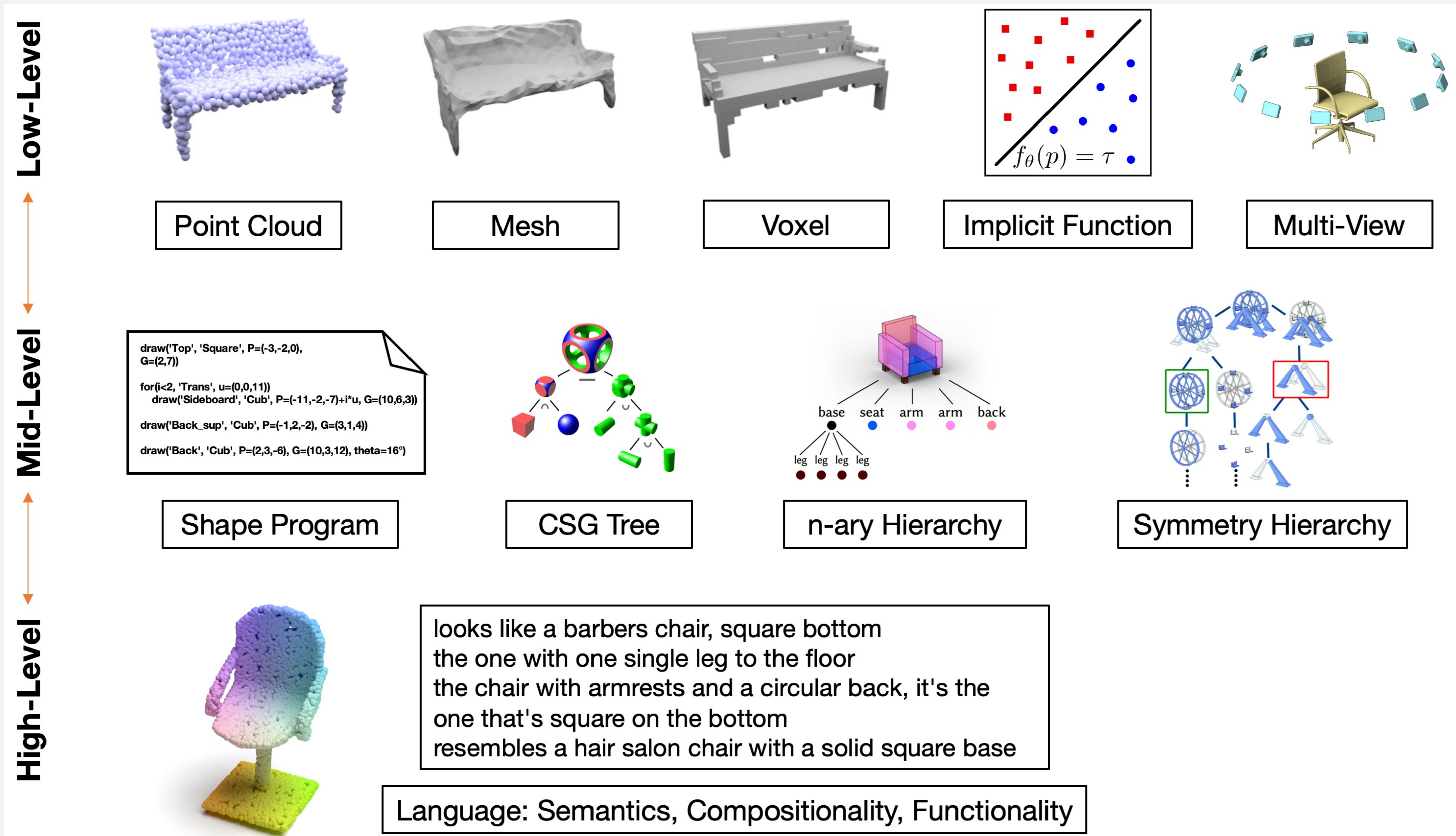




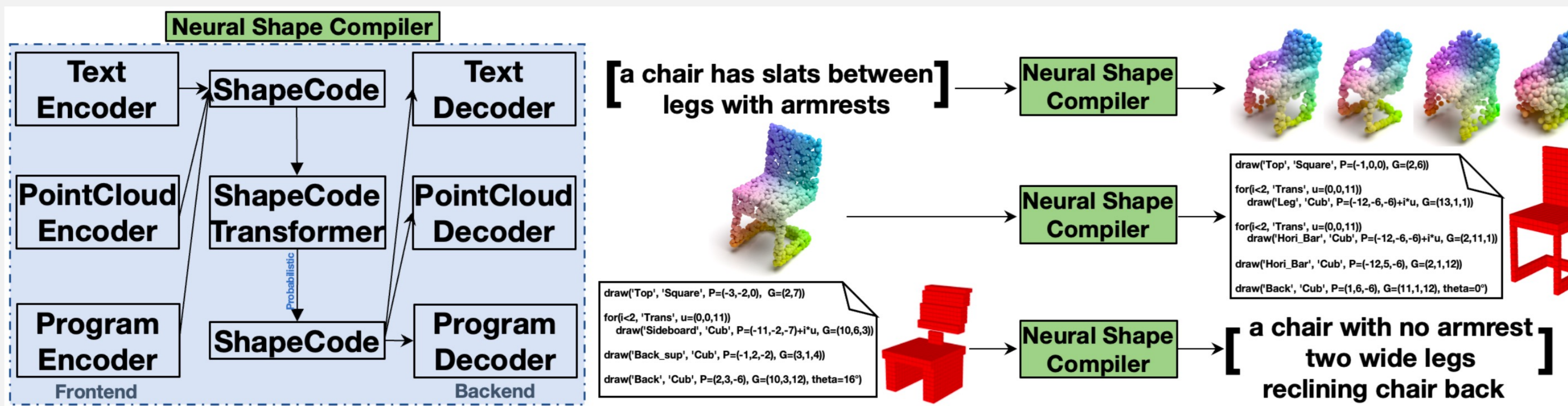
Neural Shape Compiler: A Unified Framework for Transforming between Text, Point Cloud, and Program

Tiange Luo, Honglak Lee & Justin Johnson
University of Michigan

Goal & Methods



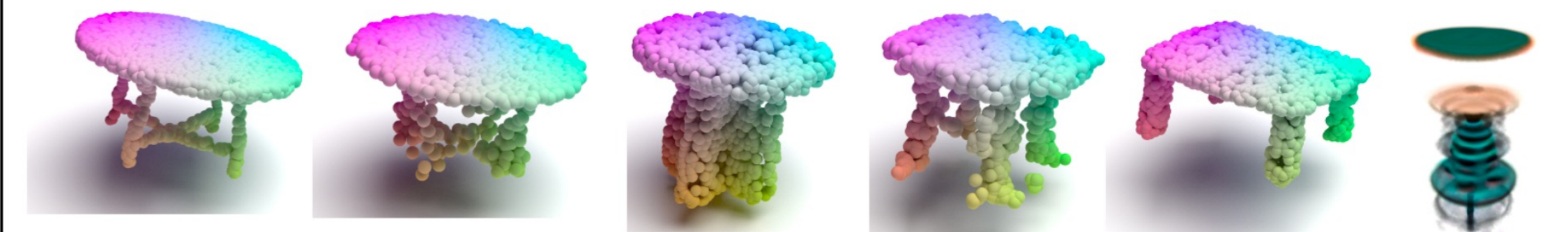
Different shape abstractions encode complementary information, and we proposed a framework to connect them.



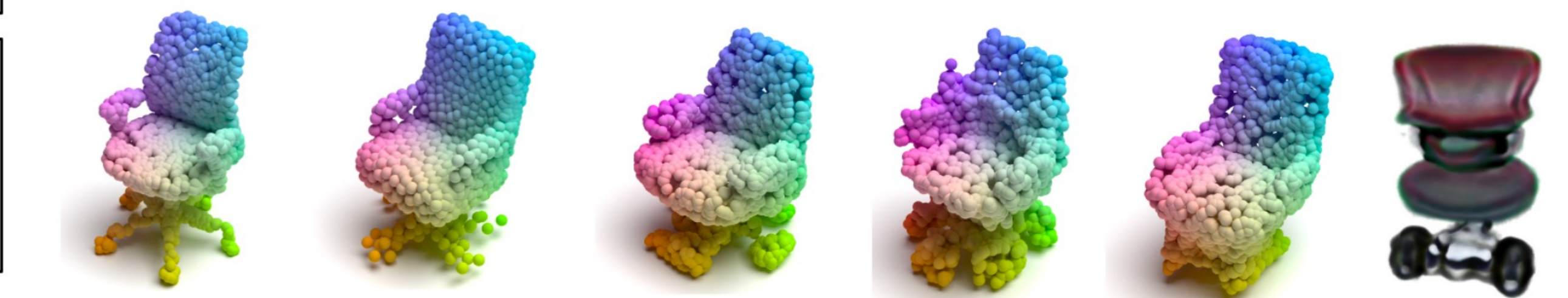
Results

Text2PointCloud

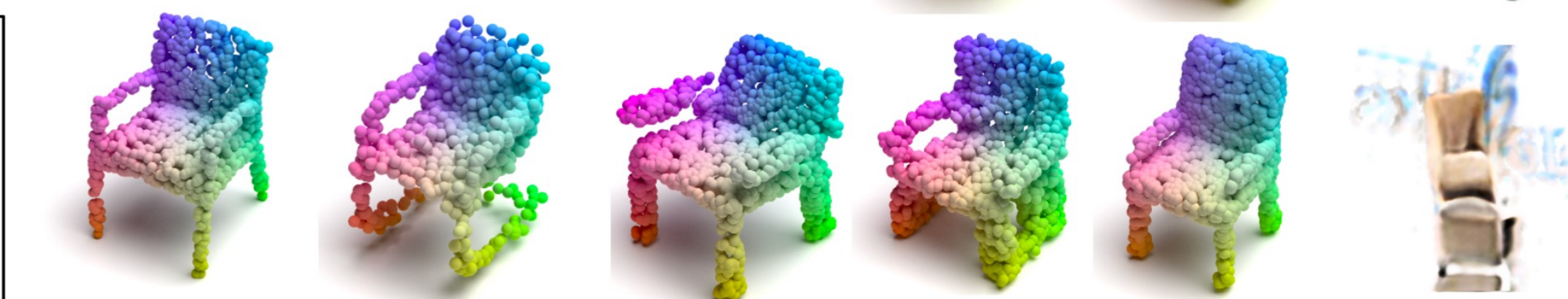
a table with circular top, four legs connected with each other by slats crossing as X



swivel chair, five wheels, office chair with armrest open frame

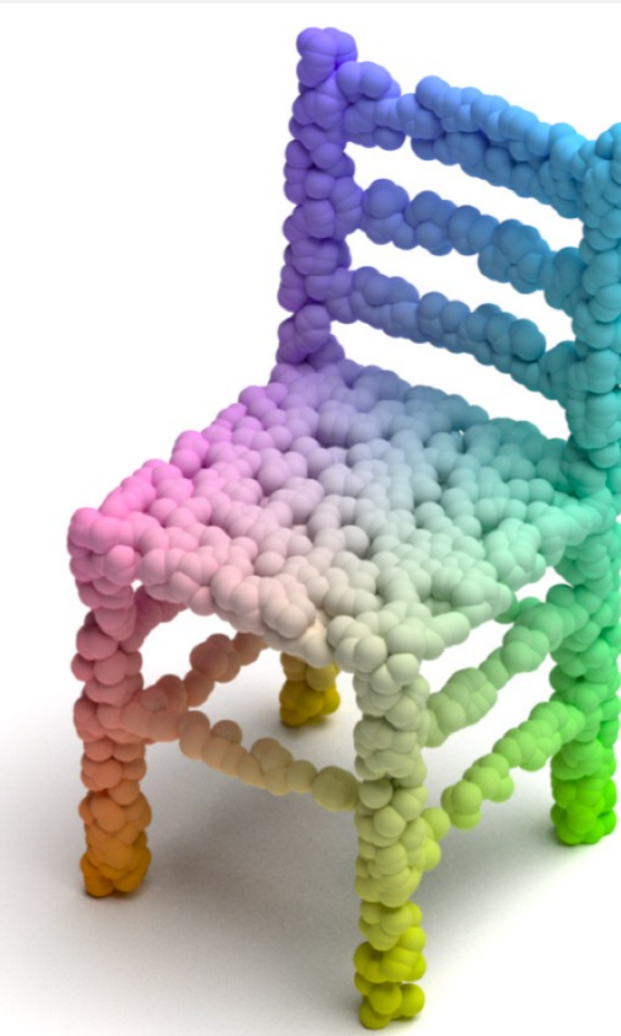


open modern style chair, armrests open frame, solid back is slightly curved



Reference Shape Compiler Shape IMLE CWGAN CLIP Forge Dream Field

Captioning

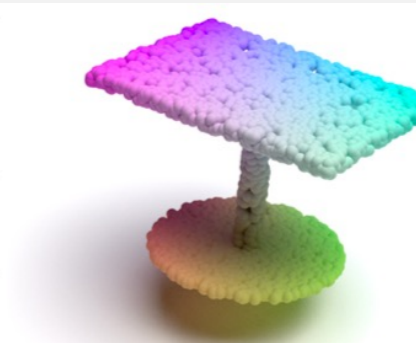


the one with 4 legs;
the one with the back and the back is the back;

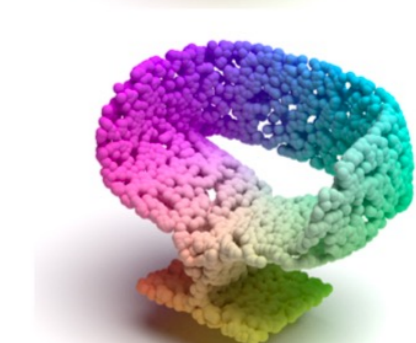
3DEncoder-LSTM

the chair with no arms;
three open spaces in the back;
it looks like it is a seat with two slats in the back;
2 horizontal bars in the back;
has extra bar connecting wire legs;
has a slatted back;
two horizontal slats, no arms, it has 4 legs;
open back chair has slits not an angle;
square with design on back;
the chair isnt rounded;

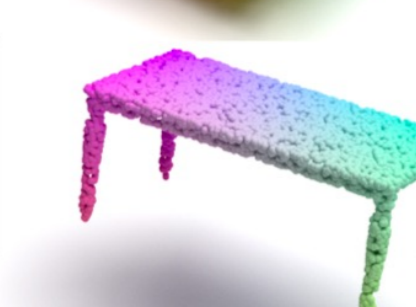
Neural Shape Compiler



circular table. it has a unique look. it has a circular stand;
a table which has a circular top and the base with the legs attached to a circular base;



no legs, sits lowest to ground, a pyramid around back;
one piece legs and weird shape;

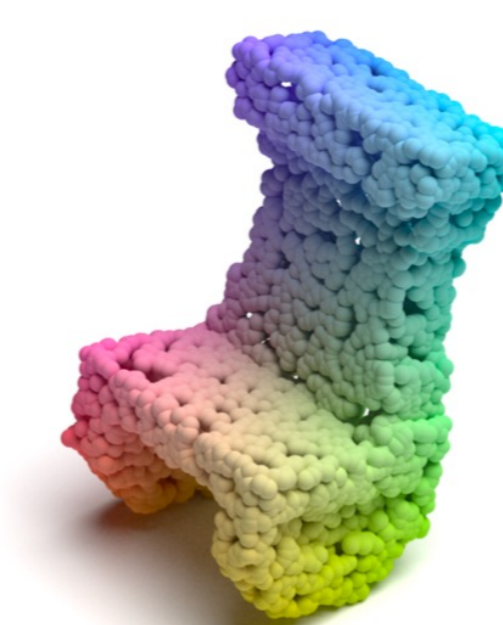


a rectangular table with laminated table top and four legs;
this is a four plain table with spindle legs;

Neural Shape Compiler

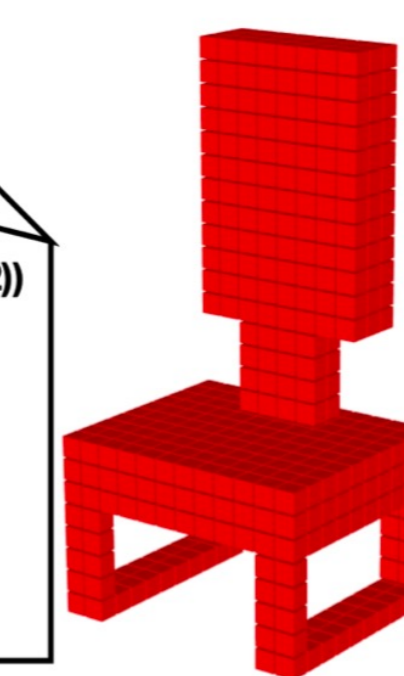
PointCloud2Program

Chair



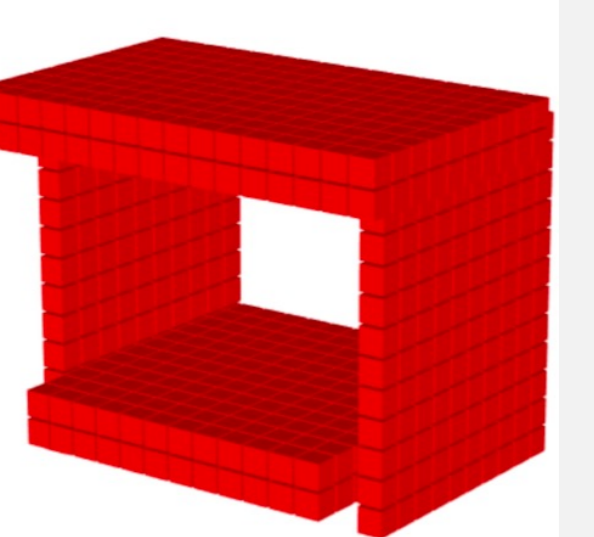
Neural Shape Compiler

```
draw('Top', 'Rec', P=(-7,0,0), G=(3,5,6))
for(i<2, 'Trans', u1=(0,0,10))
for(i<2, 'Trans', u2=(0,9,0))
draw('Leg', 'Cub', P=(-12,-5,-6)+i*u1+u2, G=(8,1,2))
for(i<2, 'Trans', u=(0,0,10))
draw('Hori_Bar', 'Cub', P=(-12,-5,-6)+i*u, G=(1,10,2))
draw('Back_sup', 'Cub', P=(-4,-2,-2), G=(4,2,4))
draw('Back', 'Cub', P=(0,2,-4), G=(12,3,8), theta=0°)
```

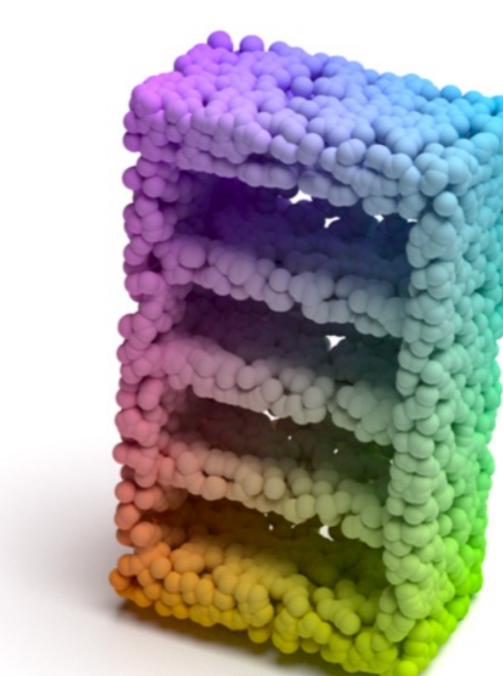


Neural Program Generator

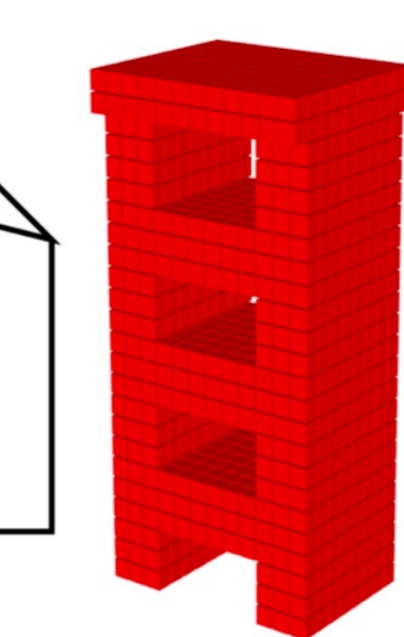
```
draw('Top', 'Rec', P=(0,-1,0), G=(2,5,8))
for(i<2, 'Trans', u=(0,0,13))
draw('Sideboard', 'Cub', P=(-11,-1,-7)+i*u, G=(12,5,1))
draw('Base', 'Square', P=(-10,0,0), G=(2,6))
```



Cabinet



```
draw('Top', 'Rec', P=(10,0,0), G=(2,5,6))
for(i<2, 'Trans', u=(0,0,9))
draw('Sideboard', 'Cub', P=(-12,0,-6)+i*u, G=(22,4,3))
for(i<3, 'Trans', u=(6,0,0))
draw('Layer', 'Rec', P=(-9,0,0)+i*u, G=(3,4,6))
```



```
draw('Top', 'Rec', P=(9,0,0), G=(1,6,8))
for(i<2, 'Trans', u1=(0,0,12))
for(i<2, 'Trans', u2=(0,6,0))
draw('Leg', 'Cub', P=(-12,-4,-7)+i*u1+u2, G=(23,2,1))
for(i<2, 'Trans', u=(0,0,11))
draw('Hori_Bar', 'Cub', P=(-9,-5,-7)+i*u, G=(0,7,2))
```

