Style Compatibility For 3D Furniture Models

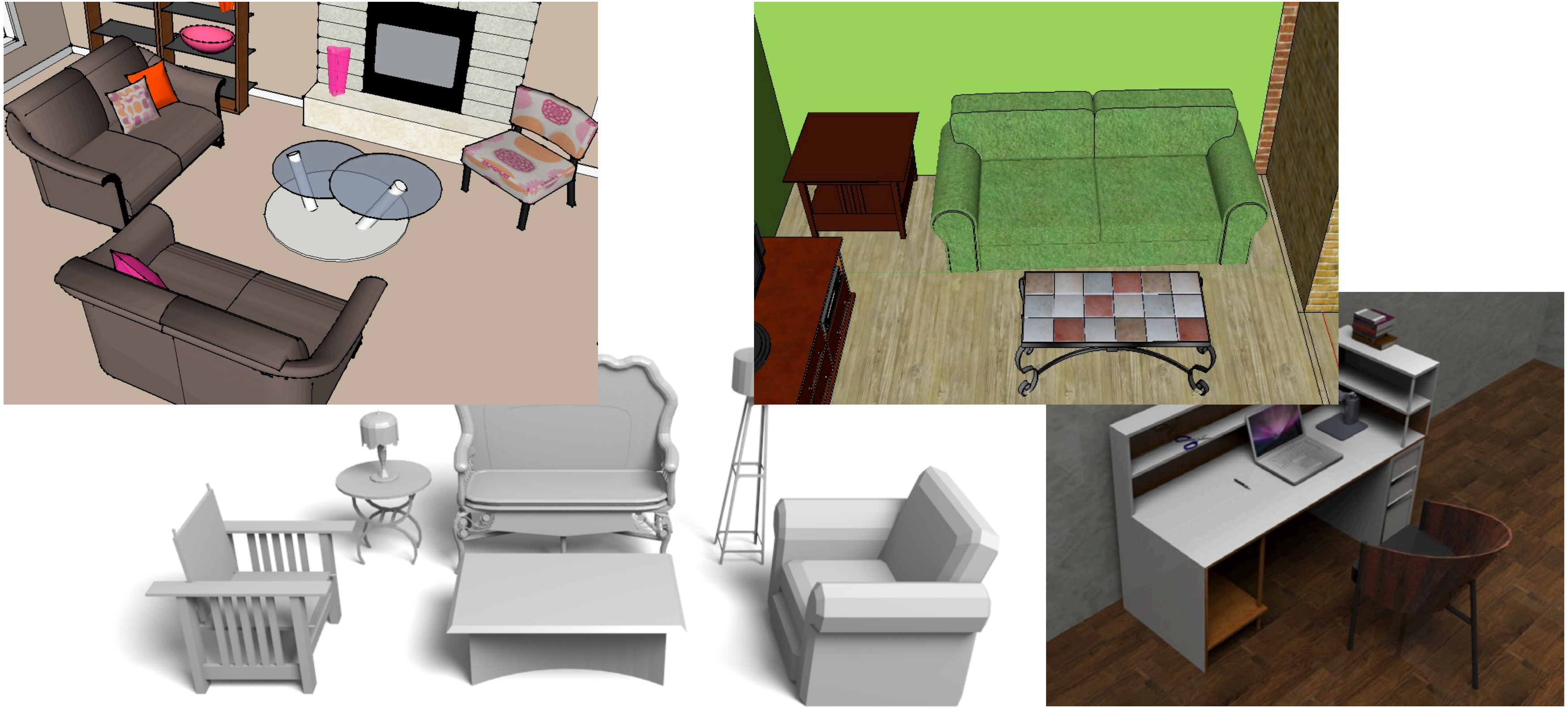
Tianqiang Liu[1] Aaron Hertzmann[2]

Wilmot Li[2] Thomas Funkhouser[1]

[1]Princeton University

[2]Adobe Research

Motivation



Motivation

Motivation

Stylistically incompatible

Motivation

Stylistically compatible

Goal

Modeling pairwise style compatibility

How likely is it that a person would put these two furniture pieces together, when furnishing an apartment?

Goal

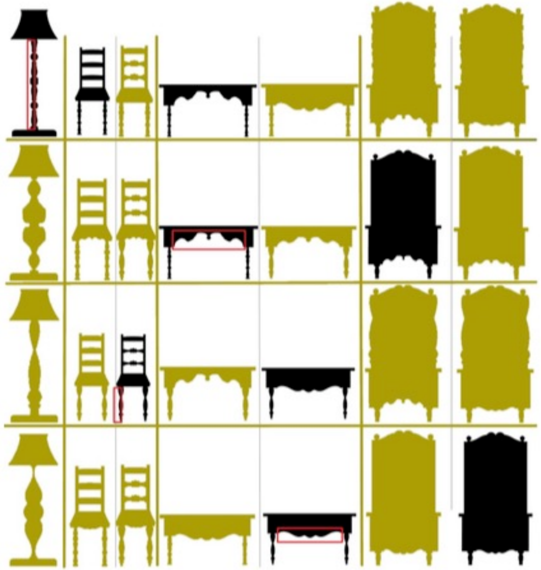
 

Extract feature vectors

$$d(x_i, x_j) = Scalar$$

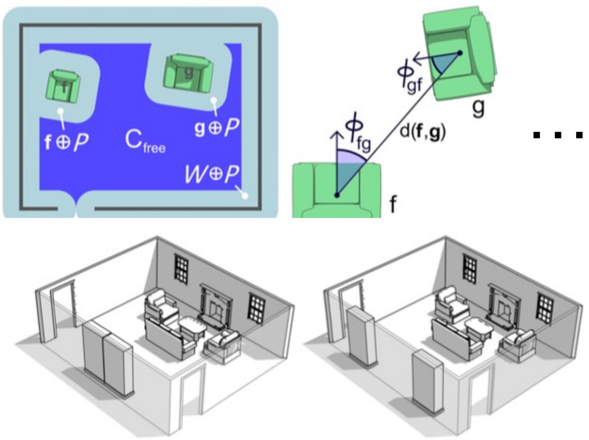
Previous work – shape style

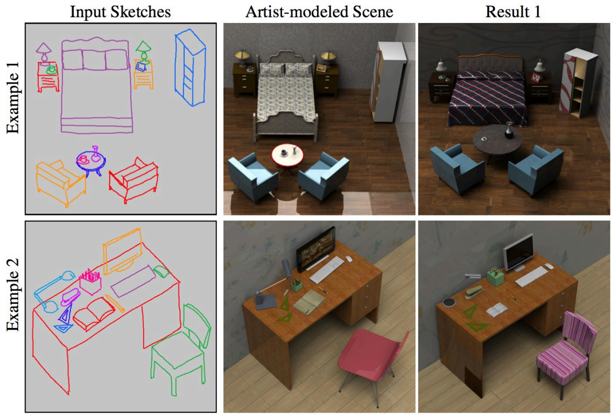
[Xu et al. 2010]



[Li et al. 2013]

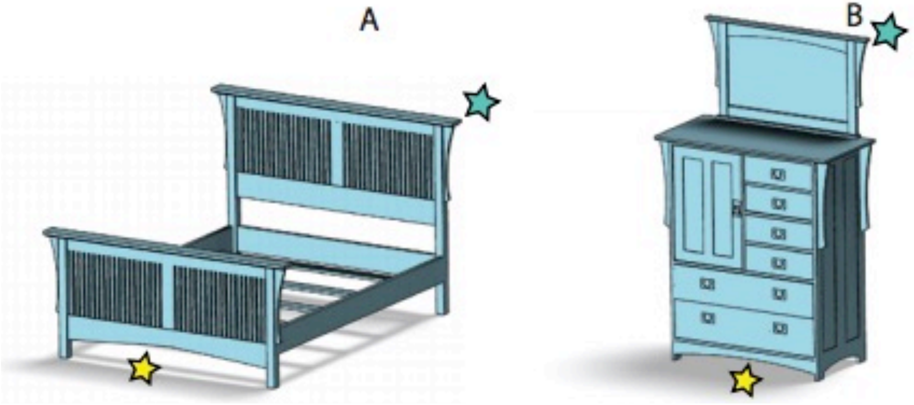
Previous work – virtual world synthesis





[Merrell et al. 2011] [Fisher et al. 2012] [Xu et al. 2013]

Concurrent work – style similarity



[Lun et al. 2015]
(previous talk in this session)

Challenges

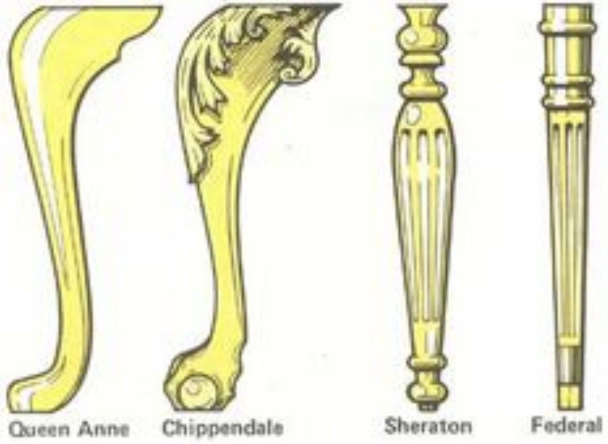
- Hard to design a hand-tuned function
- Coupled with functionality
- Requiring comparisons across object classes

Challenges

- Hard to design a hand-tuned function
- Coupled with functionality
- Requiring comparisons across object classes

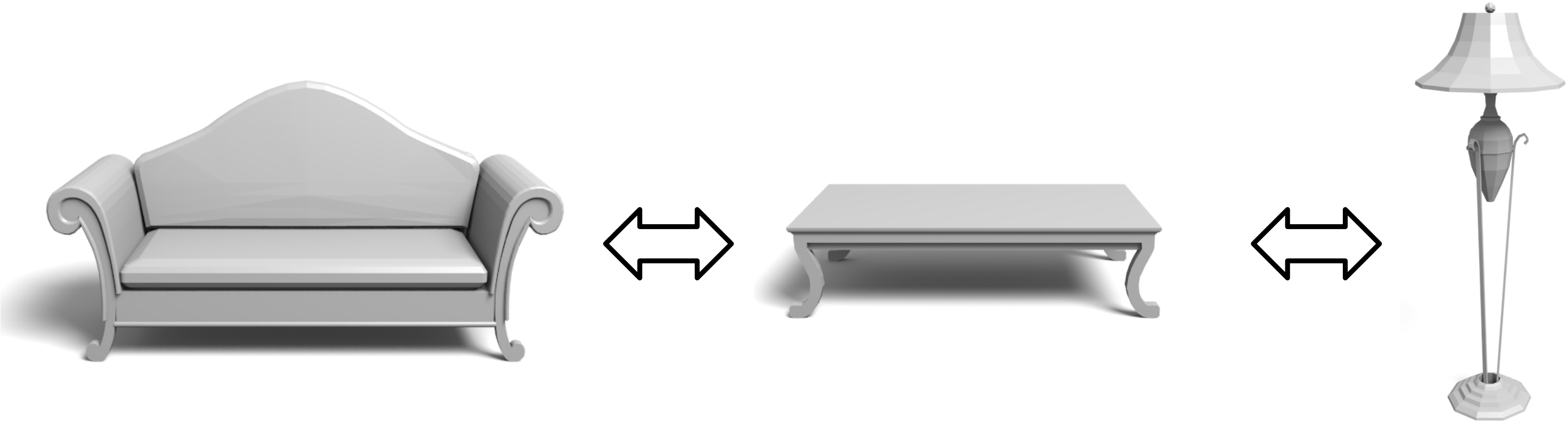
Challenges

- Hard to design a hand-tuned function
- Coupled with functionality
- Requiring comparisons across object classes



Challenges

- Hard to design a hand-tuned function
- Coupled with functionality
- Requiring comparisons across object classes

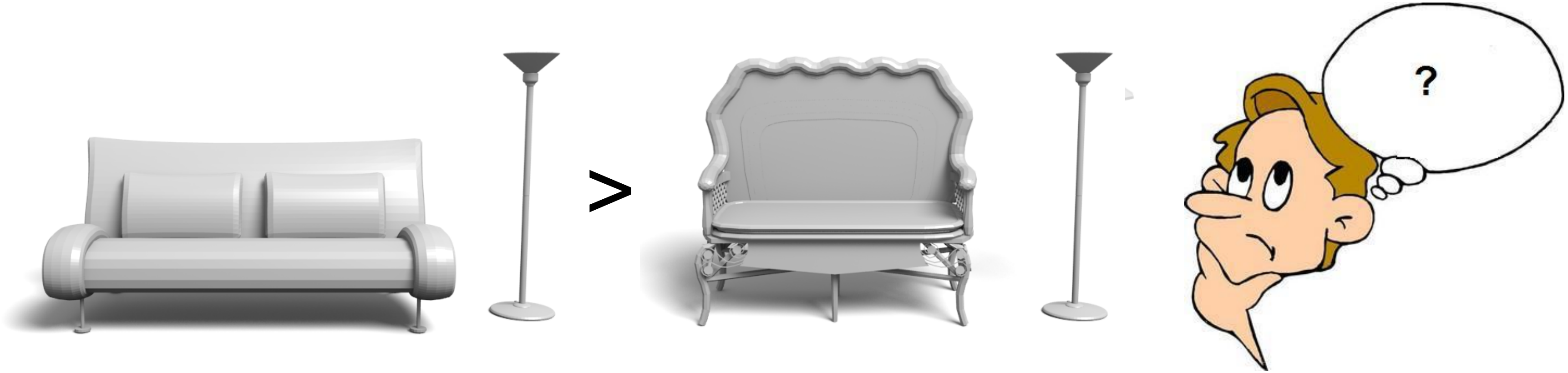


Key ideas

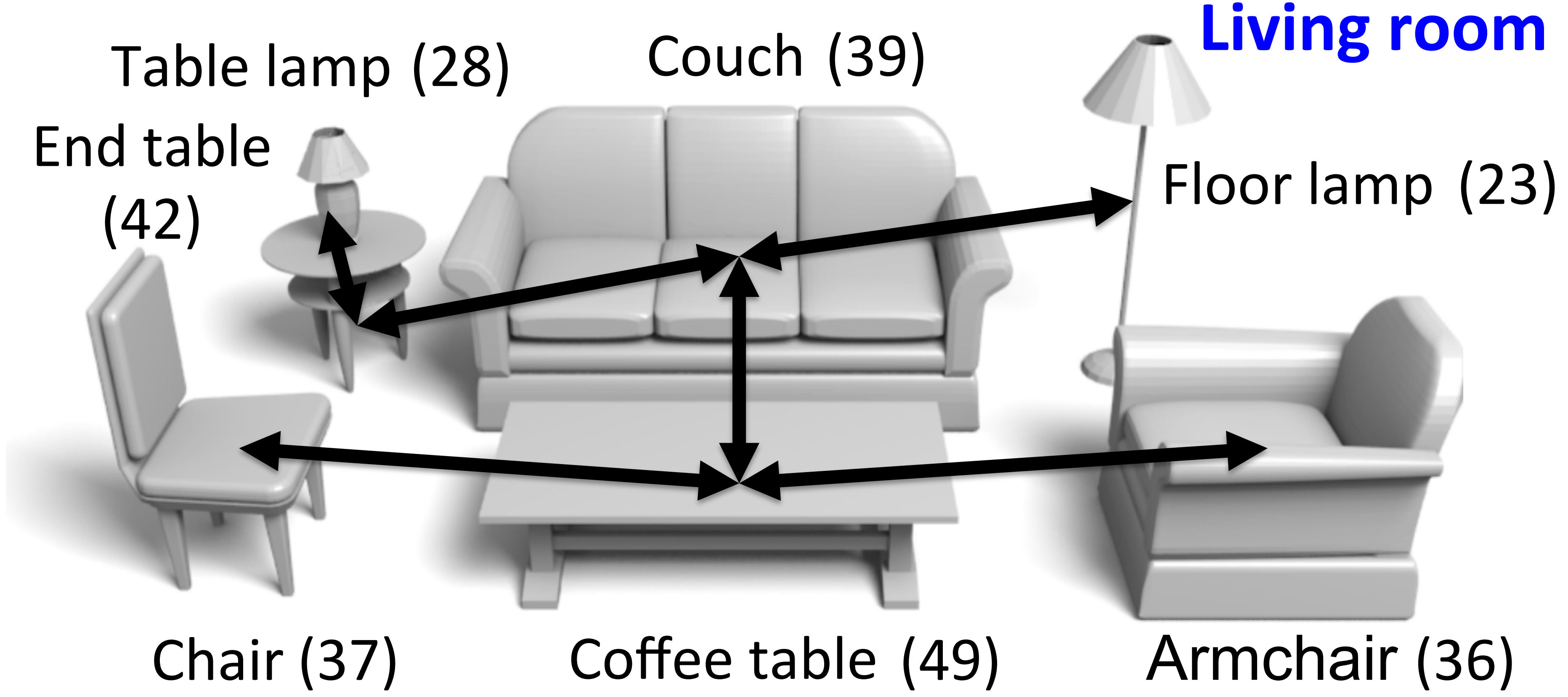
- Crowdsourcing compatibility preferences
- Part-based geometric features
- Learning object-class specific embeddings

Key ideas

- Crowdsourcing compatibility preferences
- Part-based geometric features
- Learning object-class specific embeddings

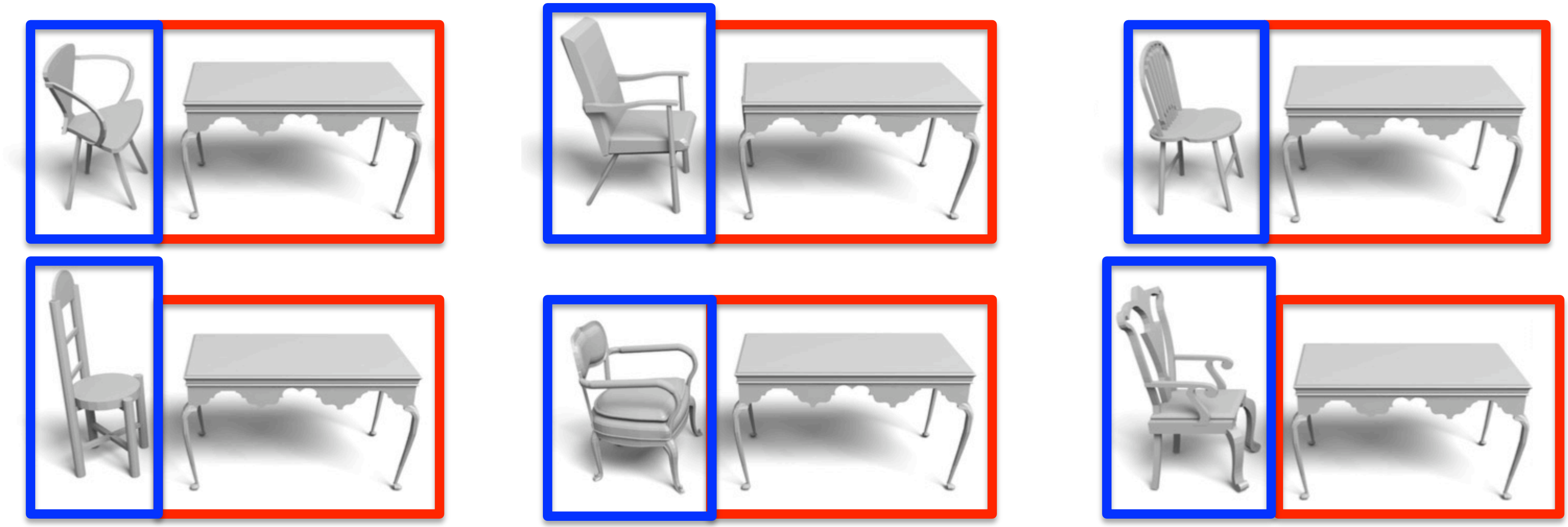


Crowdsourcing compatibility preferences



Crowdsourcing compatibility preferences

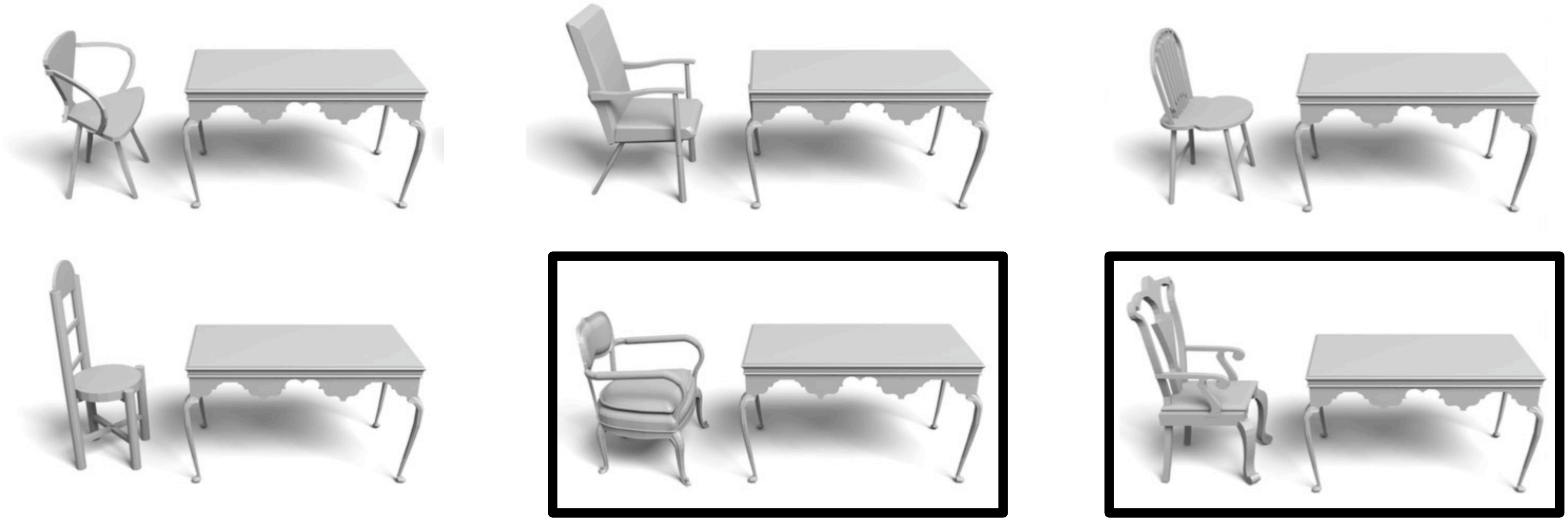
Design of user study [Wilber et al. 2014]

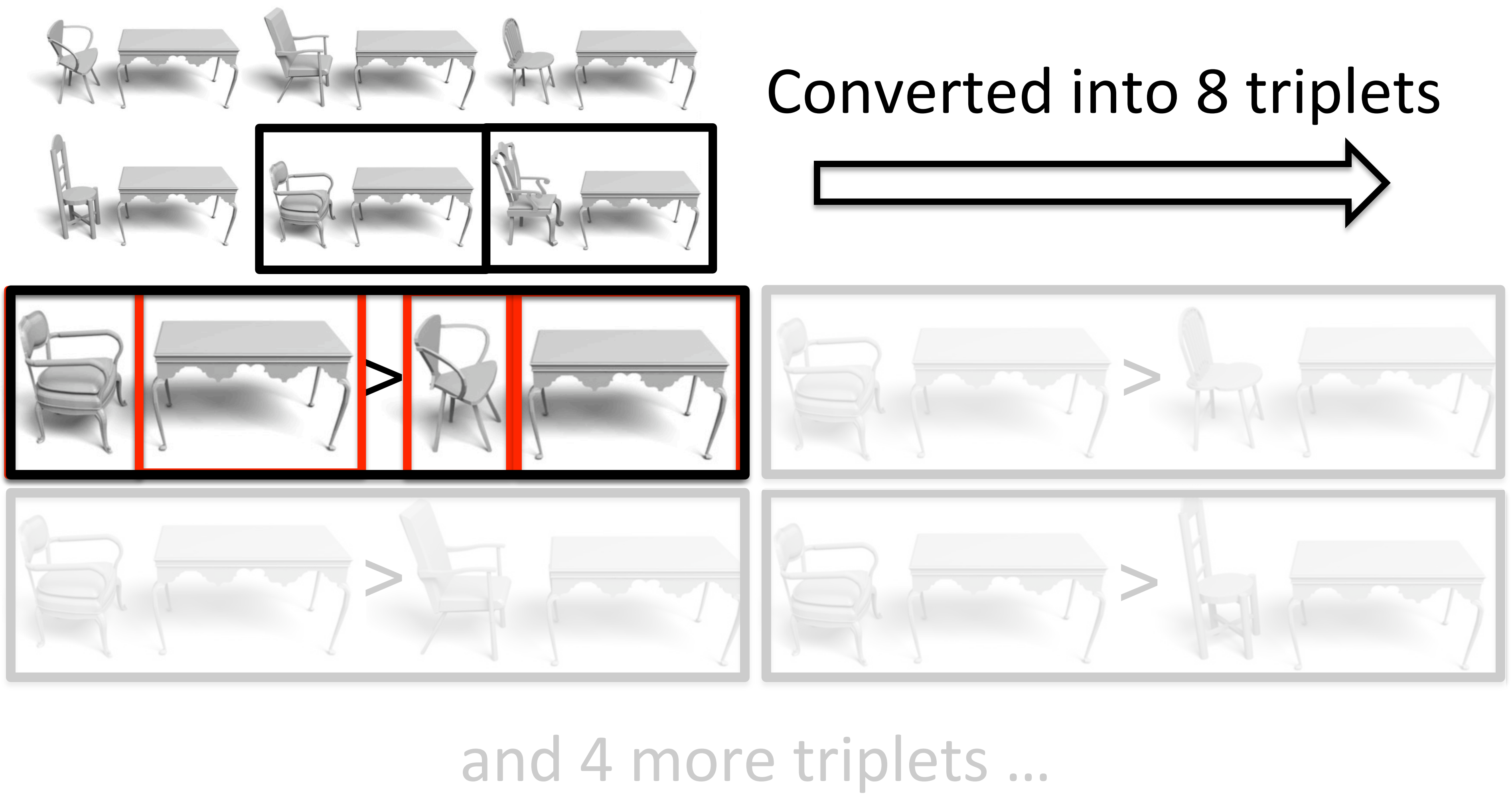


Please select the two most compatible pairs.

Crowdsourcing compatibility preferences

Rater's selection

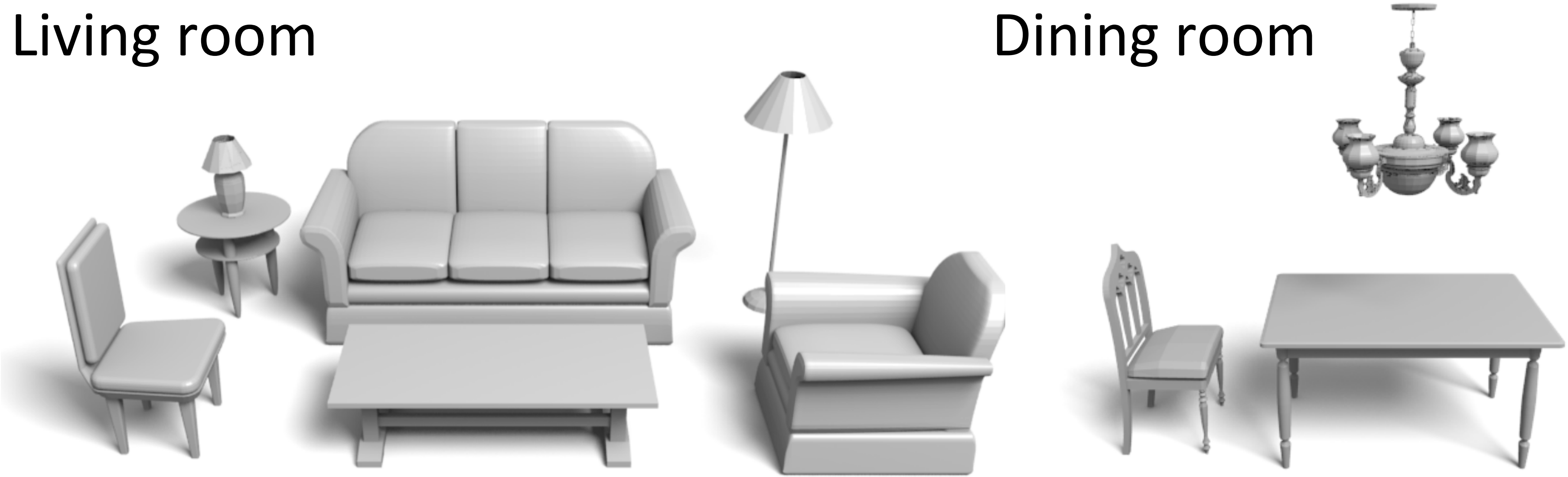



Converted into 8 triplets
>
>
>
>
and 4 more triplets ...

Crowdsourcing compatibility preferences

Living room

Dining room



Collected *63,800* triplets for living room

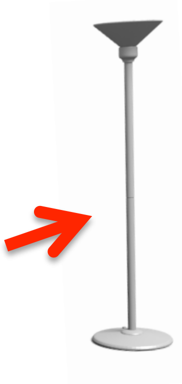
and *20,200* for dining room

Key ideas

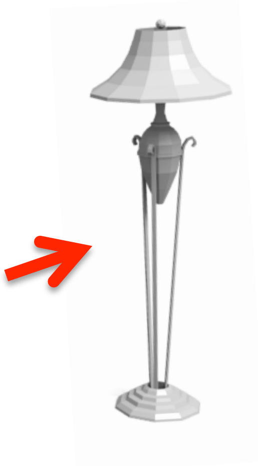
- Crowdsourcing compatibility preferences
- Part-aware geometric features
- Learning object-class specific embeddings

Part-aware geometric features

Contemporary



Antique

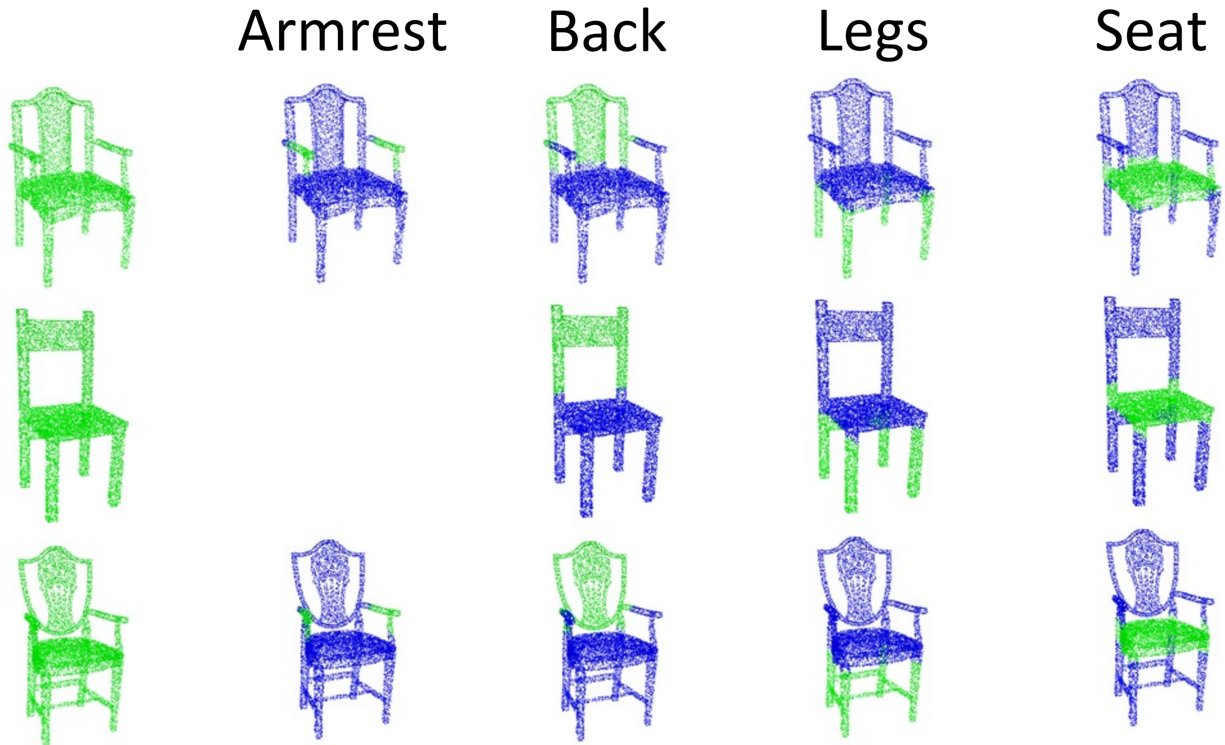


Part-aware geometric features

- Consistent segmentation
- Computing geometry features for each part
- Concatenating features of all parts

Part-aware geometric features

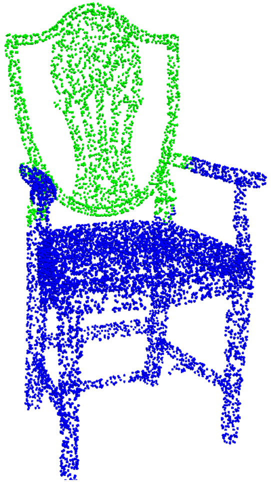
Step 1: Consistent segmentation [Kim et al. 2013]



Part-aware geometric features

Step 2: Computing geometry features for each part

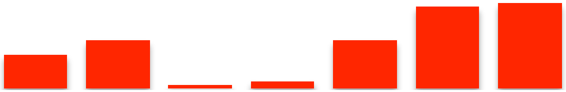
Back



Curvature histogram

Bounding box dimensions

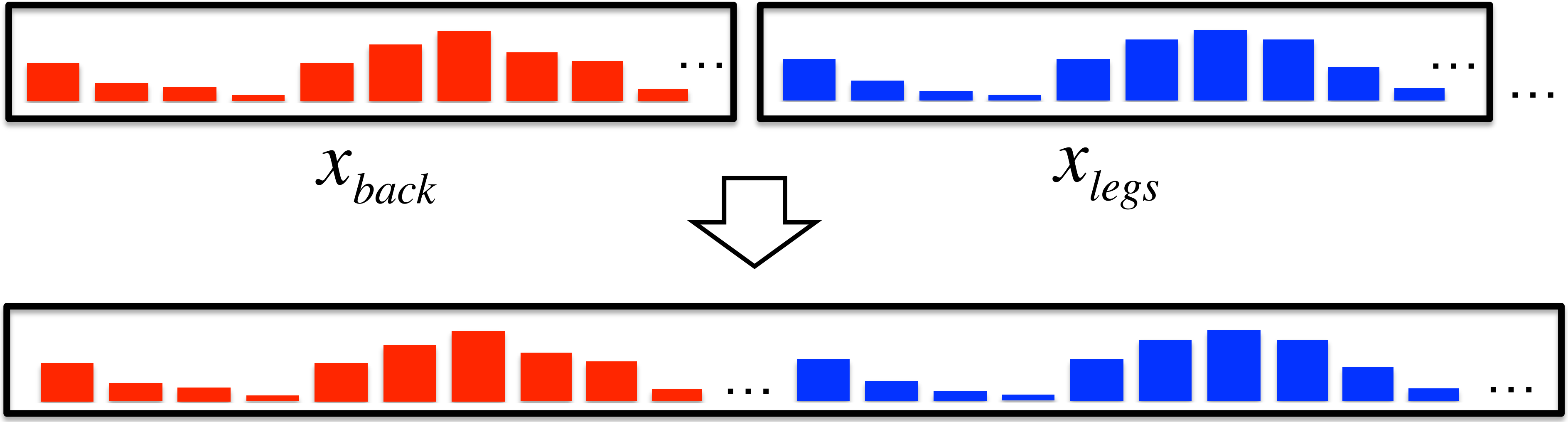
Shape diameter histogram



Normalized surface area

Part-aware geometric features

Step 3: Concatenating features of all parts



$$x = [x_{back}, x_{legs}, \ldots]$$

Key ideas

- Crowdsourcing compatibility preferences
- Part-aware geometric features
- Learning object-class specific embeddings

Learning object-class specific embeddings

Previous approach [Kulis 2012]: **Symmetric embedding**

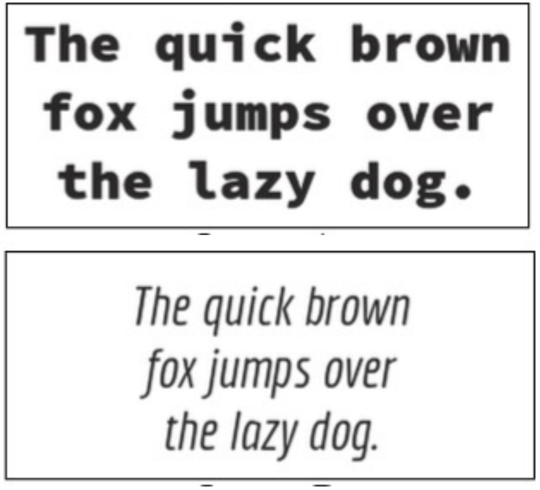
$$d_{symm}(x_i, x_j) = \left\| W(x_i - x_j) \right\|_2$$

d_{symm} is the compatibility distance

x_i, x_j are feature vectors of two shapes

Learning object-class specific embeddings

Previous approach [Kulis 2012]:



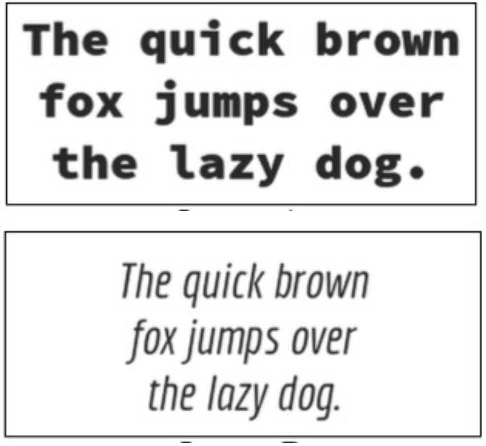
Fonts
[O'Donovan et al. 2014]

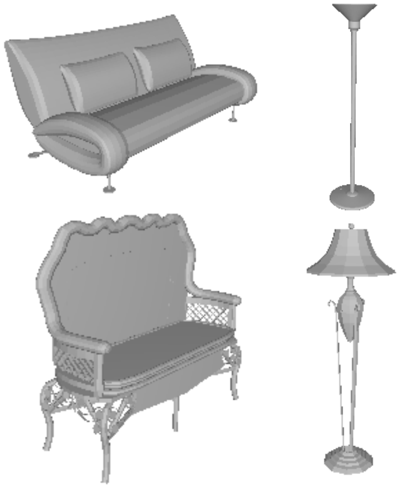
Illustration styles
[Garces et al. 2014]

Learning object-class specific embeddings

Assumptions of the previous approach

- Feature vectors have same dimensionality.
- Corresponding dimensions are comparable.





Learning object-class specific embeddings

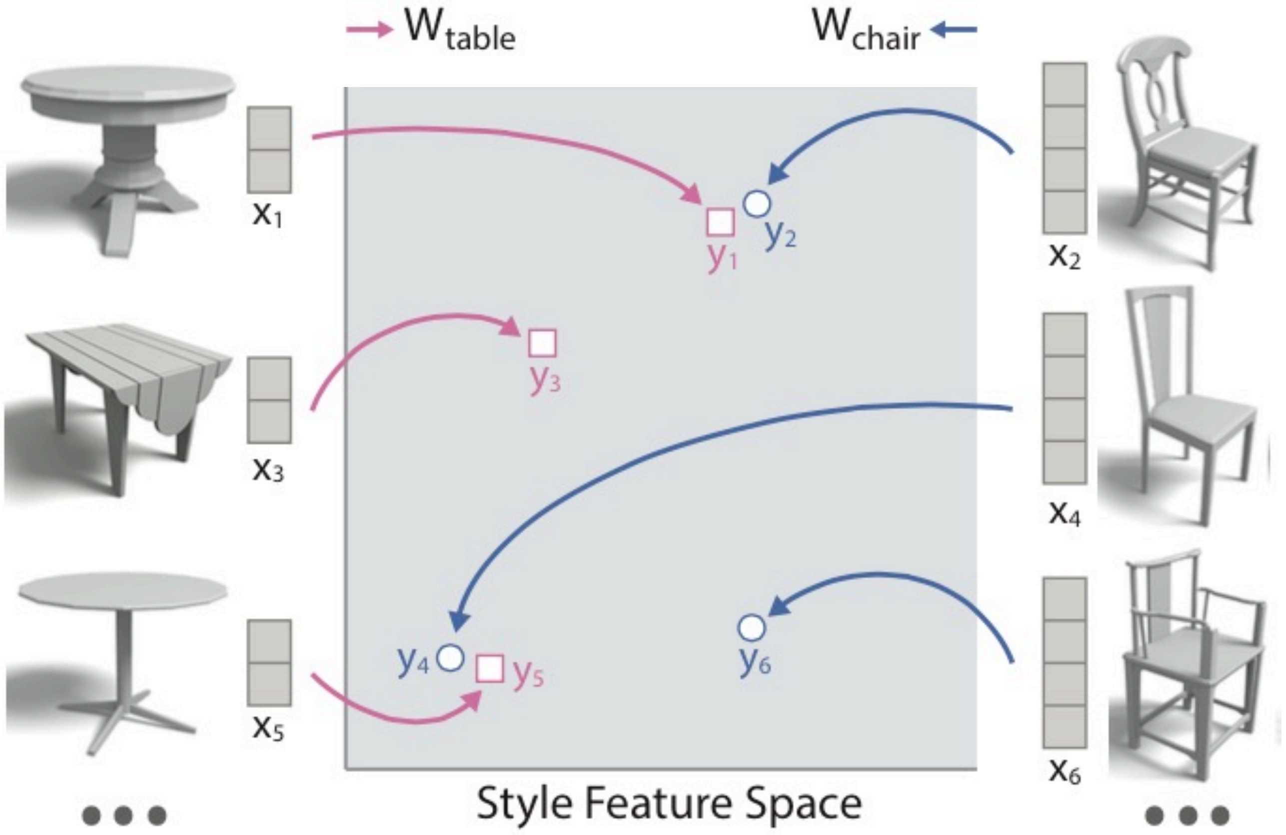
Our approach: **Asymmetric embedding**

$$d_{asymm}(x_i, x_j) = \left\| W_{c(i)} x_i - W_{c(j)} x_j \right\|_2$$

$c(i)$ is the object class of x_i

$c(j)$ is the object class of x_j

Learning object-class specific embeddings



Learning object-class specific embeddings

Learning procedure [O'Donovan et al. 2014]

- Using a logistic function to model rater's preferences
- Learning by maximizing the likelihood of the training triplets with regularization

Outline

- Key ideas
- Results of triplet prediction
- Applications

Results of triplet prediction

Test set: triplets that human agree upon

- 264 triplets from dining room
- 229 triplets from living room

Results of triplet prediction

Method	Dining room	Living room
Chance	50%	50%
No part-aware, Symmetric	63%	55%
Part-aware, Symmetric	63%	65%
No part-aware, Asymmetric	68%	65%
Part-aware, Asymmetric (Ours)	73%	72%
People	93%	99%

Results of triplet prediction

Method	Dining room	Living room
Chance	50%	50%
No part-aware, Symmetric	63%	55%
Part-aware, Symmetric	63%	65%
No part-aware, Asymmetric	68%	65%
Part-aware, Asymmetric (Ours)	73%	72%
People	93%	99%

Results of triplet prediction

Method	Dining room	Living room
Chance	50%	50%
No part-aware, Symmetric	63%	55%
Part-aware, Symmetric	63%	65%
No part-aware, Asymmetric	68%	65%
Part-aware, Asymmetric (Ours)	73%	72%
People	93%	99%

Outline

- Key ideas
- Results of triplet prediction
- Applications

Applications

- Style-aware shape retrieval
- Style-aware furniture suggestion
- Style-aware scene building

Applications

- Style-aware shape retrieval
- Style-aware furniture suggestion
- Style-aware scene building

Style-aware shape retrieval

Query model

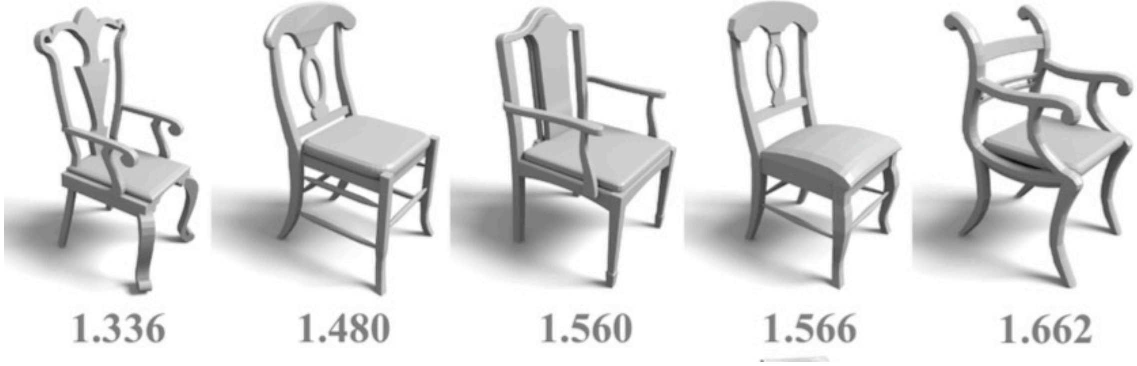
Dining chair

?

Style-aware shape retrieval

Query model

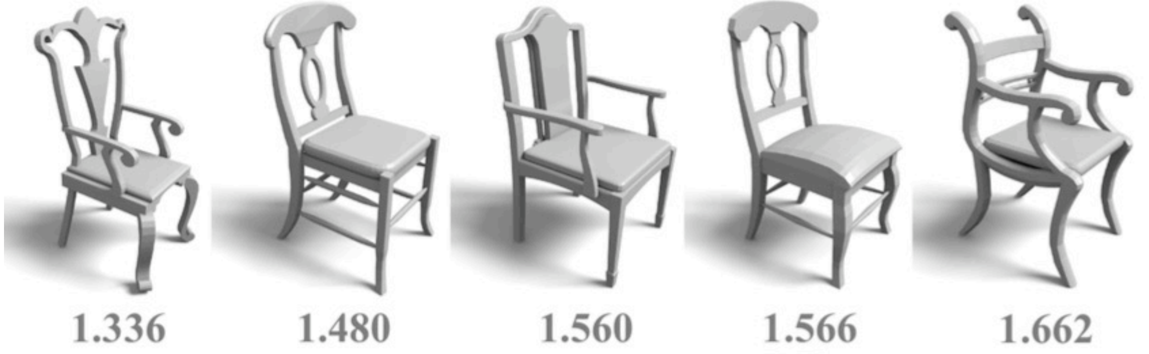
Dining chair



Style-aware shape retrieval

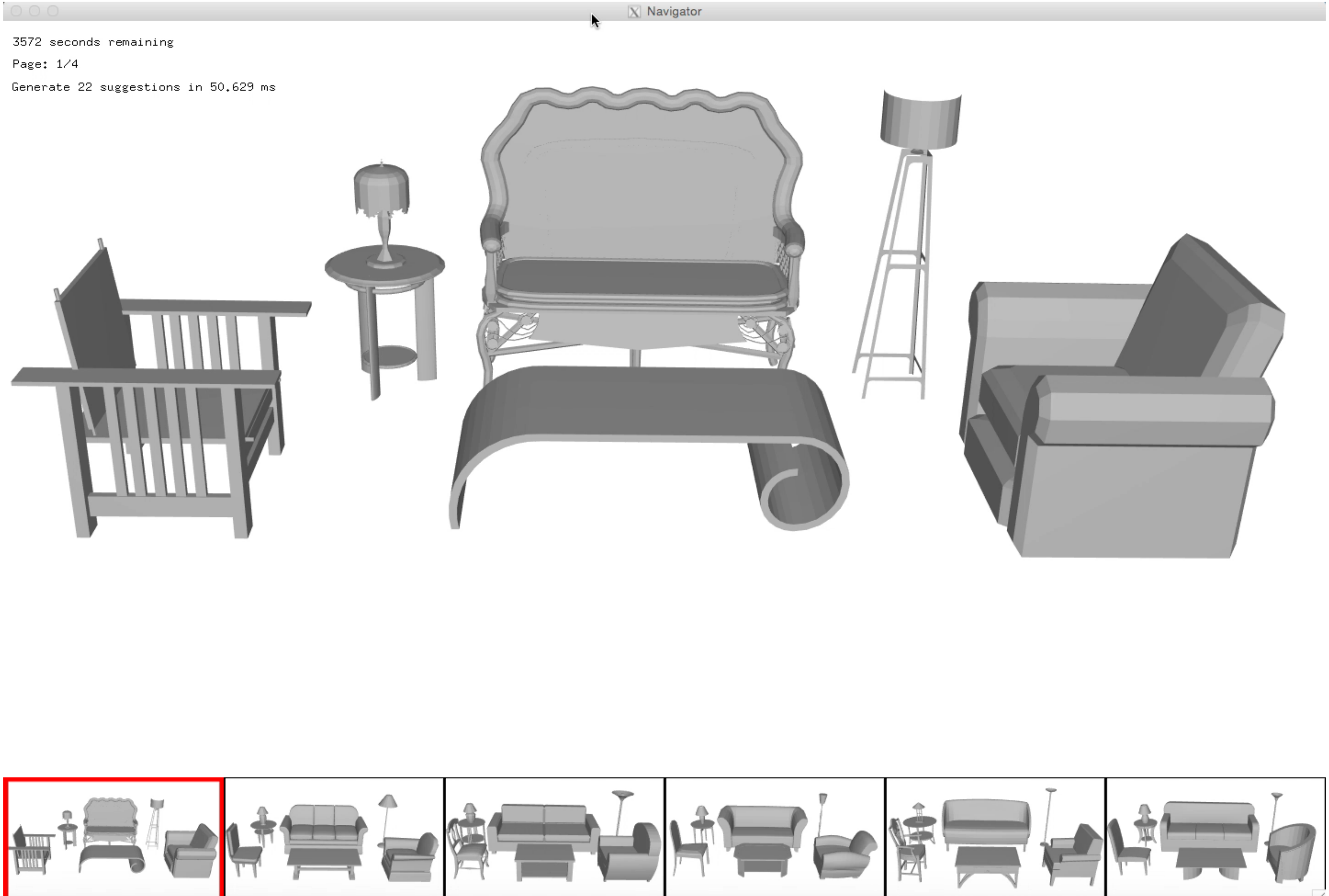
Query model

Dining chair



(Most incompatible chairs)

Style-aware scene modeling

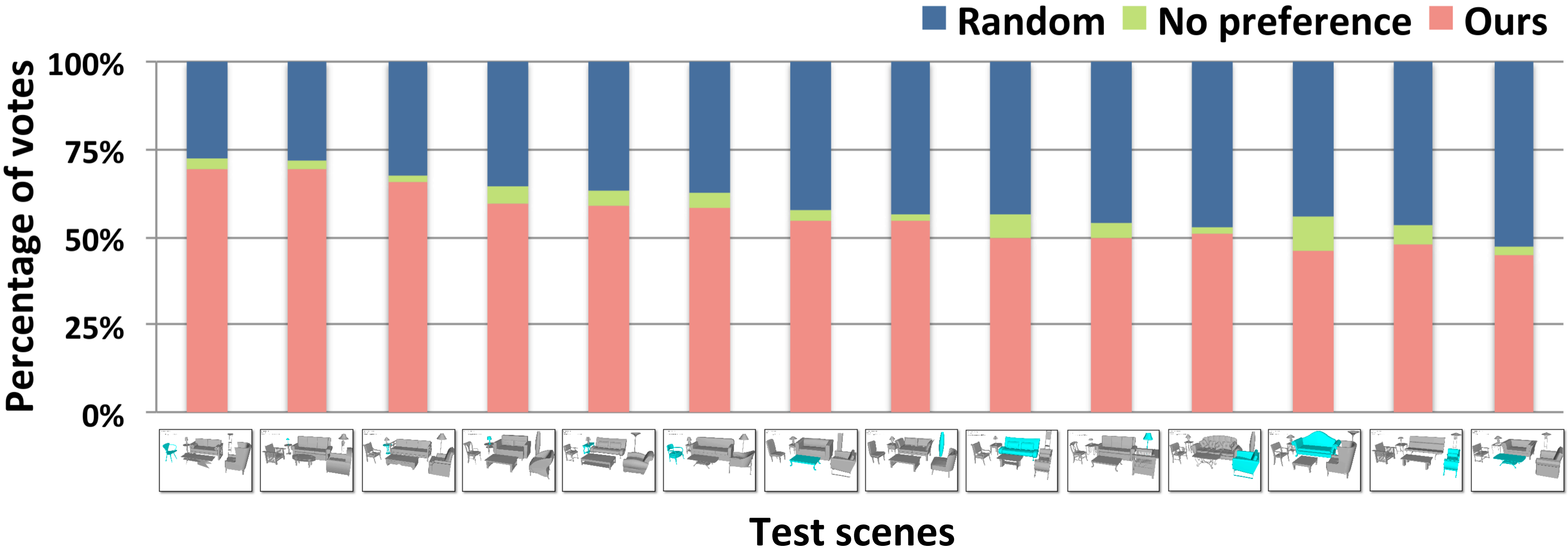

Navigator
3572 seconds remaining
Page: 1/4
Generate 22 suggestions in 50.629 ms

Style-aware scene building

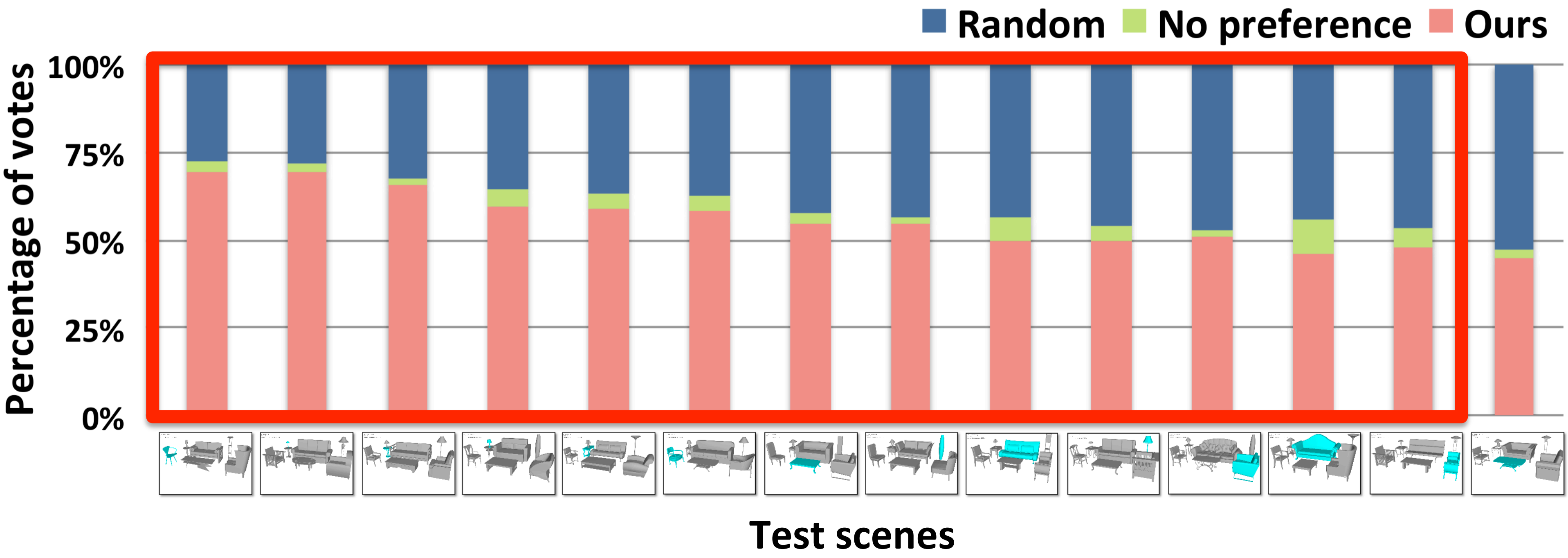
User study

- 12 participants, each works on 14 tasks.
- Half of the tasks are assisted by our metric, and the other half are not.
- Results from both conditions are compared on Amazon Mechanical Turk

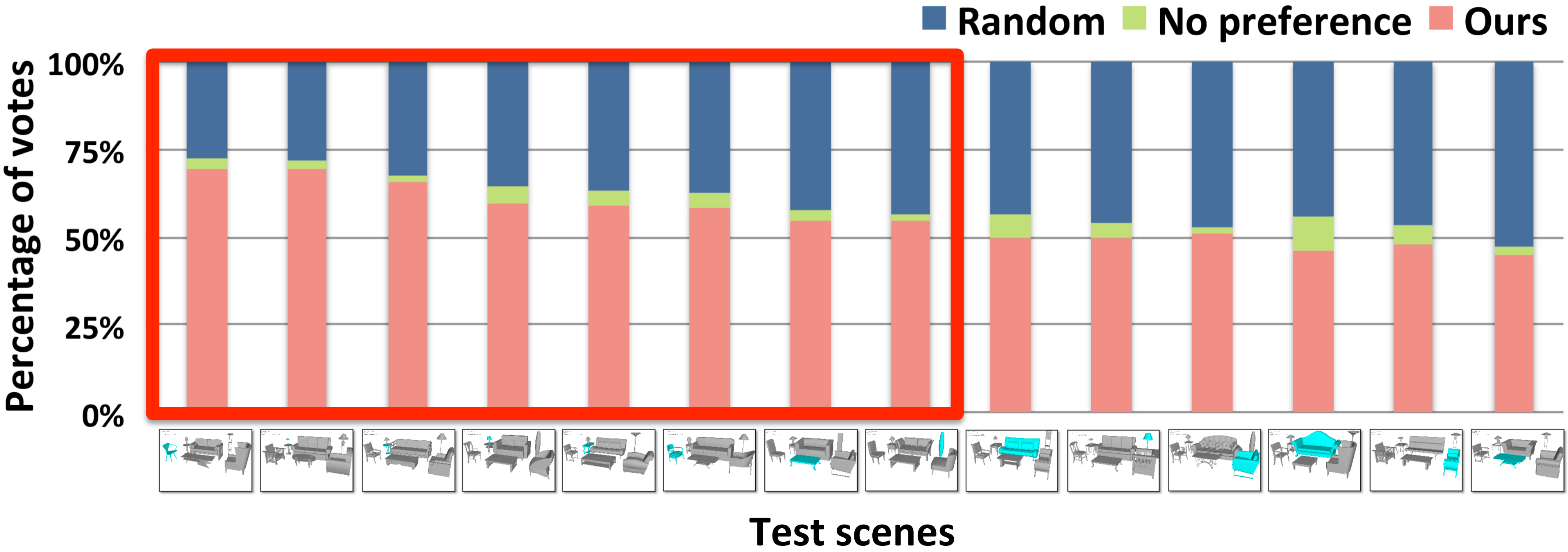
Style-aware scene building



Style-aware scene building



Style-aware scene building



Take-away messages

It is possible to learn a compatibility metric for furniture of different classes.

- Part-aware geometric features
- Asymmetric embedding of individual object classes

The learned compatibility metric is effective in style-aware scene modeling.

- Shape retrieval
- Interactive scene building

Take-away messages

It is possible to learn a compatibility metric for furniture of different classes.

- Part-aware geometric features
- Asymmetric embedding of individual object classes

The learned compatibility metric is effective in style-aware scene modeling.

- Shape retrieval
- Interactive scene building

Take-away messages

It is possible to learn a compatibility metric for furniture of different classes.

- Part-aware geometric features
- Asymmetric embedding of individual object classes

The learned compatibility metric is effective in style-aware scene modeling.

- Shape retrieval
- Interactive scene building

Limitations and future work

- Modeling fine-grained style variations

Duncan Phyfe style with eagle motif

(Courtesy: Carswell Rush Berlin)

Sheraton style with lyre motif

Limitations and future work

- Modeling fine-grained style variations
- Investigating how other properties determine style

Limitations and future work

- Modeling fine-grained style variations
- Investigating how other properties determine style
- Investigating style compatibility in other domains

OR

Acknowledgements

Data and code

- Trimble and Digimation
- Vladimir Kim and Evangelos Kalogerakis

Discussion

- Adam Finkelstein and Peter O’Donovan

Funding

- Adobe, Google, Intel, NSF

Project webpage

http://gfx.cs.princeton.edu/pubs/Liu_2015_SCF