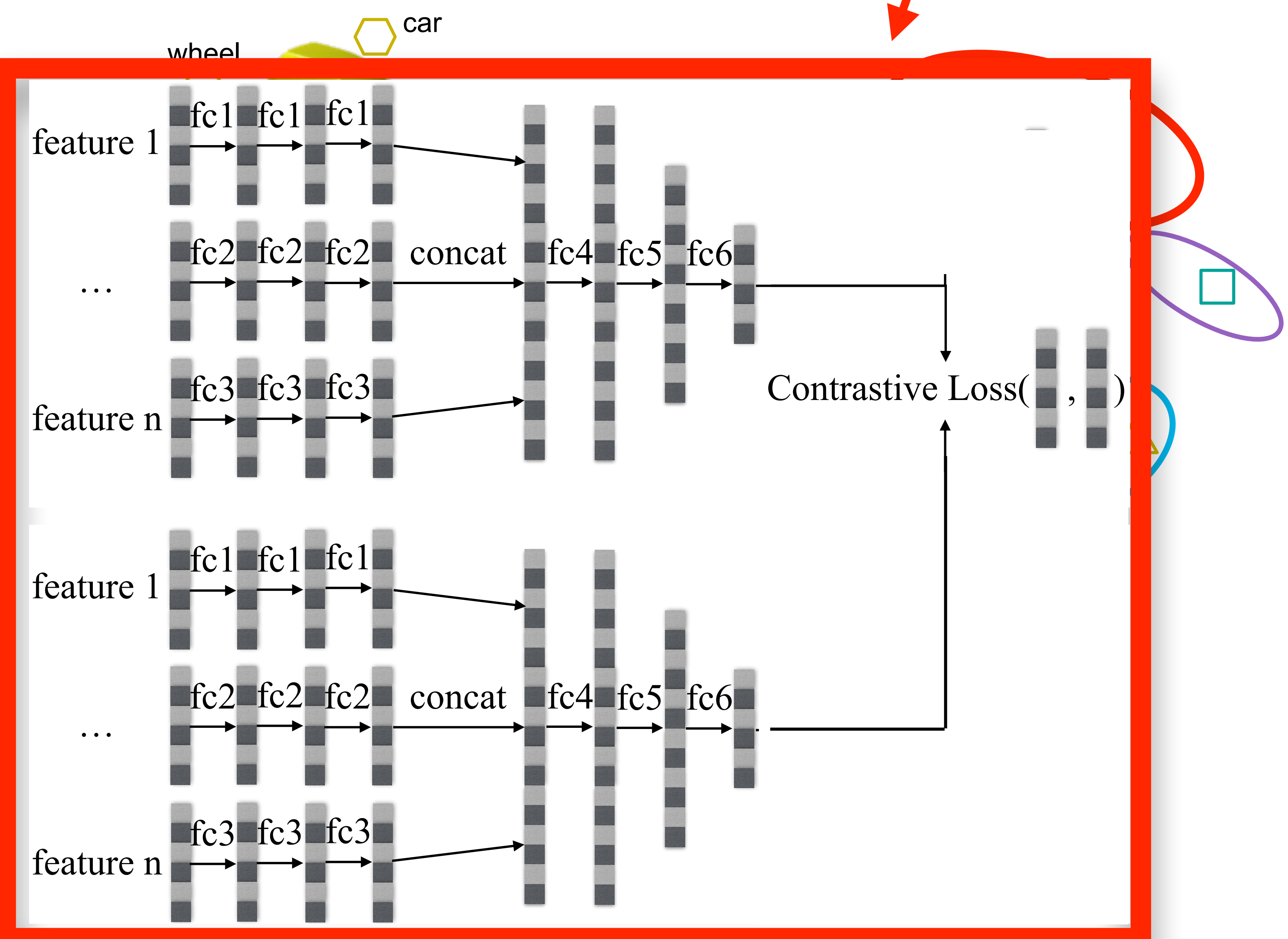
$$E_s = \sum_{\text{part 1}} \sum_{\text{part 2}} ||f(\text{part 1}) - f(\text{part 2})||$$ iff almost identical or tags are the same



Objective Function

$$E(\theta, p, c, M) = \lambda_c E_c + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

$$E_s = \sum_{\text{part 1}} \sum_{\text{part 2}} \boxed{||f(\text{part 1}) - f(\text{part 2})||}$$ iff almost identical or tags are the same

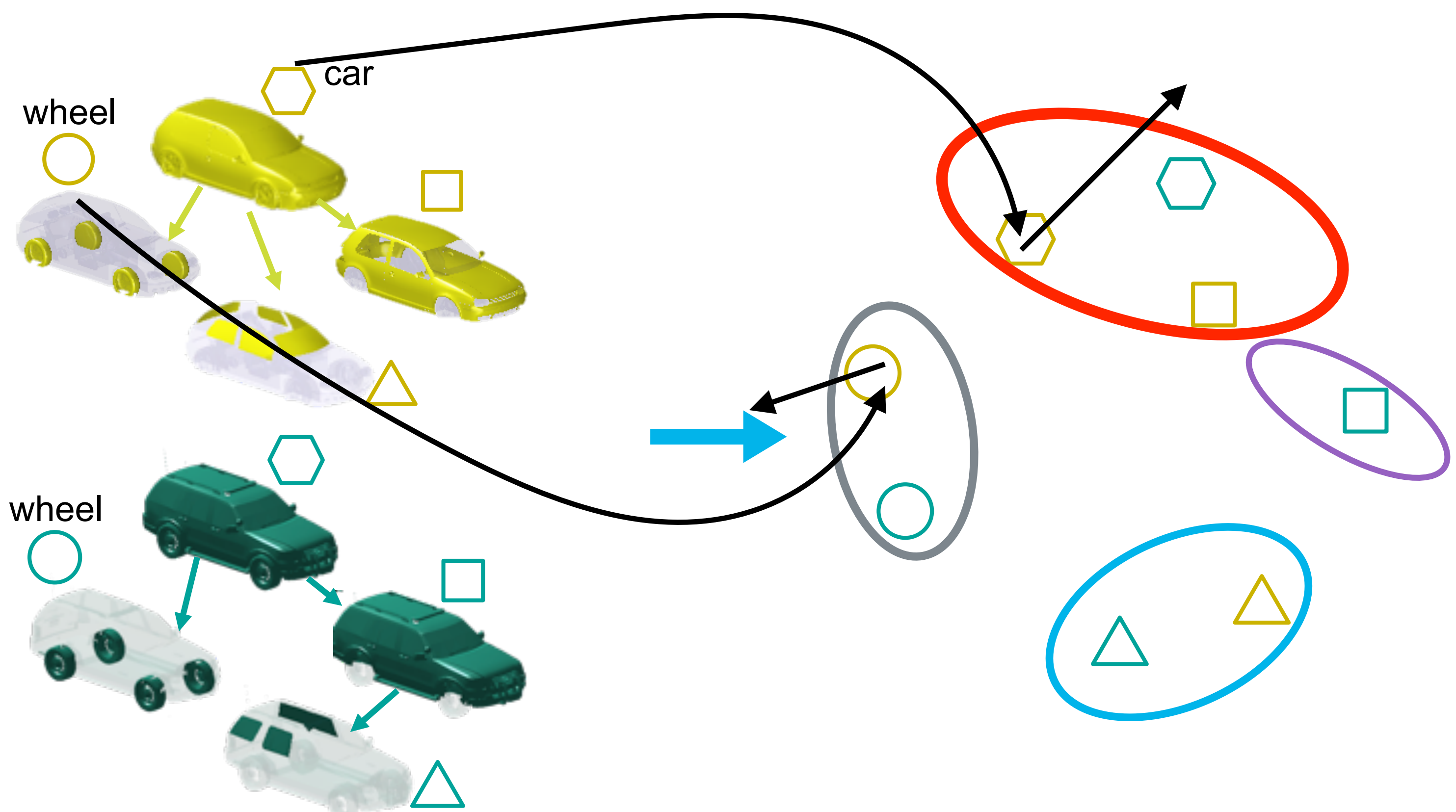


Objective Function

$$E(\theta, p, c, M) = \lambda_c E_c + \lambda_s E_s + \boxed{\lambda_d E_d} + \lambda_m E_m - H$$

$$E_d = \sum_{\text{part 1}} \sum_{\text{part 2}} \max(0, \sigma - ||f(\text{part 1}) - f(\text{part 2})||$$

iff on the same shape
but tags are different

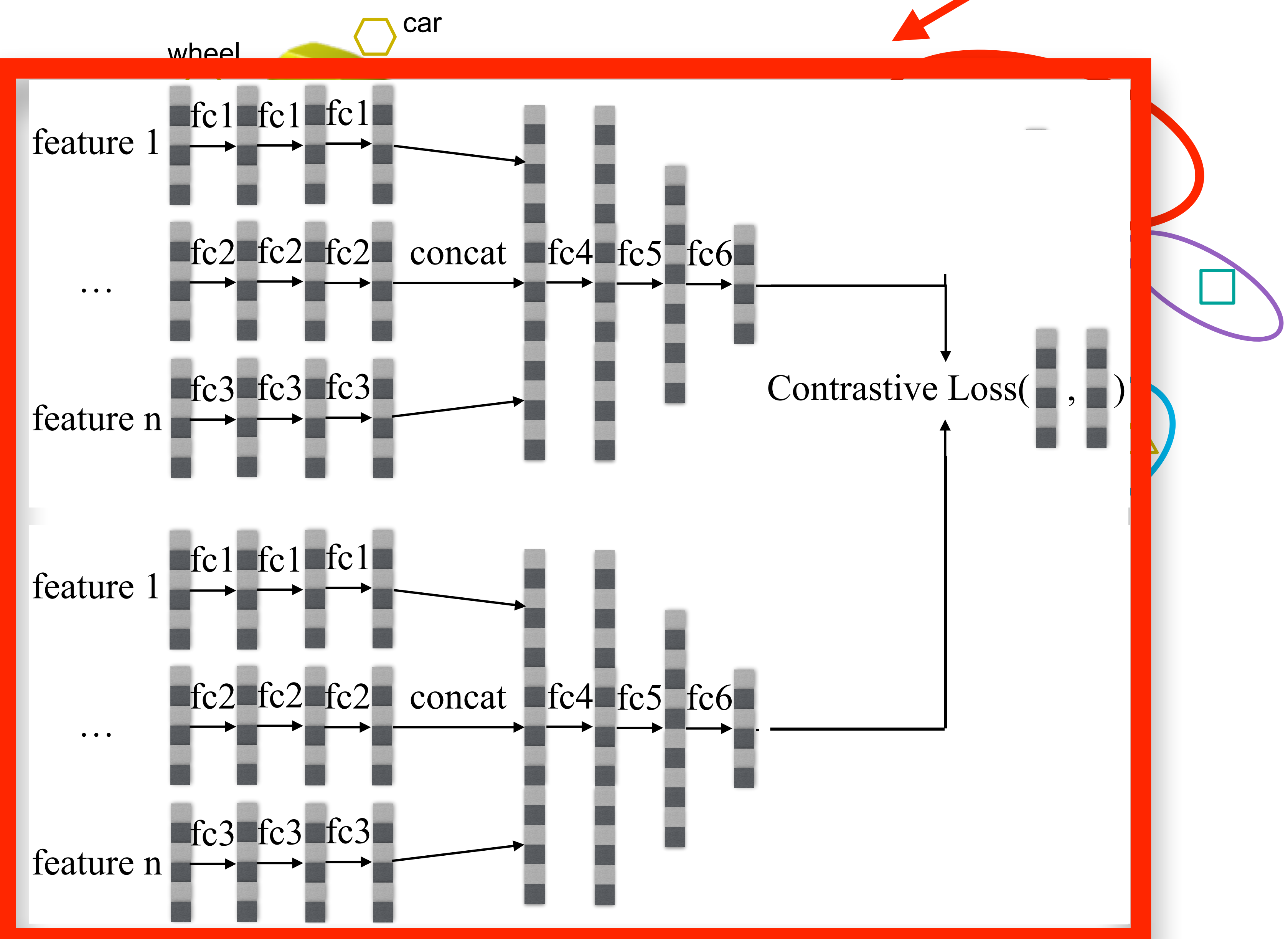


Objective Function

$$E(\theta, p, c, M) = \lambda_c E_c + \lambda_s E_s + \lambda_d E_d + \lambda_m E_m - H$$

$$E_d = \sum_{\text{part 1}} \sum_{\text{part 2}} \max(0, \sigma - \boxed{||f(\text{part 1}) - f(\text{part 2})||})$$

iff on the same shape but tags are different



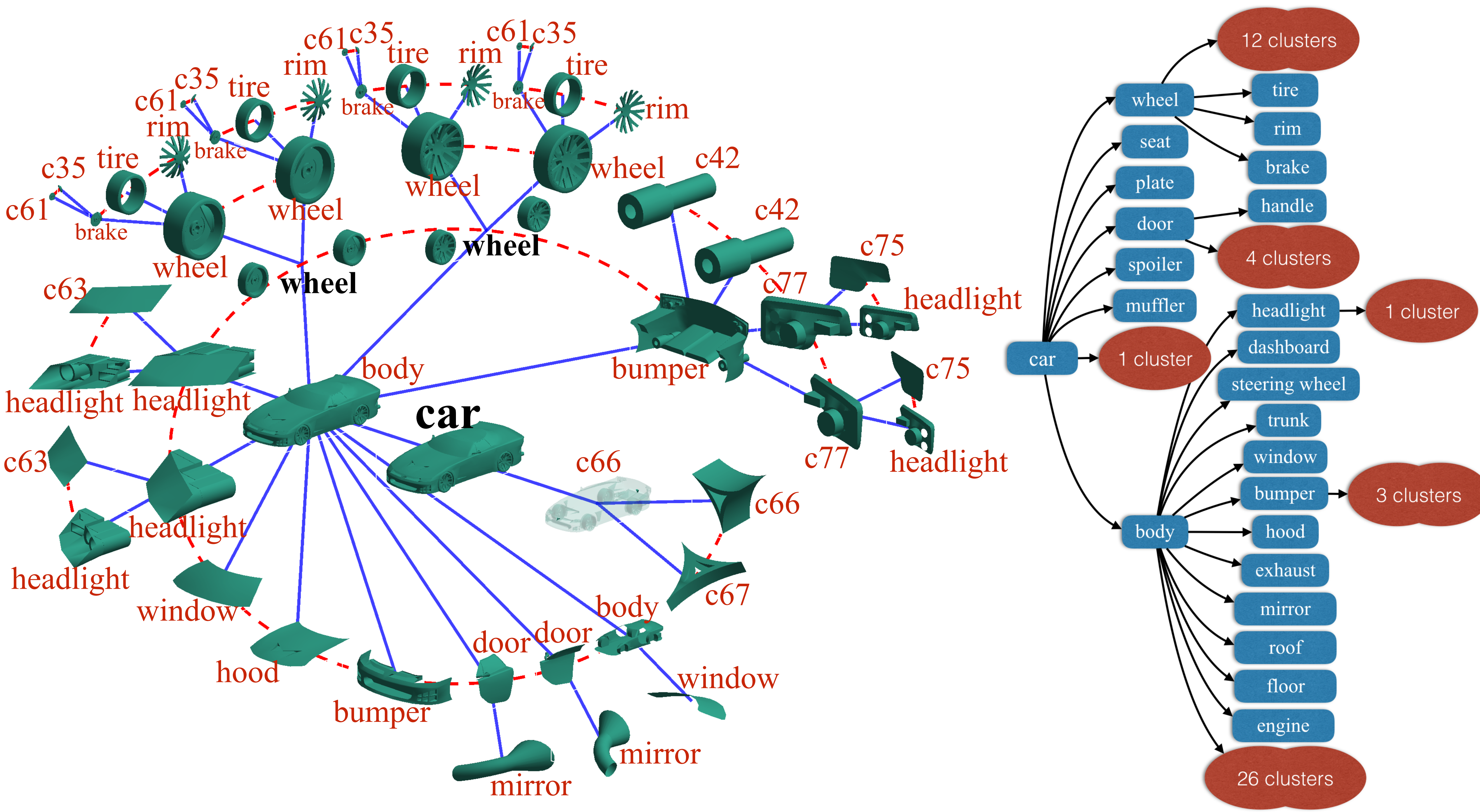
Objective Function

$$E(\theta, p, c, M) = \lambda_c E_c + \lambda_s E_s + \lambda_d E_d + \boxed{\lambda_m E_m - H}$$

Parent-child relationships & regularization

EM optimization:
E step: optimize for p
M step: optimize for θ, c, M

Part-based Analysis Results

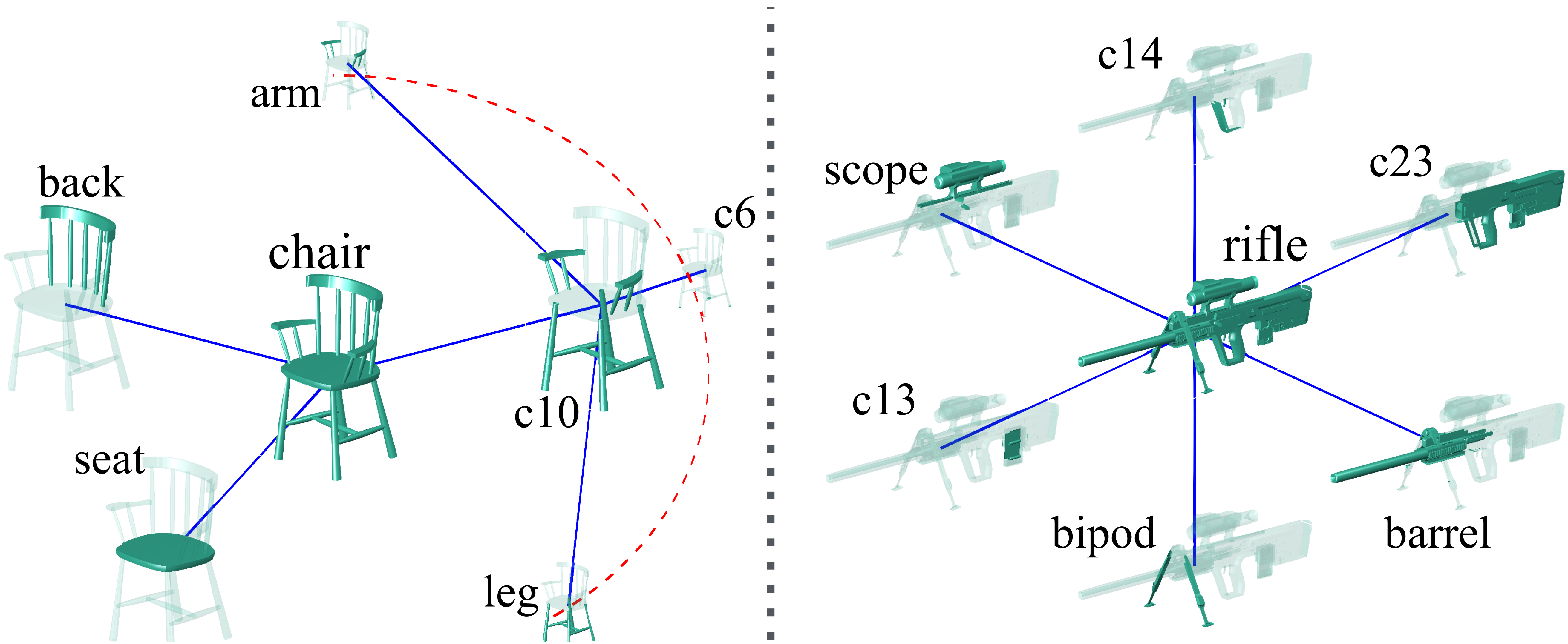


Hierarchical Mesh Segmentation

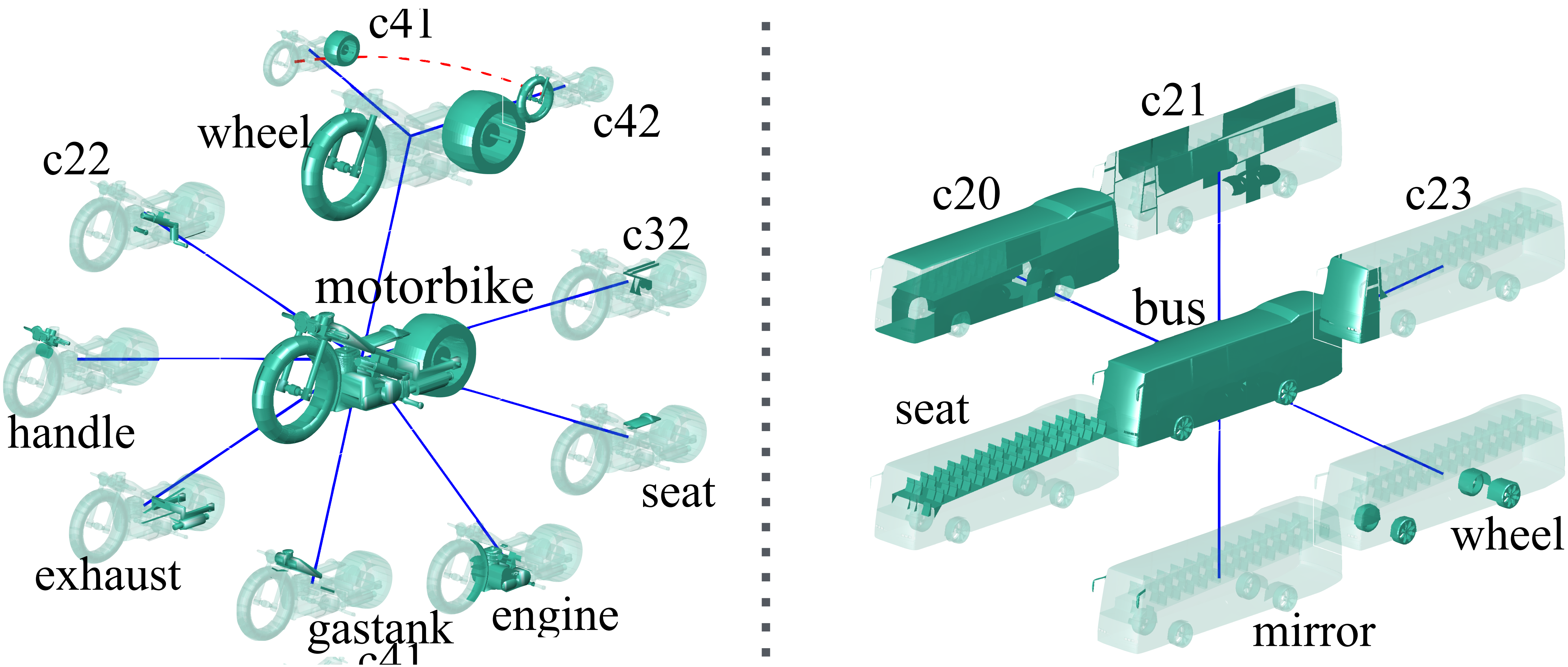
What if part was not in a separate node?

- NN to predict per-face labels (use part-base analysis tags)
- Preserve parent-child relationships
- MRF to get smooth boundaries
- Use connected components if available

Results


arm
back
chair
c10
c6
seat
leg
c14
scope
c23
rifle
c13
bipod
barrel

Results


c41
wheel
c42
c22
motorbike
c32
handle
seat
exhaust
gastank
engine
c21
c20
c23
bus
seat
wheel
mirror

Evaluation

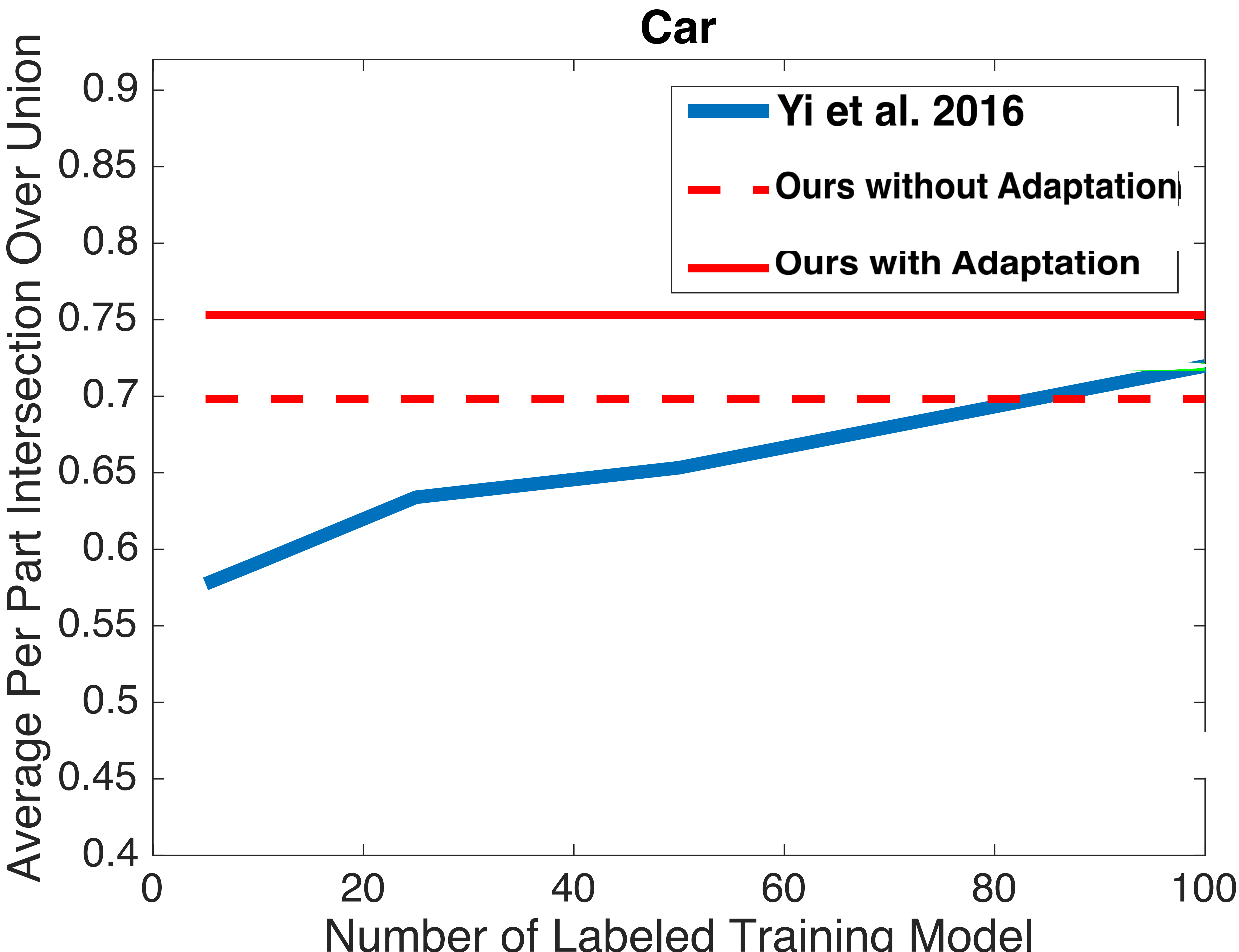
Clustering (Normalized Mutual Information ≤ 1)

- Chance = 0.034
- Features = 0.348
- Ours = 0.573

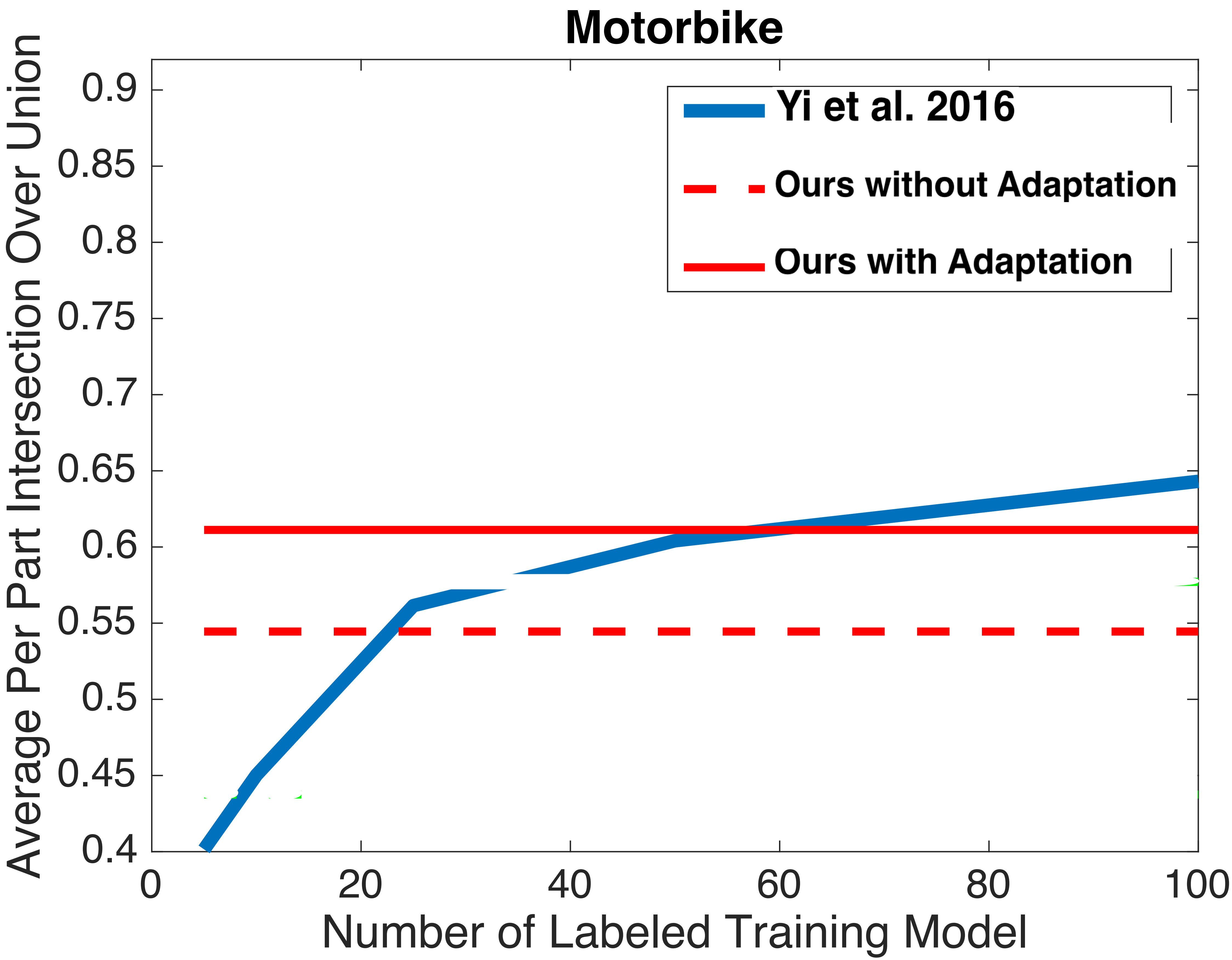
Part Tagging (accuracy ≤ 1)

- Chance = 0.139
- Features = 0.823
- Ours = 0.910

Crowd vs Ambient Data

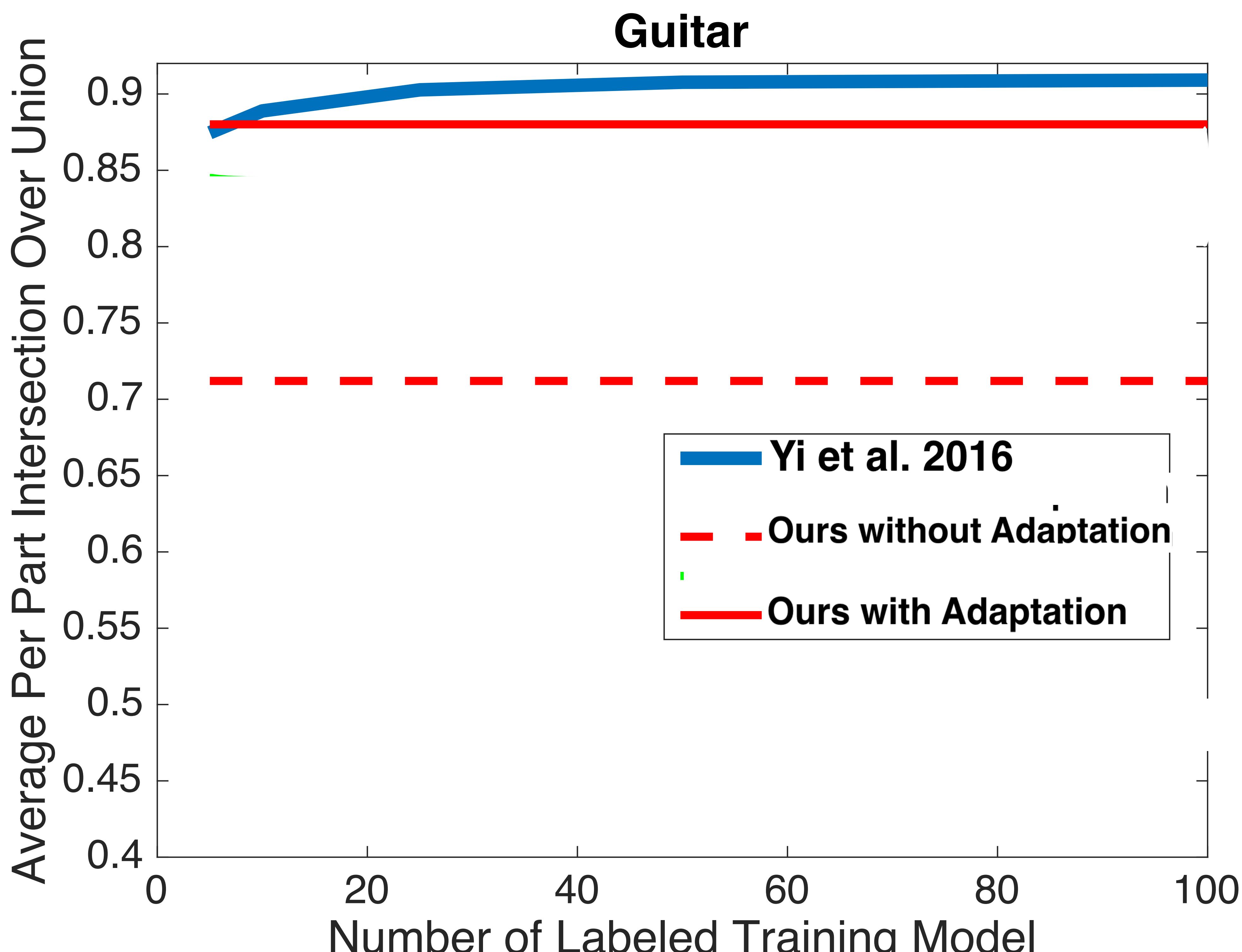


Crowd vs Ambient Data



Crowd vs Ambient Data

Not all ambient data is equally useful



Use Case

Fe PROJECT FELIX

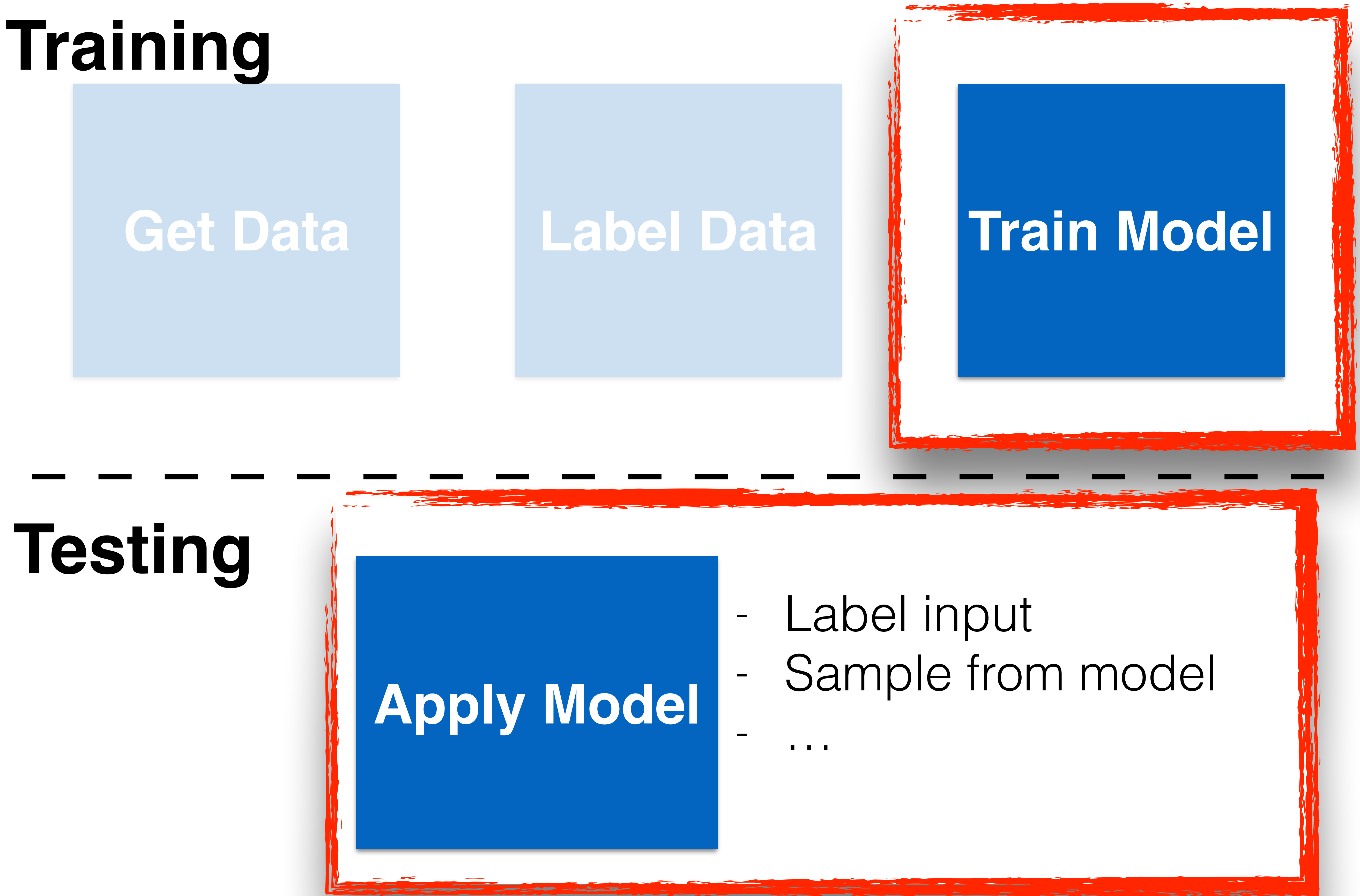
Summary

Data is as important as algorithms

Make data collection easier

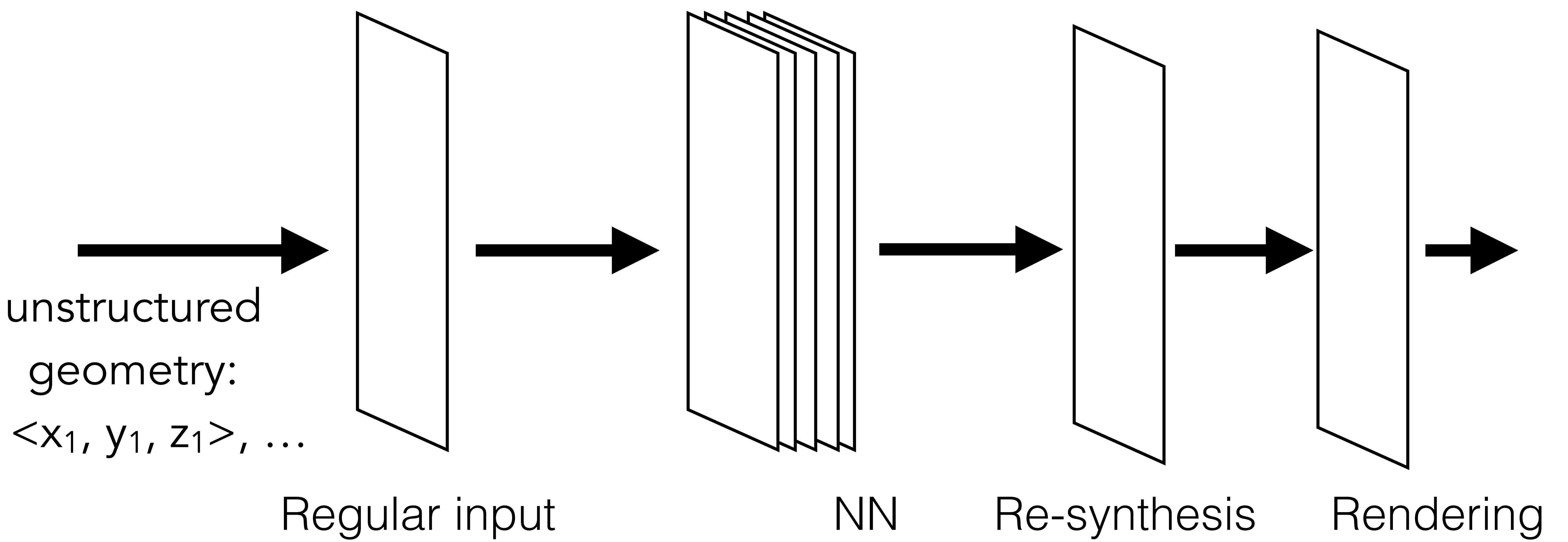
- Use all available meta-data
- Crowd-source as necessary
- Automatically propagate and then manually verify labels

Future Work


Training
Get Data
Label Data
Train Model
Testing
Apply Model
- Label input
- Sample from model
- …

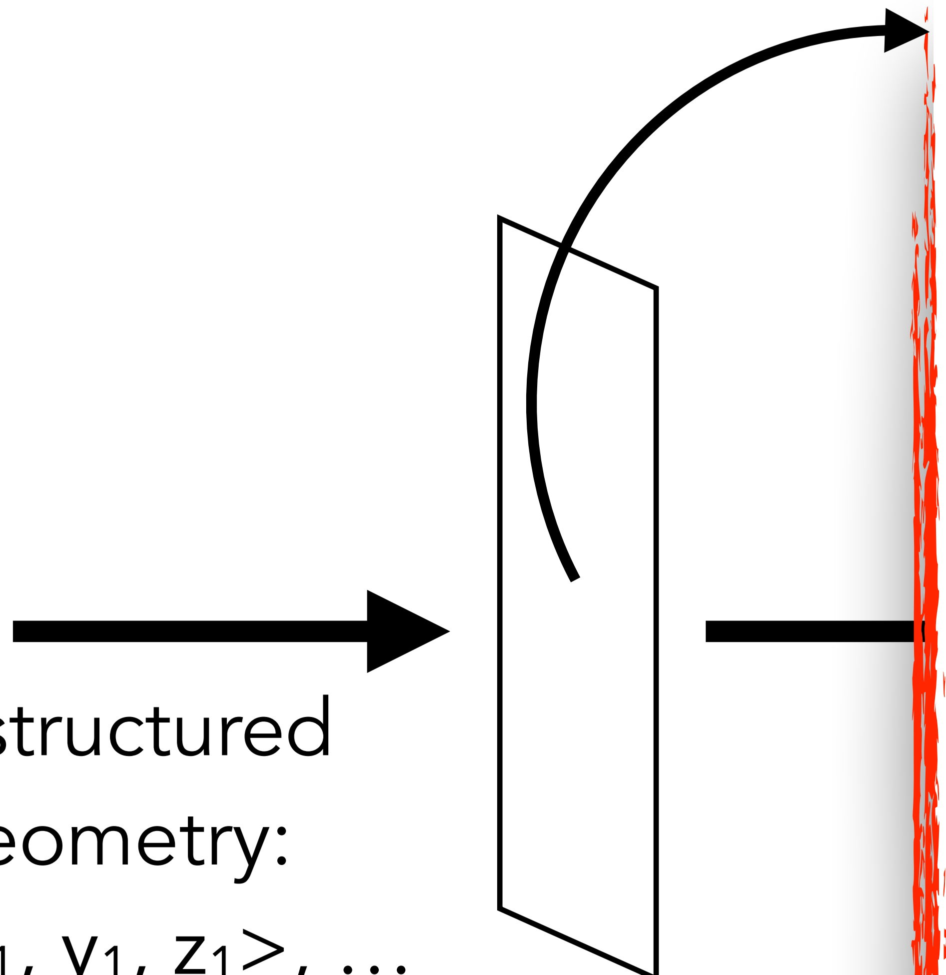
Deep Learning for Geometry

Geometry processing as neural network layers



Deep Learning for Geometry

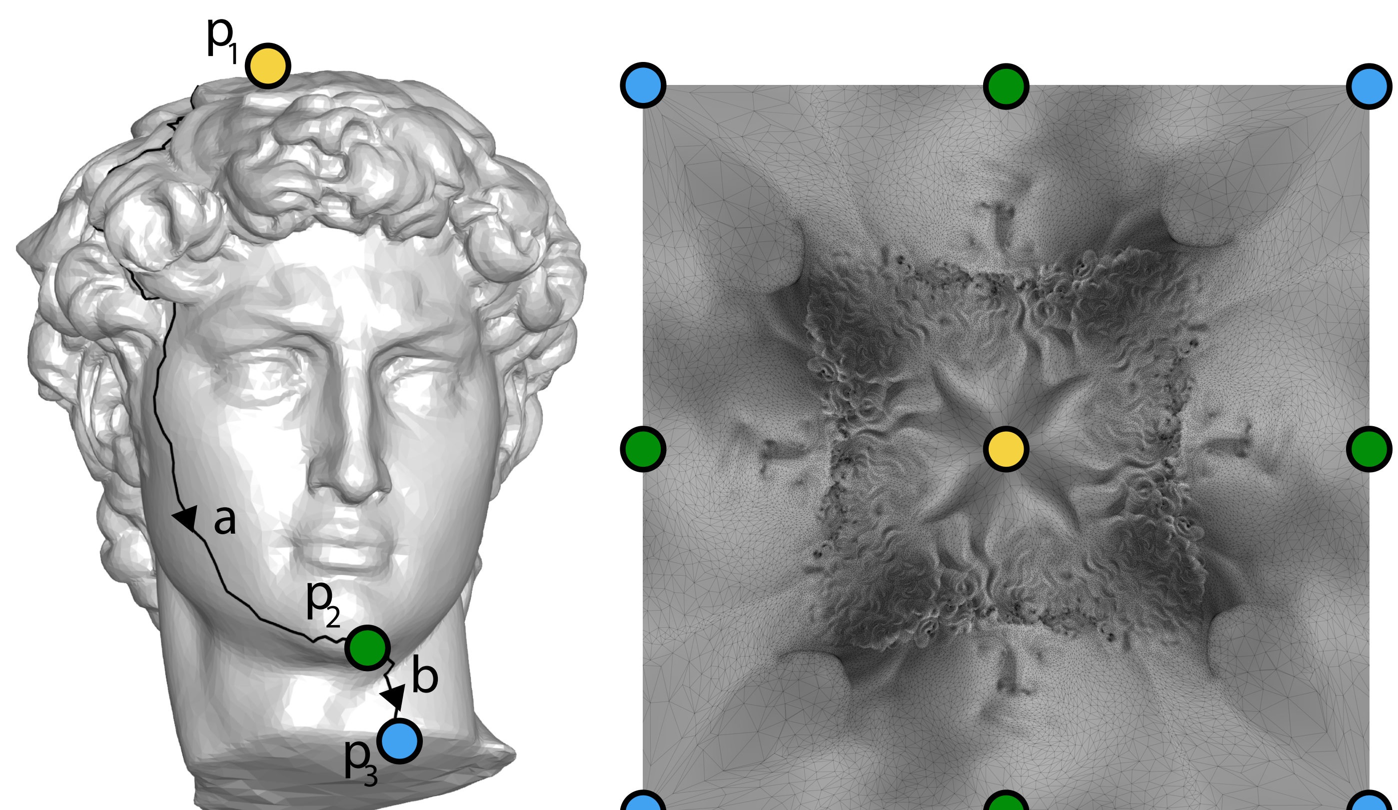
Geometry processing as neural network layers



unstructured geometry: $<x_1, y_1, z_1>$, …

Regular input

- Make input more regular:
Parameterization
(translation-invariant, seamless)



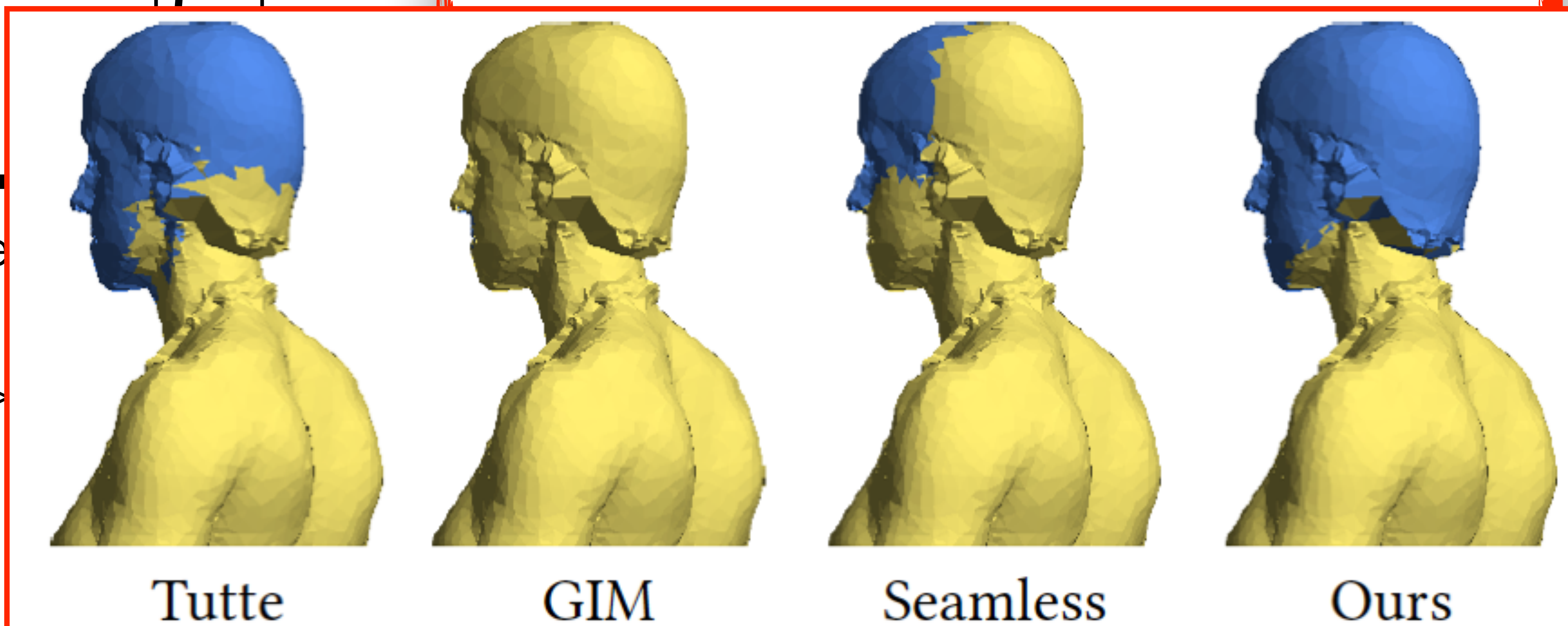
Convolutional Neural Networks on Surfaces via Seamless Toric Covers. Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G. Kim, and Yaron Lipman. SIGGRAPH 2017

Deep Learning for Geometry

Geometry processing as neural network layers

- Make input more regular:
Parameterization
(translation-invariant, seamless)

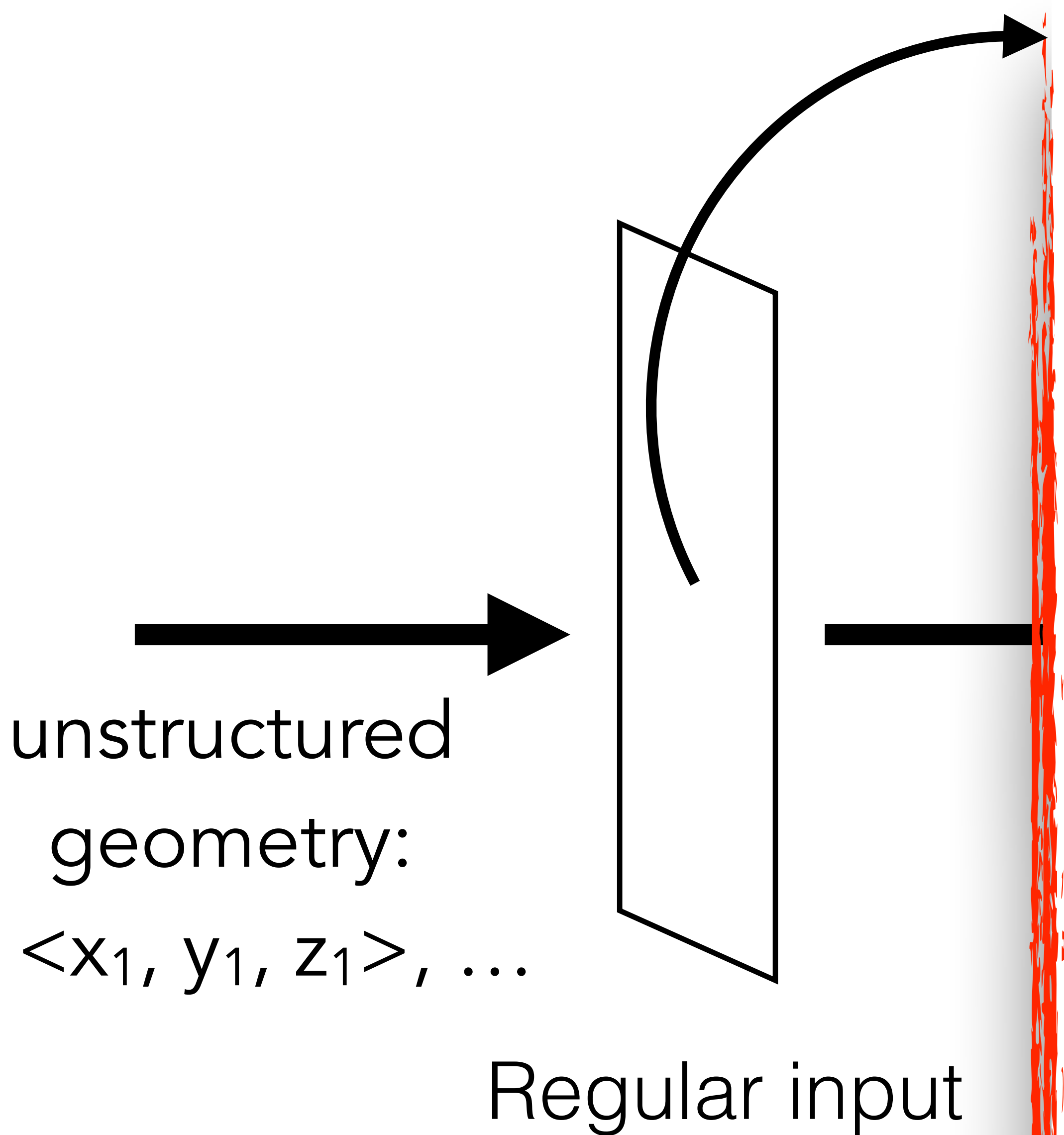
unstructure
geometry
$<x_1, y_1, z_1>$



Convolutional Neural Networks on Surfaces via Seamless Toric Covers. Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G. Kim, and Yaron Lipman. SIGGRAPH 2017

Deep Learning for Geometry

Geometry processing as neural network layers



- Make input more regular:
Parameterization

- Make input more consistent:
Correspondence

- Learn how to parameterize and optimize for consistency
Parametric Layer

Convolutional Neural Networks on Surfaces via Seamless Toric Covers. Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G. Kim, and Yaron Lipman. SIGGRAPH 2017

Deep Learning for Geometry

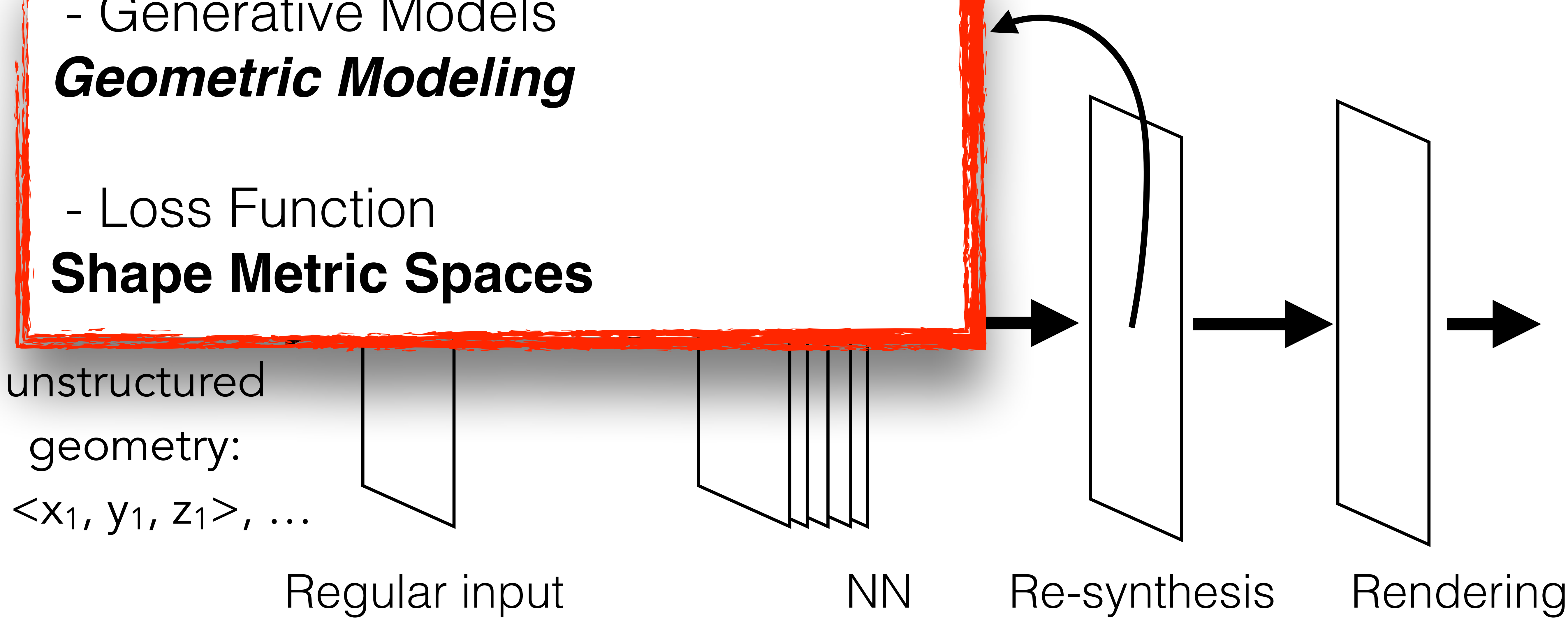
Geometry processing as neural network layers

- Generative Models

Geometric Modeling

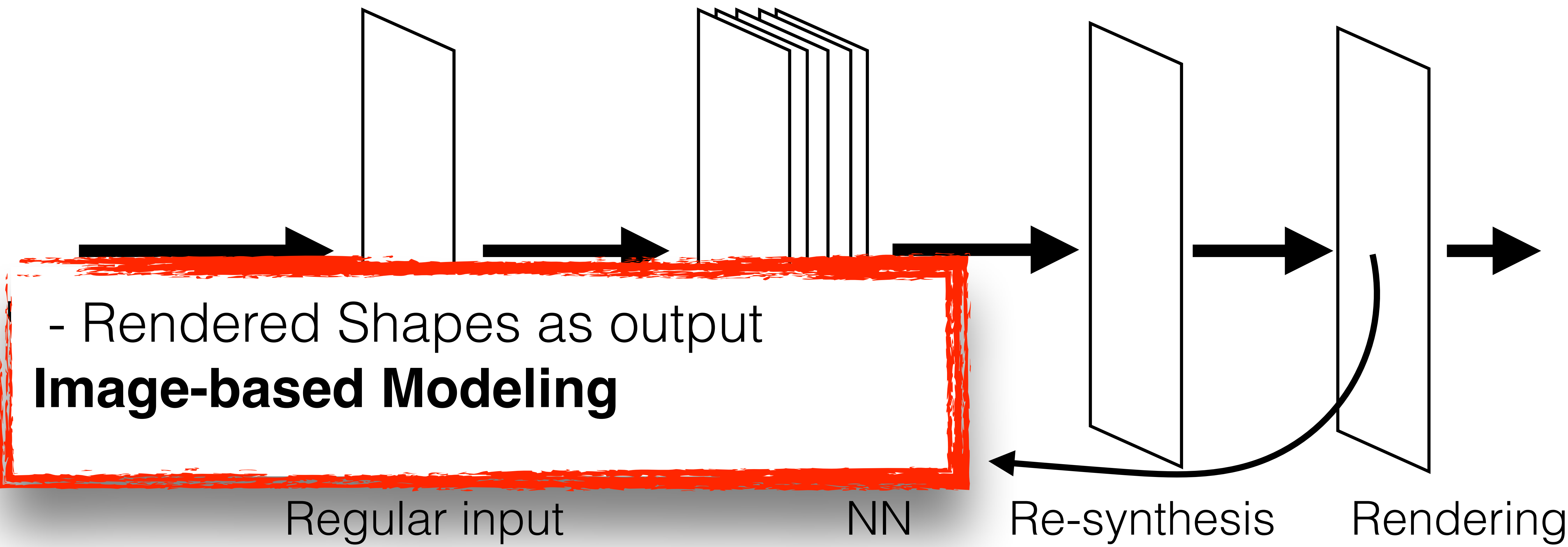
- Loss Function

Shape Metric Spaces



Deep Learning for Geometry

Geometry processing as neural network layers

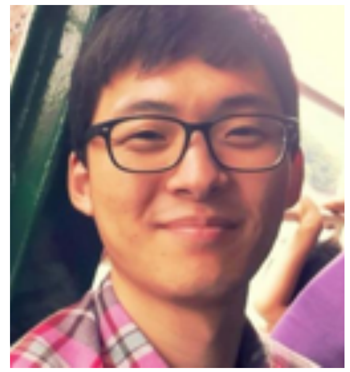


Acknowledgements

Two main papers:

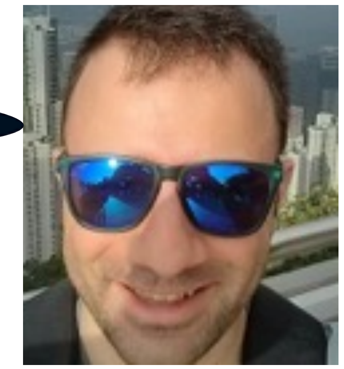
Li Yi

Applications:



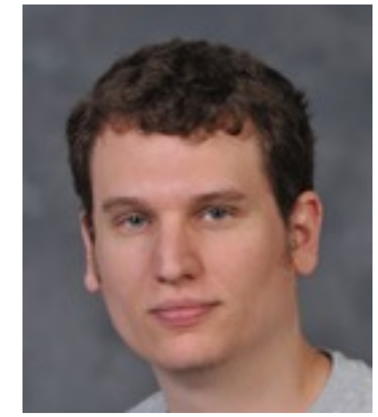
Minhyuk Sung

Aric Bartle



Melinos Averkiou

Future Work:



Haggai Maron

Danielle Ezuz

Other contributors (alphabetical)

Noam Aigerman
Roland Angst
Floraine Berthouzoz
Duygu Ceylan
Stephen DiVerdi
Nadav Dym
Thomas Funkhouser
Leonidas Guibas

Meirav Galun
Aaron Hertzmann
Qixing Huang
Danny Kaufman
Yaron Lipman
Wilmot Li
Cewu Lu
Niloy J. Mitra

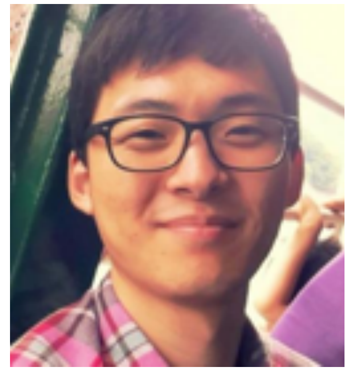
Hao Su
Alla Sheffer
I-Chao Shen
Miri Trope
Nicholas Vining
Mengyan Yan
Ersin Yumer

Acknowledgements

Two main papers:

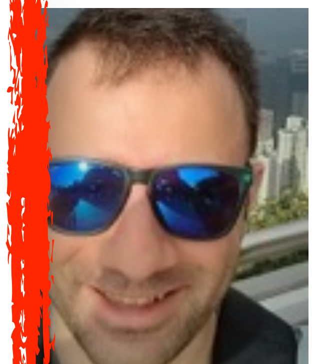
Li Yi

Applications:



Minhyuk Sung

Aric Bartle



Melinos Averkiou

Future Work:

Haggai Maron

Danielle Ezuz

Other contributors (alphabetical)

Interns @ Adobe

Noam Aigerman
Roland Angst
Floraine Berthouzoz
Duygu Ceylan
Stephen DiVerdi
Nadav Dym
Thomas Funkhouser
Leonidas Guibas

Meirav Galun
Aaron Hertzmann
Qixing Huang
Danny Kaufman
Yaron Lipman
Wilmot Li
Cewu Lu
Niloy J. Mitra

Hao Su
Alla Sheffer
I-Chao Shen
Miri Trope
Nicholas Vining
Mengyan Yan
Ersin Yumer

Researchers @ Adobe