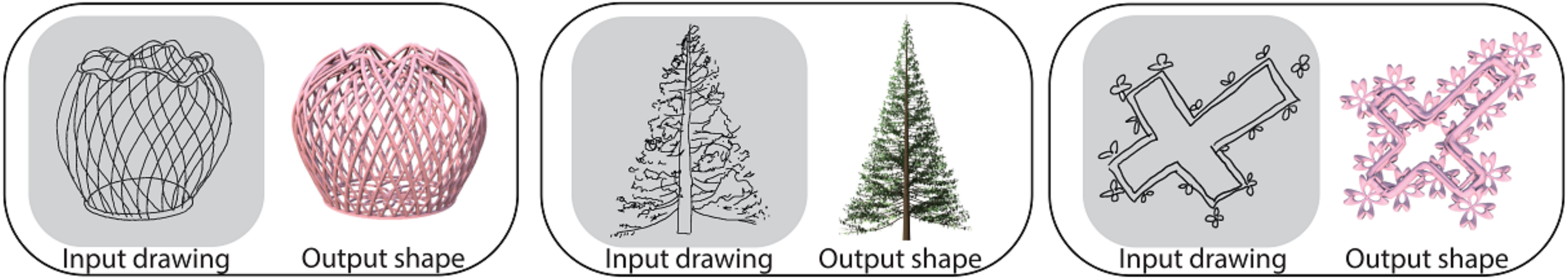
Shape Synthesis from Sketches via Procedural Models and Convolutional Networks



Haibin Huang[1] Evangelos Kalogerakis[1] M. Ersin Yumer [2] Radomir Mech [2]

[1]University of Massachusetts Amherst

[2]Adobe Research

Creating Detailed Visual Content is Hard!

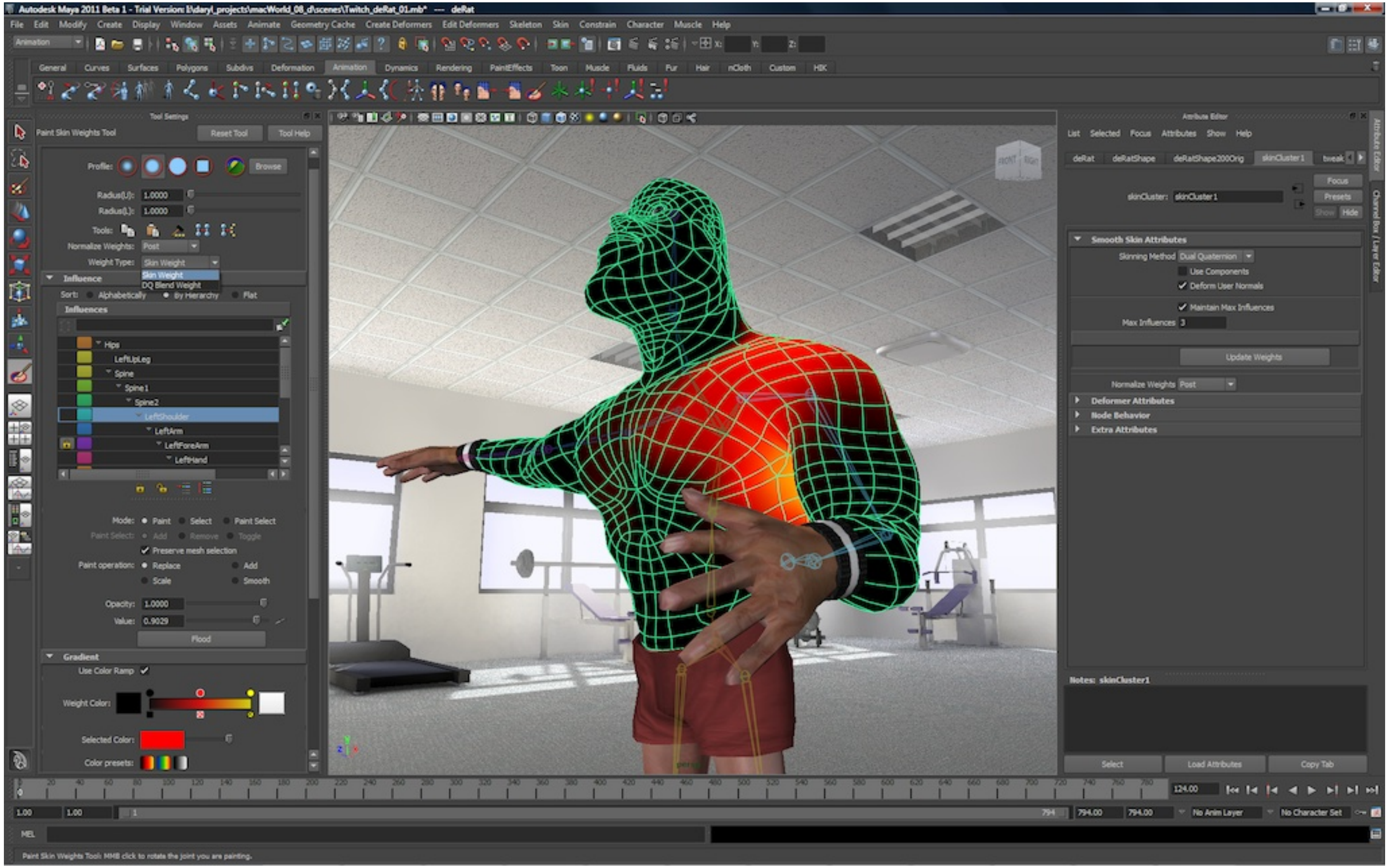
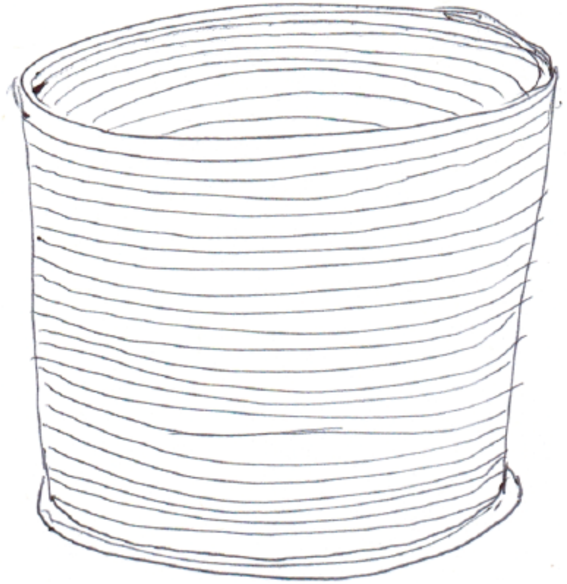


Image from Autodesk 3D Maya

Goal: Shape Synthesis from Sketches

How can we help users generate detailed shapes from simple inputs?

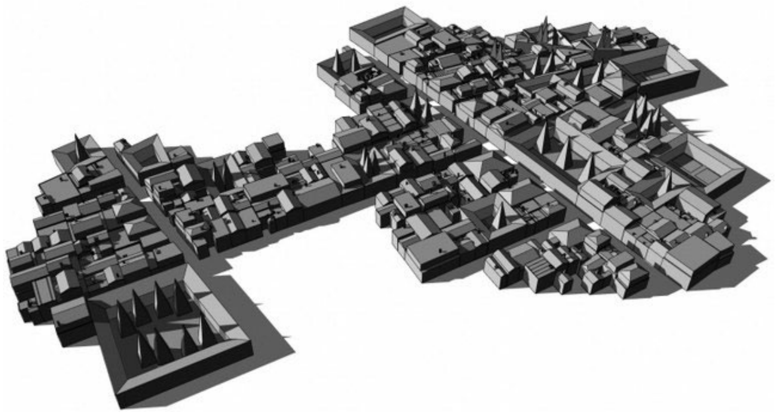


Input sketch

Output model

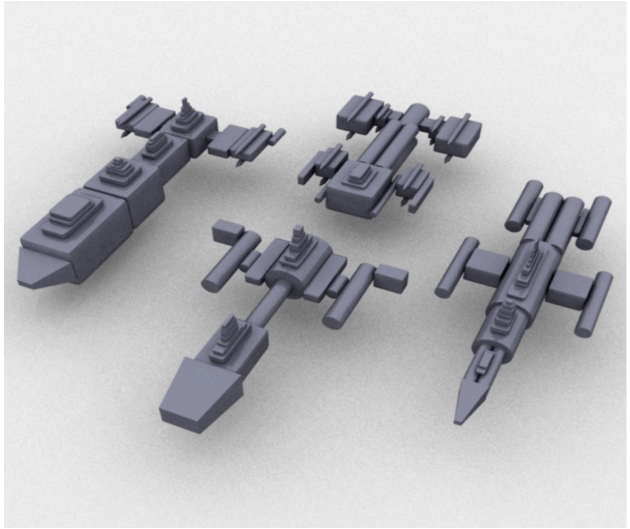
Motivation : Procedural Models

[Laubwerk kit]



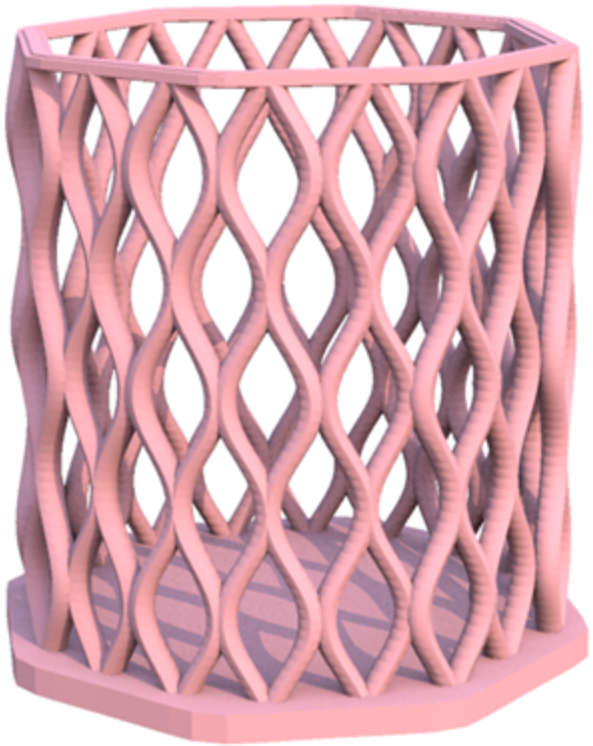
[CityEngine]

[VoxelStudio]



[Ritchie et al. 2015]

Motivation : Procedural Models



Container Type

Discrete parameters

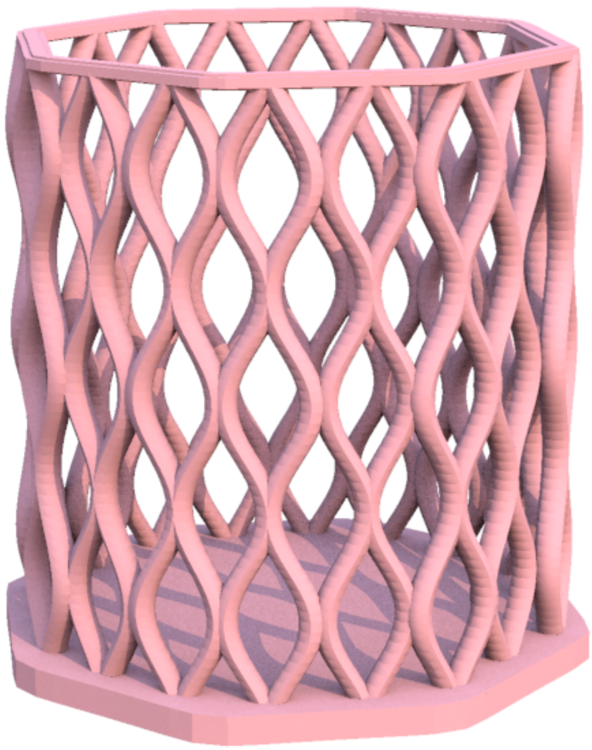
Container Size

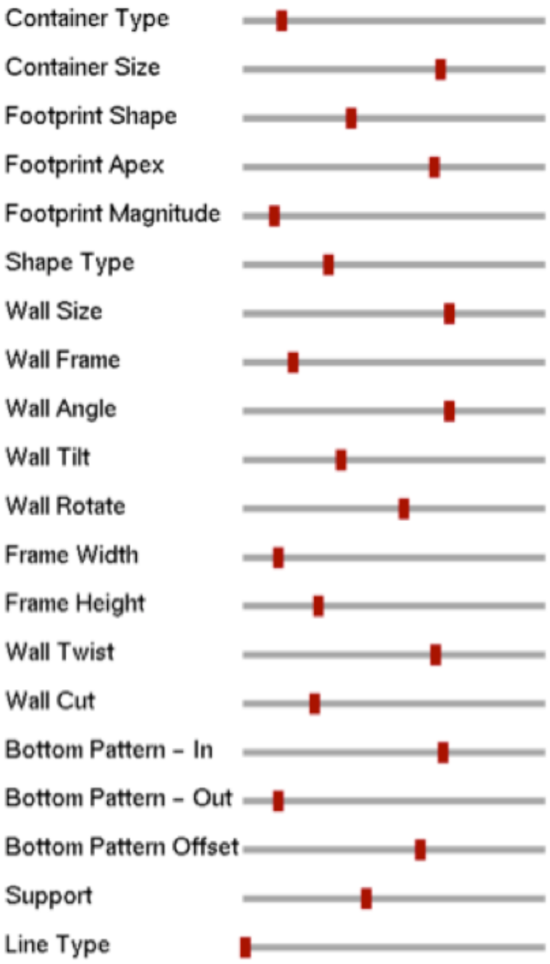
Wall Angle

Wall Size

Continuous parameters

Motivation : Procedural Models

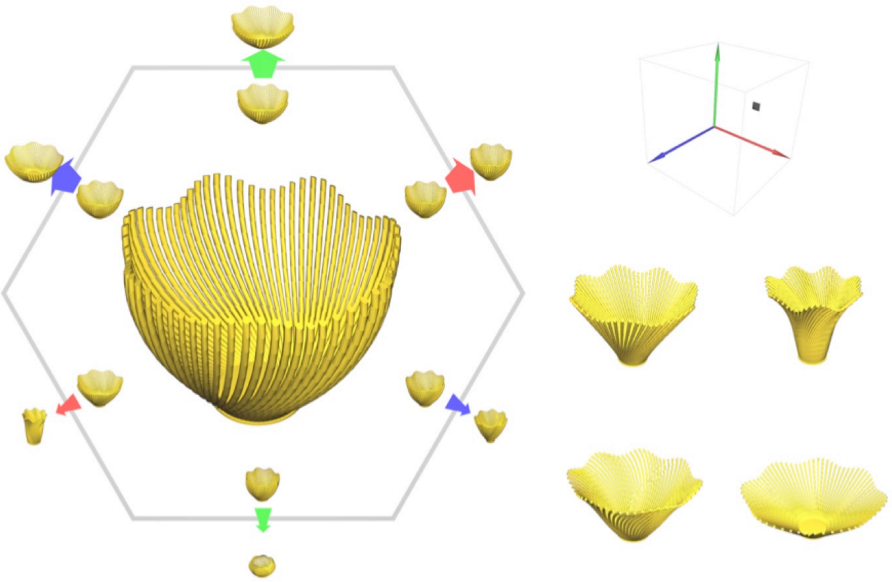




User Controlled Parameters

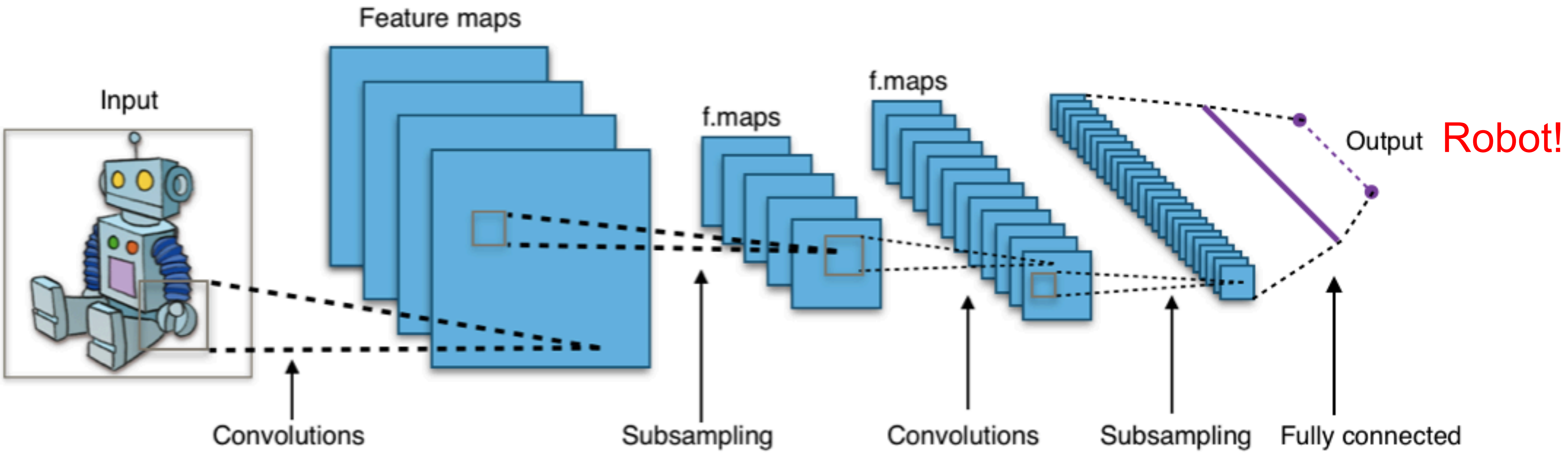
Prior work: Procedural Models

Talton et al.
Exploratory modeling with collaborative design spaces
SIGGRAPH 2009



Yumer et al.
Procedural modeling using autoencoder networks
UIST 2015

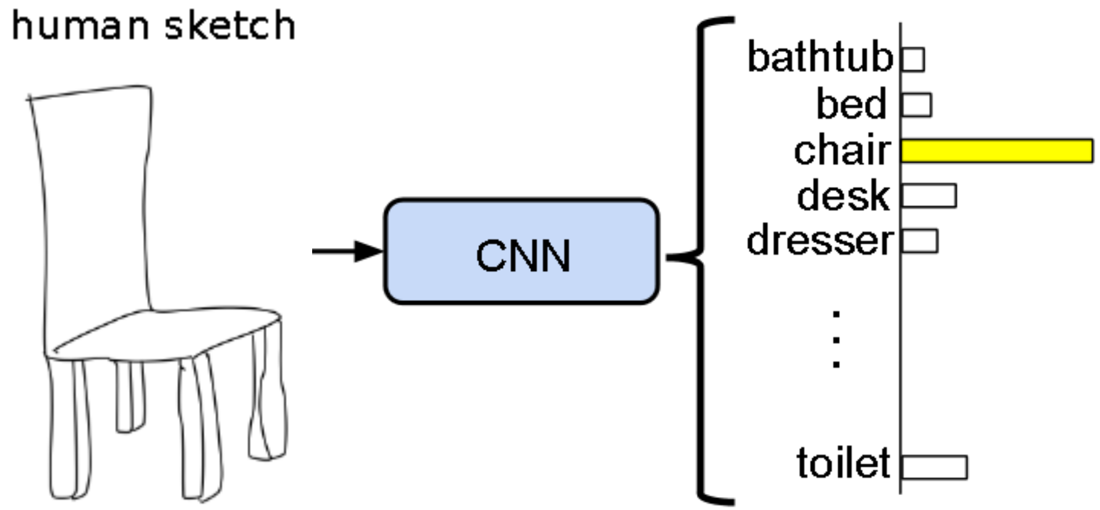
Motivation: CNNs can learn complex functions



https://en.wikipedia.org/wiki/Convolutional_neural_network

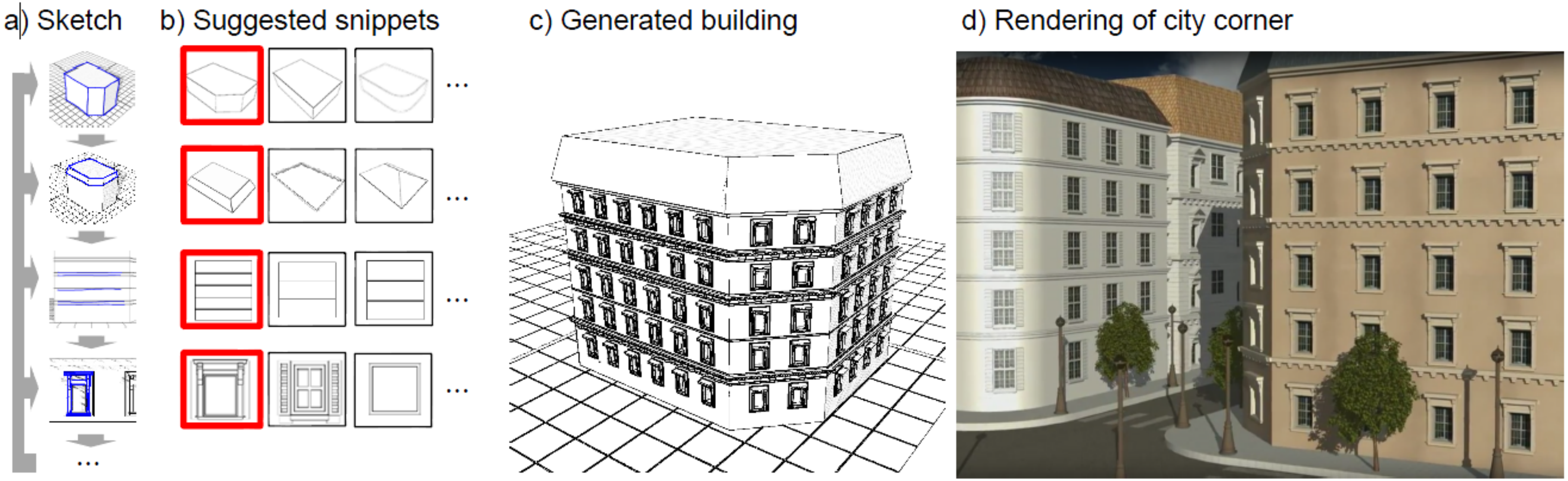
Prior work: CNNs for sketch recognition & shape retrieval

Wang et al.,
Sketch-based 3D Shape Retrieval using CNNs
CVPR 2015



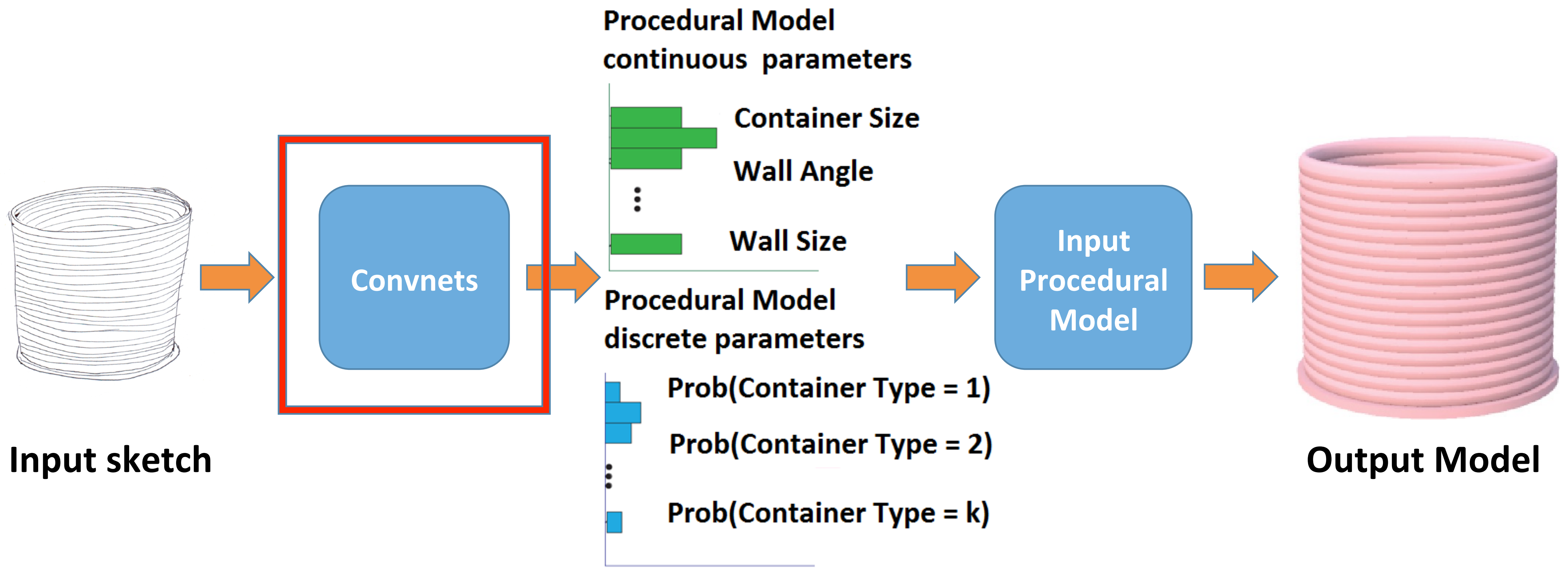
Su et al.
Multi-view CNNs for 3D Shape Recognition
ICCV 2015

Concurrent work: Urban procedural model design from sketches

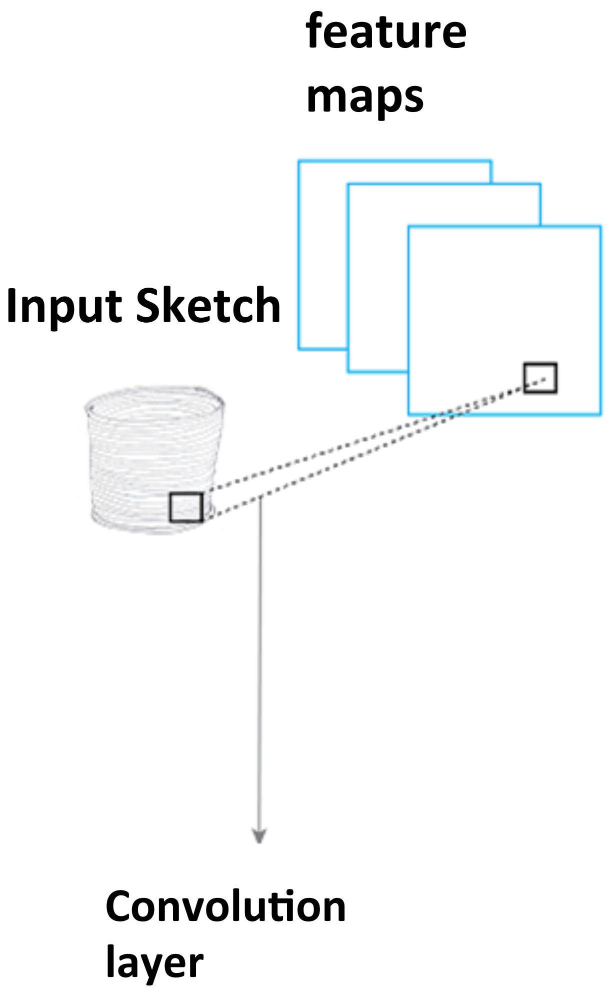


Nishida et al.
Interactive Sketching of Urban Procedural Models
SIGGRAPH 2016

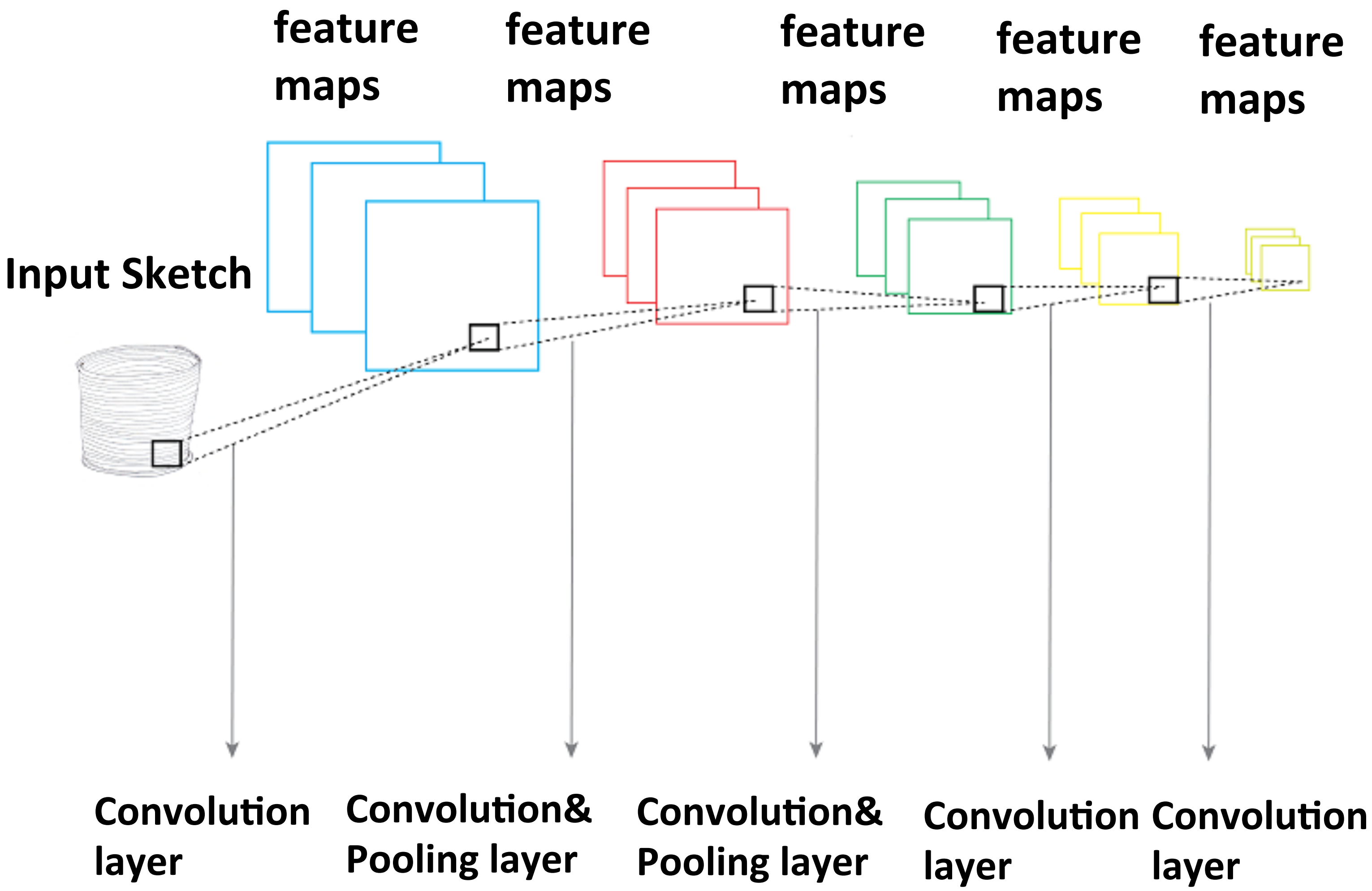
Best of both worlds: Convnets + Procedural Modeling



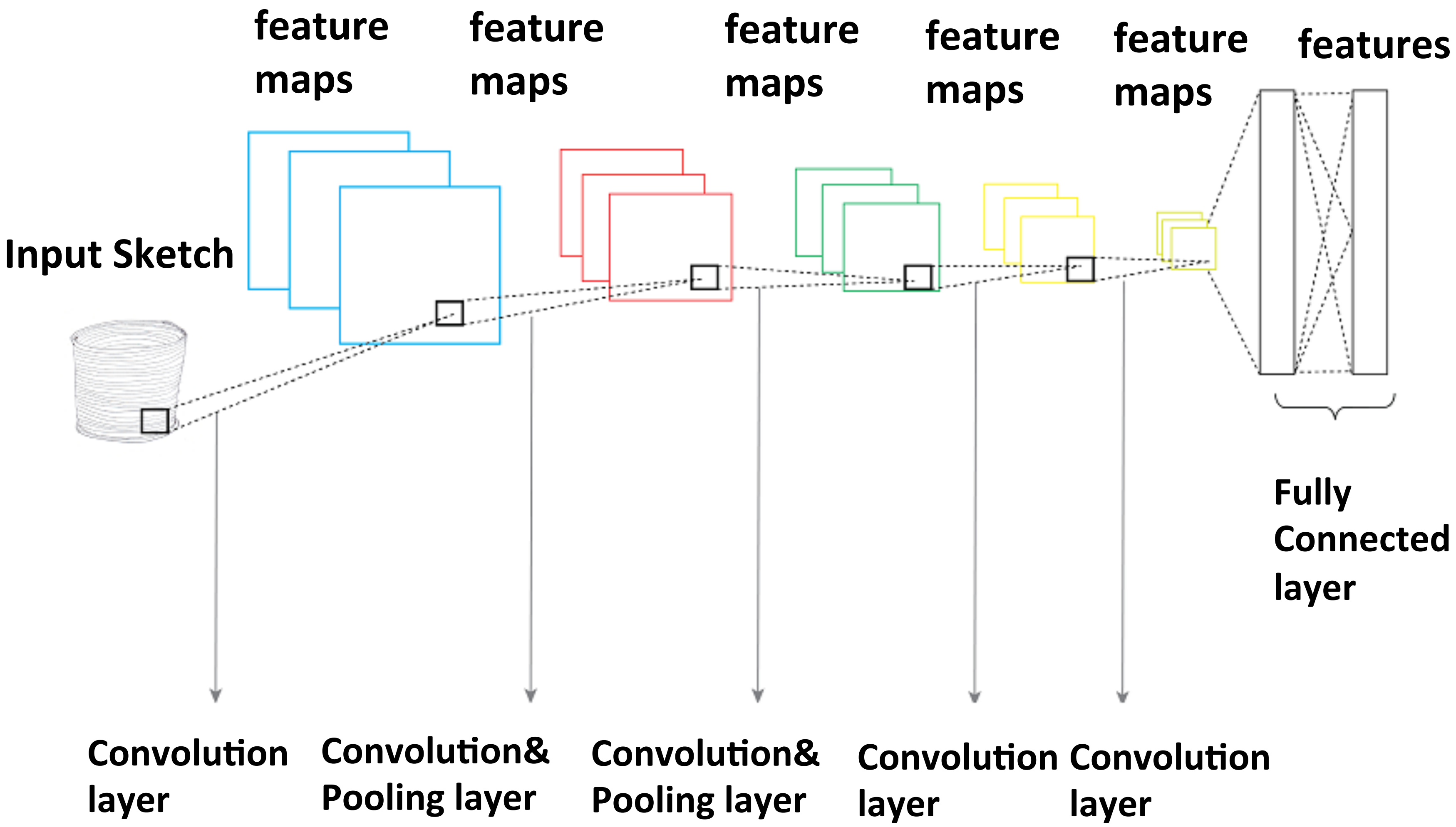
Overview: Learning architecture



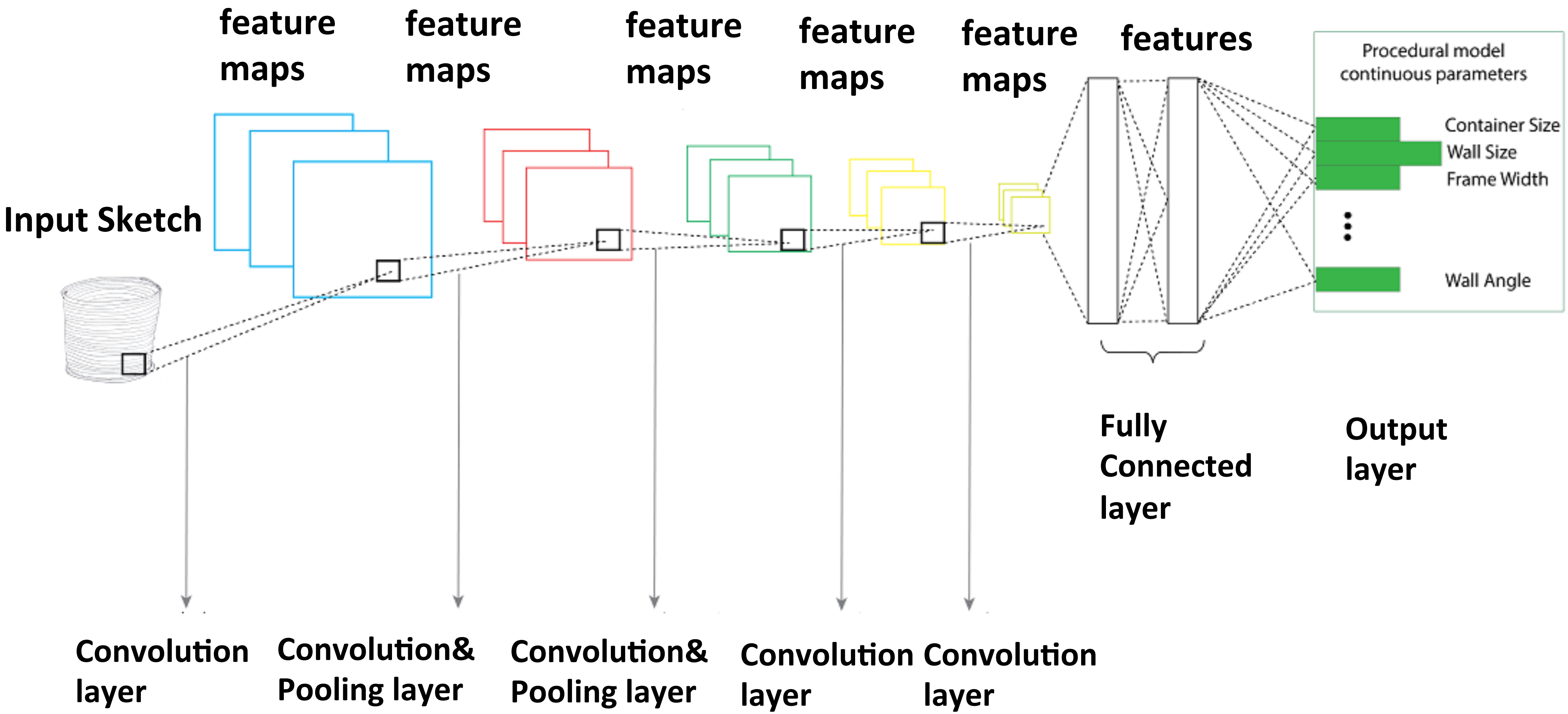
Overview: Learning architecture



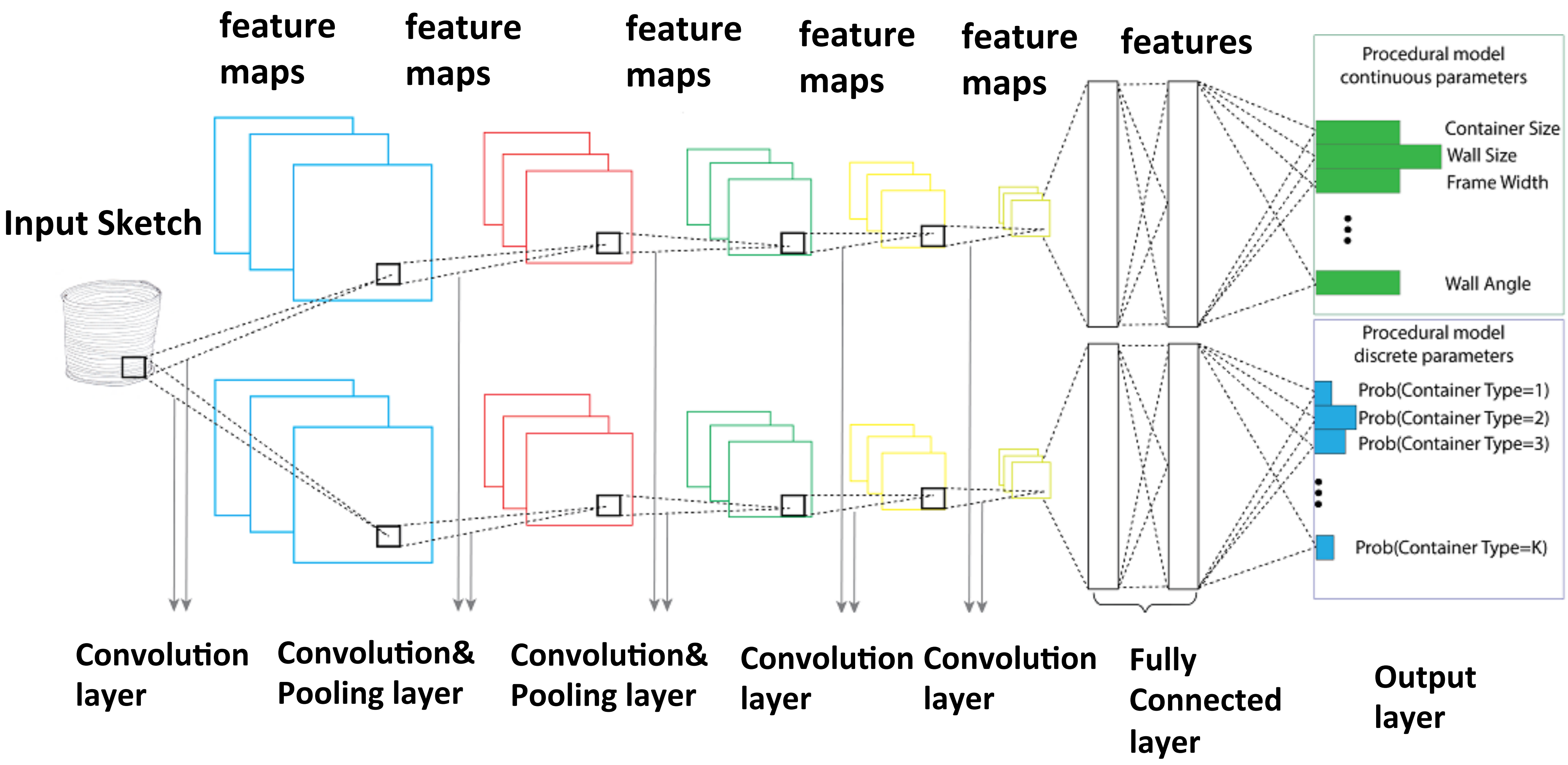
Overview: Learning architecture



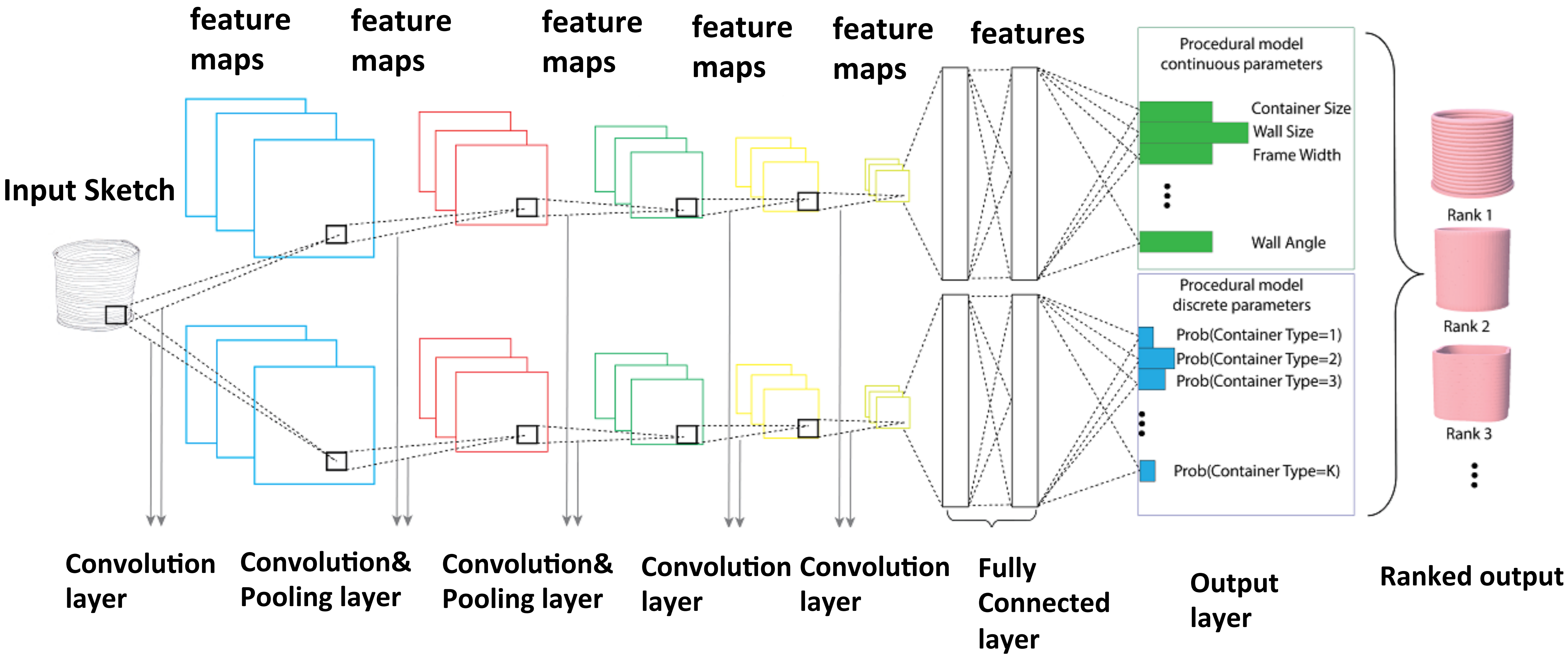
Overview: Learning architecture



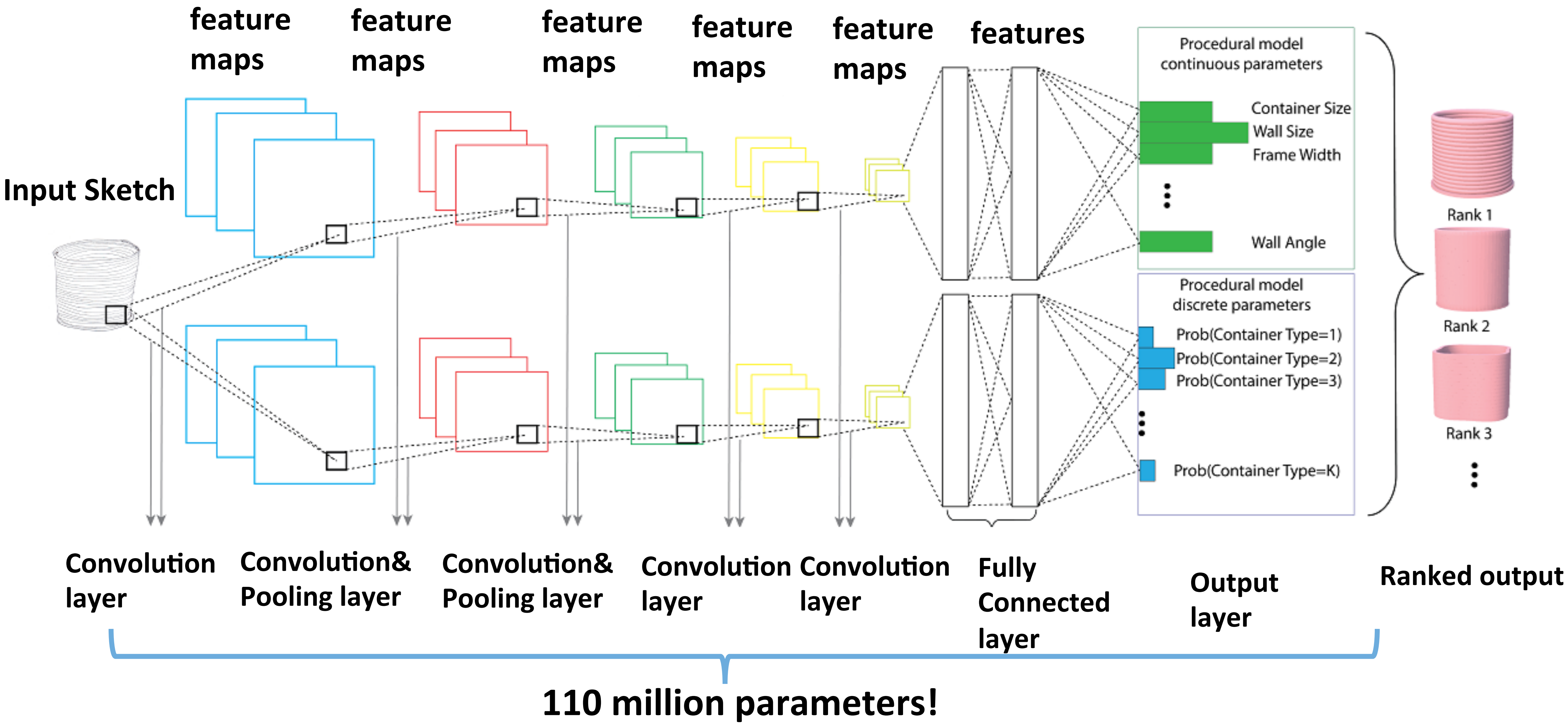
Overview: Learning architecture



Overview: Learning architecture

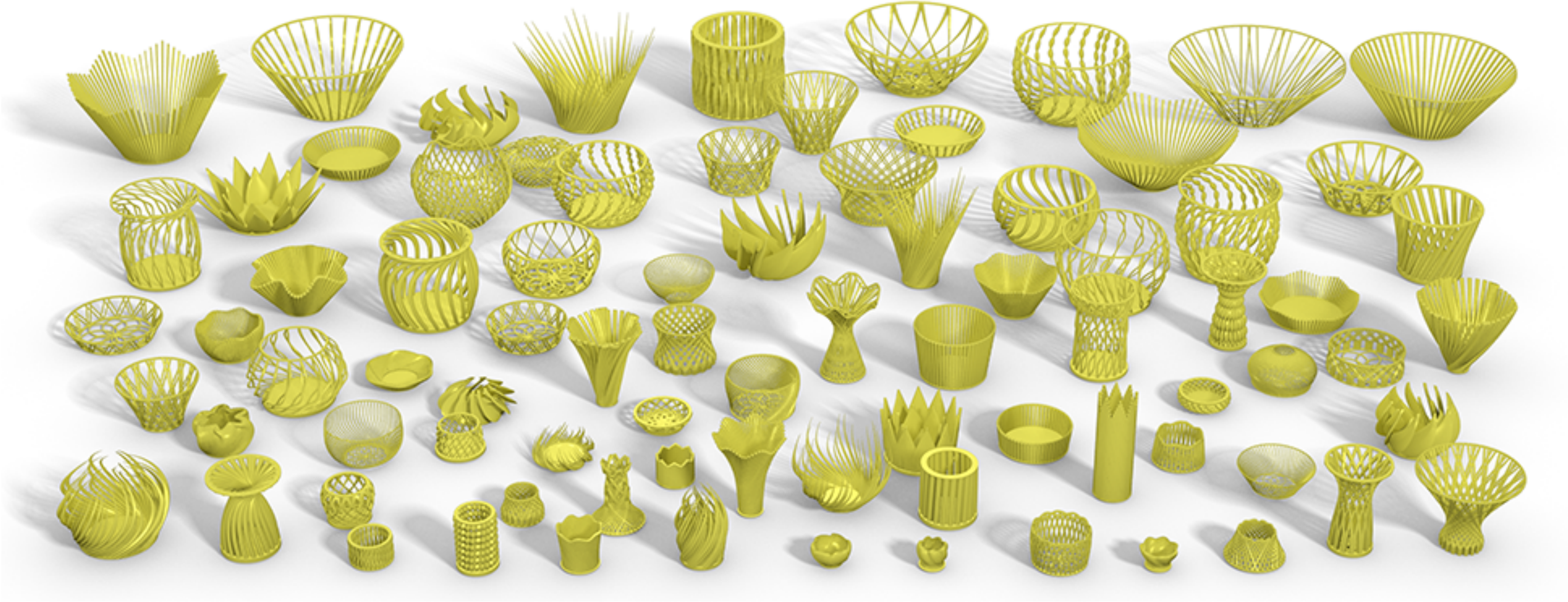


Overview: Learning architecture

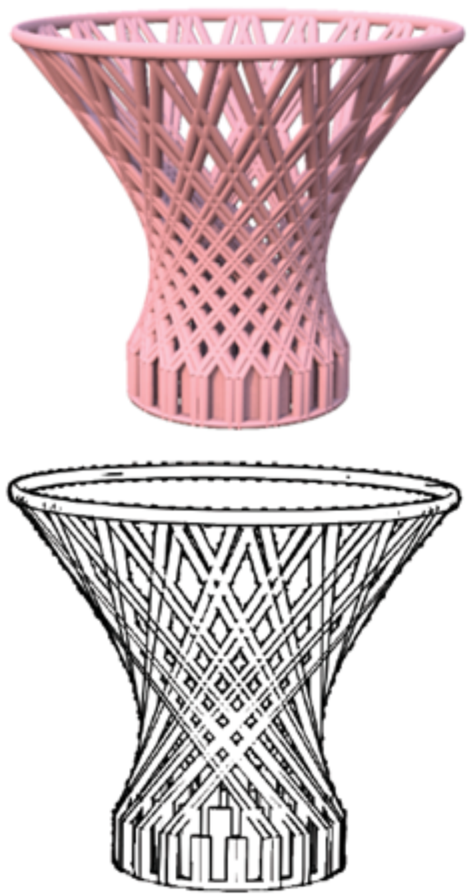


Training procedure: Synthetic Training Sketch Generation

Uniform sampling the parameter space

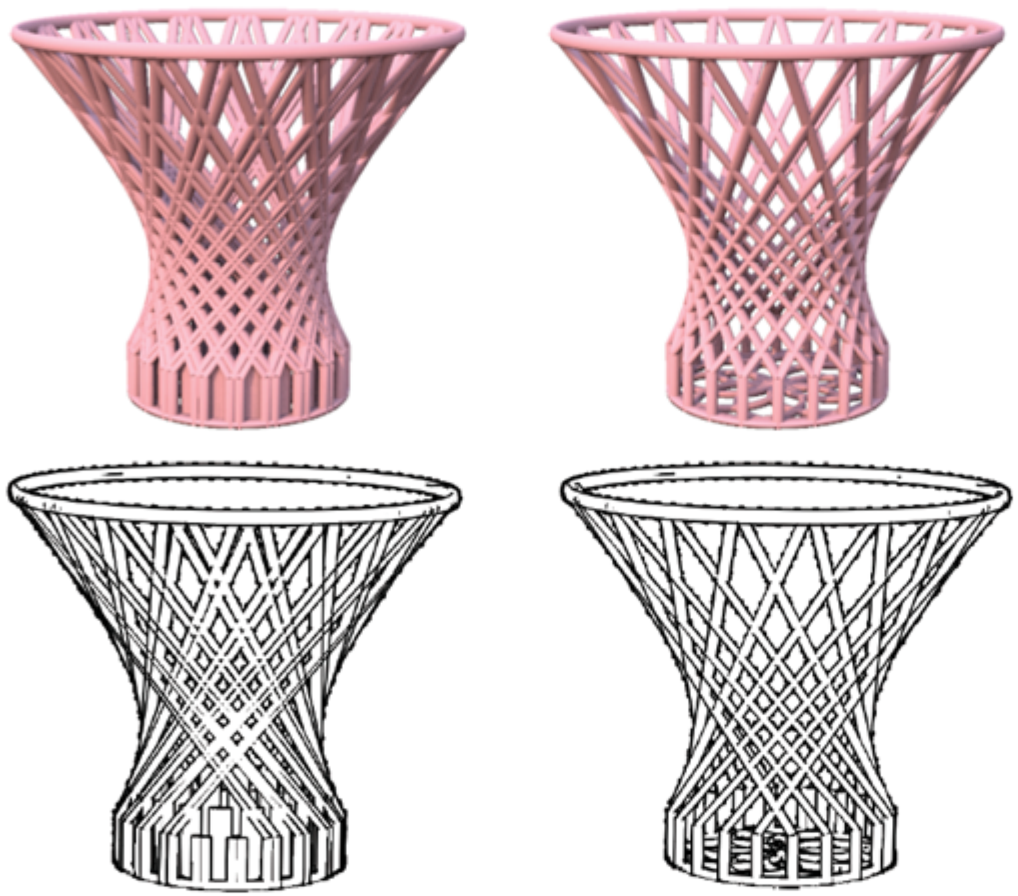


Training procedure: Synthetic Training Sketch Generation



original shape

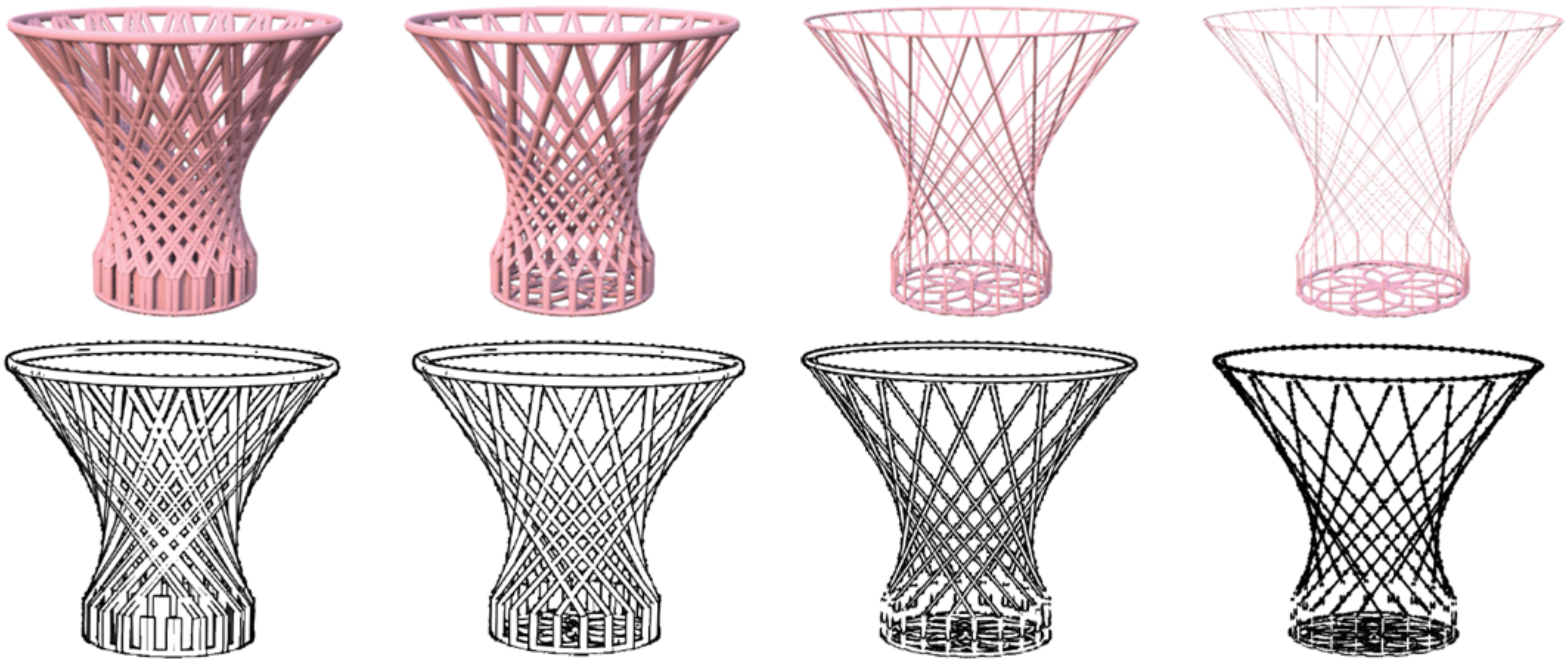
Training procedure: Synthetic Training Sketch Generation



original shape

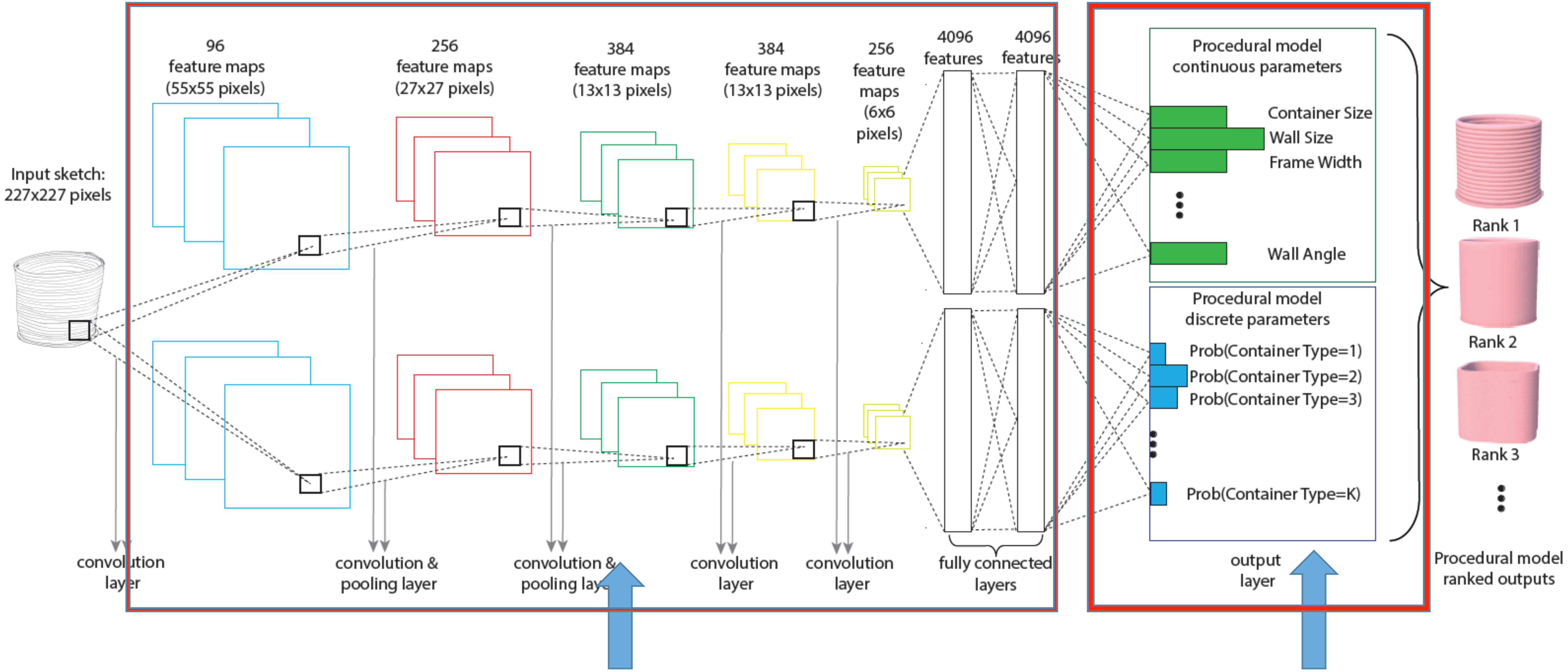
no contractions
50% subsampling

Training procedure: Synthetic Training Sketch Generation



original shape | no contractions 50% subsampling | 1 contraction 50% subsampling | 3 contractions 50% subsampling

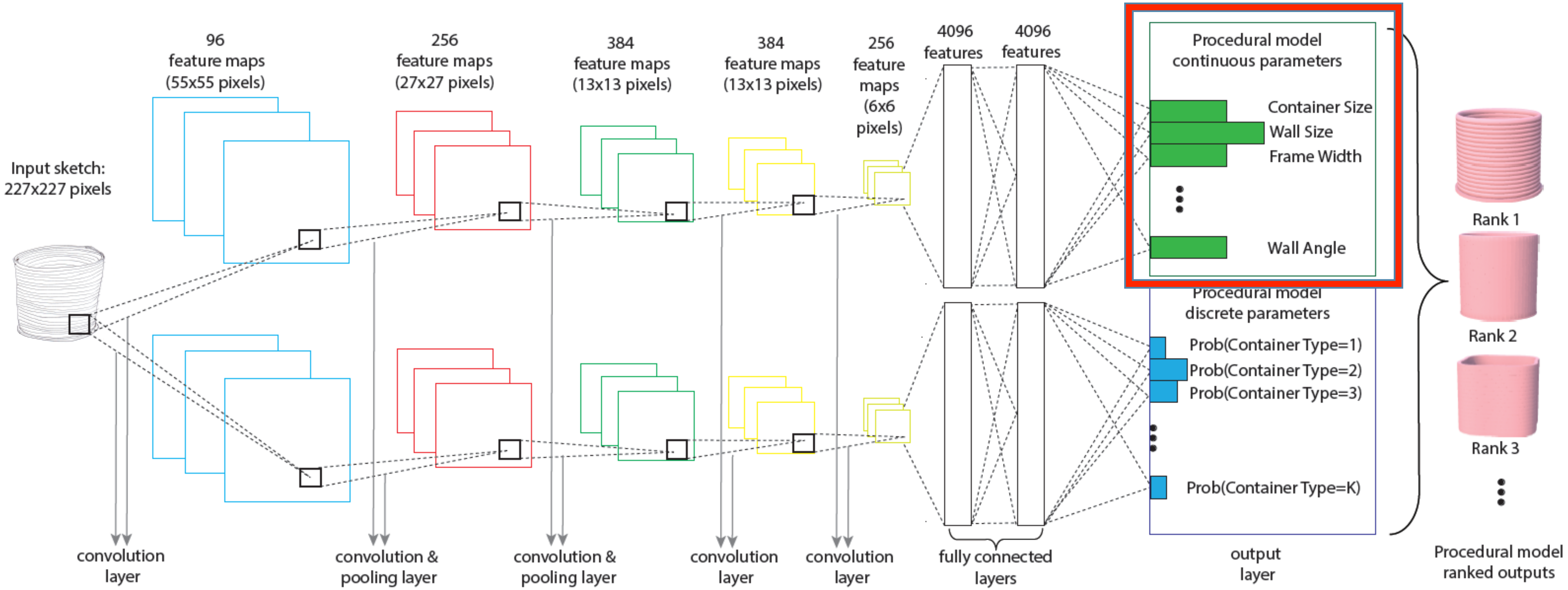
Training procedure: Pre-training and fine-tuning



Initialized from AlexNet

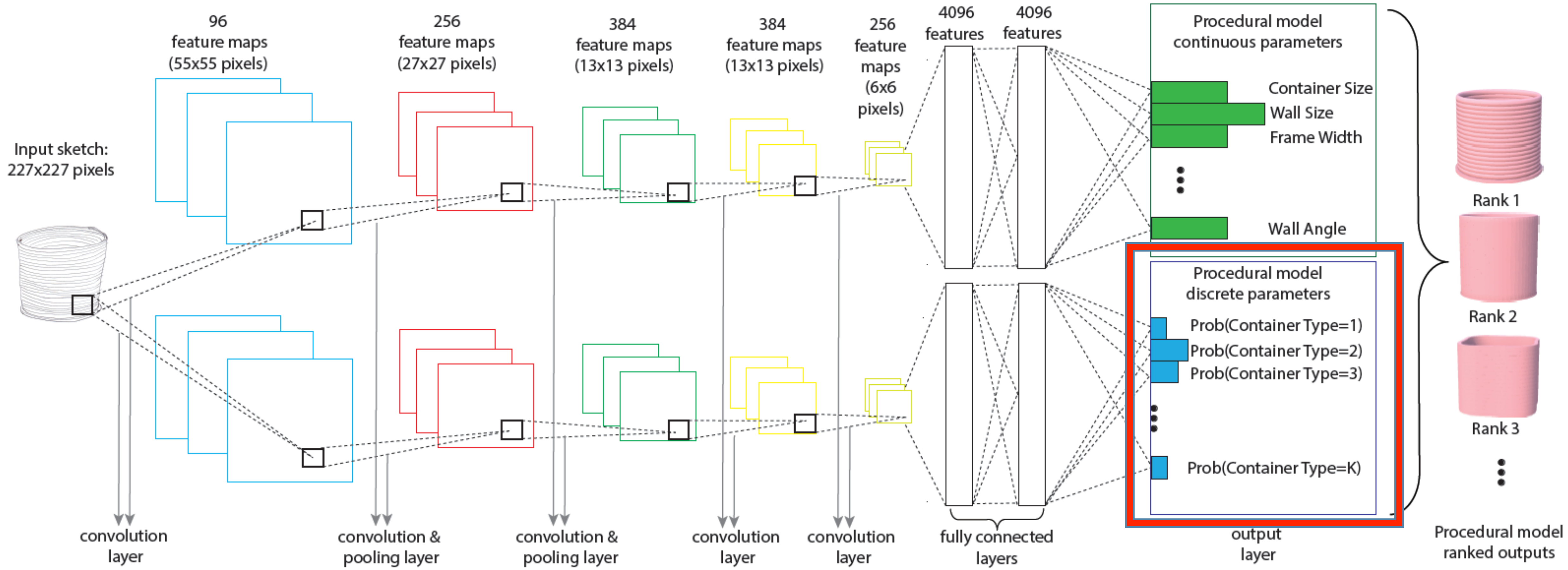
Trained from scratch

Training procedure: Pre-training and fine-tuning



$$E_r(\boldsymbol{\theta}_1) = \sum_{s=1}^{S}\sum_{c=1}^{C}[\delta_{c,s} == 1]||O_{c,s}(\boldsymbol{\theta}_1) - \hat{O}_{c,s}||^2 + \lambda_1||\boldsymbol{\theta}_1||^2 \quad (4)$$

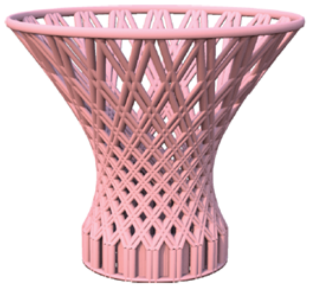
Training procedure: Pre-training and fine-tuning



$$E_c(\boldsymbol{\theta}_2) = -\sum_{s=1}^{S}\sum_{r=1}^{R} ln(Prob(D_{s,r} = \hat{d}_{s,r}; \boldsymbol{\theta}_2)) + \lambda_2 ||\boldsymbol{\theta}_2||^2 \quad (5)$$

Datasets: Deco framework

1 . 3D containers



2 . 3D jewelry

3 . 2D trees

Data Statistics

Statistics	Containers	Trees	Jewelry
# training shapes	$30k$	$60k$	$15k$
# training sketches	$120k$	$240k$	$60k$
# continuous parameters	24	20	15
# discrete parameter values/classes	27	34	13
training time (hours)	12	20	9
runtime stage time (sec)	1.5	1.6	1.2

TABLE 1: Dataset statistics

Data Statistics

Statistics	Containers	Trees	Jewelry
# training shapes	$30k$	$60k$	$15k$
# training sketches	$120k$	$240k$	$60k$
# continuous parameters	24	20	15
# discrete parameter values/classes	27	34	13
training time (hours)	12	20	9
runtime stage time (sec)	1.5	1.6	1.2

TABLE 1: Dataset statistics

Data Statistics

Statistics	Containers	Trees	Jewelry
# training shapes	$30k$	$60k$	$15k$
# training sketches	$120k$	$240k$	$60k$
# continuous parameters	24	20	15
# discrete parameter values/classes	27	34	13
training time (hours)	12	20	9
runtime stage time (sec)	1.5	1.6	1.2

TABLE 1: Dataset statistics

Data Statistics

Statistics	Containers	Trees	Jewelry
# training shapes	$30k$	$60k$	$15k$
# training sketches	$120k$	$240k$	$60k$
# continuous parameters	24	20	15
# discrete parameter values/classes	27	34	13
training time (hours)	12	20	9
runtime stage time (sec)	1.5	1.6	1.2

TABLE 1: Dataset statistics

Data Statistics

Statistics	Containers	Trees	Jewelry
# training shapes	$30k$	$60k$	$15k$
# training sketches	$120k$	$240k$	$60k$
# continuous parameters	24	20	15
# discrete parameter values/classes	27	34	13
training time (hours)	12	20	9
runtime stage time (sec)	1.5	1.6	1.2

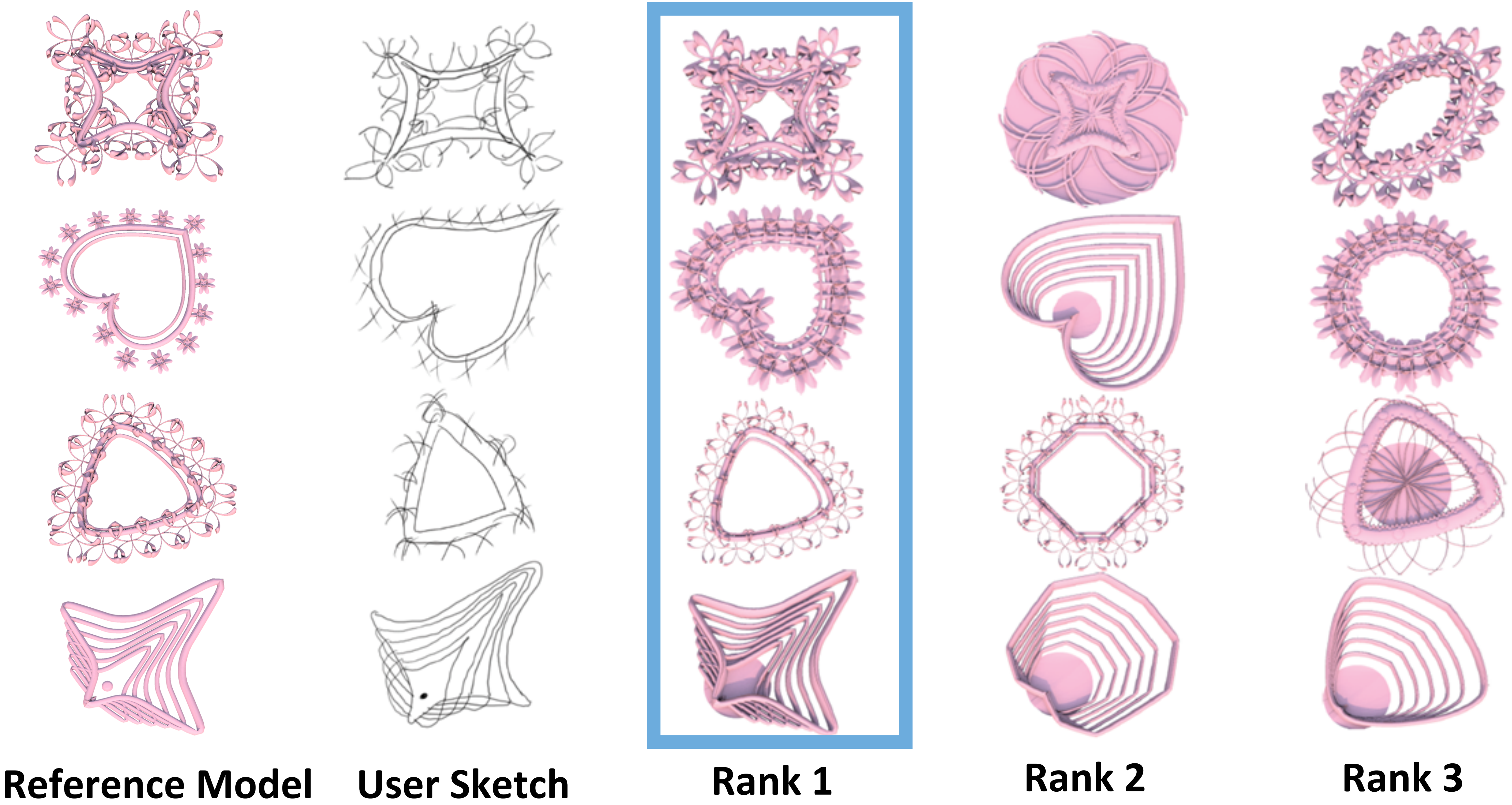
TABLE 1: Dataset statistics

Data Statistics

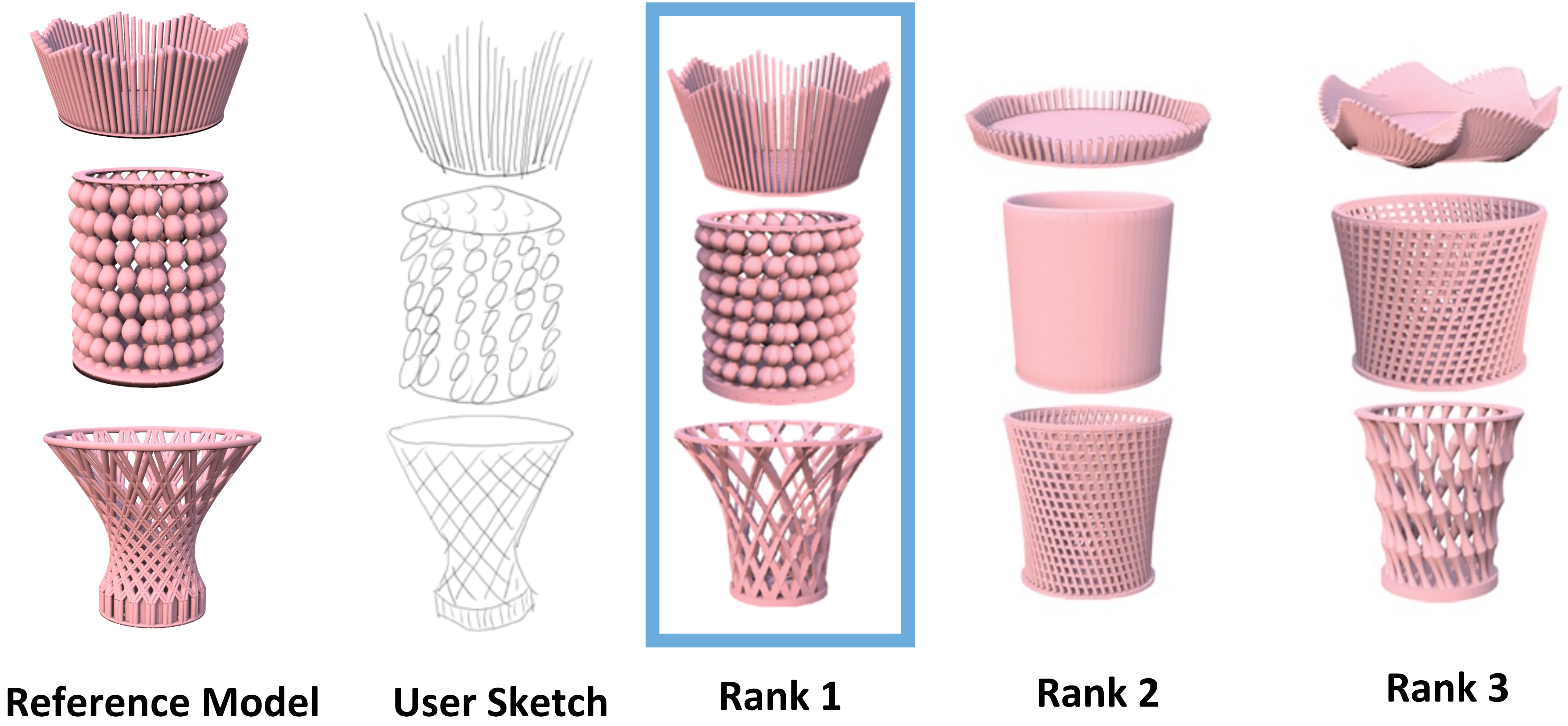
Statistics	Containers	Trees	Jewelry
# training shapes	$30k$	$60k$	$15k$
# training sketches	$120k$	$240k$	$60k$
# continuous parameters	24	20	15
# discrete parameter values/classes	27	34	13
training time (hours)	12	20	9
runtime stage time (sec)	1.5	1.6	1.2

TABLE 1: Dataset statistics

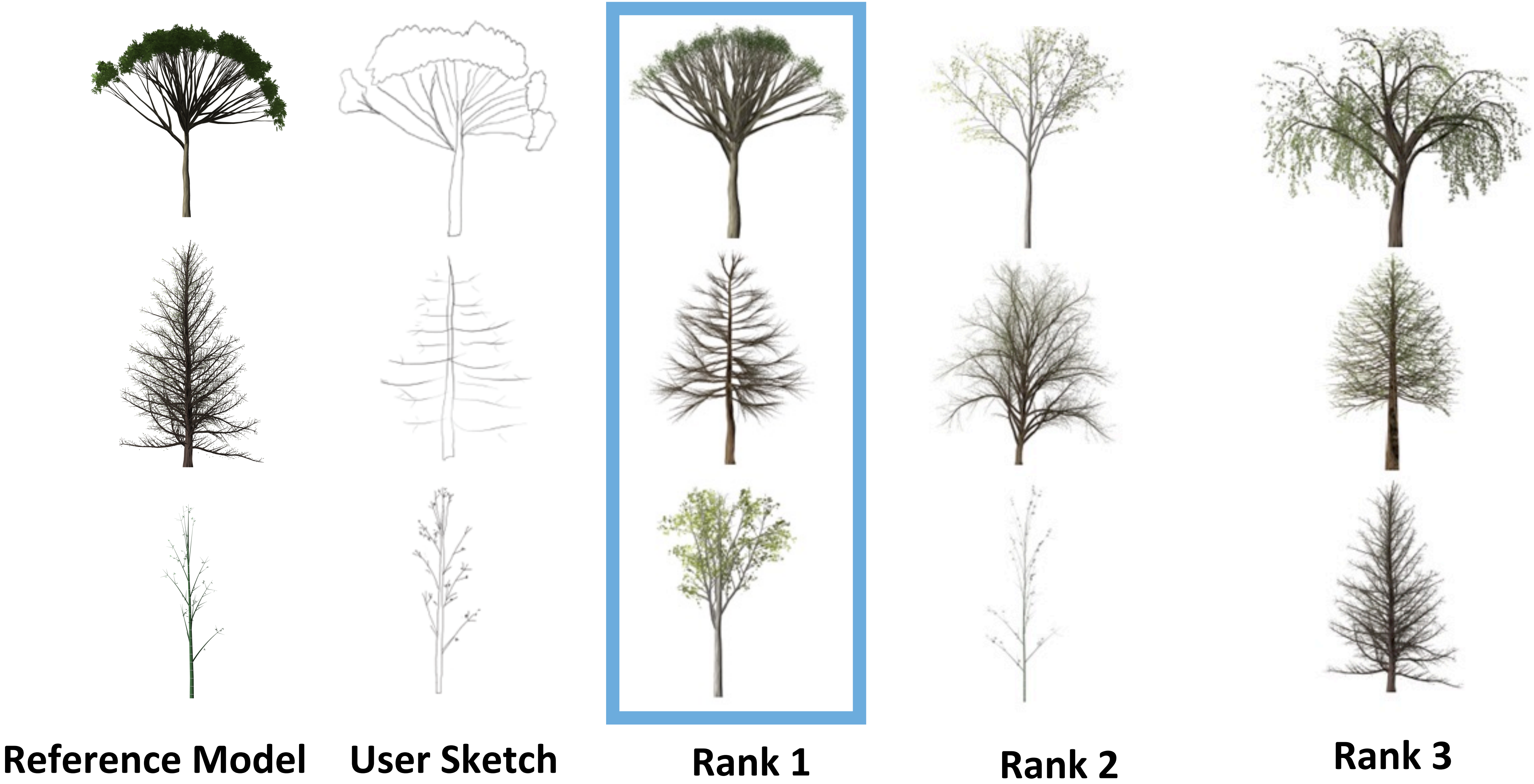
Results: Pendants



Results: Containers



Results: Trees



Comparisons with alternatives: Discrete parameters

Method	Containers	Trees	Jewelry	Average
Nearest neighbors (Fisher)	33.3%	26.7%	46.7%	35.6%
Nearest neighbors (CNN)	26.7%	33.3%	40.0%	33.3%
SVM (Fisher)	33.3%	46.7%	60.0%	46.7%
SVM (CNN)	40.0%	33.3%	53.3%	42.2%
Our method	**80.0%**	**73.3%**	**86.7%**	**80.0%**

TABLE 3: Top-3 classification accuracy for PM discrete parameters predicted by the examined methods on our user study line drawings.

The higher the better

Comparisons with alternatives: Discrete parameters

Method	Containers	Trees	Jewelry	Average
Nearest neighbors (Fisher)	33.3%	26.7%	46.7%	35.6%
Nearest neighbors (CNN)	26.7%	33.3%	40.0%	33.3%
SVM (Fisher)	33.3%	46.7%	60.0%	46.7%
SVM (CNN)	40.0%	33.3%	53.3%	42.2%
Our method	**80.0%**	**73.3%**	**86.7%**	**80.0%**

TABLE 3: Top-3 classification accuracy for PM discrete parameters predicted by the examined methods on our user study line drawings.

The higher the better

Comparisons with alternatives: Discrete parameters

Method	Containers	Trees	Jewelry	Average
Nearest neighbors (Fisher)	33.3%	26.7%	46.7%	35.6%
Nearest neighbors (CNN)	26.7%	33.3%	40.0%	33.3%
SVM (Fisher)	33.3%	46.7%	60.0%	46.7%
SVM (CNN)	40.0%	33.3%	53.3%	42.2%
Our method	**80.0%**	**73.3%**	**86.7%**	**80.0%**

TABLE 3: Top-3 classification accuracy for PM discrete parameters predicted by the examined methods on our user study line drawings.

The higher the better

Comparisons with alternatives: Discrete parameters

Method	Containers	Trees	Jewelry	Average
Nearest neighbors (Fisher)	33.3%	26.7%	46.7%	35.6%
Nearest neighbors (CNN)	26.7%	33.3%	40.0%	33.3%
SVM (Fisher)	33.3%	46.7%	60.0%	46.7%
SVM (CNN)	40.0%	33.3%	53.3%	42.2%
Our method	**80.0%**	**73.3%**	**86.7%**	**80.0%**

TABLE 3: Top-3 classification accuracy for PM discrete parameters predicted by the examined methods on our user study line drawings.

The higher the better

Comparisons with alternatives: Discrete parameters

Method	Containers	Trees	Jewelry	Average
Nearest neighbors (Fisher)	33.3%	26.7%	46.7%	35.6%
Nearest neighbors (CNN)	26.7%	33.3%	40.0%	33.3%
SVM (Fisher)	33.3%	46.7%	60.0%	46.7%
SVM (CNN)	40.0%	33.3%	53.3%	42.2%
Our method	**80.0%**	**73.3%**	**86.7%**	**80.0%**

TABLE 3: Top-3 classification accuracy for PM discrete parameters predicted by the examined methods on our user study line drawings.

The higher the better

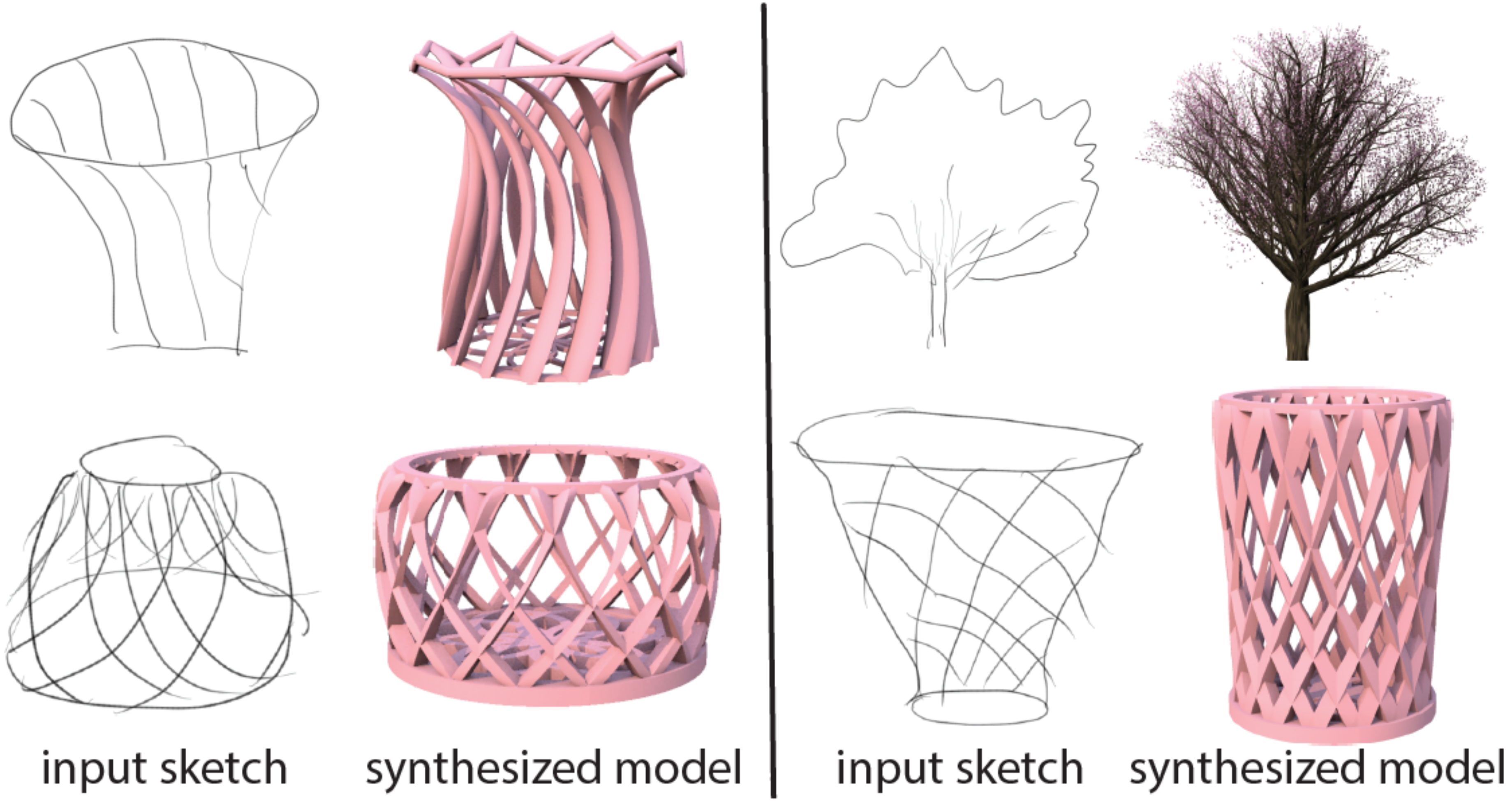
Comparisons with alternatives: continuous parameters

Method	Containers	Trees	Jewelry	Average
Nearest neighbors (Fisher)	32.1%	36.3%	29.1%	32.5%
Nearest neighbors (CNN)	29.3%	34.7%	27.5%	30.3%
RBF (Fisher)	30.5%	35.6%	28.9%	37.1%
RBF (CNN)	31.4%	34.2%	27.6%	31.1%
Our method	**12.7%**	**15.6%**	**8.7%**	**12.3%**

TABLE 4: PM continuous parameter error (regression error) of the examined methods on our user study line drawings.

The lower the better

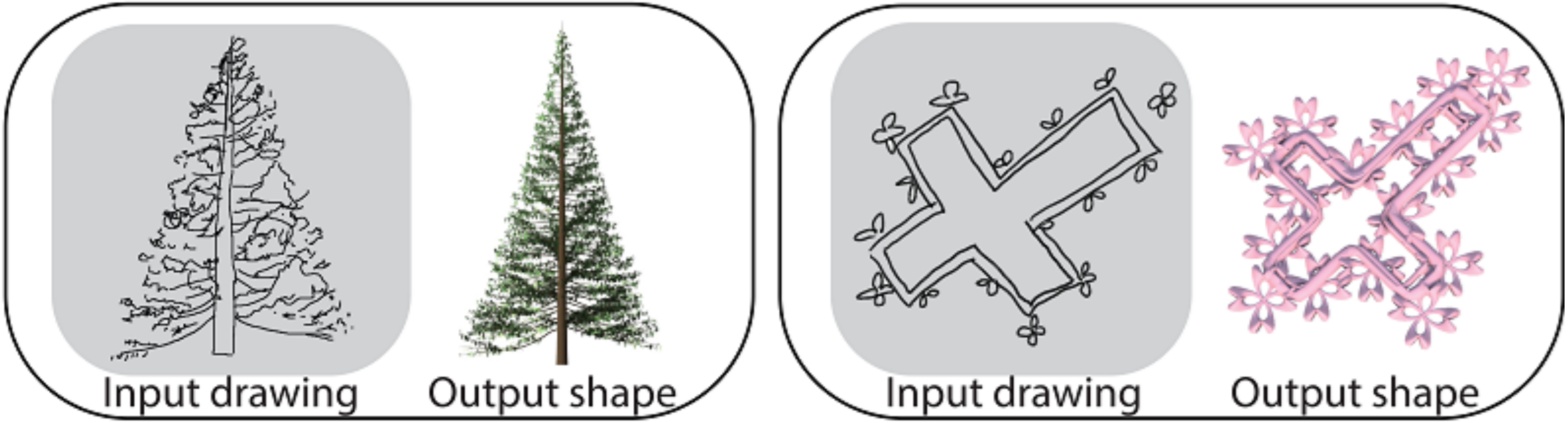
Limitations



Summary

A deep CNN that maps sketches to procedural model outputs

Generates detailed shapes through a parametric rule set and line drawings as input



Future work

Speed up and make the system in real-time

Simulate human style sketch to improve the performance

Allow user to edit the model after he gets the initial model

Thank you!

Acknowledgements: Daichi Ito, Olga Vesselova

Our project web page:

http://people.cs.umass.edu/~hbhuang/publications/srpm