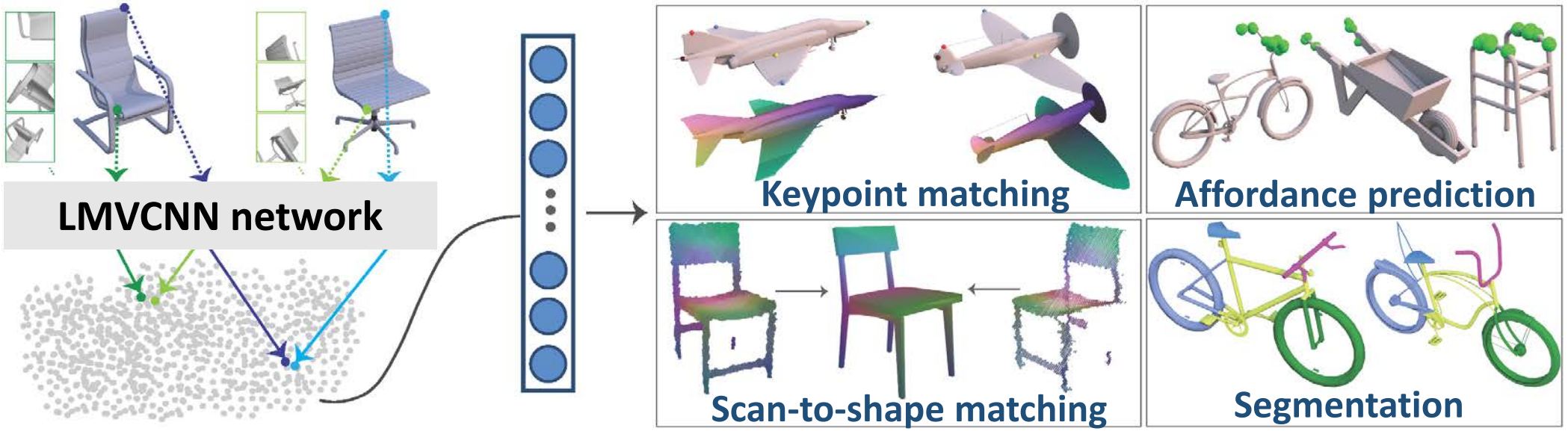
Learning Local Shape Descriptors from Part Correspondences with Multi-view Convolutional Networks



Haibin Huang[1] Evangelos Kalogerakis[1] Siddhartha Chaudhuri[2,3]

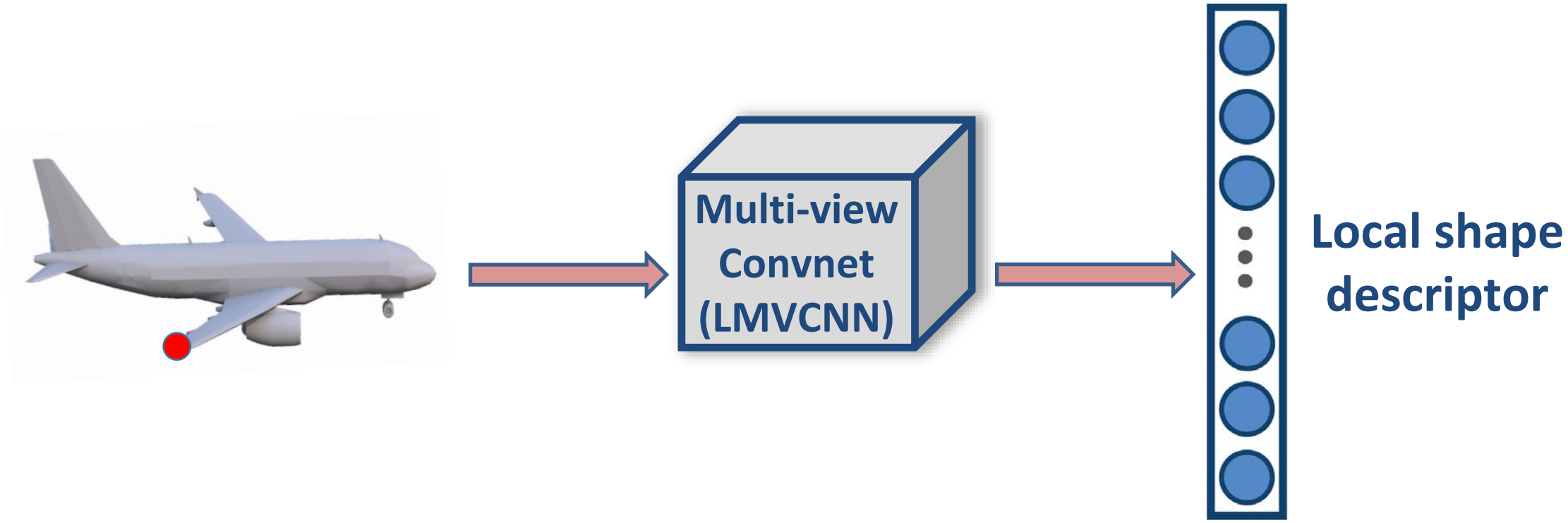
Duygu Ceylan[3] Vladimir G. Kim[3] Ersin Yumer[3]

[1]University of Massachusetts Amherst *[2]IIT Bombay* *[3]Adobe Research*

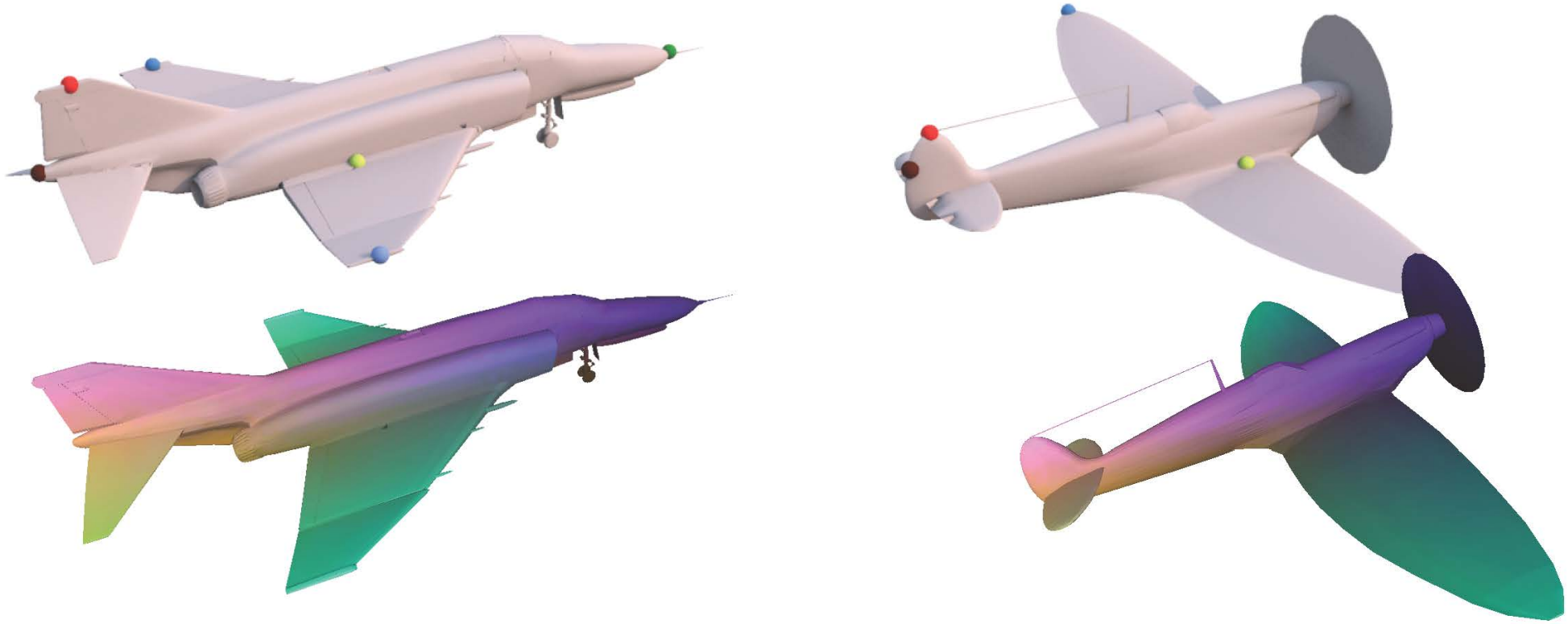
Goal: learn **local** shape descriptors

Goal: learn **local** shape descriptors

Goal: learn **local** shape descriptors

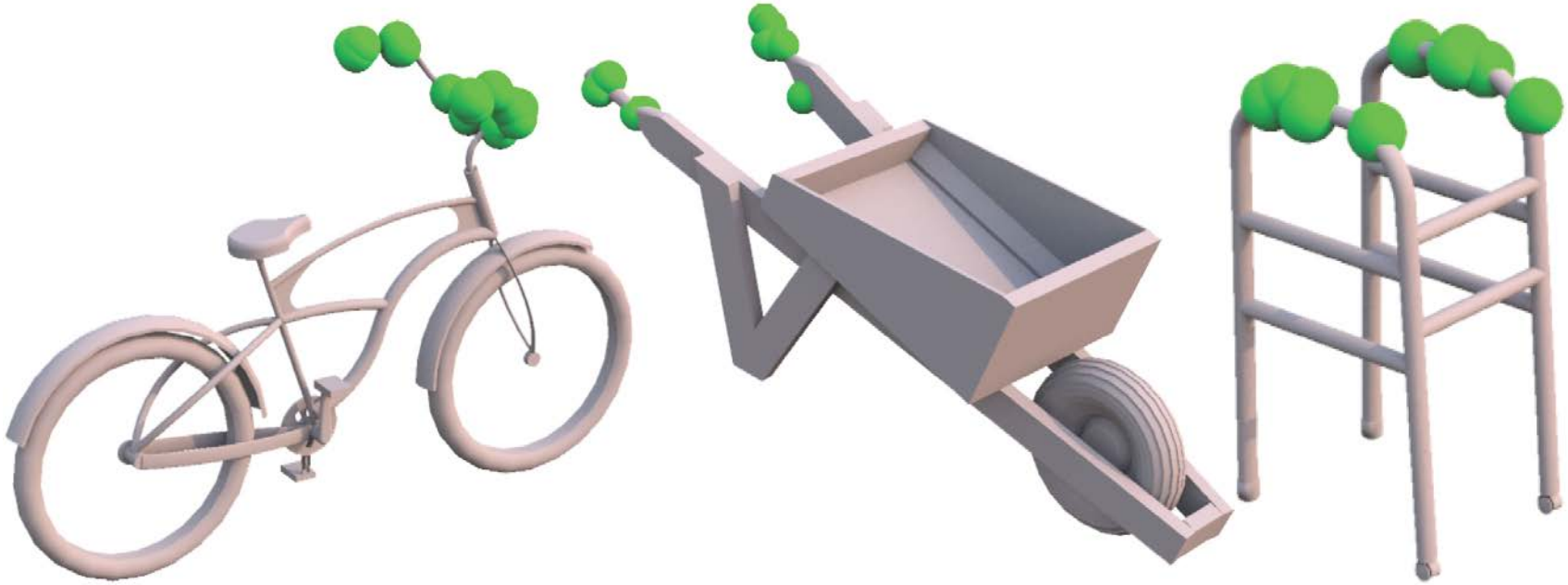


Why local shape descriptors? Keypoint detection/correspondences



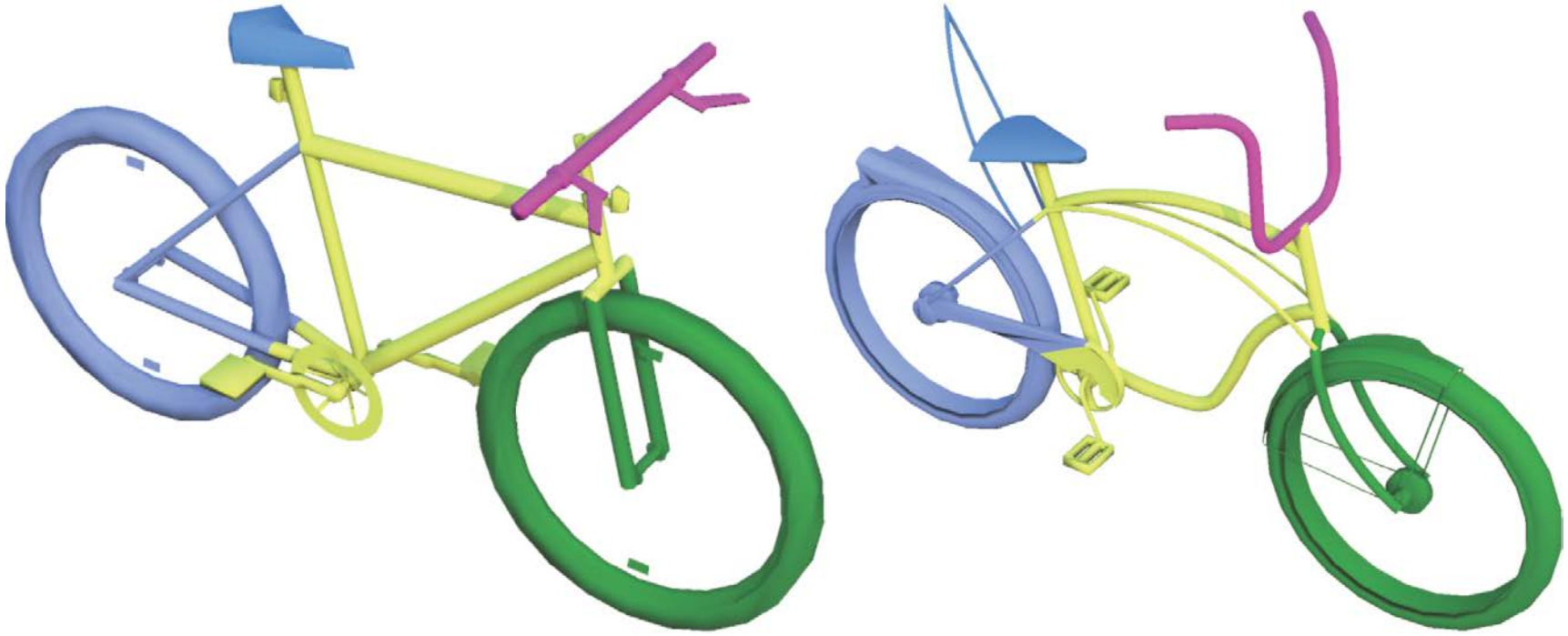
(similar colors correspond to points with similar descriptors)

Why local shape descriptors? Affordance prediction



Where do humans place their palms when they interact with these objects?

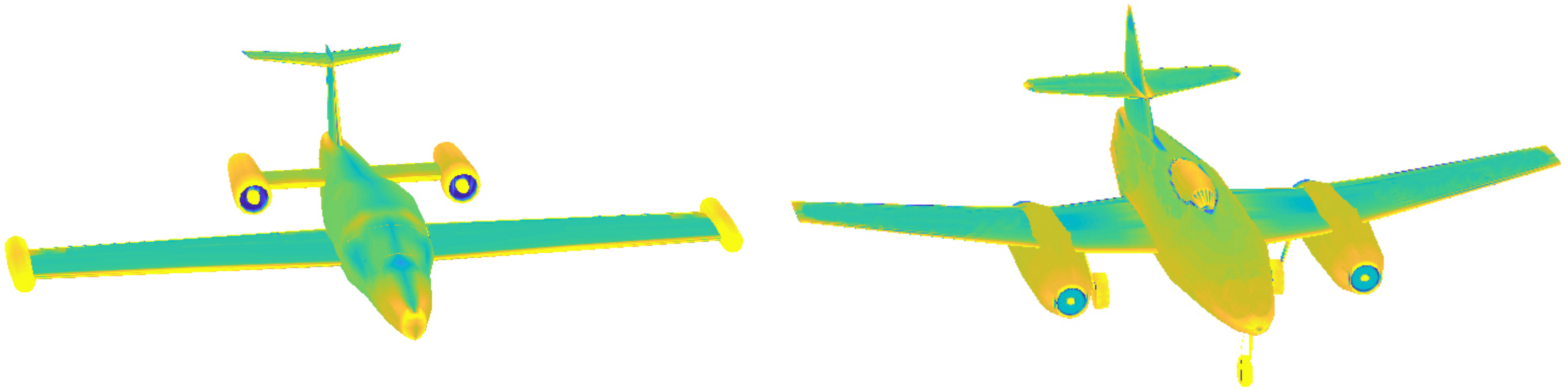
Why local shape descriptors? Shape segmentation & labeling



Classify points into labeled parts based on their descriptor

Challenges

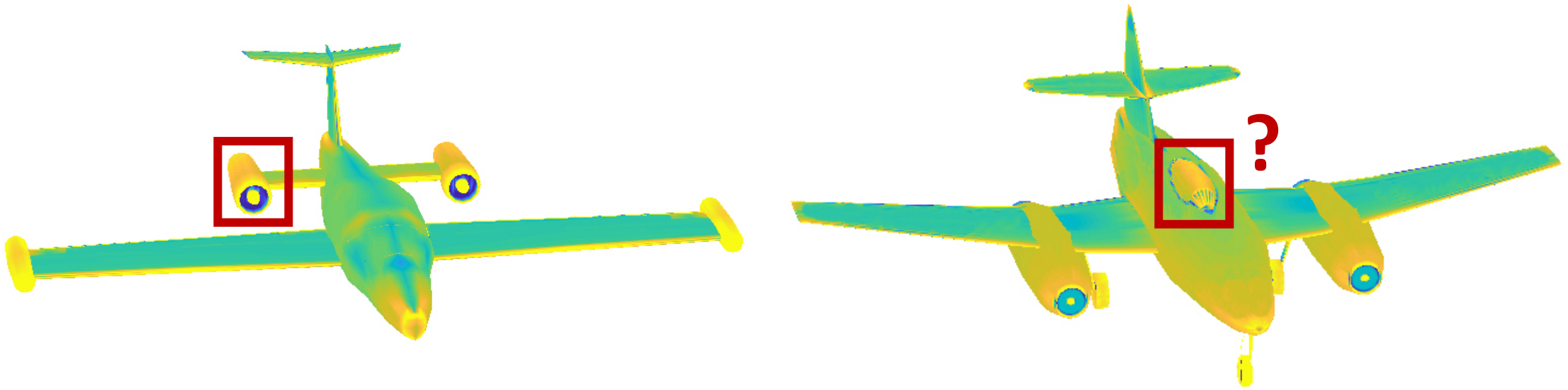
Low-level geometric cues not informative enough to yield semantic-aware descriptors



e.g., mean curvature

Challenges

Low-level geometric cues not informative enough to yield semantic-aware descriptors



e.g., mean curvature

Challenges

Low-level geometric cues not informative enough to yield semantic-aware descriptors

Large structural & geometric variability across objects, mainly man-made objects

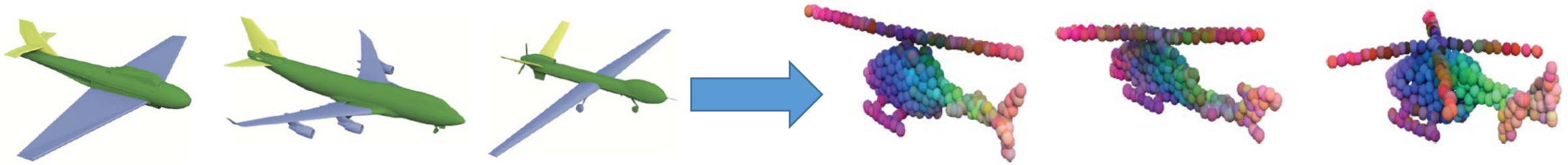
Models from Dosch Design

Challenges

Low-level geometric cues not informative enough to yield semantic-aware descriptors

Large structural & geometric variability across objects, mainly man-made objects

Generalize to novel object categories not seen during training



e.g., train on airplanes

test descriptors on helicopters

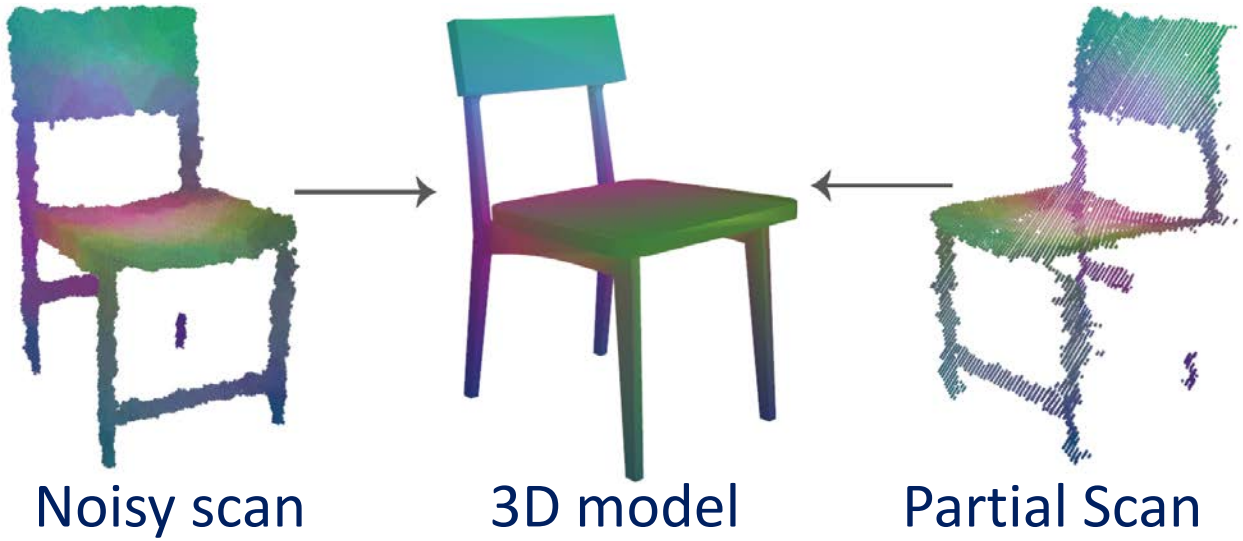
Challenges

Low-level geometric cues not informative enough to yield semantic-aware descriptors

Large structural & geometric variability across objects, mainly man-made objects

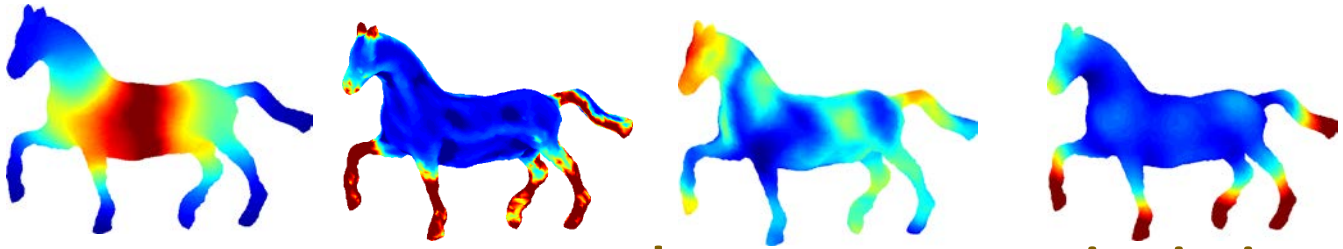
Generalize to novel object categories not seen during training

Robustness to noise and missing data



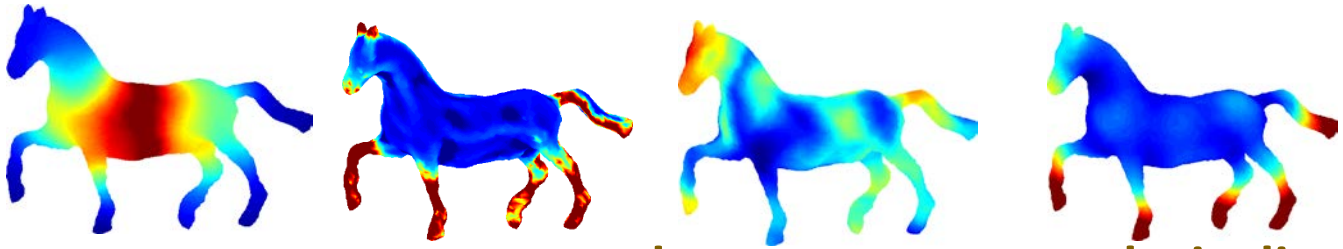
Related work

Hand-tuned geometric descriptors
see **Xu et al. EG STAR ‘16**



Related work

Hand-tuned geometric descriptors
see **Xu et al. EG STAR '16**



PCA **curvature** **shape contexts** **geodesic dist.**

Approaches (concurrent / after our submission):

Volumetric / octree-based methods: **Maturana et al. '15, Zeng et al. '17 (3DMatch), Riegler et al. '17 (OctNet), Wang et al. '17 (O-CNN), Klokov et al. '17 (kd-net) …**

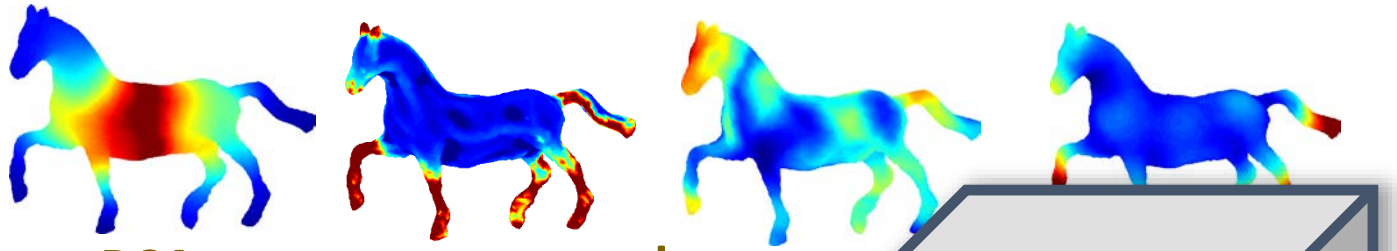
Point-based networks: **Qi et al. '17 (PointNet / PointNet++), Hua et al. '18 …**

Graph-based / spectral networks: **Yi et al. '17 (SyncSpecCNN), Bronstein et al. '17 …**

Surface embedding networks: **Maron et al. '17, Groueix et al. '18 …**

Related work

Hand-tuned geometric descriptors
see **Xu et al. EG STAR '16**



Our method: Multi-view network

Approaches (concurrent / after our submission):

Volumetric / octree-based methods: **Maturana et al. '15, Zeng et al. '17 (3DMatch), Riegler et al. '17 (OctNet), Wang et al. '17 (O-CNN), Klokov et al. '17 (kd-net) …**

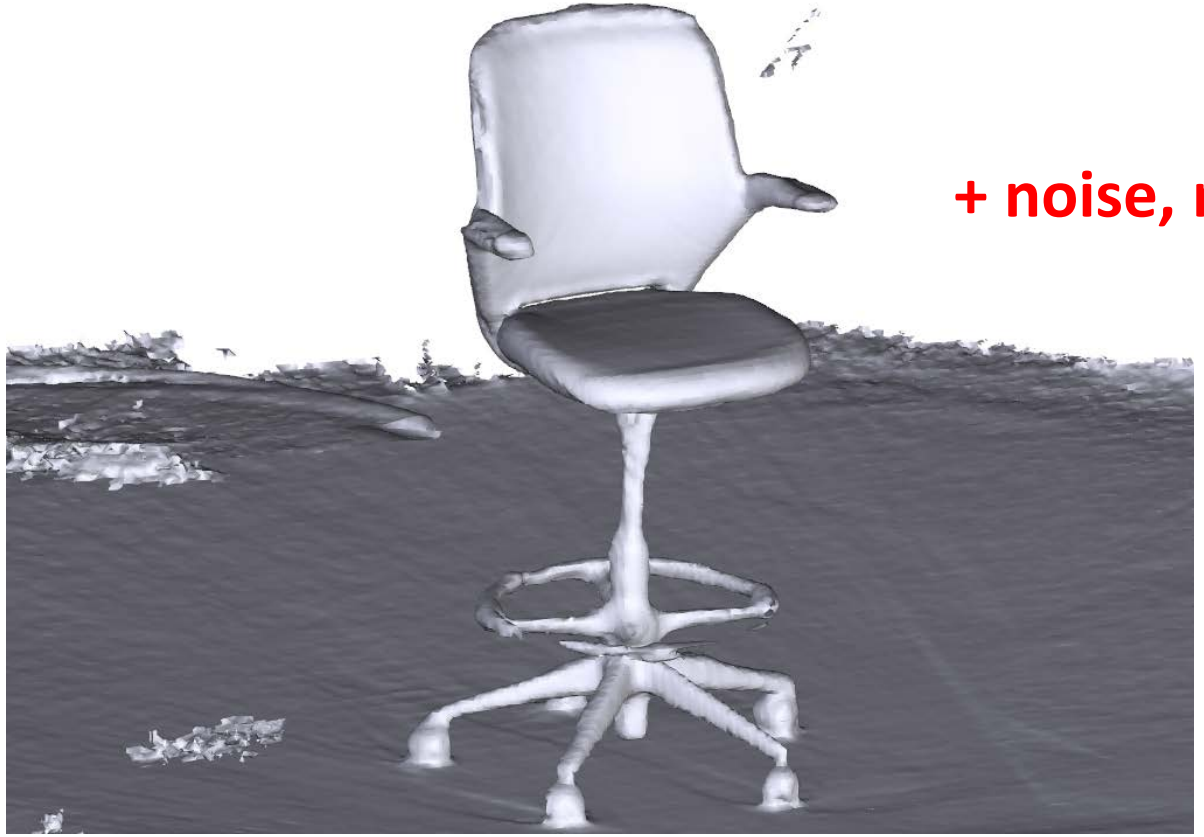
Point-based networks: **Qi et al. '17 (PointNet / PointNet++), Hua et al. '18 …**

Graph-based / spectral networks: **Yi et al. '17 (SyncSpecCNN), Bronstein et al. '17 …**

Surface embedding networks: **Maron et al. '17, Groueix et al. '18 …**

Key Observations

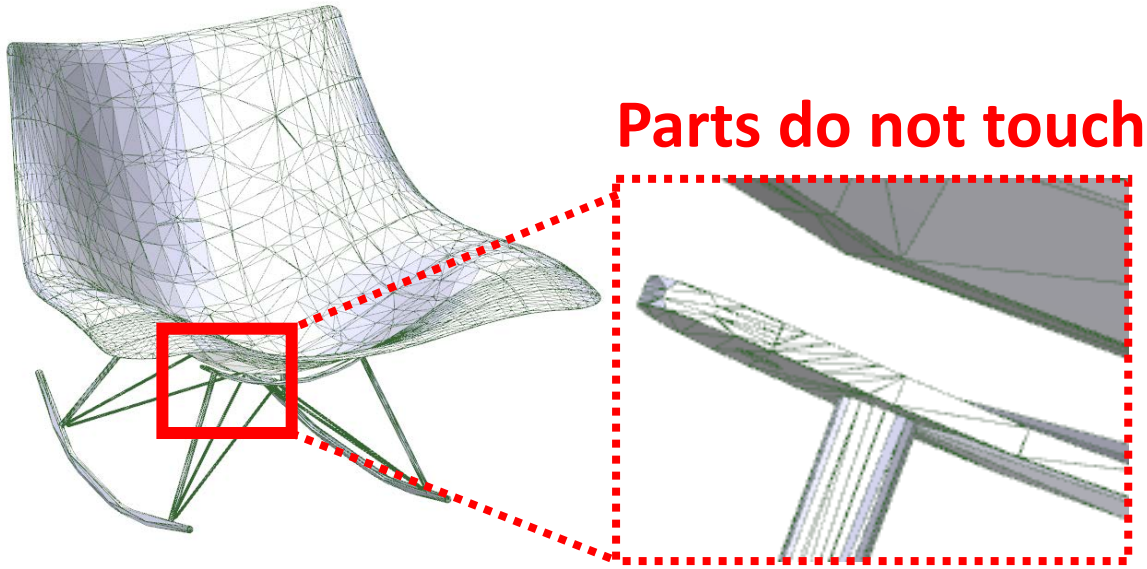
3D scans **capture the surface.**



+ noise, missing regions etc

Key Observations

3D models are often **designed for viewing.**

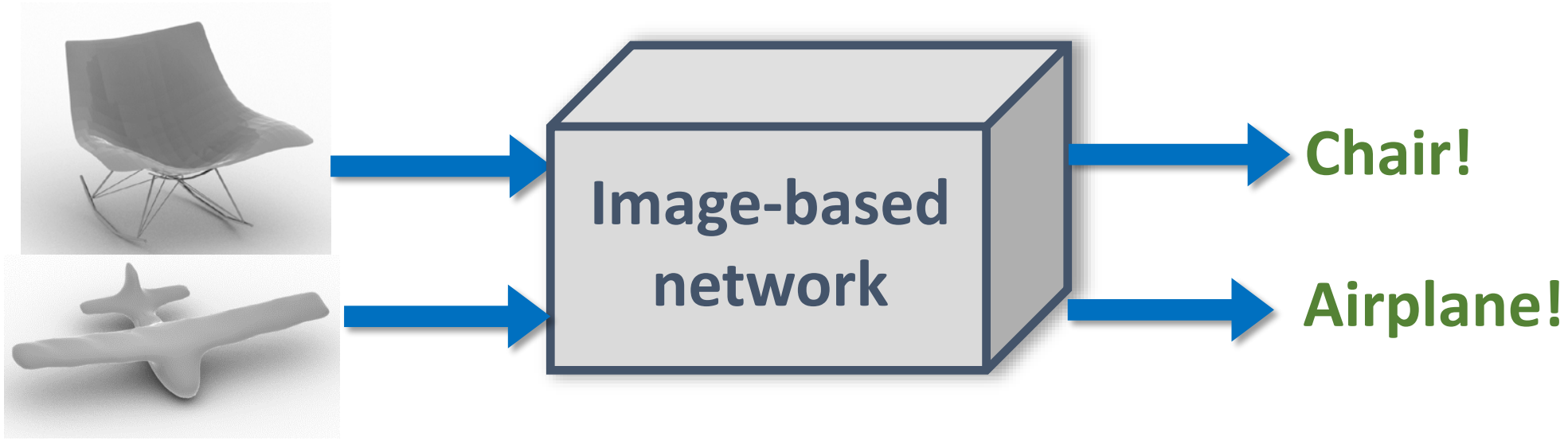


Parts do not touch!

(not easily noticeable to the viewer,
yet geometric implications on topology, connectedness...)

Key Observations

Shape renderings can be treated as **photos of objects** (without texture)



Shape renderings can be processed by powerful image-based architectures through transfer learning from massive image datasets.

(Su et al, ICCV 2015)
(Kalogerakis et al. CVPR 2017)

Key Ideas

Deep architecture for processing **rendered views of surface neighborhoods** around points **at multiple scales**. **View selection** to handle self-occlusions.

Key Ideas

Deep architecture for processing **rendered views of surface neighborhoods** around points **at multiple scales**. **View selection** to handle self-occlusions.

Trained to **embed semantically similar points close to each other** in descriptor space.

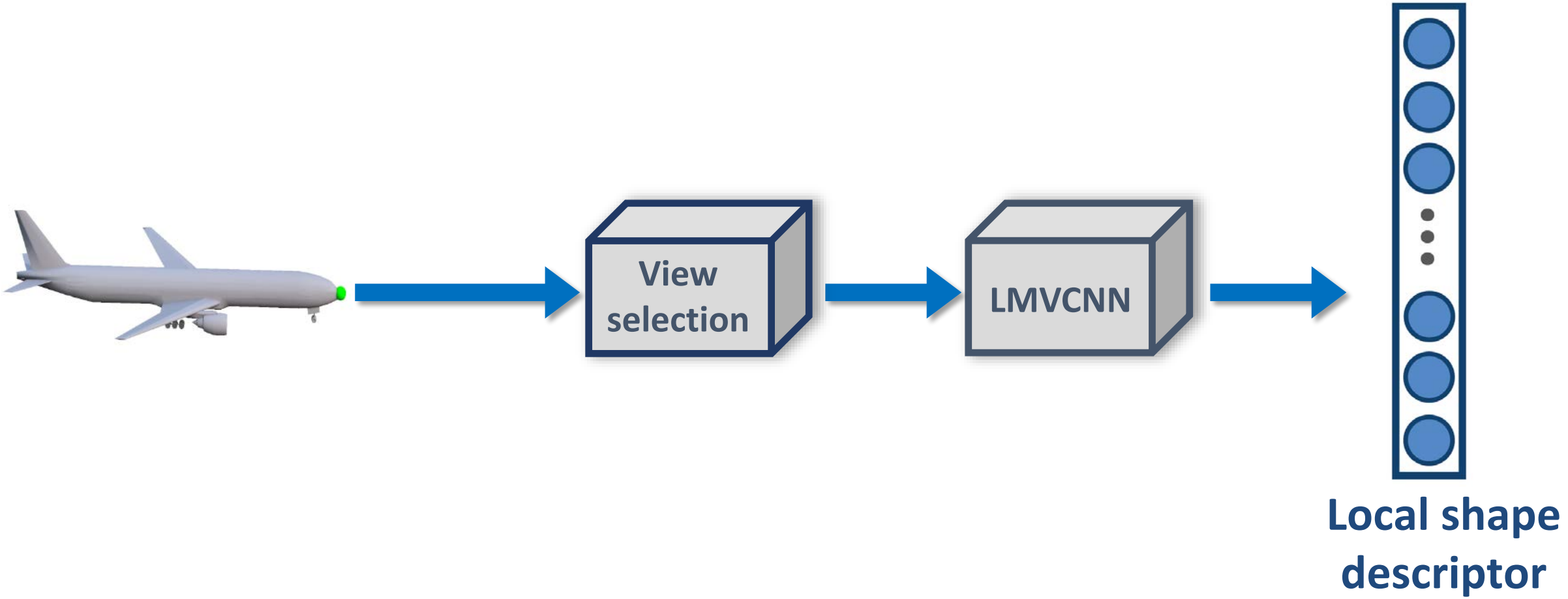
Key Ideas

Deep architecture for processing **rendered views of surface neighborhoods** around points **at multiple scales**. **View selection** to handle self-occlusions.

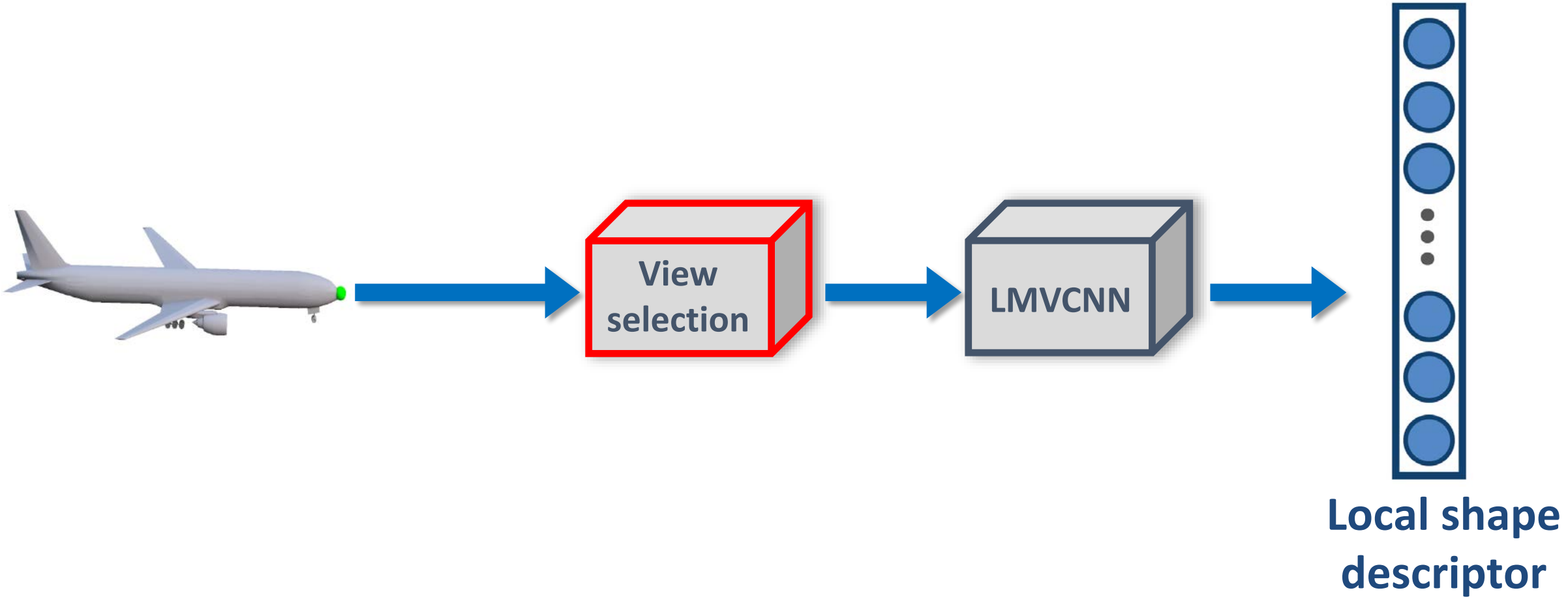
Trained to **embed semantically similar points close to each other** in descriptor space.

Massive, synthetically generated training dataset: 977M corresponding point pairs

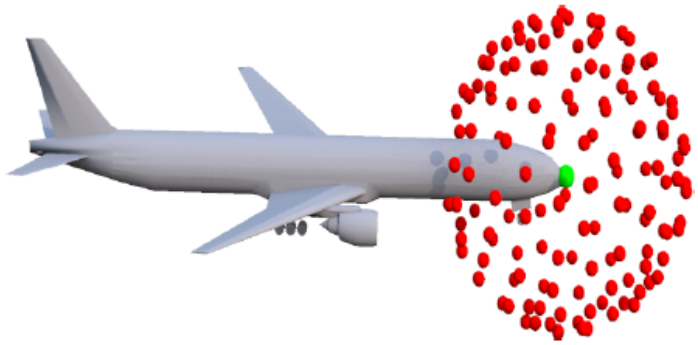
Pipeline



Pipeline

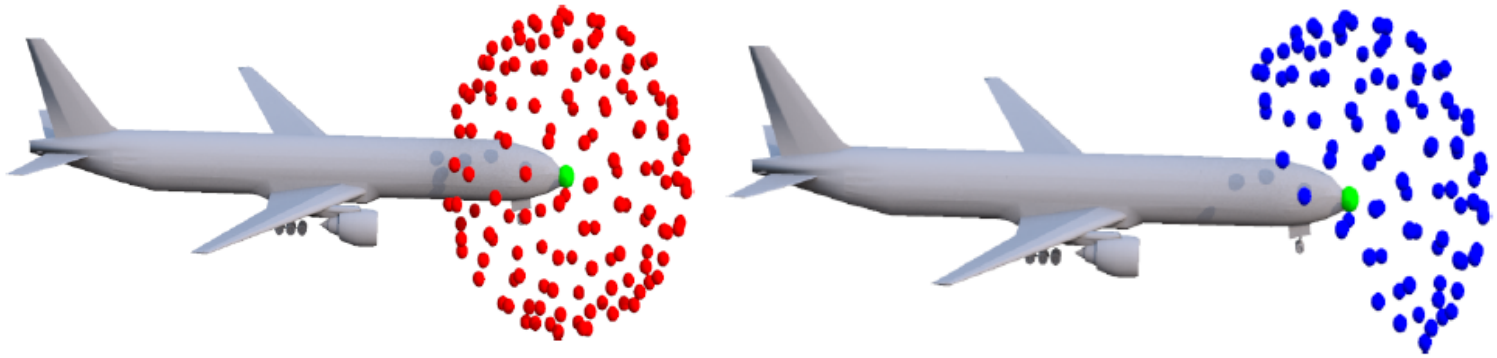


View Selection



Step 1: Uniformly sample directions on the viewing hemisphere of the input point

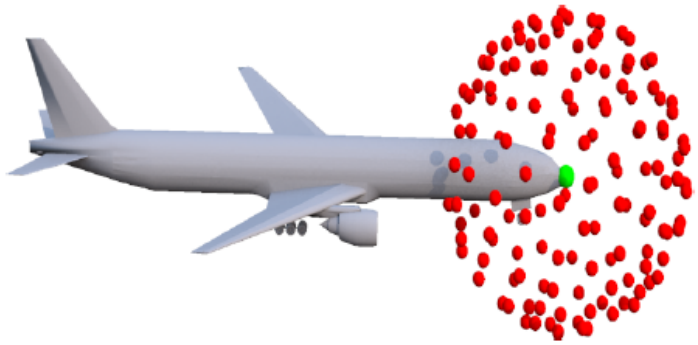
View Selection

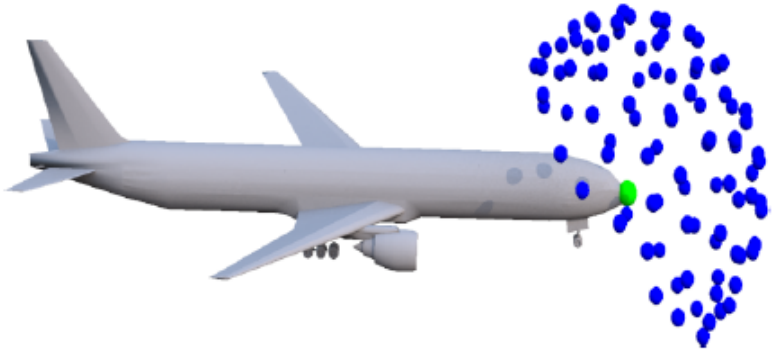


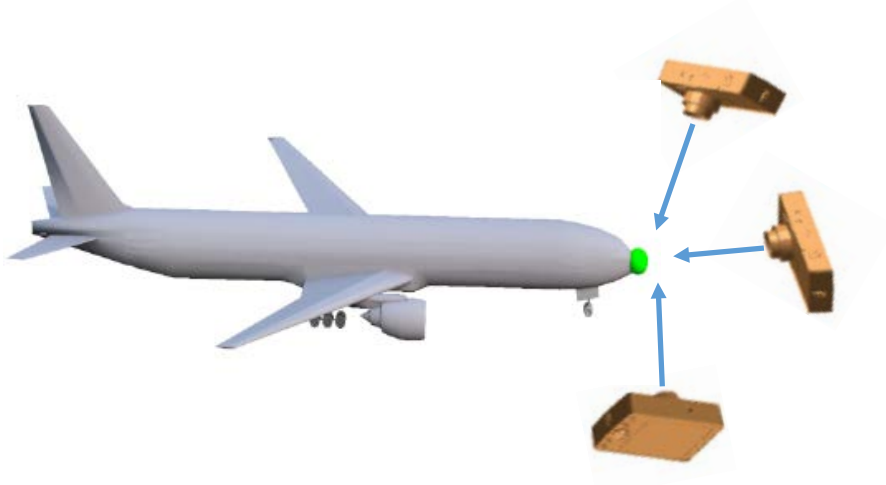
Step 1: Uniformly sample directions on the viewing hemisphere of the input point

Step 2: Find directions the point is visible from

View Selection





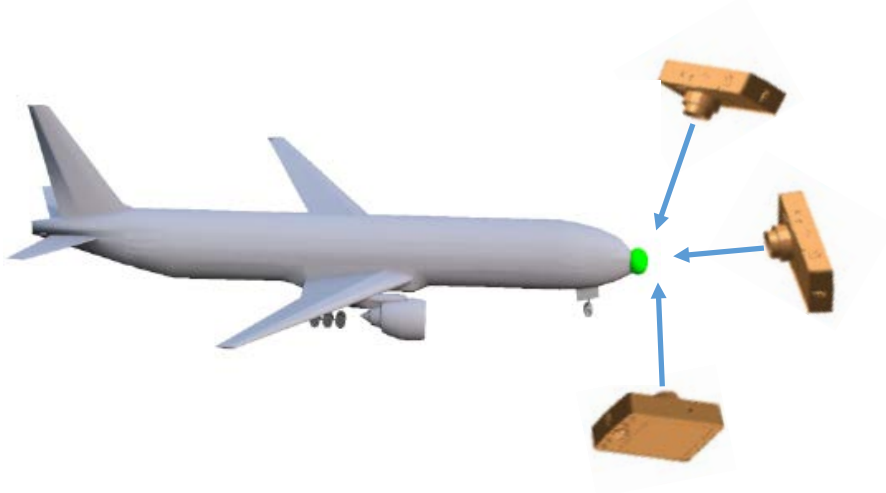


Step 1: Uniformly sample directions on the viewing hemisphere of the input point

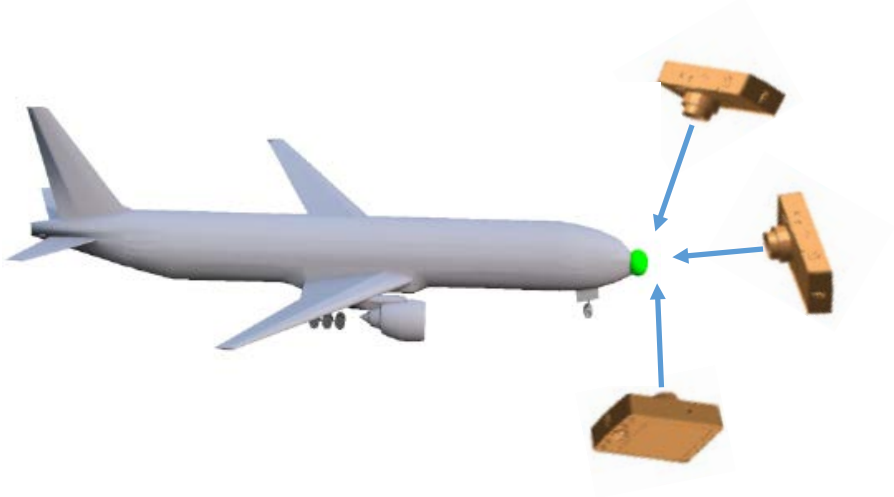
Step 2: Find directions the point is visible from

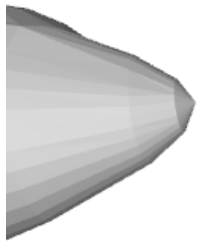
Step 3: Prune redundant views through clustering

View Selection



Rendered views

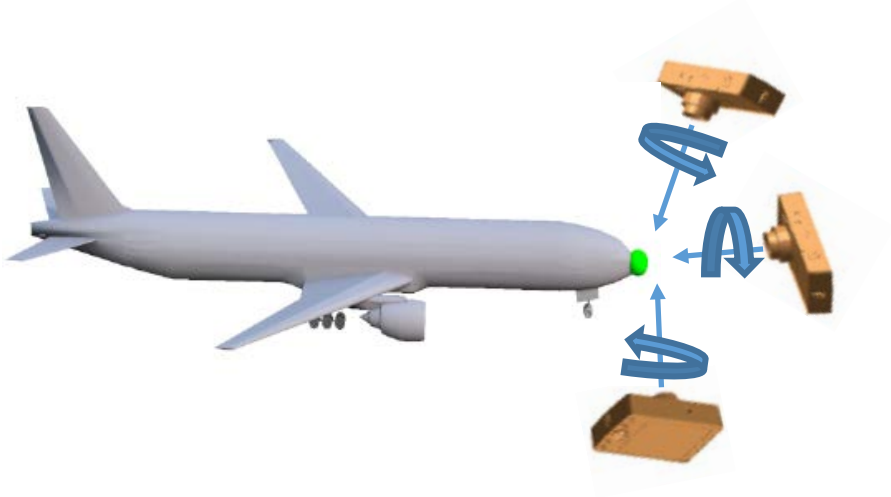


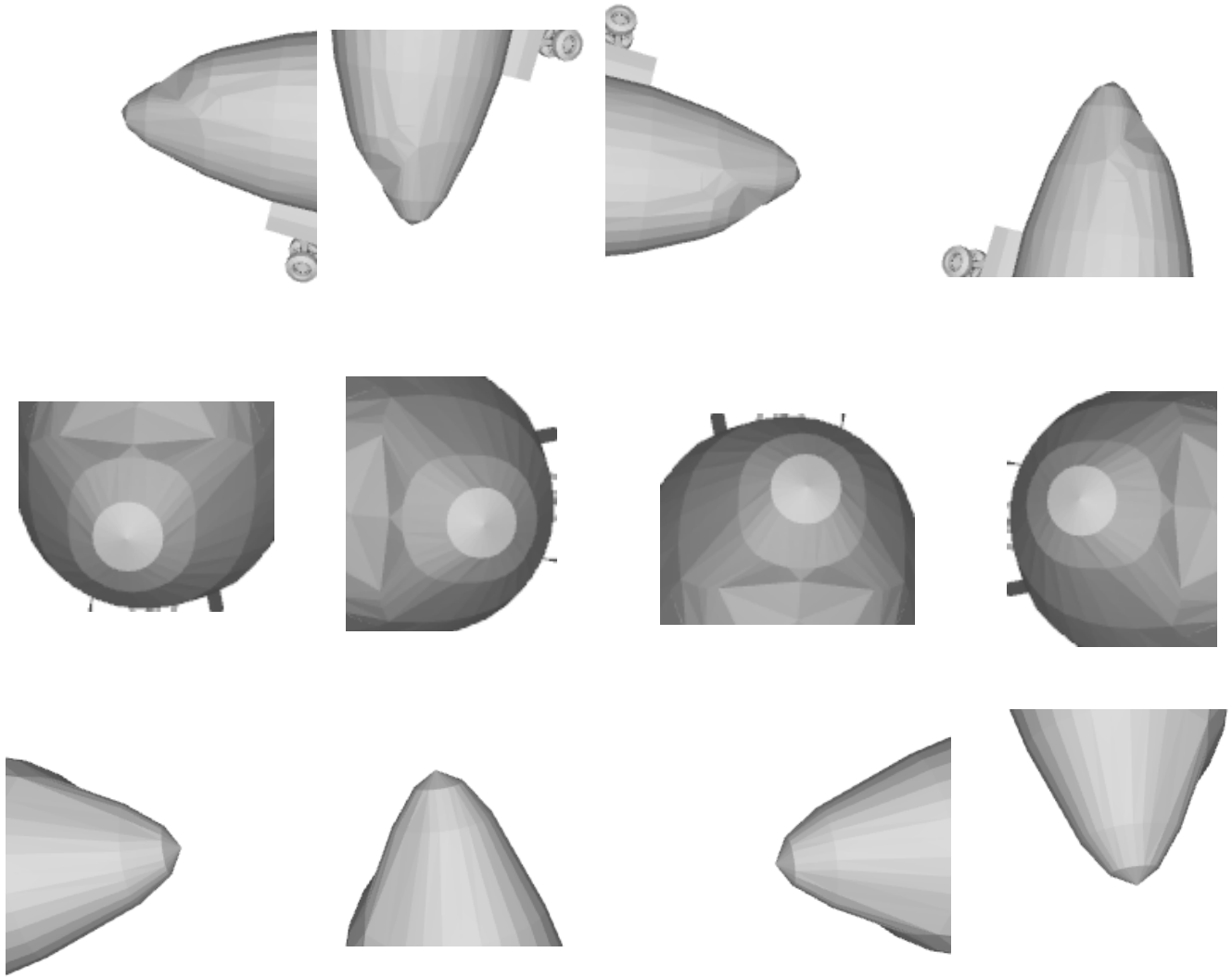


Render **shaded images** (*normal dot view vector*).

Point is at the **center of the rendered image**.

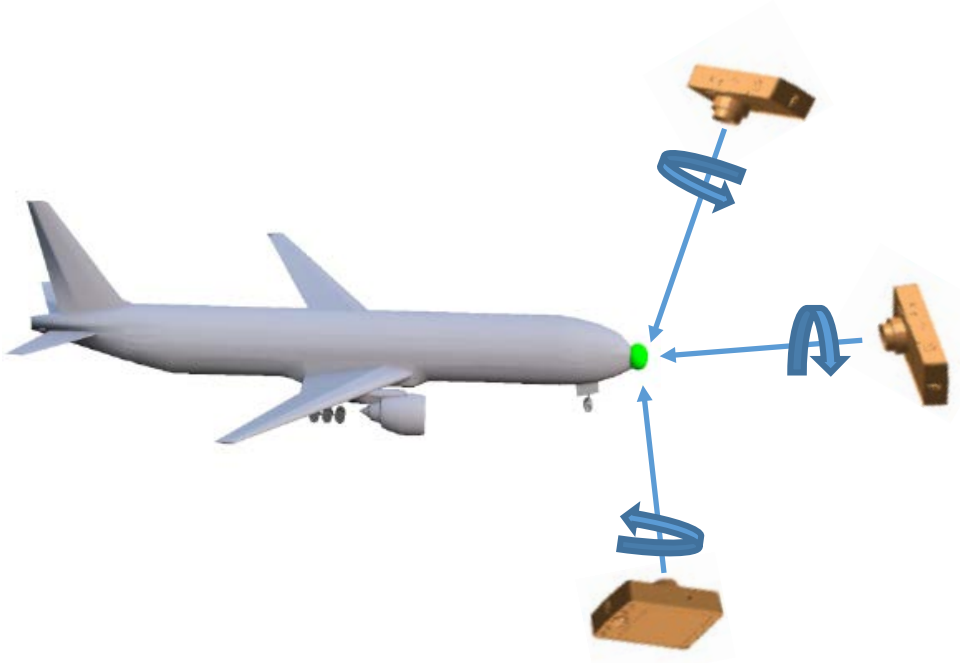
Rendered views

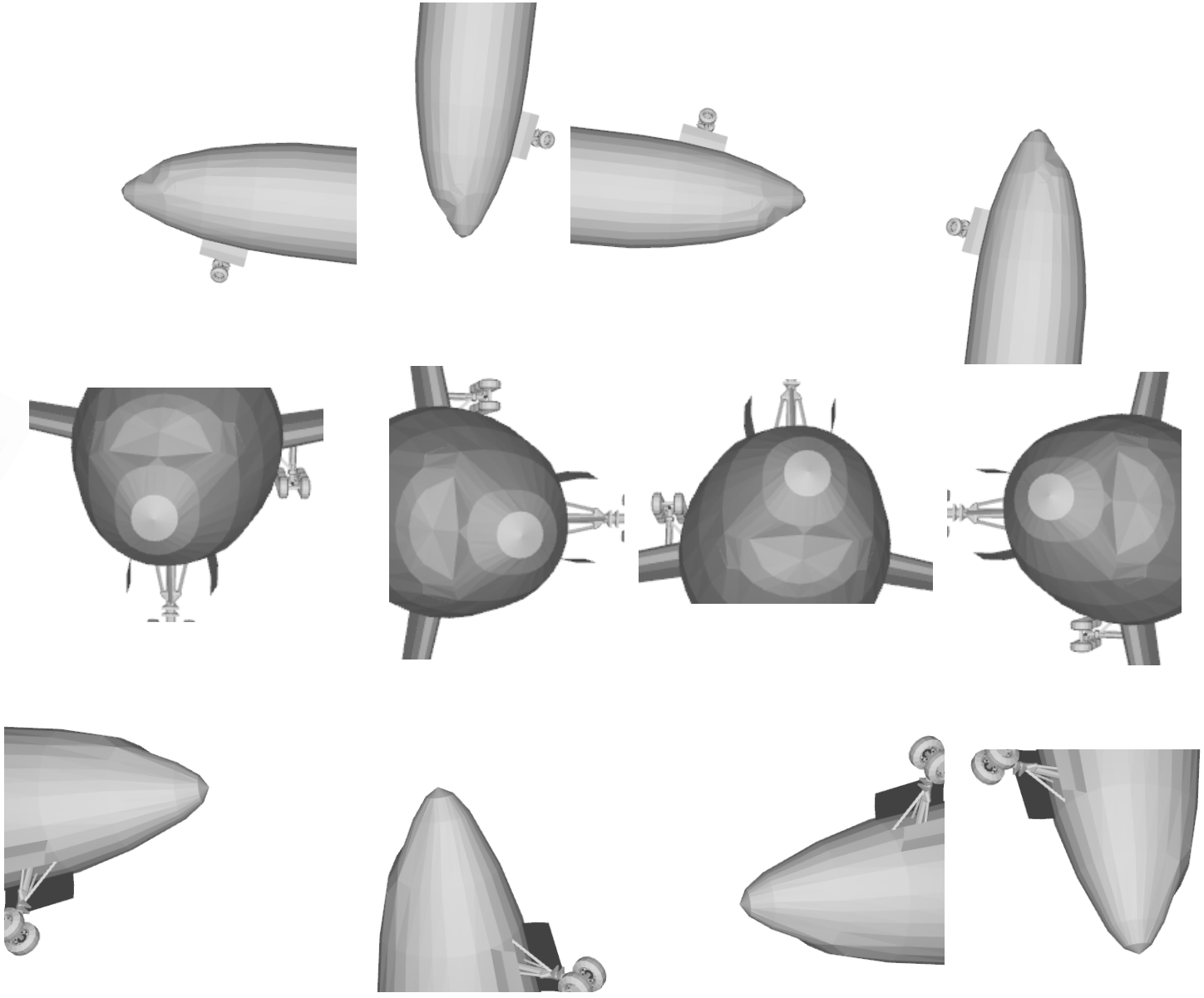




Perform in-plane camera rotations for **rotational invariance.**

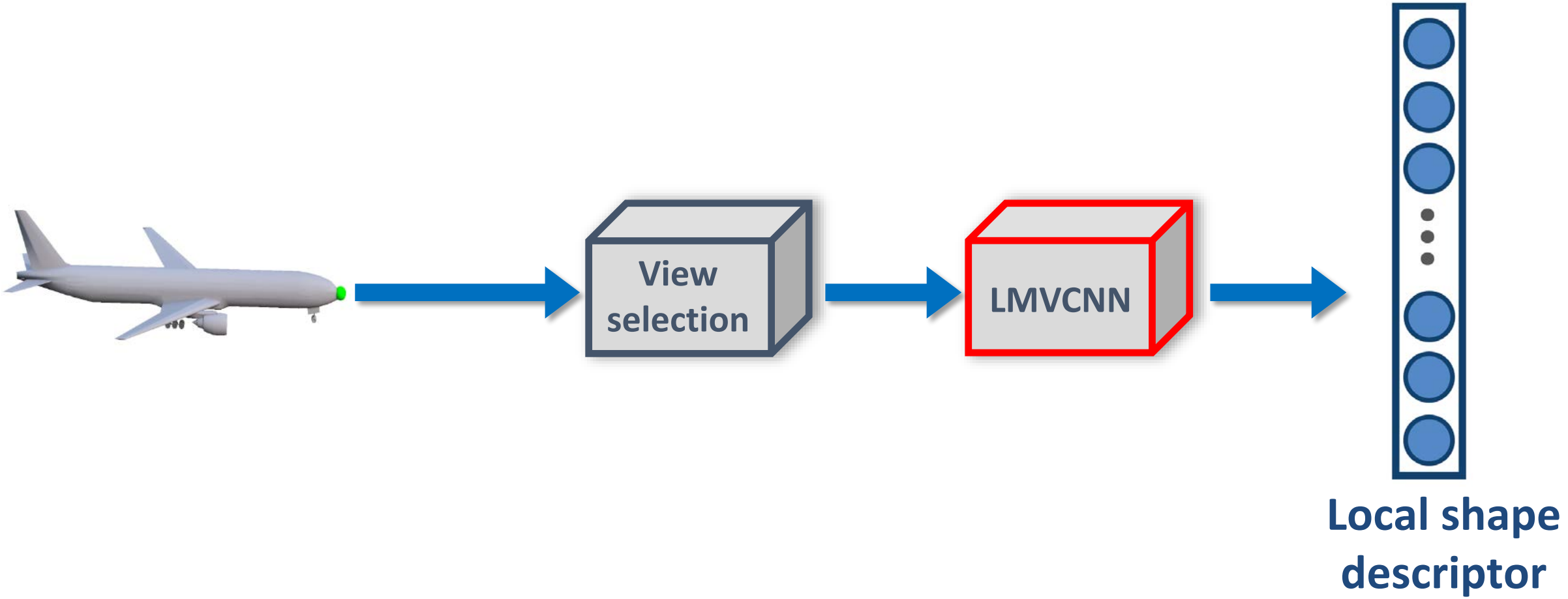
Rendered views



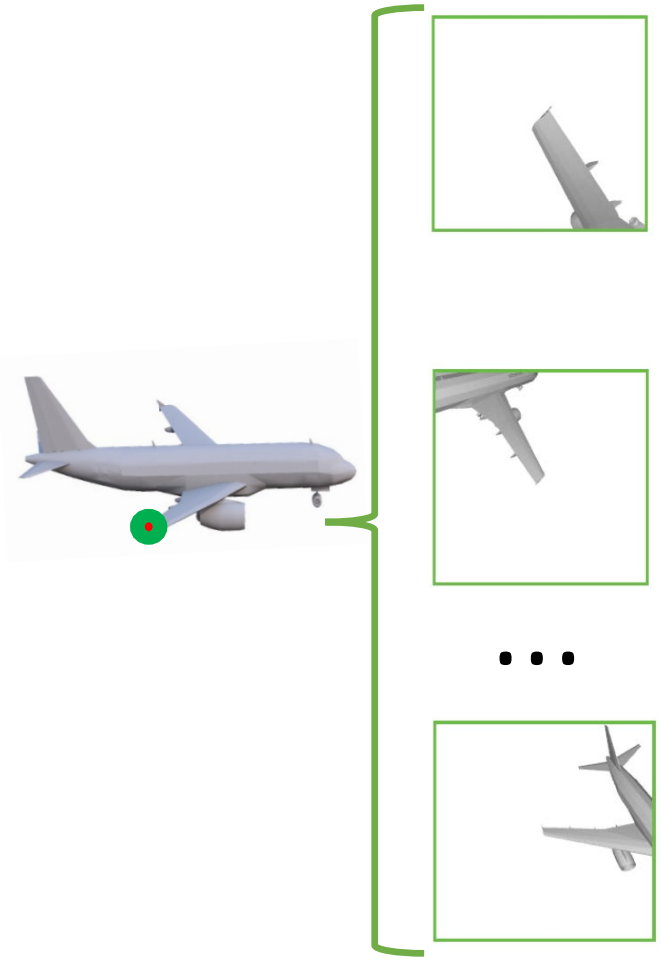


Use progressively zoomed out views to capture **multi-scale context**.

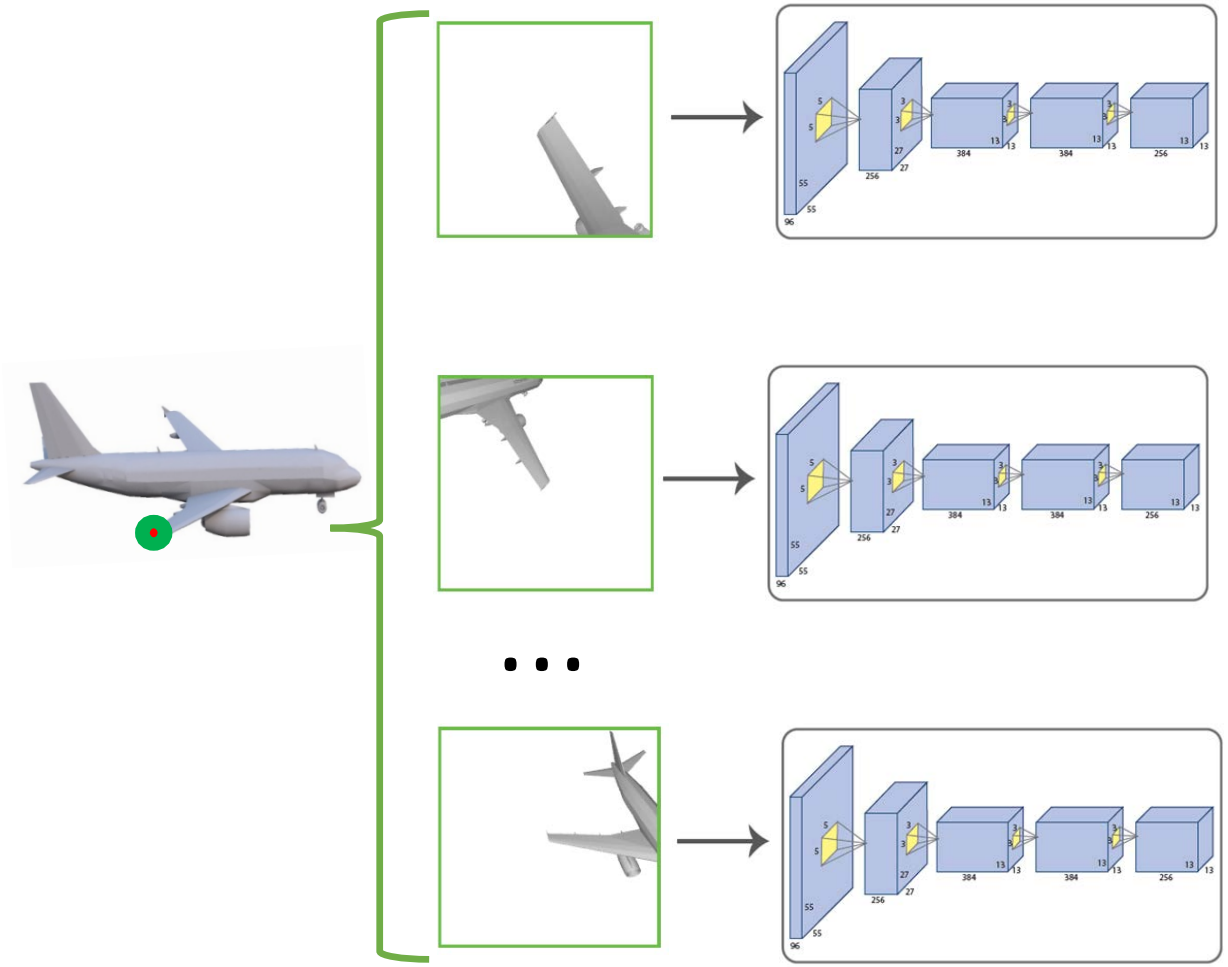
Pipeline



Network Architecture

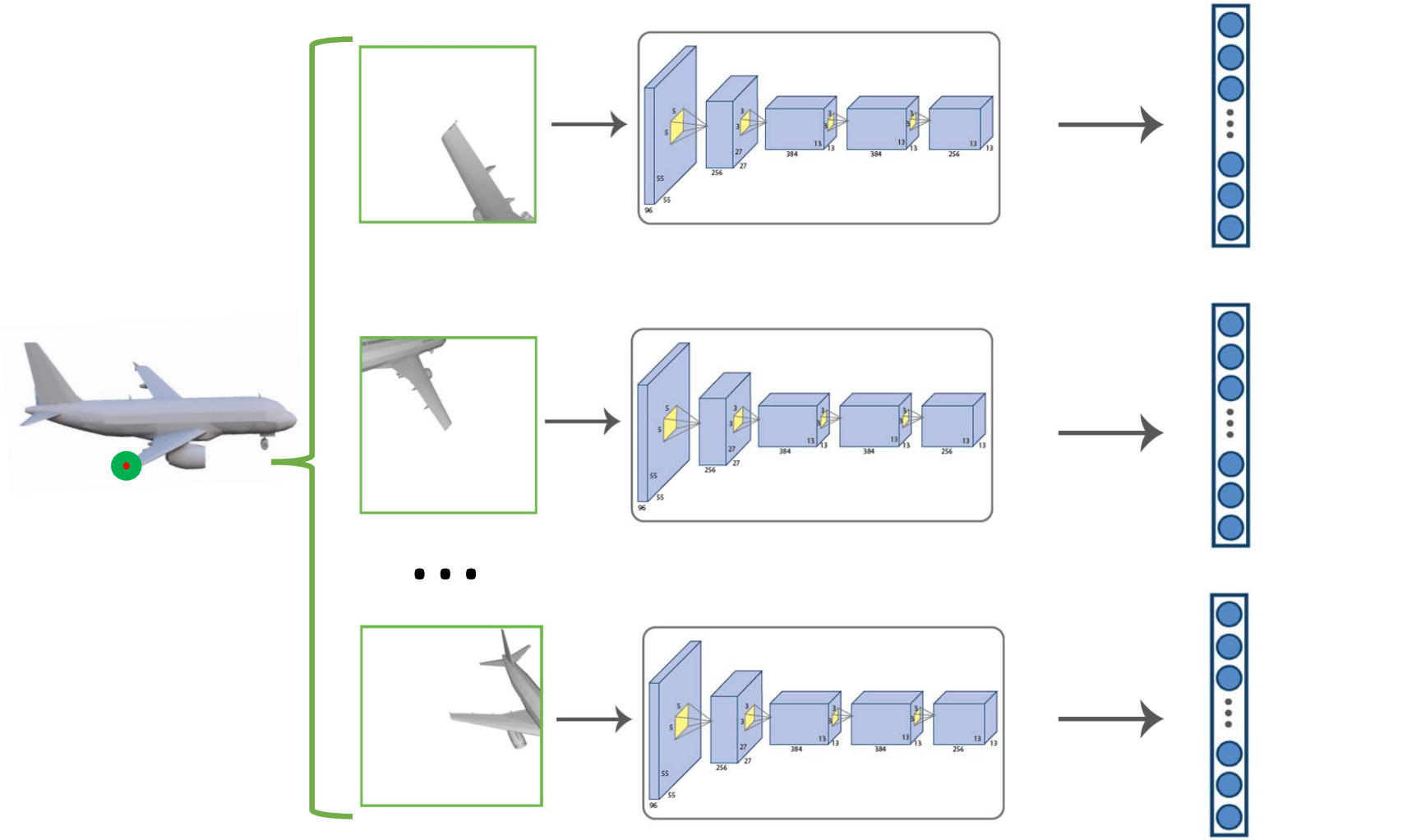


Network Architecture

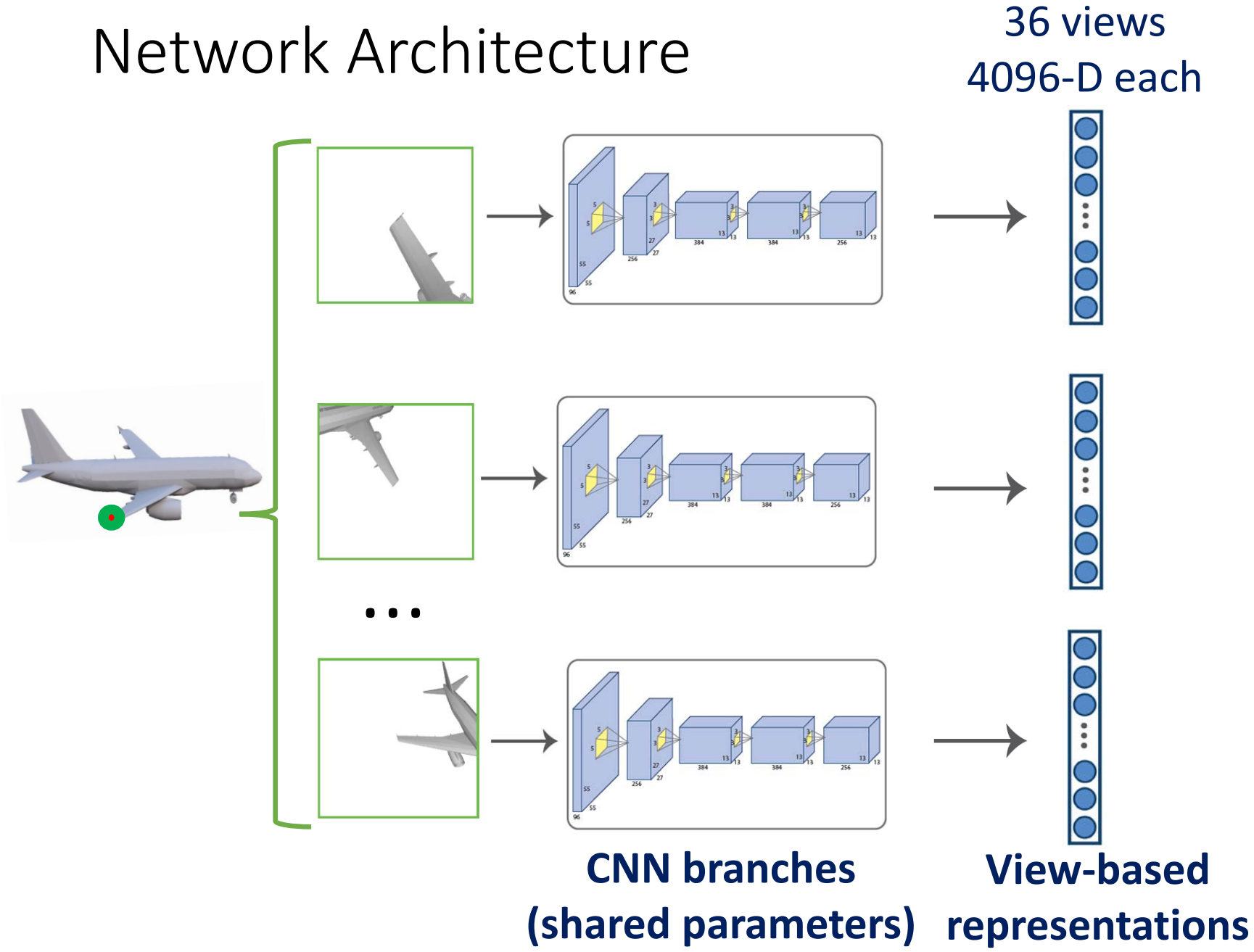


CNN branches
(shared parameters)

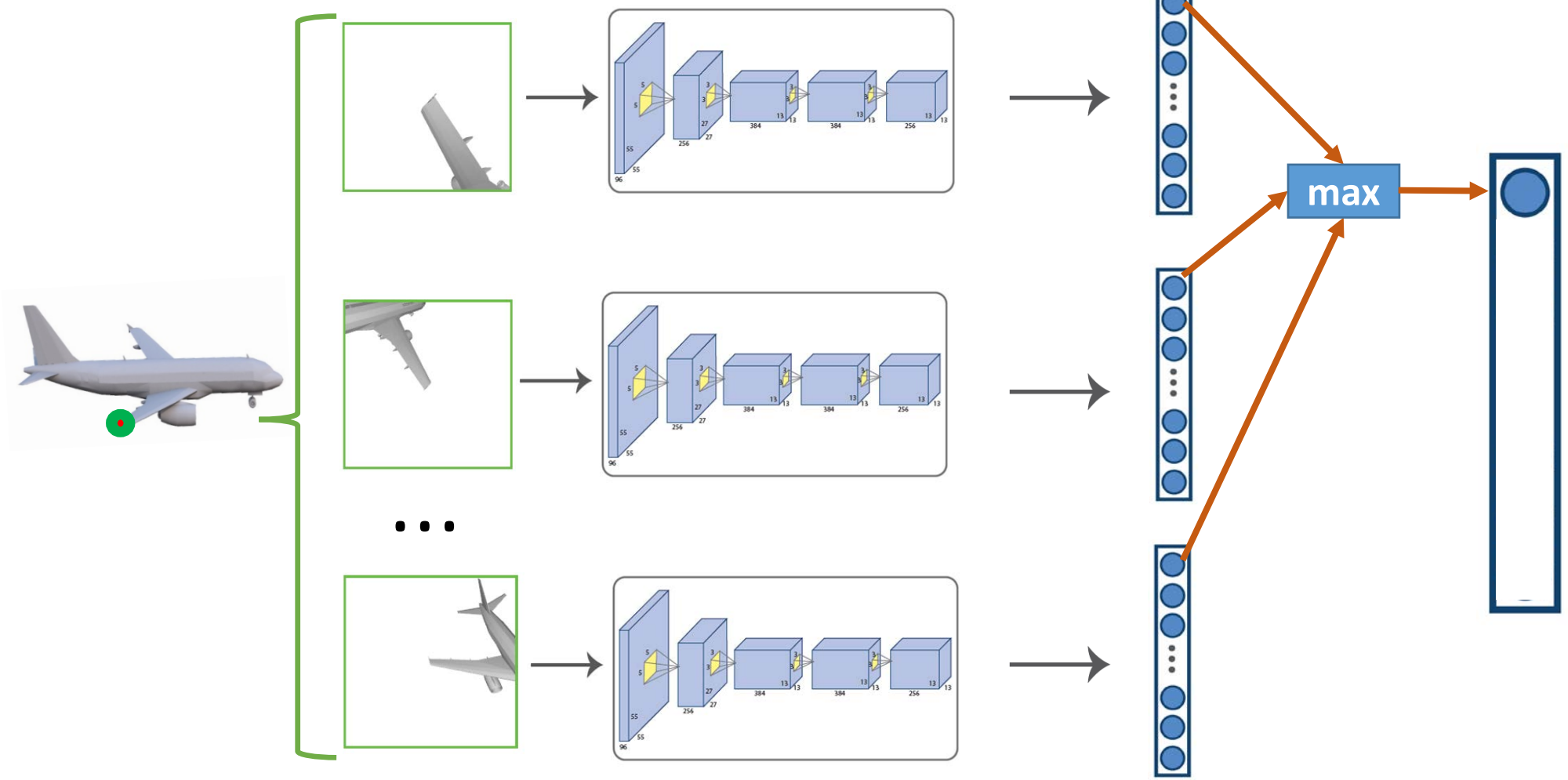
Network Architecture



Network Architecture



Network Architecture

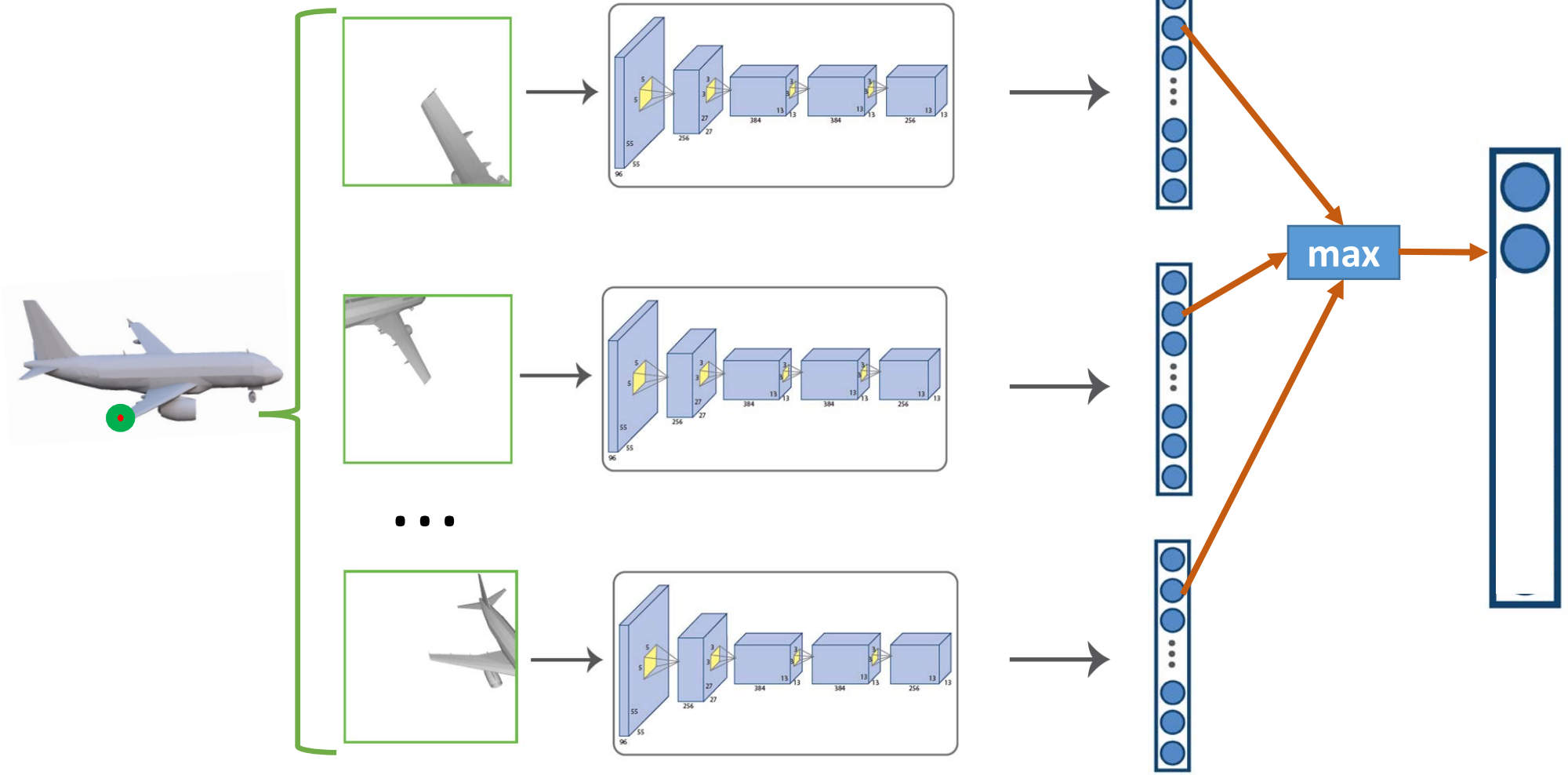


(Su et al, 2015)
(Kalogerakis et al. 2017)

Network Architecture

36 views
4096-D each

(Su et al, 2015)
(Kalogerakis et al. 2017)

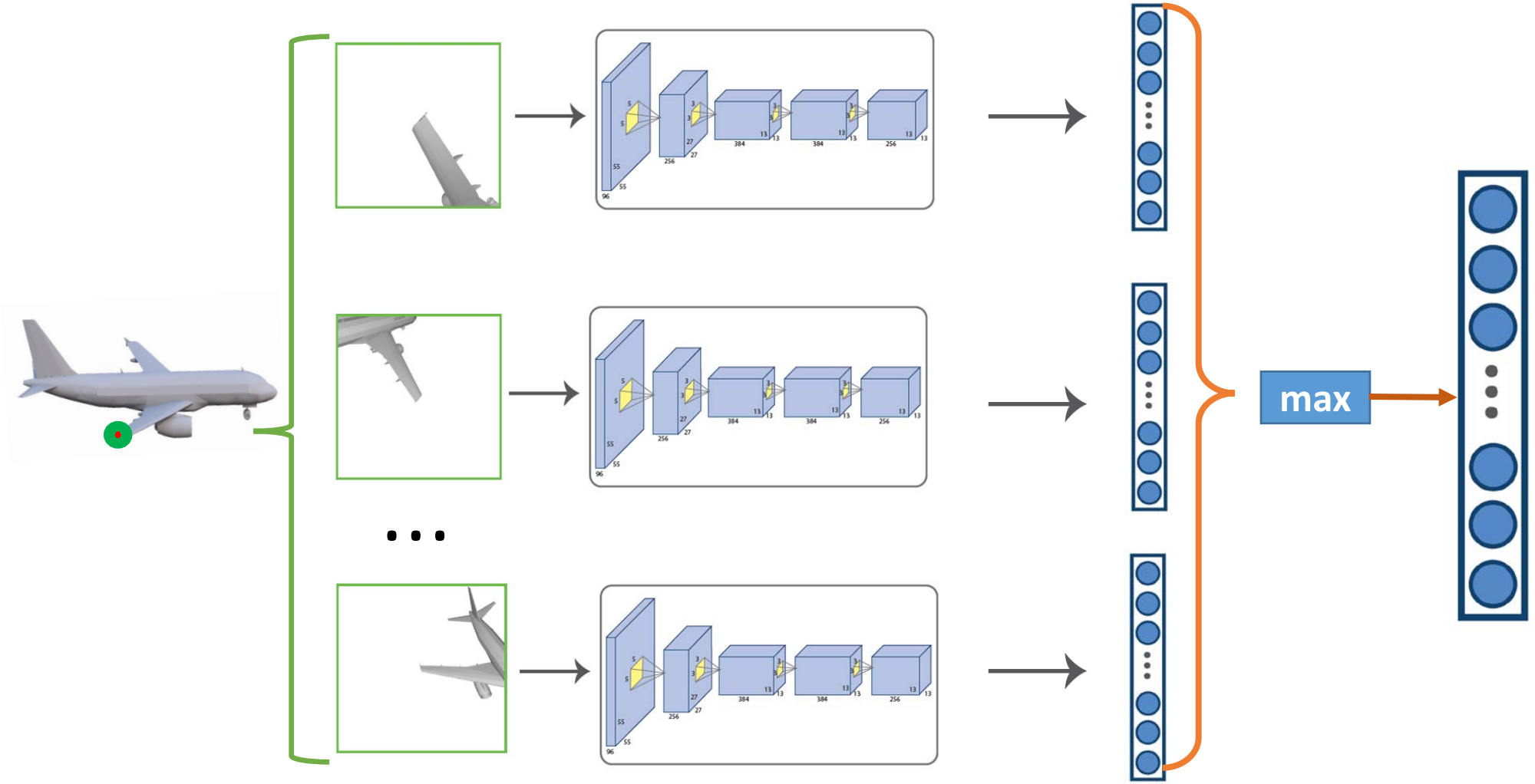


CNN branches (shared parameters)

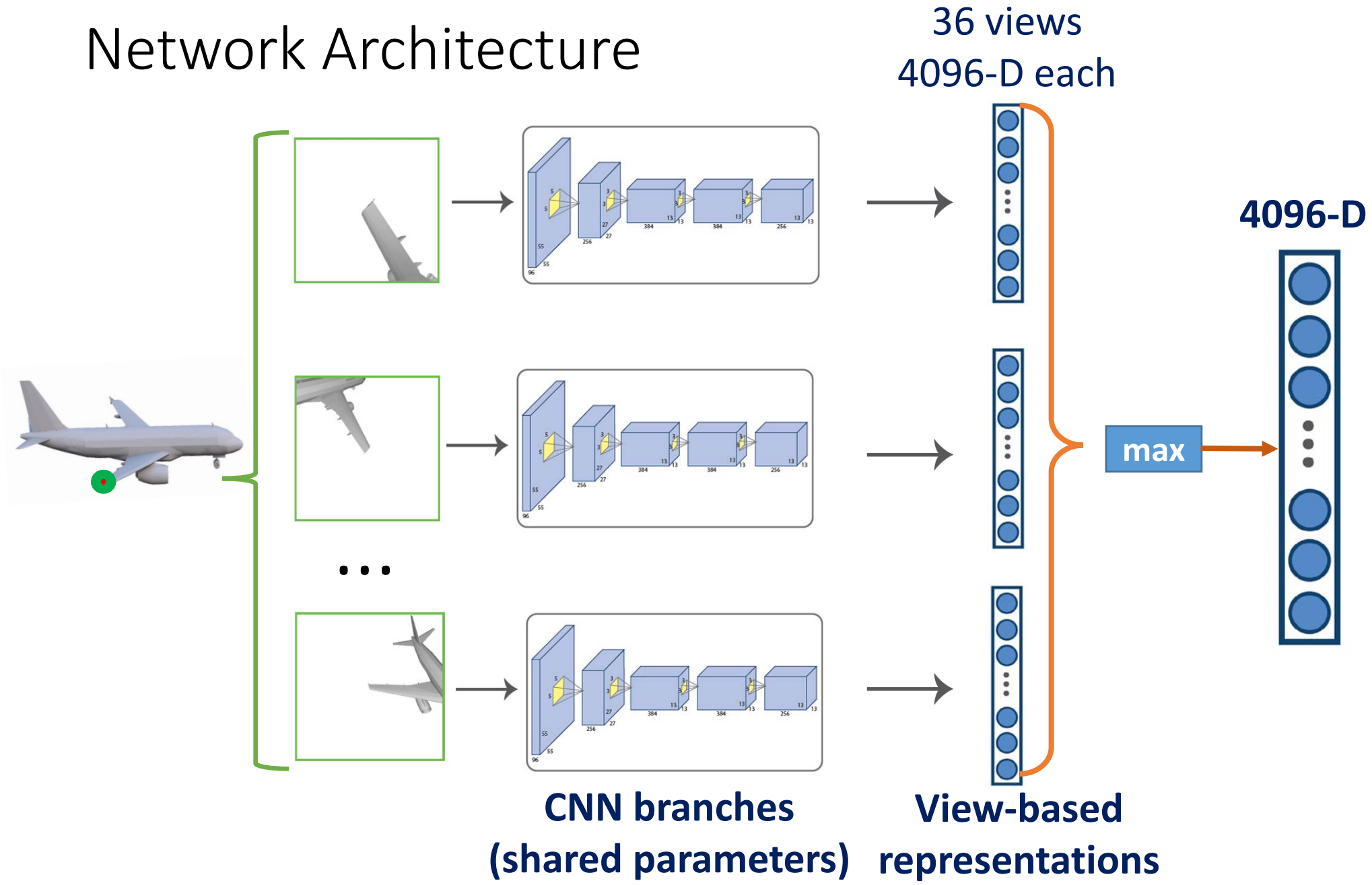
View-based representations

Network Architecture

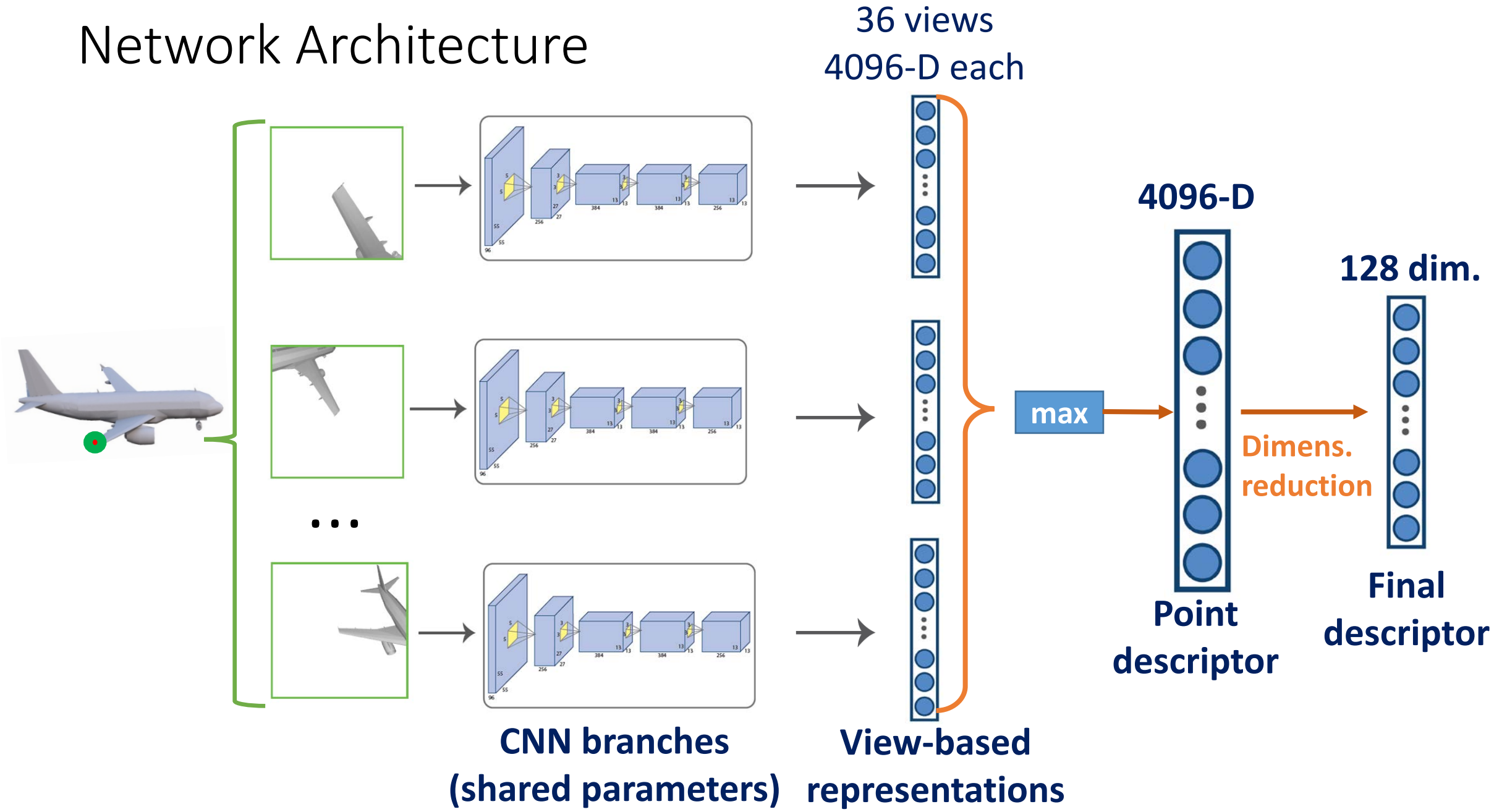
(Su et al, 2015)
(Kalogerakis et al. 2017)



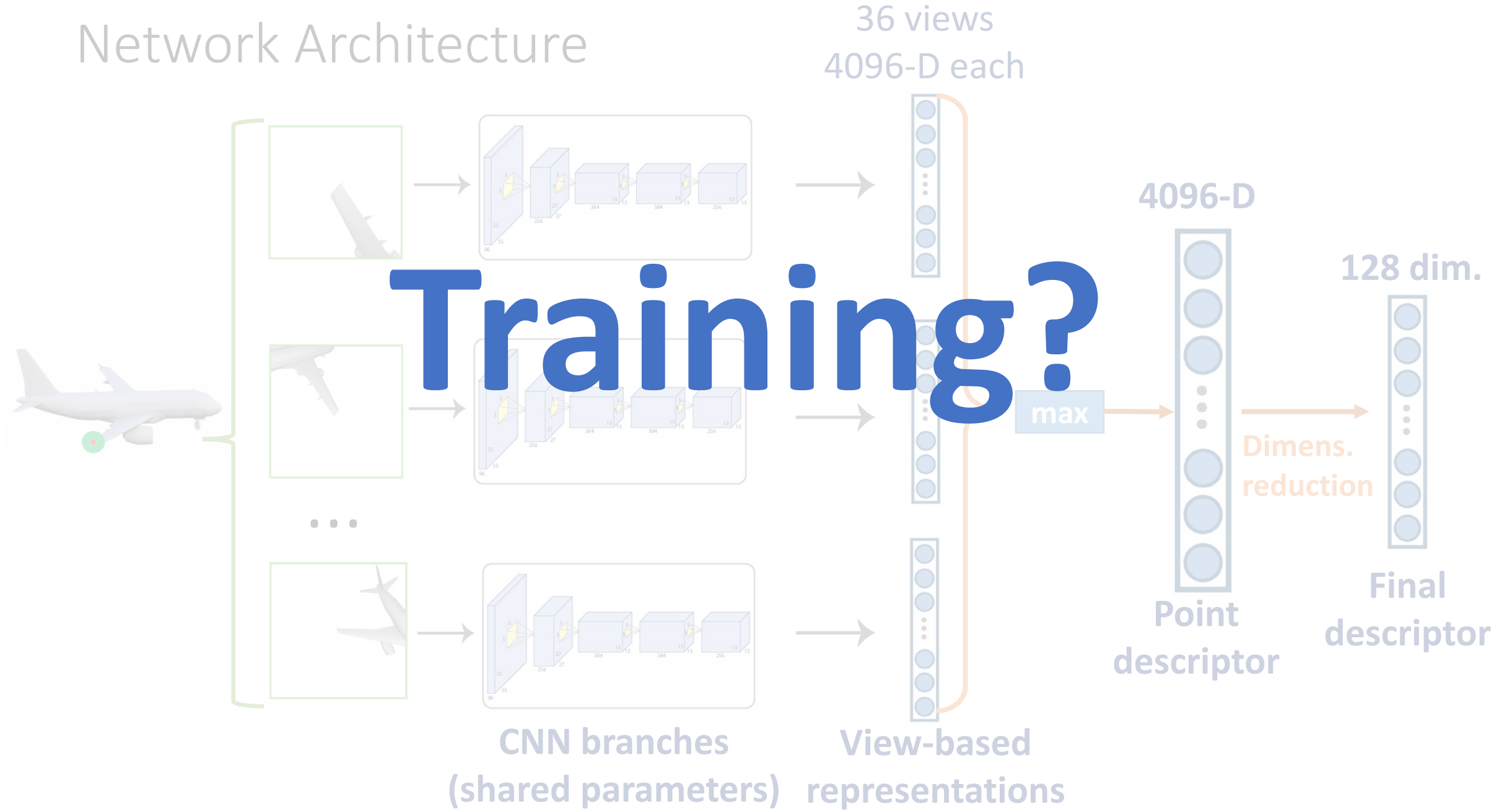
Network Architecture



Network Architecture



Network Architecture

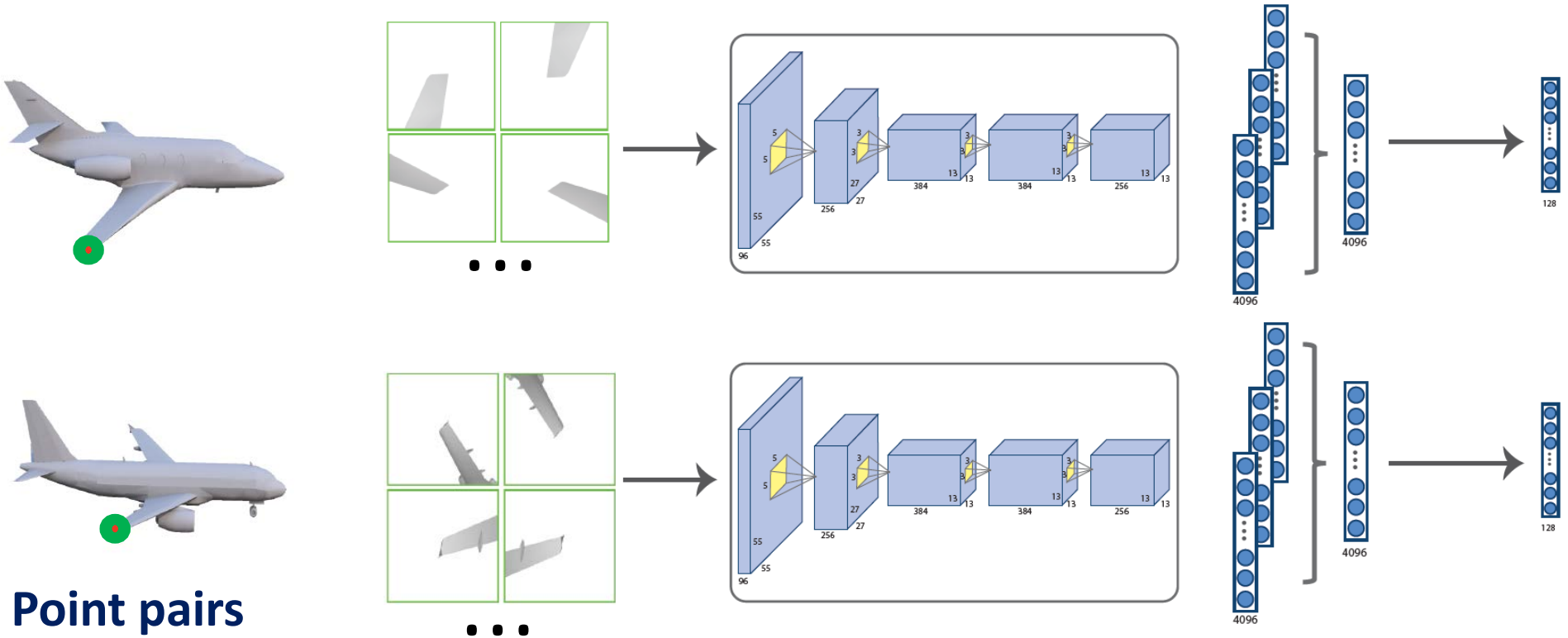


Training?

Training

Point pairs from two shapes

Training

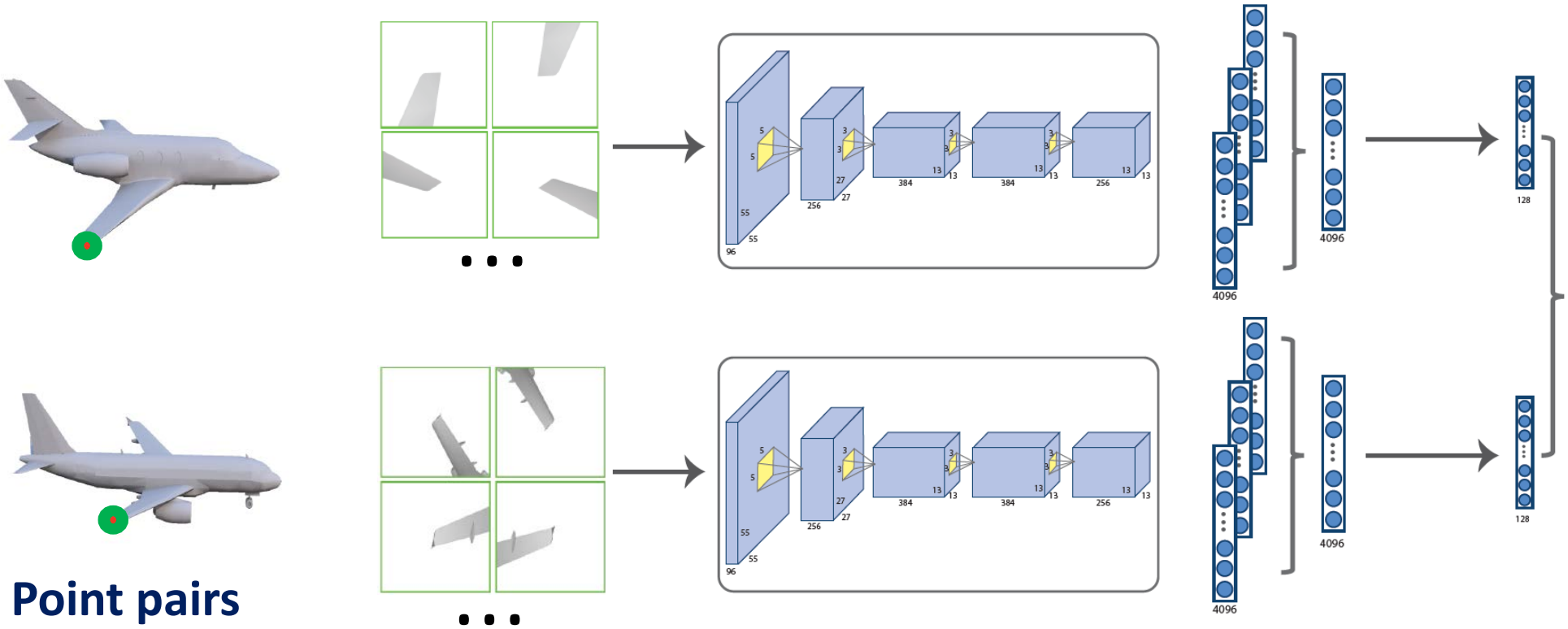


Point pairs from two shapes

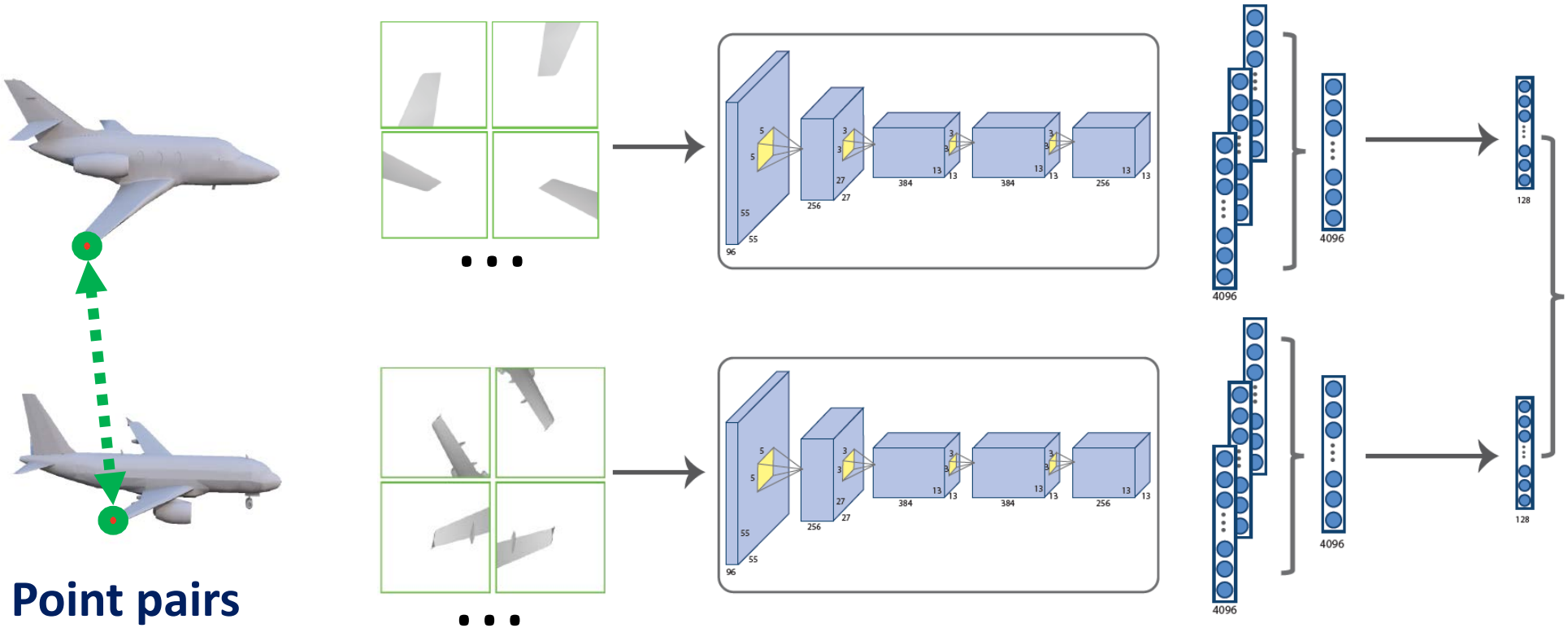
Local rendered views

“Siamese” LMVCNNs processing each point

Training


Loss function
$L(w)$
Point pairs from two shapes
Local rendered views
“Siamese” LMVCNNs processing each point

Training



Point pairs from two shapes

Local rendered views

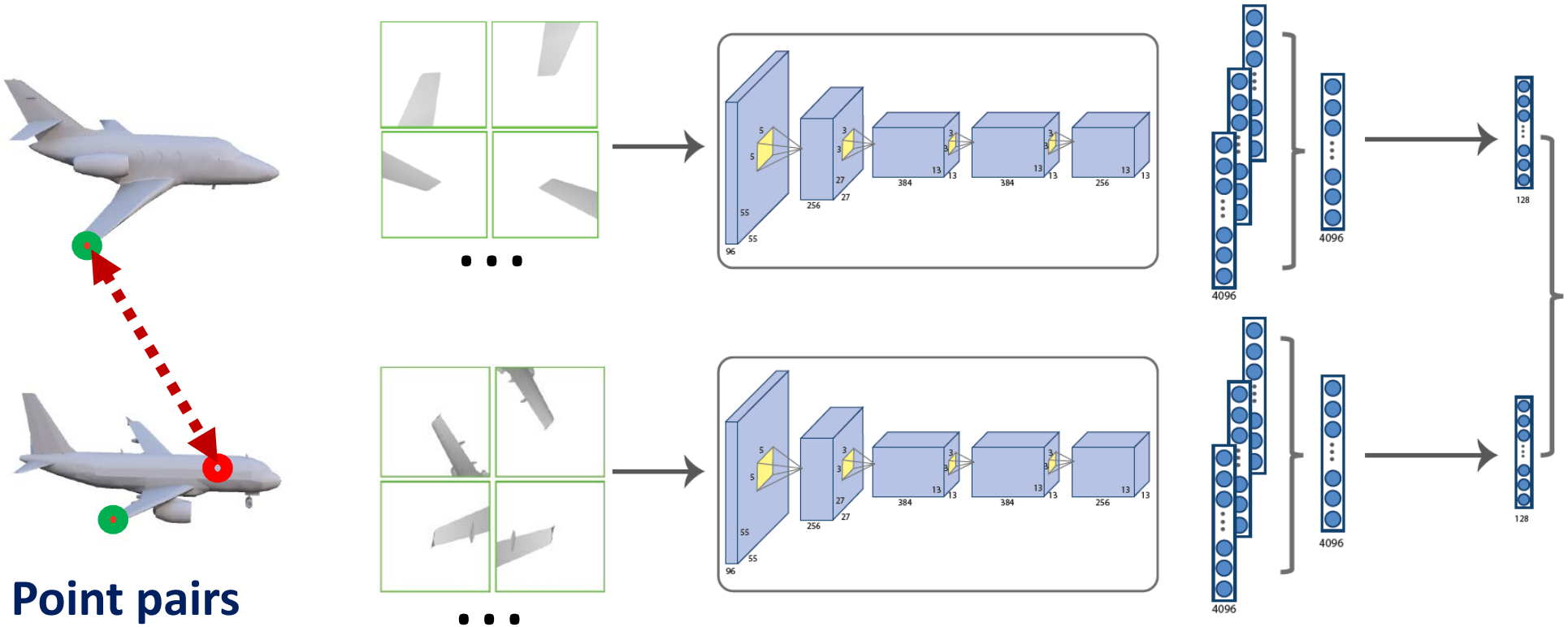
“Siamese” LMVCNNs processing each point

Loss function $L(w)$

$$L(w) = \sum_{\text{similar point pairs (a,b)}} D^2(X_a, X_b)$$

Training

Contrastive loss: (Hadsell et al, 2006)



Loss function $L(w)$

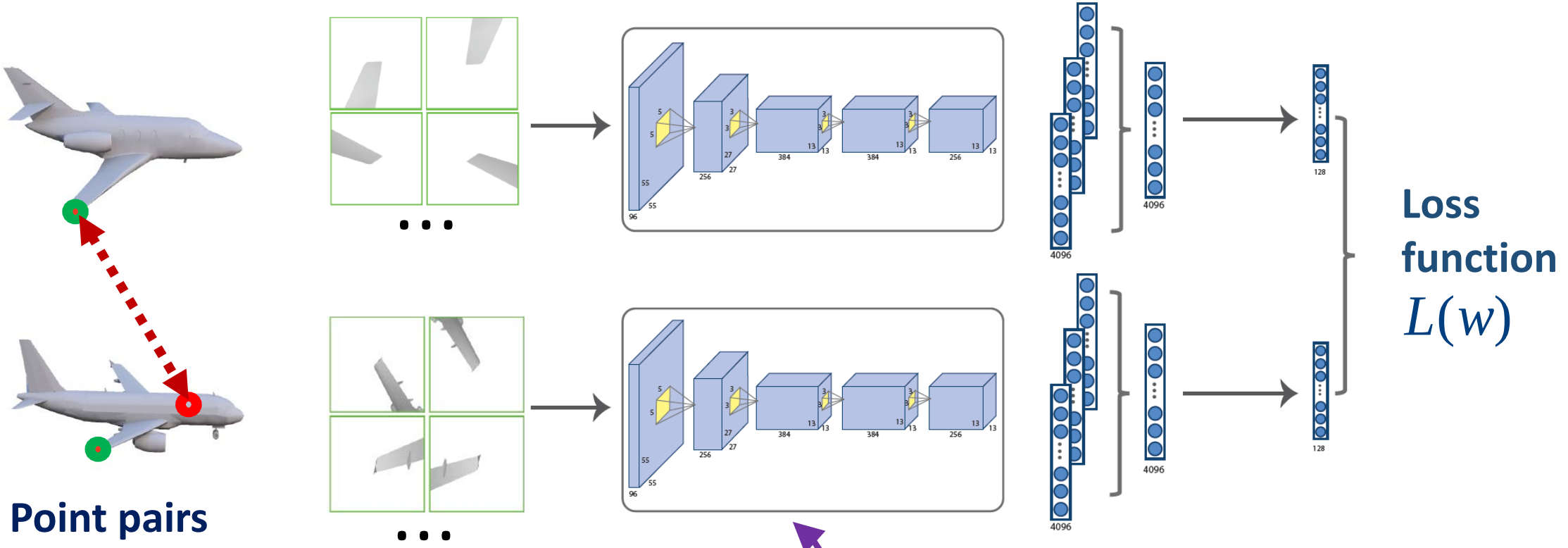
Point pairs from two shapes

Local rendered views

"Siamese" LMVCNNs processing each point

$$L(w) = \sum_{\text{similar point pairs (a,b)}} D^2(X_a, X_b) + \sum_{\text{dissimilar point pairs (a,c)}} \max(\text{margin} - D(X_a, X_c), 0)^2$$

Training


Point pairs from two shapes
Local rendered views
“Siamese” LMVCNNs processing each point
Loss function
$L(w)$
Initialize filters from their pre-trained values on ImageNet!

Training Dataset: Part Correspondences

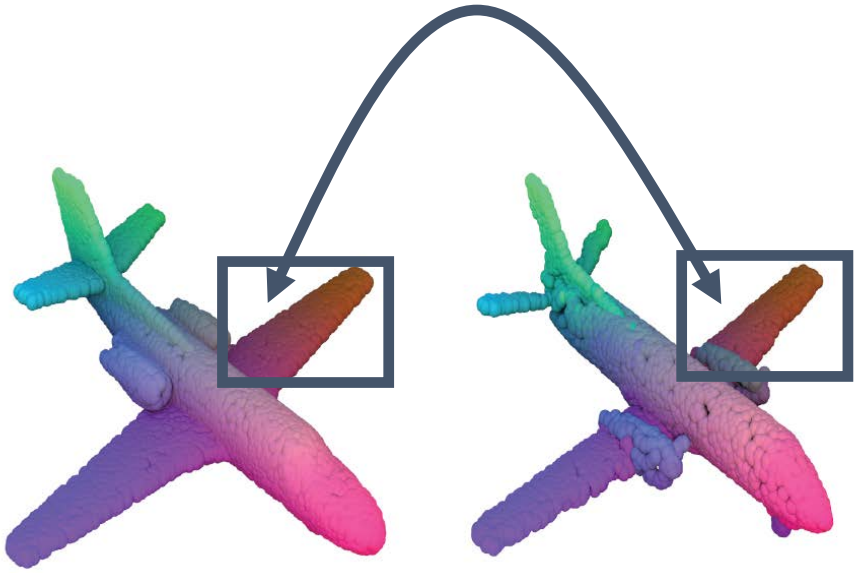
ShapeNetSem: 16 categories, 5K shapes **segmented into labeled parts**

[Yi et al. 2016]

Training Dataset

Non-rigid alignment between parts with the same **semantic label**

=> pick nearest point pairs

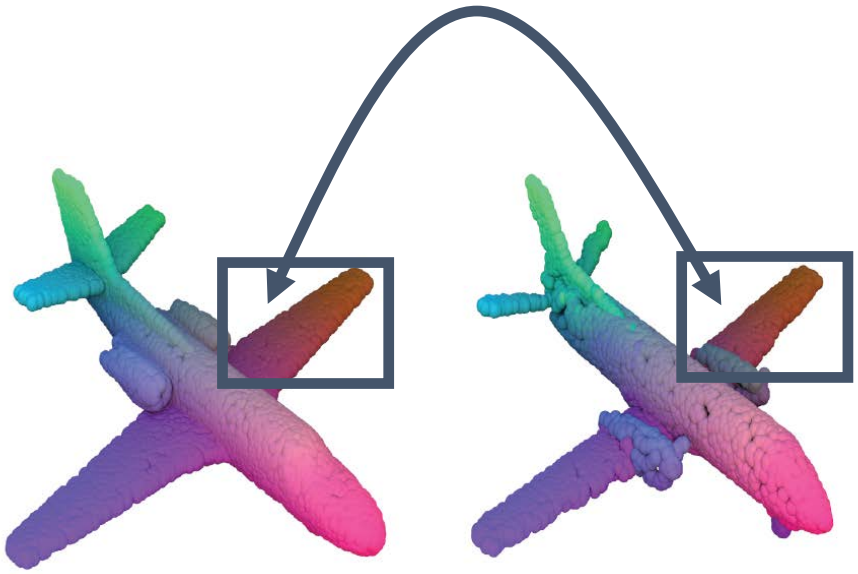


(corresponding points have same color)

Training Dataset

Non-rigid alignment between parts with the same **semantic label**

=> pick nearest point pairs



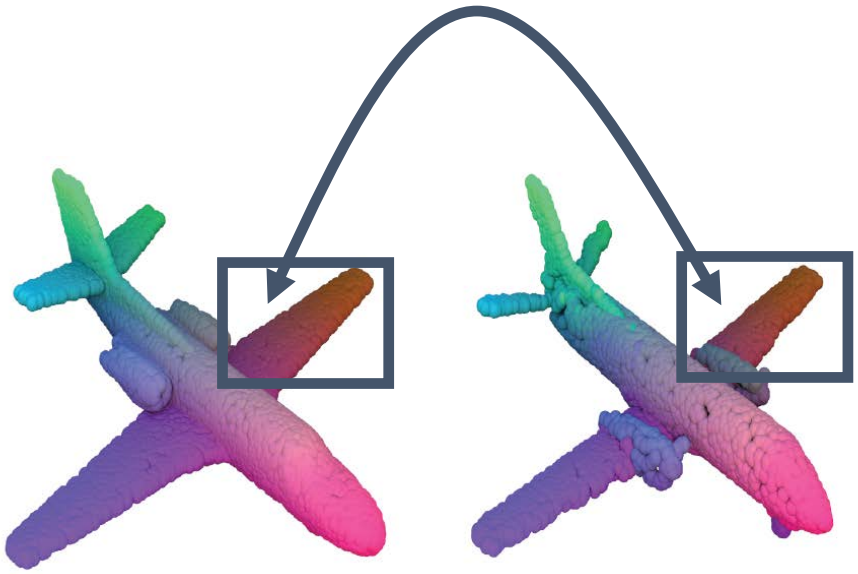
(corresponding points have same color)

ShapeNetCore Category	# shapes used	# aligned shape pairs	# corresponding point pairs
Airplane	500	9699	97.0M
Bag	76	1510	15.1M
Cap	55	1048	10.5M
Car	500	10000	100.0M
Chair	500	9997	100.0M
Earphone	69	1380	13.8M
Guitar	500	9962	99.6M
Knife	392	7821	78.2M
Lamp	500	9930	99.3M
Laptop	445	8880	88.8M
Motorbike	202	4040	40.4M
Mug	184	3680	36.8M
Pistol	275	5500	55.0M
Rocket	66	1320	13.2M
Skateboard	152	3032	30.3M
Table	500	9952	99.5M

Training Dataset

Non-rigid alignment between parts with the same **semantic label**

=> pick nearest point pairs



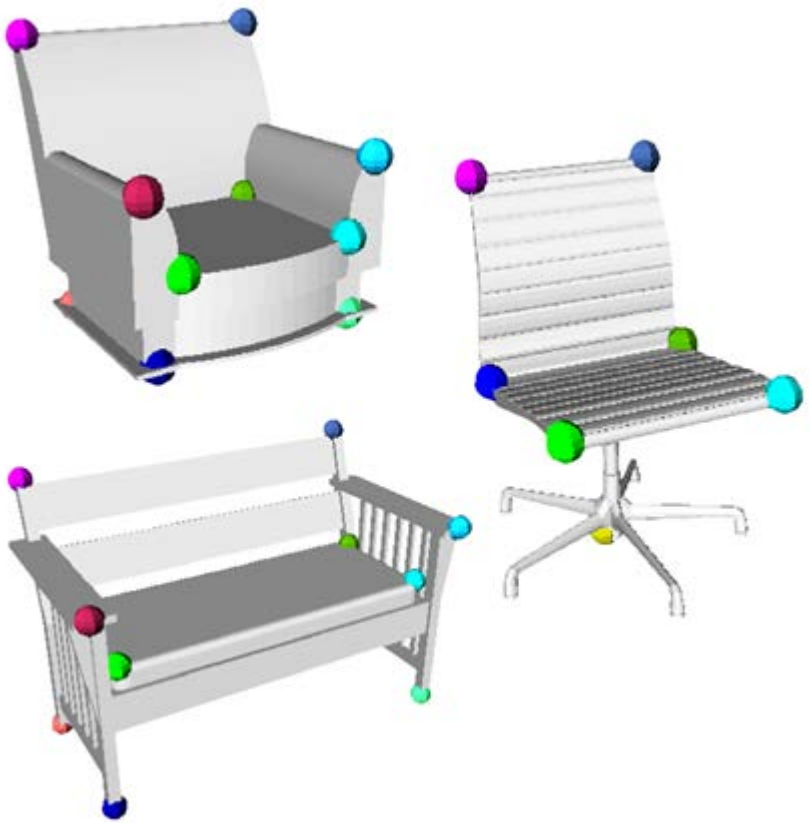
(corresponding points have same color)

ShapeNetCore Category	# shapes used	# aligned shape pairs	# corresponding point pairs
Airplane	500	9699	97.0M
Bag	76	1510	15.1M
Cap	55	1048	10.5M
Car			.0M
Chair			.0M
Earpho			8M
Guita			6M
Knife			2M
Lamp			3M
Lapto			8M
Motorbike	202	4040	40.4M
Mug	184	3680	36.8M
Pistol	275	5500	55.0M
Rocket	66	1320	13.2M
Skateboard	152	3032	30.3M
Table	500	9952	99.5M

977M corresponding point pairs

Evaluation & Applications

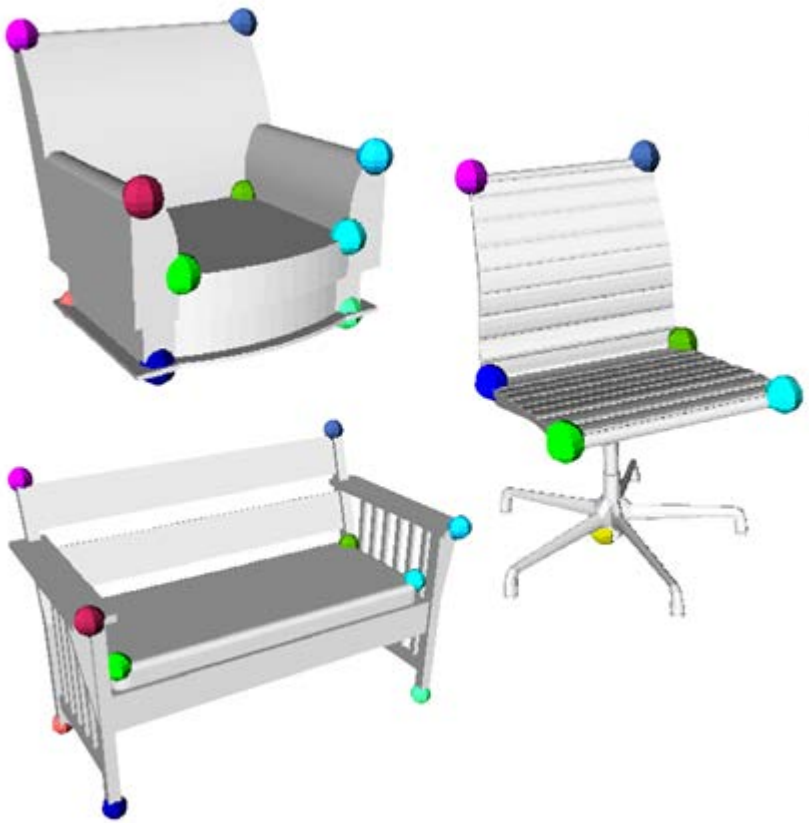
Evaluation



“BHCP” dataset: 4 categories, 404 shapes, annotated with 6-12 **corresponding feature points**

[Kim et al. 2013]

Evaluation

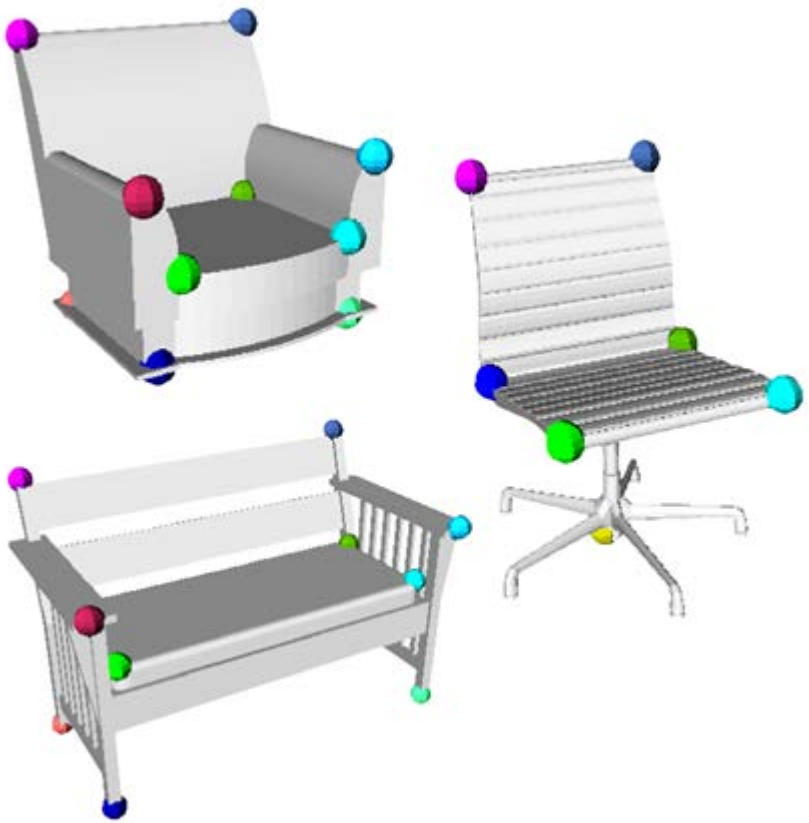


“BHCP” dataset: 4 categories, 404 shapes, annotated with 6-12 **corresponding feature points**

+ applied a **random 3D rotation** to each shape

[Kim et al. 2013]

Evaluation



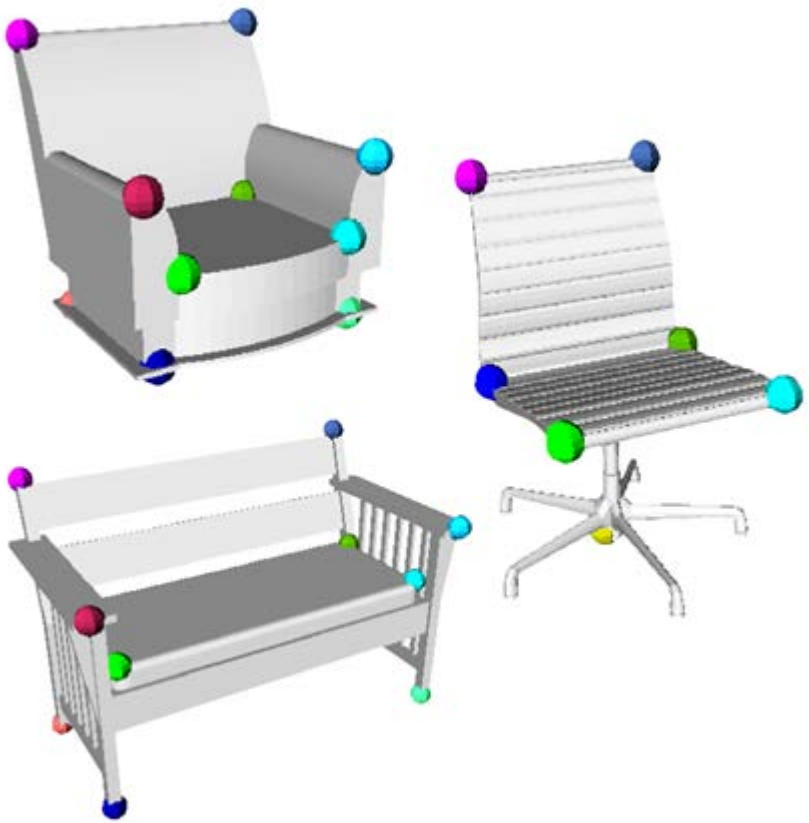
[Kim et al. 2013]

“BHCP” dataset: 4 categories, 404 shapes, annotated with 6-12 **corresponding feature points**

+ applied a **random 3D rotation** to each shape

BHCP shapes **not** included in our training datasets.

Evaluation



[Kim et al. 2013]

“BHCP” dataset: 4 categories, 404 shapes, annotated with 6-12 **corresponding feature points**

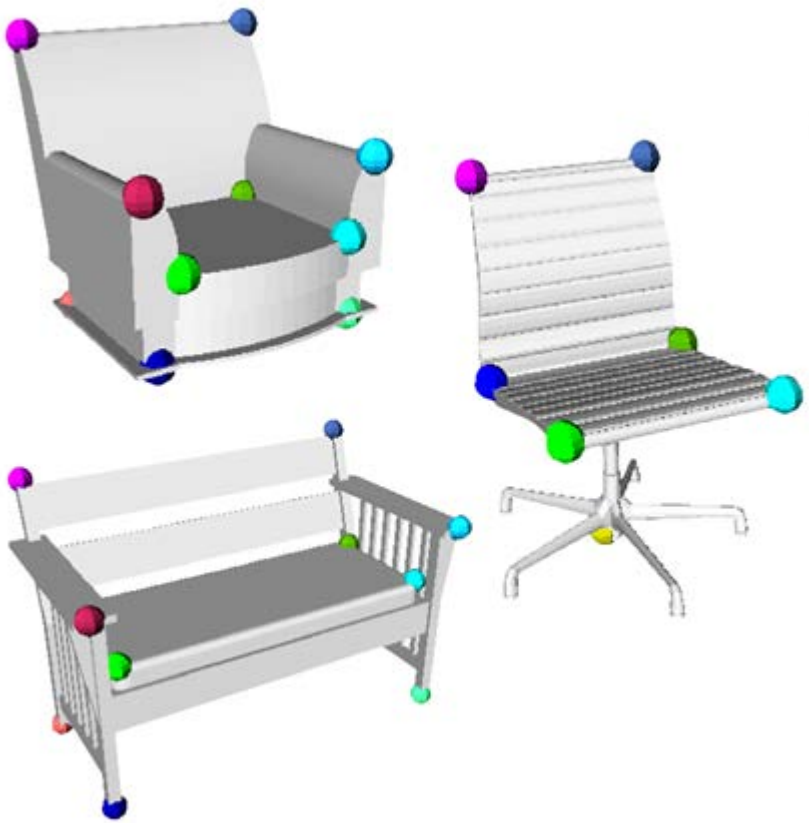
+ applied a **random 3D rotation** to each shape

BHCP shapes **not** included in our training datasets.

Three conditions:

1. Train on one ShapeNet class / test on corresponding BHCP class

Evaluation



[Kim et al. 2013]

“BHCP” dataset: 4 categories, 404 shapes, annotated with 6-12 **corresponding feature points**

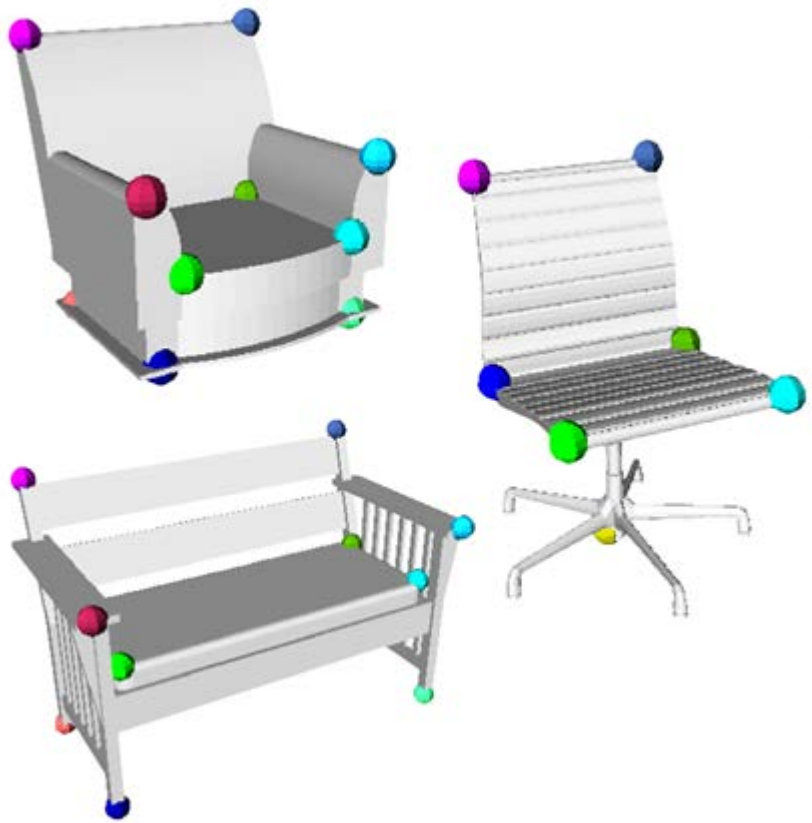
+ applied a **random 3D rotation** to each shape

BHCP shapes **not** included in our training datasets.

Three conditions:

1. Train on one ShapeNet class / test on corresponding BHCP class
2. Train on all ShapeNet classes / test on BHCP

Evaluation



[Kim et al. 2013]

"BHCP" dataset: 4 categories, 404 shapes, annotated with 6-12 **corresponding feature points**

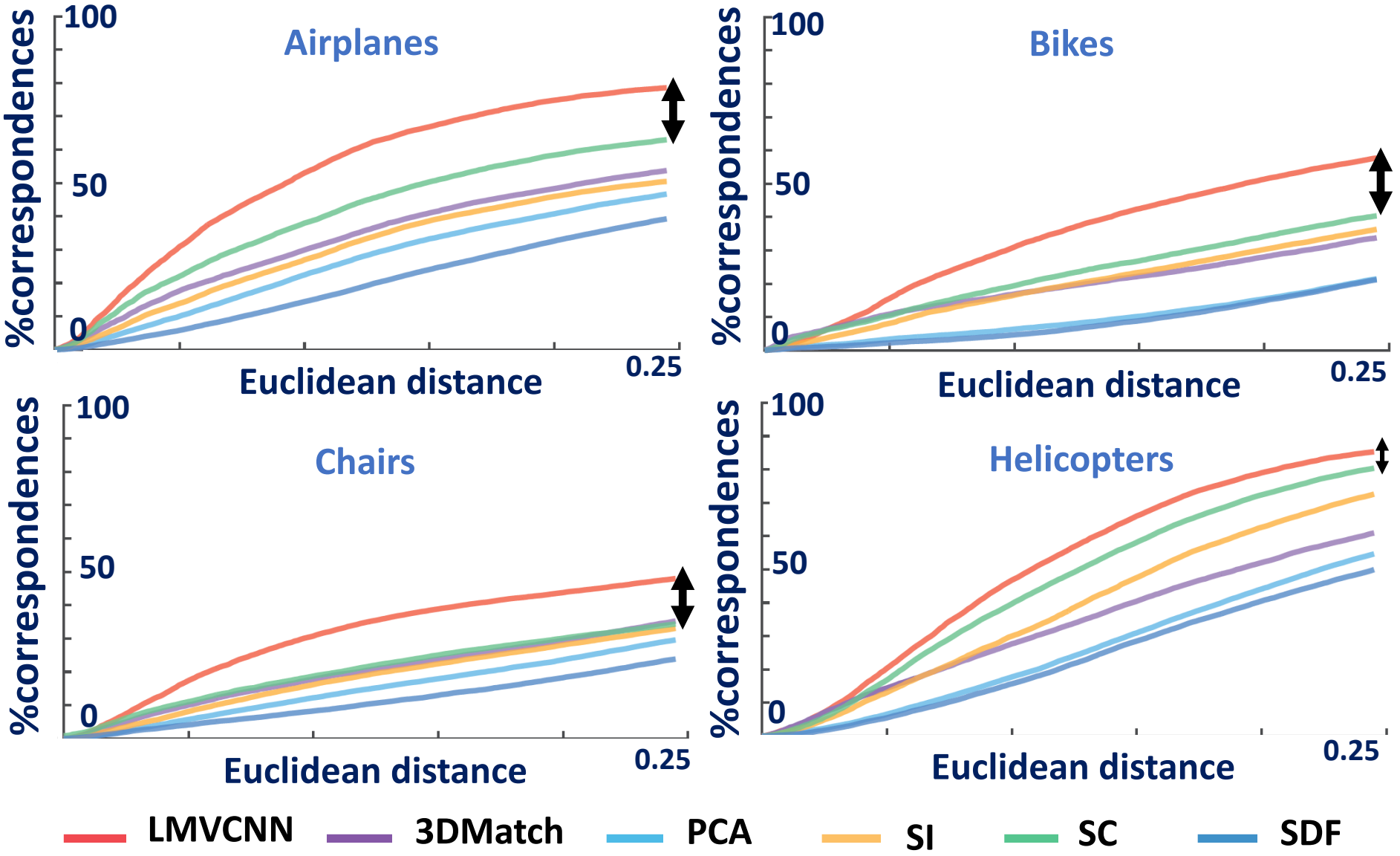
+ applied a **random 3D rotation** to each shape

BHCP shapes **not** included in our training datasets.

Three conditions:

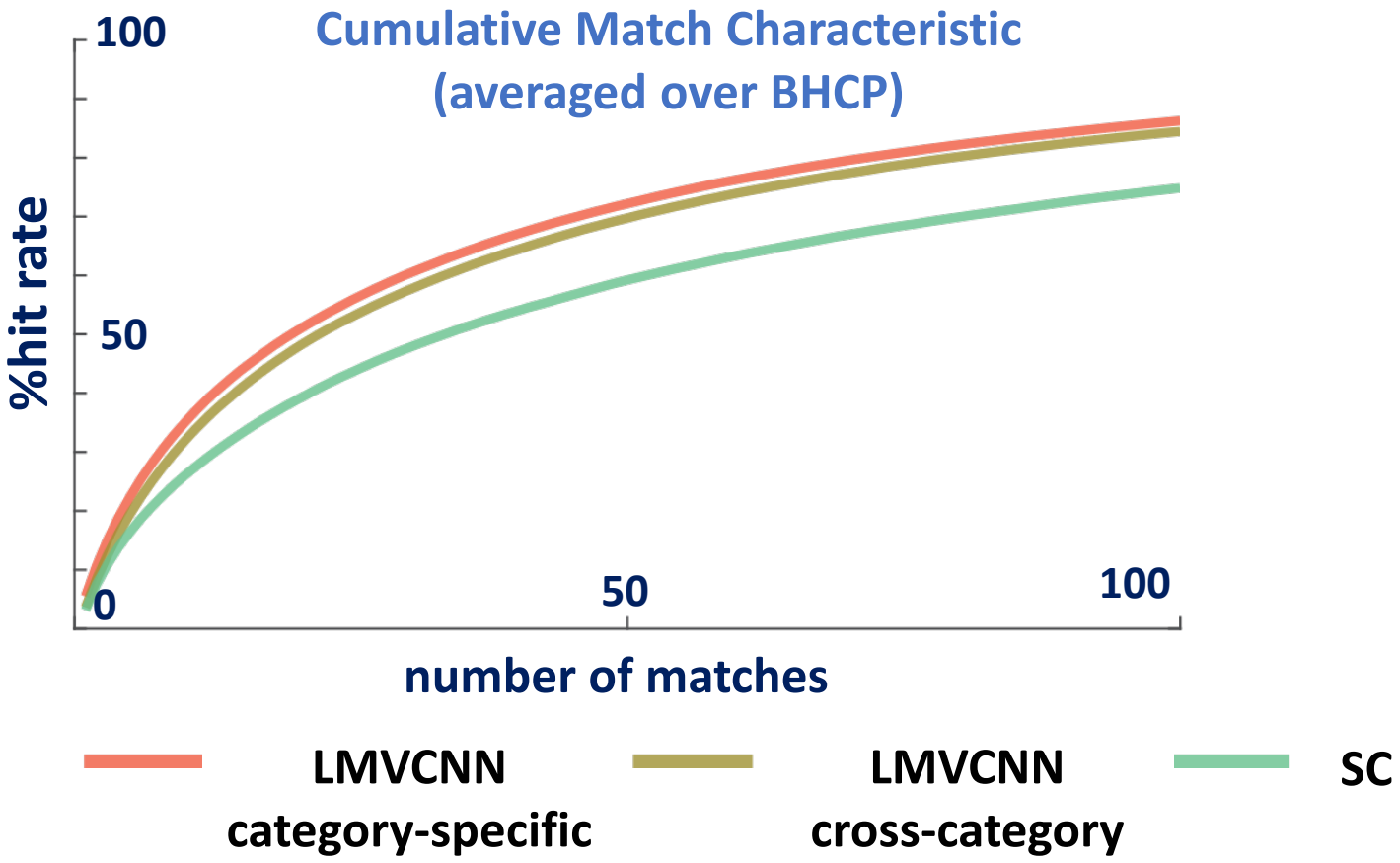
1. Train on one ShapeNet class / test on corresponding BHCP class
2. Train on all ShapeNet classes / test on BHCP
3. Train on ShapeNet classes **different** from BHCP

1. Train on one ShapeNet class / test on corresponding BHCP class

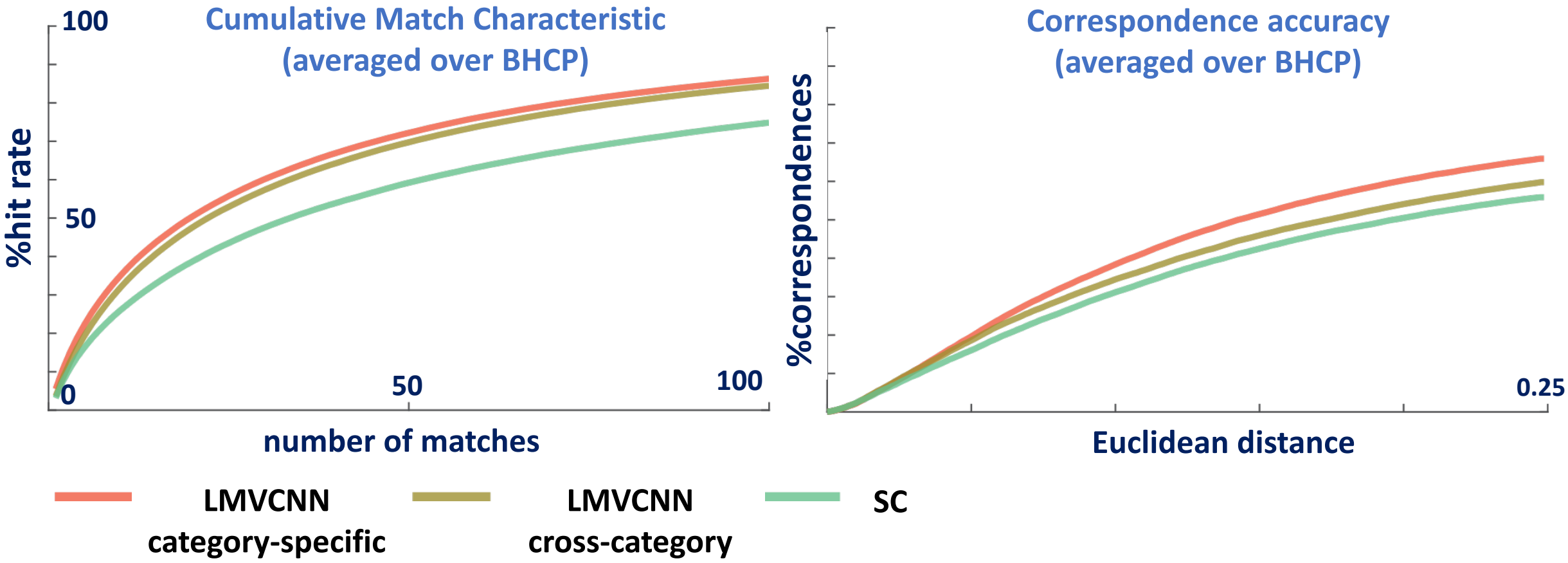


LMVCNN yields an average **+10% improvement** in correspondence accuracy

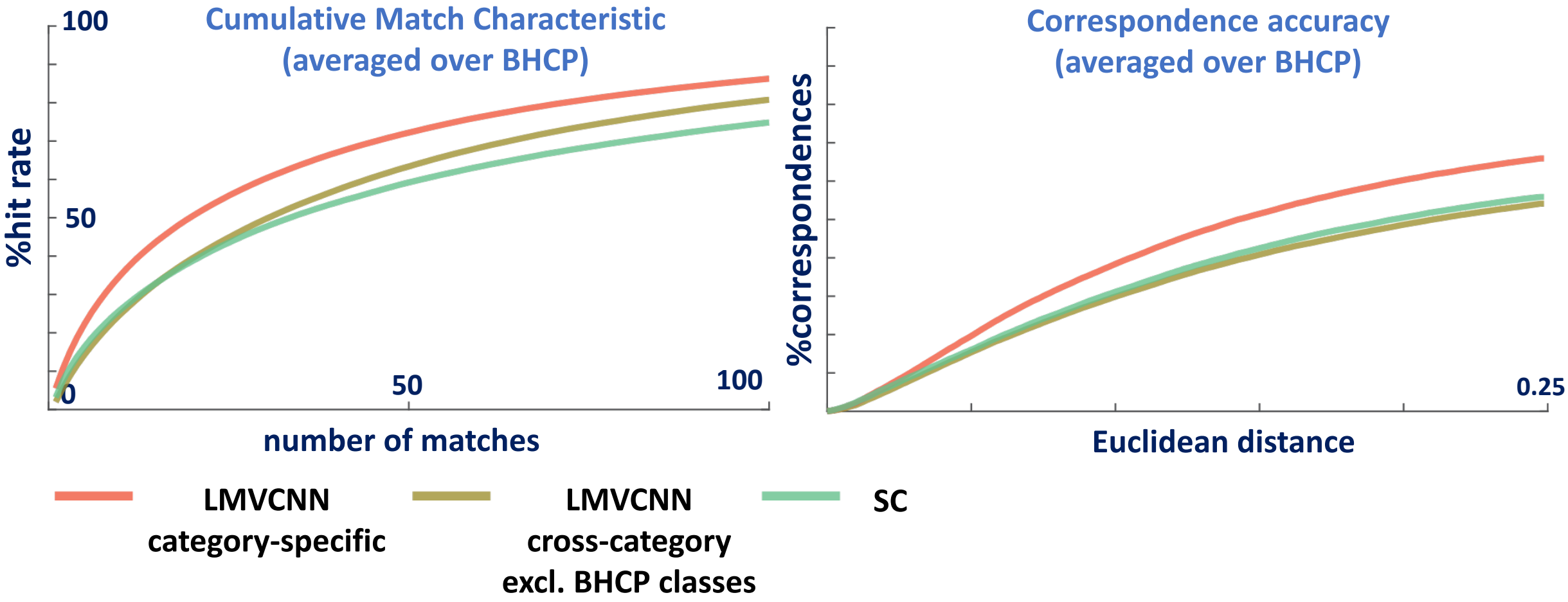
2. Train on all ShapeNet classes / test on BHCP



2. Train on all ShapeNet classes / test on BHCP



3. Train on ShapeNet classes **different** from BHCP



Applications: partial scan-to-shape matching

Trained on ShapeNet models => test on scans

(similar colors correspond to points with similar descriptors)

Note: point clouds are rendered using a sphere per point

Applications: partial scan-to-shape matching

Trained on ShapeNet models => test on scans

(similar colors correspond to points with similar descriptors)

Note: point clouds are rendered using a sphere per point

Applications: predicting affordance regions

Fine-tuned on [Kim et al. ’14]’s contact point dataset

Summary

- Point-based descriptor learning based on a **convnet operating on multi-scale local surface view projections**

Summary

- Point-based descriptor learning based on a **convnet operating on multi-scale local surface view projections**

- Leverage two **massive large sources of data** to train our network (Imagenet & correspondences we generated from segmented ShapeNet)

Summary

- Point-based descriptor learning based on a **convnet operating on multi-scale local surface view projections**
- Leverage two **massive large sources of data** to train our network (Imagenet & correspondences we generated from segmented ShapeNet)
- **Can generalize to scans** & **classes not seen during training**

Limitations

- **Surface information can be lost** in projections

Limitations

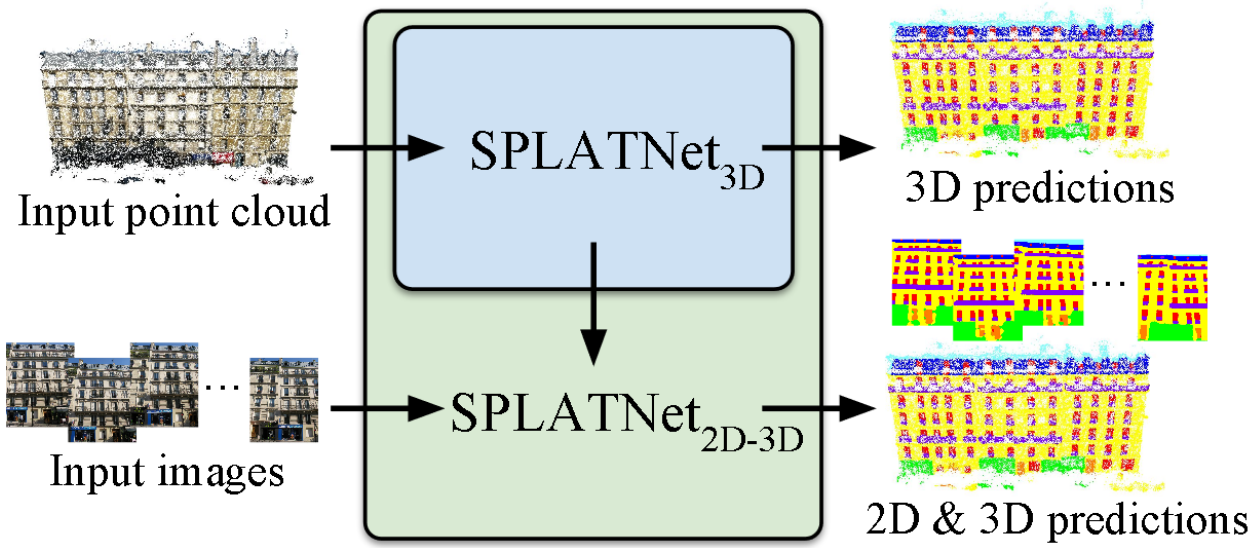
- **Surface information can be lost** in projections
- **Redundancy in processing** (same surface is visible from multiple views)

Limitations

- **Surface information can be lost** in projections
- **Redundancy in processing** (same surface is visible from multiple views)
- **Max view pooling might cause some information loss**

Limitations

- **Surface information can be lost** in projections
- **Redundancy in processing** (same surface is visible from multiple views)
- **Max view pooling might cause some information loss**
- Combine view-based with 3D-based nets, see **SplatNet**, Su et al., CVPR ‘18



Thank you!

Our project webpage with source code & dataset:

http://people.cs.umass.edu/~hbhuang/local_mvcnn/