Interactive Design of Probability Density Functions for Shape Grammars

Supplementary Material

Contents

Supplemental Material - Urban Planning Use Case

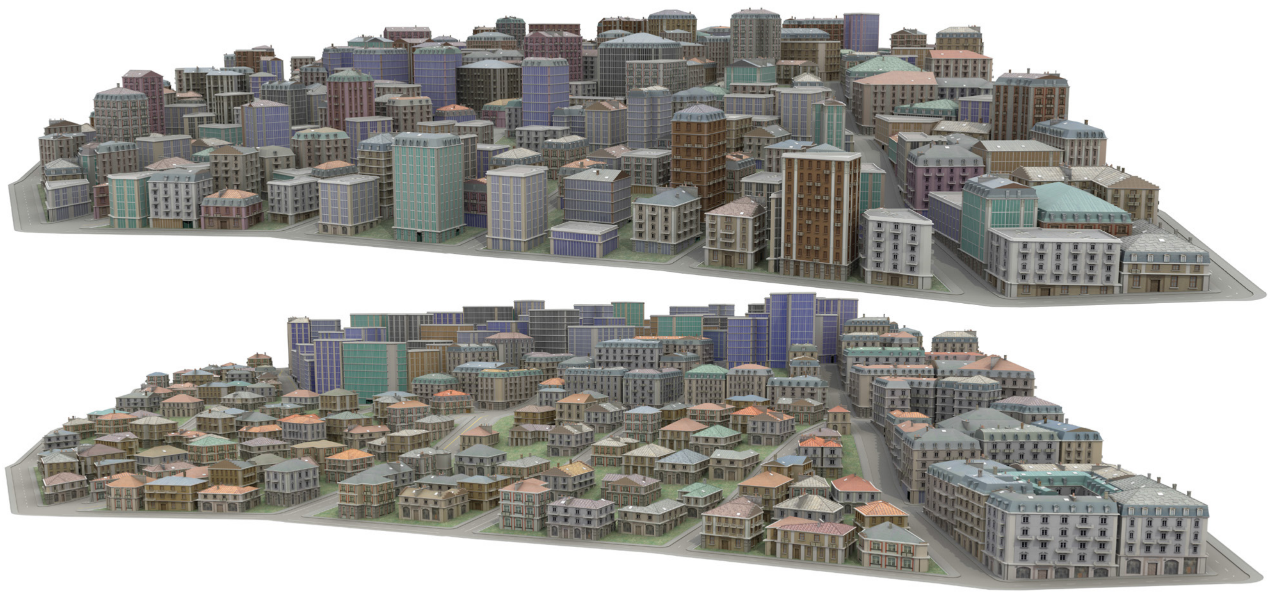


Figure 1: *We use a grammar that generates a variety of buildings for a city modeling task. (top) A direct application of the grammar leads to undesirable results. For example, office buildings are mixed with Haussmannian buildings and small residential houses without a clear structure of different neighborhoods. (bottom) We use our framework to design three different probability density functions for this grammar, which bias towards the generation of high-rise office buildings (far), downtown Haussmannian buildings (right) and residential houses (left). We can also ensure the matching of house styles, roofs, and wall colors.*

Consider the task of modeling a city using an existing shape grammar that can generate a variety of different buildings. If the grammar is general enough to cover a broad range of styles over multiple building components such as roofs, façades, windows, etc., then the initial result could look very chaotic, e.g., as in Fig. 1 (top). The buildings have random materials and a random height distribution.

Some of the generated models also do not make sense because they contain mismatching styles and colors within the model. In Fig. 2 we illustrate several such example mismatches from the chaotic city that can occur when the grammar is too general.

While it is possible for an expert to manually edit the grammar to enforce all required design constraints, it is not a scalable solution.

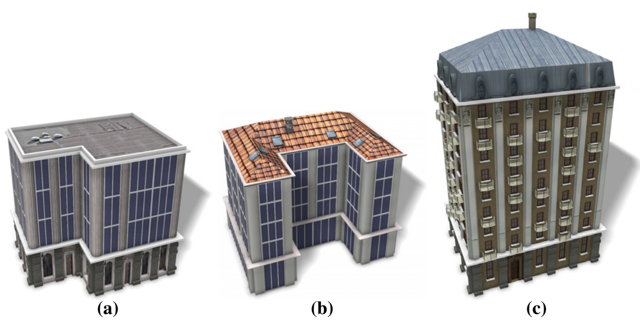


Figure 2: *Possible style mismatches of building components:*
(a) Mismatch of ground and upper floors styles. A house with an ancient ground floor should not have its higher floors in a modern office style with glass façades.
(b) Mismatching roof and façade. Modern office buildings do not usually have old red tile roofs.
(c) Too many floors. The quintessential Parisian-like Haussmannian building should not be larger than six floors.

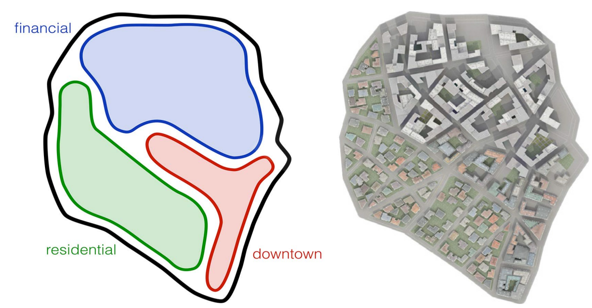


Figure 3: *Top view: (left) layout of the districts, (right) rendering.*

As more components from different architectural styles are added to the grammar, it becomes more complex to encode all their interdependencies, and maintenance of the grammar will become a tedious task due to the combinatorial explosion. With our framework even novice users can model pdfs for the building grammar that avoid these mismatches.

Imagine that we want to divide our example city into three distinct districts as shown in the top-down perspective in Fig. 3: a residential region for small suburban buildings, a finacial district with office buildings, and a downtown area featuring Haussmannian architecture. To achieve our design objective, we specify two goals: 1) each district should contain buildings appropriate for that district (e.g., a skyscraper would look strange in the residential area, and a residential building would be out of place amongst the glass buildings in the financial district), and 2) no mismatches in any buildings. With our system the user can easily achieve these goals by designing three individual pdfs, one for each district. Each pdf limits the grammar's shape space to the subset of valid models for the given district. Our resulting city in Fig. 1 (bottom) was created from a single grammar by sampling from these three pdfs to generate three different sets of buildings: short residential buildings, Hausmannian downtown buildings, and office buildings.

1. Consistency of table legs and leg bases

For tables with different number of legs, we list all consistent combinations of legs and leg bases.

One legged

Two legged

Four legged

Special types

2. Textures

Wooden textures

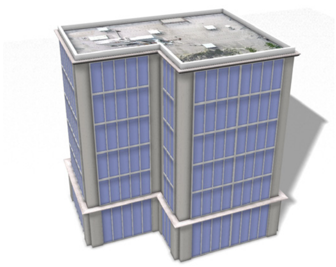
Steel textures

3. Table top styles

Rectangular top

Round top

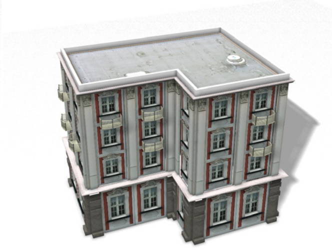
4. Building styles



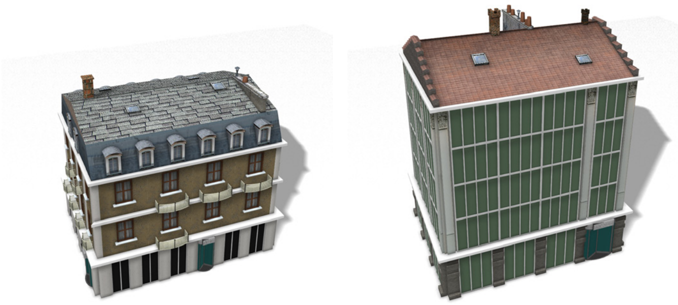
Office

R1

R2



R3



Mixed style

Design scenario F1

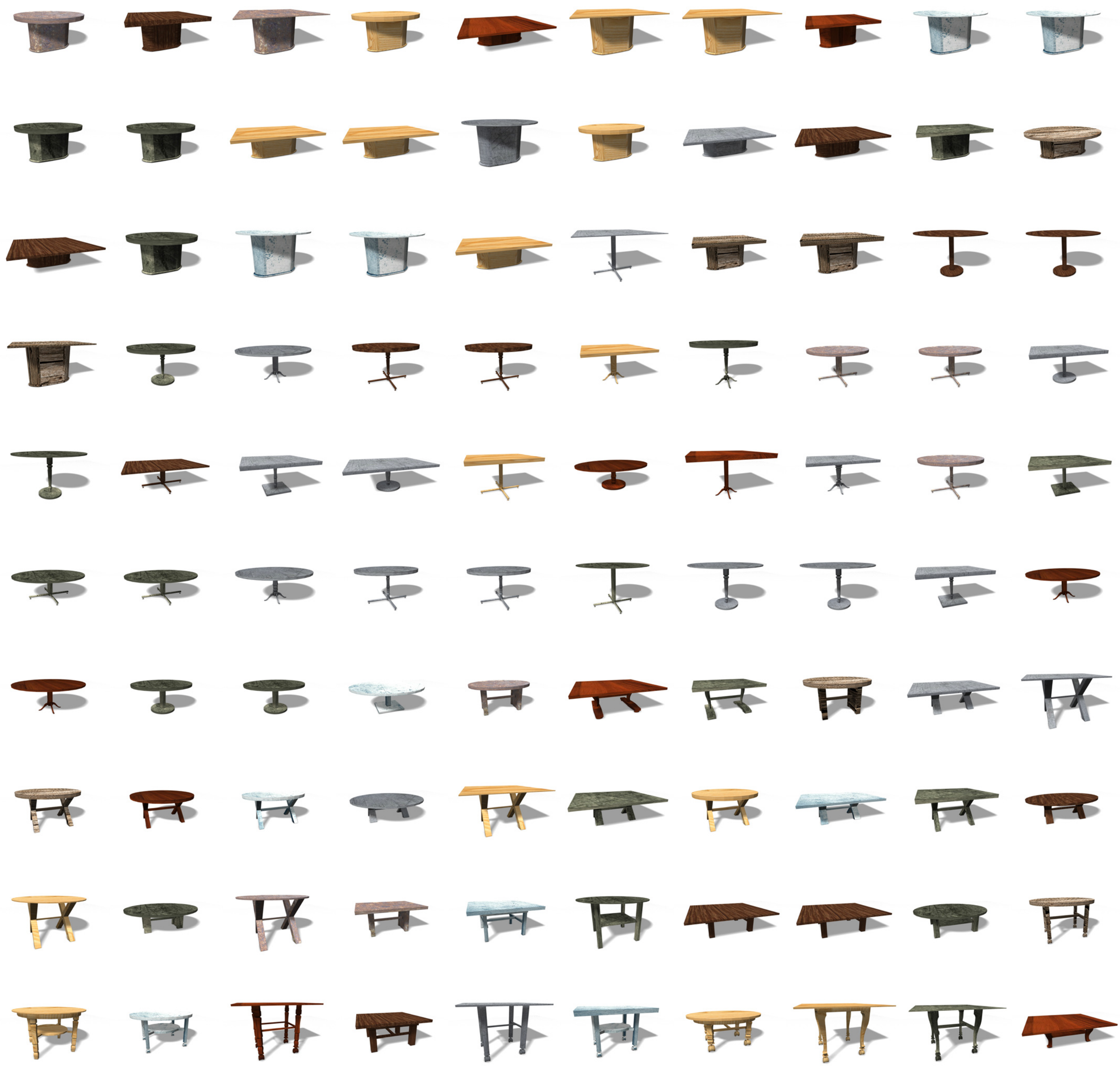
Target preferences:

Valid tables with one leg (70), two legs (20) and four legs (10). Non-valid tables (0).

A table is valid if the number of legs and the leg bases are consistent.

Example preference scores

Correlation w.r.t. ground truth



Design scenario F2

Target preferences:

Wooden tables with light color and round top: 60. Wooden tables with dark color and rectangular top: 40. Others: 0.

Example preference scores

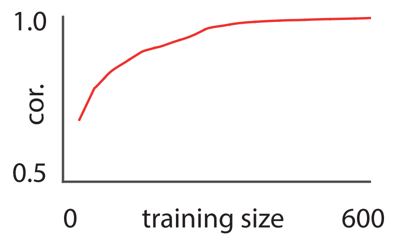
Correlation w.r.t. ground truth

Design scenario F3

Target preferences:
Valid tables with steel textures (70) and wooden textures (30). Non-valid tables (0).
A table is valid if the number of legs and the leg bases are consistent.

Example preference scores

Correlation w.r.t. ground truth

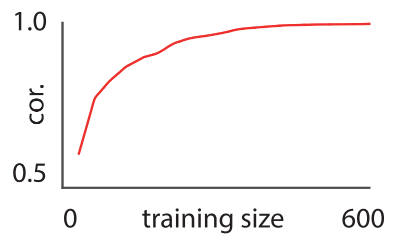


Design scenario F4

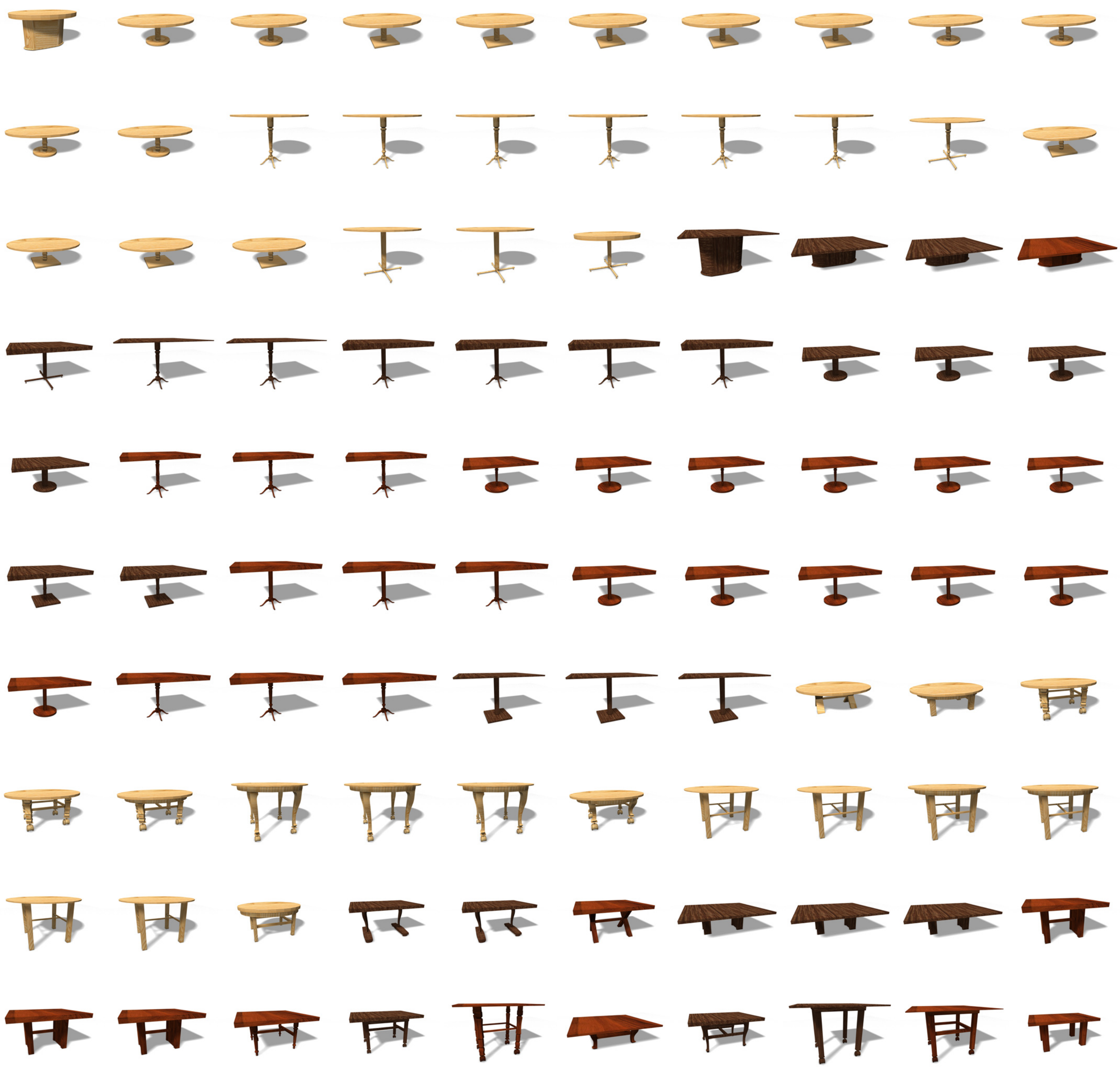
Target preferences:
Valid tables with round top (70) and rectangular top (30). Non-valid tables (0).
A table is valid if the number of legs and the leg bases are consistent.

Example preference scores

Correlation w.r.t. ground truth



Combination F1 x F2

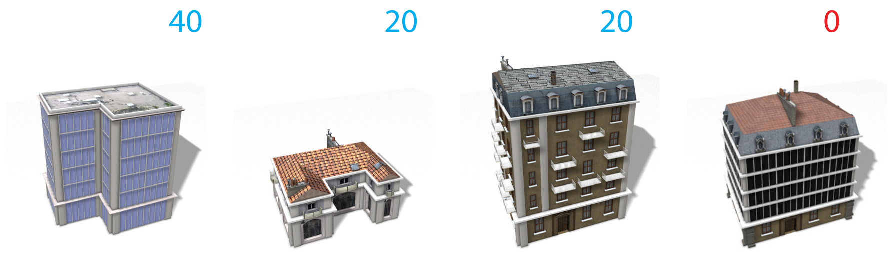


Combination F3 x F4

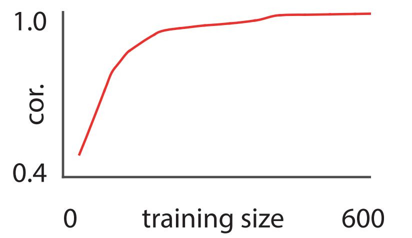
Design scenario B1

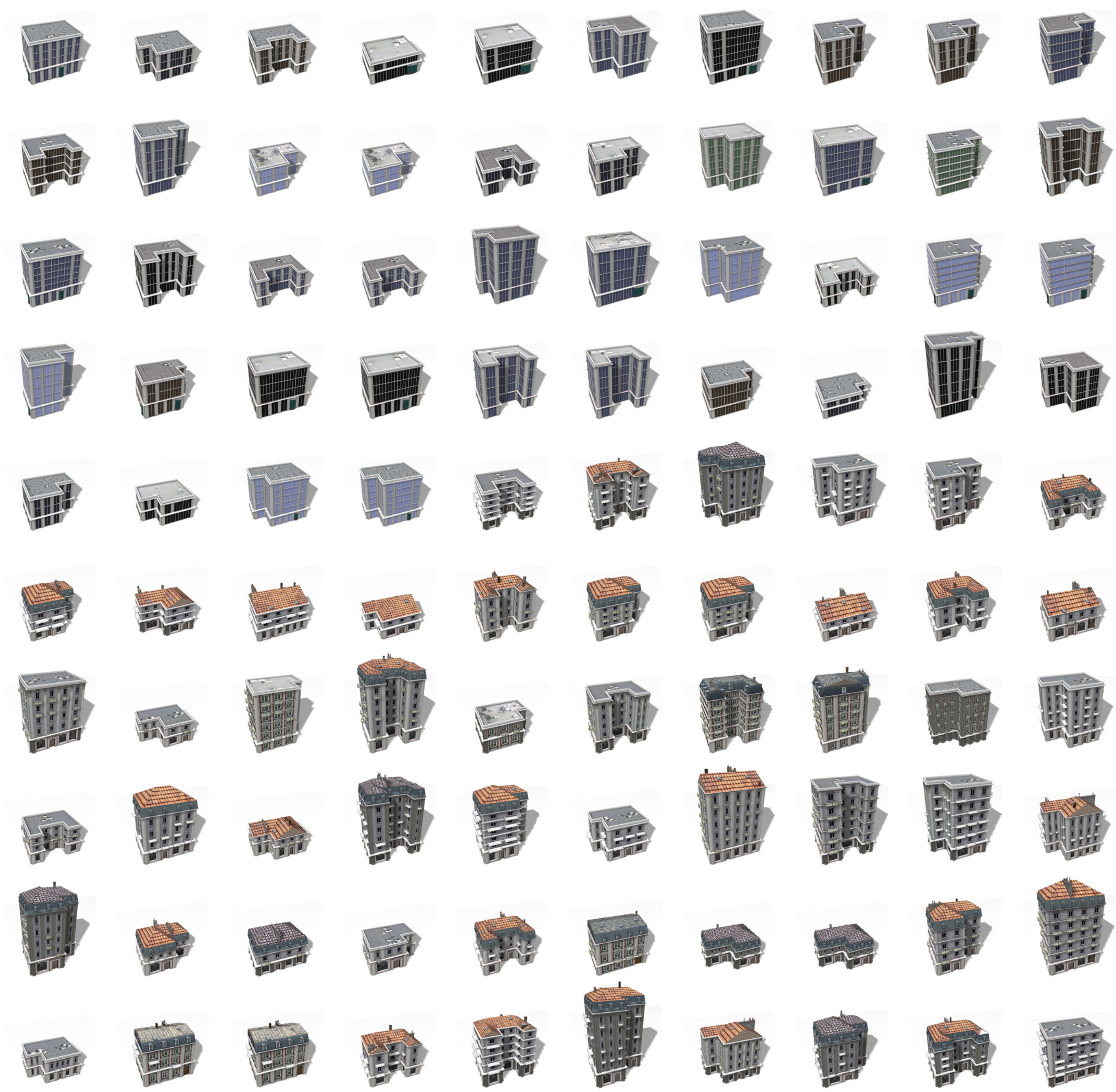
Target preferences:
Office: 40. Building style R1: 20. Building style R2: 20. Building style R3: 20. Mixed style: 0

Example preference scores



Correlation w.r.t. ground truth

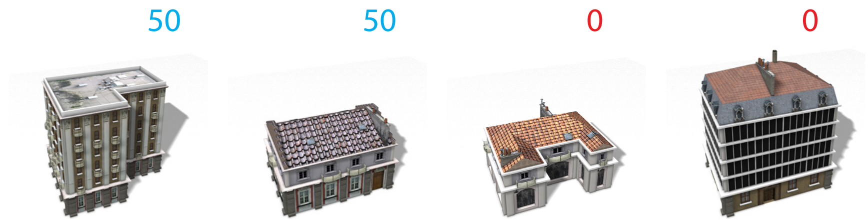




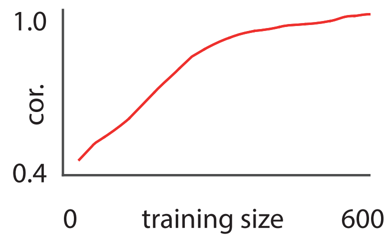
Design scenario B2

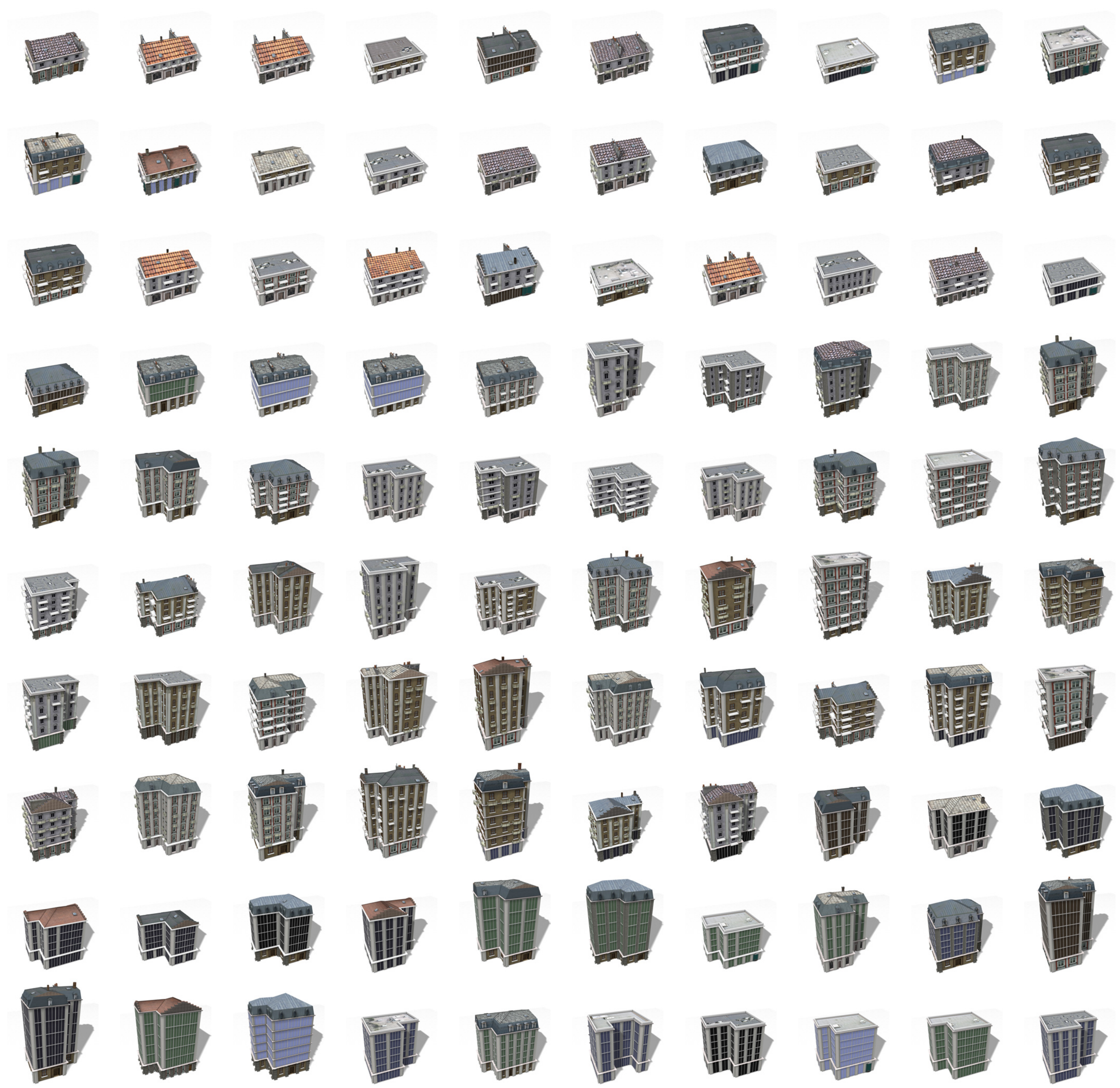
Target preferences:
Tall building (5 - 6 floors) and L-shape: 50.
Short building (2 - 3 floors) and rectangular shape: 50. Others: 0

Example preference scores



Correlation w.r.t. ground truth

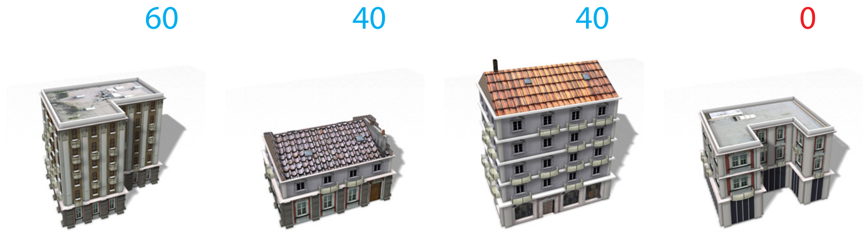




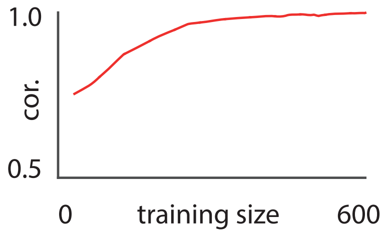
Design scenario B3

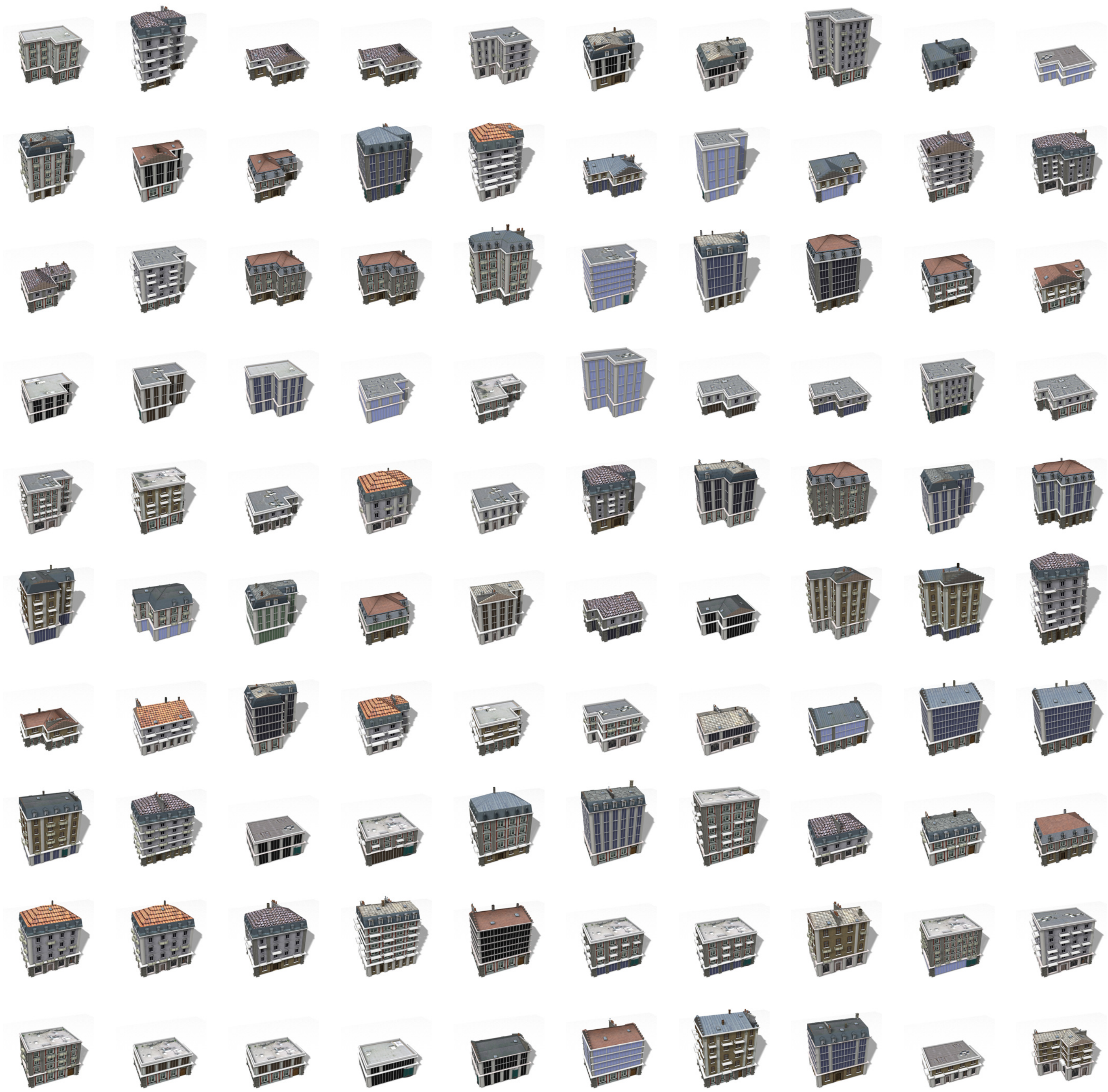
Target preferences:
L-shape: 60. Rectangular shape: 40. Others: 0

Example preference scores



Correlation w.r.t. ground truth

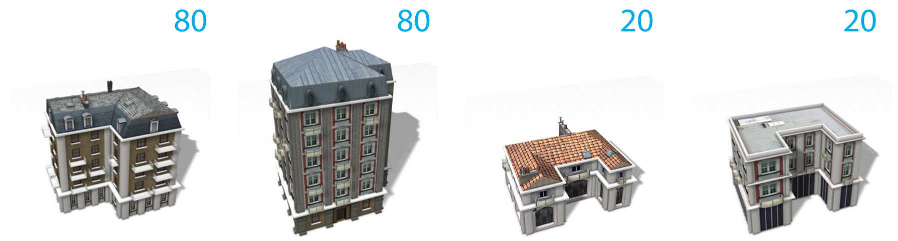




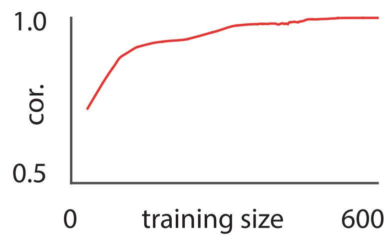
Design scenario B4

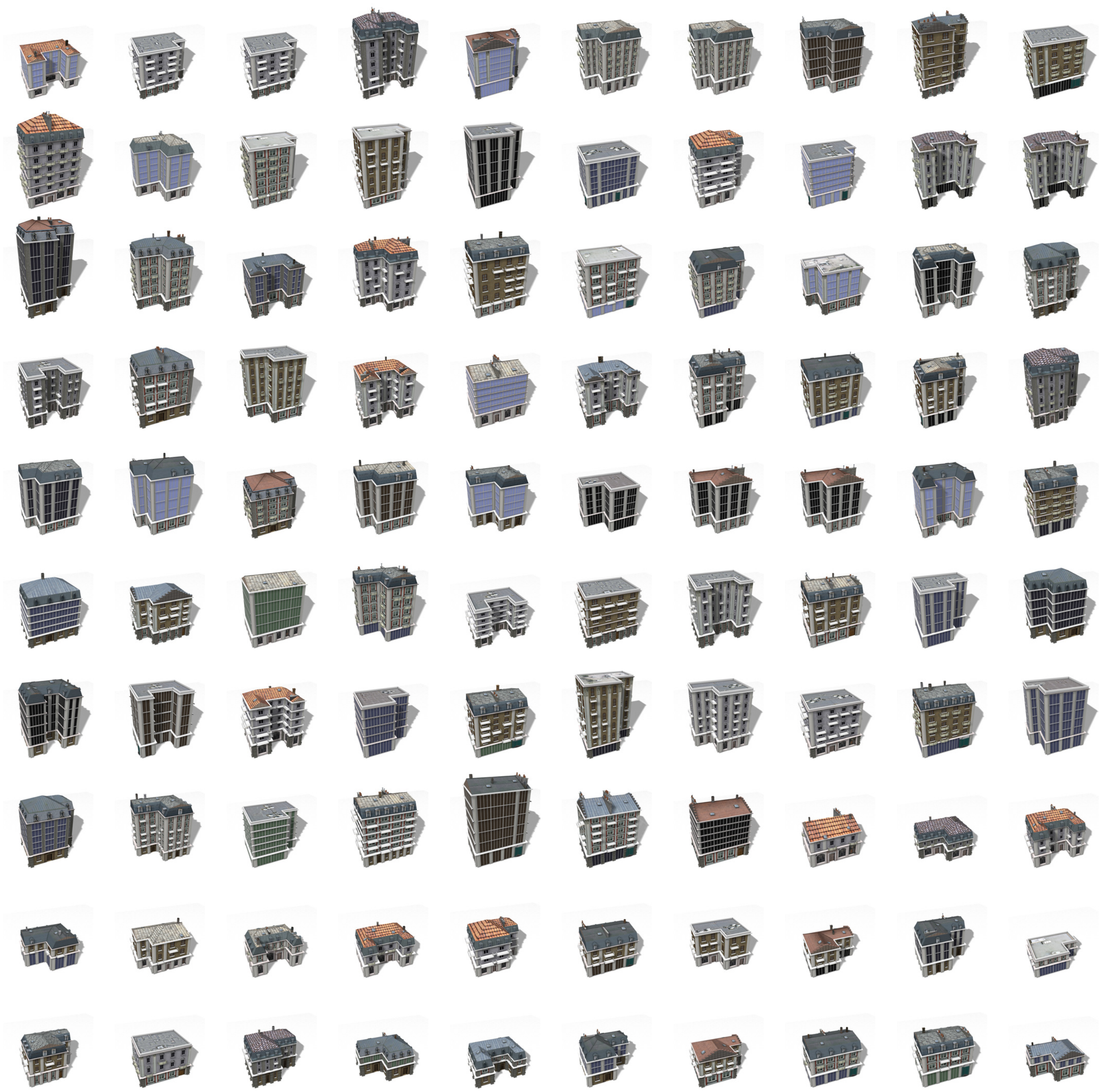
Target preferences:
Tall building (5 - 6 floors): 80. Short building (2 - 3 floors): 20

Example preference scores

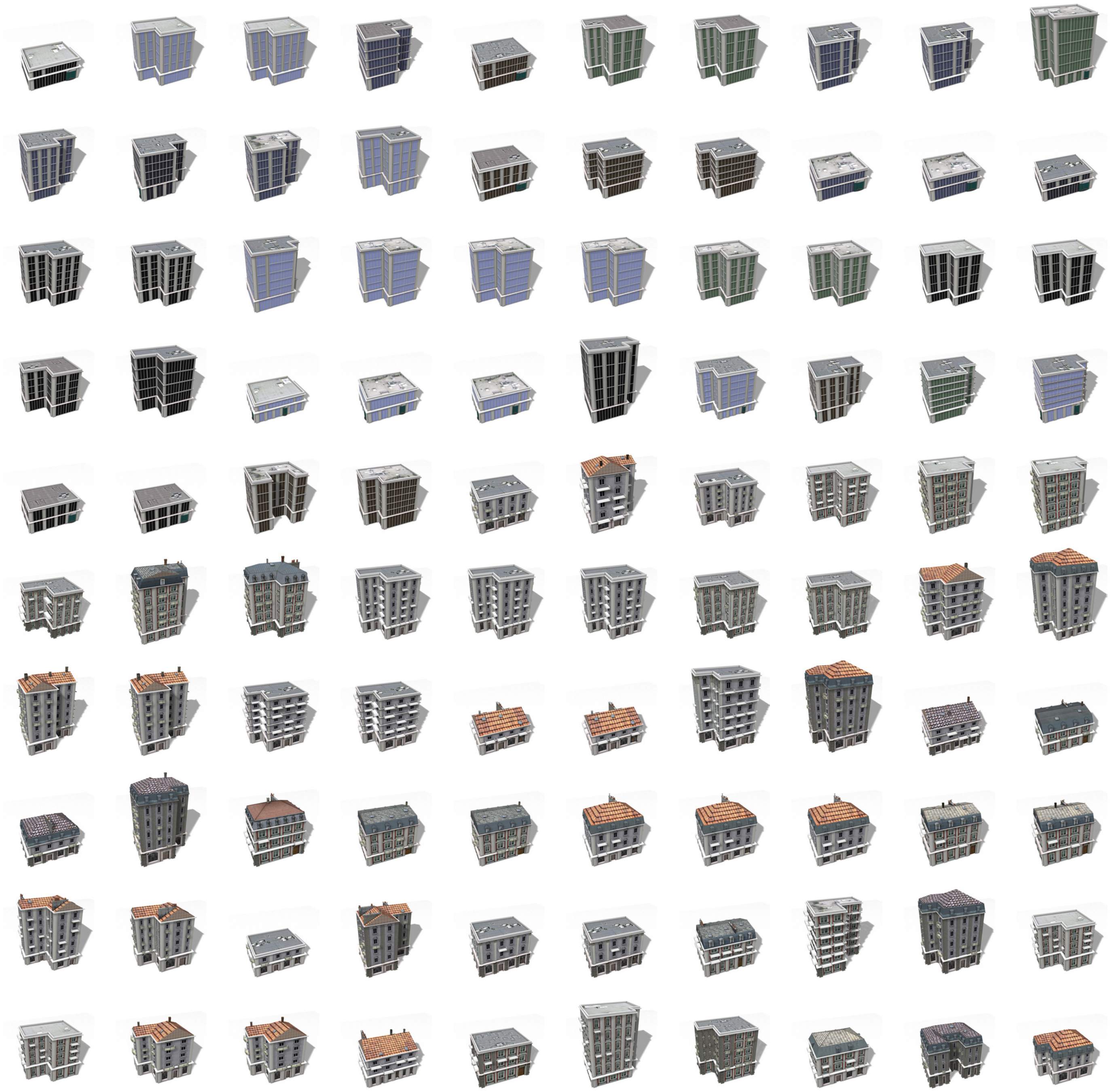


Correlation w.r.t. ground truth

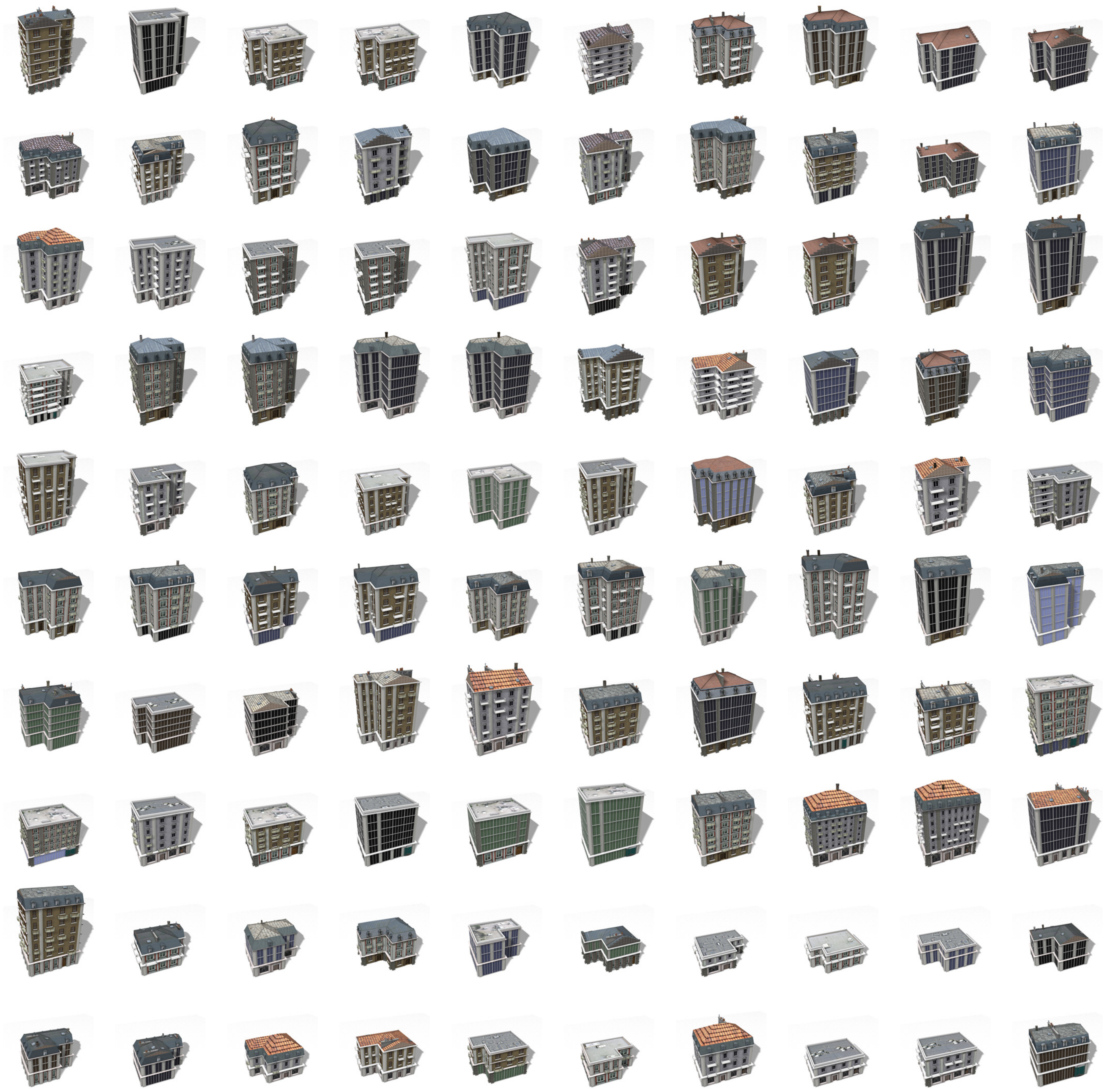




Combination B1 x B2



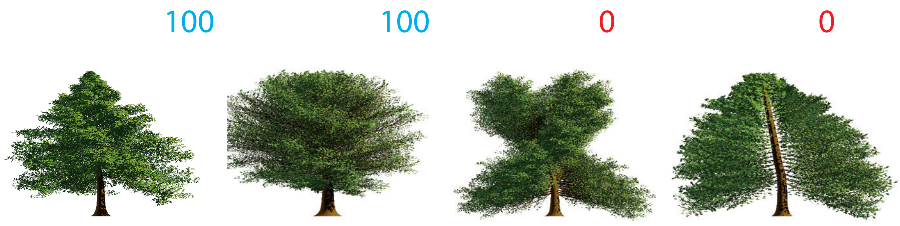
Combination B3 x B4



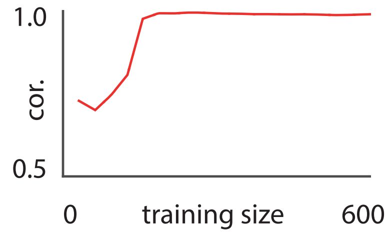
Design scenario T1

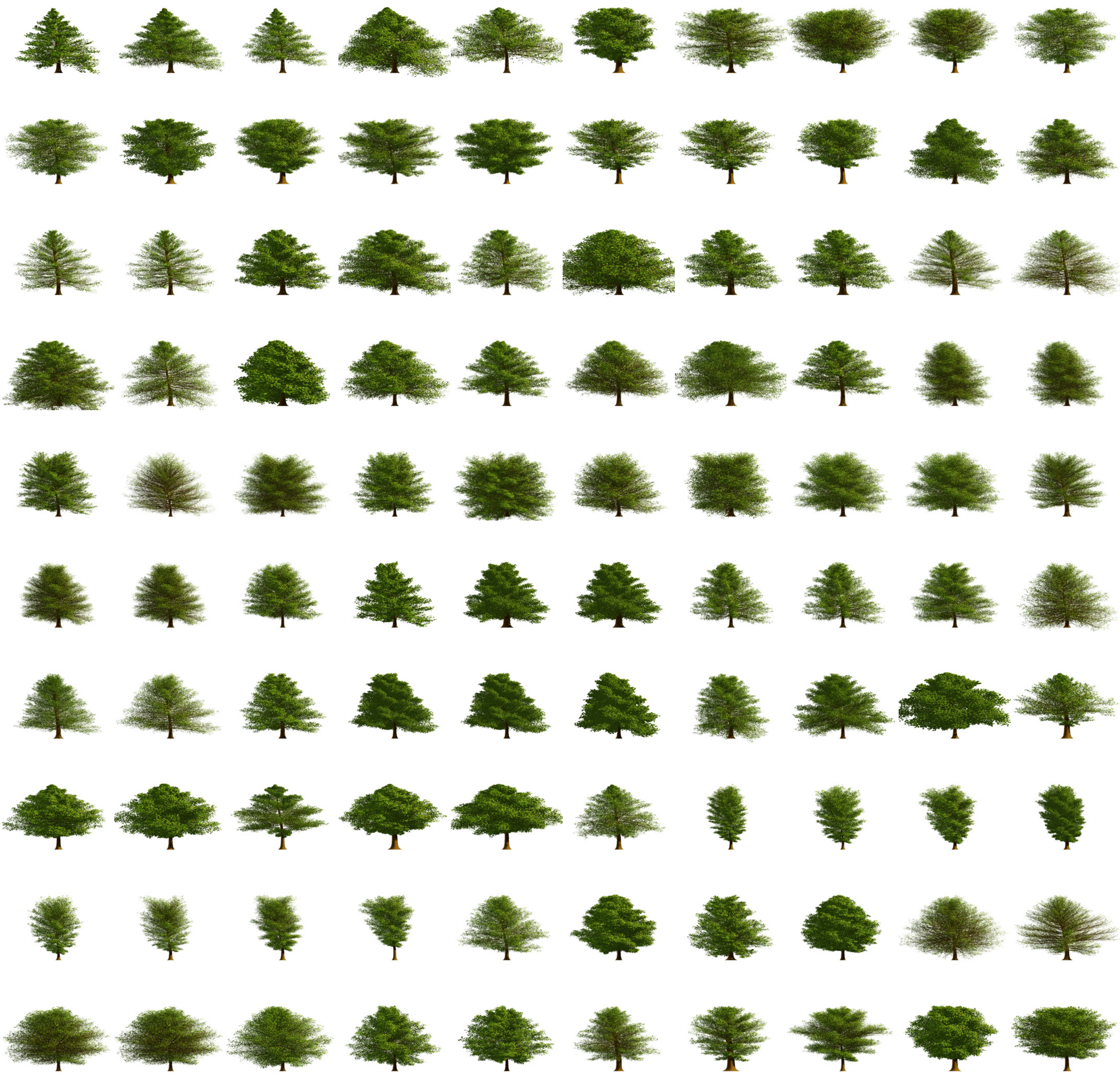
Target preferences:
Plausible tree: 100. Non-plausible tree: 0

Example preference scores



Correlation w.r.t. ground truth



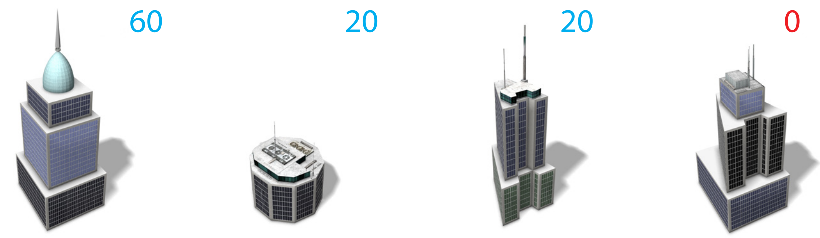


Design scenario S1

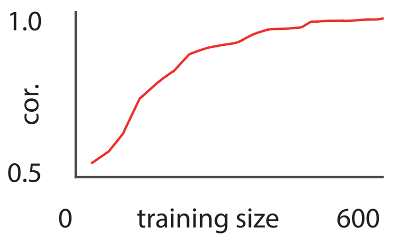
Target preferences:

Skyscrapers with only rectangular block: 60. Skyscrapers with only cylindrical blocks: 20. Skyscrapers with only V-blocks: 20. Mixture of blocks: 0

Example preference scores



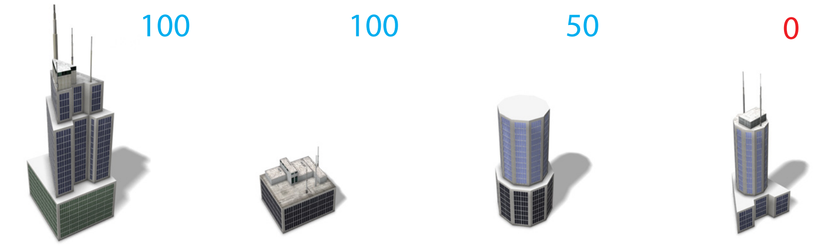
Correlation w.r.t. ground truth



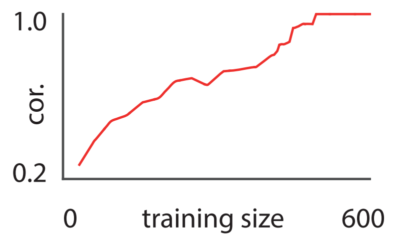
Design scenario S2

Target preferences:
Skyscrapers with rectangular base: 100, cylindrical base: 50, V-base: 0

Example preference scores



Correlation w.r.t. ground truth



Design scenario A1

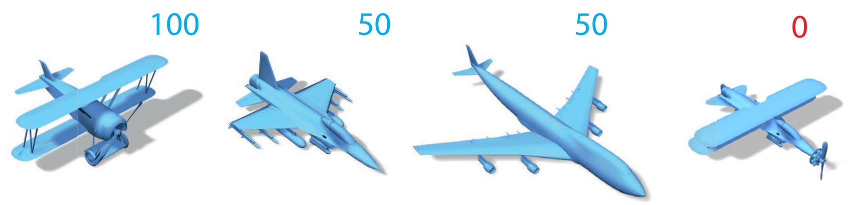
Target preferences:

Old-style airplanes: 100

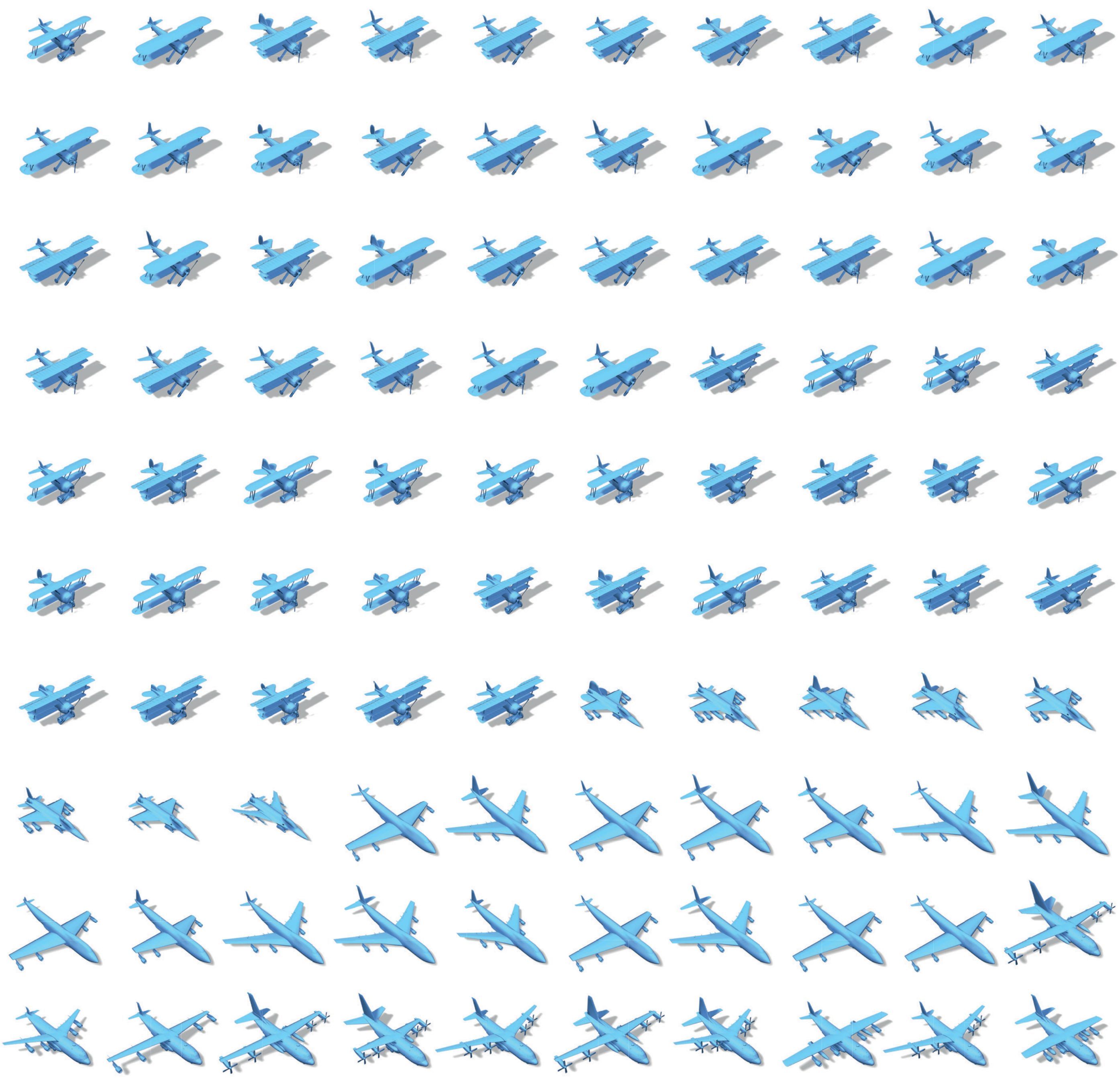
Modern airplanes (commercial, transport airplanes, jet fighter): 50

Airplanes with mismatch components: 0

Example preference scores



Correlation w.r.t. ground truth



Evaluation of grammar generation strategies

In addition to the JS divergence scores shown in the paper, we compute the correlation between the learned pdfs of parameter learning (Method 1) and structure learning (Method 2) with the target pdf for all design scenarios.

	F1	**F2**	**F3**	**F4**	**S1**	**S2**
Method 1	0.72	0.61	0.67	0.66	0.87	0.64
Method 2	0.83	0.85	0.82	0.80	0.99	0.81
	B1	**B2**	**B3**	**B4**	**A1**	
Method 1	0.77	0.42	0.96	0.55	0.70	
Method 2	0.88	0.80	0.96	0.79	0.91	

User study

Objective
The aim of this user study is to evaluate the effectiveness of our interactive framework in designing a target probability density function for a given grammar.

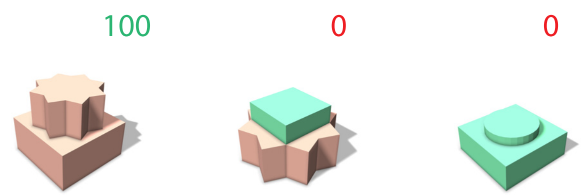
We design two tests with different levels of complexity, to evaluate our framework in the following 3 aspects:
1. Design speed: How fast can a user specify her preferences?
2. Effort: How much effort is needed to design a density function?
3. Accuracy: How accurate is the user in modeling a pdf?

Results
We performed the study on 12 users.
The design speed is measured by time spent per one ranked model. The modeling effort is evaluated by the number of ranked samples. Finally, we use Jensen-Shannon divergence w.r.t the ground truth to evaluate the modeling accuracy.

Demo and Warm-up Test
We first give a demo and ask the user to do a warm-up test to get familiar with the UI. The assistant is available for questions during this warm-up test.

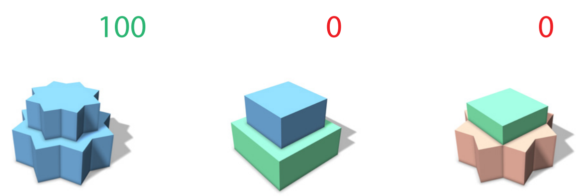
Demo. The assistant shows a demo to train the framework for the following target preference.

Target Preference.
In the toy grammar, all models having the orange color (both on top and on the bottom) are preferred. The remaining models are not wanted.



Warm-up test. The user is asked to train the framework for the following preferences, with support from the assistant.

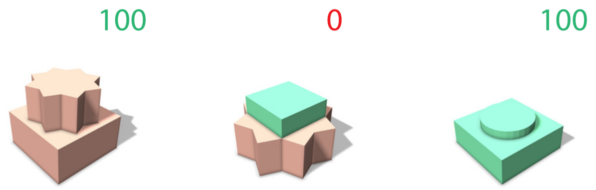
Target Preference.
In the toy grammar, all models having the blue color (both on top and on the bottom) are preferred. The remaining models are not wanted.



Test 1. Toy grammar

The user is asked to give preference scores for models shown in the UI to train the system for the following preference. The test is stopped by the user when she feels the system has correctly learned the target preference.

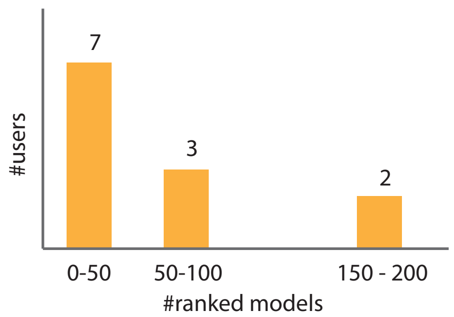
Target Preference.
In the toy grammar, all models having the same colors on top and on the bottom are preferred. The remaining models are not wanted.

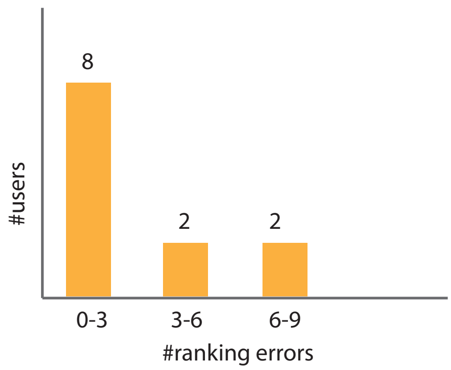


Results

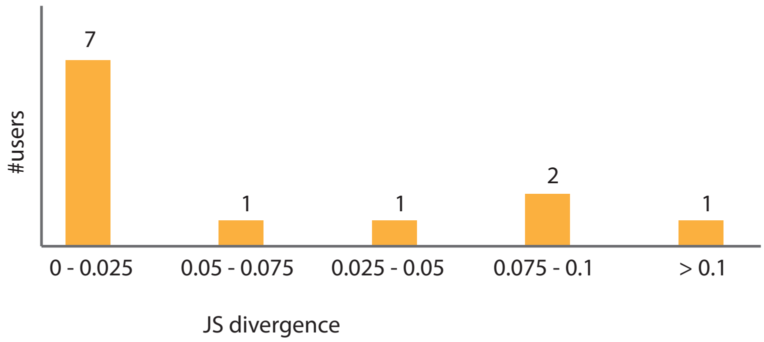
Speed.
All users spent less than 2 seconds per ranked model.
9/12 users needed less than 1 second per model.

Effort





Accuracy

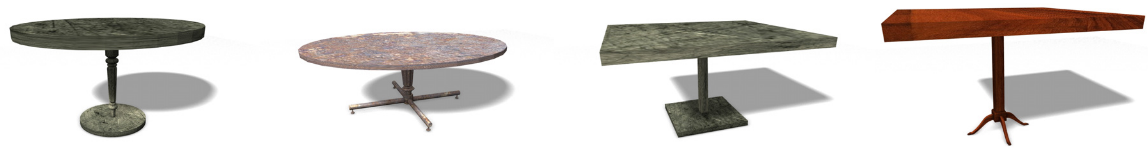


Test 2. Furniture grammar

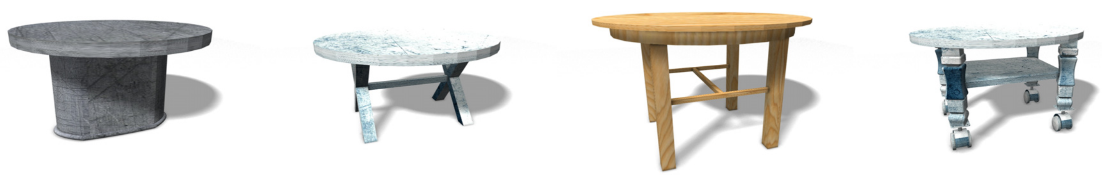
The user is asked to give preference scores for models shown in the UI to train the system for the following preference. The test is stopped by the user when she feels the system has correctly learned the target preference.

Target Preference.
In the furniture grammar, all models where the base of the table is compatible with the number of legs are preferred. The remaining models are not wanted.

Preferred tables



One legged tables with bases



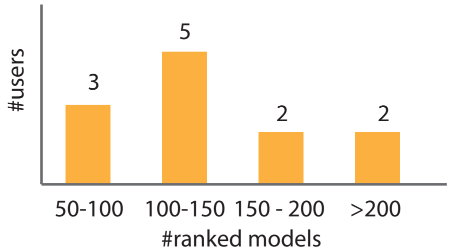
Tables with a big leg Crisscross tables Four legged tables without bases

Unwanted tables

Results

Speed.
All users spent less than 2 seconds per ranked model.
7/12 users needed less than 1 second per model.

Effort



Accuracy