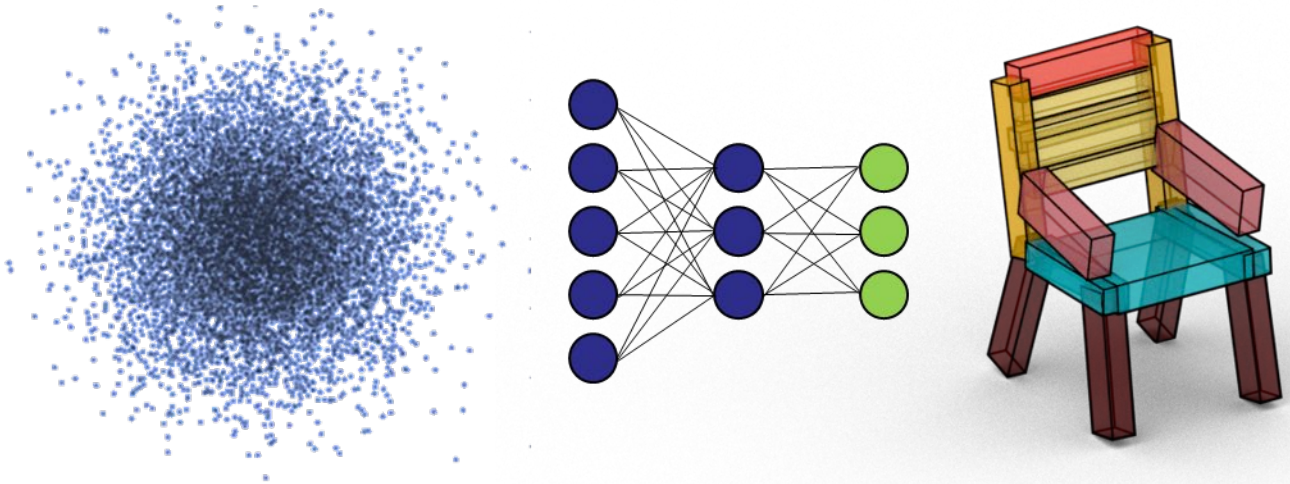
CS348n: Neural Representations and Generative Models for 3D Geometry



Leonidas Guibas
Computer Science Department
Stanford University

Class Logistics

- Project presentation will be during the next class, Wed, March 9. The will be on Zoom only.

- Projects are due Friday, March 11, 11:59 am.

Last Time:
Neural Scene Generation

Scene Generation Raises Many Issues

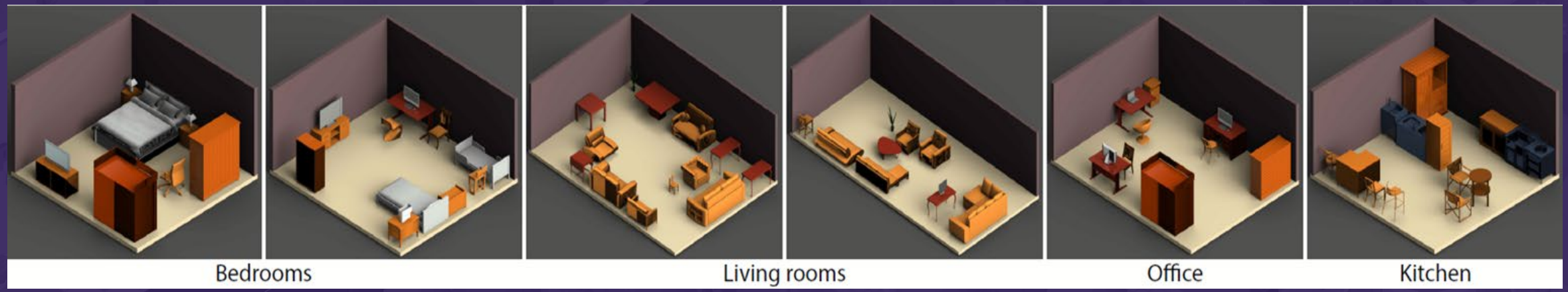
- Object selection
- Object placement
- Scene hierarchies and object groupings
- Scene affordances

thrive SIGGRAPH2019
LOS ANGELES • 28 JULY – 1 AUGUST

GRAINS: GENERATIVE RECURSIVE AUTOENCODERS FOR INDOOR SCENES

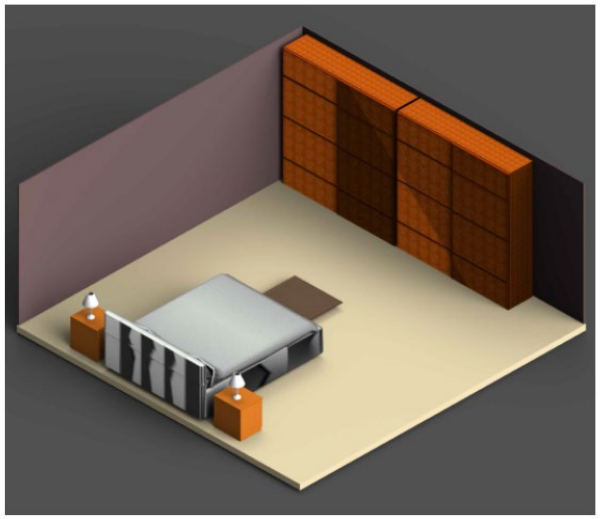
Manyi Li [1,2], **Akshay Gadi Patil** [2], Kai Xu [3,4], Siddhartha Chaudhuri [5,6], Owais Khan [6], Ariel Shamir [7], Changhe Tu [1], Baoquan Chen [8], Daniel Cohen-Or [9], Hao (Richard) Zhang [2]

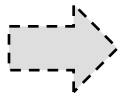
[1] Shandong University [2] Simon Fraser University [3] National University of Defense Technology [4] AICFVE Beijing Film Academy
[5] Adobe Research [6] IIT Bombay [7] The Interdisciplinary Center [8] Peking University [9] Tel-Aviv University

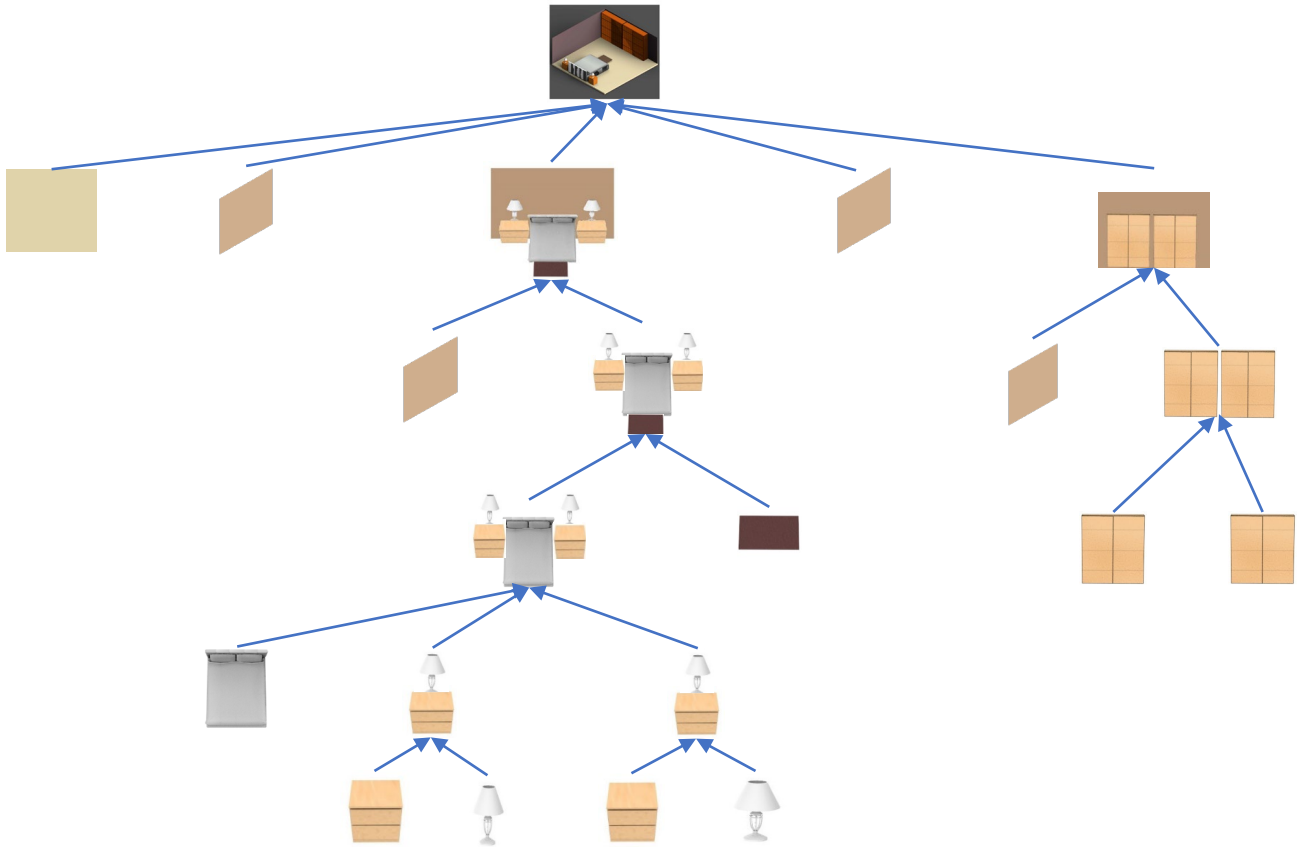


© 2019 SIGGRAPH. ALL RIGHTS RESERVED.

Scene Representation







Hierarchical Scene Graph

Scene Representation

- Leaf nodes: objects
- Internal nodes: groups

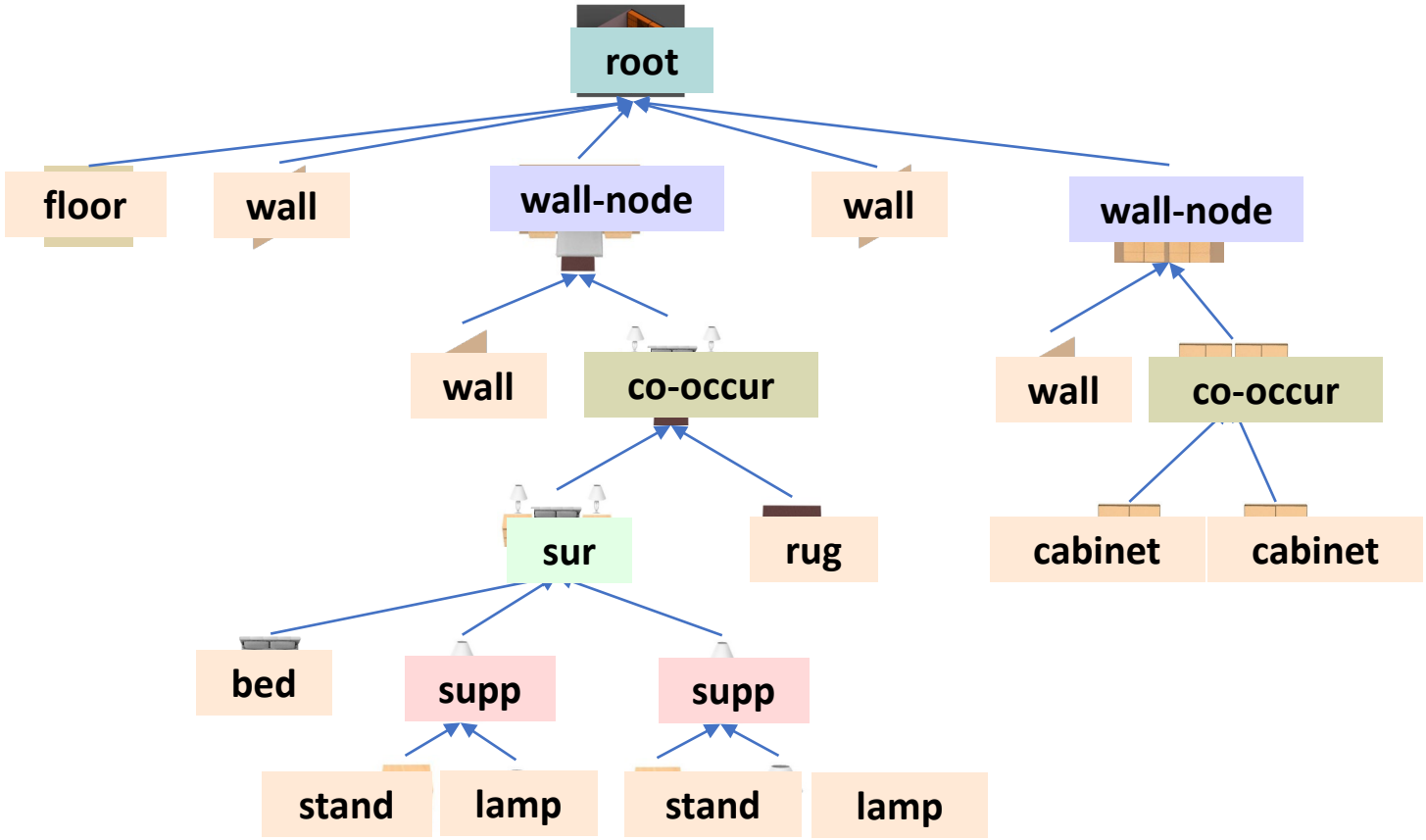
support node

surround node

co-occur node

wall node

root node



Step3: Build internal nodes and compute their OBBs

NETWORK

- Network: ***Recursive*** Variational Autoencoder

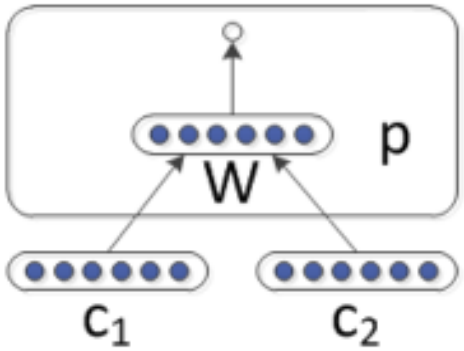
  Recursive Neural Networks (RvNN):

  -- Repeatedly merge two or more nodes into one

  -- Each node has an n-D feature vector, computed recursively

$$p = MLP_{\theta_i}(c_1, c_2)$$

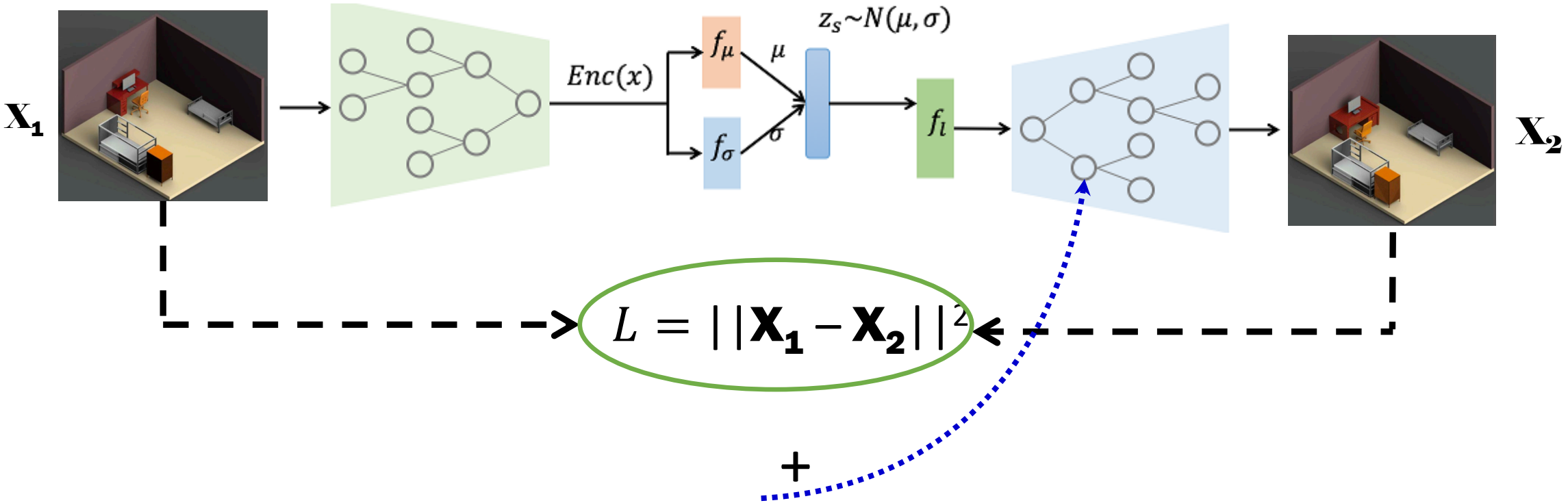
  -- End up with a fixed dimensional code for the root node, encoding entire tree



We use different MLPs for different types of relational encoders/decoders

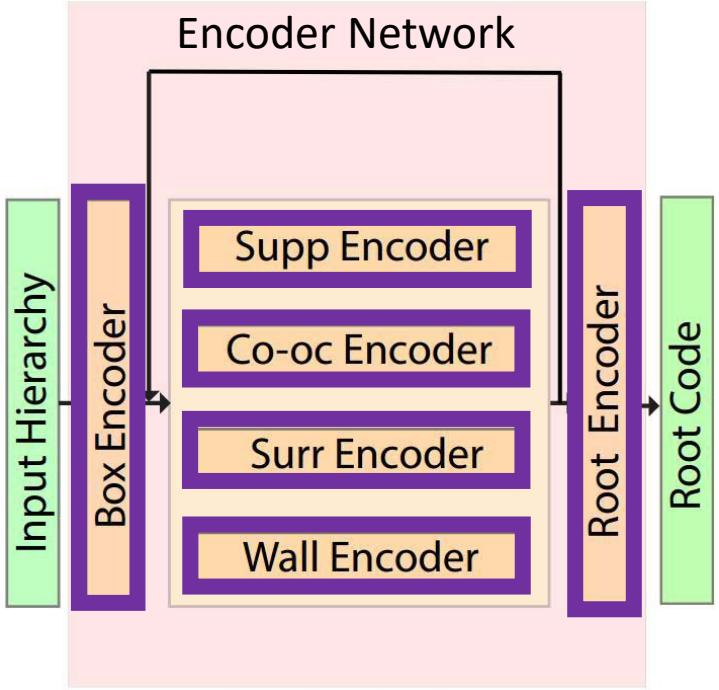
GRAINS Network

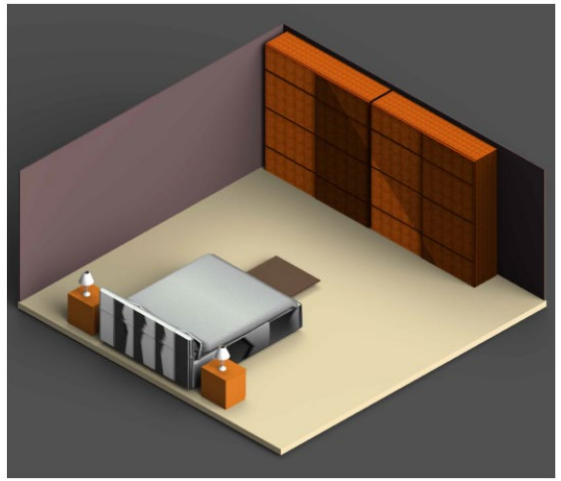
- Network: ***Recursive*** Variational Autoencoder

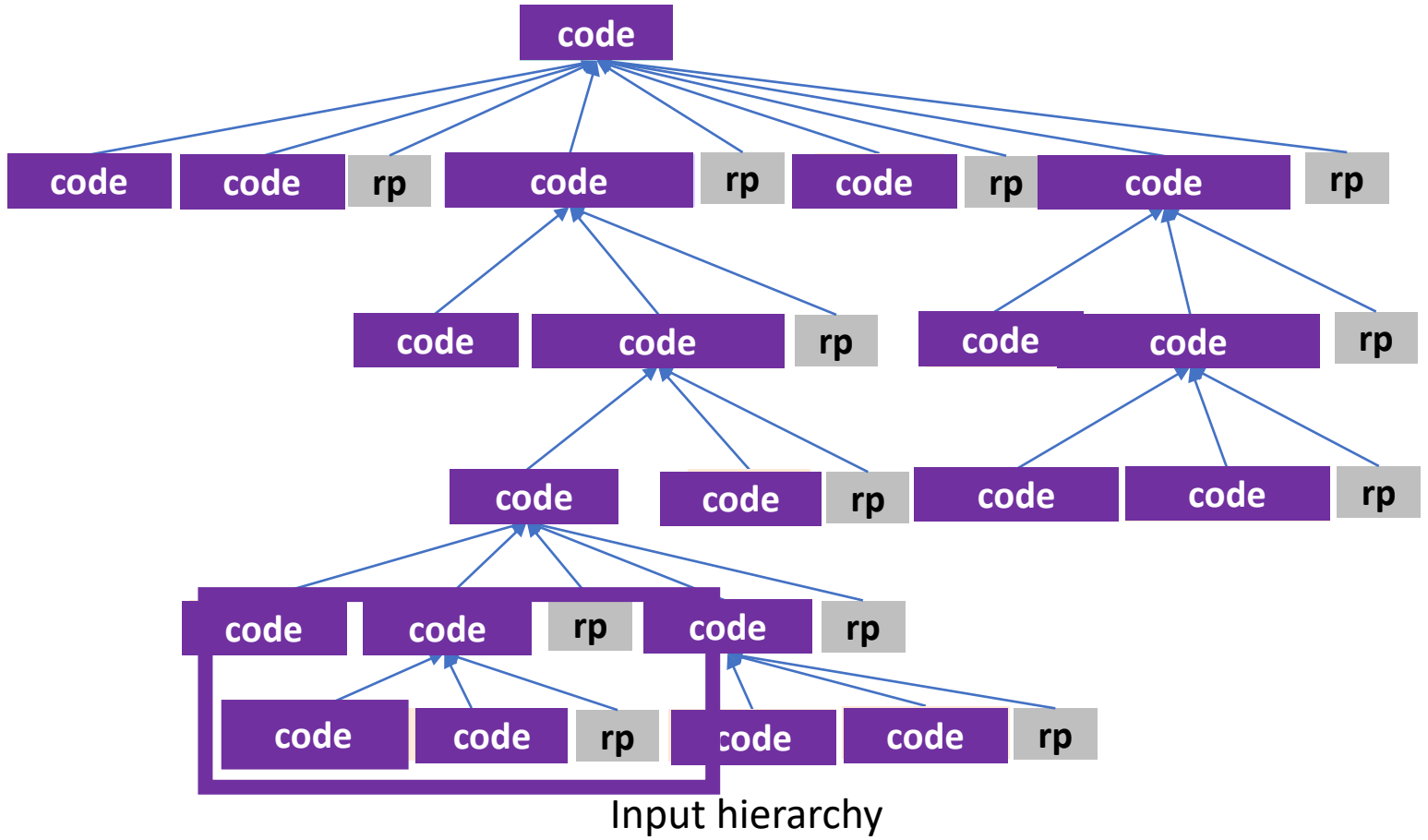


A node classifier is trained in parallel to decide which of the 5 decoders (support, surround etc.) to be invoked.

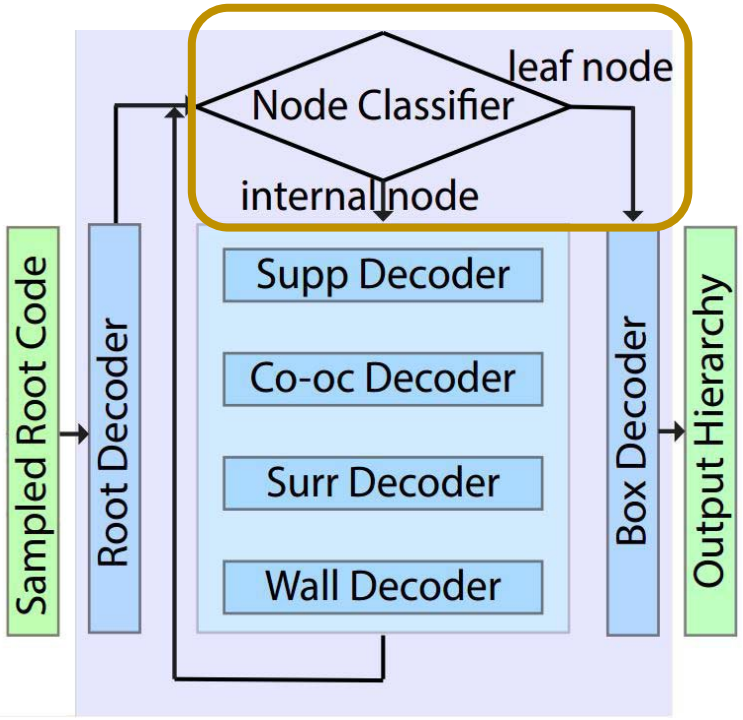
Encoding Process



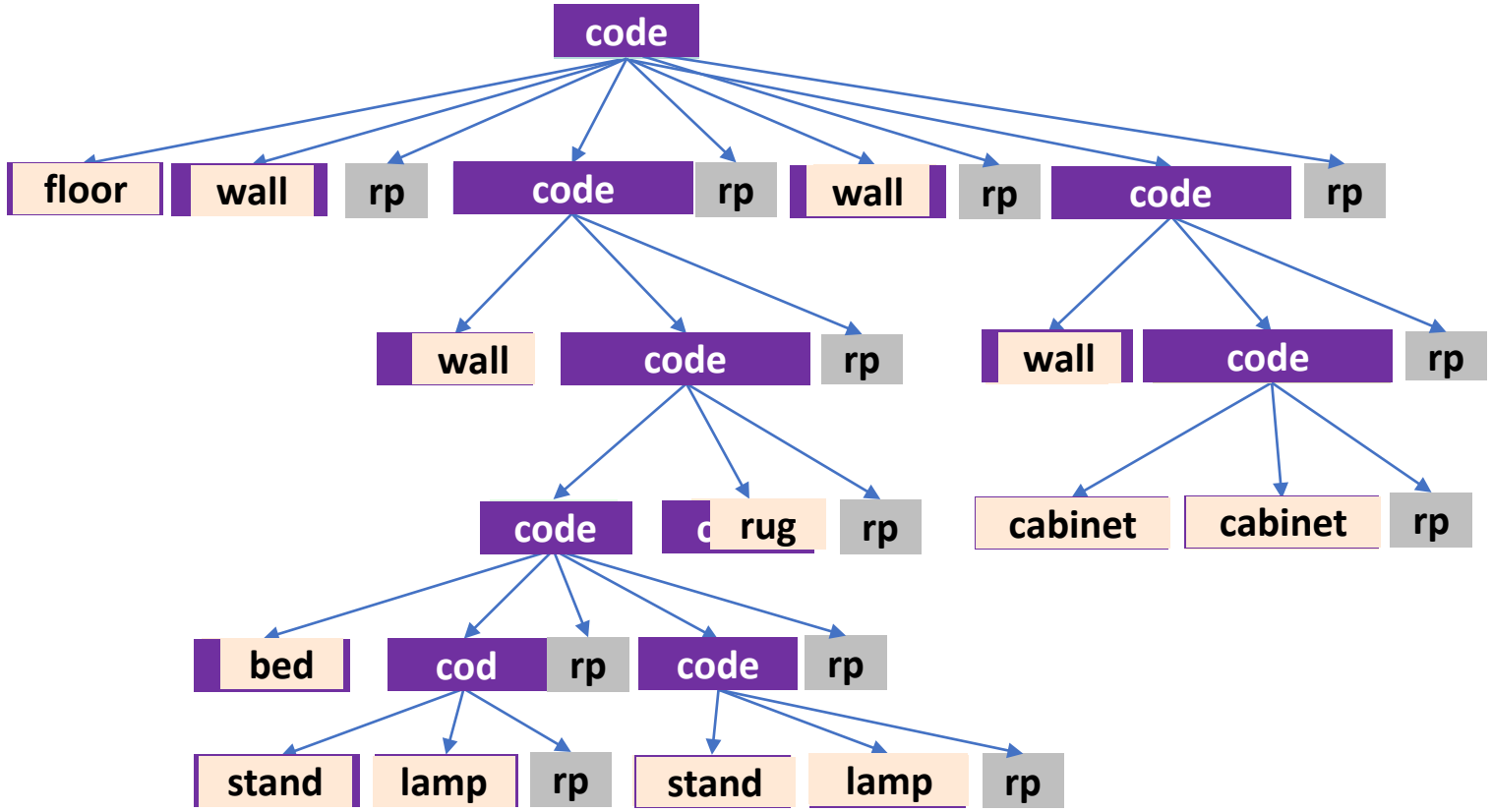




Decoding Process



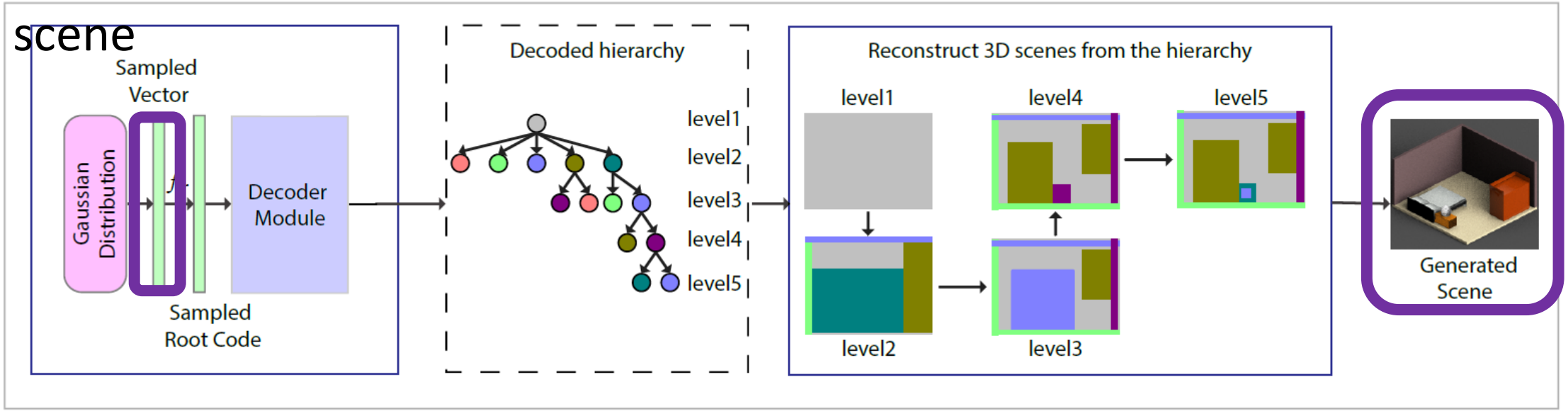
Decoder Network



Output hierarchy

Generation Pipeline

- The network learns to map a random vector to a plausible indoor scene



Generation pipeline

Scene Representation Matters

- Indoor scenes are complex and diverse

Appropriate scene representation is the key to learning

Our key points:

(1) *hierarchical* **recursive** structure

(2) relative position format

GRAINS Summary

- First Deep Hierarchical Generative Model of Indoor Scenes
- Fast and efficient indoor scene generation
- Scene representation matters in learning structural data
- Data Augmentation - Generate a huge quantity of scenes for use in data augmentation tasks

Scene Generation for ML Training

Meta-Sim: Learning to Generate Synthetic Datasets

Amlan Kar[1,2,3] Aayush Prakash[1] Ming-Yu Liu[1] Eric Cameracci[1] Justin Yuan[1]
Matt Rusiniak[1] David Acuna[1,2,3] Antonio Torralba[4] Sanja Fidler[1,2,3*]

[1]NVIDIA [2]University of Toronto [3]Vector Institute [4] MIT

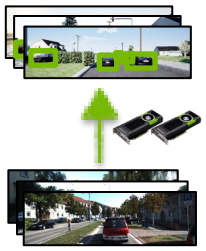
ICCV 2019

Meta-Sim2: Unsupervised Learning of Scene Structure for Synthetic Data Generation

ECCV 2020

Jeevan Devaranjan[*1,3], Amlan Kar[*1,2,4], and Sanja Fidler[1,2,4]

[1]NVIDIA [2]University of Toronto [3]University of Waterloo [4]Vector Institute



How to Simulate?

Scene Grammar (Example)

Road — Lanes

Lanes — Lane Lanes | ε

Lane — Cars Sidewalk | ε

Cars — *car* Cars | ε

Sidewalk — People | ε

People — *person* People | ε

Creating the scene graph

Road - Lanes

Lanes - Lane, Lanes

Lanes - ε

Lane - Cars, Sidewalk

Sidewalk - People

People - person, People

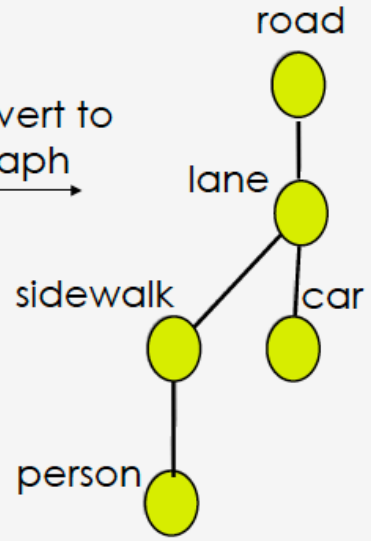
People - ε

Cars - car, Cars

Cars - ε

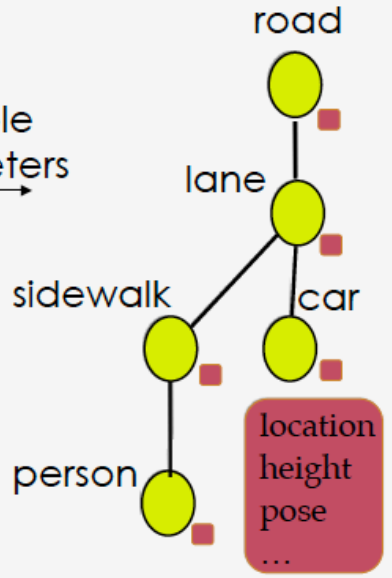
Sampled Rules

Convert to graph

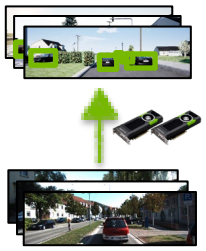


Corresponding Scene Structure

Sample Parameters

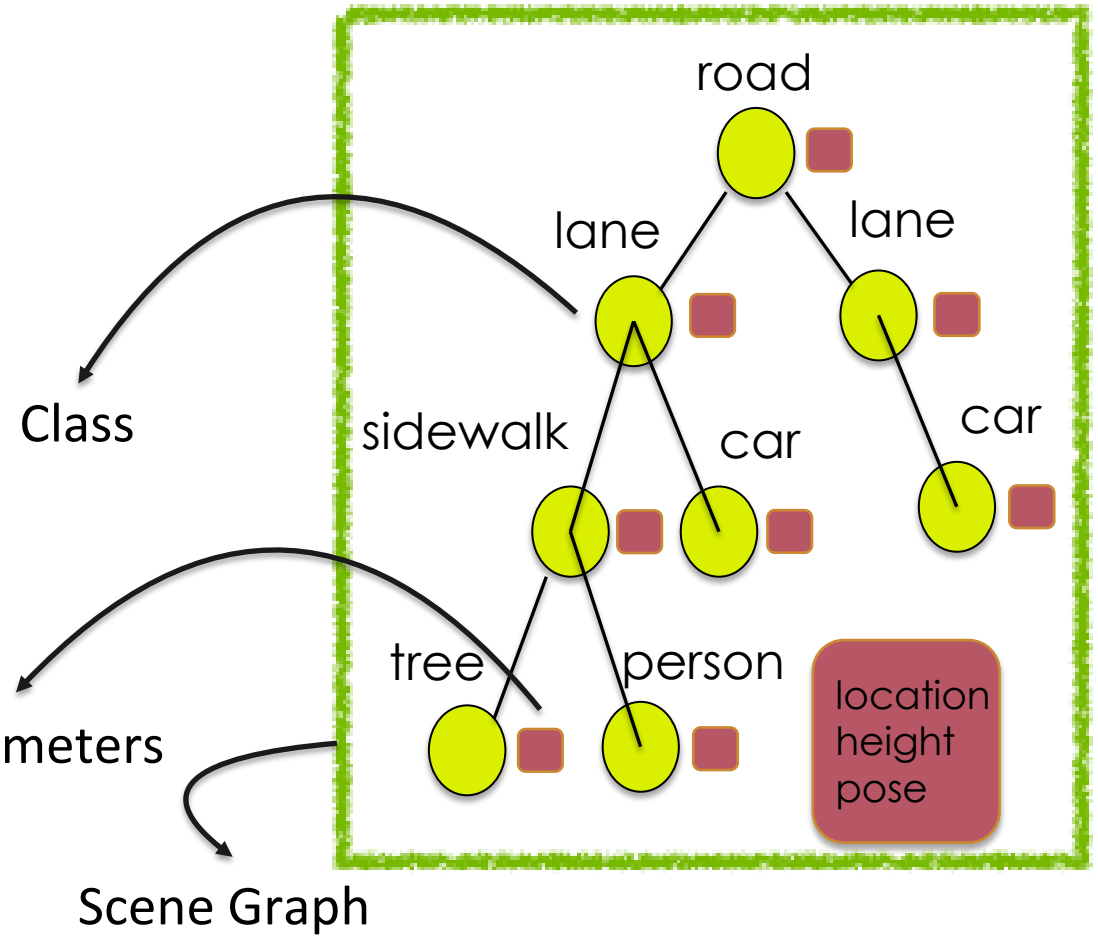


Scene Graph



Simulating with a Probabilistic Grammar

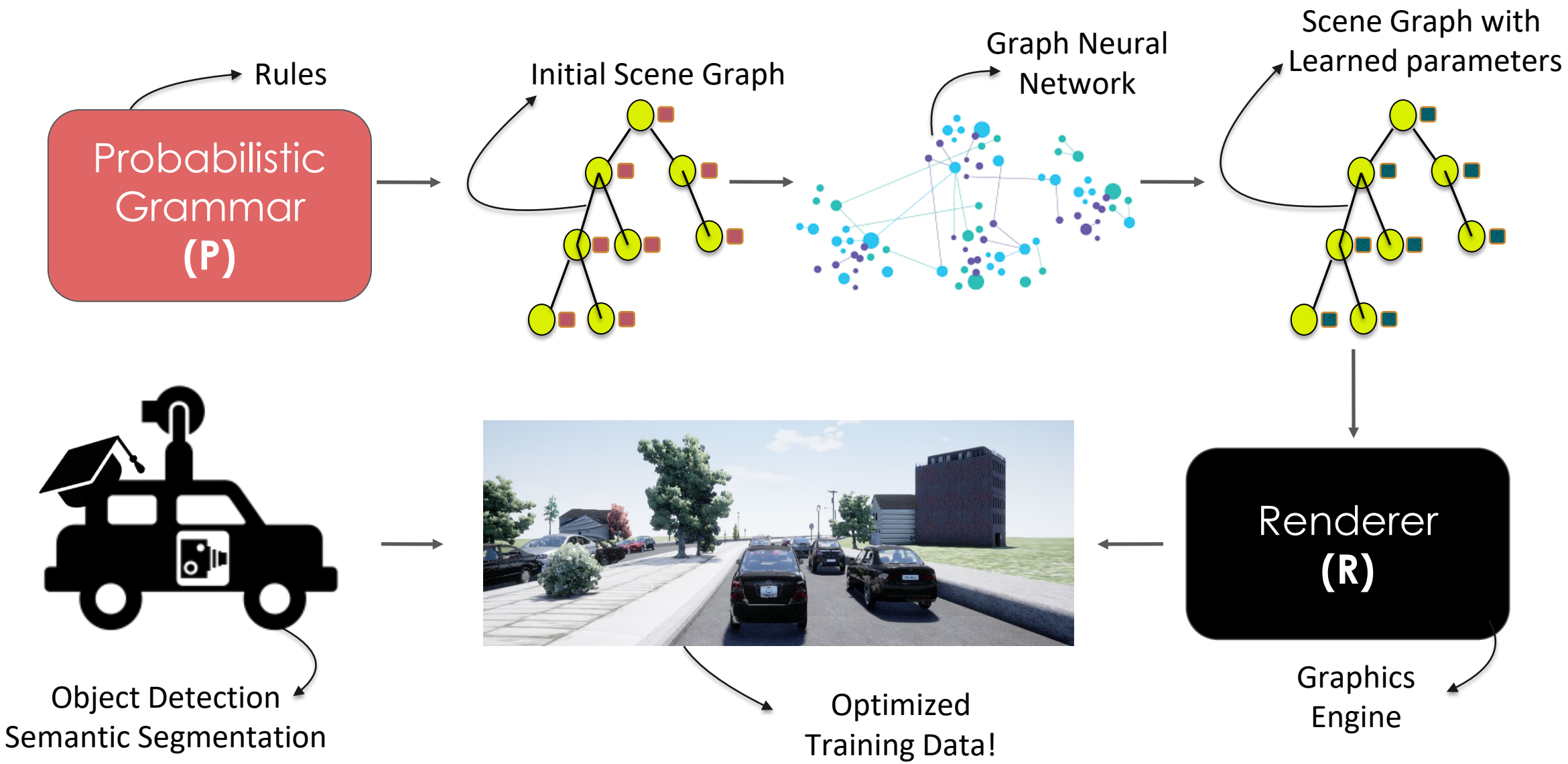
Probabilistic Grammar

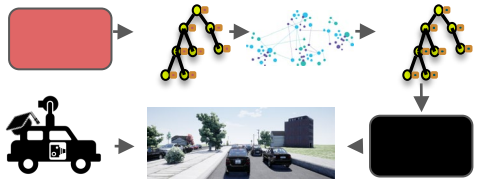


Meta-Sim

Problem Statement

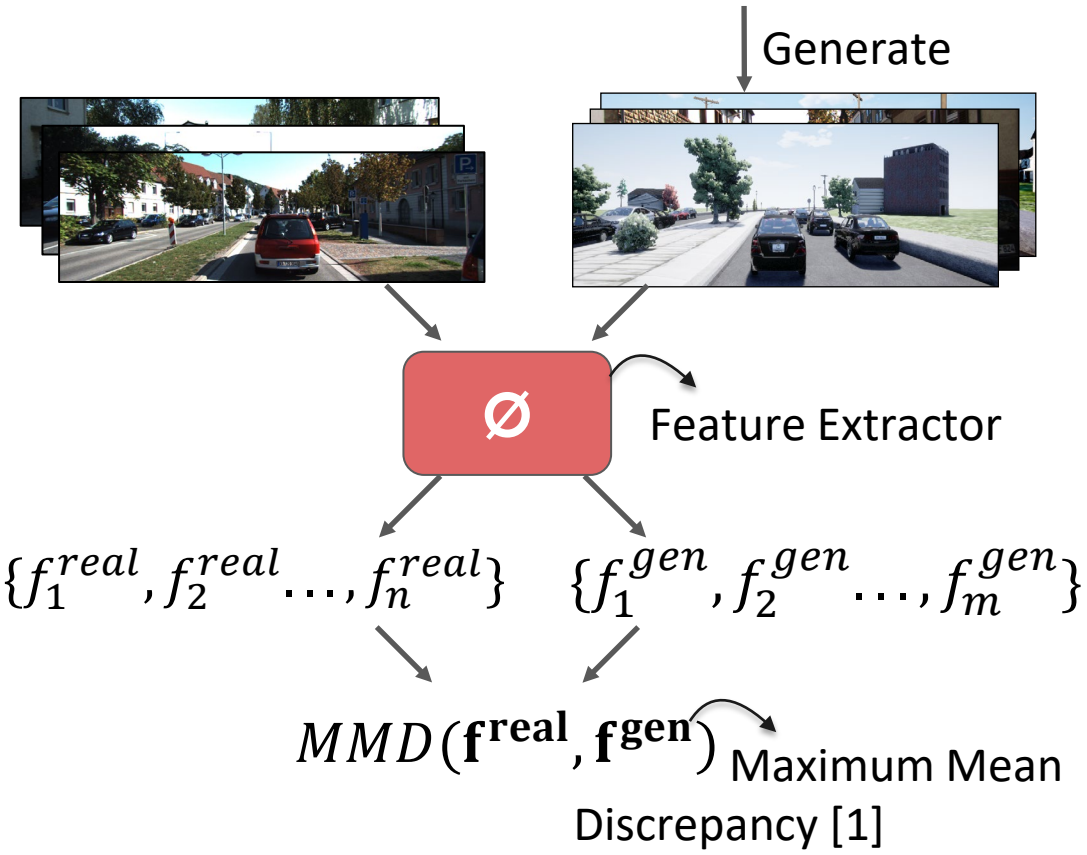
Meta-Sim


Rules
Probabilistic Grammar (P)
Initial Scene Graph
Graph Neural Network
Scene Graph with Learned parameters
Renderer (R)
Graphics Engine
Optimized Training Data!
Object Detection
Semantic Segmentation

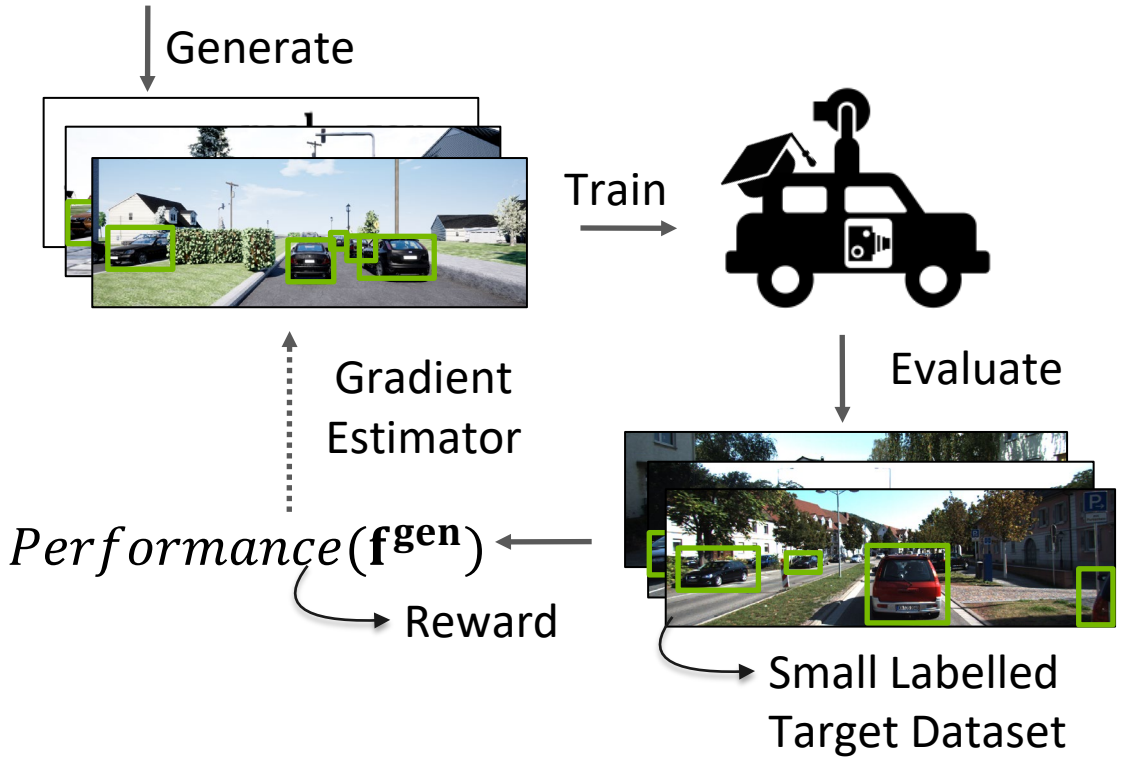


Meta-Sim

Distribution Matching



Task Loss



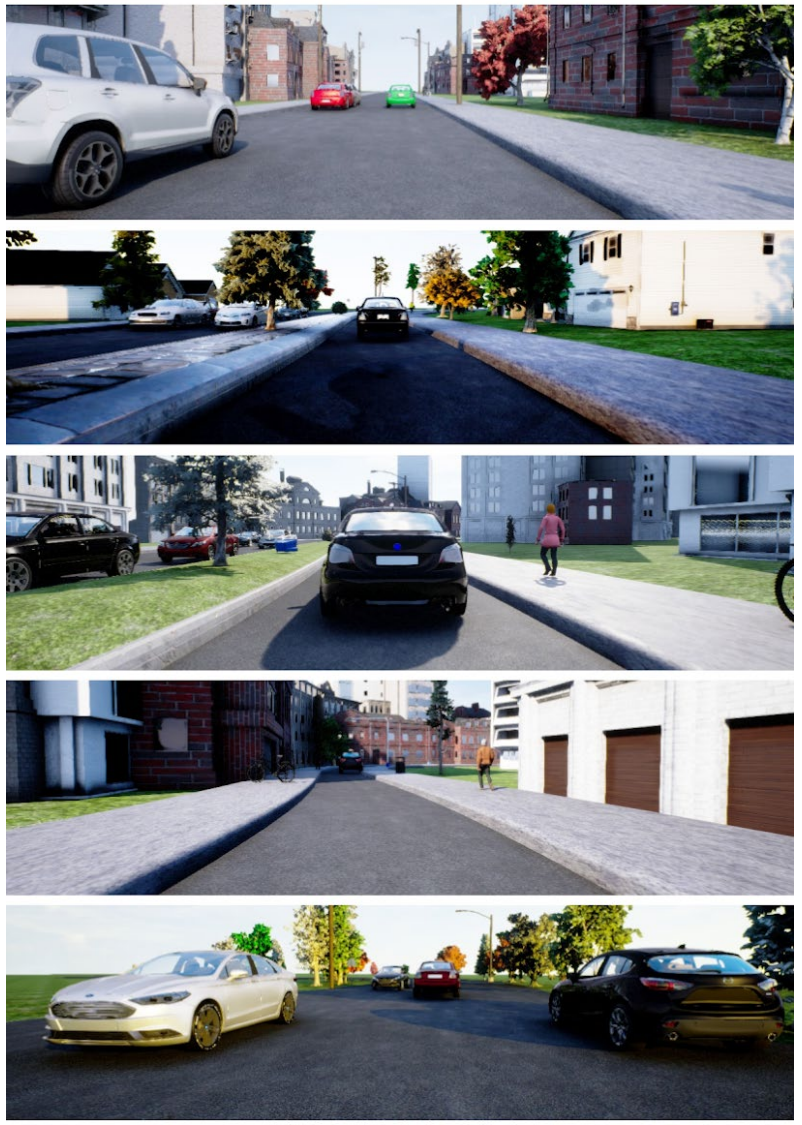
[1] A Kernel Two-Sample Test, Gretton et al. JMLR '13

Results

3D Driving Scenes

3D Driving Scenes

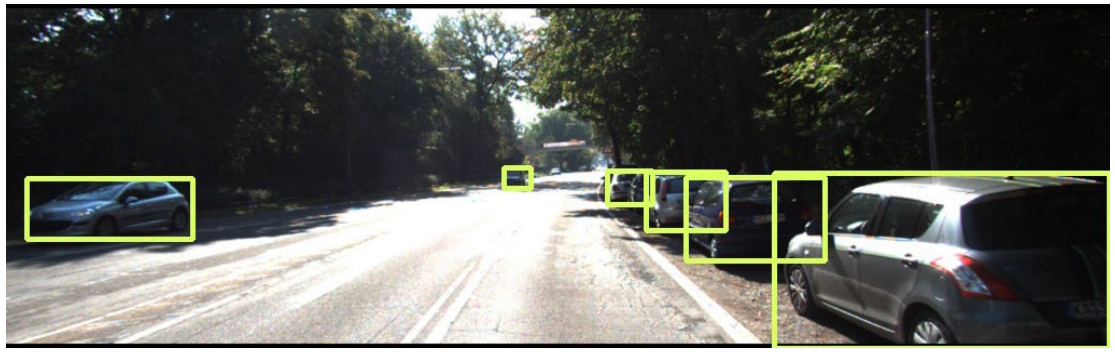
Probabilistic Grammar

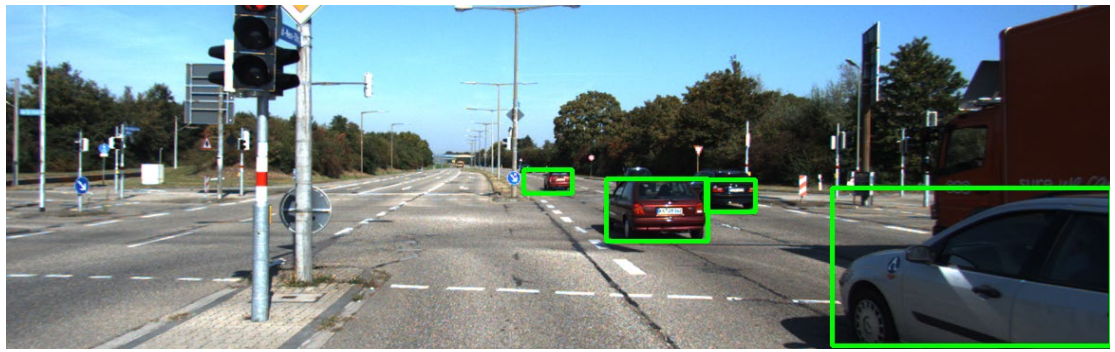


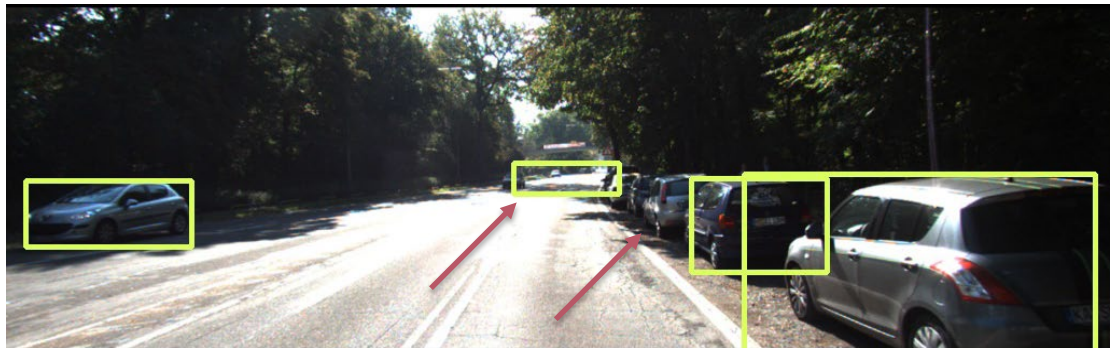
Meta-Sim

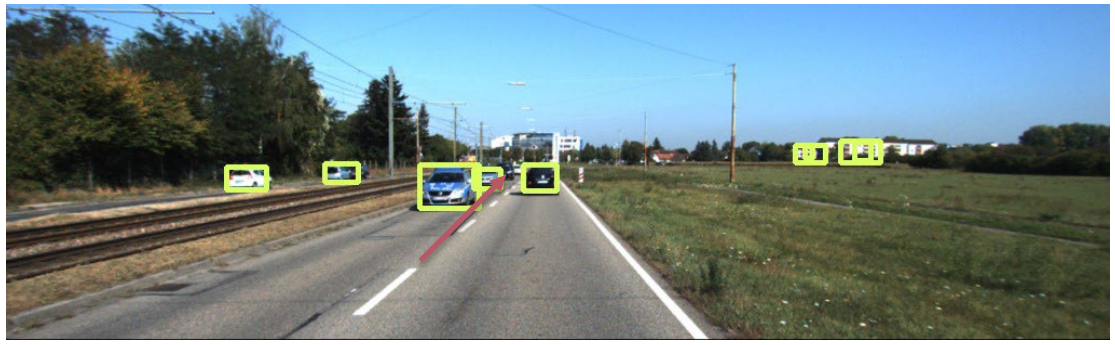
Random Real Images (KITTI)

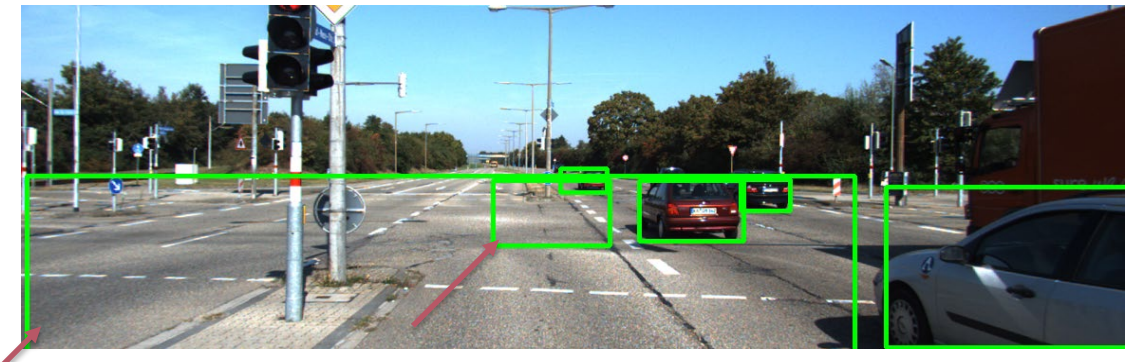
KITTI











Meta-Sim

Probabilistic Grammar

Meta-Sim2

Learn the Derivation

Meta-Sim2

Rules

Probabilistic Grammar **(P)**

Scene Graph with learned structure

Graph Neural Network

Scene Graph with learned structure and parameters

Renderer **(R)**

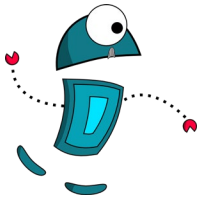
Graphics Engine

Learned Synthetic Scenarios

Target Images

Learning Signal

UNSUPERVISED



Meta-Sim2

Generative Process

Scene Grammar (Example)

Road — Lanes

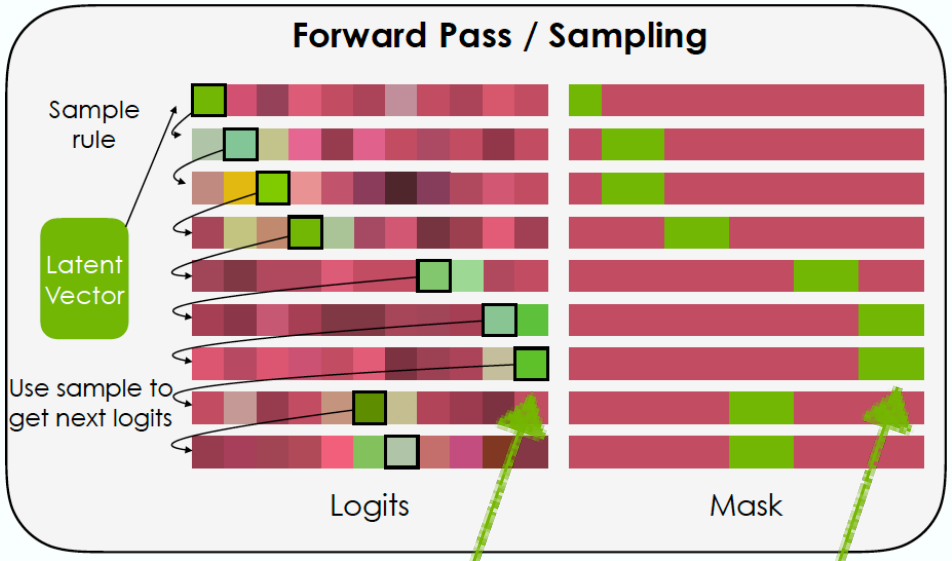
Lanes — Lane Lanes | ε

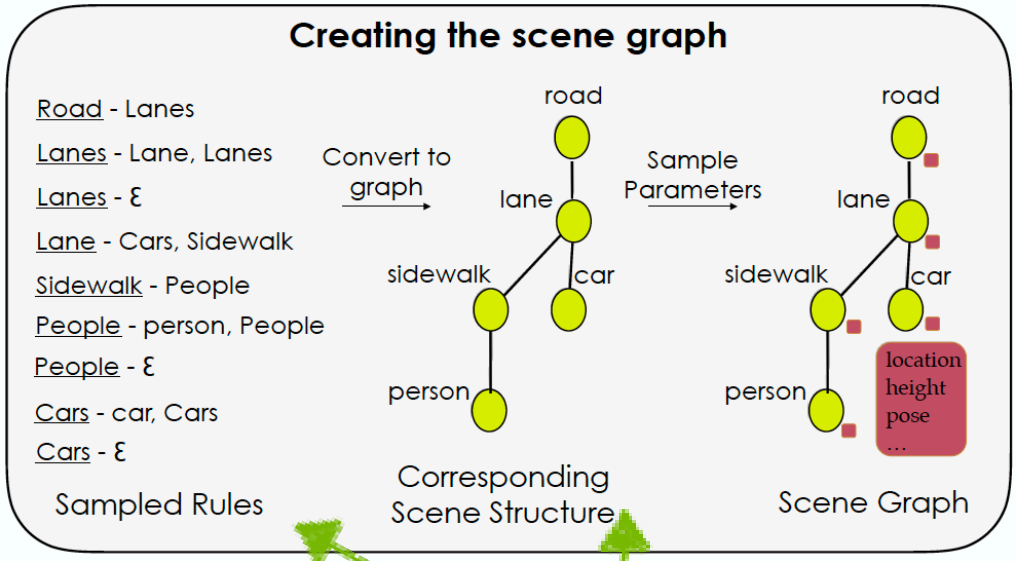
Lane — Cars Sidewalk | ε

Cars — *car* Cars | ε

Sidewalk — People | ε

People — *person* People | ε



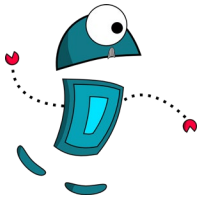


Sample rule from a flat list of all rules with a probability **masked** by valid rules

Every grammar string corresponds to a scene graph structure

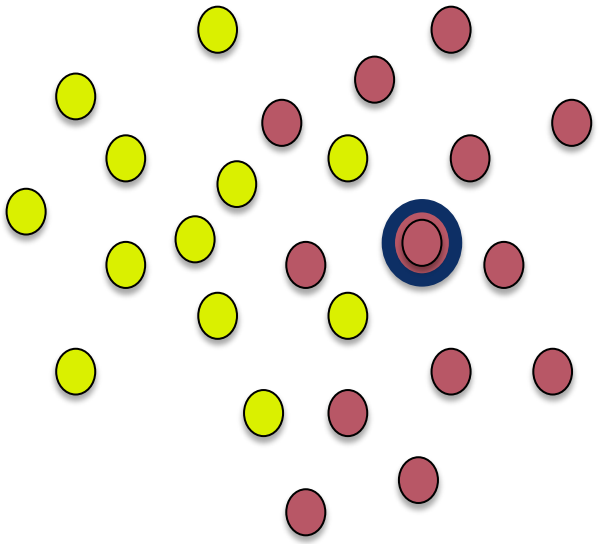
Generating logits using a **sampled** rule allows generating **context-dependent** scene graphs

Non-differentiable sequential decisions!
Use **Reinforcement Learning** to train



Meta-Sim2

Training: Computing a reward per scene



Scenes are point-clouds in a feature space

Real Scenes

Synthetic Scenes

Step 1: Compute likelihood of under all = p (is synthetic)

Step 2: Compute likelihood of under all = p (is real)

Step 3: Use log (likelihood-ratio) as reward **per scene** => in expectation it is the reverse-KL between the two distributions

Approximation: Compute likelihood non-parametrically using Kernel Density Estimates (RBF Kernel)

Approximation: Compute likelihood using a **batch** of generated real and synthetic samples

Results

Qualitative

Probabilistic Grammar

Meta-Sim2

Random Real Images (KITTI)

Language-Driven Shape and Scene Generation

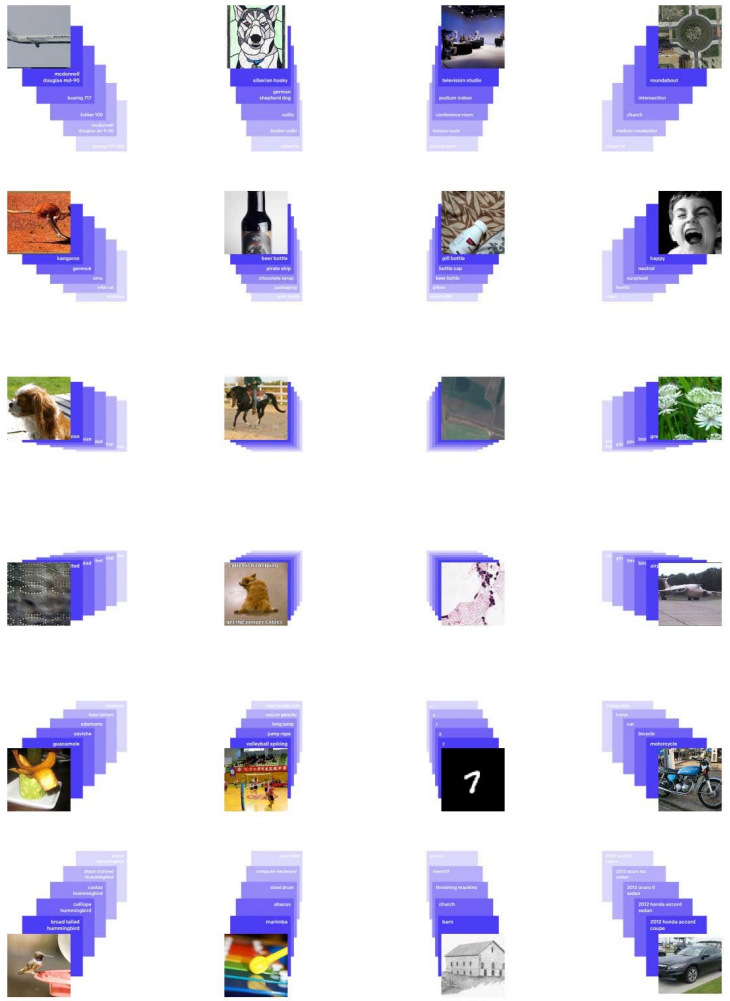
Language

- Makes 3D content creation and editing / modification universally accessible
- Raises many difficult issues, as language is often ambiguous and underspecific
- Relations between language elements (adjectives, nouns) and geometry elements and attributes is highly non-trivial

CLIP: Connecting Text and Images

- CLIP learns visual concepts from natural language supervision
  - typical vision datasets are labor intensive and costly to create while teaching only a narrow set of visual concepts;
  - standard vision models are good at one task and one task only, and require significant effort to adapt to a new task

- CLIP is a “foundation model”:
  - task and class agnostic visual understanding
  - general purpose, with good zero-shot performance (like GPT-2, GPT-3)



CLIP: Contrastive Language-Image Pre-training)

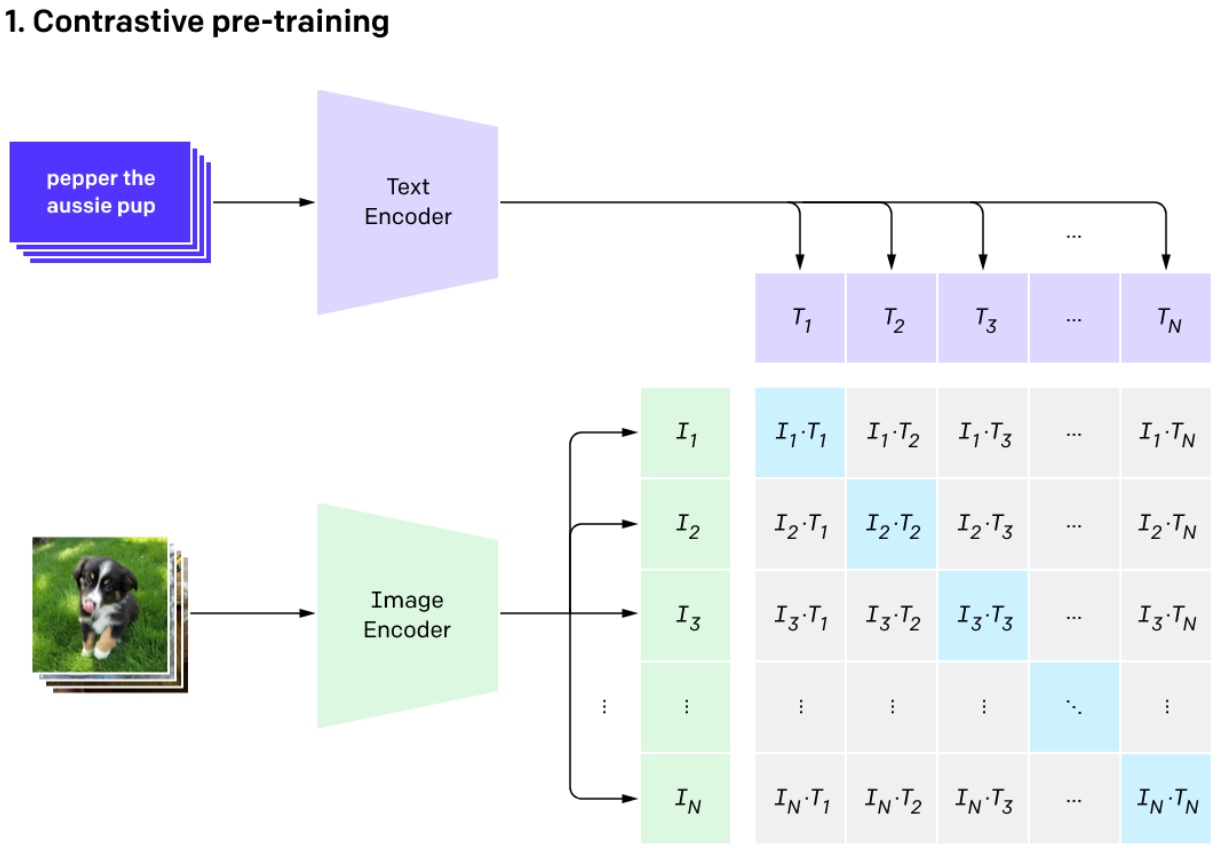
CLIP General Purpose Training

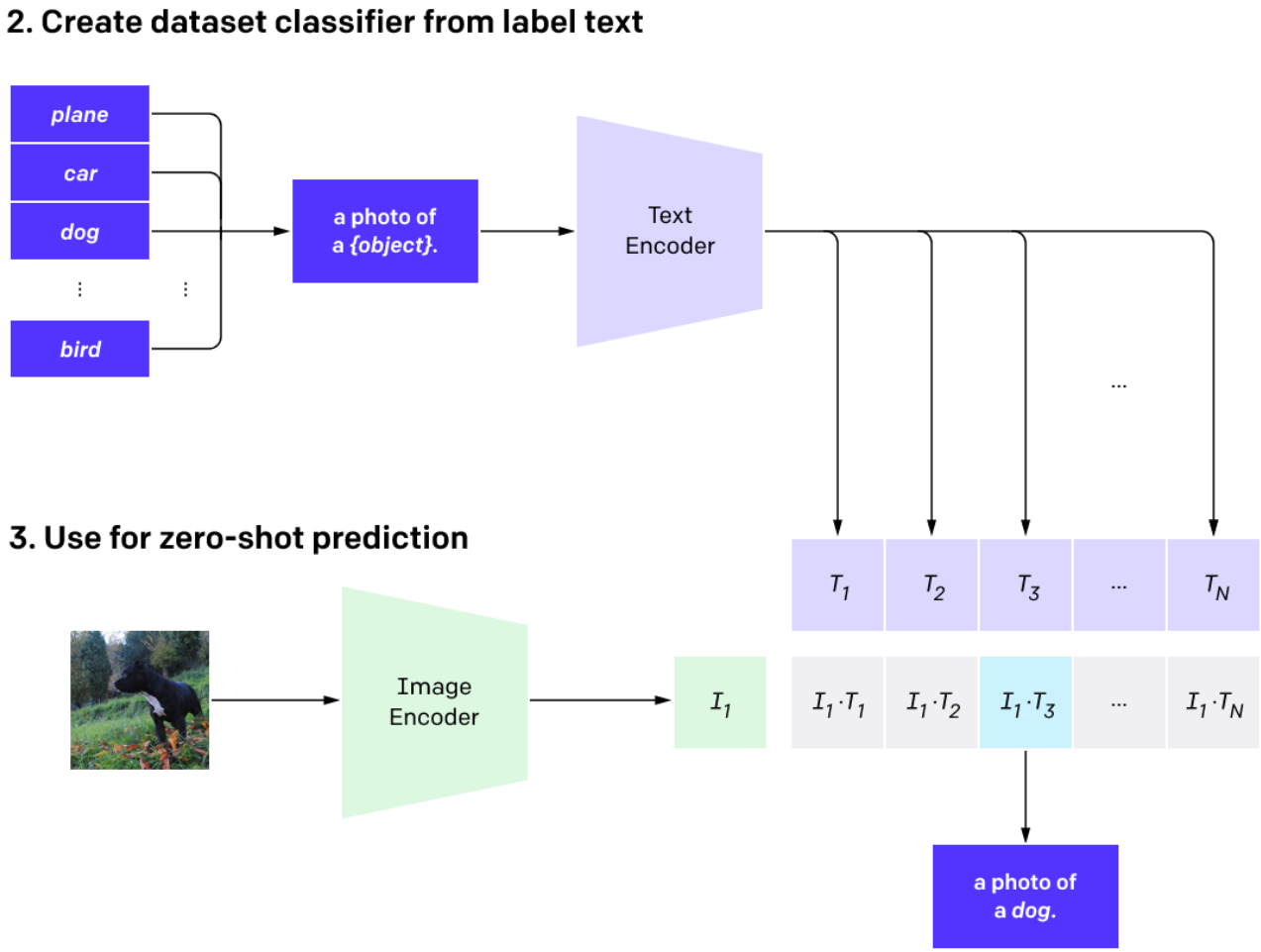
- CLIP is trained on a wide gamut of images with a wide variety of natural language supervision that's abundantly available: the text paired with images found across the internet

- Allows learning of a wide variety of concepts in images and their association with names

- Proxy training task for CLIP: given an image, predict which out of a set of 32,768 randomly sampled text snippets, was actually paired with it in the dataset.

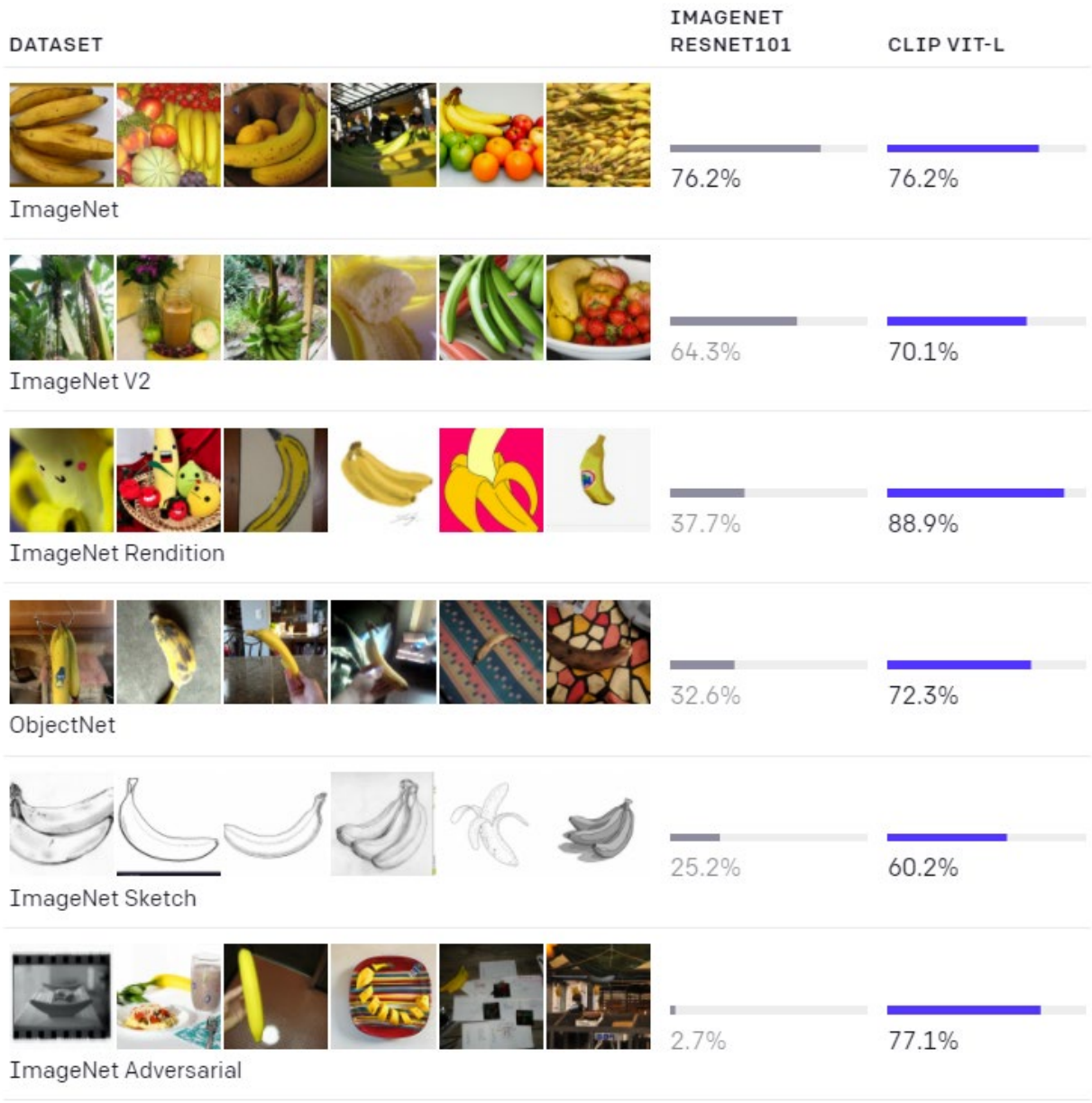
Clip Architecture





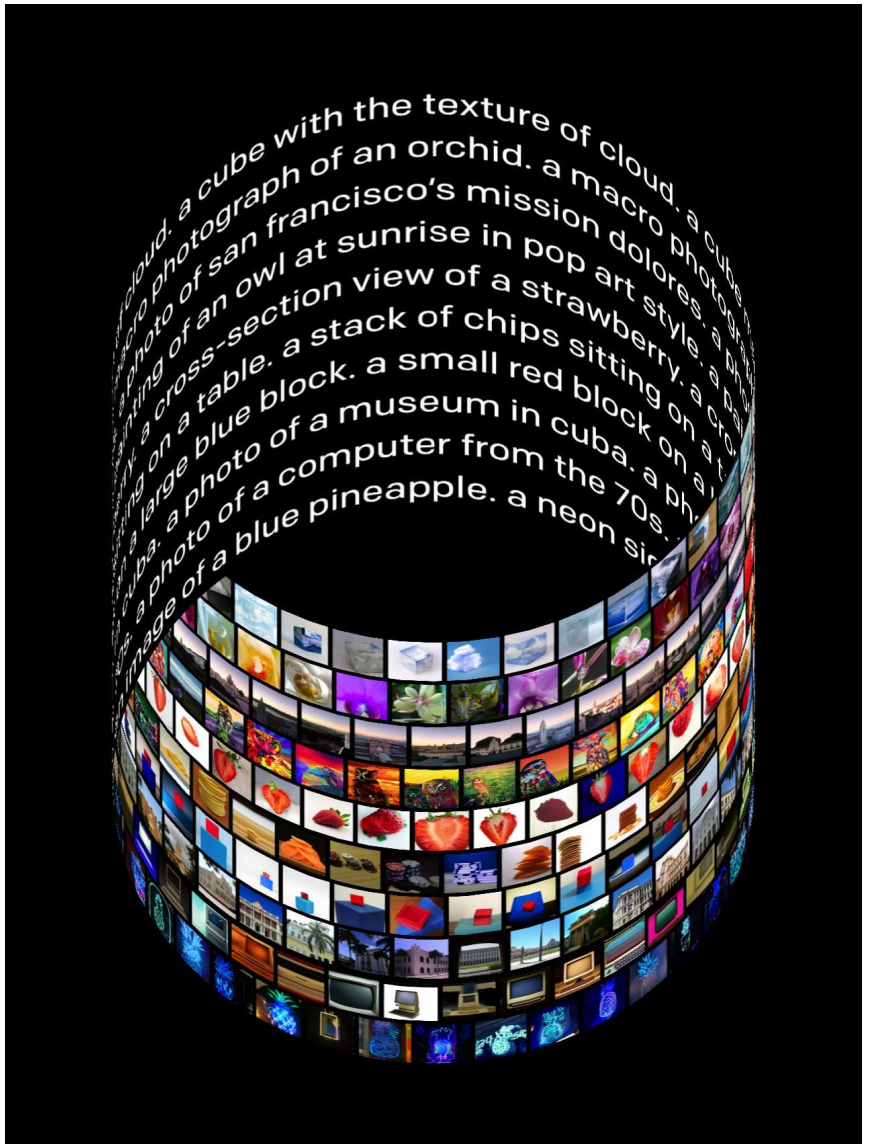
CLIP Generalizes Where Specialized Models Fail

Although both models have the same accuracy on the ImageNet test set, CLIP's performance is much more representative of how it will fare on datasets that measure accuracy in different, non-ImageNet settings. For instance, ObjectNet checks a model's ability to recognize objects in many different poses and with many different backgrounds inside homes while ImageNet Rendition and ImageNet Sketch check a model's ability to recognize more abstract depictions of objects.



DALL-E: Creating Images from Text

- DALL-E is a version o GPT-3 trained to generate images from text descriptions — gets supervision from CLIP
- DALL-E works by training a transformer to to autoregressively model text and image tokens as a single stream of data
- Stage 1 works by learning a visual codebook, training a variational autoencoder to compress images into visual tokens
- Stage 2 learns the prior distribution over text and images by building a decoder-only transformer where each image token can attend to all text tokens



DALL-E: S. Dali + WALL-E

Example Generations

TEXT PROMPT an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Edit prompt or view more images↓

TEXT PROMPT an armchair in the shape of an avocado. . . .

AI-GENERATED IMAGES

Edit prompt or view more images↓

Example Generations

TEXT PROMPT

a store front that has the word ‘openai’ written on it. . . .

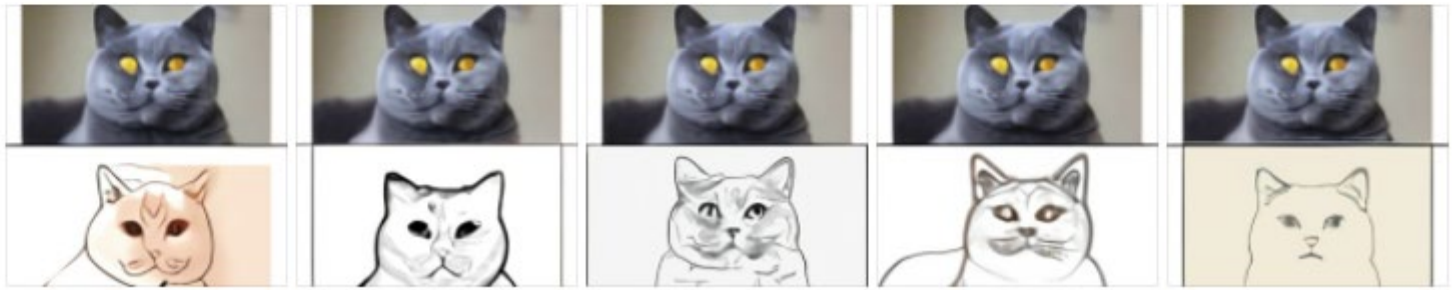
AI-GENERATED IMAGES

Edit prompt or view more images↓

TEXT & IMAGE PROMPT

the exact same cat on the top as a sketch on the bottom

AI-GENERATED IMAGES



Edit prompt or view more images↓

What is Missing?

- 3D
- Fine-grained generation control
  - realism
  - adherence to the language instructions
- Edits and modifications
  - realism
  - adherence to the language instructions
  - stability

Object Shape Differences in Language

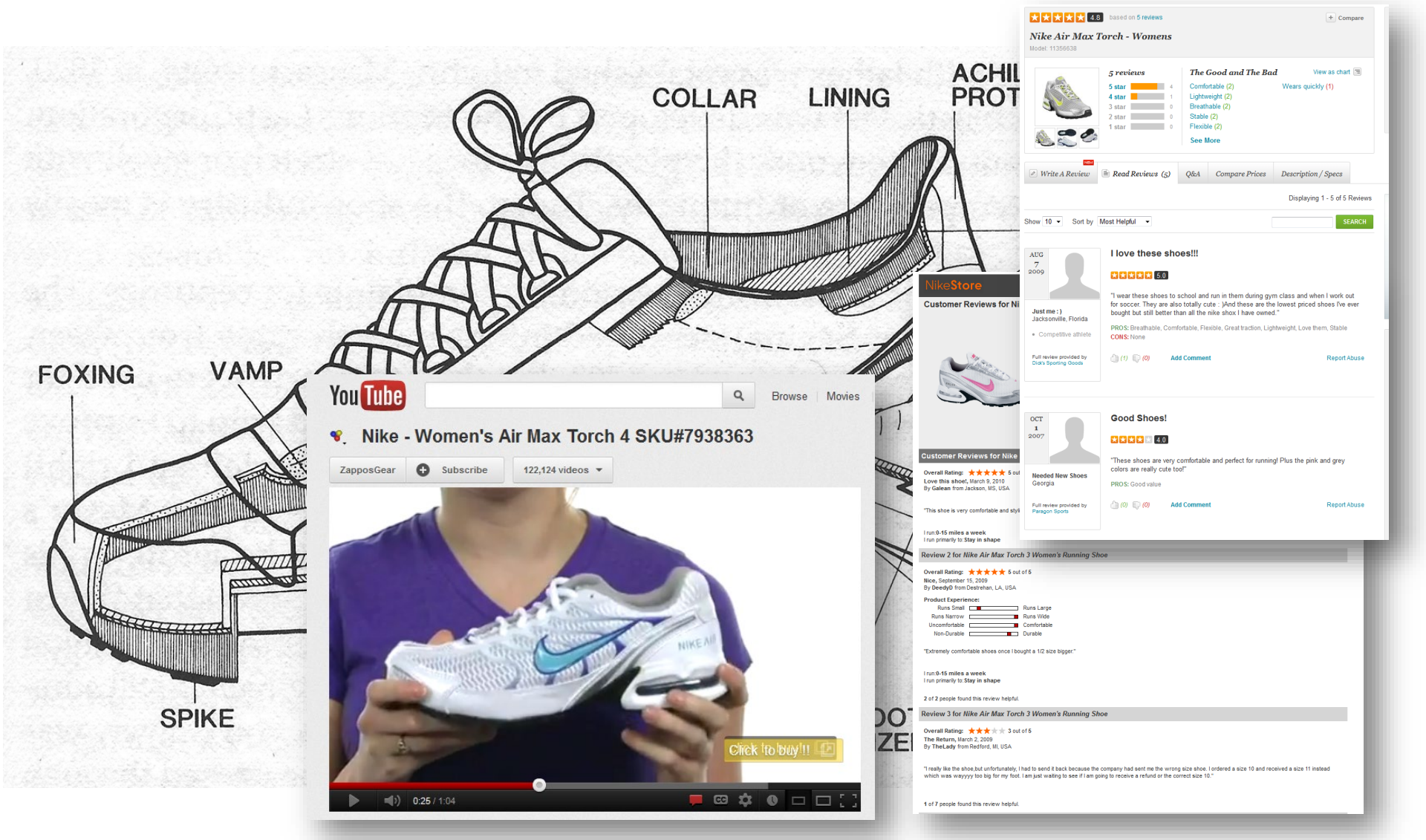
[P. Achlioptas, J. Fan, R. Hawkins, N. Goodman, L. Guibas; ICCV ‘19]

Differences in Geometry Expressed in Language

Target Object

“Gaps in the back”

Search Engines Based on Differences?



A Reference Language Game

‘looks like a sofa’

A Reference Language Game

nailed it

Utterance Examples

“it has wheels”

“has vertical lines on the back”

“rectangle back with straight legs”

“Chairs in Context” Corpus / Data Set

- 4,054 distinct contexts covering 4,511 chairs
- 78,789 utterances by 2,124 unique AMT participants

Easy: 97.2% (L.6.2)

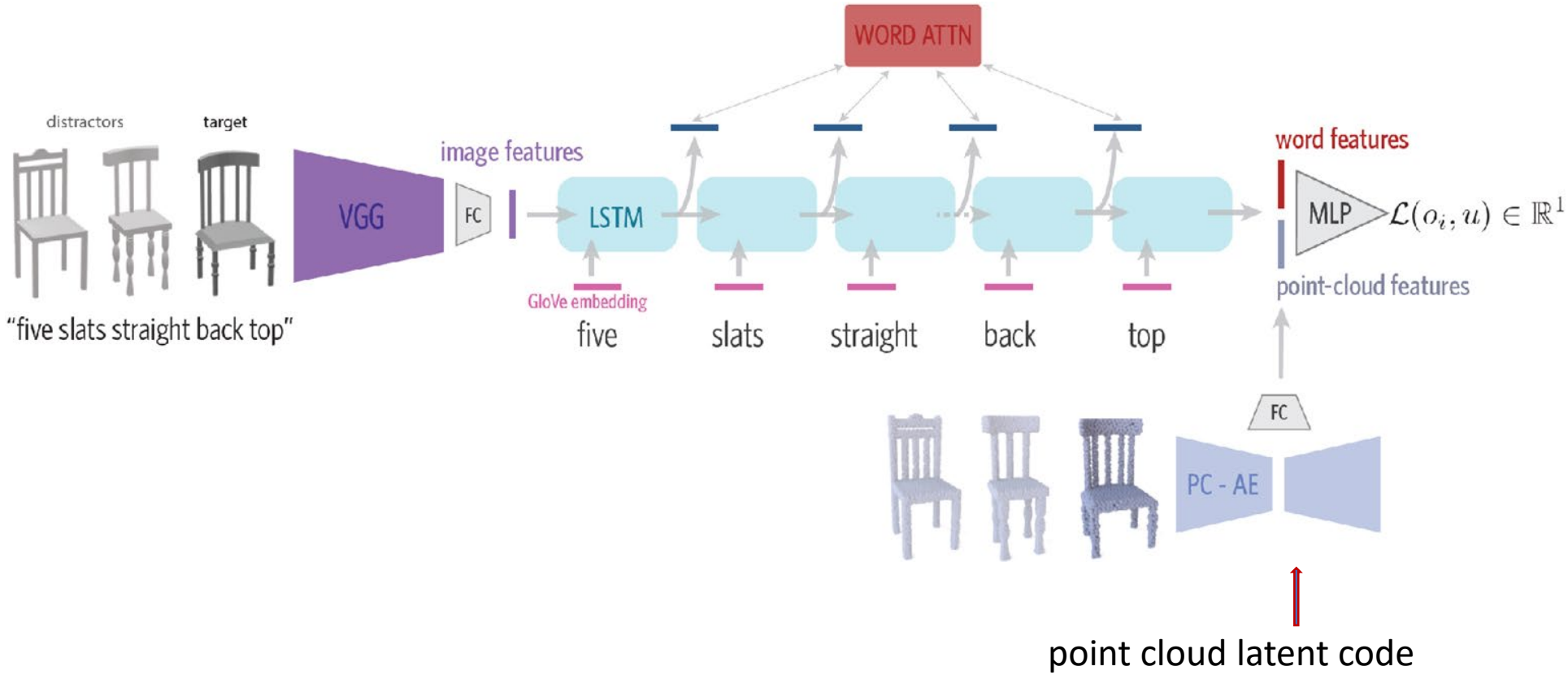
“no arms”

Hard: 94.2% (L.8.4)

“rectangle back with straight legs”

Deep Neural Listener w. Attention

- Tap on good (pre-trained) **visual/3D** representations
- **Attend** to 'important' words
- Explore effect of different **contrasting architectures**

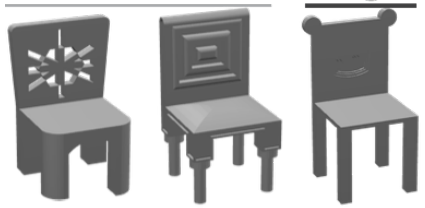


Attentive Neural Listening

Listener Failures

Metaphors

distractors target



0.39 0.27 0.34

"mickey mouse ears"

distractors target

0.48 0.13 0.39

"if it were a man's suit it would have tails"

Counting

0.63 0.02 0.35

"six bars on back"

Material

0.05 0.86 0.09

"bold steel"

Ambiguous

0.56 0.16 0.28

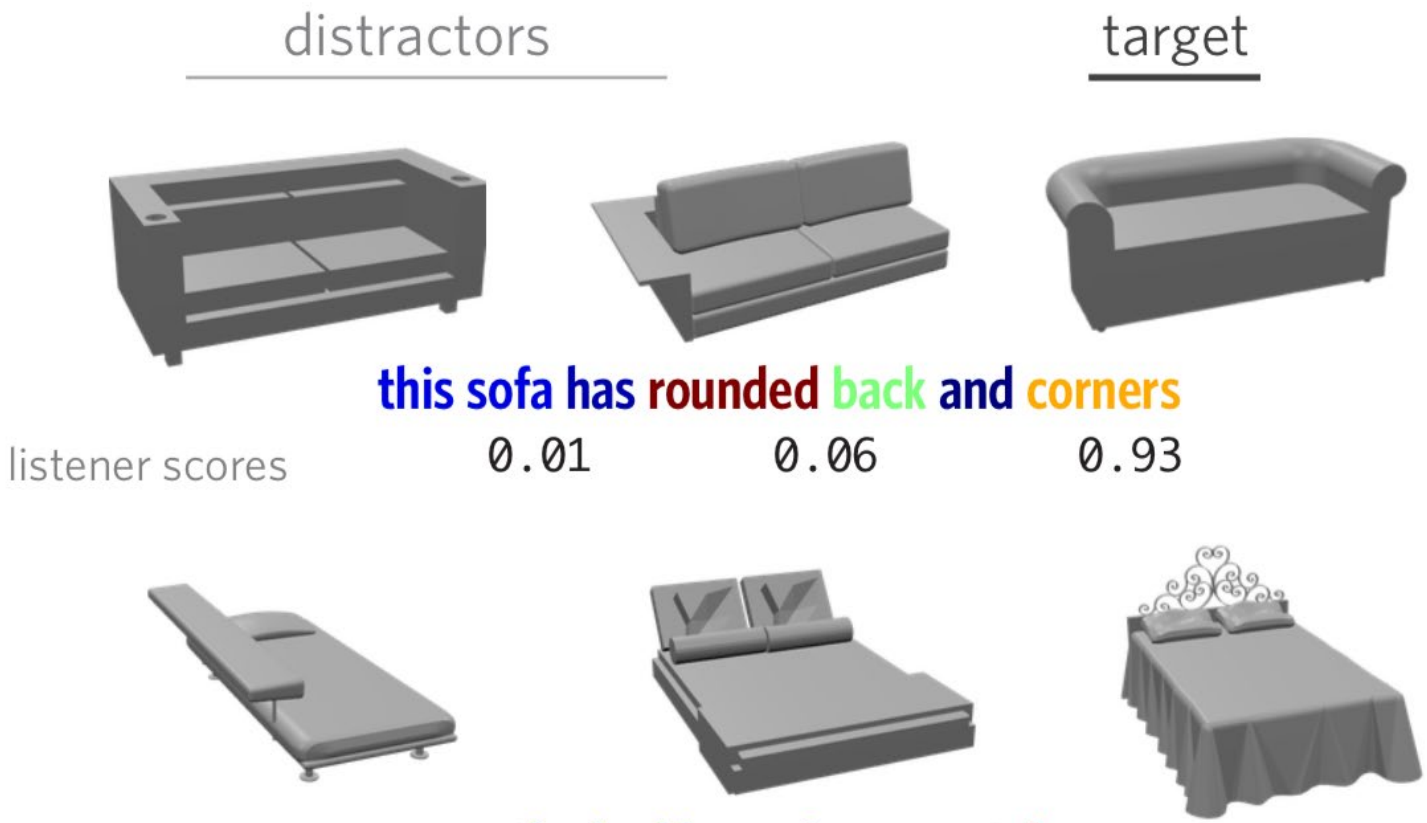
"the chair that is the skinniest"

Negation

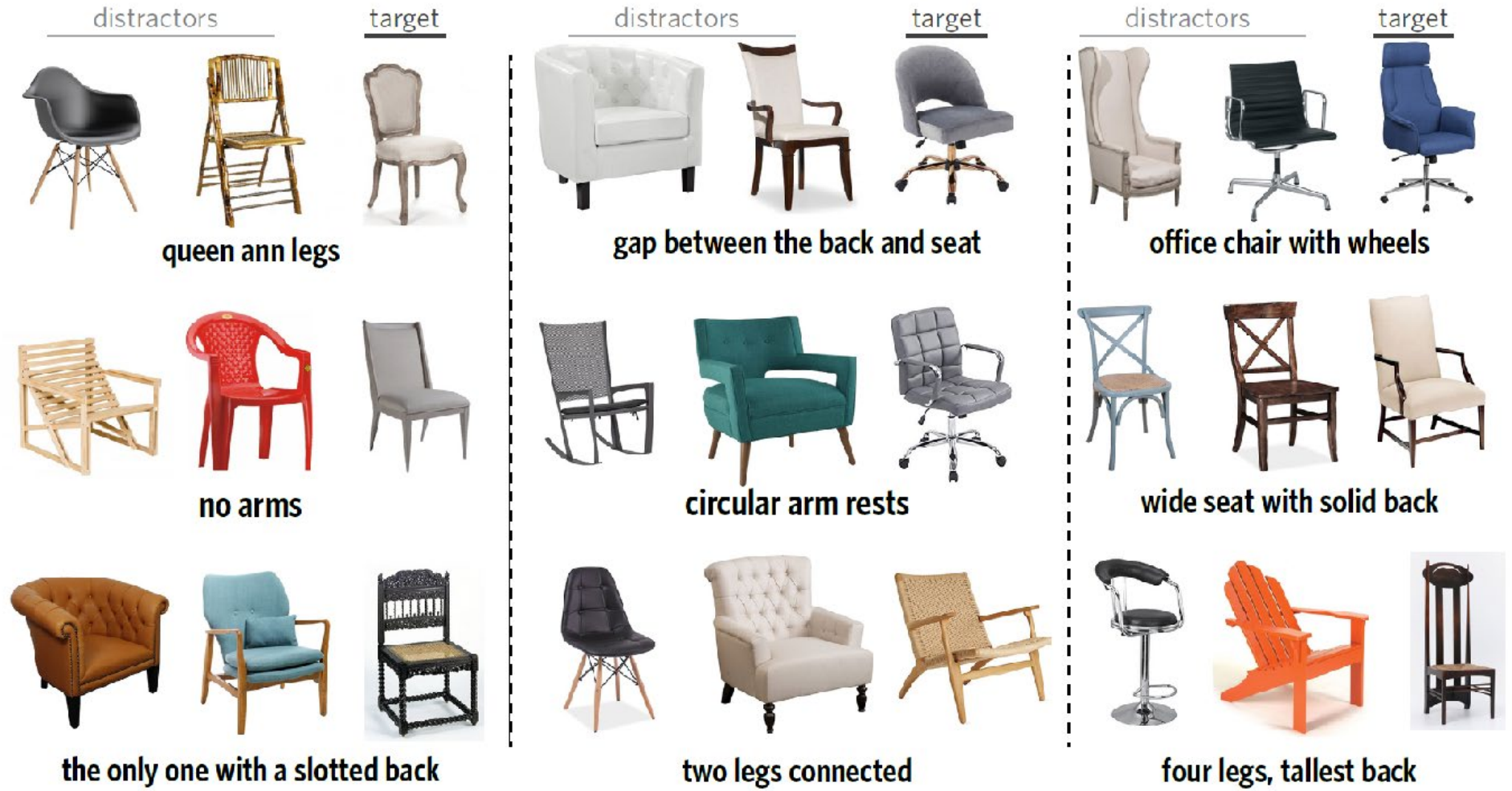
0.57 0.13 0.30

"open arms, no rails on back"

Zero-Shot Generalization to New Classes/Language

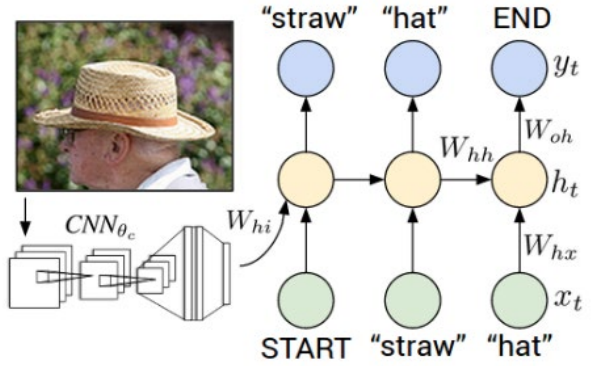


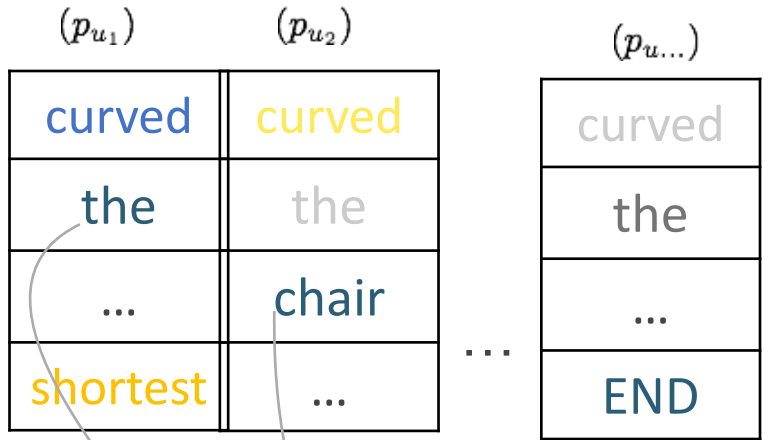
Speaker from Images

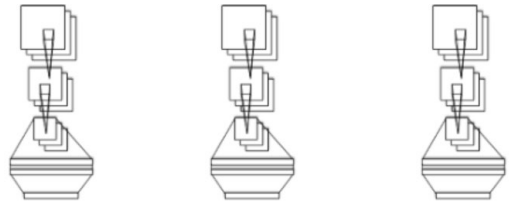


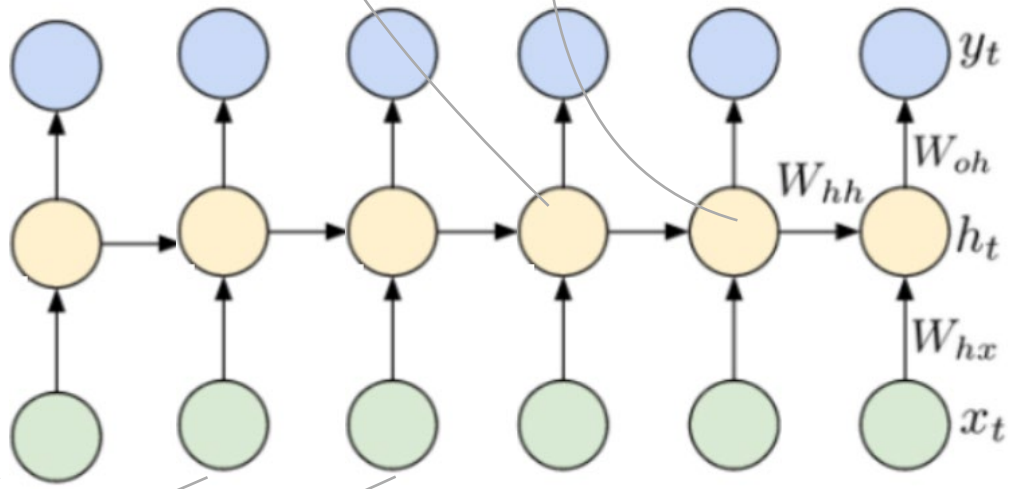
Neural Speaker

show-and-tell w. teacher forcing



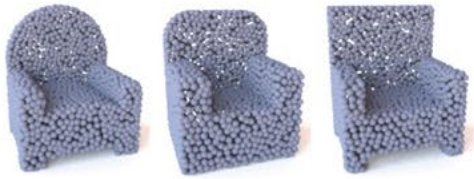




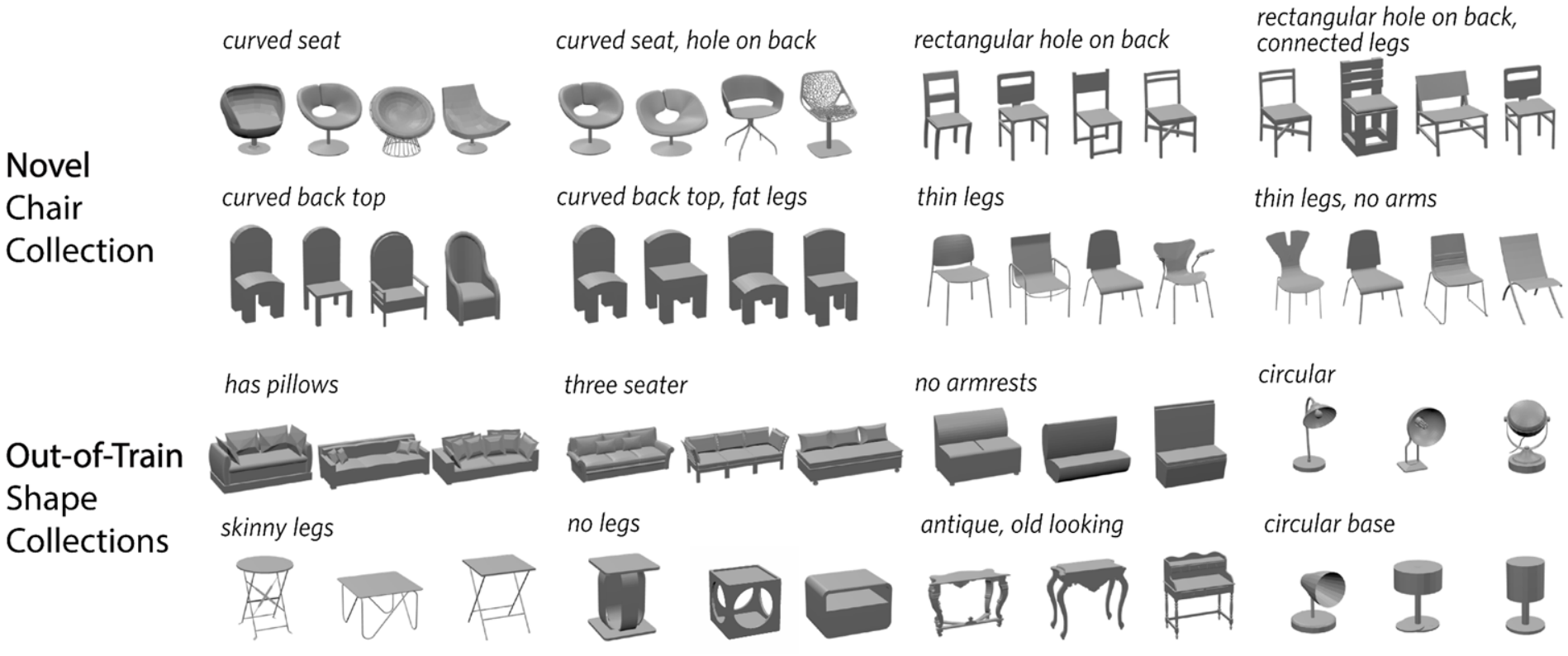


START

Unseen Speaker Examples

	distractors / target	distractors / target	distractors / target
image-based speakers			
pragmatic speaker	**square arms**	**knobby legs**	**no arm rests**
literal speaker	with the tall-est back and seat	the one with the thick-est legs	the one with high-est back
point-cloud based speakers			
pragmatic speaker	**most square back**	**thick-est legs**	**tall-est back**
literal speaker	thin-est seat	square rack at bottom of chair	has arms

Listener Examples: Shape-Based Product Retrieval



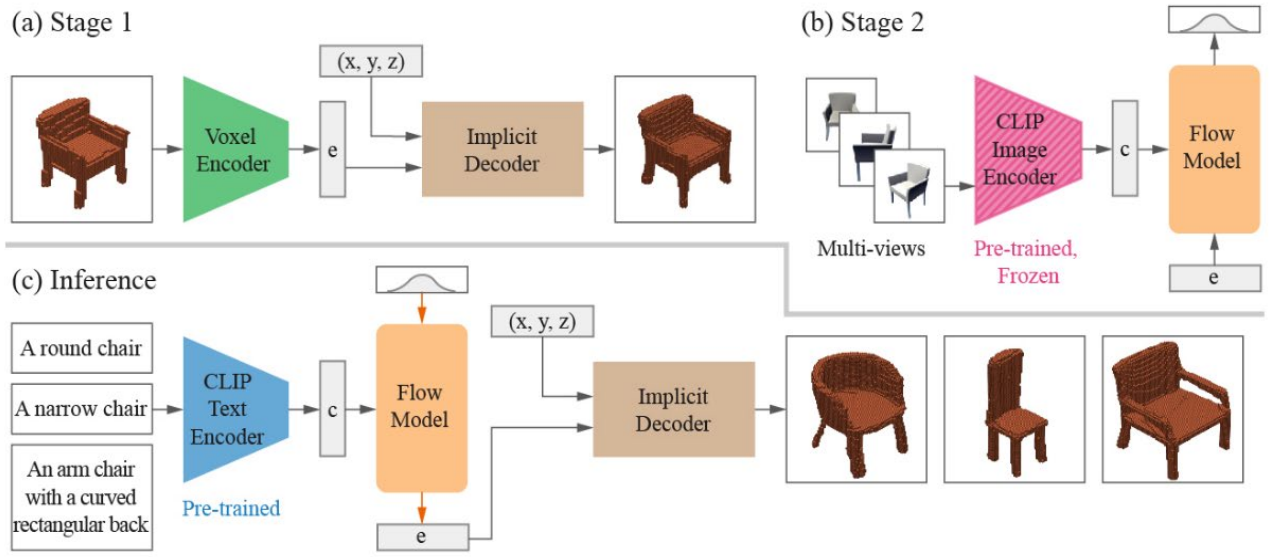
(bottom rows includes ***out-of-training*** classes)

Language-Driven 3D Shape Editing

Ian Huang

The motivation

CLIP-Forge



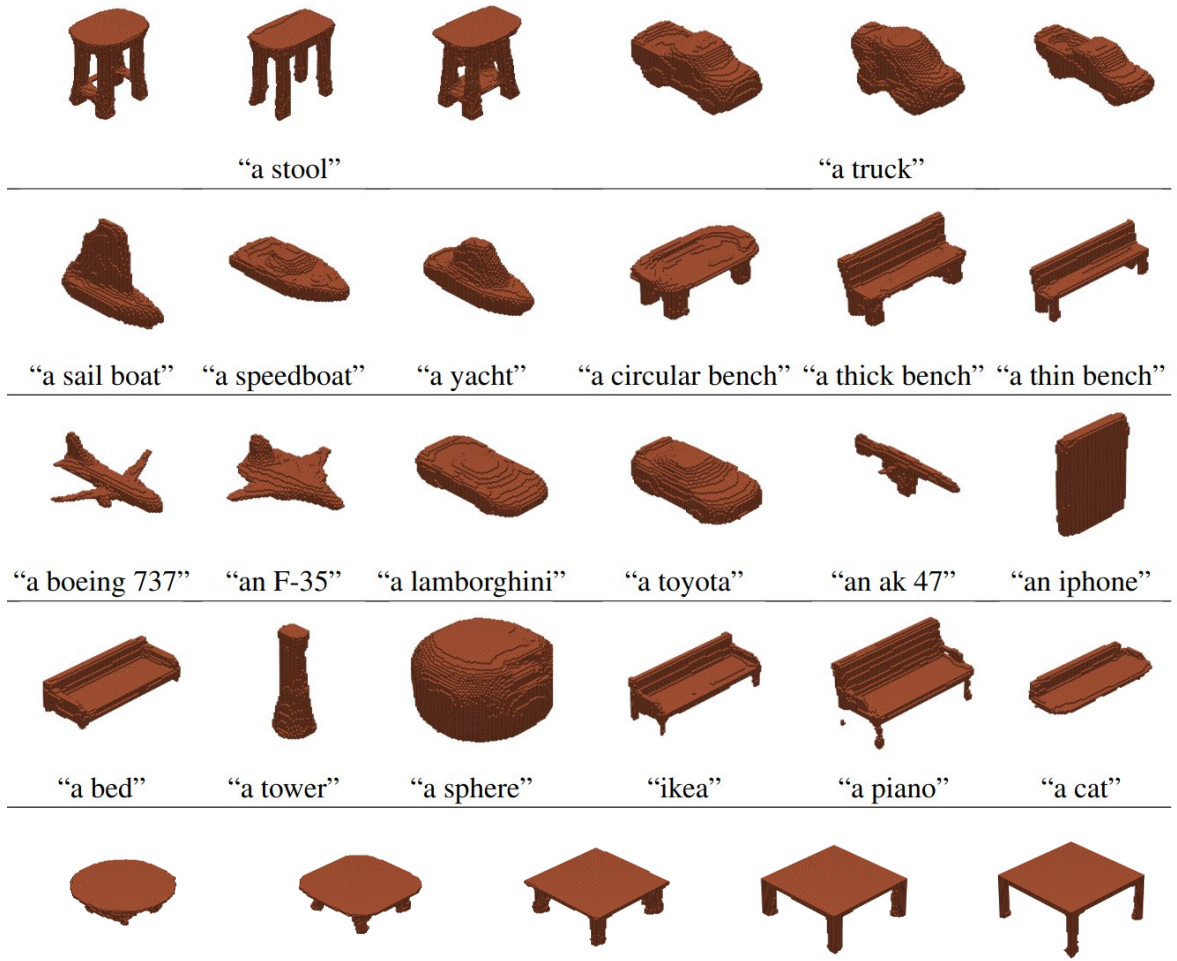
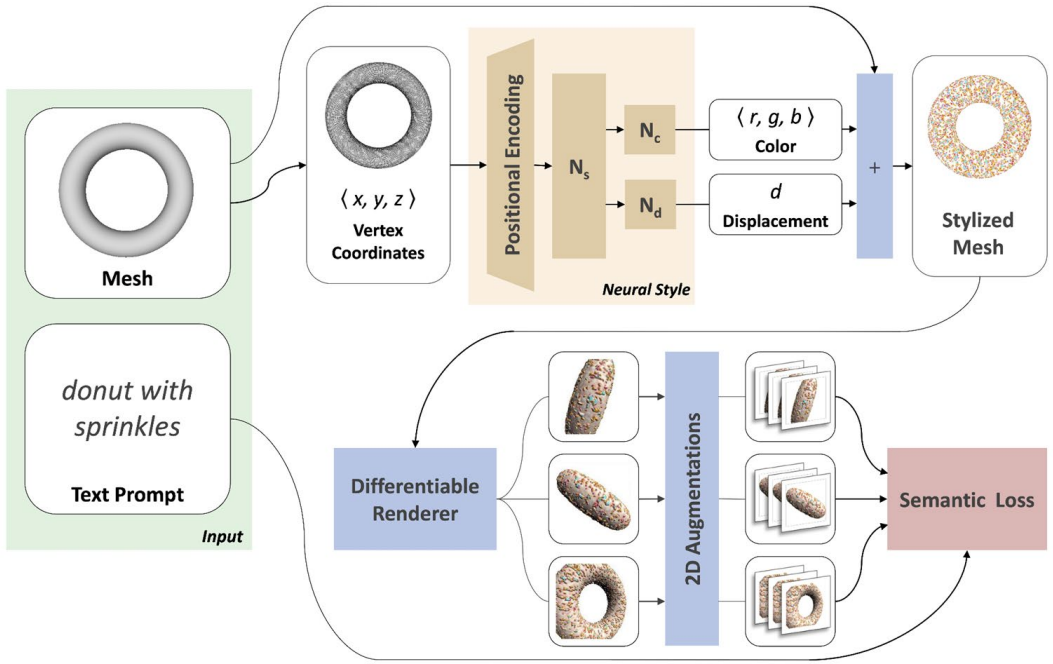


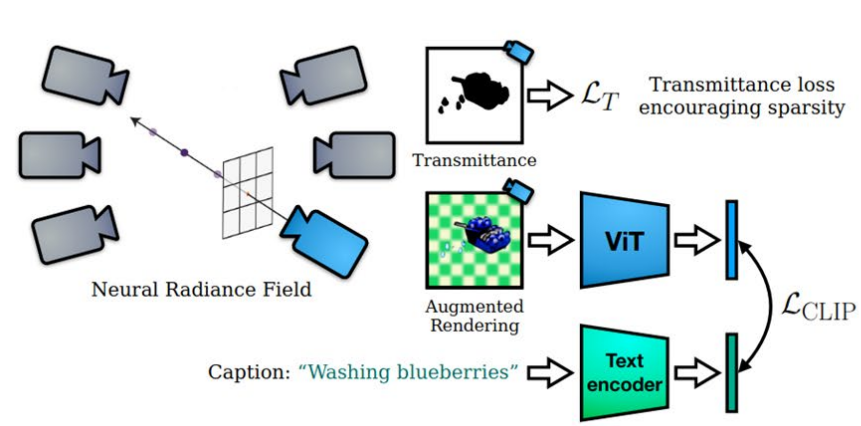
Figure 4: CLIP-Forge generations by interpolating from "a round table" to "a square table".

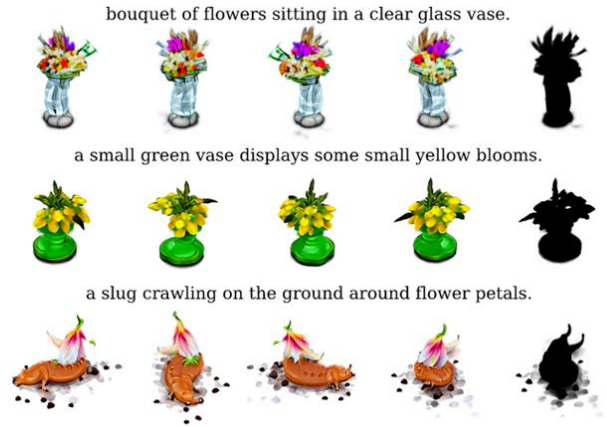
Text2mesh

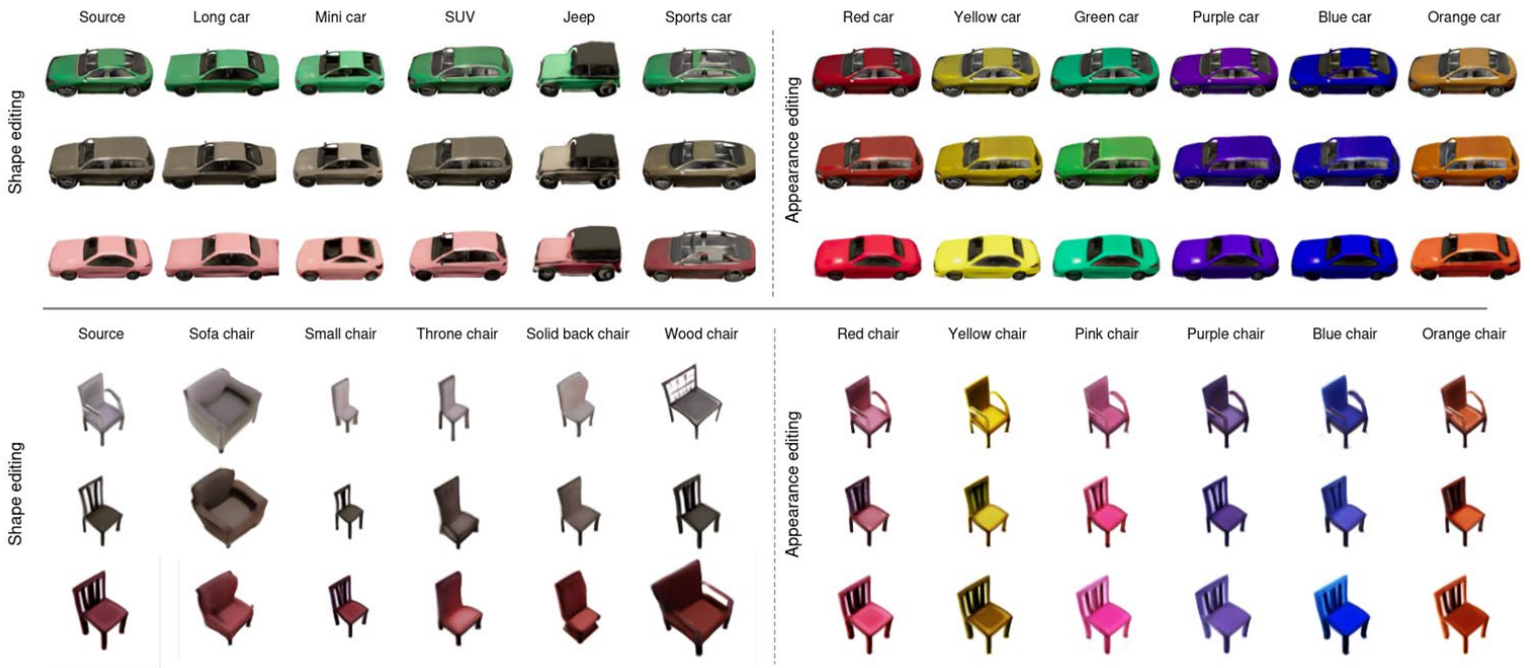

Mesh
donut with sprinkles
Text Prompt
Input
⟨ x, y, z ⟩
Vertex Coordinates
Positional Encoding
N_s
N_c
N_d
⟨ r, g, b ⟩
Color
d
Displacement
Neural Style
+
Stylized Mesh
Differentiable Renderer
2D Augmentations
Semantic Loss


Input
Steve Jobs
Yeti
Astronaut
Buzz Lightyear
Ninja
Lawyer
Messi
Batman
Hulk

DreamFields & CLIP-NERF







Solved?

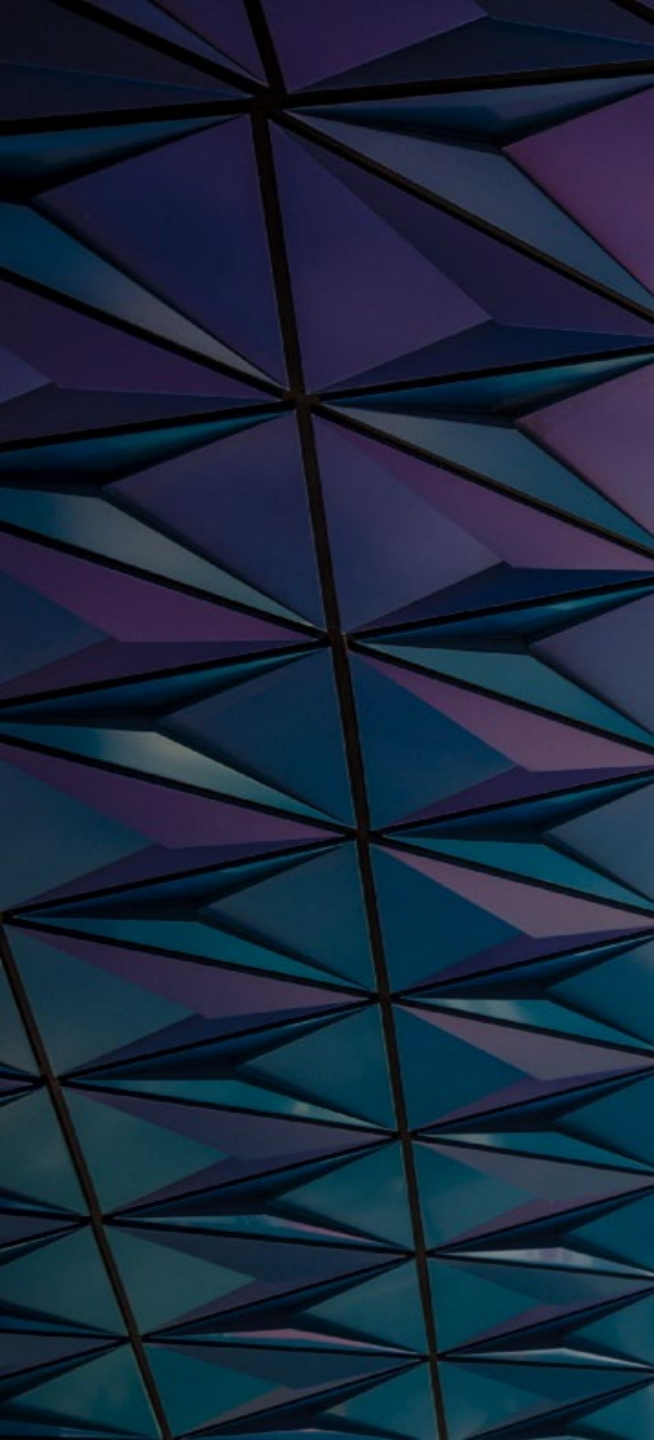
"A fireman"

"A fireman with black boots"

“No arm”

Outline

- Studying the relationship between parts and language through reference games only.
- The different datasets used for shape datasets
- Directly editing structure, supervised by ground-truth differences
- Attempting to edit by moving around an edit latent space
- Is it a data problem? Making it simpler by attempting to learn a distance metric in a joint language-geometry space.
- Our current data problem, and the our efforts to bridge the gap.

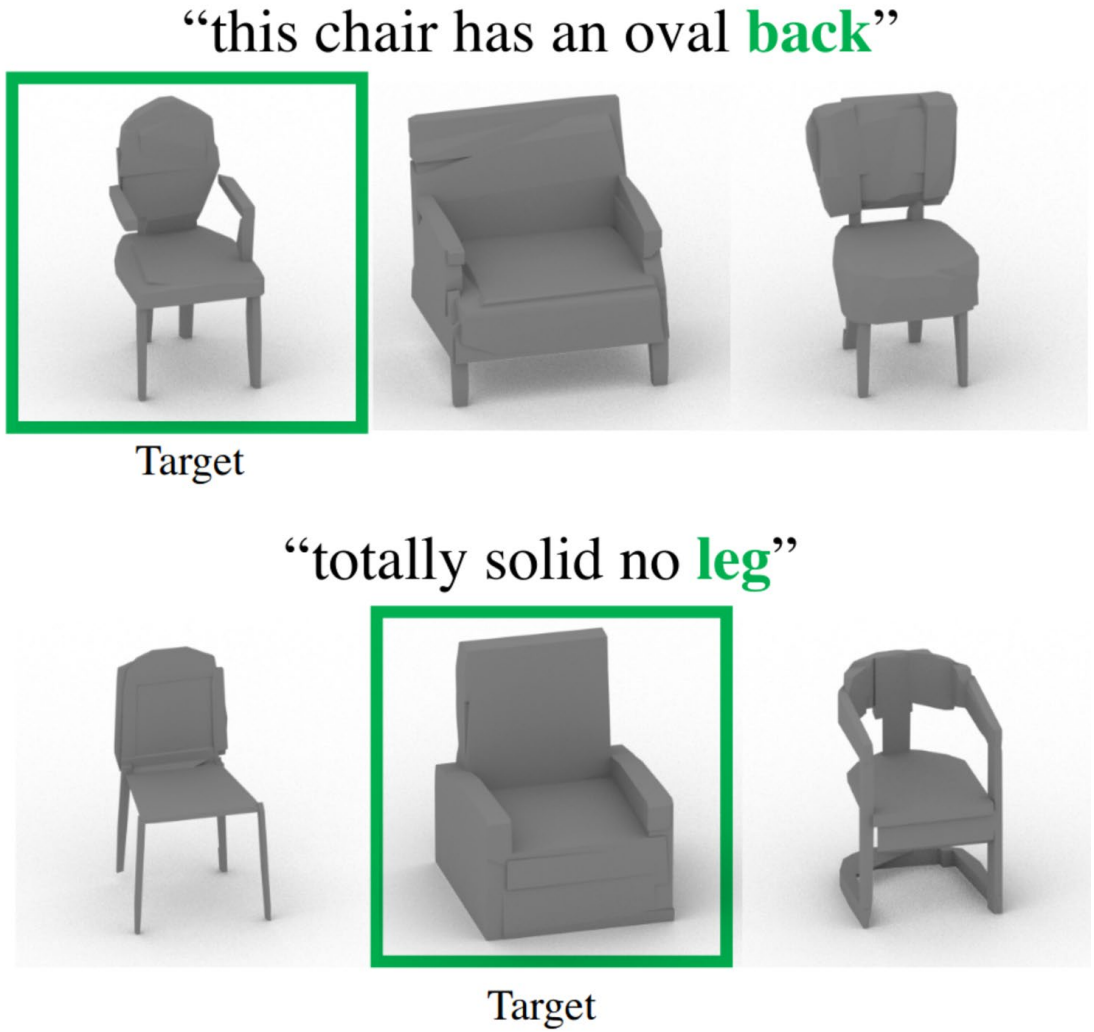


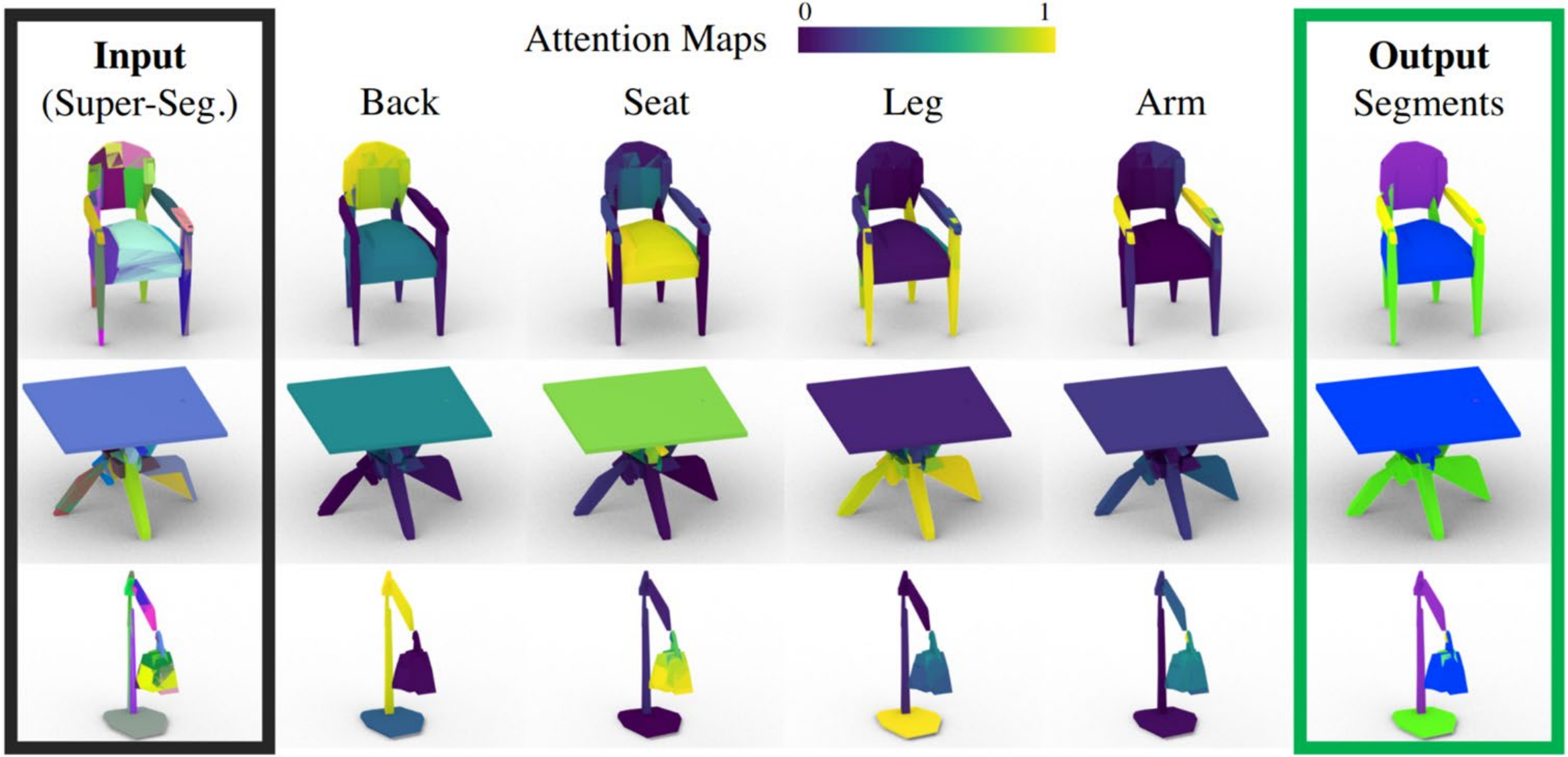
PartGlot: Learning Shape Part Segmentation from Language Reference Games

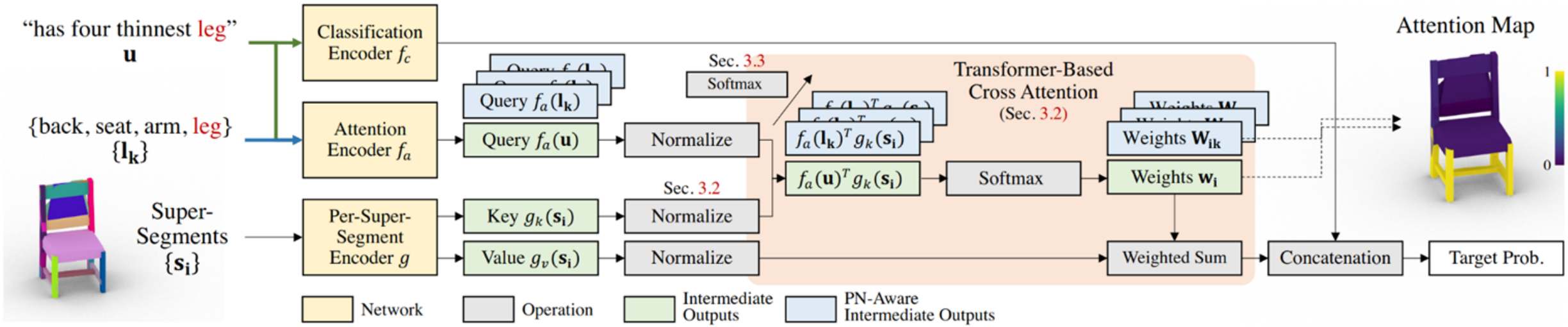
Juil Koo, Ian Huang, Panos Achlioptas, Leonidas Guibas, Minhyuk Sung

CVPR 2022

Language Reference Games

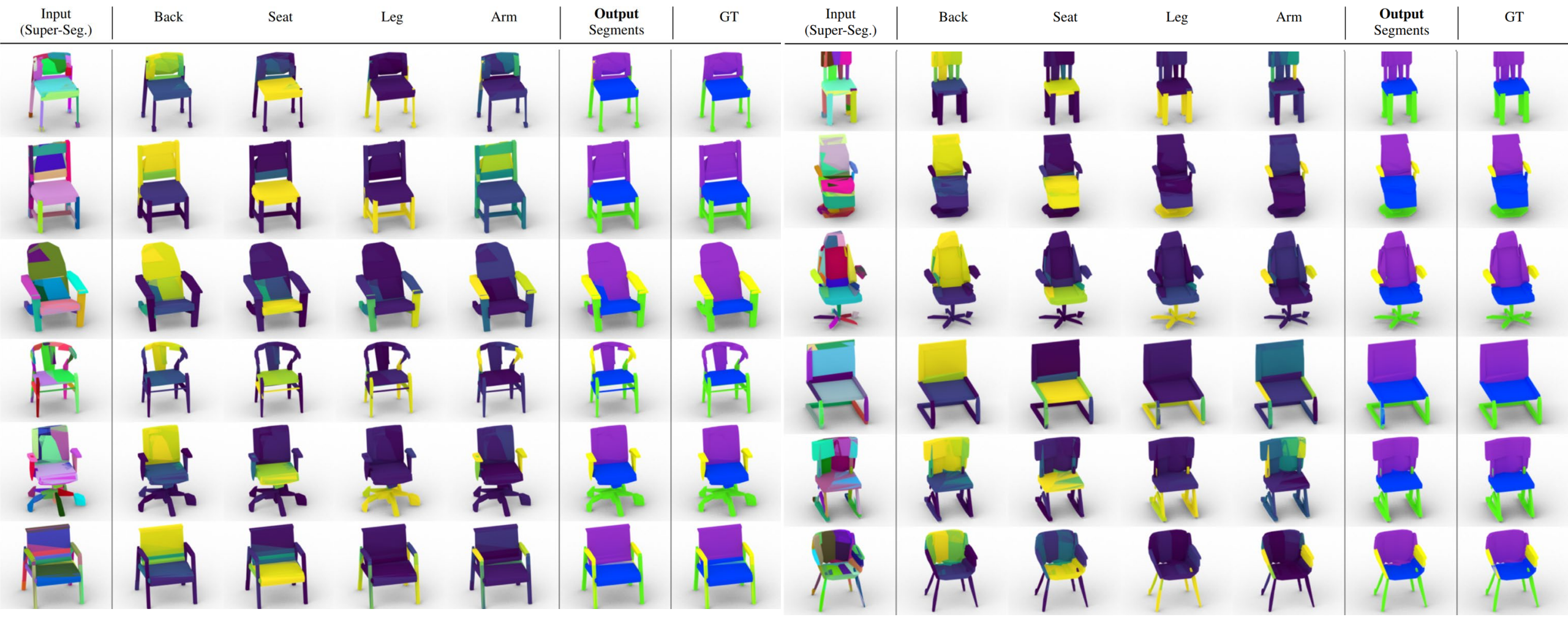



Input
(Super-Seg.)
Attention Maps
0
1
Back
Seat
Leg
Arm
Output
Segments

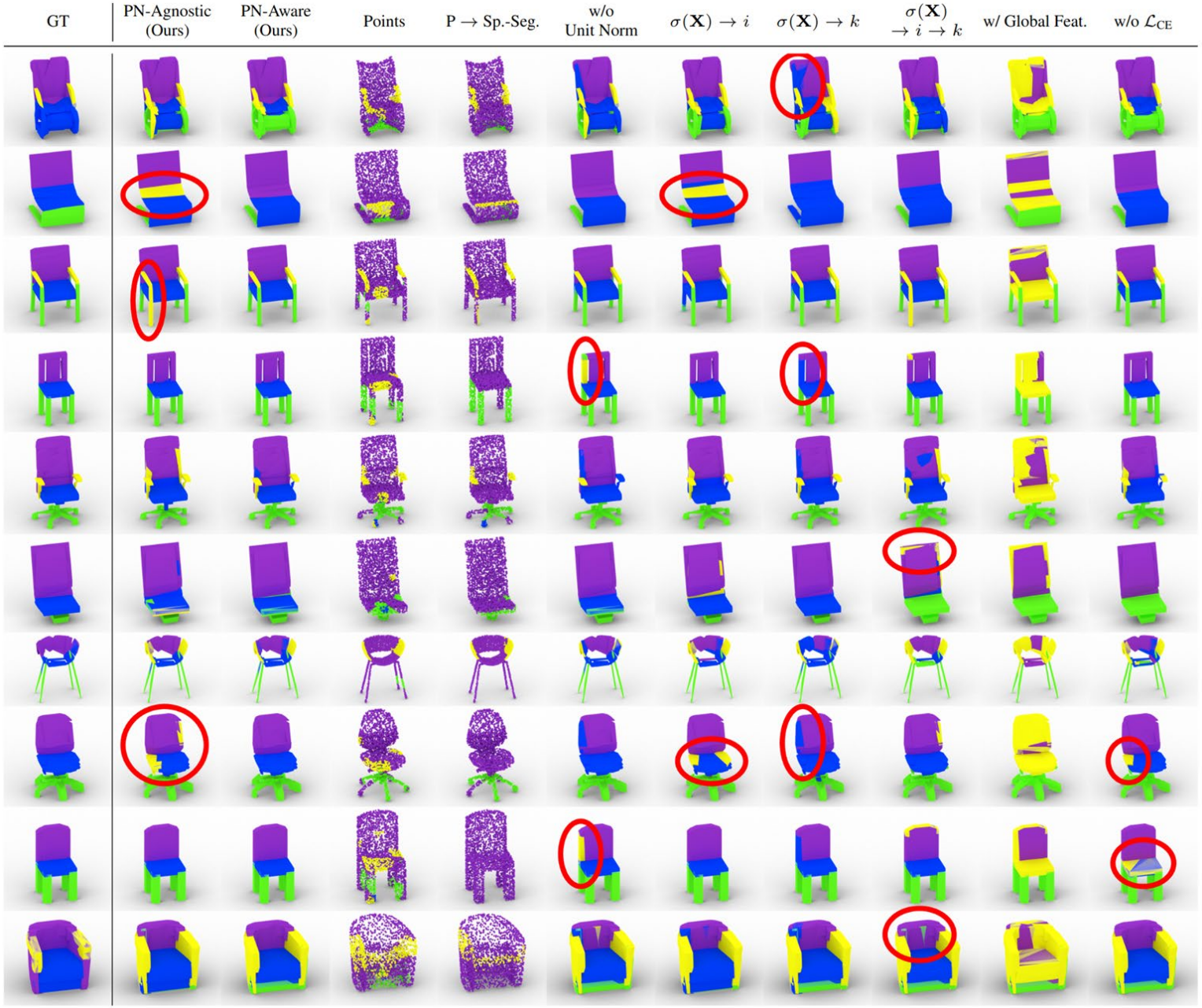


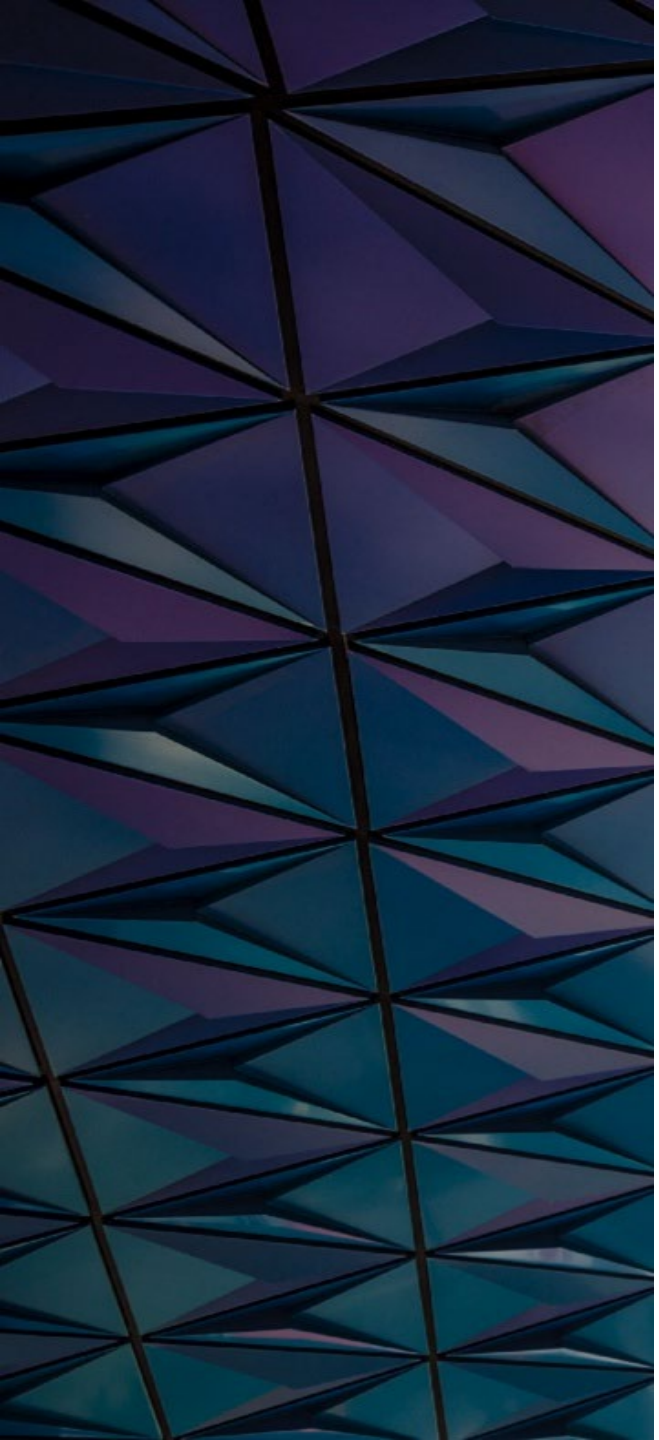
In addition to standard classification loss,

$$\mathcal{L}_{\text{CE}} = \sum_i \sum_k -\mathbb{1}\left(k = \arg\max_{k'}(\mathbf{Y_{ik'}})\right) \log(\mathbf{Y_{ik}})$$


Input (Super-Seg.)
Back
Seat
Leg
Arm
Output Segments
GT
Input (Super-Seg.)
Back
Seat
Leg
Arm
Output Segments
GT

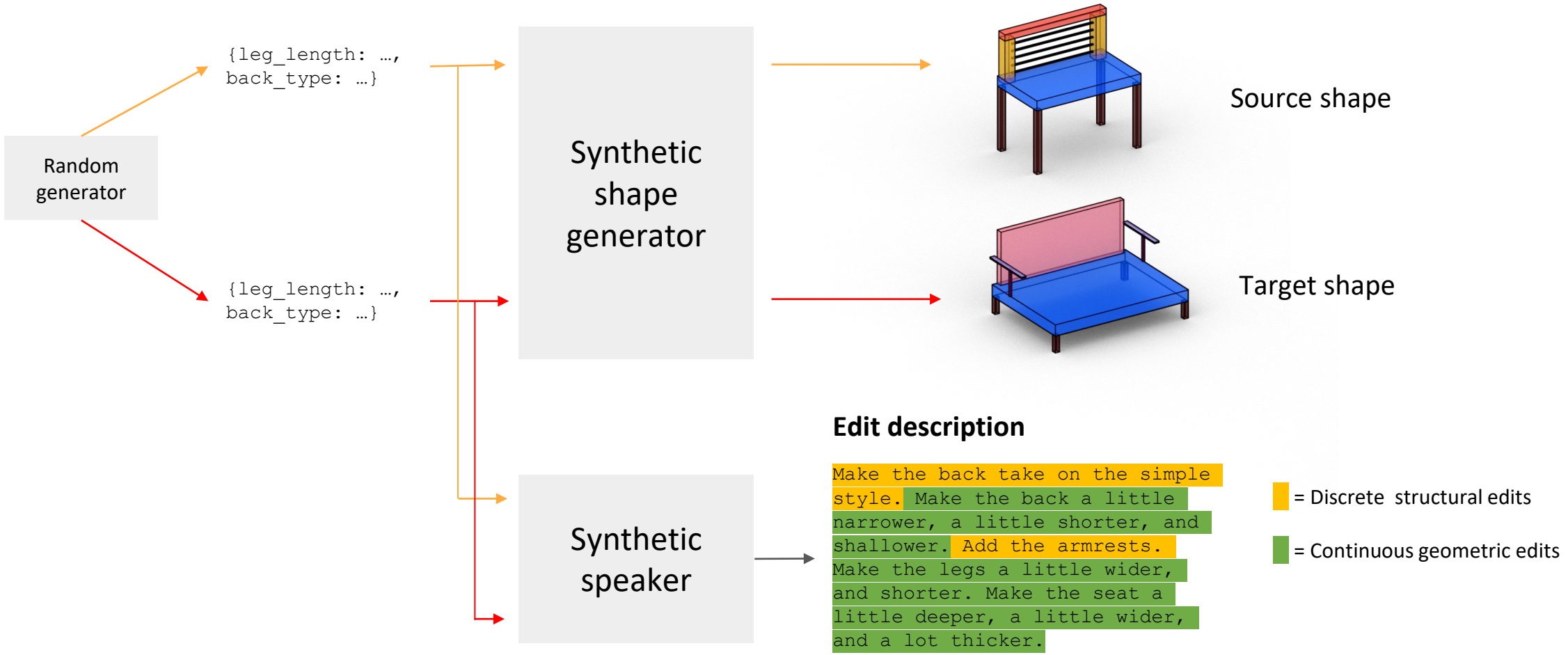
Id	Method	Segmentation mIoU(%)					Classif. Acc.(%)
		Back	Seat	Leg	Arm	Avg.	
PN-Agnostic (Sec. 3.2) vs. PN-Aware (Sec. 3.3)							
1	PN-Agnostic (Ours)	82.2	78.8	75.5	40.6	69.3	61.6
2	PN-Aware (Ours)	**84.9**	**83.6**	**78.9**	70.4	**79.4**	61.5
Points vs. Super-Segments (w/ PN-Aware)							
3	Points	40.7	0.2	38.1	10.8	22.5	57.2
4	P $\rightarrow$ Sp.-Seg.	39.2	0	44.1	63.3	36.6	57.2
5	**Sp.-Seg. (Ours)**	**84.9**	**83.6**	**78.9**	70.4	**79.4**	61.5
6	Upper Bound*	89.8	88.9	85.2	92.3	89.1	-
Ablation Study (w/ PN-Aware)							
7	w/o Unit Norm	78.5	81.0	77.4	54.4	72.8	63.0
8	$\sigma(\mathbf{X}) \rightarrow i$	80.8	77.5	75.3	56.6	72.5	**63.4**
9	$\sigma(\mathbf{X}) \rightarrow k$	73.8	76.1	75.8	**79.8**	76.4	61.9
10	$\sigma(\mathbf{X}) \rightarrow i \rightarrow k$	79.4	80.3	74.1	35.1	67.2	59.0
11	w/ Global Feat.	38.6	0.2	77.7	4.6	30.3	62.2
12	w/o $\mathcal{L}_{\text{CE}}$	82.6	79.7	77.4	71.4	77.8	59.8
Few-Shot Learning (w/ PN-Aware)							
13	k=1	85.5	83.5	78.4	73.2	80.1	59.4
14	k=8	86.1	84.2	78.9	70.6	79.9	60.0
15	k=32	86.9	84.8	79.5	76.5	81.9	59.7





Synthetic Data / Existing Datasets

Dataset 1: Synthetic Language + Synthetic Shapes



This way, we get dataset of triplets: (source, target, description) = (, ,)

Dataset 2: Synthetic Language + Partnet Shapes

source

target

```
(('back_single_surface', (), 'back_single_surface'),), 'move away from', (('leg', ('left of', 'infront of'), 'back_single_surface'), ('leg', ('left of', 'infront of'), 'seat_single_surface'))
(('back_single_surface', (), 'back_single_surface'),), 'make shallower'
(('back_single_surface', (), 'back_single_surface'),), 'shift up'
(('back_frame_vertical_bar', ('right of',), 'back_single_surface'), ('back_frame_vertical_bar', ('right of',), 'seat_single_surface')), 'ADD'
(('leg', ('right of', 'behind'), 'back_single_surface'), ('leg', ('right of', 'behind'), 'seat_single_surface')), 'make shallower'
(('leg', ('right of', 'behind'), 'back_single_surface'), ('leg', ('right of', 'behind'), 'seat_single_surface')), 'make narrower'
(('seat_single_surface', (), 'seat_single_surface'),), 'make deeper'
(('seat_single_surface', (), 'seat_single_surface'),), 'move away from', (('back_single_surface', (), 'back_single_surface'),)
(('seat_single_surface', (), 'seat_single_surface'),), 'make taller'
```

Dataset 3: Chairs in Context (ShapeGlot)

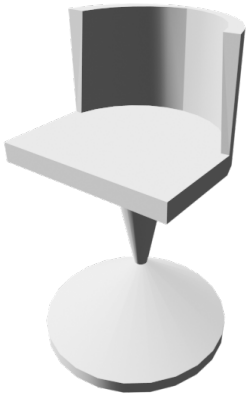
Source 1 Source 2 Target

the side of this looks like the bread on a sandwhich

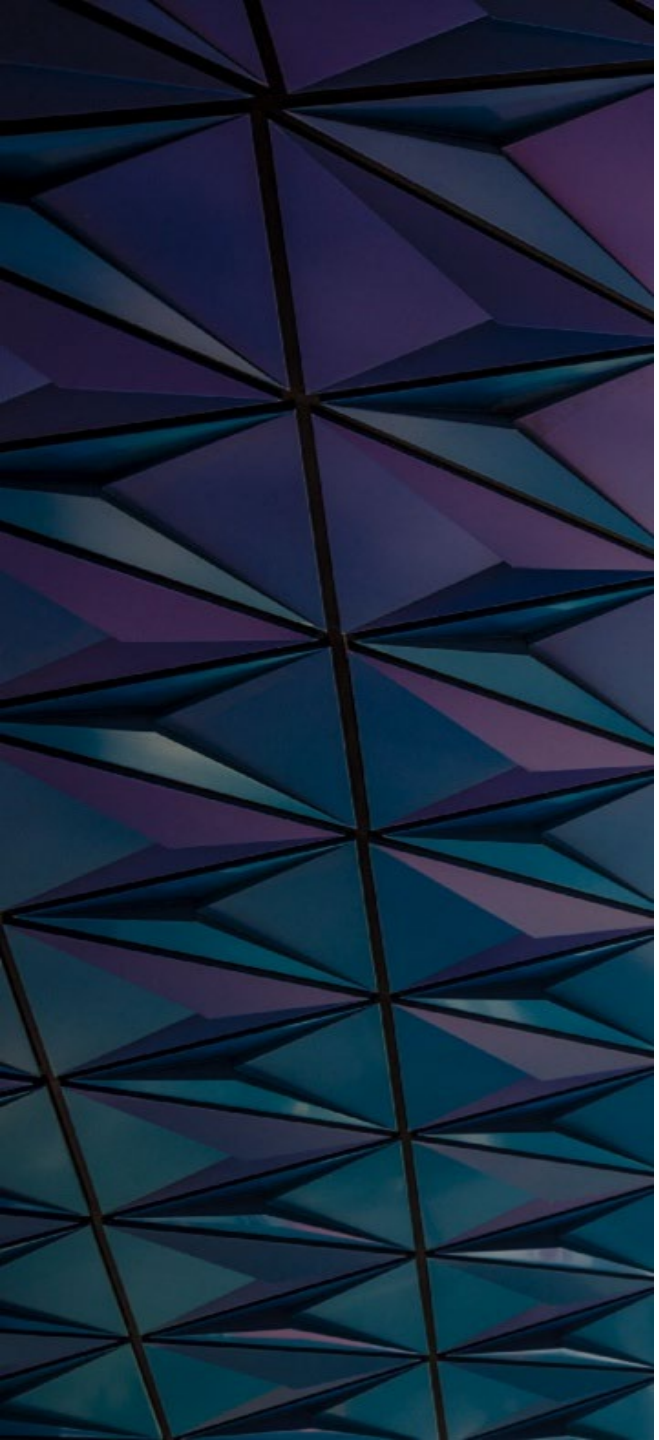
Dataset 4: PartIt (VLGrammar)

chair with back seat and seat , two thin short legs, two horizontal and two vertical leg bars.

the chair has a back, a seat, vertical arm bars, horizontal arm bars, and legs.



the parts of the chair are as follows: the chair back, the chair seat, the central support, and the pedastal base.

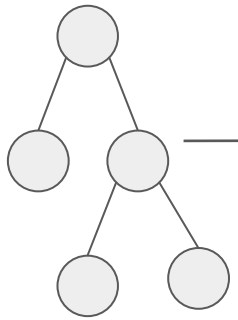


Directly editing structure

Directly editing

Edit description

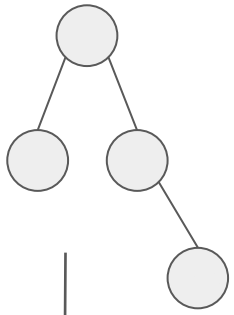
Make the back take on the simple style. Make the back a little narrower, a little shorter, and shallower. Add the armrests. Make the legs a little wider, and shorter. Make the seat a little deeper, a little wider, and a lot thicker.

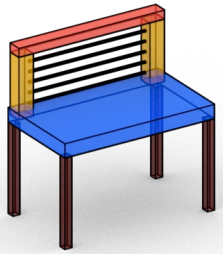


Part hierarchy

Structural diff prediction

“Given object part tree and the description, how should I change its structure?”

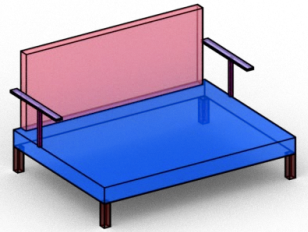




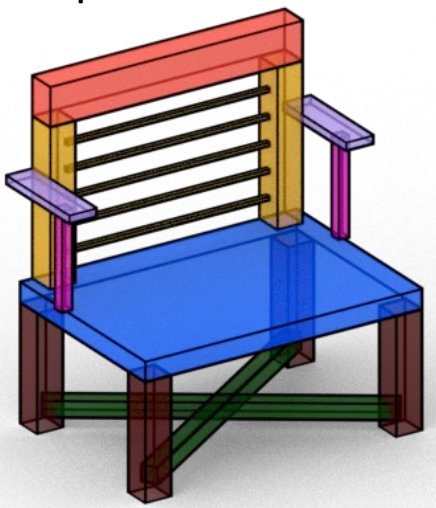
Bounding box rep.

Continuous param diff prediction

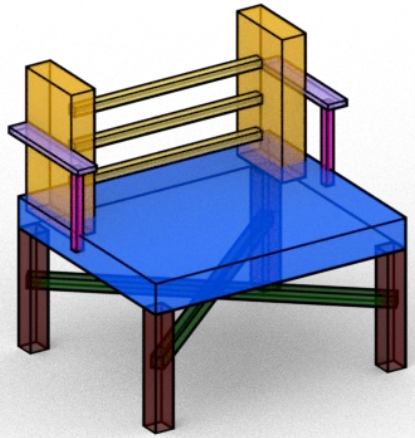
“How should I regress the source shape parameters with the predicted structure such that it’s consistent with the description?”



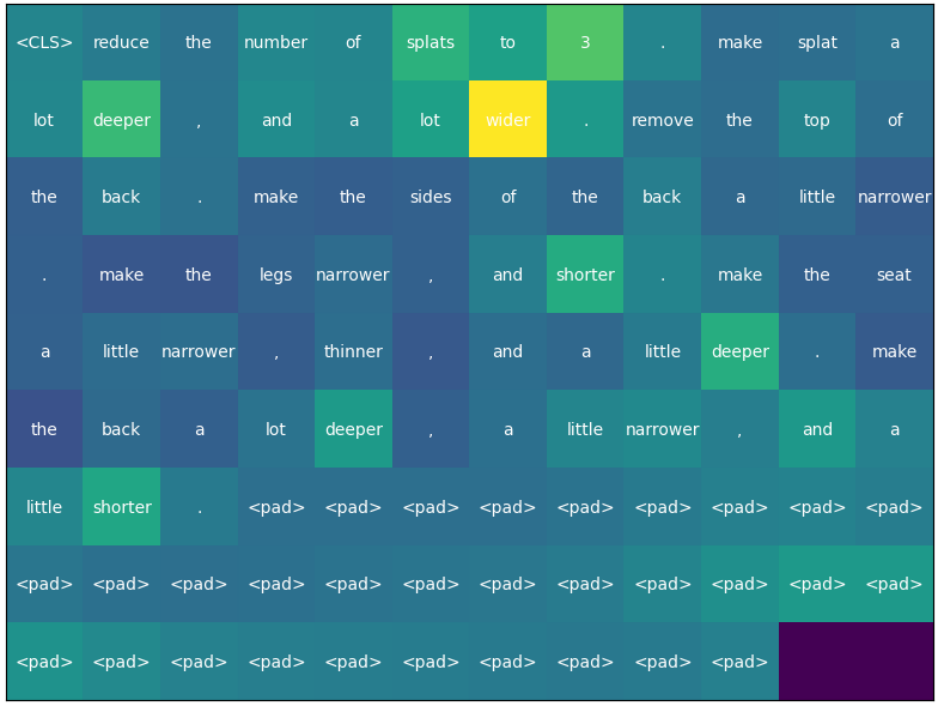
Input Source

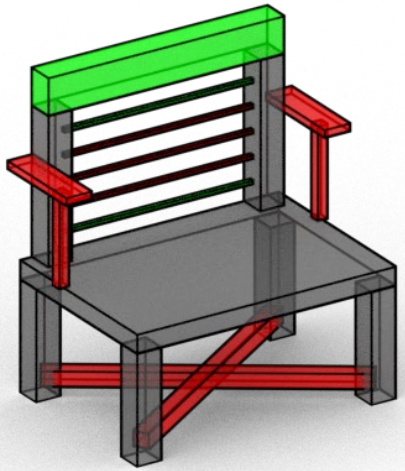


Groundtruth target



Input description (& attention)

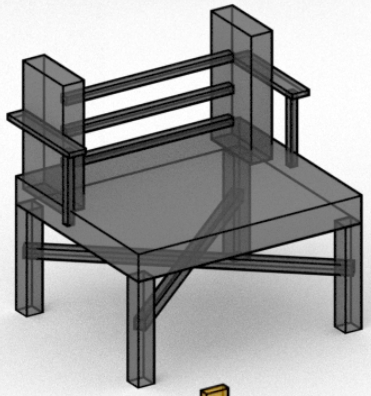




Predicted deletions

= correct

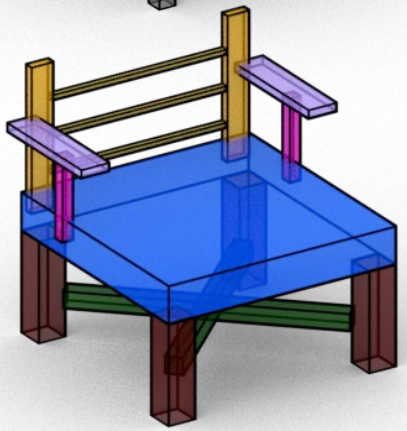
= incorrect



Predicted Insertions

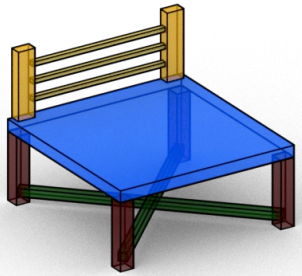
= correct

= incorrect

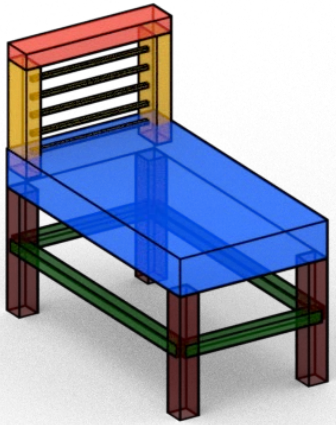


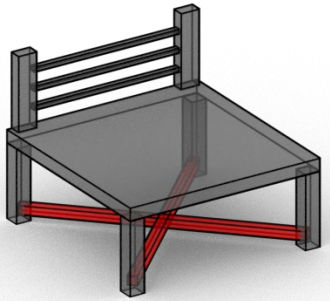
Predicted Box Params

Input Source



Groundtruth target

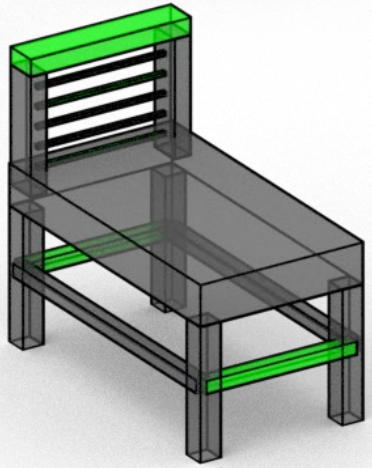




Predicted deletions

= correct

= incorrect



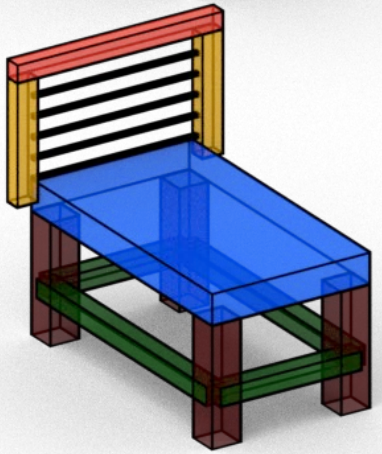
Predicted Insertions

= correct

= incorrect

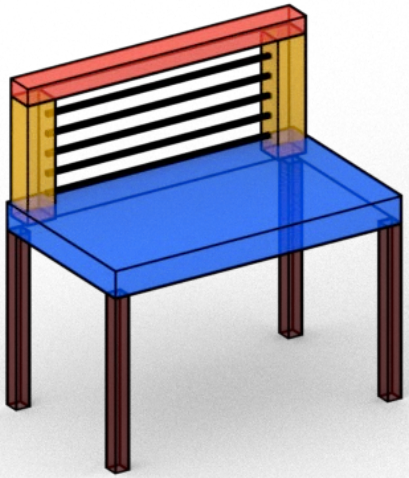
Input description (& attention)

<CLS>	give	it	2	more	splats	.	make	splat	narrower	,	and
shallower	.	add	the	top	of	the	back	.	make	the	sides
of	the	back	narrower	.	make	the	stretchers	take	on	the	o
style	.	make	the	legs	shorter	,	and	narrower	.	make	the
seat	a	little	thinner	,	a	little	shallower	,	and	narrower	.
make	the	back	narrower	,	a	lot	deeper	,	and	a	little
shorter	.	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>		

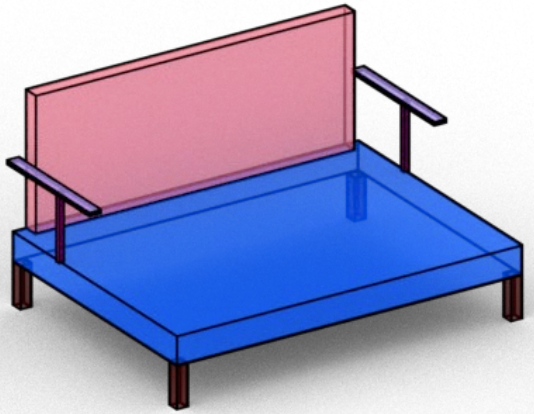


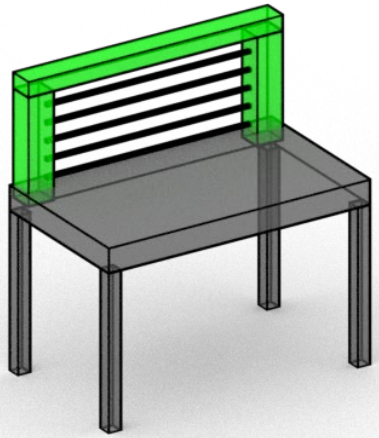
Predicted Box Params

Input Source



Groundtruth target





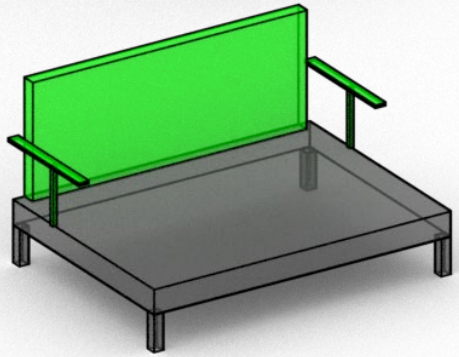
Predicted deletions

= correct

= incorrect

Input description (& attention)

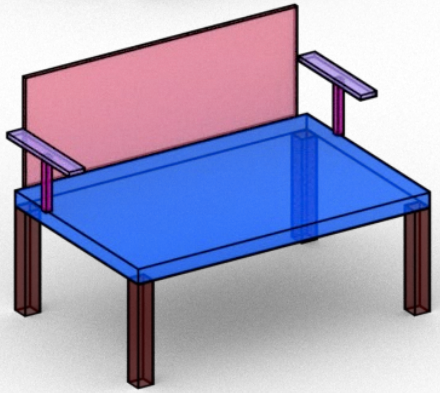
<CLS>	make	the	back	take	on	the	simple	style	.	make	the
back	a	little	narrower	,	a	little	shorter	,	and	shallower	.
add	the	armrests	.	make	the	legs	a	little	wider	,	and
shorter	.	make	the	seat	a	little	deeper	,	a	little	wider
,	and	a	lot	thicker	.	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>		



Predicted Insertions

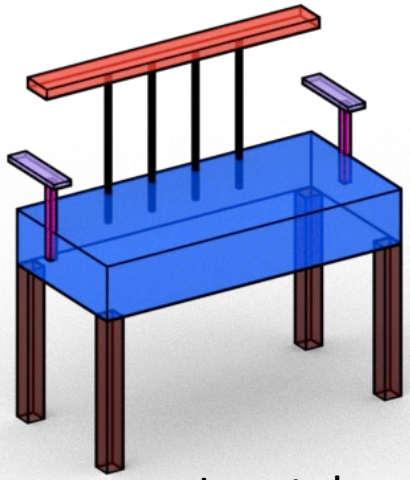
= correct

= incorrect

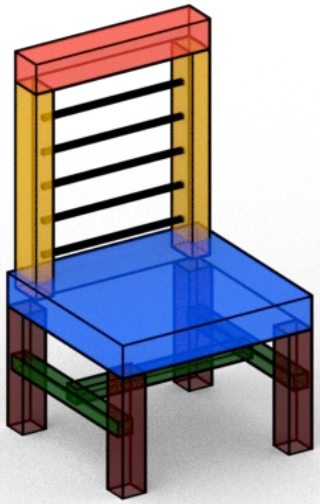


Predicted Box Params

Input Source

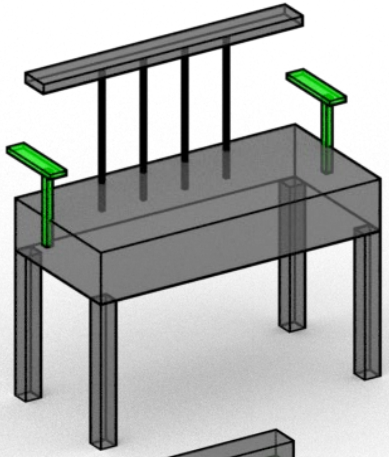


Groundtruth target



Input description (& attention)

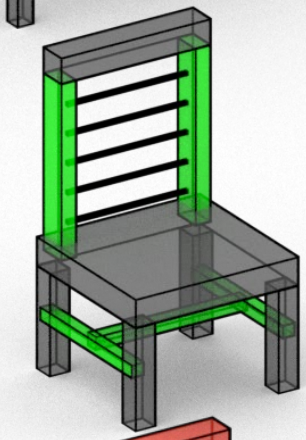
<CLS>	give	it	1	more	splats	.	make	the	back	take	on
the	h	style	.	make	the	back	a	little	deeper	,	longer
,	and	narrower	.	make	the	top	of	the	back	a	lot
longer	.	add	the	sides	of	the	back	.	add	the	stretchers
.	remove	the	armrests	.	make	the	legs	a	little	shorter	,
and	wider	.	make	the	seat	narrower	,	thinner	,	and	a
little	deeper	.	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>		



Predicted deletions

= correct

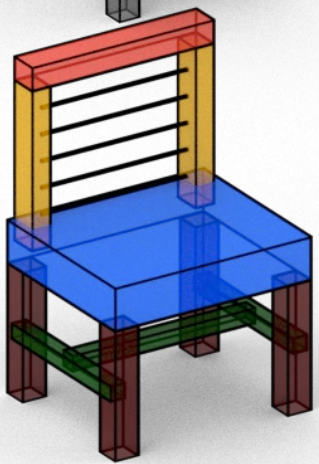
= incorrect



Predicted Insertions

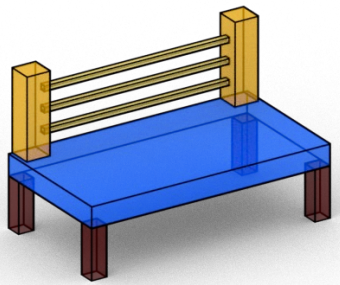
= correct

= incorrect

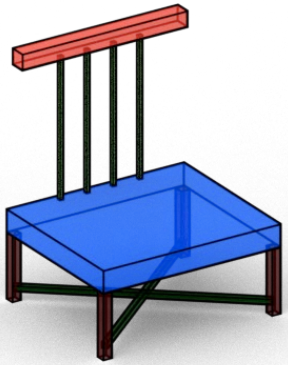


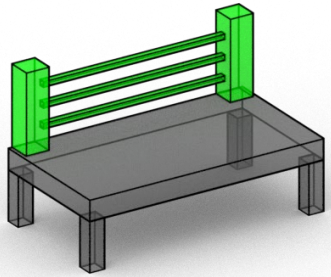
Predicted Box Params

Input Source



Groundtruth target

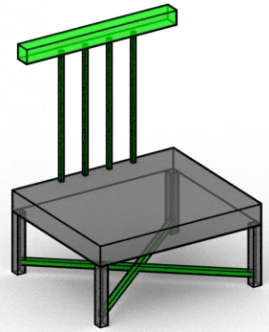




Predicted deletions

= correct

= incorrect



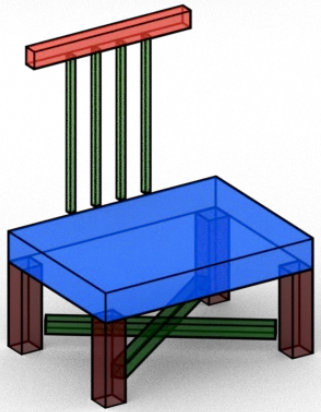
Predicted Insertions

= correct

= incorrect

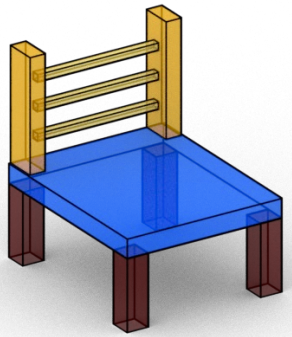
Input description (& attention)

<CLS>	give	it	1	more	splats	.	make	splat	narrower	,	and
shallower	.	make	the	back	take	on	the	t	style	.	make
the	back	a	little	longer	,	narrower	,	and	shallower	.	add
the	top	of	the	back	.	remove	the	sides	of	the	back
.	add	the	stretchers	.	make	the	legs	a	little	longer	,
and	narrower	.	make	the	seat	a	little	thinner	,	a	little
shallower	,	and	narrower	.	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>		

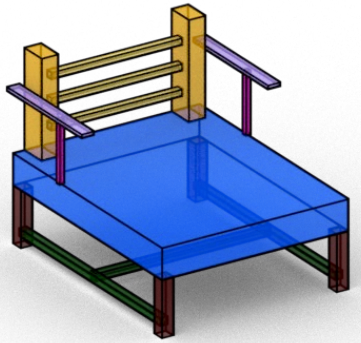


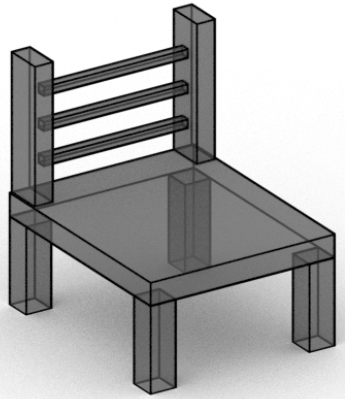
Predicted Box Params

Input Source



Groundtruth target





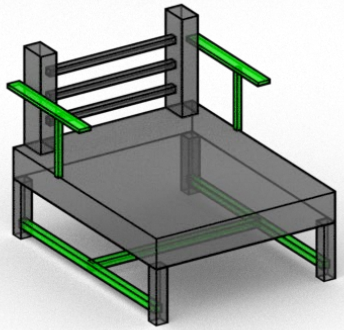
Predicted deletions

= correct

= incorrect

Input description (& attention)

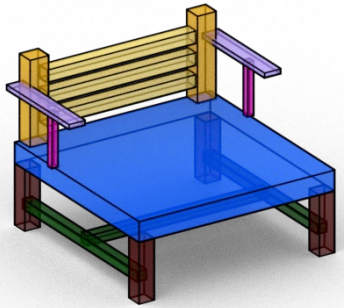
<CLS>	make	the	sides	of	the	back	a	little	wider	.	add
the	stretchers	.	add	the	armrests	.	make	the	legs	narrower	,
and	a	little	shorter	.	make	the	seat	deeper	,	a	lot
thicker	,	and	wider	.	make	the	back	a	little	shallower	,
a	little	wider	,	and	a	little	shorter	.	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>
<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>	<pad>		



Predicted Insertions

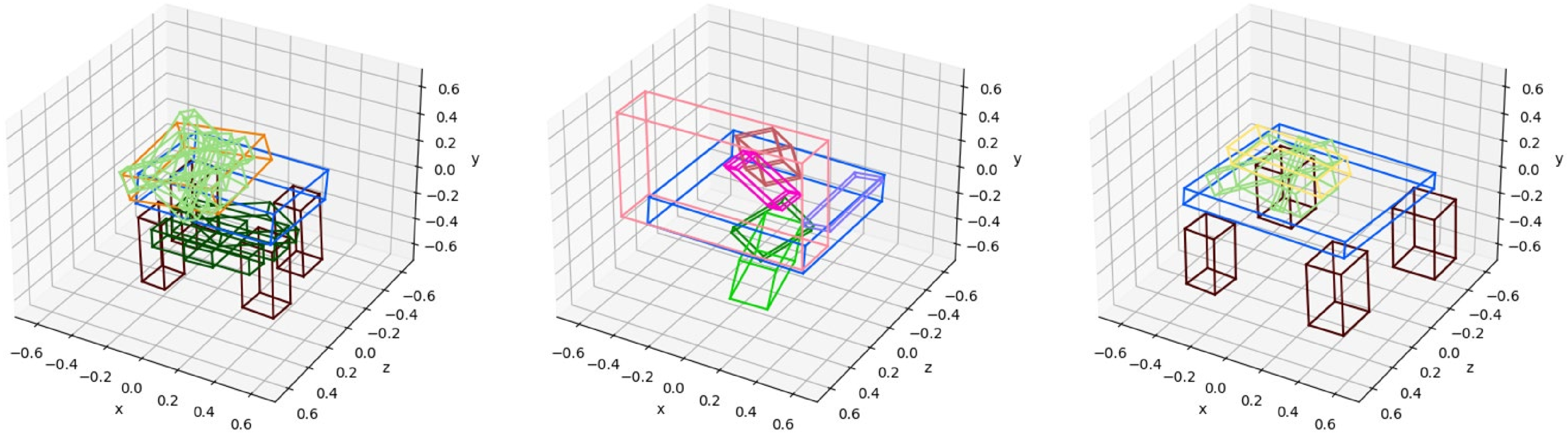
= correct

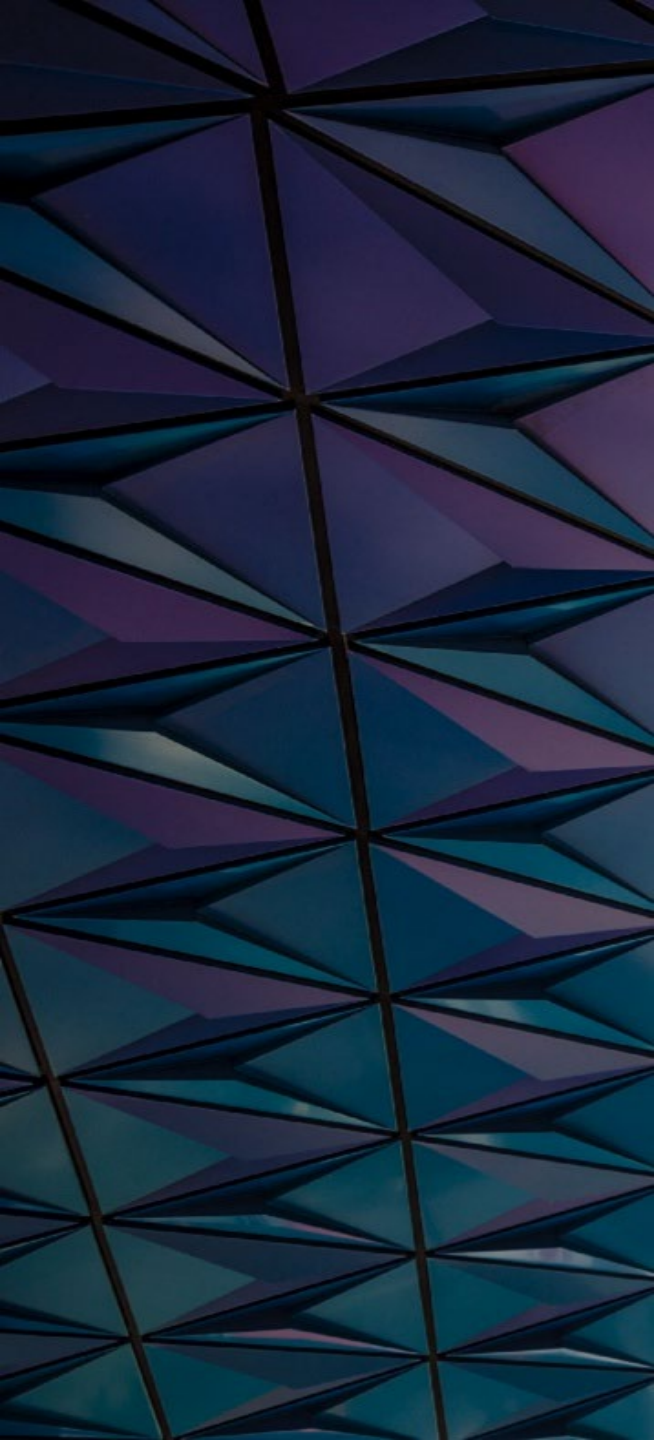
= incorrect



Predicted Box Params

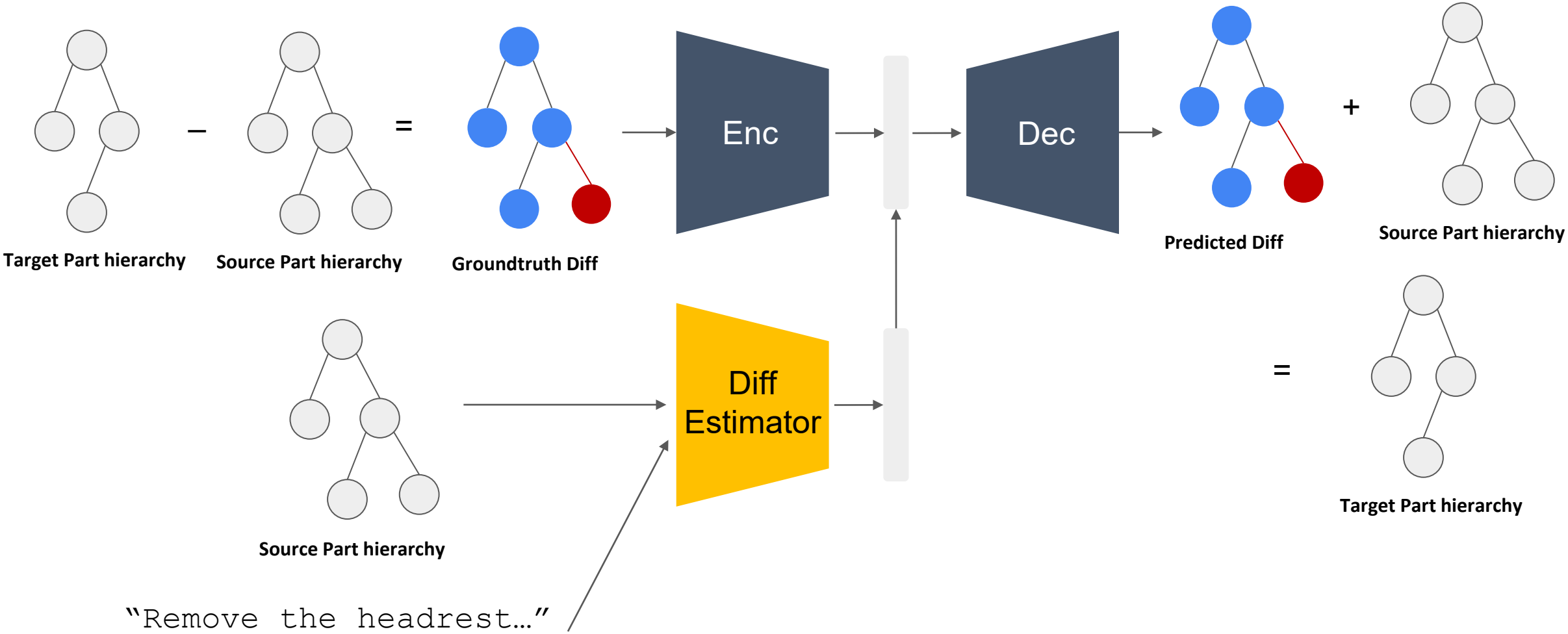
But when dealing with Partnet structures, constraints get violated a lot.

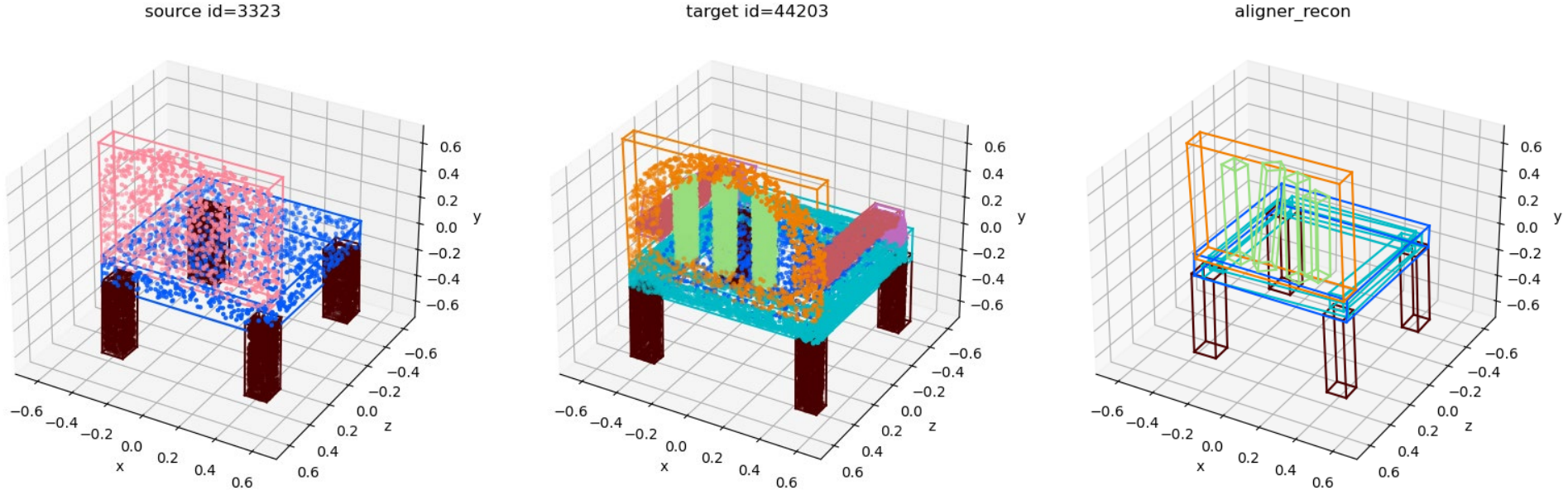




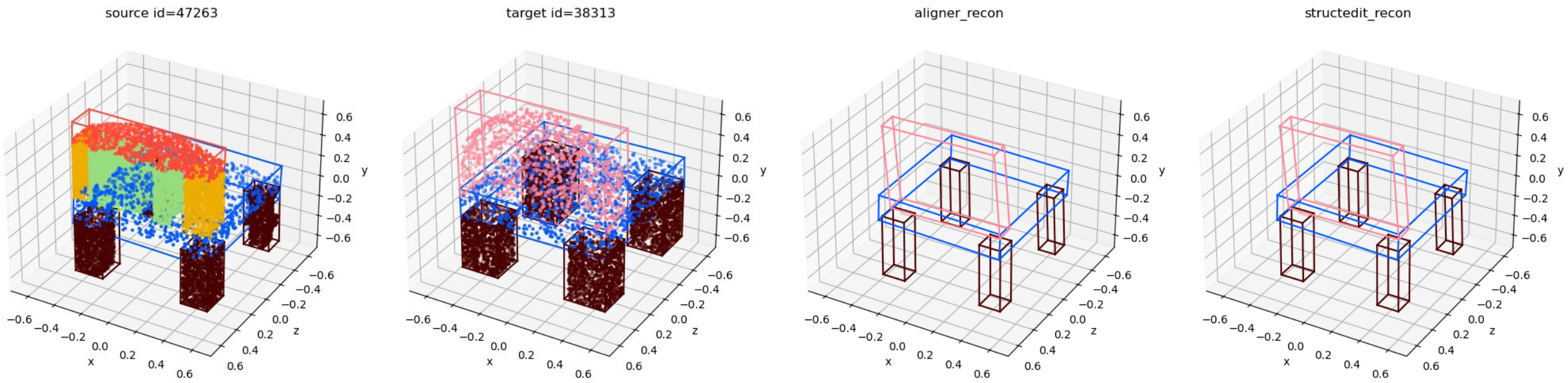
Latent space traversal

AE latent space traversal

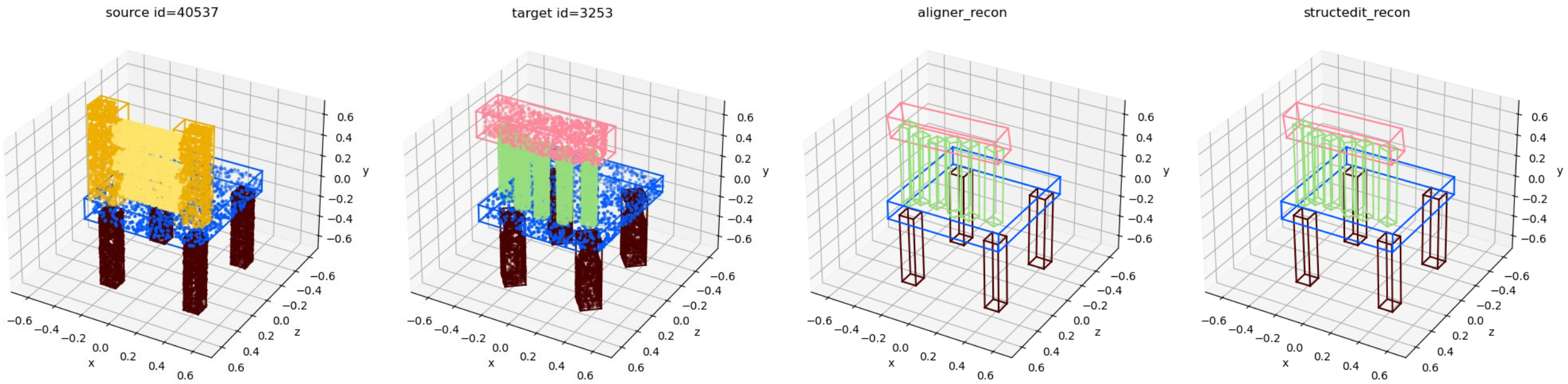




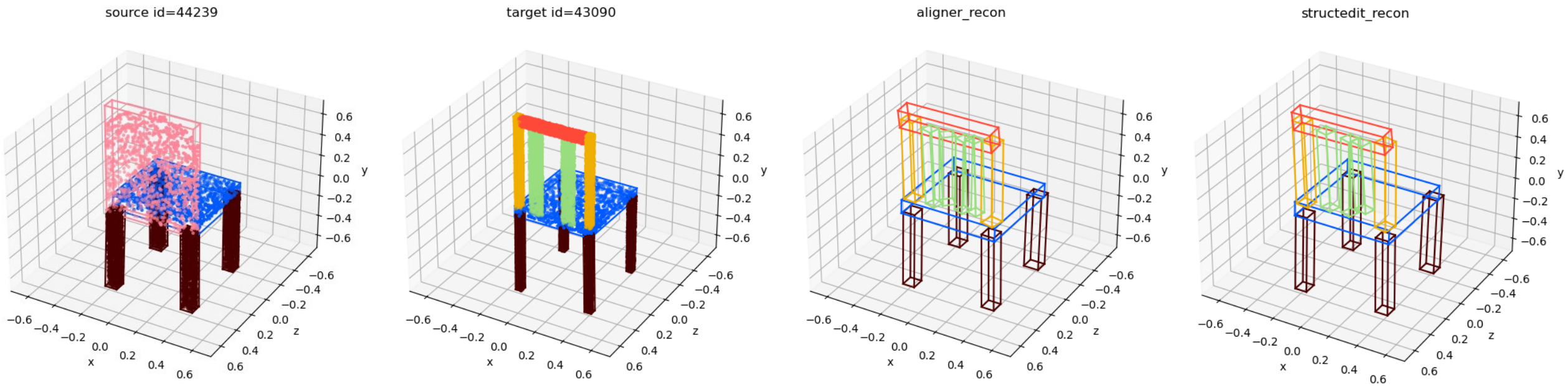
(((‘leg’, (‘left of’, ‘infront of’), ‘seat_single_surface’),), ‘make shorter’, ((‘leg’, (‘left of’, ‘infront of’), ‘seat_single_surface’),)) (((‘leg’, (‘left of’, ‘infront of’), ‘seat_single_surface’),), ‘shift forwards’, ((‘leg’, (‘left of’, ‘infront of’), ‘seat_single_surface’),)) (((‘leg’, (‘left of’, ‘behind’), ‘seat_single_surface’),), ‘move away from’, ((‘leg’, (‘left of’, ‘infront of’), ‘seat_single_surface’),)) (((‘arm_horizontal_bar’, (‘left of’,), ‘back_single_surface’), (‘arm_horizontal_bar’, (‘left of’,), ‘seat_single_surface’)), ‘DEL’, ((‘arm_horizontal_bar’, (‘left of’,), ‘back_single_surface’), (‘arm_horizontal_bar’, (‘left of’,), ‘seat_single_surface’))) (((‘leg’, (‘right of’, ‘infront of’), ‘seat_single_surface’),), ‘make wider’, ((‘leg’, (‘right of’, ‘infront of’), ‘seat_single_surface’),)) (((‘leg’, (‘right of’, ‘infront of’), ‘seat_single_surface’),), ‘move away from’, ((‘back_single_surface’, (), ‘back_single_surface’),)) (((‘leg’, (‘right of’, ‘behind’), ‘seat_single_surface’),), ‘move away from’, ((‘leg’, (‘left of’, ‘infront of’), ‘seat_single_surface’),)) (((‘seat_single_surface’, (), ‘seat_single_surface’),), ‘make taller’, ((‘seat_single_surface’, (), ‘seat_single_surface’),)) (((‘back_single_surface’, (), ‘back_single_surface’),), ‘make deeper’, ((‘back_single_surface’, (), ‘back_single_surface’),)) (((‘back_single_surface’, (), ‘back_single_surface’),), ‘make wider’, ((‘back_single_surface’, (), ‘back_single_surface’),)) (((‘seat_frame_bar’, (‘right of’,), ‘back_single_surface’), (‘seat_frame_bar’, (‘right of’,), ‘seat_single_surface’)), ‘DEL’, ((‘seat_frame_bar’, (‘right of’,), ‘back_single_surface’), (‘seat_frame_bar’, (‘right of’,), ‘seat_single_surface’))) (((‘arm_sofa_style’, (‘left of’,), ‘back_single_surface’), (‘arm_sofa_style’, (‘left of’,), ‘seat_single_surface’)), ‘ADD’, ((‘arm_sofa_style’, (‘left of’,), ‘back_single_surface’), (‘arm_sofa_style’, (‘left of’,), ‘seat_single_surface’))) (((‘back_frame_vertical_bar’, (‘right of’,), ‘back_single_surface’), (‘back_frame_vertical_bar’, (‘right of’,), ‘seat_single_surface’)), ‘DEL’, ((‘back_frame_vertical_bar’, (‘right of’,), ‘back_single_surface’), (‘back_frame_vertical_bar’, (‘right of’,), ‘seat_single_surface’)))




```
((('leg', ('left of', 'behind'), 'seat_single_surface'),), 'make wider', (('leg', ('left of', 'behind'), 'seat_single_surface'),))
((('back_frame_vertical_bar', ('right of',), 'back_frame_horizontal_bar'), ('back_frame_vertical_bar', ('right of',),
'back_single_surface'), ('back_frame_vertical_bar', ('right of',), 'seat_single_surface')), 'DEL', (('back_frame_vertical_bar', ('right
of',), 'back_frame_horizontal_bar'), ('back_frame_vertical_bar', ('right of',), 'back_single_surface'), ('back_frame_vertical_bar',
('right of',), 'seat_single_surface'))) ((('back_surface_vertical_bar', ('left of',), 'back_frame_horizontal_bar'),
('back_surface_vertical_bar', ('left of',), 'back_single_surface'), ('back_surface_vertical_bar', ('left of',), 'seat_single_surface')),
'DEL', (('back_surface_vertical_bar', ('left of',), 'back_frame_horizontal_bar'), ('back_surface_vertical_bar', ('left of',),
'back_single_surface'), ('back_surface_vertical_bar', ('left of',), 'seat_single_surface'))) ((('back_single_surface', (),
'back_single_surface'),), 'ADD', (('back_single_surface', (), 'back_single_surface'),)) ((('leg', ('right of', 'infront of'),
'seat_single_surface'),), 'shift up', (('leg', ('right of', 'infront of'), 'seat_single_surface'),)) ((('back_frame_horizontal_bar', (),
'back_frame_horizontal_bar'),), 'DEL', (('back_frame_horizontal_bar', (), 'back_frame_horizontal_bar'),))
```



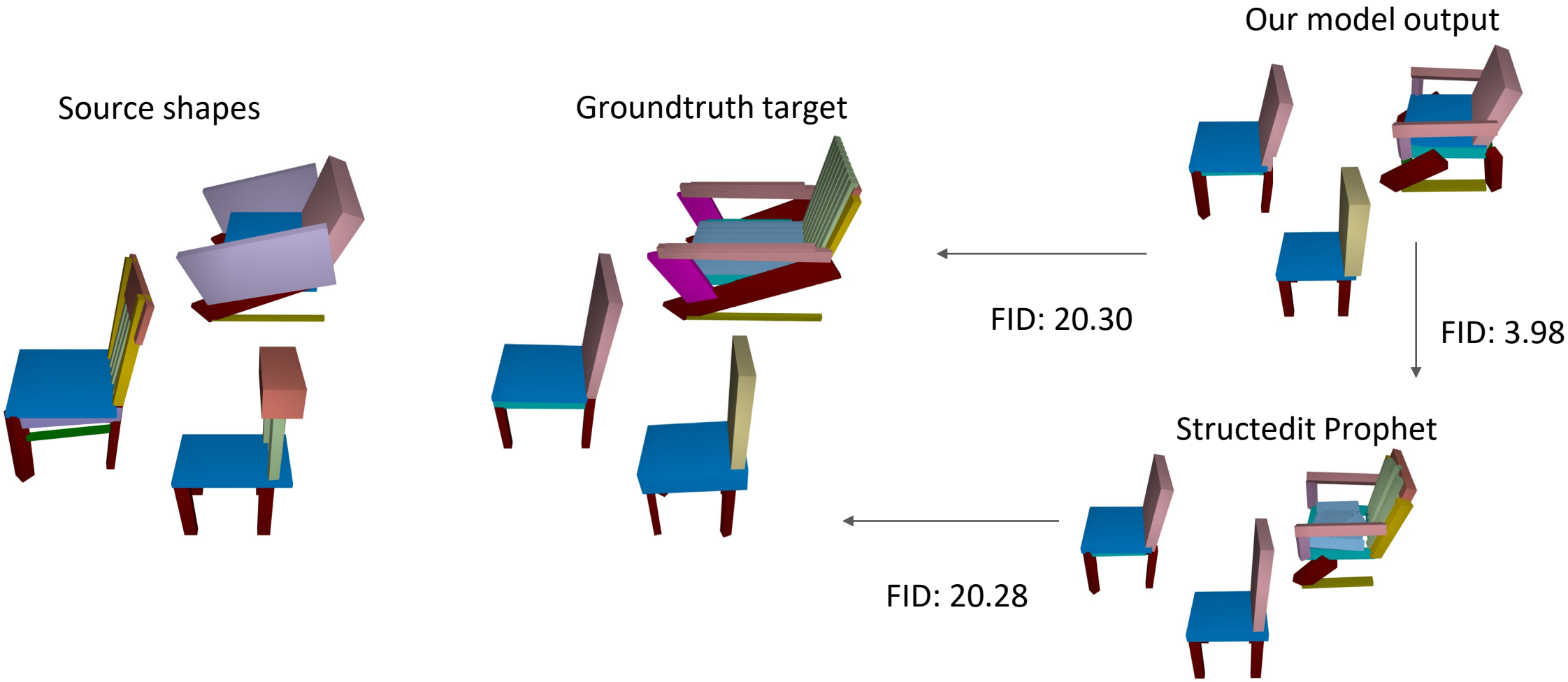

```
((('leg', ('left of', 'behind'), 'seat_single_surface'),), 'make wider', (('leg', ('left of', 'behind'), 'seat_single_surface'),))
((('back_surface_vertical_bar', ('right of', 'infront of'), 'back_single_surface'),), 'ADD', (('back_surface_vertical_bar', ('right
of', 'infront of'), 'back_single_surface'),)) ((('back_surface_vertical_bar', ('left of', 'behind'), 'back_single_surface'),), 'ADD',
(('back_surface_vertical_bar', ('left of', 'behind'), 'back_single_surface'),)) ((('back_single_surface', (),
'back_single_surface'),), 'ADD', (('back_single_surface', (), 'back_single_surface'),)) ((('back_frame_vertical_bar', ('right of',),
'back_single_surface'), ('back_frame_vertical_bar', ('right of',), 'seat_single_surface')), 'DEL', (('back_frame_vertical_bar',
('right of',), 'back_single_surface'), ('back_frame_vertical_bar', ('right of',), 'seat_single_surface'))) ((('seat_single_surface',
(), 'seat_single_surface'),), 'make shorter', (('seat_single_surface', (), 'seat_single_surface'),))
```




```
((('leg', ('left of', 'behind'), 'back_single_surface'), ('leg', ('left of', 'behind'), 'seat_single_surface')), 'make narrower', (('leg', ('left
of', 'behind'), 'back_single_surface'), ('leg', ('left of', 'behind'), 'seat_single_surface'))) ((('leg', ('behind',),
'back_frame_horizontal_bar'),), 'make shallower', (('leg', ('behind',), 'back_frame_horizontal_bar'),)) ((('leg', ('behind',),
'back_frame_horizontal_bar'),), 'make narrower', (('leg', ('behind',), 'back_frame_horizontal_bar'),)) ((('back_surface_vertical_bar', ('right
of',), 'back_frame_horizontal_bar'), ('back_surface_vertical_bar', ('right of',), 'back_single_surface'), ('back_surface_vertical_bar', ('right
of',), 'seat_single_surface')), 'ADD', (('back_surface_vertical_bar', ('right of',), 'back_frame_horizontal_bar'), ('back_surface_vertical_bar',
('right of',), 'back_single_surface'), ('back_surface_vertical_bar', ('right of',), 'seat_single_surface'))) ((('back_frame_vertical_bar', ('right
of',), 'back_frame_horizontal_bar'), ('back_frame_vertical_bar', ('right of',), 'back_single_surface'), ('back_frame_vertical_bar', ('infront
of',), 'back_single_surface'), ('back_frame_vertical_bar', ('right of',), 'seat_single_surface')), 'ADD', (('back_frame_vertical_bar', ('right
of',), 'back_frame_horizontal_bar'), ('back_frame_vertical_bar', ('right of',), 'back_single_surface'), ('back_frame_vertical_bar', ('infront
of',), 'back_single_surface'), ('back_frame_vertical_bar', ('right of',), 'seat_single_surface'))) ((('back_single_surface', (),
'back_single_surface'),), 'DEL', (('back_single_surface', (), 'back_single_surface'),)) ((('back_frame_horizontal_bar', (),
'back_frame_horizontal_bar'),), 'ADD', (('back_frame_horizontal_bar', (), 'back_frame_horizontal_bar'),)) ((('seat_single_surface', (),
'seat_single_surface'),), 'make shorter', (('seat_single_surface', (), 'seat_single_surface'),))
```

Findings so far

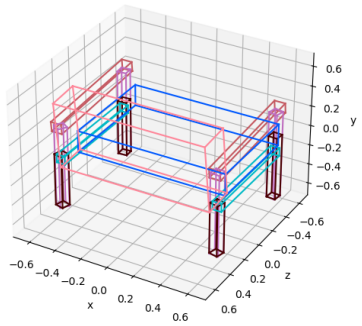
VAE traversal works better on real-shapes.



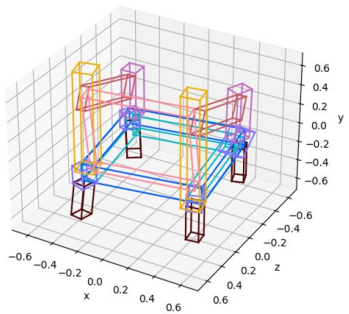
But trying the same technique on CiC fails…

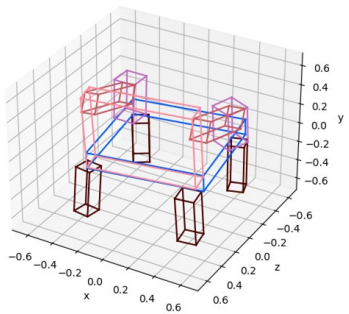
On the shapeglot dataset, there is too many geometric differences between distractors and the target that are not described by **innate constraints** or the **human-provided language.** VAE traversal “fails”.

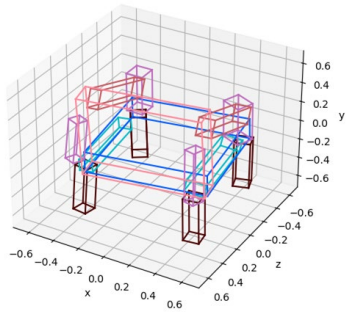
Groundtruth target

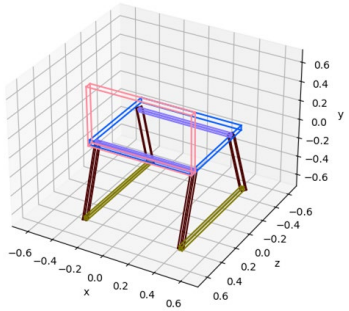


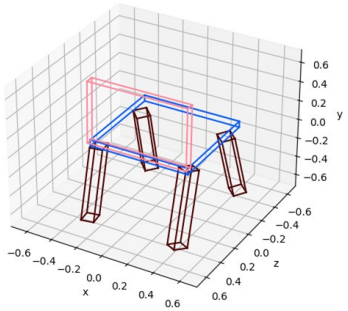
`this bars on the seat 5 horizontal backings`

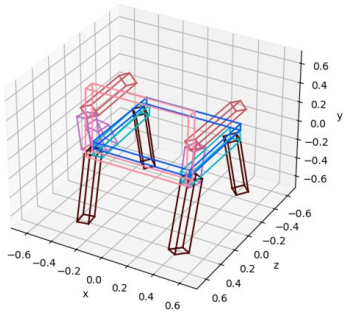












Source shape

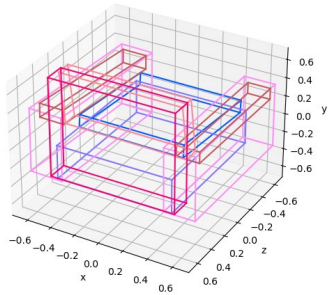
Our model output

Structedit Prophet

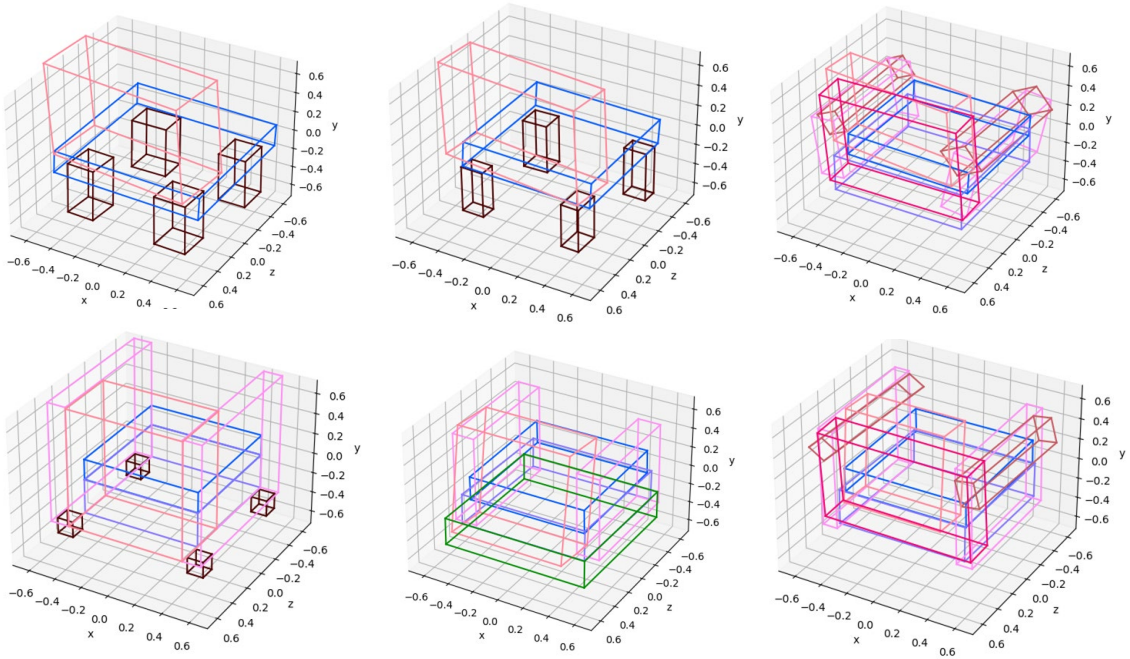
But trying the same technique on CiC fails…

On the shapeglot dataset, there is too many geometric differences between distractors and the target that are not described by **innate constraints** or the **human-provided language.** VAE traversal “fails”.

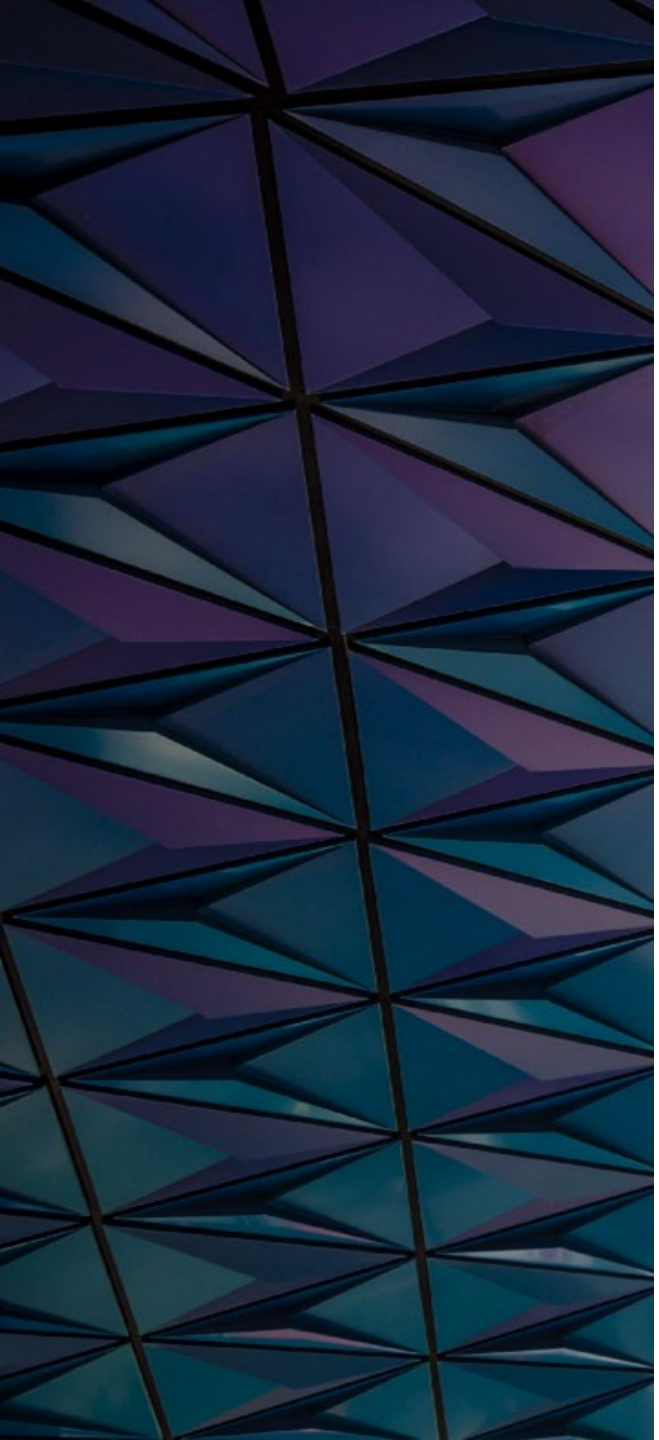
Groundtruth target



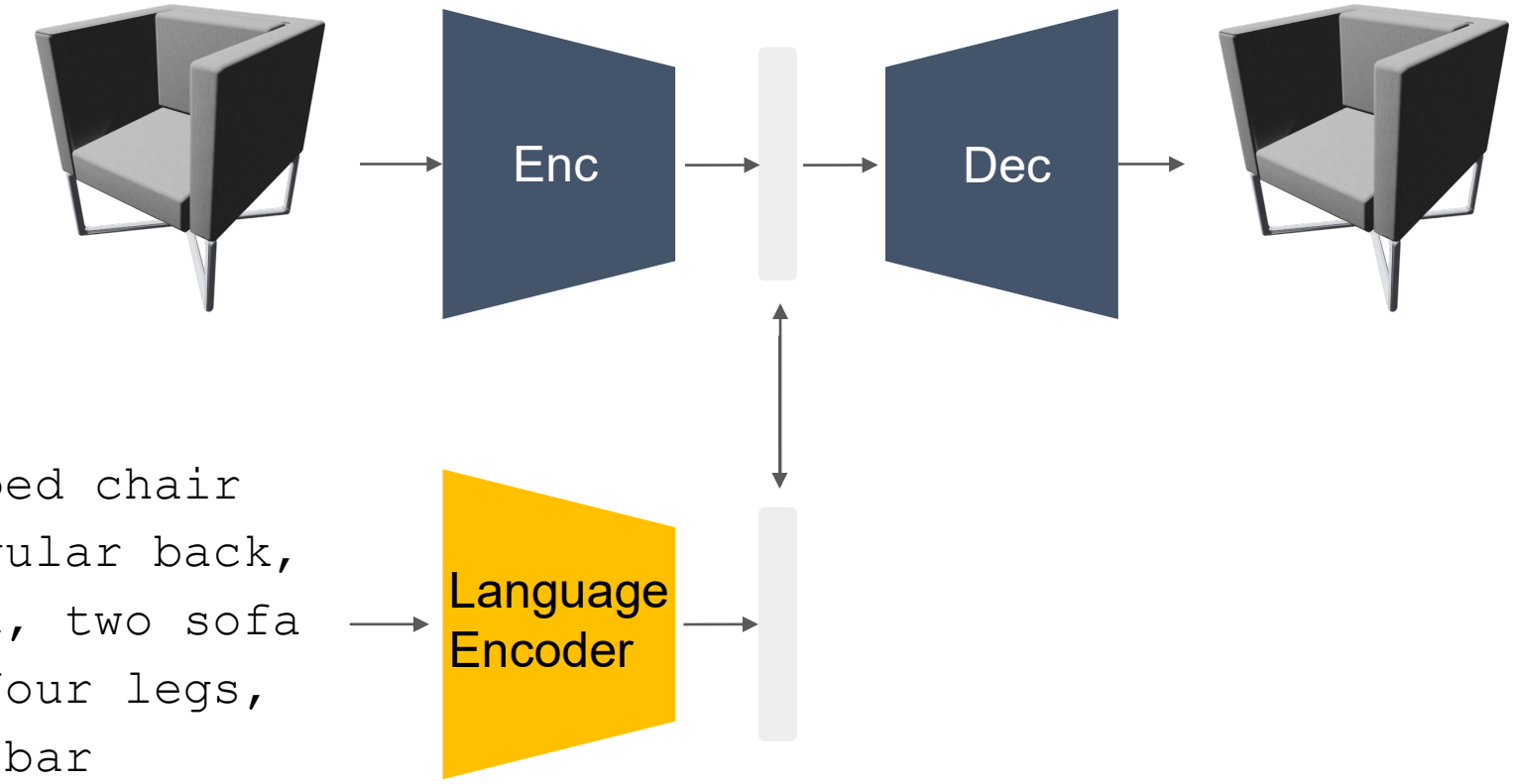
```
fat chair on ground
short back and little
legs very boxy
```



Source shape | Our model output | Structedit Prophet



Contrastive approach for a joint space



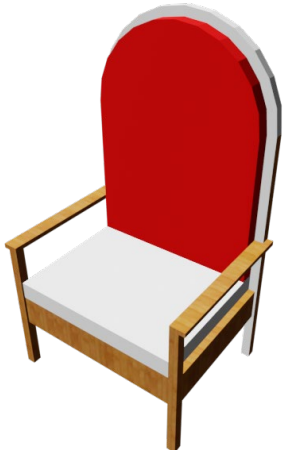
the cube shaped chair has a rectangular back, a square seat, two sofa style arms, four legs, and four leg bar supports.

chair with curved back seat and seat with four thick legs.

this chair has a regular back, a seat, as well as four straight legs.

a classic vertical bar back topped by a horizontal bar for stability and a crisp look, a gently molded seat for comfort and support, all four legs are reinforced for with leg bars for added strength and longevity.

this chair has a back, a seat, four legs, and three leg bars.



a chair consists of five parts: a back; a seat; two vertical arm bars; two horizontal arm bars; and four legs.

single chair including parts such as a chair back, seat, arm near vertical bar, arm horizontal bar and legs.

chair with back seat and seat , two thin short legs, two horizontal and two vertical leg bars.

this rocking chair has two vertical legs, and a back rest held on with two horizontal bars.

the chair shown has a back, a seat, and four legs.

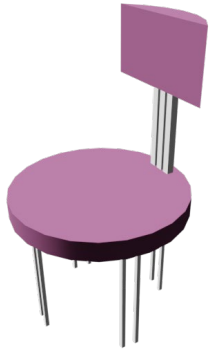
the parts of the chair are as follows: the chair back, the chair seat, the left and right arms, the central support, and four legs.

chair with back seat and seat , two thin short legs, two horizontal and two vertical leg bars.

Original

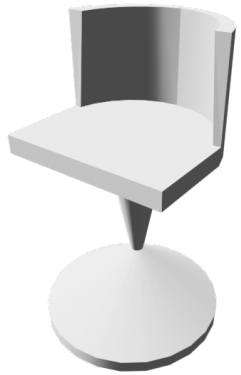
Retrieved

this chair consists of a chair back, a chair seat, and eight legs.



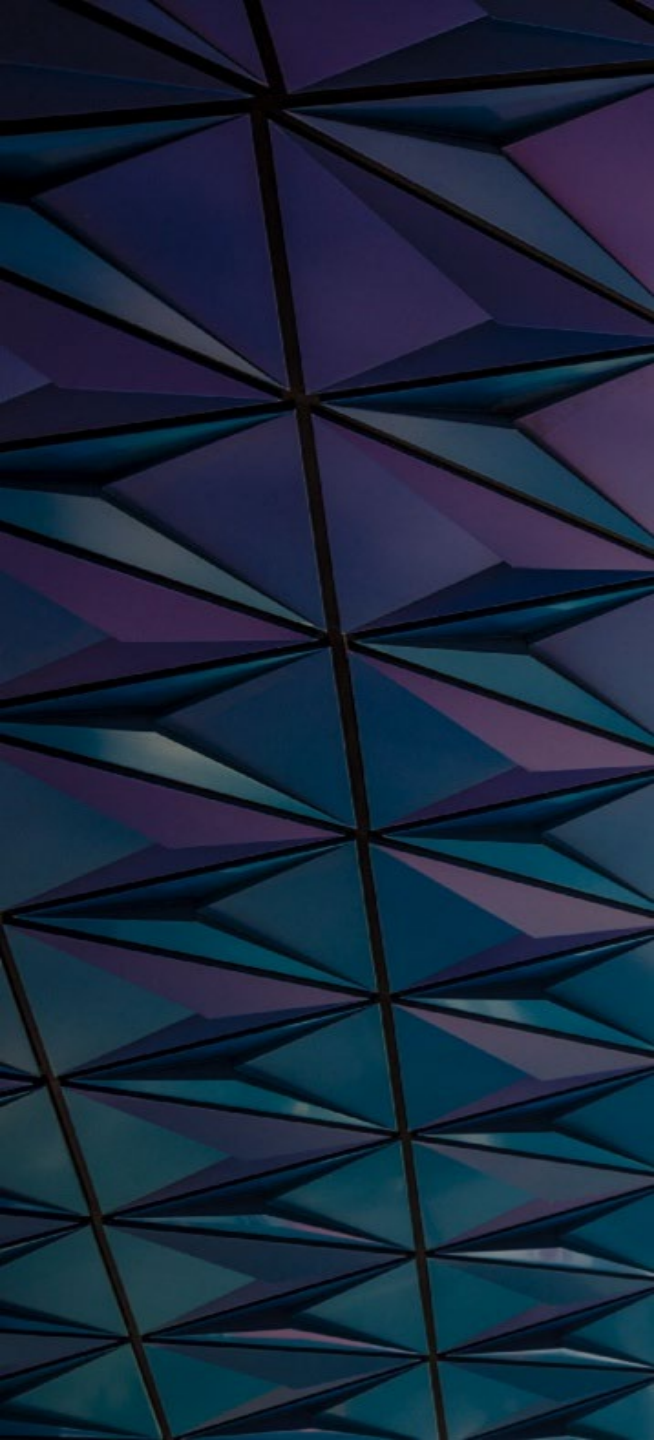
this chair has 2 vertical back frame bars, 2 horizontal back frame bars, a square seat, 4 long legs and 2 leg bars.

the parts of the chair are as follows: the chair back, the chair seat, the central support, and the pedastal base.



Original

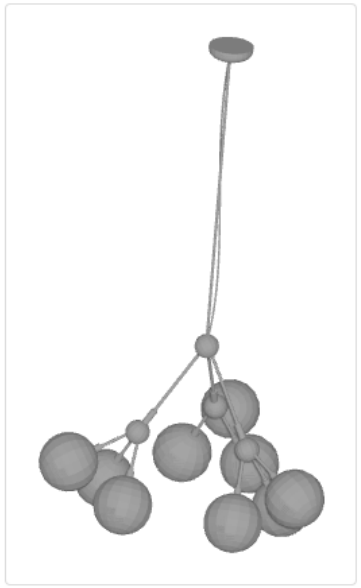
Retrieved



Language-Assisted 3D Shape Edits and Deformations

Panos Achlioptas, Ian Huang, Minhyuk Sung, Sergey Tulyakov, Leonidas Guibas

Distractor



[1] the wall attachment is thicker

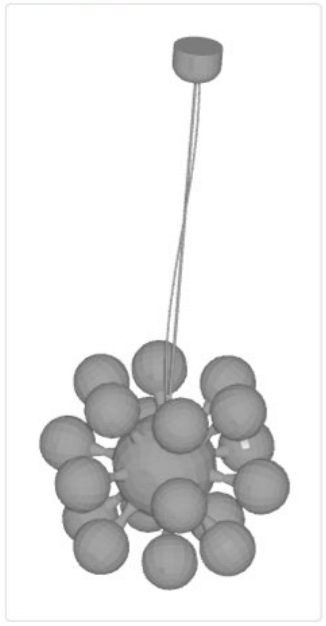
[2] there is a center ball

[3] there are more balls

[4] the shape is less abstract

First Annotator

Target



[1] The target has more spheres.

[2] It has one larger sphere with smaller spheres around it

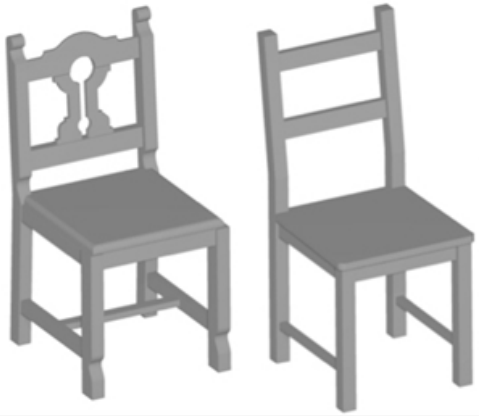
[3] The support it hangs from is thicker.

[4] Its spheres are not grouped into three clusters.

Second Annotator

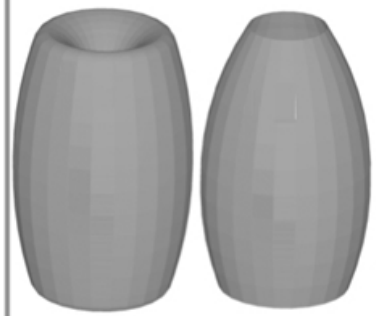
	Class	Unique Submissions	Collected Utterances
0	airplane	4560	18223
1	bag	834	3327
2	bathtub	3097	12183
3	bed	2991	14900
4	bench	6669	25758
5	bookshelf	3263	16114
6	bottle	1968	5897
7	cabinet	984	3860
8	cap	1263	3775
9	chair	15215	74767
10	clock	2320	9231
11	display	4909	14703
12	dresser	6740	33567
13	faucet	2560	10192
14	flowerpot	2492	9867
15	guitar	3013	12024
16	knife	1696	6783
17	lamp	13726	54269
18	mug	832	2488
19	person	564	2820
20	pistol	1208	4832
21	scissors	480	1438
22	skateboard	912	2724
23	sofa	13041	51996
24	table	21359	89182
25	trashbin	1372	5346
26	vase	3290	13074

Distractor **Target**



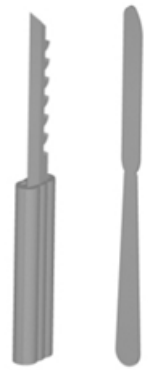
The target is thinner
There is no design on the back
The legs are not niched
The bottom does not have a cross bar
The seat is thinner

Distractor **Target**



The lip of the target is thinner
The neck of the target is narrower than the distractor's
The body of the target is slightly taller
The mouth of the target has a smaller opening

Distractor **Target**

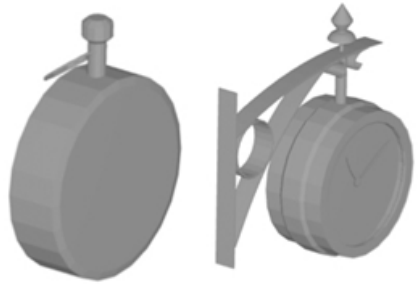


Its handle is thinner
Its blade is not serrated
It looks like a butter knife
Its blade is not pointed

Distractor **Target**

It has a top arm
The lampshade is tapered
The body is not as ornate
The base is thicker

Distractor **Target**



It has an arm for wall mounting
It is not a pocket watch
It has hour and minute hands
It is deeper

Distractor **Target**

The arm swivels
The aspect ratio is 4:3
The stand is a square

Distractor

Target

Part words

Geometric words

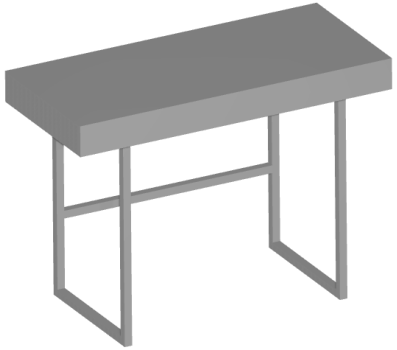
The backrest is comprised of two flat rectangular panels, separated by a thin space for a modern design.

Dimensional words

Local feature words

Stylistic words

Distractor



Target

the table is symmetrical about the long axis

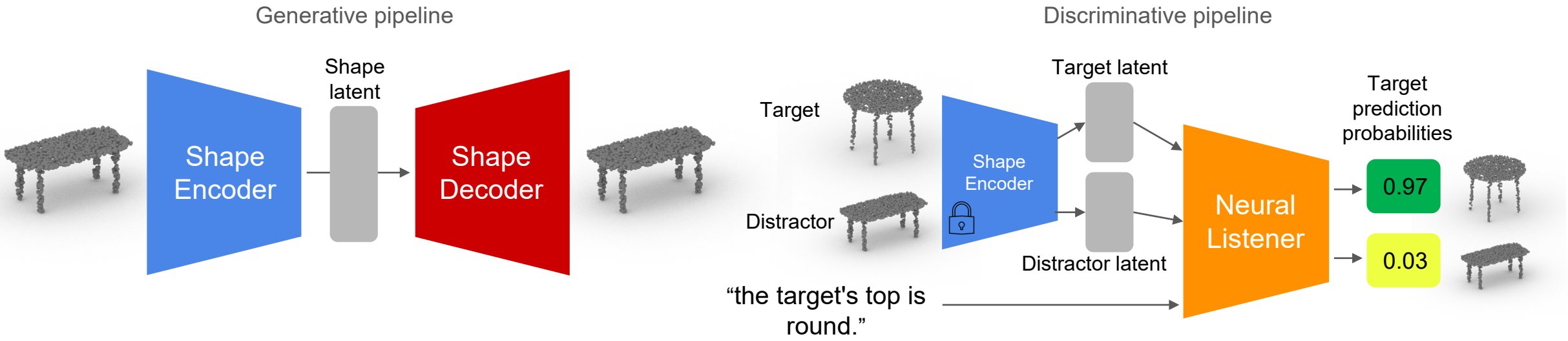
Looking at how language is used…

	stylistic	dimensional	geometric
parts	0.040	0.270	0.204
local features	0.004	0.009	0.016
holistic	0.017	0.083	0.039

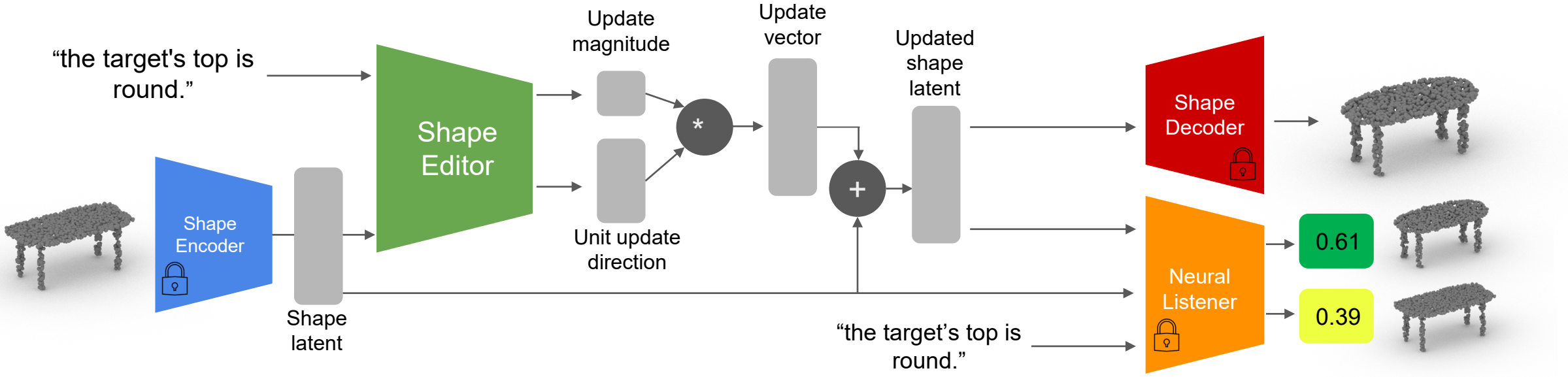
	all	all easy	all hard
parts	0.59	0.55	0.63
local features	0.53	0.50	0.55
holistic	0.72	0.69	0.75

class	part	local	holistic	stylistic	dimension	geometric
all	0.81	0.05	0.18	0.06	0.36	0.25
all easy	0.78	0.04	0.20	0.07	0.30	0.25
all hard	0.82	0.06	0.16	0.05	0.40	0.24
airplane	0.87	0.02	0.13	0.02	0.31	0.33
chair	0.88	0.05	0.11	0.08	0.33	0.27
lamp	0.83	0.04	0.17	0.06	0.38	0.30
table	0.76	0.06	0.23	0.06	0.41	0.24

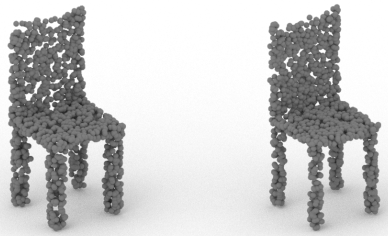
1st Stage: Generation & Discrimination



2nd Stage: Edit Prediction & Decoding

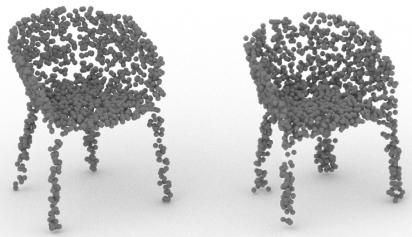


Input Output



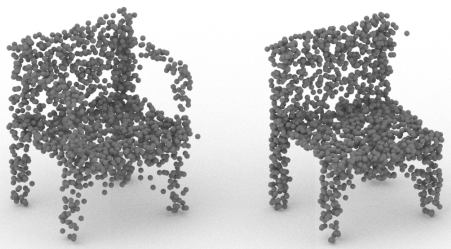
its legs are much thinner.

Input Output



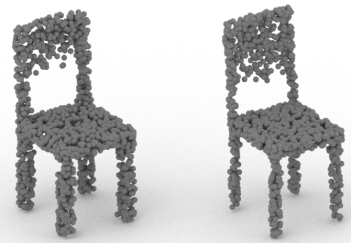
it appears more sturdy.

Input Output

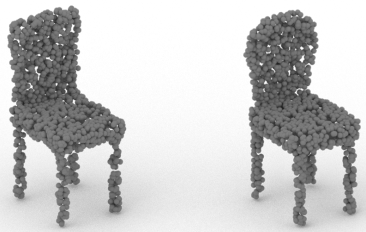


no arms.

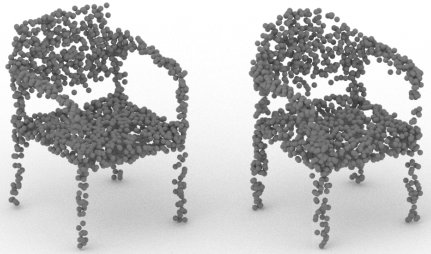
Input Output



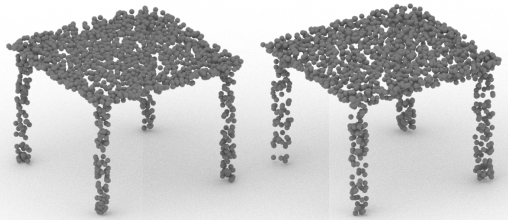
thin legs half the backrest is solid.



the backrest of the chair is curved.

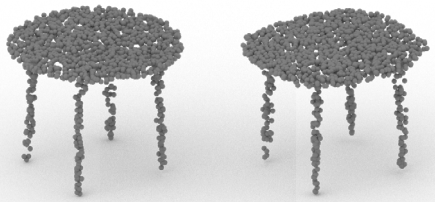


circle in the back.

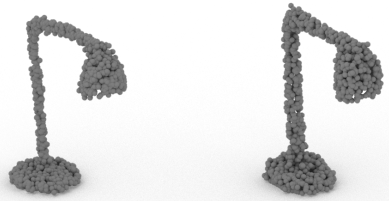


legs are thicker.

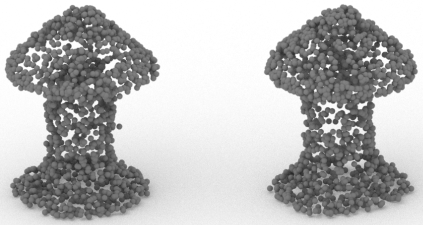
the target 's top is a semi circle.



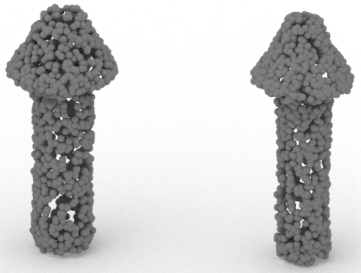
it has a rectangular top.



the frame is much bigger.

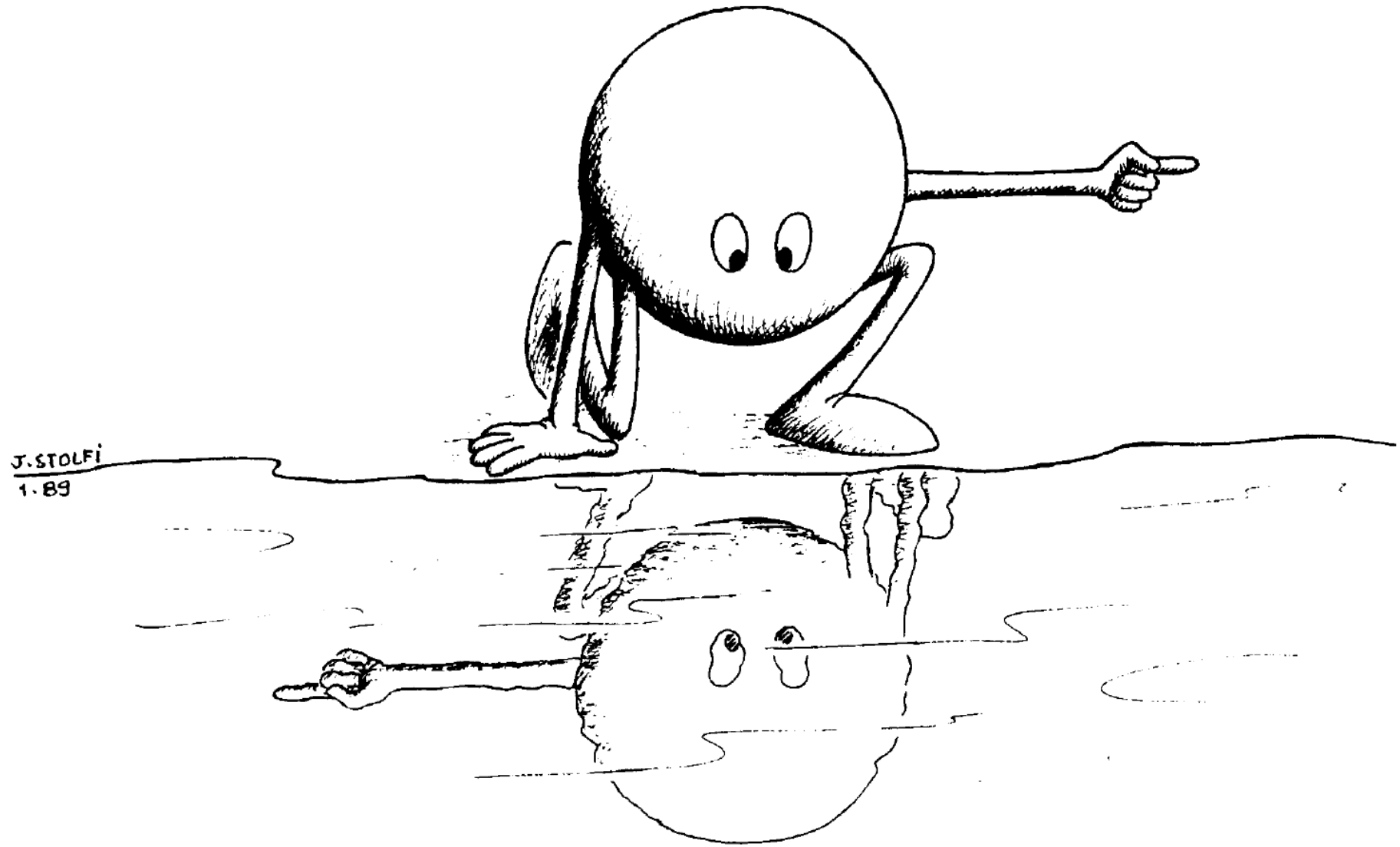


the base is less round.



the pole is thinner.

We have a long way to go!


J. STOLFI
1.89