Learning Object Arrangements in 3D Scenes

Nicolás Rondán Zambra

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2018

Abstract

Learning Scene Arrangements in 3D scenes is a fundamental task for the fields of scene understanding and 3D scene generative models. Scene arrangements have been used in the past for building generative models as well as generating semantic parsing graphs for indoor scenes. In this dissertation we study how spatial scene arrangements can be used to build a hierarchical generative model for scenes and the benefits of doing so. Moreover, we study how scene arrangements can be used to find the hierarchical parsing graph of scenes. We propose a baseline hierarchical model and we compare it to a baseline flat model by evaluating how the probability of scenes under both models behaves.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Prof. Chris Williams for his advice, encouragement and guidance along this project. Moreover, I want to thank Paul Henderson for providing code related to his work which was used as a baseline for this work, and for his insightful advice on how to build baseline generative models for scenes. I would also like to express my gratitude to the Chevening Scholarships, the UK government's global scholarship programme, funded by the Foreign and Commonwealth Office (FCO) and partner organisations, for funding my studies at the University of Edinburgh. Finally, I want to thank my family and friends for their support during this programme.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Nicolás Rondán Zambra*)

Table of Contents

Chapter 1

Introduction

In this chapter the main objectives of this dissertation are presented and the motivation behind them is explained. Moreover, the contributions of this research are introduced and the outline of this dissertation is explained.

1.1 Motivation and Objective

Learning scene arrangements is an active research area in the field of machine learning, computer vision and computer graphics (CG). It is well known that humans are able to understand how objects are arranged in scenes, and decide whether a scene looks realistic or not. Moreover, humans are able to recognise whether objects in scenes are related to each other and should be interpreted as groups or not. Representing this knowledge is a task that researchers have sought to address in recent years but one that has still not been completely solved. Being able to model the inherent structure of scenes and represent the knowledge that governs scene arrangements is of particular interest for synthetic 3D generative models, for scene analysis in robotics and semantic analysis of scenes for machine learning purposes. The motivation of this dissertation is to understand how the inherent structure of scenes can be modelled for indoor scenes of houses and analysing which are the benefits of doing so.

Scene understanding and learning scene arrangements can be used to rearrange objects, generate new synthetic scenes or parse unseen scenes. When analysing indoor scenes it is usually the case that objects in a scene belong to functional groups such as tables and chairs, sofas and televisions or beds and side stands. Following this idea, there have been several proposals (Merrell et al., 2011; Xu et al., 2002; Yu et al., 2011) focused on how to rearrange a given set of objects in a scene, by forming groups

in order to make it more realistic. Learning how to rearrange objects can be used as a basis for generating new scenes. However, scene generation implies sampling new objects which usually are grouped together and spatially arrange them in order to create new scenes. Several generative models have been proposed in the recent years (Henderson and Ferrari, 2017; Handa et al., 2016; Wang et al., 2018; Qi et al., 2018), that could be potentially used by interior designers to come up with new ideas of how to rearrange indoor spaces, and moreover, could also be used in video games for synthetic environment generation. Nevertheless, even if these models are able to synthesise new scenes they do not particularly address the problem of parsing real scenes. The question of scene understanding and parsing has been addressed by Liu et al. (2014) by generating semantic parsing trees for 3D scenes. In addition, Yang et al. (2017); Zhao and Zhu (2013) have tackled this problem by analysing images of scenes and creating semantic parsing trees based on functional groups and geometric relationship. Scene parsing and understanding is of particular interest for robotics since autonomous agents have to understand their surroundings and make decisions.

In order to learn how objects are arranged in real scenes it is necessary to have human generated examples of scenes. Gathering large-enough data-sets of 3D scenes, that account for the diversity of objects and layouts present in real-world data used to be prohibitively expensive. This meant that, previously data-sets of 3D scenes used to be composed only by a couple of hundreds of examples as the RGBD data-set introduced by Silberman et al. (2012). Recently, however, the SUNCG data-set was introduced (Song et al., 2017), this is composed by 45,000 human generated 3D scenes of houses and has opened the path for new studies under the field of scene generative models and scene understanding.

Even if several generative models have been proposed in the recent years, it is the case that we need perceptual tests to evaluate these models. This means that we still need human supervision to decide whether a generative model is performing good and new scenes look realistic. Moreover, when addressing the question of how to parse scenes it is the common consensus that hierarchical models should be used and that scenes should be addressed as hierarchical structures. In this dissertation we are motivated by these particular topics. *Is it possible to evaluate the probability of a scene under a model and is there a benefit in using hierarchical models rather than flat structures to parse scenes?* Our hypothesis is that hierarchical models are a better representation of the inherent structure behind scenes and these models extract more valuable information from scenes compared to flat models.

The objective of this research is driven by this question. We will test our hypothesis by proposing two simple generative models, which should be able to evaluate the probability of scenes. One of these models will interpret scenes as flat structures with no spatial relationship between objects and the second one will model scenes as hierarchical structures which allow scene arrangements. Moreover, we will analyse how scene arrangements can be learned based on a previous method used by Henderson and Ferrari (2017) and Fisher et al. (2012), which involves clustering objects using a Gaussian Mixture Model. Furthermore, we propose a strategy to learn the most probable hierarchical interpretation of a scene under our model. Finally, we will use the SUNCG database to train our models and compare the probability of scenes under both models in order to analyse the benefits of using hierarchical models when interpreting indoor scenes.

1.2 Contributions

The contributions of this dissertation are as follows:

1. We present a simple generative model for 3D scenes that interprets scenes as flat structures were the spatial relationship between objects is not modelled and the probability of unseen scenes under the model can be evaluated.

2. We present a second generative model for 3D scenes which accounts for the possibility of arrangements in scenes and multiple hierarchical interpretations. This model is properly normalised and parametric so that given an hierarchical interpretation the probability a scene can be evaluated under the model.

3. Further analysis is done, with focus on how scene arrangements can be learned using Gaussian Mixture Models.

4. We propose a strategy for learning the hierarchical parsing graph of a scene under our hierarchical model.

5. The benefits of using a hierarchical model are analysed in depth by comparing this to a flat model. Moreover, further analysis is done on how the probability of scenes changes as groups of objects are clustered in our hierarchical model.

1.3 Dissertation Outline

In Chapter 2, the background knowledge required to understand this dissertation project is explained. In Chapter 3, we present the recent and more classic related works in the field of scene understanding and 3D scenes generative models. Following this in Chapter 4, the resources needed in order to implement our generative models and develop the experiments done are explained. Once this is done in Chapter 5, the definition for the generative models that are going to be used to analyse the benefits of using a hierarchical model are presented and, in Chapter 6, we expose the experiments designed to test our hypothesis and their results. Finally, in Chapter 7 we give our conclusions on the experiment's results and provide further discussion on our models selection and design as well as proposing future research works that could follow this dissertation.

Chapter 2

Background

In this chapter we explain the background knowledge that is necessary in order to understand the methods proposed in Chapter 5 and the experiments done in Chapter 6.

2.1 The Infinite Gaussian Mixture Model

The infinite Gaussian Mixture model was proposed by Rasmussen (2000) as an extension to the finite Gaussian Mixture model with k defined elements defined by equation:

$$\mathcal{P}(y|\mu_1,\cdots,\mu_k,\Sigma_1,\cdots,\Sigma_k,\pi_1,\cdots,\pi_k) = \sum_{j=1}^{k} \pi_i \mathcal{N}(\mu_j,\Sigma_j). \tag{2.1}$$

The finite Gaussian Mixture Model can be used in practice to approximate the distribution of any multidimensional data-set y composed by $\{y_1,\cdots,y_n\}$ points by a combination of k different multivariate Gaussian distributions with parameters μ_j and covariance Σ_j using mixing coefficients π_j. Moreover, this model can be used to cluster points in a data-set given the assumption that each data-point y_i in the data-set is assigned an indicator c_i which indicates which of the k clusters $\mathcal{N}(\mu_j,\Sigma_j)$ generated the data-point. In the finite mixture model the mixing coefficients π_j need to be positive and sum to one. This model is useful for modelling unknown distributions, however, one of the main drawbacks of using a finite mixture model is that parameter k needs to be set beforehand in order to fit the model. Furthermore, the model fitting is usually done by Expectation Maximisation which can converge to local minima.

In order to sort out this drawbacks in Rasmussen (2000) it is proposed to use a Bayesian inference framework to approximate the distribution of y using a Gaussian Mixture Model with infinite elements ($k \to \infty$). In practice for a finite data-set a finite number of clusters k will be used. The main advantages of using this approach is that

the number of clusters k needed to represent the distribution of the data, is learned during the process of fitting the model and does not has to be set by the user. This is quite useful if there is no prior knowledge about the possible number of clusters in the data-set.

The model is fitted in the following way. To begin with, the author sets up a Bayesian inference procedure to learn the parameters of a Mixture model with k constant elements. In order to do this, priors are defined on the parameters μ_j, Σ_j and π_j given constant k . Having set the priors, it is possible to derive the posterior distribution for the parameters μ_j, Σ_j, π_j and the indicators c_i using the likelihood of the data given by Equation 2.1. Once the posteriors are defined, it is to possible sample from these using different sampling techniques as Gibbs Sampling or Adaptive Rejection Sampling (ARS) (Gilks and Wild, 1992). Having set the framework, the author derives the formulation for the priors given the limit $k \to \infty$ and by doing this it derives the posterior distribution of the parameters of the model μ_j, Σ_j and π_j, and hidden variables c_i when $k \to \infty$. Once this is done, by sampling from the posterior it is possible to define a finite approximation of the infinite mixture model that approximates the distribution of the data from data-set y.

Using this technique it is possible to approximate any distribution as a Gaussian Mixture Model with no prior definition of k the number of clusters. Nevertheless, in practice it is necessary to set an hyper-parameter α which defines the prior on the concentration of the clusters. As stated by the author, using and infinite mixture model has the advantage that the number of clusters is automatically learned. Moreover, by using MCMC sampling techniques as Gibbs sampling local minima convergence is mitigated when fitting the model. Finally, using this technique simplifies the problem of having to work with finite models with unknown number of components as proposed by Richardson and Green (1997).

2.2 Tukey's Fences

In data analysis Tukey's fences are a methodology for detecting outliers in the data samples. This method is based on the box plots to represent the distribution of data proposed by Tukey (1977). Box plots and Tukey's fences are based on the Interquartile Range (IQR) concept. The IQR is a statistical measure defined by the difference

between the third quartile and the first quartile of the data:

$$IQR = Q_3 - Q_1. \tag{2.2}$$

The IQR can be used to detect outliers and Tukey proposed to build a fence using the IQR given by the following formula:

$$[Q1 - k(Q3 - Q1), Q_3 + k(Q3 - Q1)] = [Q1 - kIQR, Q_3 + kIQR], \tag{2.3}$$

and label each data-point outside this interval as an outlier in the data-set. It was proposed to use $k = 1.5$, however, other values are still valid for this formula.

Chapter 3

Related Work

In this chapter, we review the related work regarding learning scene arrangements and rearranging objects in scenes, generative models for 3D scenes, and scene understanding by building semantic parsing graphs. This related work section is based on our previous research carried out for the Informatics Research Proposal (IPP) (Rondan, 2018). Firstly, we review the most relevant related work for object rearrangement, secondly, we present several of the recent proposals regarding generative models, and finally, the relevant work on scene understanding and parsing is described.

3.1 Rearranging Objects in Scenes

The first approach to solve the problem of learning scene arrangements was to generate algorithms that were able to rearrange a given set of objects in a scene. In this section we describe the relevant related work related to this task.

3.1.1 An interactive framework for object rearrangement

Interior design guidelines were used by Merrell et al. (2011) in order to generate a system that could rearrange objects in scenes. This was achieved by taking into account the fact that objects in scenes must respect a functional criteria, and modelled the functional relationships between objects based on this. Moreover, they modelled the clearance space that needs to be respected between objects, in order for these to be accessible and modelled the geometric pairwise relationships between them. Furthermore, they incorporated a visual criteria to their model, which stated objects groups should respect alignments and have focal points were group's objects are faced. All

these concepts were integrated into a density function and arrangements were sampled from this distribution using a Markov chain Monte Carlo sampler. For a particular set of objects, the algorithm will offer several sets of arrangements from which the user can choose which one to use. Moreover, the system could also work in an unassisted way; suggesting only one arrangement to users, however, the assisted mode was preferred when evaluated by professional interior designers. Although this method can output realistic scenes, given the nature of the probability distribution, it can only be sampled using Markov chain Monte Carlo techniques. Therefore, it is not possible to evaluate the probability of scenes under this model.

3.1.2 Rearranging objects as an optimisation problem

The problem of rearranging objects can also be addressed as an optimisation problem, which was studied by Yu et al. (2011). Using similar concepts as Merrell et al. (2011), Yu et al. (2011) suggested the use of interior design guidelines, spatial relationships, pairwise relationships and hierarchical relationships between objects to build a cost function. Spatial relationships modelled the distance and relative orientation of an object to its nearest wall. Pairwise relationships modelled the relationship between objects that usually appear together, such as televisions and sofas or tables and chairs. Moreover, the hierarchical relationships modelled the relations between objects that are placed one over another, such as a candelabrum and table. These relationships were learned from positive examples, and in particular, pairwise relationships were labelled by users in the training data. The cost function was optimised using simulated annealing (Kirkpatrick, 1984) and Metropolis-Hastings state-search step (Metropolis et al., 1953; Hastings, 1970). This is one of the first reported algorithms to fully automatise the problem of rearranging objects in scenes. Finally, the method was evaluated using perceptual test and the results suggest, as stated by the authors, that the participants in the experiment do not clearly prefer human arranged scenes to their automatically optimised scenes.

3.2 Generative Models

Following the work done for the rearrangement of objects in scenes, several generative models for 3D scenes were presented. In this section, we describe the most relevant models introduced until the moment of this work.

3.2.1 Scene generation based on user examples

In Fisher et al. (2012), new synthetic scenes are generated based on a minimal set of examples provided by users. In order to achieve this, contextual categories of objects that can be interchangeable in scenes based on neighbourhood similarity were defined. Moreover, the occurrence of objects in scenes was modelled using a Bayesian Network and a probability function was defined to account for parent-child relationships between objects. The spatial relationship between objects was modelled using a Gaussian Mixture Model learned from the example scenes. Furthermore, the surface placement of objects was modelled using a probability distribution, taking into account that objects, can lie on the floor, be placed on walls, or be supported by other objects. They augment the user provided examples using an existing database of scenes, and used these to learn the different distributions' parameters. Finally, scenes were synthesised by sampling from the Bayesian network and the different probability density functions. This approach was used to synthesise scenes of small environments like desks with objects above or dining tables with objects rather than complete rooms scenes modelling bigger objects' arrangements. Nevertheless, when evaluated by perceptual tests the results showed that at least 80% of the newly generated scenes were not distinguishable from human generated ones.

3.2.2 A framework for hierarchical scene generation

Recently Handa et al. (2016) proposed a framework for generating new scenes based on the work done by Merrell et al. (2011), extending this approach in order to generate scenes hierarchically. They proposed to address the scene generation task as an energy minimisation problem. In order to achieve this, an energy function was defined that accounted for pairwise relationships between objects similarly to Merrell et al. (2011), ensured visibility between objects and avoided bounding box intersections. Moreover, the energy function modelled the position and relative rotation of objects to walls. The coefficients for these terms were learned from prior scenes. For the energy function optimisation they proposed to solve it hierarchically using simulated annealing. In order to generate a new scene, they initialised this with all its objects placed in the middle, and optimised the energy function by iterations selecting random objects at each iteration. Moreover, they optimised the function hierarchically, this means they grouped objects and optimised the position of groups at each iteration. Using hierarchical structures to generate scenes gave more realistic results than incorporating

independent objects in the energy function. The authors only evaluate the proposed framework qualitatively, nevertheless, it is relevant as it introduces an approach on how to optimise objects' positions in scenes hierarchically.

3.2.3 Probabilistic model for 3D scene sampling

The scene generation problem can also be targeted using probabilistic models. In a Henderson and Ferrari (2017) a data driven method was introduced which models the occurrence of objects in scenes using different probability distributions. Their model is trained using the newly introduced data-set of scenes SUNCG (Song et al., 2017). In this method the authors generate new scenes by modelling the occurrence of objects and the spatial distribution of object conditioned to different room types. Furthermore, objects are clustered together to form arrangements and these arrangements are used in the generative process of creating new scenes. The process of generating new scenes is described in detail in this section since some ideas from this probabilistic approach are going to be used as a basis for our research on objects arrangements and scene probability evaluation.

To begin with, the authors define the probability distributions for objects classes occurrence given a particular room type. To do this, they account for five object categories: big objects placed on the floor (furniture), small objects (small objects placed on furniture, e.g. books, laptops), wall objects, ceiling objects and rugs. Having done this, the probability of object counts for each object class is defined based on Poisson distributions for big counts and categorical distributions for small counts.

To position objects in rooms, it was proposed to use a deformable cell grid, where each object will take a particular cell in the grid but the cell size is not defined. This was done taking into account that an object's size is given by their associated CAD model. Therefore, cell sizes were defined once the objects' CAD models were sampled. In order to select a particular cell for an object, the authors defined the probability of objects being placed next to a wall or in the middle of rooms. Once the cells were defined the CAD models for the objects were sampled. Finally, the cell sizes were selected to fit the sampled CAD models and padding was added sampling from an isotropic 4D Gaussian restricted to positive values.

Placing independent objects in new scenes still lacks the component that in realistic scenes objects tend to cluster together in functional groups. In order to fulfil this requirement, the authors search for patterns in the training data using the following

procedure. Given a K-tuple of objects classes that can be possible in the training data. The authors designate for each K-tuple, a base class object, and search for the occurrences of this K-tuple in the training data. They extract all the relative locations of objects around the base object for and fit an Infinite Gaussian Mixture Model (Rasmussen, 2000). The mixture model is fitted using Bayesian variational inference and each Gaussian cluster is assigned a diagonal covariance matrix. For each K-tuple of classes, this method learns the different configurations the K-tuple can take. For example, they are able to cluster the different configurations in which desks and chairs group and extract the different configurations in which beds and side stands arrange. Once objects patterns have been found, in order to sample arrangements, they save the found occurrences from the training scenes and sample from these in order to have a realistic results when generating scenes. Objects patterns are placed following the same process as independent objects.

This generative model was evaluated using perceptual tests and participants were asked to select between human generated scenes and scenes generated by the model. Participants preferred human generated scenes 70% of the time, and synthetic scenes generated by the model 30% of the time. The perfect output for this kind of tests is to get the users unable to distinguish between human generates scenes and synthetic ones, with a 50%-50% result.

3.2.4 A deep learning approach for scene generation

It is well known that deep learning models need big data-sets to be trained. With the introduction of the SUNCG data-set containing more than 250,000 scenes of rooms, the possibility for deep generative models was opened. In Wang et al. (2018) a deep generative model for synthesising new scenes was introduced. The model generates new scenes using a pipeline of three components: *Continue?*, *CategoryLocation* and *InstanceOrientation*. These components are described in the following paragraphs.

The first component is a multilayer perceptron that outputs the probability of adding a new object to a scene given the current objects and high-level features from the top-down view of the scene. These high-level features are extracted using a deep convolutional network trained with the top-down views of scenes augmented with semantic features as object category per pixel, depth per pixel, and walls, doors and windows information. In this way objects are sampled conditional to the previous objects in the scene. If a new object needs to be added, the component *CategoryLocation* makes the

decision of which object category to add and where in the scene it should be placed. The structure of *CategoryLocation* is similar to *Continue?*, however, this component outputs the probability distribution of categories given the position and the scene. In order to get these distributions, the authors discretise the scene space using a grid of cells and for each cell they output the probability distribution of categories. Given the distribution for each cell it is possible to sample from the joint distribution of categories and location. The final component *InstanceOrientation* outputs the orientation of the new object in a scene, however, in order to do this a 3D model for the selected category needs to be sampled. For this reason, the authors model the relations between CAD models, and cluster them in collections. The object's 3D model is sampled from the same collection as previously placed objects. Having done this *InstanceOrientation* outputs the probability of orientations given the sampled position.

This method is compared to other baselines which generate objects independently and has better results in perceptual studies. Moreover, when compared to human generated scenes, human generated scenes are still preferred since the generative model has some failure modes were inconsistent objects are placed (e.g. tables with no chairs or to many side stands with beds). Nevertheless, this is one of the first approaches that uses deep leaning models to extract features from real scenes and use these features in a generative process that creates new synthetic scenes.

3.3 Scene Parsing

In addition to generative models another relevant scene analysis task relates to scene understanding and generating semantic parsing graphs. In this section we review the method for generating scene graphs that is most related to our research.

3.3.1 Scene parsing with probabilistic grammars

In order to address the problem of scene parsing, Liu et al. (2014) proposed using probabilistic grammars to learn the grammar rules that rule the formation of scenes. Scenes of bedrooms with labelled objects and annotated scene graphs were used to achieve this goal. The selected parsing graphs described scenes by forming hierarchical structures with semantic groups of objects as sleeping area or storage area. The method for learning the grammars is the following. Firstly, the authors defined a grammar of the

form:

$$G =< \mathbf{L}, \mathbf{R}, \mathbf{P} >, \tag{3.1}$$

where **L** denotes for the labels of the objects and semantic groups existing in the scene, **R** denotes the rules that define the grammar and **P** includes the probabilistic parameters included in these rules. Secondly, the authors proposed to learn the grammar rules and associated probabilities from the training data by setting **L** using the occurrences of objects and groups in scenes, building **R** by analysing dependencies in groups and setting **P** based on the data-set statistics.

The problem of parsing a new scene is formulated as a dynamic programming problem for belief propagation in a pruned search space, as stated by the authors. This means they prune the search space of possible configurations and propose candidate configurations which are incorporated into an energy function, which is optimised to find the most probable configuration. Their energy function is aimed to approximate the maximum a posteriori probability (MAP) (Bishop, 2006) estimation of a parsing graph for a scene.

Finally, in order to evaluate this method, generated scene graphs were compared to ground truth data annotated by humans, and it was possible to get almost 100% accuracy for small data-sets of scenes.

Chapter 4

Resources

In this chapter we describe the different Software and Hardware resources needed to implement the models defined in the following chapter and the experiments explained in Chapter 6

4.1 Software & Hardware

The models and experiments described in this project were implemented using python 2.7.15 in addition to scientific computing libraries as numpy and scipy. Moreover, scikit-learn (Pedregosa et al., 2011) and code from Henderson and Ferrari (2017) which will be explained in section 4.3 were used to train the models defined in Chapter 5. Finally, matplotlib (Hunter, 2007) was used to generate the plots presented in Chapter 6.

All the models were trained using an Intel i5 CPU at 2.4GHz, no GPUs are needed for this project. The average training and evaluating time of the final experiments is 3 hours on the testing set.

4.2 SUNCG Dataset

The SUNCG data-set is a set of 45,000 of human-generated CG scenes of houses and apartments made available by Song et al. (2017). This is the largest available data-set of indoor scenes at the moment and includes more than 250.000 scenes of rooms in houses corresponding to different room types. These are the following: Bedroom, Living Room, Toilet, Kitchen, Bathroom, Room, Dining Room, Garage, Office, Hallway, Hall, Child Room, Balcony, Storage, Guest Room, Lobby, Entryway, Terrace, Logia,

Boiler Room, Aeration, Passenger Elevator and Freight Elevator. From the room types available the ones with the largest amount of samples are listed in Table 4.1.

Room Type	**Room Count**
Bedroom	34,655
Living Room	29,014
Toilet	27,230
Kitchen	23,720

Table 4.1: Most prevalent room type counts

Each of the 45,000 scenes is labelled and a ”json” file is provided that states the types of rooms included in the scene, which objects are included in each room, assigns a CAD model for each object and defines the bounding box of space occupied by the object in the scene. Moreover, meta-data is provided that maps each CAD model to a corresponding object class; object classes extracted from the data-set for this project are listed in Appendix A. Furthermore, a toolbox[1] is provided for visualising scenes. The data-set can be downloaded by signing a licence agreement on the available on the authors web page[2]. Samples from houses included in SUNCG can be seen in Figure 4.1 and samples from rooms showing objects with their corresponding bounding box can be seen in Figure 4.2.

[1] http://suncg.cs.princeton.edu/, accessed 13 Aug 2018
[2] https://github.com/shurans/SUNCGtoolbox, accessed 13 Aug 2018

Figure 4.1: Sample Scenes from SUNCG

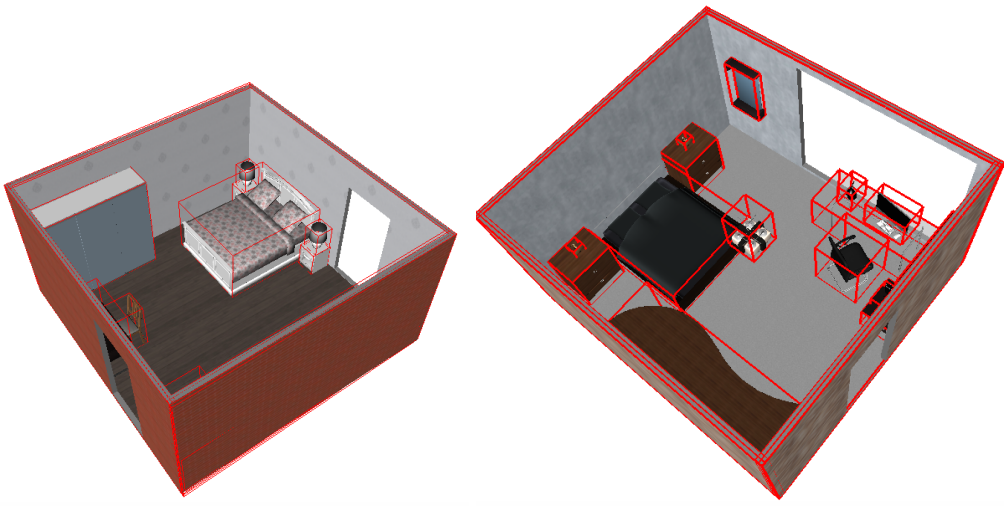


Figure 4.2: Sample rooms with objects bounding boxes

4.3 Learning Objects Arrangements in Rooms

In recent years several research groups have been working on the topic of learning how objects arrange in scenes and proposing generative models as the ones explained in the related work chapter. One important task in order to generate new scenes is to learn object arrangement patterns. Fisher et al. (2012) and Henderson and Ferrari (2017) have tried modelling objects groupings as a Gaussian Mixture model that maps relative locations of objects with respect of a reference object. The first one worked under a limited amount of training scenes, generating artificial configurations by adding jitter to real configurations in order to learn the mixture model as explained in the paper. The second one used the whole SUNCG data-set to cluster existing configurations of object patterns. We will work under this idea of modelling scene arrangements as mixture models because it can learn objects groupings and different grouping configurations in an unsupervised way. Moreover, using a Gaussian mixture model it is possible to define the probability of a particular set of objects to be considered a group. In particular we will expand on the idea presented by Henderson and Ferrari (2017), since it was already implemented using the SUNCG data-set and code from the original paper was provided by the authors.

As explained in Section 3.2.3 by analysing every K-tuple of object classes existing in the training data, extracting all the occurrences of this particular K-tuple, selecting a centre object class and measuring the relative locations of objects in the K-tuple around the centre object it is possible to learn how objects cluster. However, given the amount of scenes and object classes available in SUNCG this approach is feasible as it is not computationally practical. In practice some selected tuples of object classes that are expected to cluster together can be used to learn how these arrange in different configurations. Some examples of tuples that were analysed are (*single_bed*, *stand*) or (*stand*, *double_bed*, *stand*) or (*desk*, *chair*). Full list of class tuples used for finding patterns is listed in Appendix B.

Given a particular set of classes the provided code by Henderson and Ferrari is able to load the data-set, find all occurrences of the class tuple and extract all the relative locations between the centre object and remaining objects. Having extracted this information it fits an infinite Gaussian Mixture Model using a Dirichlet process as proposed by Rasmussen (2000) and explained in Section 2.1. In practice the mixture model is fitted using Bayesian inference using scikit-learn libraries. For a finite amount of data this results in a finite approximation of the infinite mixture model. Finally, for each

set of classes it is able to learn different configurations in which these classes cluster together. Given the finite approximation with k elements of the Infinite Gaussian mixture

$$\mathcal{P}(x) = \sum_{i=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k), \tag{4.1}$$

for each cluster $\mathcal{N}(\mu_k, \sigma_k)$ the code evaluates the area where satellite objects group. In order to evaluate this area the following decision was made by Henderson and Ferrari. Given the covariance matrix Σ_k the diagonal elements are the variances of the x and y positions of each satellite object. For each satellite object O_i given the variances $\sigma^2_{x_i}$ and $\sigma^2_{y_i}$ the area in which objects cluster is computed with the following formula:

$$area = 2\sqrt{2}\sigma_{x_i} 2\sqrt{2}\sigma_{y_i}. \tag{4.2}$$

If for a given Gaussian cluster the mentioned areas is smaller than a certain threshold ($2m^2$) for each satellite object and if the full cluster contains a minimum amount of occurrences (200). Then, the parameters μ_k and Σ_k are extracted and this cluster is saved as a particular configuration of the selected classes. The extracted configurations will be referred as motifs; in Figure 4.3 different motifs for (*stand*, *single_bed*) tuple are shown and in Figure 4.4 different motifs of (*double_bed dresser*) are presented. In each of these figures different occurrences of these particular motifs are plotted together. Finally, some real occurrences of motifs are shown in Figure 4.5. This code

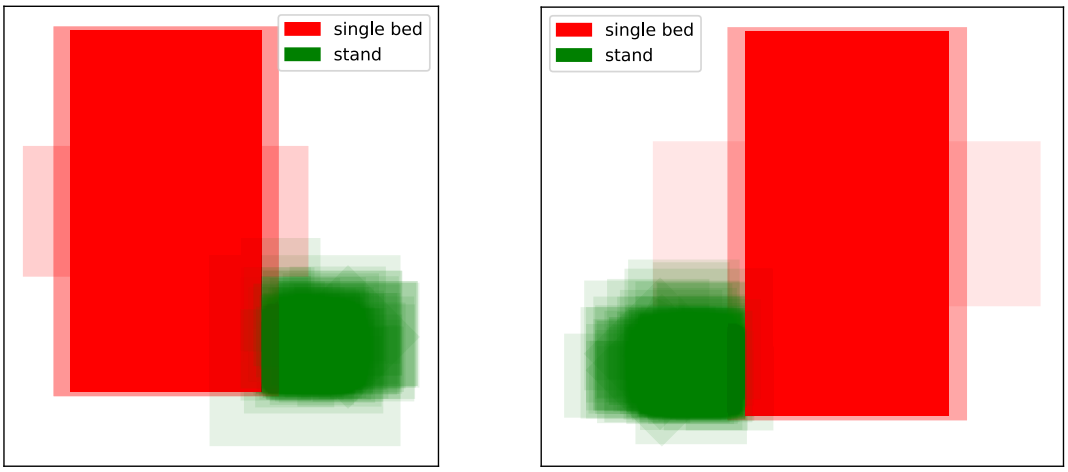


Figure 4.3: Sample motifs for *stand*, *single_bed*

will be used as a resource for this project and we will expand on the idea of using a Gaussian Mixture Model to model the occurrence of objects as groups.

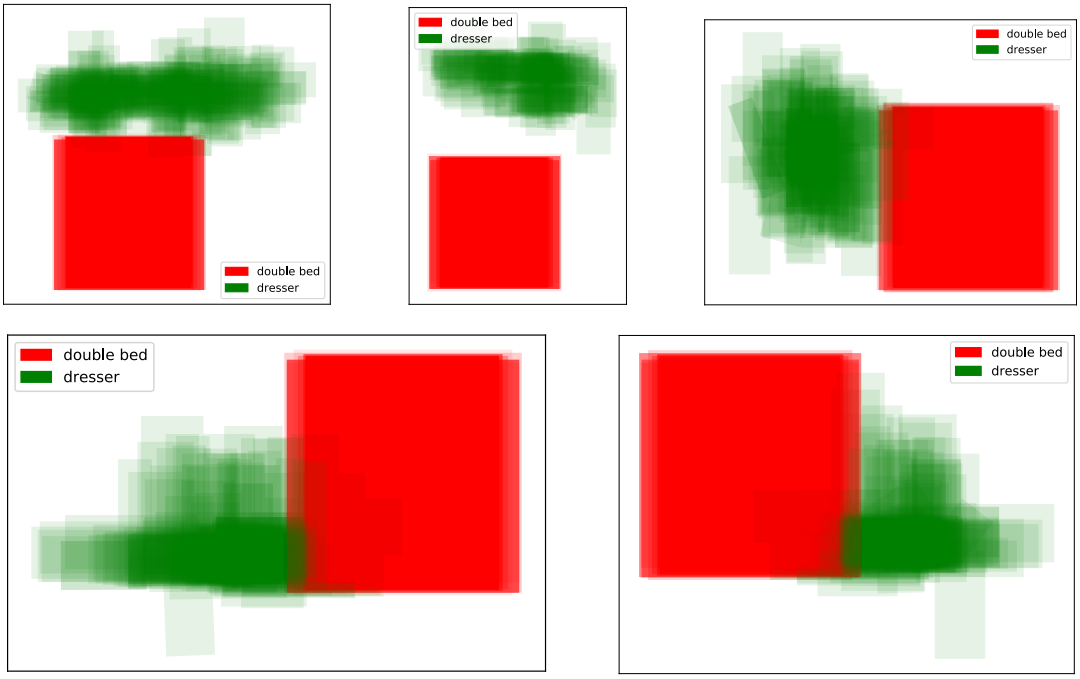
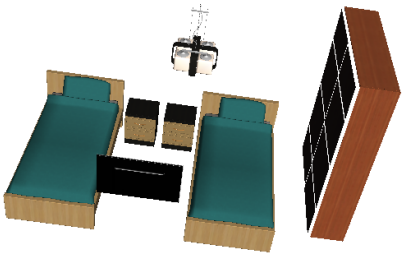
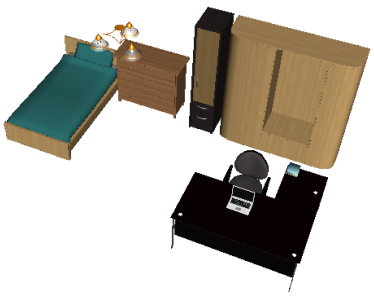


Figure 4.4: Sample motifs for *double_bed*, *dresser*



(a) Single Bed - Stand



(b) Single Bed - Dresser / Desk - Chair

(c) Sofa - Tv Stand

Figure 4.5: Example real occurrences of motifs in scenes

Chapter 5

Scene Representation Models

In this chapter we will describe the scene representation models that will be used to test the hypothesis that hierarchical models are better for learning the inherent structure of scenes than flat models. For this purpose, we will define a baseline model $\mathcal{M}_1$ which does not know about spatial relationships between scene objects and a hierarchical model $\mathcal{M}_2$ which is able to learn object arrangements and treat scenes as hierarchical structures. These models are based on previous definitions proposed in our IPP work (Rondan, 2018), however, improvements have been added to these previous definitions. Given these two models we will define the evaluation metrics which will be used to compare them.

5.1 Scope of Models

Both models will be data-driven and trained with extracted scenes from SUNCG data-set (Song et al., 2017). This data-set is the most complete data-set of 3D indoor scenes available at the moment and has been used by several research papers related to 3D scene understanding recently (Qi et al., 2018; Wang et al., 2018). SUNCG accounts for different type of objects which can be placed on walls, above other objects or on the ceiling. For the scope of this project we will simplify the problem to only work with furniture objects placed in the floor, objects classes used for this project are listed in Appendix A. This will allow us to treat a scene as a 2D structure, nevertheless these models can be extended to 3D with some simple additions. Moreover, as explained in Section 4.2, the data-set is composed of a set of houses with different room types in each house, however, for this project we will select one room type and train the model on extracted scenes from this particular room type. The room type selection is

explained in Section 6.2. Furthermore, rooms can have different shapes and numbers of walls, as a simplification only scenes with four walls will be considered. Finally, the model will be constrained to a particular room size range. This restrictions are not included in the equations defining the models but will be considered when cleaning the data and explaining the experiments in Chapter 6.

Each scene in the data-set is composed by a set of objects which have a particular object class, a position, an orientation as well as an associated CAD model which defines the size of the object. In order for the models to be complete we will need to define a distribution that models the dimensions of rooms, a distribution that models the occurrence of objects in scenes, a spatial model that defines the location and orientation of objects in the scene space and in the case of the hierarchical model we will need to define a grouping model. Moreover, since each model could be able to generate scenes that are not valid we will define a normalisation constant for each model, which defines the ratio of valid scenes to total scenes generated. The need of this normalisation constant is given by the fact that the model will generate a considerable amount of invalid scenes by definition. If we want to evaluate the real probability of valid scenes we need to normalise the model knowing the probability of a scene to be valid.

In order to define both models $\mathcal{M}_1$ and $\mathcal{M}_2$, we will make some basic definitions and will take some basic assumptions. The final objective of this chapter is to be able to define the probability of a real scene given each one of the models. For this purpose, we will treat each scene as a set of objects $\mathcal{S}$:

$$\mathcal{S} = \{O_1, O_2, \cdots, O_N\}, \tag{5.1}$$

where each object will be defined by the following set:

$$O_i = \{c_i, x_i, y_i\}, \tag{5.2}$$

where c_i is the object class and $\{x_i, y_i\}$ defines the object's position in the scene. For our modelling we will exclude rotations and CAD Models from the object representation. Nevertheless, this could be added to the models in the future. In this section we will define models $\mathcal{M}_1$ and $\mathcal{M}_2$, explain how to generate new scenes from each of the models and give a formulation for $\mathcal{P}(\mathcal{S}|\mathcal{M}_1)$ and $\mathcal{P}(\mathcal{S}|\mathcal{M}_2)$.

5.2 Baseline Model $\mathcal{M}_1$

A baseline model for representing scenes is to treat all objects belonging to a scene as being spatially independent objects. In order to test that hierarchical models are able

to learn the inherent structure of scenes, we will compare such models to this baseline flat model. This model will be referred as $\mathcal{M}_1$ throughout the document.

Given a scene $\mathcal{S}$, for model $\mathcal{M}_1$ we will define that the only possible hierarchical configuration for the objects in the scene is given by the flat parsing tree that is shown in Figure 5.1.

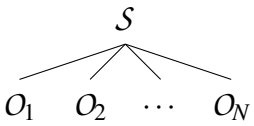


Figure 5.1: Independent model scene graph

As explained in the introduction of the chapter for the formulation of the model $\mathcal{M}_1$ it is necessary to define an object occurrence model, a spatial model that models objects position independently given it's object class, a scene size model and a normalisation constant $\mathcal{Z}_1$. This will allow to sample new synthetic scenes and evaluate the probability of exiting scenes. For model $\mathcal{M}_1$ for each object in a new scene, CAD models will be sampled uniformly from the available models corresponding to the object class.

5.2.1 Occurrence model

The occurrence model specifies the probability of a particular number of instances of an object class to be present in a scene. The standard distribution to model the frequency of an event is the Poisson distribution. For this reason, we modelled the probability of occurrence of and object class c_k with a Poisson distribution with parameter λ_{c_k}. All the λ_{c_k} parameters for each class distribution will be learned from the training data. The occurrence model will define the probability of an object with class c_k of having n_{c_k} instances in a scene. A similar strategy for object count was used by Henderson and Ferrari (2017) to sample the total number of objects for a scene.

5.2.1.1 Sampling process

For a new scene given an object class c_i the cardinality of this class can be sampled from:

$$n_{c_i} \sim Poisson(\lambda_{c_i}), \tag{5.3}$$

where parameter λ_{c_i} has been inferred from the training set of scenes. In practice given the distribution of objects in the training set, most classes will have zero instances in a new scene.

5.2.1.2 Probability evaluation

Given a scene $\mathcal{S}$ set as expressed in Equation 5.1, in order to evaluate the probability of occurrence of these particular objects under model $\mathcal{M}_1$ we need to evaluate the probability of the number of occurrences per class. Given all possible object classes c_k and n_{c_k} the cardinality of each class in the scene, the occurrence probability of $\mathcal{S}$ under $\mathcal{M}_1$ is:

$$\mathcal{P}_{occ}(\mathcal{S}|\mathcal{M}_1) = \prod_k Poisson(n_{c_k}; \lambda_{c_k}). \tag{5.4}$$

5.2.2 Spatial model

This model will define the spatial distribution of objects under $\mathcal{M}_1$. One of the simplest ways to model the spatial distribution is to divide each scene in a discrete grid of cells. For this reason, scenes will be divided into a 2D grid of $N \times M$ cells as seen in Figure 5.2, where objects will be approximated to be placed in the centre of each cell. Using the training data a categorical distribution can be learned in order to represent the probability of an object with class c_i to be placed in a grid cell g_{hj}. A similar grid division strategy was used by Wang et al. (2018) for finding the probability of object categories to be placed in the different cells. Nevertheless, they used deep learning models to learn the probability of object classes given the position. In this model we will use a categorical distribution.

g_{00}		$\cdots$		g_{0M}
$\vdots$		g_{hj}		$\vdots$
g_{N0}		$\cdots$		g_{NM}

Figure 5.2: Discretised scene in cell grid

The cell grid size parameters N, M will be selected using a grid parameter search strategy under a validation set of scenes and testing which grid cell division has better results.

5.2.2.1 Sampling process

For each object O_i and given its class c_i a cell g_{hj} will be sampled from:

$$g_{hj} \sim Categorical(c_i) \tag{5.5}$$

After sampling a cell the object position x_i, y_i will be set in the centre of the sampled cell. Finally, Orientation θ_i will be sampled uniformly from a discrete orientations $[0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}]$.

5.2.2.2 Probability evaluation

Given a scene $\mathcal{S}$ and given the area for each cell in the grid $\mathcal{A}_c$ the spatial probability $\mathcal{P}_{loc}$ of an object being placed in a particular cell g_{hj} as:

$$\mathcal{P}_{loc}(O_i|\mathcal{M}_1) = Categorical(\{x_i, y_i\} \in g_{hj}|c_i)/\mathcal{A}_c. \tag{5.6}$$

Taking this into account the final spatial probability of a scene S is:

$$\mathcal{P}_{loc}(\mathcal{S}|\mathcal{M}_1) = \prod_{i=1}^{N}(Categorical(\{x_i, y_i\} \in g_{hj}|c_i)/\mathcal{A}_c). \tag{5.7}$$

5.2.3 Room size model

Given the training data we will learn the distribution of a scene sizes. Each scene will be defined by dimensions X, Y referring to the width and length of rooms. Since X, Y dimensions can be interpreted as continuous variables and modelling the height of scenes its not interesting for the purpose of this project, we will model X, Y dimensions as a 2D Gaussian with mean μ_D and covariance Σ_D. Parameters μ_D and Σ_D are inferred from the training data.

5.2.3.1 Sampling process

When generating new scenes dimensions X, Y will be sampled from the defined distribution and samples that are outside the boundaries of the training data scenes dimensions will be rejected. This will avoid sampling extremely small scenes or extremely big ones.

$$(X, Y) \sim \mathcal{N}(\mu_D, \Sigma_D). \tag{5.8}$$

5.2.3.2 Probability evaluation

The probability of a scene having particular dimension can be evaluated using a truncated Gaussian, however for the purposes of this project this will not be interesting. Since both models $\mathcal{M}_1$ and $\mathcal{M}_2$ will be used to compare the probability of a scene. And the probability of the room size will be the same under both models.

5.2.4 Normalisation constant

Given that model $\mathcal{M}_1$ will be able to generate more scenes that those which are possible in the real world we will define a normalisation constant to evaluate the real probability of scenes under this model. The definition of the normalisation constant $\mathcal{Z}_1$ is the following:

$$\mathcal{Z}_1 = \frac{\text{\#Valid Scenes}}{\text{\#Total Possible scenes}}. \tag{5.9}$$

The value of this constant will be computed experimentally by sampling scenes from the model and finding the ratio of valid scenes to total scenes like in a rejection sampling strategy. A scene is considered invalid when there is occlusion between objects, or intersection between objects and walls. If there are no intersections a new sampled scene is considered valid.

5.2.5 Scene sampling

The process of sampling a new scene is the following:

- Sample scene dimensions (X, Y).
- For each possible class sample the number of instances n_{c_k} in the scene.
- For each object instance O_i sample a CAD model uniformly from the available CAD models in the data-set.
- For each object instance O_i sample the object location and rotation in the scene (x_i, y_i, θ_i)

5.2.6 Final probability

For model $\mathcal{M}_1$ given a scene $\mathcal{S}$ as described by set in Equation 5.1 the probability of the scene under model $\mathcal{M}_1$ is:

$$\mathcal{P}(\mathcal{S}|\mathcal{M}_1) = \frac{1}{\mathcal{Z}_1}\mathcal{P}_{occ}(\mathcal{S}|\mathcal{M}_1))\mathcal{P}_{loc}(\mathcal{S}|\mathcal{M}_1), \tag{5.10}$$

$$\mathcal{P}(\mathcal{S}|\mathcal{M}_1) = \frac{1}{\mathcal{Z}_1}\prod_j Poisson(c_j;\lambda_{c_j})\prod_i Categorical(x_i,y_i|c_i)/\mathcal{A}_c, \tag{5.11}$$

where index j iterates over all possible classes and index i over existing scene objects.

5.3 Hierarchical Model $\mathcal{M}_2$

In order to test the hypothesis of this project a hierarchical model was designed . This model needed to be able to generate new synthetic scenes as well as evaluating the probability of real world scenes. The designed model will be referred to $\mathcal{M}_2$ throughout this document. In contrast to the baseline model $\mathcal{M}_1$, in model $\mathcal{M}_2$ for each scene $\mathcal{S}$ given by Equation 5.1 we will consider different hierarchical interpretations. Each hierarchical interpretation will be referred as a configuration $\mathcal{C}$ and will be described by a parsing tree composed of as a set of independent objects O_i and object clusters $\mathcal{Y}_k$. The structure of a parsing tree for a configuration $\mathcal{C}$ of $\mathcal{S}$ under this model can be seen in Figure 5.3. For this project we will only work with one level of hierarchy. Each possible possible configuration $\mathcal{C}$ of $\mathcal{S}$ will be defined by a set:

$$\mathcal{C} = \{O_1, \mathcal{Y}_1, \cdots, , \mathcal{Y}_K, O_N\}, \tag{5.12}$$

where each group $\mathcal{Y}_K$ is a group of objects belonging to $\mathcal{S}$ which are clustered together:

$$\mathcal{Y}_k = \{O_{k1}, \cdots, O_{kN_k}\} \quad : \quad O_{ki} \in \mathcal{S} \quad \forall i, \tag{5.13}$$

and each $\mathcal{Y}_k$ will be described by m_k a *motif* class, $\{x_k, y_k\}$ which define the position of the group in the scene and $\mathcal{P}_{m_k}$ a probability distribution that models the spatial distribution of the objects $\{O_{k1}, \cdots, O_{kN_k}\}$ given $\{x_k, y_k\}$. This probability distribution will be defined in section 5.3.3.

Each *motif* class represents how a particular set of objects classes can cluster together in different shapes as explained in section 4.3. As an example an object class tuple (single bed, stand) can take different valid spatial configurations that will be called *motifs* and the will belong to the pattern (single bed, stand).

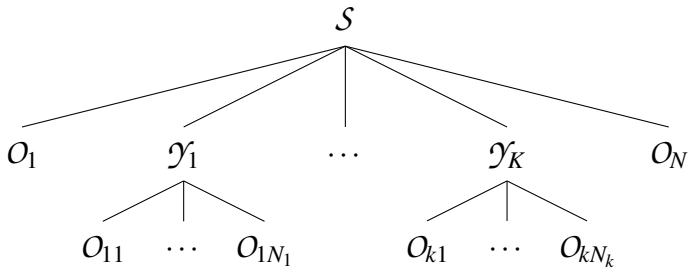


Figure 5.3: Scene graph with object arrangements for a configuration $\mathcal{C}$

For the formulation of the model $\mathcal{M}_2$ as in model $\mathcal{M}_1$ we will define an occurrence model, a spatial model and a room size model, as well as a normalisation constant. Moreover, since this model accounts for objects groupings it is necessary to model the spatial distributions of objects within the different grouping clusters, this will be called the grouping model. Finally, for each independent object a CAD model will be sampled uniformly from the available models corresponding to the object class, and for a grouping $\mathcal{Y}_k$ CAD models corresponding to the grouping objects $\{O_{k1}, \cdots, O_{kN_k}\}$ will be sampled from an existing occurrence of the particular motif. Each of these distributions is detailed in the following sections.

5.3.1 Occurrence model

The occurrence model specifies the probability of a particular number of instances of an object class to be present in a scene. As well as modelling the probability of independent objects, the occurrence model defines the probability of a particular grouping *motif* to be present in the scene. When sampling the occurrence of objects and groups a particular configuration $\mathcal{C}$ will be implicitly sampled.

Using the same strategy as in model $\mathcal{M}_1$ the probability of and object class c_i or *motif* class m_k to be present in the scene will be modelled with a Poisson distribution. The parameters λ_{c_i}, λ_{m_k} for the different objects and motifs class distributions will be learned from the occurrences of objects and motifs in the training data.

5.3.1.1 Sampling process

For a new scene given an object class c_i the cardinality of this object class can be sampled from:

$$n_{c_i} \sim Poisson(\lambda_{c_i}), \tag{5.14}$$

and given a motif class m_k the cardinality of this motif class in the new scene can be sampled from:

$$n_{m_k} \sim Poisson(\lambda_{m_k}). \tag{5.15}$$

The sampling process consists on sampling a number of instances from each know object class and each know grouping motif class. In practice most object classes and motif classes will have zero instances in any given scene.

5.3.1.2 Probability evaluation

For a scene $\mathcal{S}$ and a particular parsing configuration $\mathcal{C}$ the probability of occurrence of these particular objects and groups under model $\mathcal{M}_2$ is computed as below. For each possible object class c_i and each possible motif class m_k in the model where the cardinality of objects in $\mathcal{C}$ with class c_i is n_{c_i} and the number of instances of groups in $\mathcal{C}$ with motif class m_k is n_{m_k} (for classes not present cardinality n_{c_i} or n_{m_k} will be zero) the probability of occurrence of this particular configuration set $\mathcal{C}$ of scene $\mathcal{S}$ under model $\mathcal{M}_2$ is:

$$\mathcal{P}_{occ}(\mathcal{C}|\mathcal{M}_2) = \prod_i Poisson(n_{c_i};\lambda_{c_i}) \prod_k Poisson(n_{m_k};\lambda_{m_k}). \tag{5.16}$$

5.3.2 Spatial model

In order to represent the spatial distribution of objects and objects clusters a grid strategy will be used as in model $\mathcal{M}_1$. The modelling for independent object's position is the same as in model $\mathcal{M}_1$. And the probability of a particular group $\mathcal{Y}_k$ to be placed in a particular cell g_{hj} will be modelled with a categorical distribution. Given that the model accounts for objects groupings and independent objects, the parameters of the categorical distribution which models the independent objects will be different from those in $\mathcal{M}_1$. In order to define the spatial probability of a scene a particular configuration $\mathcal{C}$ has to be decided beforehand which will define object classes c_i an motif classes m_k.

5.3.2.1 Sampling process

For each independent object O_i and each group $\mathcal{Y}_k$ the positions in the grid x_i, y_i and x_k, y_k will be sampled from:

$$\{x_i, y_i\} \sim Categorical(c_i), \tag{5.17}$$

$$\{x_k, y_k\} \sim Categorical(m_k). \tag{5.18}$$

As in $\mathcal{M}_1$ rotations θ_i and θ_k will be uniformly randomly sampled from $[0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}]$.

5.3.2.2 Probability evaluation

Given a scene $\mathcal{S}$ represented by configuration $\mathcal{C}$ and given the area $\mathcal{A}_c$ of each cell in the scene. The spatial probability $\mathcal{P}_{loc}$ of an independent object being placed in a particular cell g_{hj} and the spatial probability particular grouping $\mathcal{Y}_k$ to be placed in a particular cell g_{hj} is evaluated by the following equations:

$$\mathcal{P}_{loc}(O_i|\mathcal{M}_2, \mathcal{C}) = Categorical(\{x_i, y_i\} \in g_{hj}|c_i)/\mathcal{A}_c. \tag{5.19}$$

$$\mathcal{P}_{loc}(\mathcal{Y}_k|\mathcal{M}_2, \mathcal{C}) = Categorical(\{x_k, y_k\} \in g_{hj}|m_k)/\mathcal{A}_c. \tag{5.20}$$

5.3.3 Grouping model

Model $\mathcal{M}_2$ accounts for object groupings. Given a group $\mathcal{Y}_k$ with objects $\{O_{k2}, \cdots, O_{kN_k}\}$ and given that the group has *motif* class m_k the spatial distributions of objects this grouping is given by a probability distribution $\mathcal{P}_{m_k}$. This distribution is learned using the method explained in Section 4.3 which takes one object of the O_{kN_k} group objects as the centre object for the cluster, we will refer to this object as O_{k1}, and models the position of the remaining "satellite" objects $\{O_{k2}, \cdots, O_{kN_k}\}$ as an Infinite Gaussian Mixture Model explained in Section 2.1. From this mixture model some clusters are extracted as motifs and the distribution each motif m_k is modelled as a multivariate Gaussian with mean μ_{m_k} and covariance Σ_{m_k}.

5.3.3.1 Sampling process

Once the group position $\{x_k, y_k\}$ has been sampled then the position $\{x_{k1}, y_{k1}\}$ from object O_{k1} will be set to $\{x_k, y_k\}$ and relative positions $\{x_{k2}, y_{k2}, \cdots, x_{kN_k}, y_{kN_k}\}$ can be sampled from:

$$\{x_{k2}, y_{k2}, \cdots, x_{kN_k}, y_{kN_k}\} \sim \mathcal{N}(\mu_k, \Sigma_k). \tag{5.21}$$

This sampled positions will be relative to $\{x_k, y_k\}$, therefore, they need to be transformed in global positions when placing objects $\{O_{k2}, \cdots, O_{kN_k}\}$ in a new room.

5.3.3.2 Probability Evaluation

For evaluating the probability of a particular spatial configuration of a group of objects with class m_k the probability is derived as follows:

$$\mathcal{P}_{loc}(O_{k1},\cdots,O_{kN_k}|\mathcal{M}_2,\mathcal{C}) = \mathcal{P}_{mk}(O_i,\cdots,O_{kN_k}|\mathcal{Y}_k)\mathcal{P}_{loc}(\mathcal{Y}_k|\mathcal{M}_2,\mathcal{C}). \tag{5.22}$$

As mentioned beforehand in practice $\mathcal{P}_{mk}$ will be a multivariate Gaussian with parameters μ_{m_k} and Σ_{m_k} so:

$$\begin{aligned}&\mathcal{P}_{m_k}(O_{k1},\cdots,O_{kN_k}|\mathcal{Y}_k) = \mathcal{P}_{m_k}(O_{k2},\cdots,O_{kN_k}|O_{k1})\\&= \mathcal{N}(x_{k2},y_{k2},\cdots,x_{kN_k},y_{kN_k};\mu_k,\Sigma_k).\end{aligned} \tag{5.23}$$

For the satellite object rotations $(\theta_{k2},\cdots,\theta_{kN_k})$, these are taken from a existing sampled occurrence from available occurrences of motif class m_k

5.3.4 Room Size model

In this case the room size model is the same as in model $\mathcal{M}_1$ respecting the same sampling process for new scenes.

5.3.5 Normalisation constant

Normalisation $\mathcal{Z}_2$ constant for model $\mathcal{M}_2$ is calculated as in Equation 5.9, but sampling scenes from model $\mathcal{M}_2$.

5.3.6 Scene sampling

The scene sampling process is the following:

- Sample a room size dimensions X,Y
- Sample number of instances n_{c_i} for each object class c_i.
- Sample cardinality for each motif class n_{m_k} for each motif class m_k.
- Having sampled all n_{c_i} and n_{m_k} $\mathcal{C}$ is set.
- For each object instance O_i sample the object location in the scene (x_i,y_i,θ_i)
- For each object instance O_i sample a CAD model uniformly from the available CAD models in the data-set.
- For each object group instance $\mathcal{Y}_k$ sample an exiting motif instance in the data-set and get the CAD models.

- For each group instance $\mathcal{Y}_k$ sample the grouping location and rotation in the scene (x_k, y_k, θ_k).
- For each group instance $\mathcal{Y}_k$ set the location for centre object O_{N1} and sample the relative locations of the remaining objects $O_{N2}, \cdots, O_{Nk}$. The relative rotations are taken from the sampled occurrence in the data-set.

5.3.7 Final probability

The final probability of a scene under model $\mathcal{M}_2$ is computed as follows:

$$\mathcal{P}(\mathcal{S}|\mathcal{M}_2) = \mathcal{P}(O_1, O_2, \cdots, O_N|\mathcal{M}_2). \tag{5.24}$$

However, in order to evaluate the probability of scene $\mathcal{S}$ under model $\mathcal{M}_2$ we will have to account for the different configurations $\mathcal{C}$ that a scene $\mathcal{S}$ can take under the model. We will call $\mathcal{C}_s$ the set of possible configurations for a scene $\mathcal{S}$. Given $\mathcal{C}_s$ the final probability is:

$$\mathcal{P}(\mathcal{S}|\mathcal{M}_2) = \frac{1}{\mathcal{Z}_2} \sum_{\mathcal{C} \in \mathcal{C}_s} \mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C}) \mathcal{P}(\mathcal{C}|\mathcal{M}_2). \tag{5.25}$$

From the definitions of the model we can derive that:

$$\mathcal{P}(\mathcal{C}|\mathcal{M}_2) = \mathcal{P}_{occ}(\mathcal{C}|\mathcal{M}_2), \tag{5.26}$$

moreover, once $\mathcal{C}$ is given then all objects classes c_i and motif classes m_k are set so:

$$\mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C}) = \mathcal{P}_{loc}(\mathcal{S}|\mathcal{M}_2, \mathcal{C}) = \prod_i \mathcal{P}_{loc}(O_i|\mathcal{M}_2, \mathcal{C}) \prod_k \mathcal{P}_{loc}(O_{k1}, \cdots, O_{kN_k}|\mathcal{M}_2, \mathcal{C}), \tag{5.27}$$

where index i over existing independent scene objects and index k over existing object groups in scene. Given equation 5.28, we will search through $\mathcal{C}_s$ in order to find $\mathcal{C}^*$ that maximises $\mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C})\mathcal{P}(\mathcal{C}|\mathcal{M}_2)$. It is sensible to think there is one particular configuration $\mathcal{C}^*$ that generated the target scene $\mathcal{S}$ and maximises the probability of the scene. We will search for this particular configuration and given $\mathcal{C}^*$ we will lower bound the probability of $\mathcal{S}$ under model $\mathcal{M}_2$. This can be derived from Equation 5.24 and this relation is expressed in the following equation:

$$\mathcal{P}(\mathcal{S}|\mathcal{M}_2) \geq \frac{1}{\mathcal{Z}_2} \mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C}^*) \mathcal{P}(\mathcal{C}^*|\mathcal{M}_2). \tag{5.28}$$

5.3.8 Learning the hierarchical structure of scenes

Model $\mathcal{M}_2$ admits several interpretations of a scene. In order to approximate the probability of a scene $\mathcal{S}$ under this model, the configuration $\mathcal{C}^*$ that maximises the probability of the scene needs to be found. We will assume that scene $\mathcal{S}$ was generated by $\mathcal{C}^*$ and that this configuration represent the correct parsing tree for the scene. Searching for the highest probability configuration under $\mathcal{M}_2$ can be used to learn the parsing tree of a scene. This follows a similar logic as the approach taken by Liu et al. (2014) were they find the best parsing graph under their probabilistic grammars model by estimating the MAP solution for the probability of a configuration given a scene . For our model this can be derived using Bayes rule:

$$\mathcal{P}(\mathcal{C}|\mathcal{S},\mathcal{M}_2) \propto \mathcal{P}(\mathcal{C}|\mathcal{M}_2)\mathcal{P}(\mathcal{S}|\mathcal{C},\mathcal{M}_2). \tag{5.29}$$

Therefore finding the configuration that maximises the probability of $\mathcal{S}$ is also finding the highest probability configuration.

Finding $\mathcal{C}^*$ for a scene can be solved doing a exhaustive search and computing the probability for all possible configurations. The number of possible configurations in a scene $\mathcal{S}$ is given by the set of objects in the scene, the possible class tuples that form patterns that appear in the scene and the available motifs for each class tuple. Moreover, if the scene set has repetition in objects (e.g. several beds, several stands or dressers) the number of possible configurations increases considerably. The exact number of configuration becomes difficult to track and the exact formula for this number is dependent on the repetitions of particular objects in each scene. Nevertheless, in practice we can find $\mathcal{C}^*$ doing a combinatorial search by building a tree with all the possible configurations. In this search tree, each path is a possible configuration and on each leaf node we will save the final probability of $\mathcal{S}$ for the given the path: $\mathcal{P}(S|\mathcal{M}_2,\mathcal{C})$. The tree is built avoiding repetitions when possible and pruning the tree paths whenever it is possible in order to reduce the search time.

To give further explanation about this search, we will build over an example which involves repetitions for a scene composed by the following set:

$$\mathcal{S} = \{\textit{single bed 1}, \textit{single bed 2}, \textit{stand 1}, \textit{stand 2}, \textit{wardrobe}\} \tag{5.30}$$

and for this example the only class tuple which can form groupings is (*single bed*, *stand*). For this pattern we will consider possible motifs: *"motif-A"* and *"motif-B"*. Without considering different motifs our search tree has 7 possible paths shown in Figure 5.4. If motifs are considered for every node composed by a single bed and a

stand will split in two, this will give us 17 possible configurations. Given this example if we consider more motifs per class tuple and more grouping patterns the number of configurations per scene increases exponentially for big objects sets.

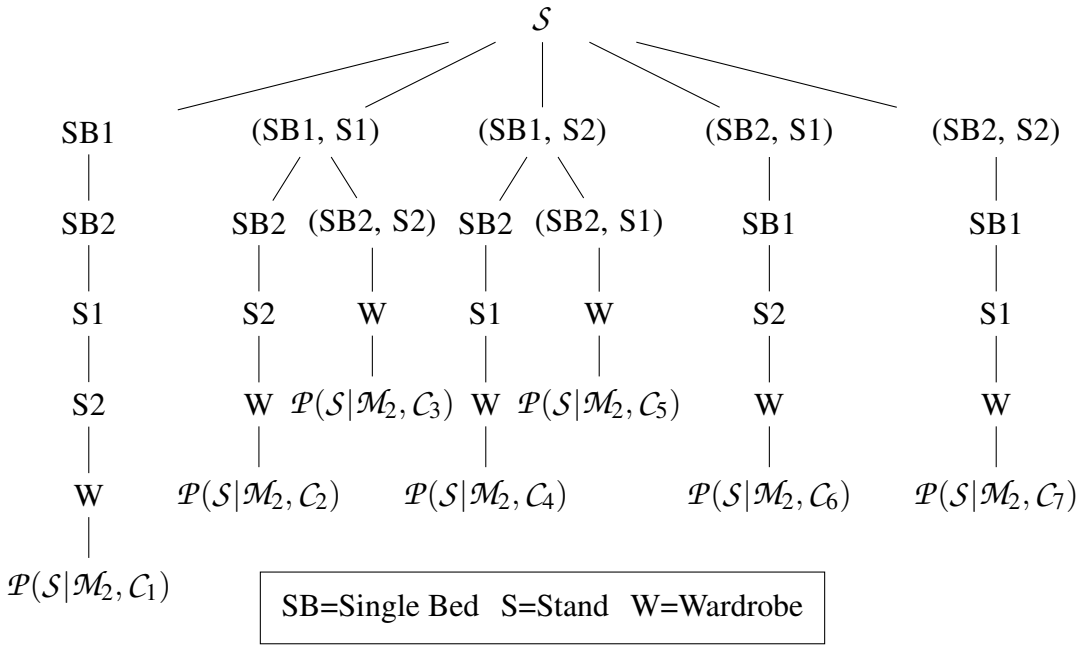


Figure 5.4: Example decision tree for a given scene $\mathcal{S}$ without motifs

Once the tree is built and every $\mathcal{P}(S|\mathcal{M}_2, \mathcal{C})$ is computed we compute the probability of each path $\mathcal{P}(C|\mathcal{M}_2)$. Having done this is is possible to find $\mathcal{C}^*$ which maximises the probability of a scene under the model, which is also the highest probability scene graph interpretation under model $\mathcal{M}_2$.

5.4 Evaluation Metrics

For evaluating our models we will use a hold-out strategy, where we will use 70% of the available input data for training 20% for validation purposes and hyper-parameter selection and 10% for testing purposes. Once the hyper-parameters are selected we will retrain the model with the training and validation set all together and compute the evaluation metrics under the testing set.

For the comparison of both models we will define that if $\mathcal{P}(S|\mathcal{M}_2, \mathcal{C}^*) < \mathcal{P}(S|\mathcal{M}_1)$ scene $\mathcal{S}$ prefers hypothesis $\mathcal{H}_1$, which means that $\mathcal{M}_1$ is a better model for representing scenes. On the other hand if $\mathcal{P}(S|\mathcal{M}_2, \mathcal{C}^*) > \mathcal{P}(S|\mathcal{M}_1)$ we will say that scene $\mathcal{S}$ prefers hypothesis $\mathcal{H}_2$, which means that model $\mathcal{M}_2$ is a better model for representing scenes.

The evaluation metrics we will report are the number of scenes which prefer $\mathcal{H}_1$ and $\mathcal{H}_2$. Moreover, we will report the average $\log \mathcal{P}(S|\mathcal{M}_2, \mathcal{C}^*)$ and the average $\log \mathcal{P}(S|\mathcal{M}_1)$ under our testing set. Finally, for the analysis of our models we will report the normalisation constants $\mathcal{Z}_1$ and $\mathcal{Z}_1$ and for model $\mathcal{M}_2$ we will report the number of groupings found.

Chapter 6

Experiments & Results

In this chapter we will explain the experiments designed to test models $\mathcal{M}_1$ and $\mathcal{M}_2$ and report their results. Firstly, we will detail the data cleaning process. Secondly, we will expose different experiments done to learn scene arrangements. Moreover, we will do a further analysis on the hierarchy learning process. Finally, we will present the results and compute the evaluation metrics defined in Chapter 5 as well as show examples from new sampled scenes from both models.

6.1 Cleaning the Data

SUNCG data-set has over 45,000 scenes of houses which in total provide more than 250,000 different room scenes. For this project we selected to work on with scenes of Bedrooms. This selection was done by using the code explained in Chapter 4 in order to extract motifs from different room types and select the room type with most potential arrangements. Target room types were: Bedrooms, Dining Rooms, Living Rooms and Kitchens. These were selected based on the potential for objects arrangements to appear and the available number of instances for each room. Moreover, since rooms can have multiple room types we select rooms with extrictly one room type. For the target rooms, the number of different object arrangement motifs found and the available room counts are shown in Table 6.1. Given these statistics, it was decided to use Bedrooms for testing our models.

Having selected the target room type, it was decided to split the original set of scenes into three different sub-sets divided by room size. This was done taking into account that room sizes in the data-set are very variable and object count is correlated to the room size. The data-set of bedrooms was split into three different sets: *small*

Room Type	Room Count	Possible Motifs Found
Bedroom	36233	54
Living Room	15675	25
Kitchen	12331	31
Dining Room	6402	29

Table 6.1: Most prevalent room type counts

rooms, *medium rooms*, and *big rooms*. The division was done by percentiles based on room's area and using a tukey's fence (explained in Chapter 2) with $k = 2$ to remove outliers. Given the size of the data-set, it contains some noisy bedroom scenes that can have more than $100m^2$ area and more than 100 objects. We removed outliers to work with reasonably sized scenes and realistic object counts. The resulting room count, area range, and average object count for each set is shown in Table 6.2.

Room Size	Room Count	Area Range (m^2)	Avg. Objects/Room
Small Rooms	12590	0-18	7.8
Medium Rooms	12578	18-27	11.1
Big Rooms	10690	27-58	13.7

Table 6.2: Room divided by sizes

Once the rooms were divided by area range, the three sub-sets were split in training, validation and testing sets with proportions 70-20-10 in order to perform all the experiments in this chapter. The split count for each sub-set is shown in Table 6.3.

Room Size	Training Count	Validation Count	Testing Count
Small Rooms	8731	2579	1279
Medium Rooms	8869	2489	1219
Big Rooms	7511	2058	1120

Table 6.3: Subsets division for training validation and test

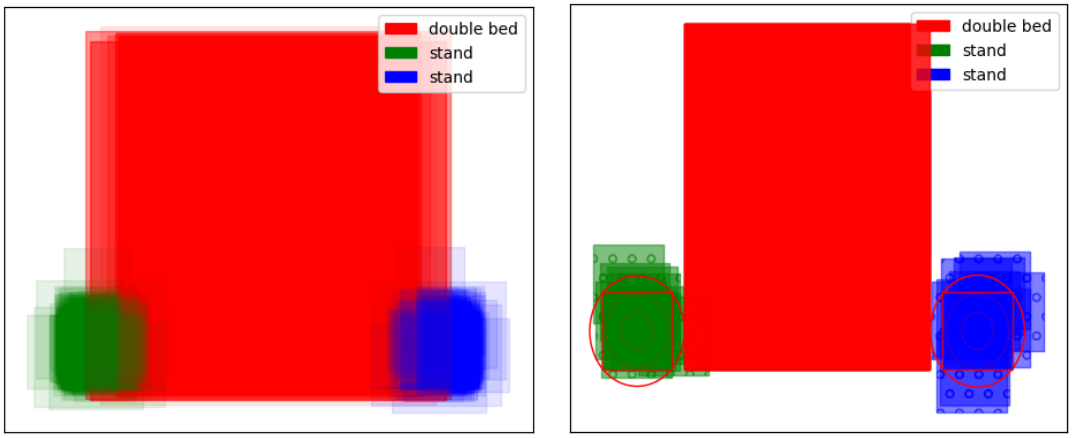
6.2 Learning Pattern Motifs from Scenes

The code provided by Henderson and Ferrari (explained in Section 4.3) was used to learn objects arrangements in Bedrooms. As explained Henderson and Ferrari approach can be used to cluster scene arrangement occurrences. In this project we are also interested in modelling the distribution of objects in groups and finding out the probability of objects being part of a group or appearing as independent objects. For this reason, several experiments were done following this idea.

Firstly, we explored the idea of modelling arrangements as a Gaussian Mixture Model and objects being dependent on an anchor object but independent from each other in a group. This can be done by fitting the mixture model to the training data and for each cluster from Equation 4.1 generating a diagonal covariance matrix. After fitting the model, it was possible to sample new objects arrangements from the different clusters of the Mixture model. In Figure 6.1a, it can be seen how new samples of a motif composed of one double bed and two side stands are distributed, and the covariance lines for each satellite object are plotted. Moreover, real occurrences for the selected motif are plot in Figure 6.1b. In these figures it can be seen that the new sampled positions for the satellite objects (stands) differs from the distribution of the real occurrences. This is given by the fact than when clustering different motifs using diagonal covariance matrices can be useful, however, it does not have realistic results when sampling new instances. As an example, when sampling from a configuration of double-bed, stand, stand if using diagonal covariance matrices, side tables will not be properly aligned between each other. This is because use of diagonal a covariance assumes that side stands' positions are not strictly correlated. However, it is usually is the case that their position is correlated. This process was done for the full list of target class tuples; the list of motifs extracted using diagonal covariance matrices and new sampled examples from these can be seen in Appendix B.2.

Given these results, we extended the original modelling to learn full covariance matrices under the Gaussian Mixture Model. With this approach it possible to model the correlation of objects positions between each others within a particular motif. In Figure 6.2a the original occurrences of a motif composed by two stands and a double bed are shown, and new sampled instances are shown in Figure 6.2b. In Figures 6.3a and 6.3b the same is shown for a particular motif of one single bed and one stand.

Considering the differences between learning objects arrangements using diagonal and full covariance matrices, we decided to use full covariance matrices since these

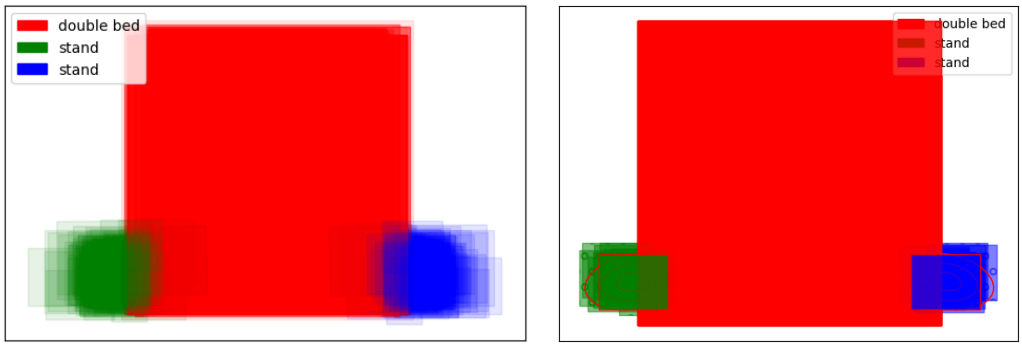


(a) Real Occurrences (b) Sampled Occurrences

Figure 6.1: Sampling from model with diagonal covariance matrices in a motif composed by a double bed and two stands compared to real occurrences

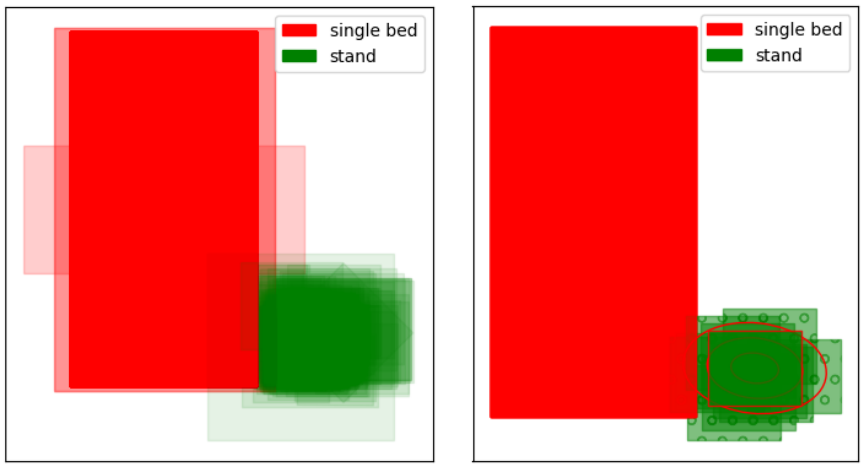
are able to extract more information about the spatial relationships between objects in the group. The full set of extracted motifs and some new sampled occurrences are included in Appendix B.3.

Finally, in order to fully define the process of sampling new object arrangement instances, rotations and a CAD models for the group need to be sampled. It was decided that all rotations for satellite objects and object CAD models for the full group were sampled from an existing occurrence of the selected motif. In practice this gives more realistic results since relative rotations are usually related to the particular CAD models in the grouping. Nevertheless, it is possible to learn the relative rotation of objects as part of the Mixture Model, this has been done in the past by Fisher et al. (2012). In this project some experiments were done adding rotations to the mixture model and results are shown in Appendix B.4. However, as mentioned before, sampling relative rotations from a real occurrence seems to have more realistic results than sampling from the mixture model and fitting the mixture model adding rotations as variables in practice decreases the number of clusters found for each set of classes.



(a) Real Occurrences (b) Sampled Occurrences

Figure 6.2: Sampling from model with full covariance matrices in a motif composed by a double bed and two stands compared to real occurrences



(a) Real Occurrences (b) Sampled Occurrences

Figure 6.3: Sampling from model with full covariance matrices in a motif composed by a sing bed and one stand compared to real occurrences

6.3 Training the models

The process of training the models consist on extracting statistics from the different room sets. Firstly, for training $\mathcal{M}_1$ the global object count per object class was extracted in order to estimate the λ_{c_k} parameters of the Poisson distributions. For the categorical distributions, the count of appearances per class in each cell was extracted. Secondly, the process of learning scene arrangements in the training sets was done. Therefore, different motifs of object patterns were learned and occurrences of groups labelled in the training set. Having done this, $\mathcal{M}_2$ was trained using the count of independent

objects and counts of motifs occurrences found. Moreover, the count of independent objects and object groups corresponding to a particular motif for each cell in the grid were used to learn the categorical distributions.

In order to avoid scenes with zero probability during evaluation, we set a prior probability of $1/1000$ for each cell. This means that objects in the testing set will have a small probability of appearing in unseen positions.

Once this is done all, the parameters for our probability distributions in $\mathcal{M}_1$ and $\mathcal{M}_2$ are set and new scene can be sampled. Moreover, the experiments for hyper-parameter selection on the validation set were performed and these are described in the following section.

6.4 Hyper Parameter Selection - Grid Size

In order to do the final experiments, it was necessary to select a grid size for the model. To do this the models were trained for several $N \times M$ grid sizes for each room set. Even though $N \neq M$ in general, for simplicity $N = M$ was used, and the grid sizes varied from 5×5 to 10×10. This was done for the small and medium size rooms sets. The big rooms set is excluded due to the computation complexity of evaluating the big size rooms under the validation set. Doing the tree search from section 5.3.8 under the big rooms can take up to 6 hours for each grid size.

After training the models for each grid size, the normalisation constants were computed sampling 20,000 scenes. These constants will give an idea of how good the model is for sampling new synthetic scenes. Moreover, we computed the $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_1)$ and $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C}^*)$ for each scene in the validation set and we report the average $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_1)$ and average $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C}^*)$ for each grid size to have a measure of how probable scenes are under model $\mathcal{M}_1$ and $\mathcal{M}_2$. The number of scenes which prefer $\mathcal{H}_1$ and $\mathcal{H}_2$ is reported in addition to the number of groupings found for each grid size. The results for the small and medium room sets under each grid size are shown in Tables 6.4 and 6.5 respectively.

From these results, it is sensible to select the grid size 5×5. This selection was done based on the formulation of this project , were we want to evaluate the probability of scenes under a hierarchical model. Taking this into account we selected the grid size that enabled to find more object arrangement occurrences. Moreover, we considered the normalisation and for 5×5 grid size the ratio of valid scenes is the higher. For these reasons, grid size 5×5 will be used for the final tests.

Grid Size	5×5	6×6	7×7	8×8	9×9	10×10
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)$	**-14.531**	**-14.349**	**-14.195**	**-14.193**	**-14.253**	**-14.116**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)$	**-12.736**	**-12.743**	**-12.724**	**-12.804**	**-12.820**	**-12.842**
Constant $\mathcal{Z}_1$	0.0257	0.02315	0.0205	0.0199	0.02175	0.02135
Constant $\mathcal{Z}_2$	0.0345	0.02945	0.0302	0.0306	0.03035	0.03135
Groupings found	**1257**	**1201**	**1153**	**1134**	**1149**	**1122**
Scenes prefer $\mathcal{H}_1$	1228	1336	1328	1360	1363	1356
Scenes prefer $\mathcal{H}_2$	1351	1243	1251	1219	1216	1223

Table 6.4: Validation Results on Small Rooms based on grid size

Grid Size	5×5	6×6	7×7	8×8	9×9	10×10
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)$	**-23.056**	**-22.702**	**-22.504**	**-22.423**	**-22.231**	**-22.252**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)$	**-20.269**	**-20.570**	**-20.559**	**-20.704**	**-20.549**	**-20.675**
Constant $\mathcal{Z}_1$	0.0205	0.0195	0.01525	0.0138	0.01505	0.0153
Constant $\mathcal{Z}_2$	0.02885	0.024	0.0229	0.0194	0.0208	0.0215
Groupings found	**1670**	**1409**	**1371**	**1344**	**1347**	**1370**
Scenes prefer $\mathcal{H}_1$	862	1111	1121	1209	1202	1242
Scenes prefer $\mathcal{H}_2$	1627	1378	1368	1280	1287	1247

Table 6.5: Validation Results on Medium Rooms based on grid size

6.5 Final Results

After selecting the grid hyper-parameter, the models were retrained using the validation and training set all together; the final statistics extracted to fit the models are reported in Appendix A. Having trained the models, the final normalisation constants were computed using 100,000 samples. In this section the final values for these constants are reported, samples from valid scenes for the models are shown and plots for the log probability of the scenes under each model are presented. Furthermore, we analyse how the probability of scenes changes as the hierarchical graph is being built. Finally, we analyse the results and how the probability changes given the number of groupings found in a scene.

6.5.1 Training and sampling

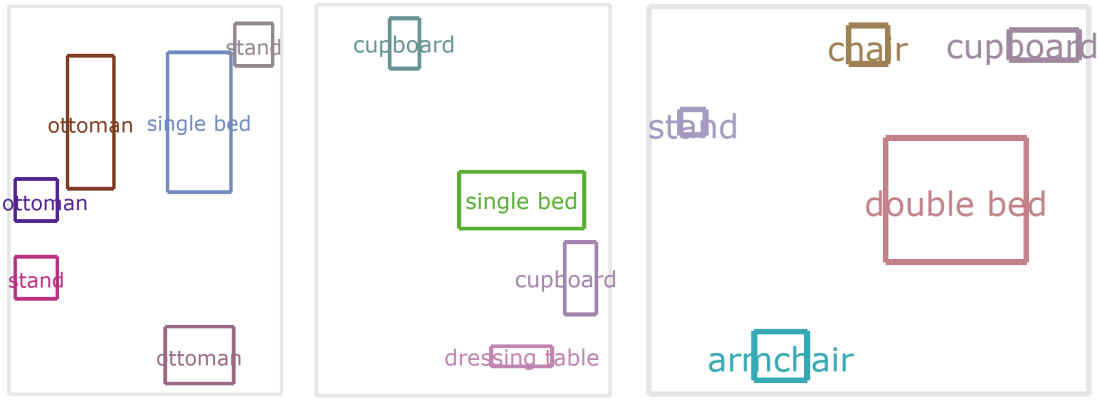
The models were retrained trained using the train and validation set in conjunction, after sampling 100,000 scenes for each room sizes the final normalisation constants computed are reported in Table 6.6.

Room Size	Small	Medium	Big
$\mathcal{Z}_1$	0.02538	0.02166	0.01741
$\mathcal{Z}_2$	0.03298	0.02845	0.02555

Table 6.6: Normalisation constants after sampling 100,000 scenes

These constants show that both models $\mathcal{M}_1$ and $\mathcal{M}_12$ have a high rejection rate and generate more invalid scenes than valid. However, it can be seen that there is a difference in the acceptance rate of both models and that model $\mathcal{M}_2$ generates more valid scenes that model $\mathcal{M}_1$ suggesting that $\mathcal{M}_2$ is a better model for scene modelling.

From the sampling process in models $\mathcal{M}_1$ and $\mathcal{M}_2$ some valid floor plans were extracted for each room size and these are presented in Figure 6.4 and 6.5. Further examples of valid and invalid sampled scenes are reported in Appendix C.

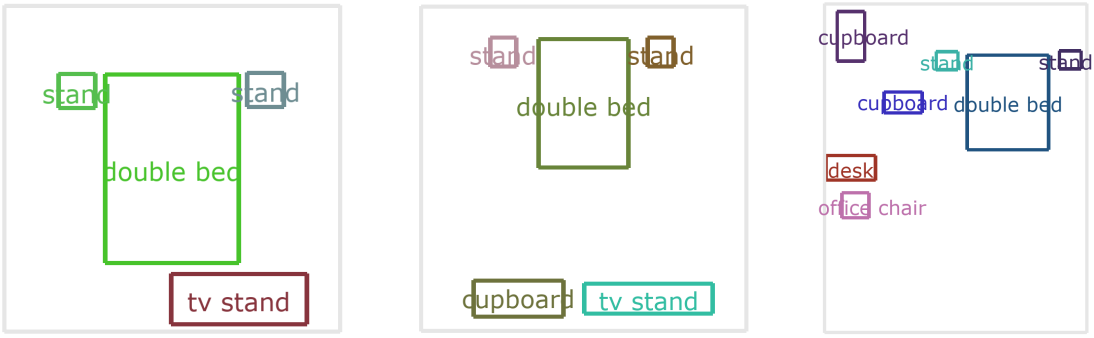


(a) Small room $\mathcal{M}_1$ (b) Medium room $\mathcal{M}_1$ (c) Big room $\mathcal{M}_1$

Figure 6.4: Example valid scenes sampled from $\mathcal{M}_1$

6.5.2 Learning the scene hierarchical graph

In this section we will explain how the probability of a real scene from the testing set varies for the different configurations in can take. We will analyse the scene $\mathcal{S}$ shown



(a) Medium room $\mathcal{M}_2$ (b) Medium room $\mathcal{M}_2$ (c) Big room $\mathcal{M}_2$

Figure 6.5: Example valid scenes sampled from $\mathcal{M}_2$

in Figure 6.6; this is composed by seven furniture objects: single-bed, single-bed, stand, stand, office-chair, desk, wardrobe. Given this scene, a simplified version of the configurations tree (explained in Section 5.3.8) is presented in Figure 6.7. In practice the implemented tree has 18 different configurations after pruning nodes; this number is given by the different combinations of beds and stands and for each combination of these there are two possible motifs. Since there is only one occurrence of a pattern composed by one office-chair and one desk the decision about which motif of this pattern to use can be done in the tree node. This means some paths can be pruned in this tree. In this simplified tree severalconfigurations are presented, starting from a flat configuration were all objects are independent adding arrangements until the highest probability configuration is found. From this figure, it can be clearly seen how the log probability of the scene under model $\mathcal{M}_2$ increases each time a new arrangement is added to generate a new configuration. Finally, the final scene graph interpretation is shown in Figure 6.8.

6.5.3 Evaluating the final probability

In this section we will present the final results under the testing set after training both models on the three different room sizes and learning the hierarchical structure of scenes.

To begin with, it is sensible to compare the probability of a flat configuration $\mathcal{C}_F$ were objects are not grouped under $\mathcal{M}_2$ compared to the probability of scenes under $\mathcal{M}_1$. This was done to have a starting point to compare how probabilities evolve as objects groupings are found in scenes. We computed these probabilities for the three different room sizes and the resulting plots are shown in Figure 6.9. Moreover, results

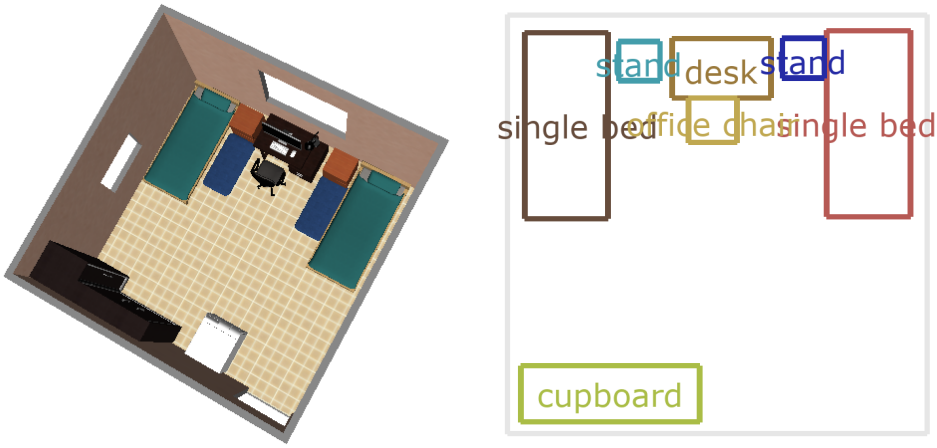


Figure 6.6: Sample scene

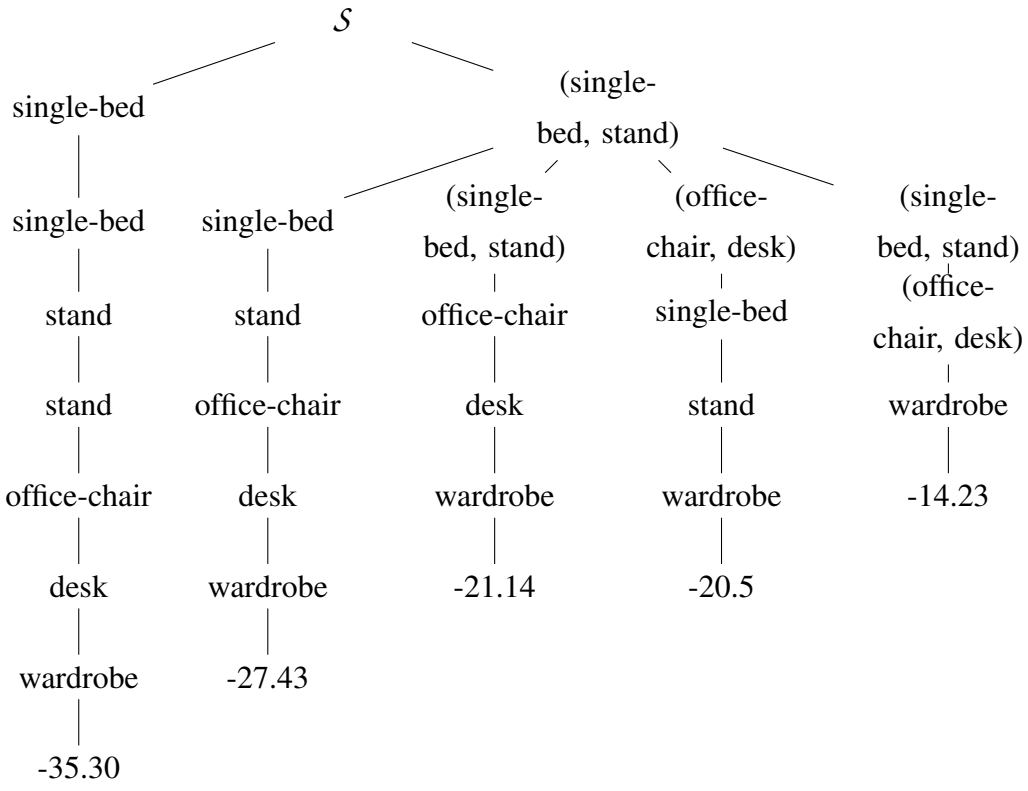


Figure 6.7: Decision tree for scene hierarchy with $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C})$ given different configurations for scene $\mathcal{S}$.

for the final average log probability of scenes on the testing set under both models given the selected configuration are reported in Table 6.7.

After the computing probability of the flat configuration of objects under $\mathcal{M}_2$ and compare it to the probability of $\mathcal{S}$ under $\mathcal{M}_1$, it can be concluded that if analysing

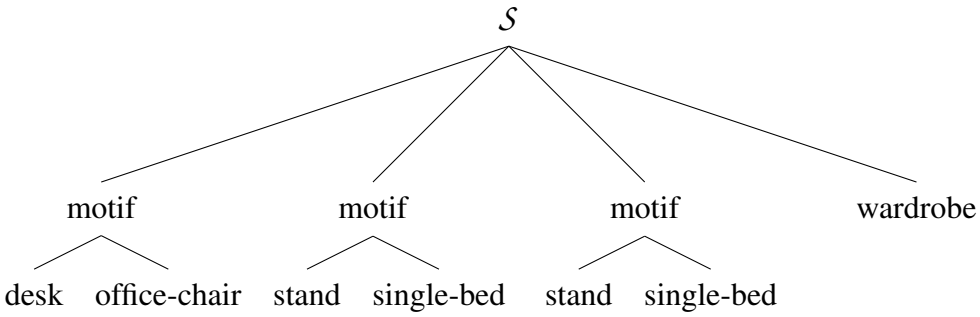


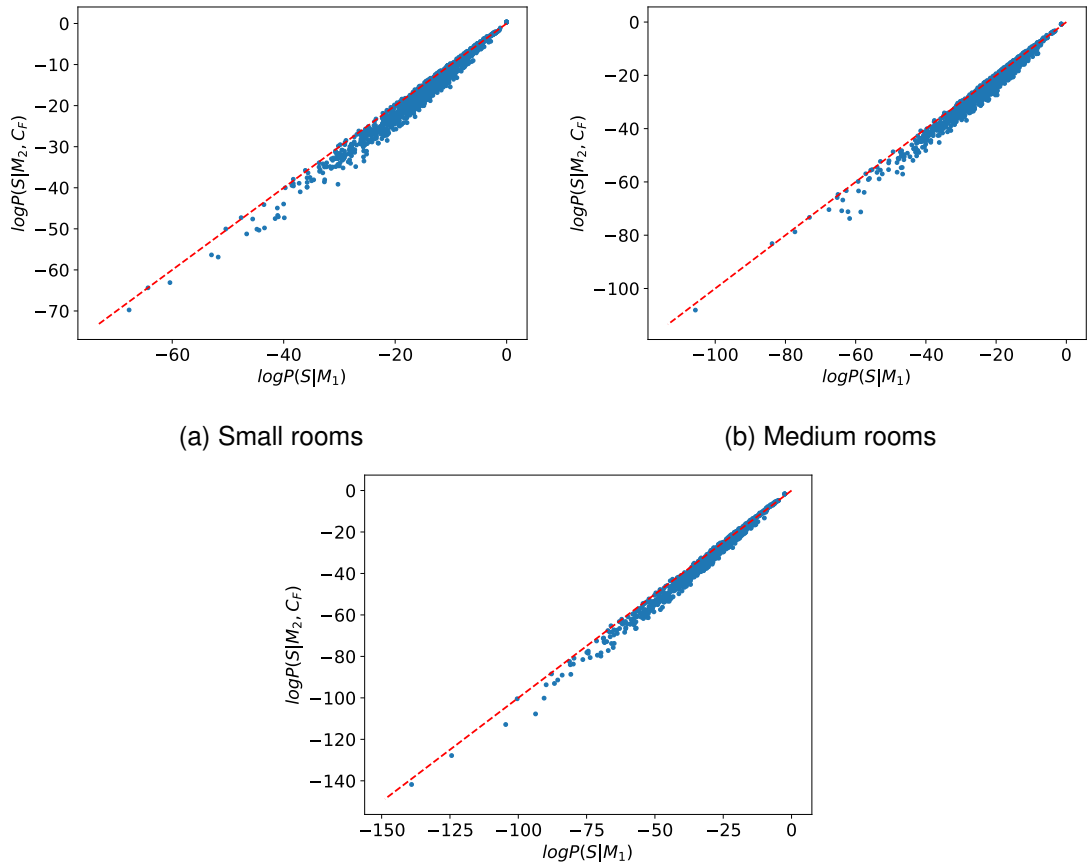
Figure 6.8: Scene hierarchical graph

Room Size	**Small**	**Medium**	**Big**
Avg. $\log P(\mathcal{S}\|\mathcal{M}_1)$	-14.32	-22.85	-31.22
Avg. $\log P(\mathcal{S}\|\mathcal{M}_2, \mathcal{C}_F)$	-16.04	-25.15	-33.69
Scenes prefer $\mathcal{H}_1$	1103	1047	974
Scenes prefer $\mathcal{H}_2$	174	172	146

Table 6.7: Probability of scenes treated as independent objects

scenes without hierarchical structures then model $\mathcal{M}_1$ is preferred. The results show that the average $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_1)$ is higher in every case to the average $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C}_F)$. Moreover, when analysing which hypothesis $\mathcal{H}_1$ or $\mathcal{H}_2$ is preferred, $\mathcal{H}_1$ is preferred in 86.3% of the small scenes, 85.9% of the medium scenes and 86.9% of the big scenes. In Figure 6.9 scenes that prefer $\mathcal{H}_1$ lie below the red line in the plots.

Taking the flat configuration probability as a baseline, the process of learning the hierarchical structure of scenes was carried. Having done this, it was possible to evaluate $\mathcal{P}(S|\mathcal{M}_2, \mathcal{C}^*)$ for each scene. These results are plotted on Figure 6.10 labelled by the number of objects groupings found (these will be used for further analysis) and they are tabulated in Table 6.8. Analysing these results, it can be seen that representing scenes as hierarchical structures reverts the trend seen in Figure 6.9. In this scenario the average $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_2, \mathcal{C}^*)$ is higher than the $\log \mathcal{P}(\mathcal{S}|\mathcal{M}_1)$. Moreover, more scenes prefer $\mathcal{H}_2$ to $\mathcal{H}_1$ in the three testing sets. For the best hierarchical configuration $\mathcal{C}^*$, 54.9% of the small scenes, 71.4% of the medium scenes and 77.5% of the big scenes prefer $\mathcal{H}_2$ over $\mathcal{H}_1$. In Figure 6.10 scenes that prefer $\mathcal{H}_2$ lie above the red line in the plots. This suggests model $\mathcal{M}_2$ is a better model for analysing the structure of scenes and its able to better learn the inherent structure of scenes.



(a) Small rooms

(b) Medium rooms

(c) Big rooms

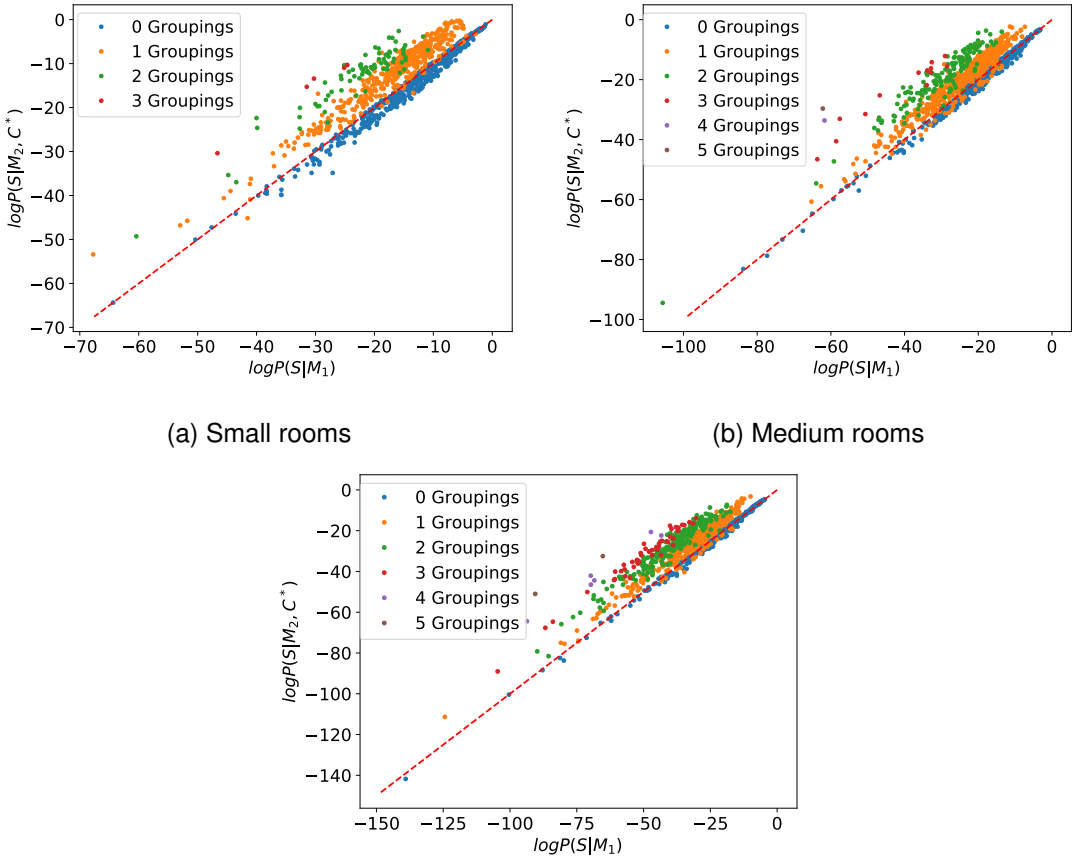
Figure 6.9: Flat Configuration Probability under $\mathcal{M}_1$ and $\mathcal{M}_2$

Some further exploration was done under both models. It was studied how the probability of scenes conditional to the number of groupings found behaved. These results are shown in Figure 6.10 and in Tables 6.9, 6.10 and 6.11 it is reported how the probability of scenes varies depending on the number of groupings found for the small, medium and big rooms respectively. Inspecting these results, it can be seen that the ratio between the probability of scenes under model $\mathcal{M}_2$ and $\mathcal{M}_1$ increases as the number of arrangements found increases and this trend is valid in all room sizes. This behaviour supports the theory that scenes should be interpreted as hierarchical graphs taking into account the relationships between objects and that model $\mathcal{M}_2$ is better for representing the inherent structure of scenes.

These results also show that even if the ratio between the probability under $\mathcal{M}_2$ and $\mathcal{M}_1$ increases, the global probability values decrease as scenes become more complex and have more objects and groups. Therefore, it is sensible to study how the probability of scenes behaves based on the number of objects per scene. Further exploration was

Room Size	**Small**	**Medium**	**Big**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)$	**-14.32**	**-22.85**	**-31.22**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)$	**-12.54**	**-19.45**	**-25.77**
Scenes prefer $\mathcal{H}_1$	577	349	252
Scenes prefer $\mathcal{H}_2$	702	870	868
Groupings found	**630**	**930**	**1104**

Table 6.8: Testing set results after finding configuration $\mathcal{C}^*$



(a) Small rooms

(b) Medium rooms

(c) Big rooms

Figure 6.10: Results with number of groupings labelled

done by labelling the probability of scenes given the number of objects present in the scene. Figures relating these probabilities can be seen in Appendix D. Nevertheless, due to the way the models are designed it is expected that the probability of scenes will decrease as the number of objects in a scene increases. On the other hand, bigger counts of objects support finding more arrangements as shown by results where more

Groupings Found	**0**	**1**	**2**	**3**
Scenes Count	737	455	80	5
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)$	**-11.662**	**-16.981**	**-22.543**	**-31.624**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)$	**-12.519**	**-12.498**	**-12.742**	**-16.066**

Table 6.9: Small rooms results based on groupings found

Groupings Found	**0**	**1**	**2**	**3**	**4**	**5**
Scenes Count	477	572	155	13	1	1
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)$	**-18.029**	**-24.417**	**-29.851**	**-41.308**	**-61.721**	**-62.187**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)$	**-18.770**	**-19.915**	**-19.271**	**-23.786**	**-33.630**	**-29.650**

Table 6.10: Medium rooms results based on groupings found

Groupings Found	**0**	**1**	**2**	**3**	**4**	**5**
Scenes Count	368	468	226	50	6	2
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)$	**-22.899**	**-31.108**	**-39.633**	**-49.439**	**-65.366**	**-77.903**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)$	**-23.475**	**-25.671**	**-27.826**	**-32.009**	**-40.062**	**-41.734**

Table 6.11: Big rooms results based on groupings found

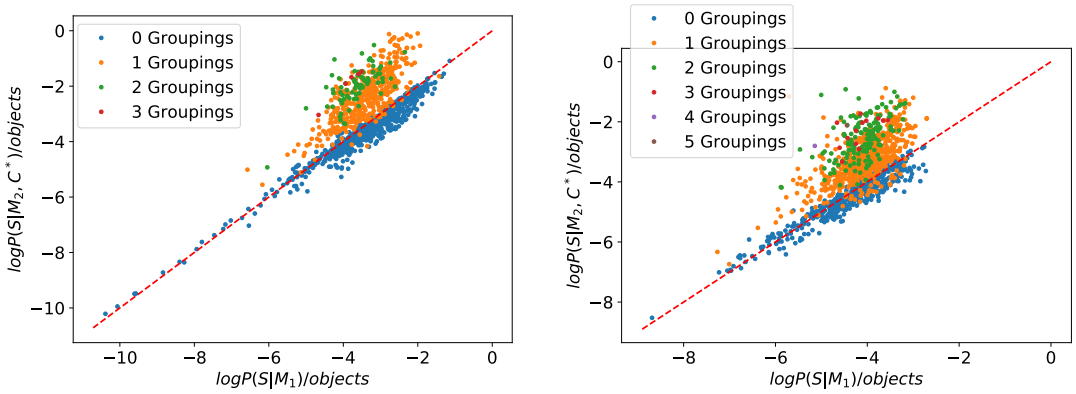
groupings were found under the medium and big size rooms. For this reason, if we want to explore how the effect of arrangements affects the probability of scenes it is necessary to mitigate the effect of the object count in scenes. Therefore, we normalised the probability values by the object count. The relationship between the probability of a scene and its object count is of exponential nature given that each object adds a new term to the spatial probability. Therefore, we decided to compute the following values:

$$\log \mathcal{P}(S|\mathcal{M}_1)/\#Objects \tag{6.1}$$

$$\log \mathcal{P}(S|\mathcal{M}_2, \mathcal{C}^*)/\#Objects, \tag{6.2}$$

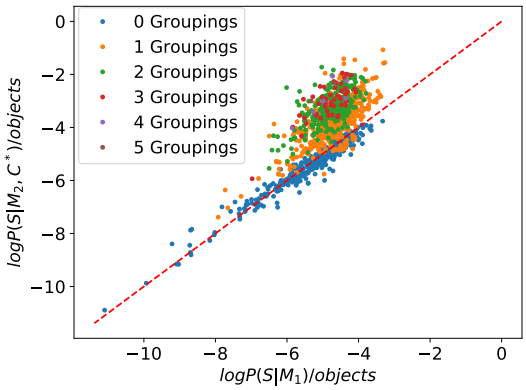
and explored how the probability of scenes evolves for the different number of objects' arrangements found in scenes normalised by the number of objects in each scene. These results are plotted in Figure 6.11 labelled by number of groups found and plots labelled by object count are included in Appendix D. Moreover, results analysed by number of groups found are tabulated in tables 6.12, 6.13 and 6.14 for each testing

set. From these results, it is even more clear how the probability of scenes increases whenever an arrangement is found and added to a configuration. In the previous results we could infer this from the ratio between $\log \mathcal{P}(S|\mathcal{M}_2, \mathcal{C}^*)$ and $\log \mathcal{P}(S|\mathcal{M}_1)$. However, when normalising by the number of objects it can be clearly seen how $\log \mathcal{P}(S|\mathcal{M}_2, \mathcal{C}^*)/Objects$ increases as more groups are found.



(a) Small rooms

(b) Medium rooms



(c) Big rooms

Figure 6.11: Results with number of groupings labelled normalised by number of objects per scene

Groupings Found	**0**	**1**	**2**	**3**
Scenes Count	737	455	80	5
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)/Obj$	**-3.370**	**-3.416**	**-3.631**	**-3.898**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)/Obj$	**-3.598**	**-2.401**	**-1.981**	**-1.932**

Table 6.12: Small rooms results normalised by object based on groupings found

Groupings Found	**0**	**1**	**2**	**3**	**4**	**5**
Scenes Count	477	572	155	13	1	1
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)/Obj$	**-4.389**	**-4.065**	**-4.126**	**-4.166**	**-5.143**	**-4.441**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)/Obj$	**-4.539**	**-3.228**	**-2.561**	**-2.323**	**-2.803**	**-2.118**

Table 6.13: Medium rooms normalised by object results based on groupings found

Groupings Found	**0**	**1**	**2**	**3**	**4**	**5**
Scenes Count	368	468	226	50	6	2
Avg. $\log \mathcal{P}(S\|\mathcal{M}_1)/Obj$	**-5.310**	**-4.685**	**-4.813**	**-4.826**	**-4.920**	**-4.994**
Avg. $\log \mathcal{P}(S\|\mathcal{M}_2, \mathcal{C}^*)/Obj$	**-5.399**	**-3.769**	**-3.283**	**-3.042**	**-2.942**	**-2.660**

Table 6.14: Big rooms results normalised by object based on groupings found

All the results in this section support the hypothesis that hierarchical models are able to extract the inherent structure of scenes and learn how objects arrange in scenes.

Chapter 7

Conclusions

In the previous chapters we defined two probabilistic models for 3D scene sampling and evaluation and provided several experiments which test the probability of real world scenes under both models. Moreover, we presented the results of these experiments under different room sizes data-sets and made further analysis of how our hierarchical model can be used to learn the hierarchical structure of scenes.

In this chapter we put our results into context, analysing how they fit in with the related work in the field and summarise our contributions to the field of generative modelling of 3D scenes. Moreover, we do some further discussion on the possible drawbacks of our experiments and finally, we present some future research ideas that could follow this project.

7.1 Contributions

In this dissertation we presented two baseline generative models for 3D scenes, one that models scenes with no hierarchical interpretation and one that models scenes as hierarchical structures and accounts for the relationship between objects. These two models are properly normalised and are tractable enough that the probability of scenes can be evaluated. The main contribution to the field is that we provide two properly defined baseline models that are able to evaluate the probability of real scenes. Moreover, we test an already presented method for learning scene arrangements and we use objects spatial relationships to learn the hierarchical representation of a scene under our hierarchical model. Finally, we tested our hypothesis that hierarchical models are better for representing the inherent structure of scenes compared to flat models. These contributions are going to be discussed in this section.

Much of the literature in the area as Henderson and Ferrari (2017); Fisher et al. (2012) has focused on the problem of generating new scenes and getting realistic results, however, in practice the likelihood of scenes under these models is intractable. This is given by the fact that most models at some level introduce human crafted rules to make scenes look more realistic and by doing this the models become non-parametric. Moreover, modelling the representation space of 3D scenes with probability distributions is a high dimensional problem which has not been completely solved. In this dissertation we presented two models which are parametric, normalised and tractable, therefore, the probability of real scenes can be evaluated. On the other hand, these models have a high rejection rate generating more invalid scenes than valid scenes and the generated scenes are not comparable with humanly generated scenes. Nevertheless, these results were expected since the motivation of this project was related to the evaluation of scenes rather than defining realistic generative models. Nowadays, when evaluating generative models these are evaluated using perceptual tests, which involve human participants deciding whether synthetic scenes or human generated scenes look more realistic. The models presented in this research, and particularly model $\mathcal{M}_2$ could be used as a baseline to evaluate generative models by comparing the probability of synthetic scenes to real scenes under the model. Nevertheless, this possibility has to be addressed by future research.

Learning scene arrangements using Gaussian Mixture models based on pairwise spatial relationships between objects in combination with finding the maximum probability hierarchical representation provides a new method for learning scene graphs. To our knowledge this approach has not been reported in the past. The closest to our method, is the work by Liu et al Liu et al. (2014) which uses probabilistic grammars models to learn the hierarchical graph of scenes. Moreover, based on work by Henderson and Ferrari (2017) we did further exploration over the idea of using an Infinite Gaussian mixture model to model pairwise relationship between objects. This approach was also recently used by Wang et al. (2018) for comparison to evaluate their generative model, and in the past by Fisher et al. (2012) using a finite mixture model. In our research we explored different ways in which objects arrangements can be model using this method and decided that using full covariance matrices yields better results when using the learned probability distribution to evaluate objects arrangement as well as sampling new ones.

Both models were developed in order to test the hypothesis that hierarchical models are able to learn the inherent structure of scenes. From the results obtained we can

clearly conclude that modelling scenes using hierarchical structures based on spatial objects arrangements learns valuable information from scenes, and therefore, scenes evaluated under model $\mathcal{M}_2$ have a higher average probability than when evaluated under model $\mathcal{M}_1$. This suggests that model $\mathcal{M}_2$ is better for modelling the indoor scene space than model $\mathcal{M}_1$. Finally, when testing how the probability of scenes changes as the groupings are added, we can conclude that clustering objects increases the probability of scenes under model $\mathcal{M}_2$ which implies that clustering objects and building hierarchical structures is a more realistic way of modelling scenes. Therefore, the benefits of using a hierarchical model are clearly expressed by the results of this project and we can conclude that hierarchical models are better for representing the inherent structure of scenes than flat models.

7.2 Discussion

In this section we discuss about our models formulation and suggest improvements to them given the results obtained.

In analysing our results, the first drawback of our models seems to be the use of a categorical distribution on a discrete grid of cells for modelling the spatial distribution of object classes in scenes. This can generate big discrete jumps in the probability values for adjacent cells. It is sensible to think that a more realistic approach will be to use a well defined continuous PDF function to model the spatial distribution of object classes in scenes. Our motivation for using the categorical distribution was simplicity, however, it should be possible to extend this to the continuous domain in future work.

The SUNCG data-set provides a significant amount of data and has enabled several new data driven methods to sample new scenes and some deep learning approaches to this problem. However, the provided data is quite noisy and contains scenes with alien objects. For example, objects as sinks and toilets or kitchen appliances appear in scenes labelled as bedrooms which was our target study room type. Having said this, although a data cleaning process was done before training our models a more extensive filtering process can lead to better results when sampling new rooms.

Doing an exhaustive search in order to find the maximum probability configuration $\mathcal{C}^*$ is a simple solution to the problem and is effective for small object counts. Nevertheless, it was the case that some complex scenes with more than 20 objects took more than 30 minutes to be solved. This problem could be addressed in the future as a dynamic programming problem in order to find the optimal solution faster.

Finally, further improvement to the models could be done by learning probability distributions over the rotations of objects and object's CAD models. This can lead to more realistic results when generating scenes new scene.

7.3 Future Research Work

Given the results of our research in this section we analyse some possible research directions based on our project.

The first sensible suggestion to continue with this research is to explore how more levels of hierarchical grouping work when learning arrangements and generating the hierarchical graph. Given that motif occurrences are labelled when doing the process of learning object arrangements is should be possible to cluster motifs together using a Gaussian Mixture Model. Moreover, in this particular project a constrained number of classes was used to learn object arrangements. However, the number of classes that can form groupings in scenes is not completely explored, this could also be explored in the future. Furthermore, the same experiments presented in this project could be expanded for different room types.

As mentioned beforehand our models are effective at evaluating the probability of real scenes, and these could be used in the future to evaluate and compare models. This possibility could be explored by comparing the probability of synthetic scenes generated by various generative models to the probability of human generated scenes. Moreover, the result should be correlated with the results obtained from perceptual tests, to validate if the probability results from our models suggest the same results as human supervised validation. Nevertheless, in order to do so it is sensible to think that rotations for objects and CAD models should be included in the modelling process.

Finally, some recent research such as Wang et al. (2018) have address the scene synthesis problem as conditional problem, were each object is sampled conditional to the prior sampled objects. In our case for evaluating the probability of new scenes it is interesting to model the occurrence probability of objects as a joint distribution. In our occurrence models, the presence of an object is sampled independently for each object class. Given the amount of data provided by the SUNCG data set it would be sensible to try to estimate the joint probability distribution for the occurrence of objects in scenes, at least for furniture objects. This could be done using a mixture model or estimating the distribution with Neural Autoregressive Density Estimation (NADE) Uria et al. (2016) or a Restricted Boltzmann Machine.

Appendix A

Dataset Statistics

A.1 Statistics model $\mathcal{M}_1$

Object Class	Count	Object Class	Count	Object Class	Count
desk	3121	dresser	2483	chair	1405
car	33	bicycle	1	sink	105
bench_chair	8	shower	37	tv_stand	1171
armchair	1165	ottoman	1175	dressing_table	1582
motorcycle	15	game_table	2	kitchen_cabinet	115
tripod	5	wardrobe_cabinet	9719	stand	6872
trash_can	13	shoes_cabinet	318	storage_bench	2
wood_board	10	basketball_hoop	9	single_bed	4159
baby_bed	662	double_bed	6300	fence	10
table	1	coffee_table	1125	fireplace	63
chair_set	4	goal_post	7	sofa	509
toilet	73	office_chair	1988	bathtub	34
workplace	42	bunker_bed	908	gym_equipment	120
dining_table	112	table_and_chair	8		

Table A.1: Class counts for small rooms under model $\mathcal{M}_1$

Object Class	**Count**	**Object Class**	**Count**	**Object Class**	**Count**
desk	4017	dresser	3568	chair	2180
car	33	sofa	1399	bench_chair	11
shower	10	tv_stand	2421	armchair	2299
ottoman	2556	dressing_table	2638	motorcycle	18
game_table	24	kitchen_cabinet	179	tripod	6
wardrobe_cabinet	11411	stand	9707	trash_can	11
storage_bench	2	double_bed	8065	wood_board	6
basketball_hoop	16	single_bed	3418	baby_bed	696
sink	44	fence	30	table	4
coffee_table	1979	fireplace	195	chair_set	2
goal_post	13	shoes_cabinet	744	toilet	25
kitchen_set	2	office_chair	2672	bathtub	17
workplace	122	bunker_bed	953	gym_equipment	195
dining_table	190	table_and_chair	28		

Table A.2: Class counts for medium rooms under model $\mathcal{M}_1$

Object Class	**Count**	**Object Class**	**Count**	**Object Class**	**Count**
desk	3645	dresser	3240	fireplace	309
car	22	sink	52	bench_chair	10
shower	19	tv_stand	2798	armchair	3185
ottoman	3028	dressing_table	2890	motorcycle	16
game_table	49	kitchen_cabinet	755	tripod	3
wardrobe_cabinet	10299	stand	8647	trash_can	9
storage_bench	8	wood_board	17	basketball_hoop	27
single_bed	2721	baby_bed	576	double_bed	7484
fence	29	coffee_table	2696	chair	2147
chair_set	7	goal_post	10	shoes_cabinet	966
toilet	34	kitchen_set	18	sofa	2340
office_chair	2532	bathtub	21	table_and_chair	91
bunker_bed	738	gym_equipment	287	dining_table	256
drinkbar	1	workplace	245		

Table A.3: Class counts for big rooms under model $\mathcal{M}_1$

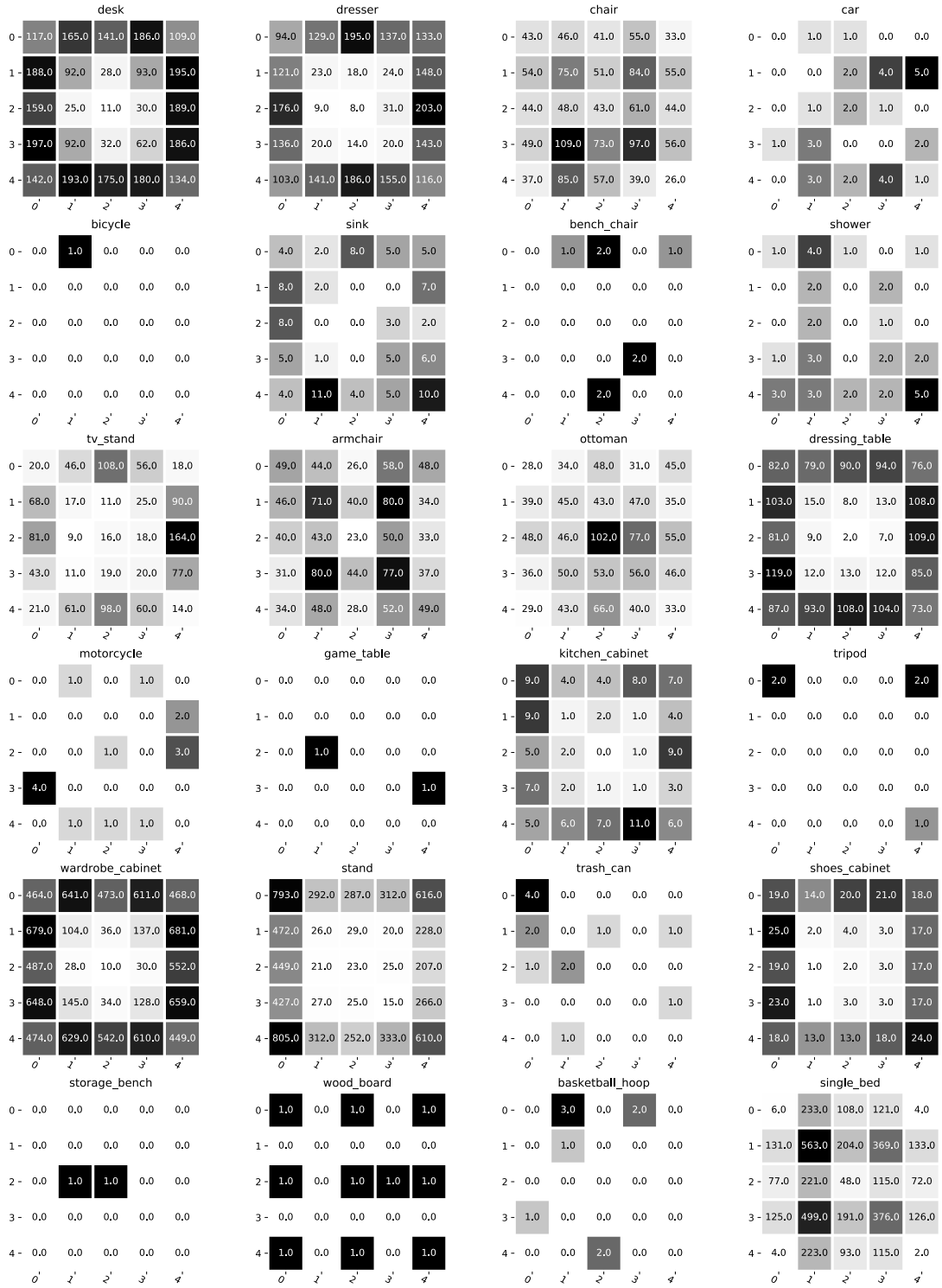


Figure A.1: Cell grid with object count per class for small Rooms $\mathcal{M}_1$

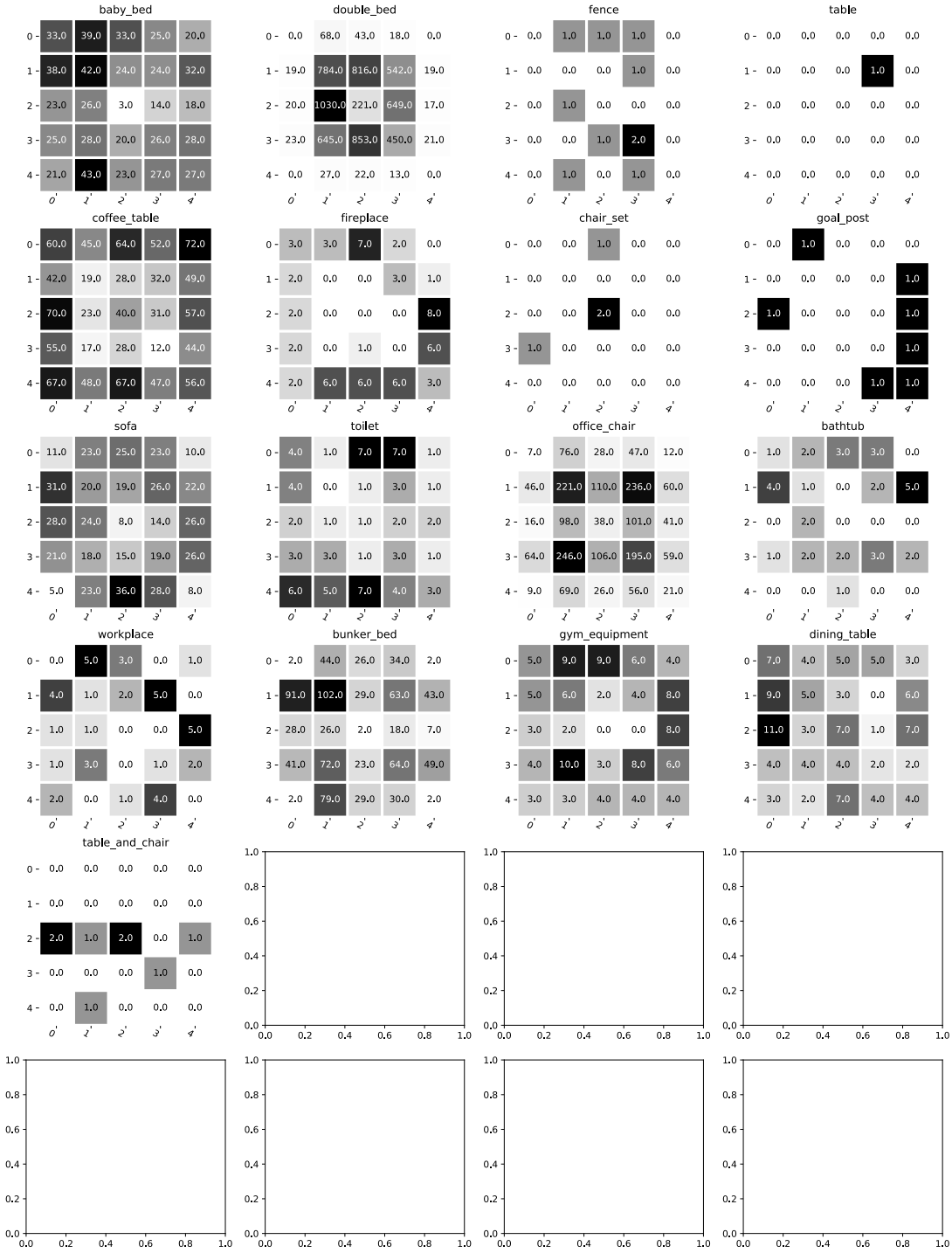


Figure A.2: Cell grid with object count per class for small Rooms $\mathcal{M}_1$

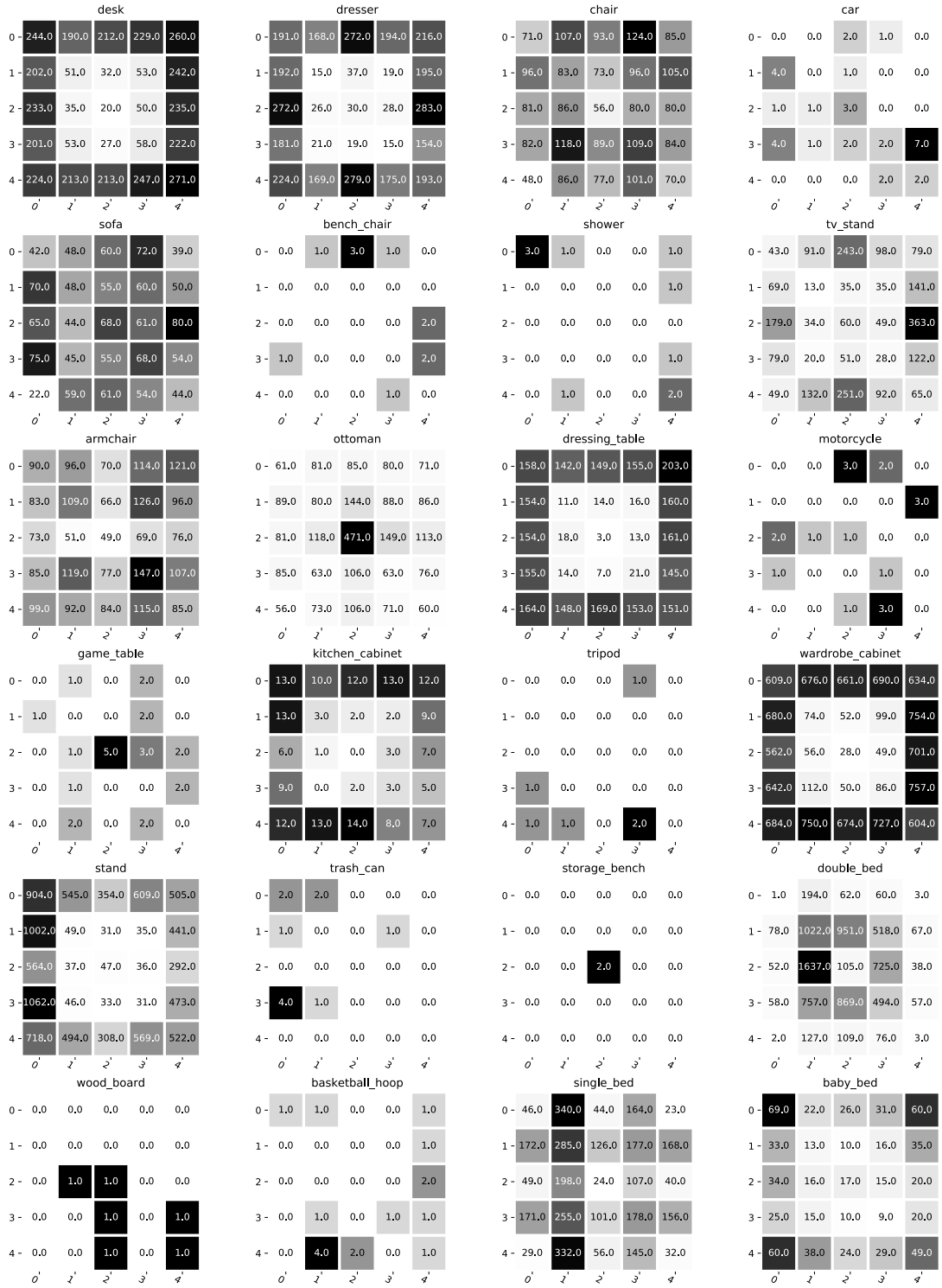


Figure A.3: Cell grid with object count per class for Medium Rooms $\mathcal{M}_1$

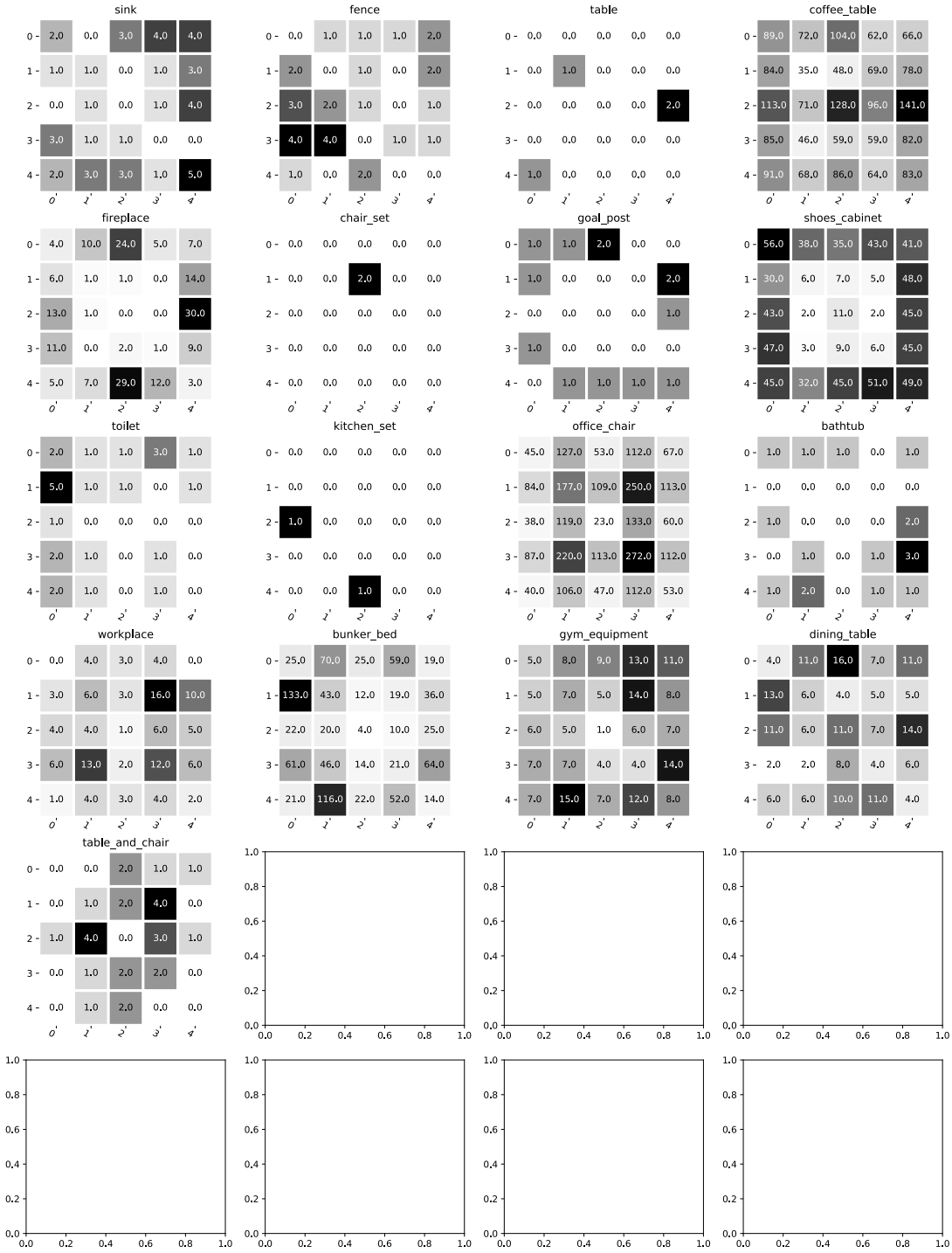


Figure A.4: Cell grid with object count per class for Medium Rooms $\mathcal{M}_1$

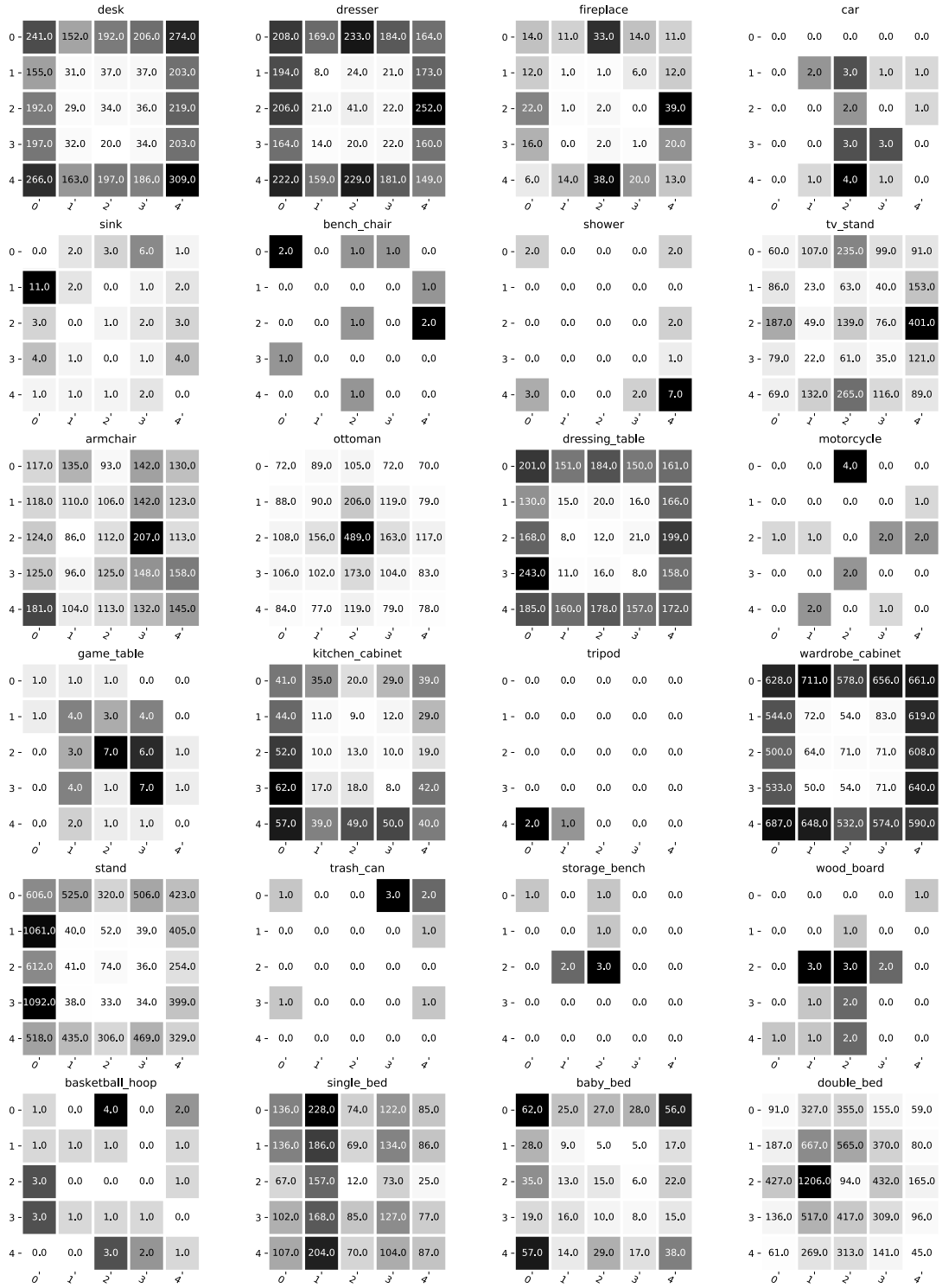


Figure A.5: Cell grid with object count per class for Big Rooms $\mathcal{M}_1$

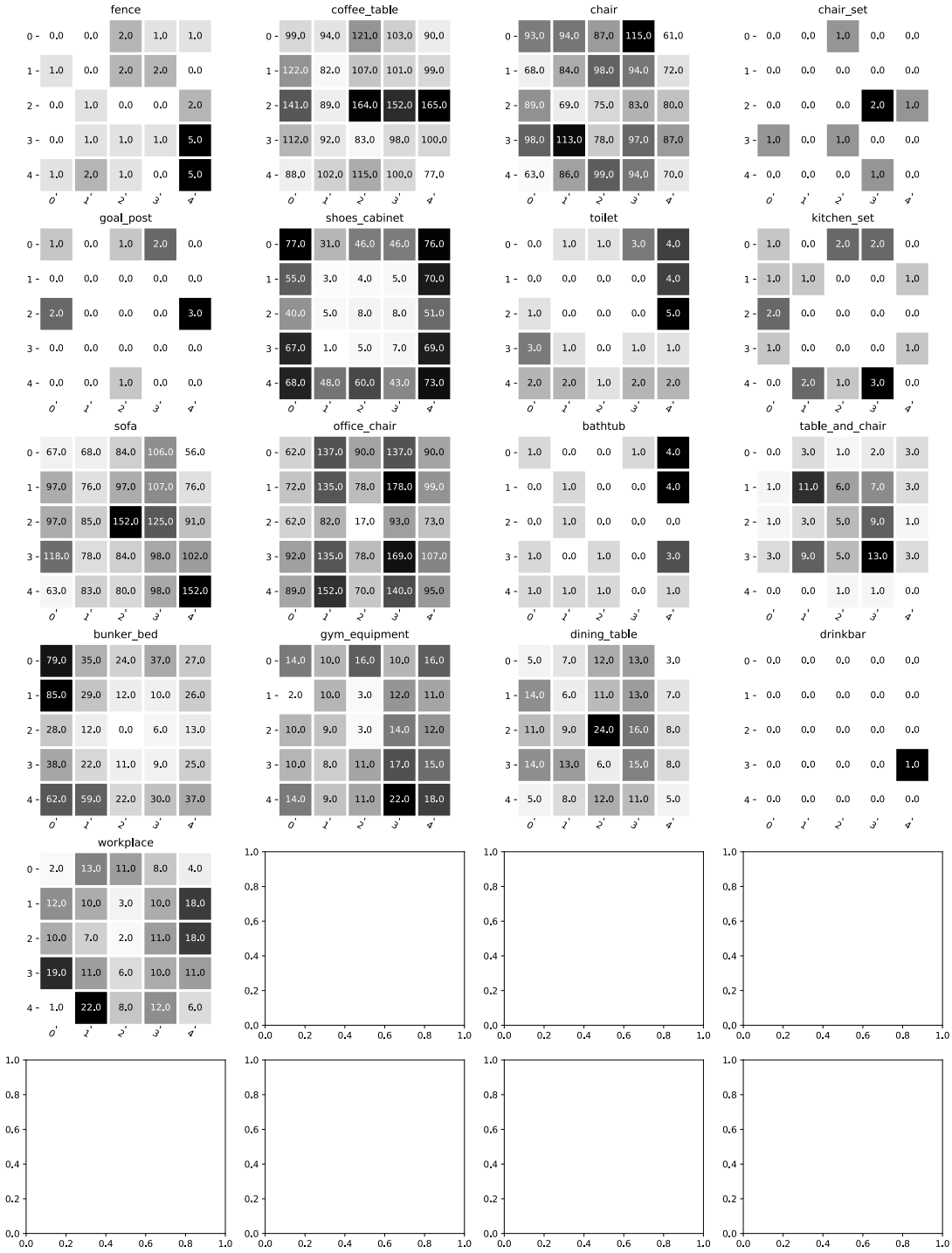


Figure A.6: Cell grid with object count per class for Big Rooms $\mathcal{M}_1$

A.2 Statistics model $\mathcal{M}_2$

Object Class	Count	Object Class	Count	Object Class	Count
desk	900	dresser	1623	chair	938
car	33	bicycle	1	sink	105
bench_chair	8	shower	37	tv_stand	1171
armchair	1165	ottoman	1175	dressing_table	1582
motorcycle	15	game_table	2	kitchen_cabinet	115
tripod	5	wardrobe_cabinet	9719	stand	2387
trash_can	13	shoes_cabinet	318	storage_bench	2
wood_board	10	basketball_hoop	9	single_bed	2326
baby_bed	662	double_bed	3915	fence	10
table	1	coffee_table	1125	fireplace	63
chair_set	4	goal_post	7	sofa	509
toilet	73	office_chair	234	bathtub	34
workplace	42	bunker_bed	908	gym_equipment	120
dining_table	112	table_and_chair	8		

Table A.4: Class counts for independent objects in small rooms under model $\mathcal{M}_2$

Motif	Count	Motif	Count
233c3c40-bed5-4361-8f9a-ad96428619b2	462	fd9f9408-1ff1-47db-8228-0534e4cec7ca	340
4c87a529-5454-4515-872c-8ad580a32df7	132	a4b565e4-d3e9-41cf-94fd-374c9aba5ec0	465
c34751e4-ac2f-41b6-9e0d-fefe1b1f4c03	95	78b55f48-9599-4cf2-99e6-d3ab3499ced4	98
b87f69a5-e526-4bdd-9a5c-ec34127fdd5e	372	84fc9496-3e7d-4ced-b048-debe9515598a	237
8ea30cdd-78a6-4232-a424-4737a3866214	115	0c36a617-bf63-4c36-baef-a95455331e20	237
cfe70ef8-a93c-4641-b728-d64bc9abffdc	73	2e0a141d-d0a4-4bbe-9ac2-be611a8597ca	573
99456681-3f1a-4c48-ba76-dd6c0f9dc0f7	89	8f21bb71-ee45-4e1c-bb93-176dca1573dd	166
f9f66d53-28af-46be-b28c-118bd9ee2d6f	139	a39b1fa7-17bc-4da7-a564-d5f454e4b2f7	85
7367b48c-9b5b-402f-9b75-216ca00c97e8	58	692d2588-cbf0-48e0-b641-aa8844ad1fdc	1664
c62a116b-6f1f-4f8c-b936-bb28916e984e	95	3ca4fb05-d61d-458f-b5ff-b16165569c00	106
9875db7f-1af8-4b81-b885-8b50fec340ed	90	ad87619d-7d2b-4412-ab06-4e84faf988d7	367
03943e84-fd0b-42b1-8970-42367069c3c9	55		

Table A.5: Motifs occurrences counts in small rooms under model $\mathcal{M}_2$

Object Class	Count	Object Class	Count	Object Class	Count
desk	1161	dresser	2935	chair	1601
car	33	sofa	1399	bench_chair	11
shower	10	tv_stand	2421	armchair	1671
ottoman	2556	dressing_table	2638	motorcycle	18
game_table	24	kitchen_cabinet	179	tripod	6
wardrobe_cabinet	11411	stand	2613	trash_can	11
storage_bench	2	double_bed	4613	wood_board	6
basketball_hoop	16	single_bed	1645	baby_bed	696
sink	44	fence	30	table	4
coffee_table	1821	fireplace	195	chair_set	2
goal_post	13	shoes_cabinet	744	toilet	25
kitchen_set	2	office_chair	395	bathtub	17
workplace	122	bunker_bed	953	gym_equipment	195
dining_table	190	table_and_chair	28		

Table A.6: Class counts for independent objects in medium rooms under model $\mathcal{M}_2$

Motif	Count	Motif	Count
f6ff3122-983f-4f89-9426-427778ec27bc	204	bc092a6f-0083-4a84-927b-f4add001bd71	104
ebc2451d-1554-42d4-833f-fc0dd5cacbd9	130	fd007943-1b79-452a-9ccf-39982c66f00a	40
d7751799-a7c7-4bb9-a985-f09ff079ae03	465	15517aa8-62ec-44ed-a24f-b76cb1ceaafe	40
5f6bf163-1090-4879-8958-688aaa9af1d4	158	ffb06229-67c2-48d3-8efd-ff3bca3b8e3e	56
7024f768-1295-4431-a106-c285406161d5	54	5133c79c-dda3-4e1d-b810-d53fdd94178a	38
4bd692d5-bebc-4da8-b8db-135a56bc897a	153	20519d7d-921f-46eb-9790-af132c39f070	133
98a750ba-7265-40fe-a177-dde56dbebbaf	270	736f8a3e-79c0-42c4-a922-2bdb8e1f6456	41
2aa61777-2459-4a17-b42e-7882a3cf7b4c	79	aee028df-d7f3-4ee8-9d39-a03f530d65a6	44
f715e71e-c781-4cc0-b3ae-cb30124667e5	24	3fdf4331-9ad0-4bac-8d25-190b4646b441	906
f1e50d99-283e-40a2-ade6-87c03dcfb494	35	cae7d1c4-df61-4561-9b15-3518ba50008d	290
d441fda1-5379-438f-a480-b057e4be7186	162	44b88cfd-d866-47d4-847b-2a47299a971a	397
f7765cad-e3db-4258-8745-fa14235b9626	342	e13b9781-f22b-4327-adb1-cc60698f2026	2685
a952390d-afc8-4f24-b8fe-335fde6a0891	59	d23da005-97dc-4567-848d-d83cf378ded4	22
63c50d51-6d77-4ed6-a76f-f3fea5fcb773	305	5bccca56-6859-4580-9b8d-0f4987423d3e	60
5ae0633a-bf14-4228-bba3-01800c5f044e	500	4709f83d-b46b-421f-90d1-04f4963a7834	80
6a949ee8-deb5-432c-bf4a-71a629a64ae4	132	cf427cae-1445-4247-b824-6b2272ccaa85	95

Table A.7: Motifs occurrences counts in medium rooms under model $\mathcal{M}_2$

Object Class	Count	Object Class	Count	Object Class	Count
desk	1109	dresser	2778	chair	1691
car	22	sink	52	bench_chair	10
shower	19	tv_stand	2558	armchair	1984
ottoman	3028	dressing_table	2890	motorcycle	16
game_table	49	kitchen_cabinet	755	tripod	3
wardrobe_cabinet	10299	stand	2424	trash_can	9
storage_bench	8	wood_board	17	basketball_hoop	27
single_bed	1308	baby_bed	576	double_bed	4278
fence	29	coffee_table	2006	table_and_chair	91
fireplace	309	chair_set	7	goal_post	10
shoes_cabinet	966	toilet	34	kitchen_set	18
sofa	1317	bathtub	21	workplace	245
bunker_bed	738	gym_equipment	287	dining_table	256
drinkbar	1	office_chair	452		

Table A.8: Class counts for independent objects in big rooms under model $\mathcal{M}_2$

Motif	Count	Motif	Count
ca925e25-148a-4585-a426-a9f07fbeac16	90	7af4edaa-6b89-4199-a33a-3ea354191acd	161
91cf2a6a-4939-4945-b0b6-9e8b93f6276d	30	f4a355ad-2ddf-4fd9-9e95-28ebad631271	23
ea8fe884-575a-4860-885e-366e928ac7b5	25	ff956359-2d07-444d-b9a9-f68bf30c8365	262
dcb9fcb0-bdea-4fe5-882a-9a7a58c28a4c	60	58b86a16-c7b0-4193-88f6-ed4fb73dbba4	155
170c8c01-2029-4b81-b7df-6938b6cd1009	309	c666b0a0-b1e2-45c1-a288-4c9cbb7f2310	30
492a1079-d9af-4c64-9d1d-bad5a8cee27d	43	86dc6bd9-0f67-4191-826e-b8cc05ae50f7	57
99a327df-48fe-41ae-a5bf-c7cf5b00d488	45	1f29cba1-dc69-4b3d-a0cf-f27494c3ecc7	48
680e60aa-f58e-4042-b51e-b3e3ce9f1e9f	2345	51103616-50d2-4f4b-814e-6f0c02816eef	244
38fcfe54-723b-4cc0-aa7c-a04f47d5a6c3	832	1e966779-bf68-482b-a508-48c82aa6f690	77
ca17353b-7036-4c0c-8981-13261152f861	62	6d99557e-6759-4e4b-94e8-e865bd64183f	38
ceece28c-d342-4bfb-b12b-21f87961b8ce	60	9ca6b7a8-b6b0-4461-880b-084817e7e558	473
53907426-c2e6-45e4-9e87-b4de4d3307ab	47	47c122ae-f43e-4a7f-9e26-3ce539bfec85	317
57b4278d-8dee-4067-9948-2e8fd45f3d15	353	954dc8c3-0f20-4b30-863d-bb3c02c3d439	156
f2c2d2c1-cf06-45ea-b5fd-cd5c1c0708a7	399	8bee4038-d215-4053-a30f-cd38ec048213	74
1b05d50a-fa4e-41c7-9f42-07b7cf82e5ca	51	dacd76f3-a7b2-450d-9403-186f29a63ae6	126
4fc4336e-411d-48a4-a127-5ca8213fb580	351	a4278932-c148-4d01-a510-bac8fd3848f3	126
0c5a94a1-1400-422b-bd34-3f20e5fa0333	141	61787d43-6a95-4734-b0de-5b758f339138	240
644ba11f-3931-4881-931e-e84d3689cfb2	45	4aadb471-ab02-4c96-819c-d58117d4965e	55
2adae9d6-07cc-4b6f-b756-c786ed9c1568	57	4b2db7b6-640b-400a-853d-5139606dd0cf	160
0ead0e91-2d1f-4ae4-be47-a6568e870d64	178	7204631a-071e-49d4-bf64-8abbbc3cf34b	48

Table A.9: Motifs occurrences counts in big rooms under model $\mathcal{M}_2$

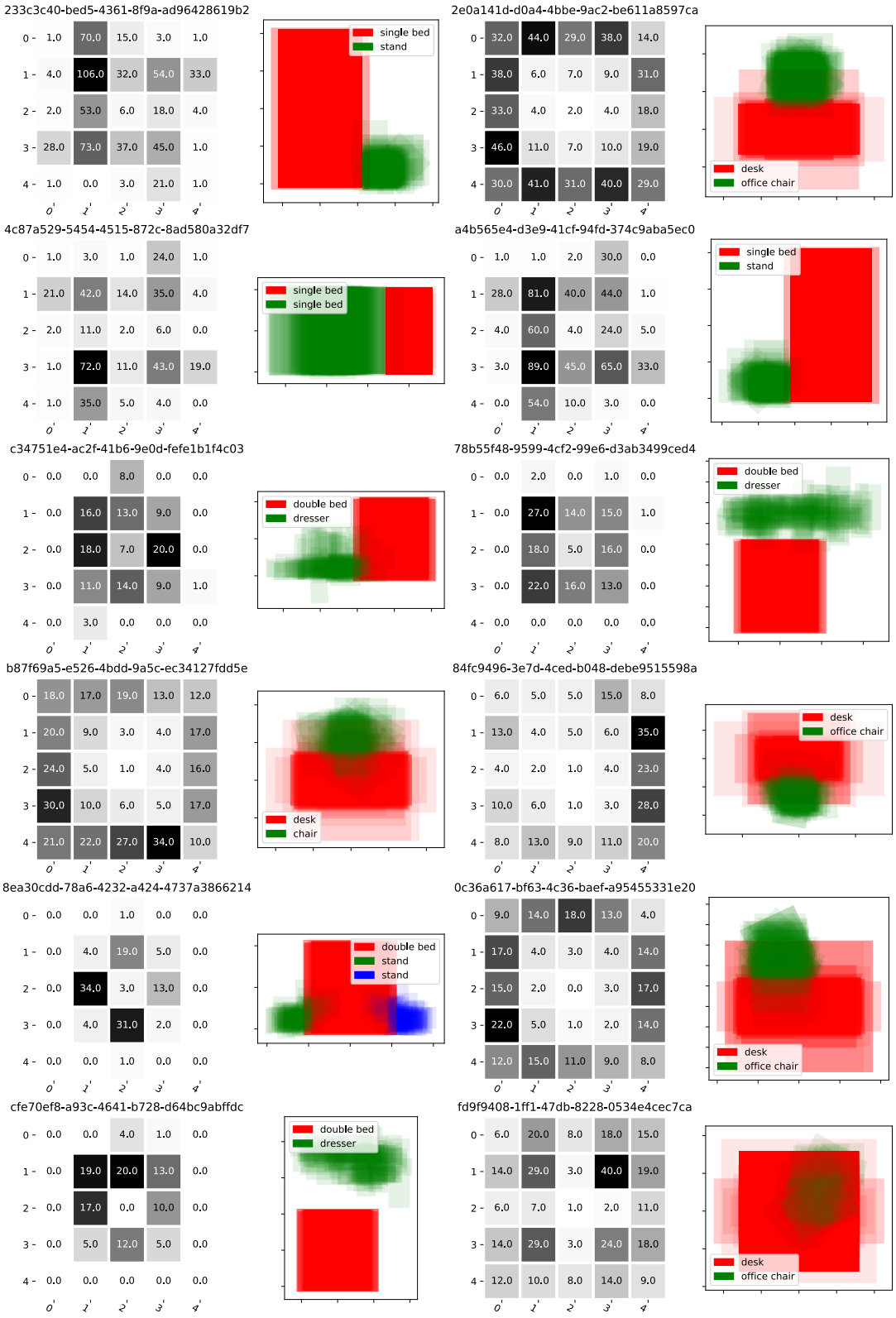


Figure A.7: Count of Motifs occurrences in grid for small Rooms $\mathcal{M}_2$

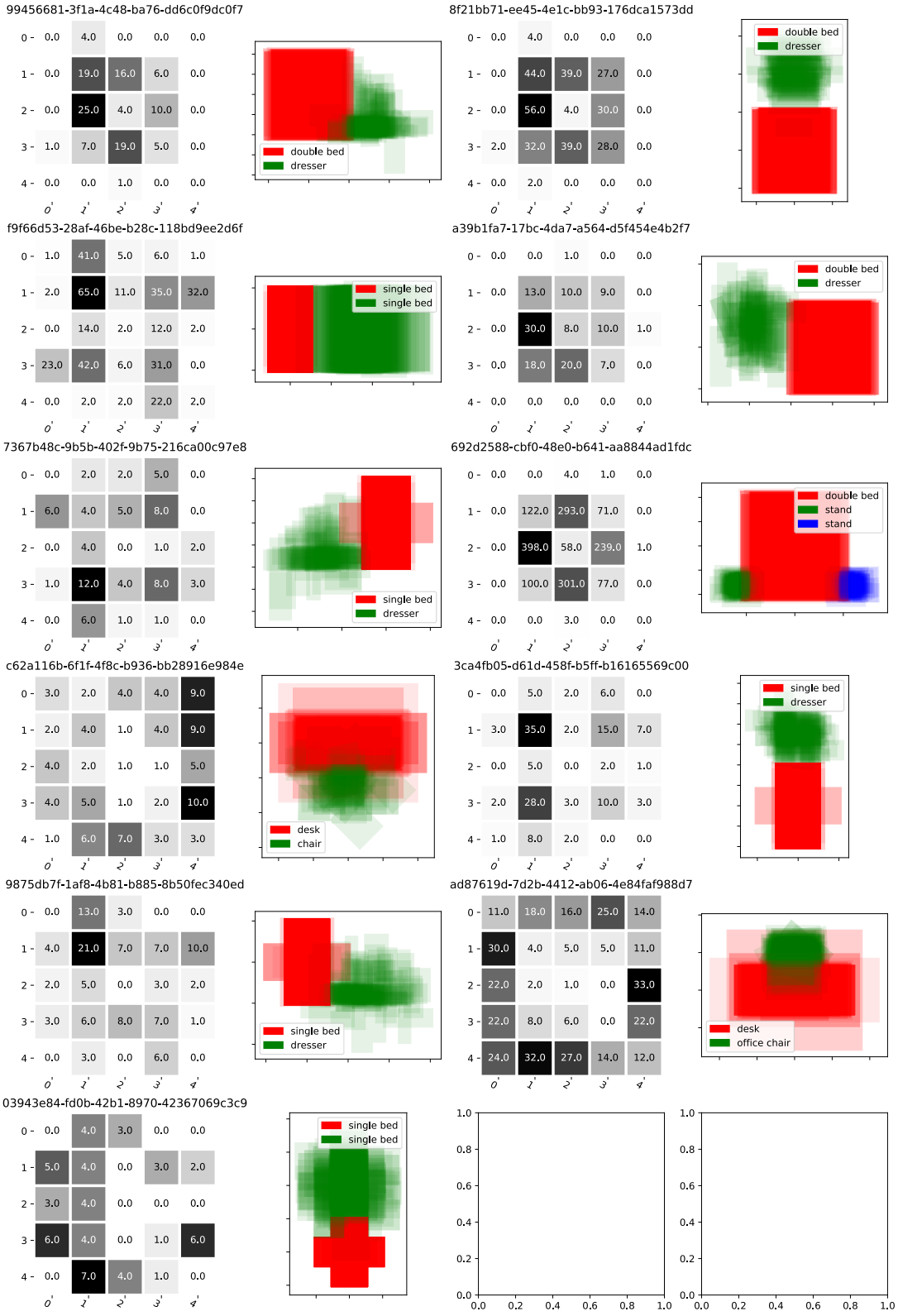


Figure A.8: Count of Motifs occurrences in grid for small Rooms $\mathcal{M}_2$

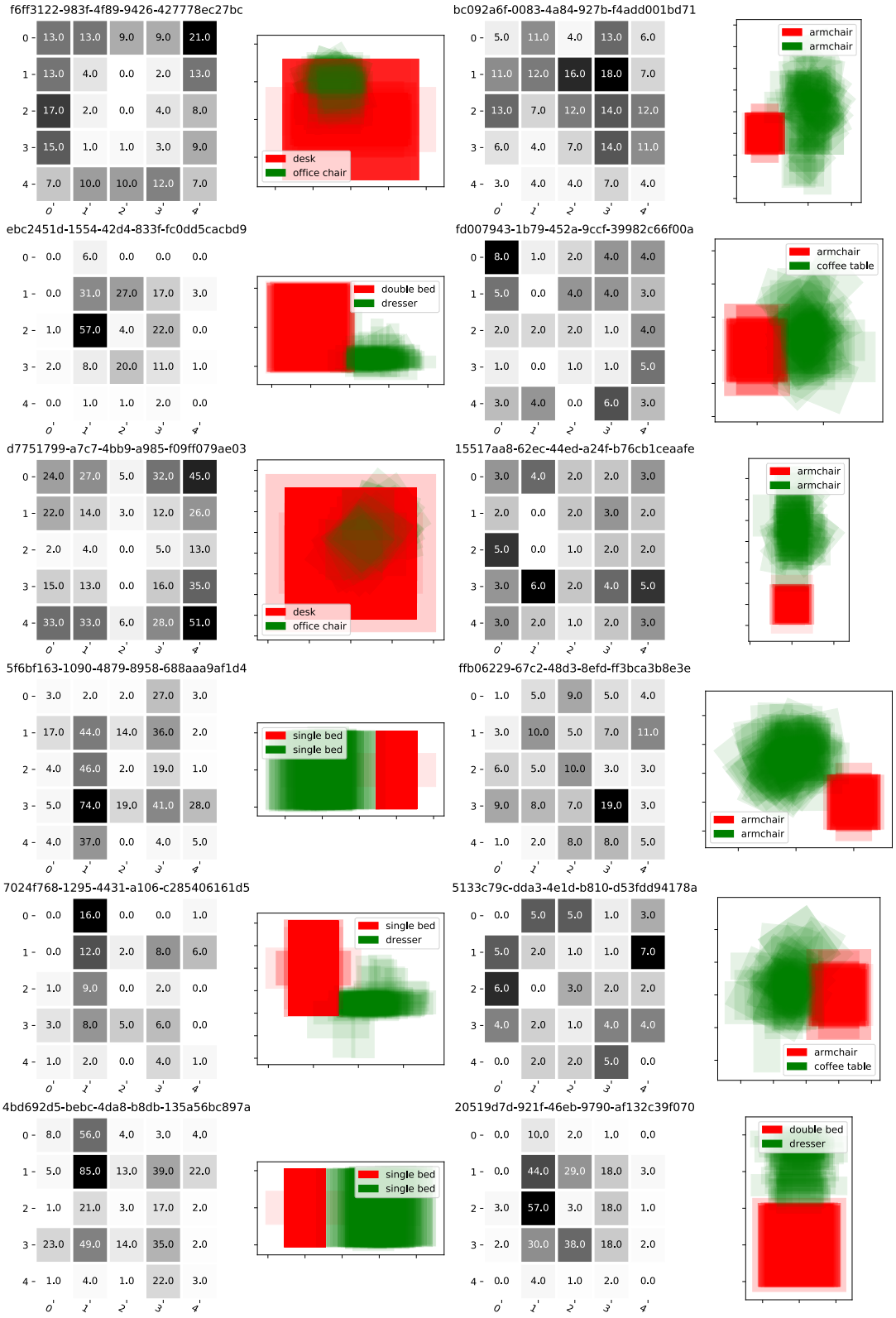


Figure A.9: Count of Motifs occurrences in grid for medium Rooms $\mathcal{M}_2$

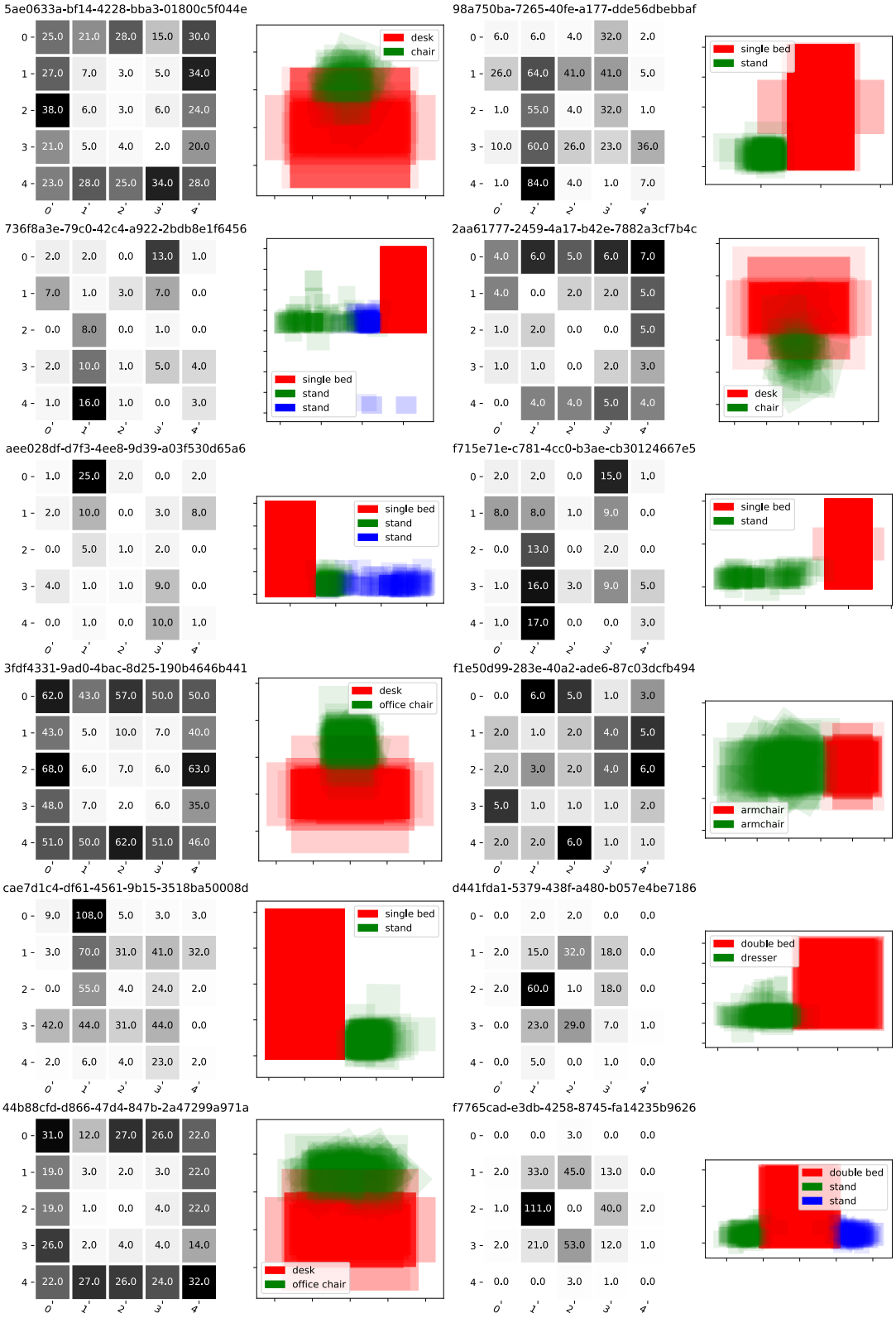


Figure A.10: Count of Motifs occurrences in grid for medium Rooms $\mathcal{M}_2$

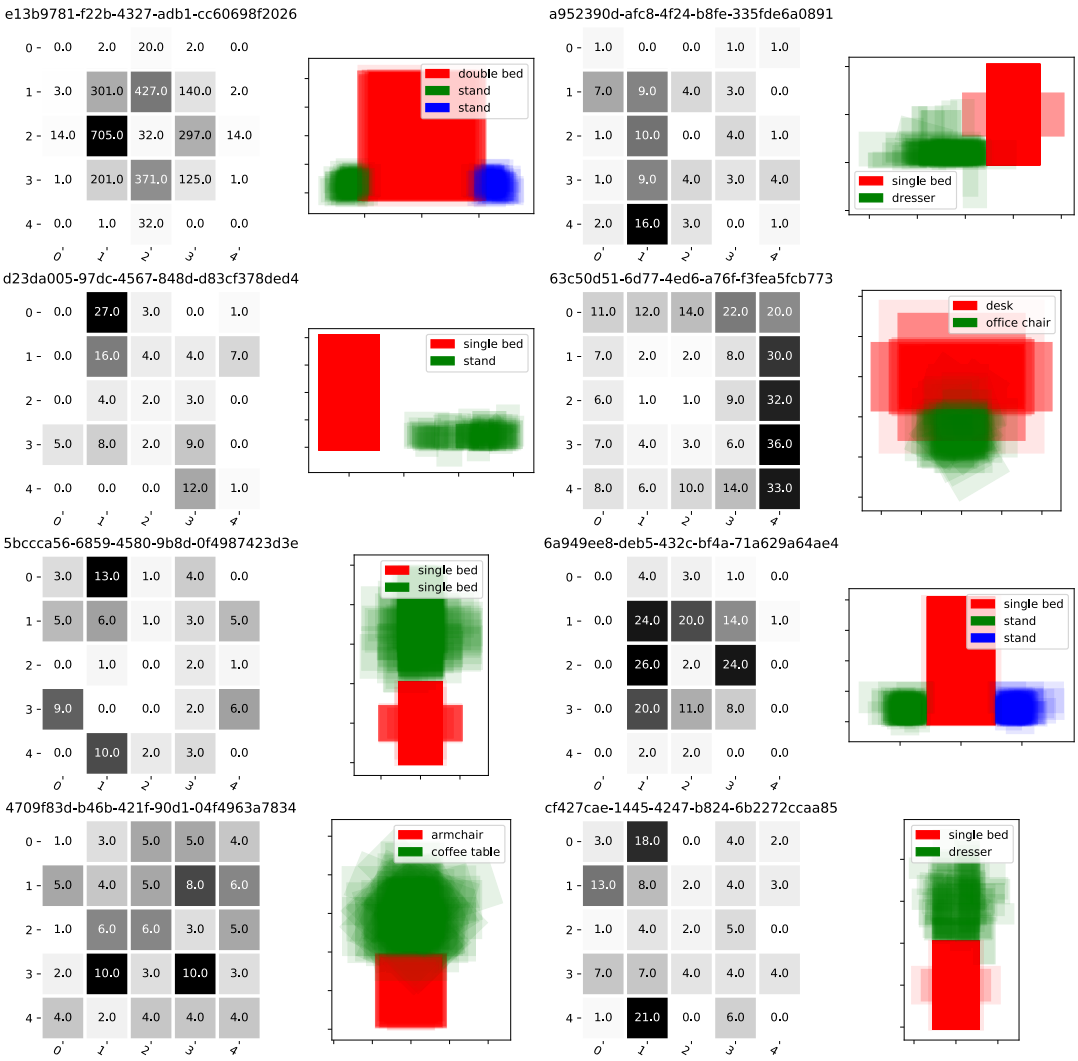


Figure A.11: Count of Motifs occurrences in grid for medium Rooms $\mathcal{M}_2$

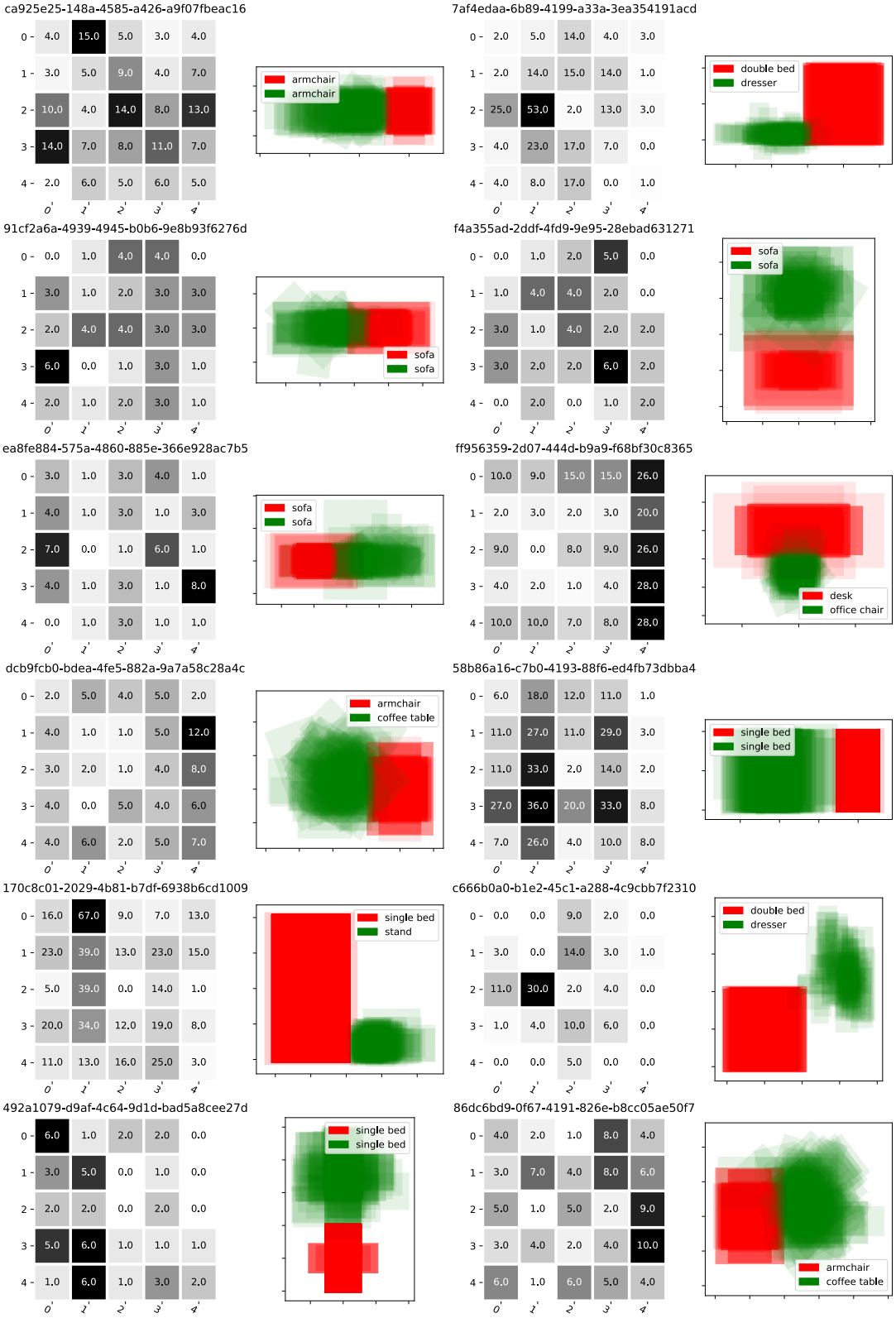


Figure A.12: Count of Motifs occurrences in grid for big Rooms $\mathcal{M}_2$

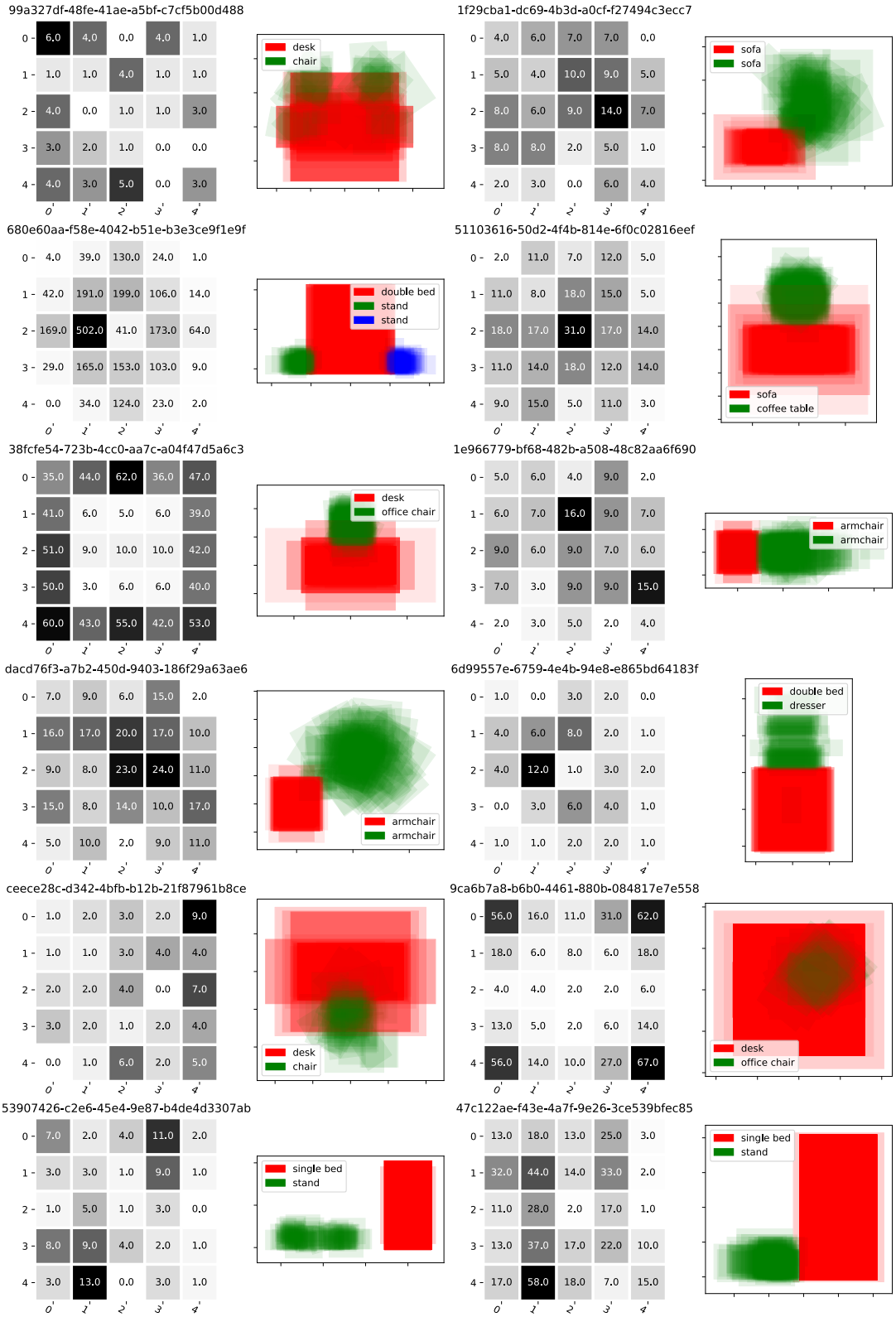


Figure A.13: Count of Motifs occurrences in grid for big Rooms $\mathcal{M}_2$

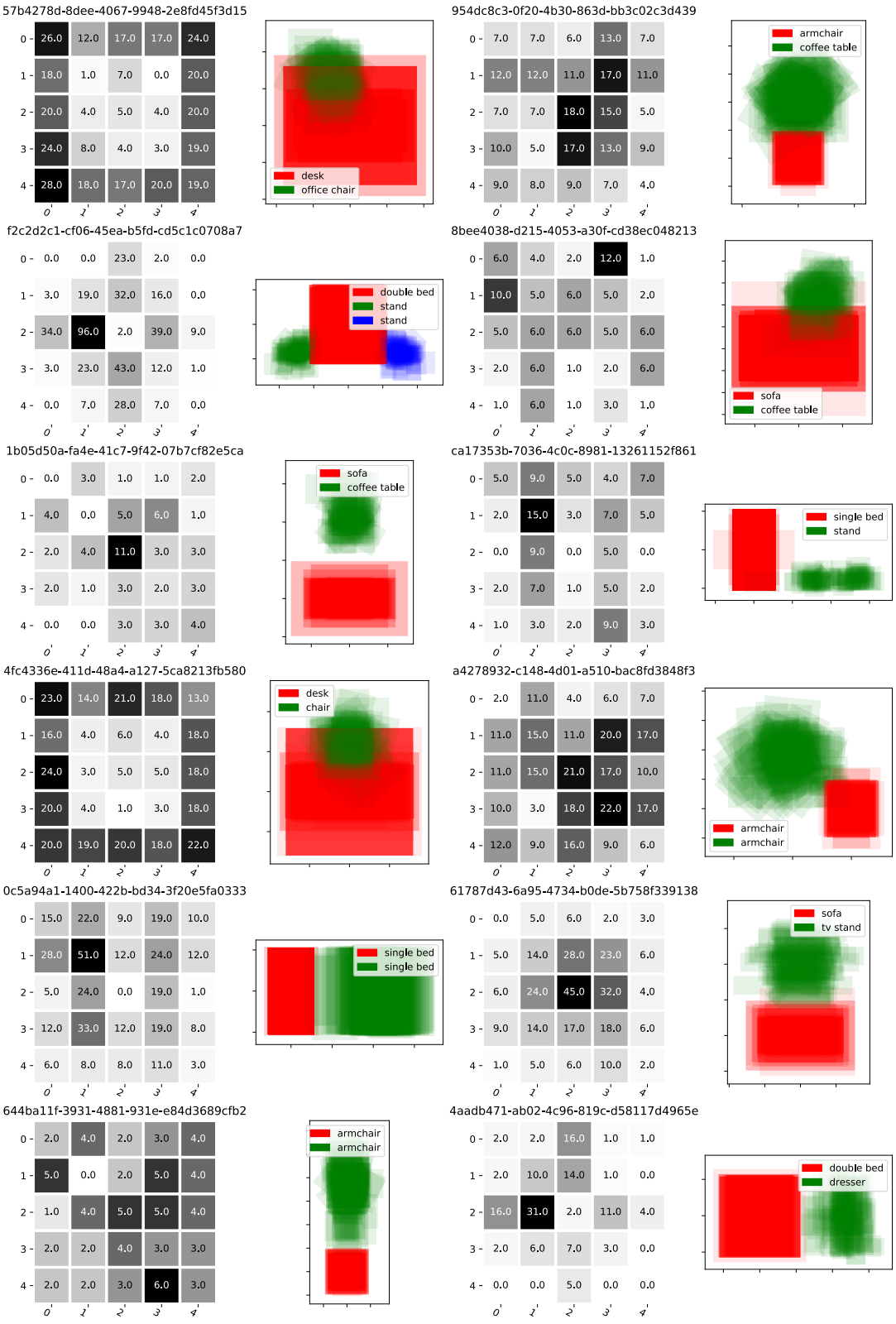


Figure A.14: Count for Motifs occurrences in grid for big Rooms $\mathcal{M}_2$

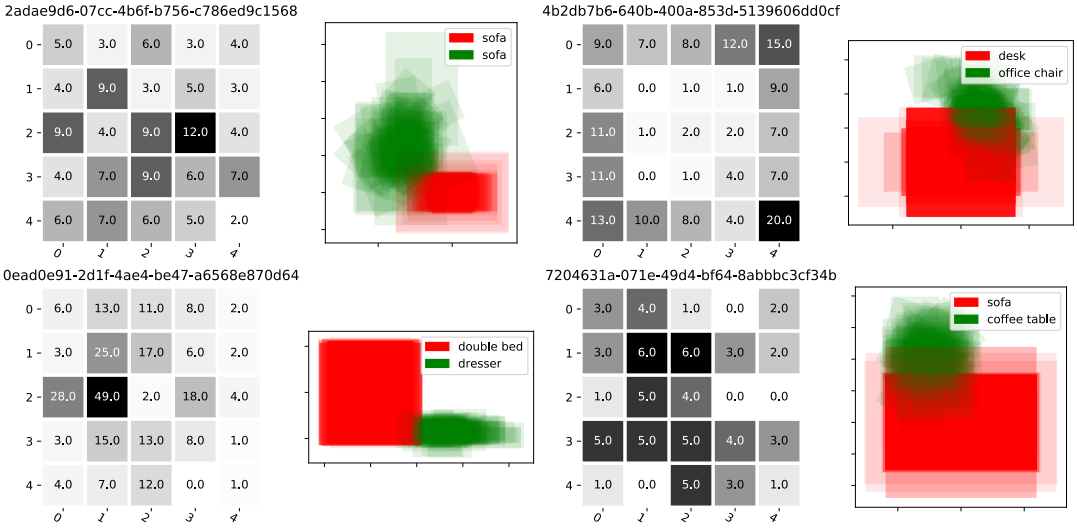


Figure A.15: Count for Motifs occurrences in grid for big Rooms $\mathcal{M}_2$

Appendix B

Learning Objects Arrangements in Practice

B.1 List of Class Clusters

- ('bed/double_bed', 'stand/stand', 'stand/stand')
- ('bed/double_bed', 'dresser/dresser')
- ('bed/single_bed', 'stand/stand', 'stand/stand')
- ('bed/single_bed', 'stand/stand')
- ('bed/single_bed', 'dresser/dresser')
- ('bed/single_bed', 'bed/single_bed')
- ('desk/desk', 'chair/office_chair')
- ('desk/desk', 'chair/chair')
- ('workplace/workplace', 'chair/office_chair')
- ('workplace/workplace', 'chair/office_chair', 'chair/office_chair')
- ('sofa/sofa', 'sofa/sofa')
- ('sofa/sofa', 'tv_stand/tv_stand')
- ('sofa/sofa', 'table/coffee_table')
- ('chair/armchair', 'chair/armchair')
- ('chair/armchair', 'table/coffee_table')
- ('music/piano', 'ottoman/ottoman')
- ('vehicle/car', 'vehicle/car')
- ('gym_equipment/gym_equipment', 'gym_equipment/gym_equipment')

B.2 Learning with Diagonal Covariances

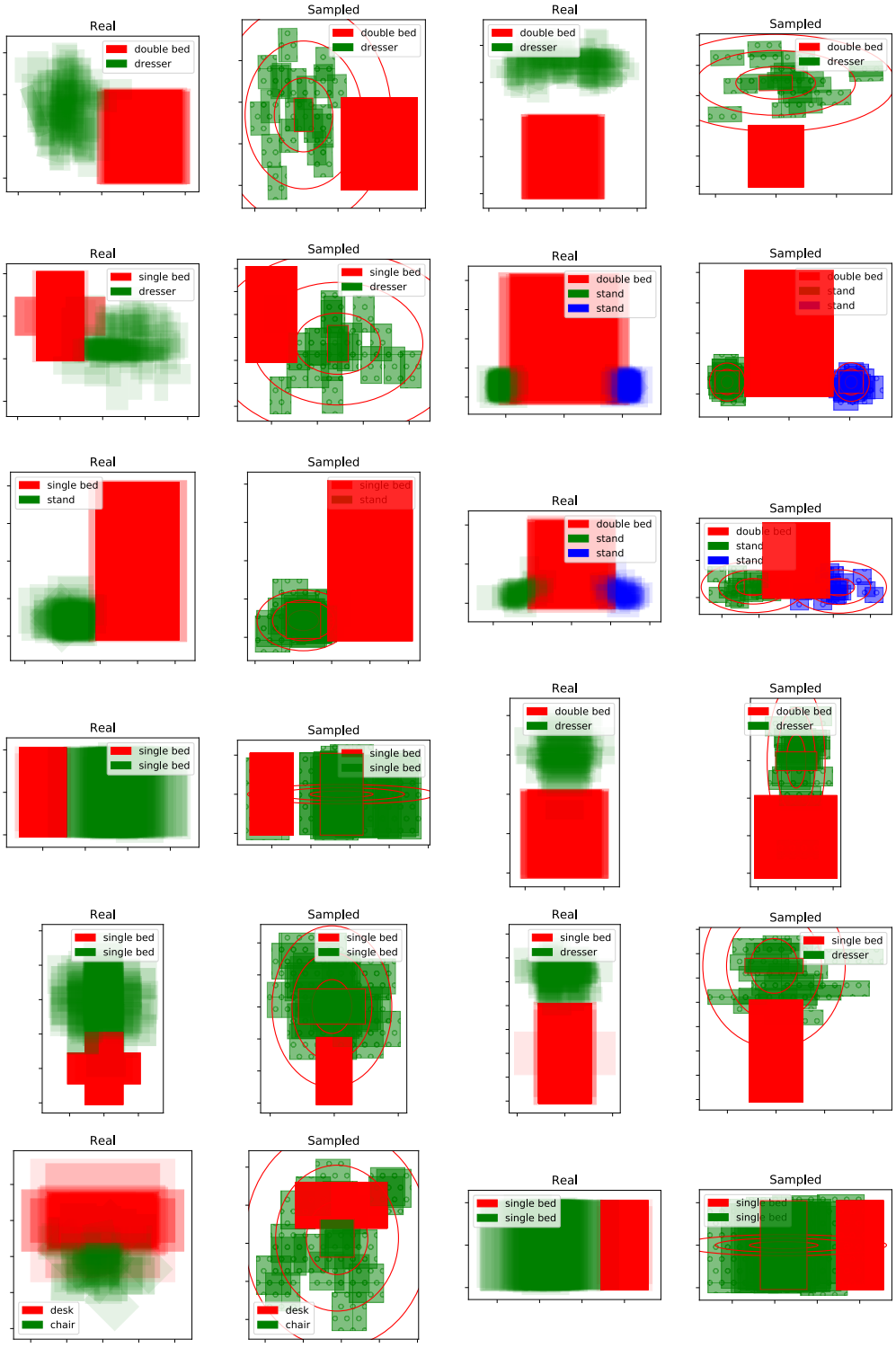


Figure B.1: Motifs for small rooms learned with diagonal covariance matrices. Real occurences and samples

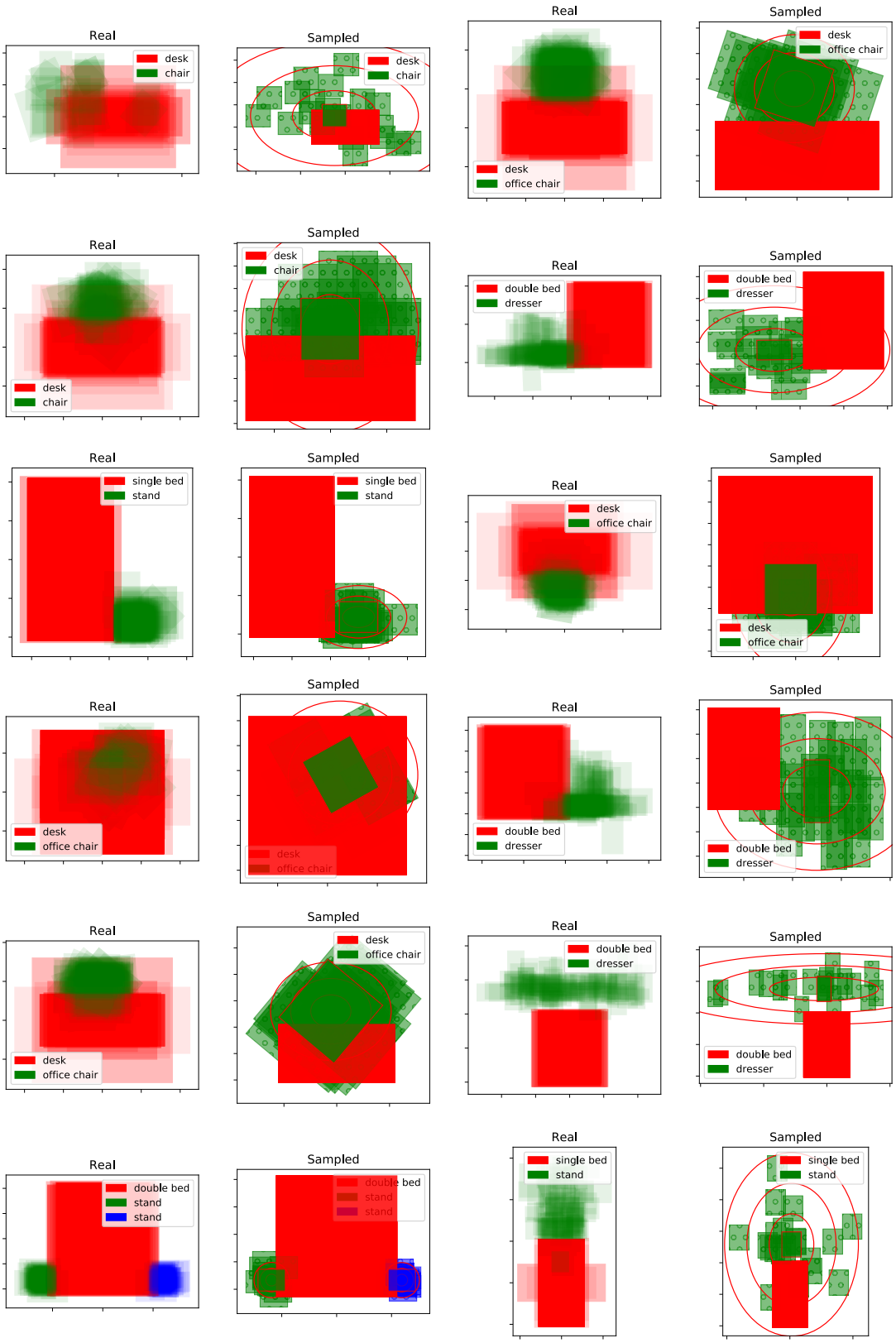


Figure B.2: Motifs for small rooms learned with diagonal covariance matrices. Real occurences and samples

B.3 Learning with Full Covariances

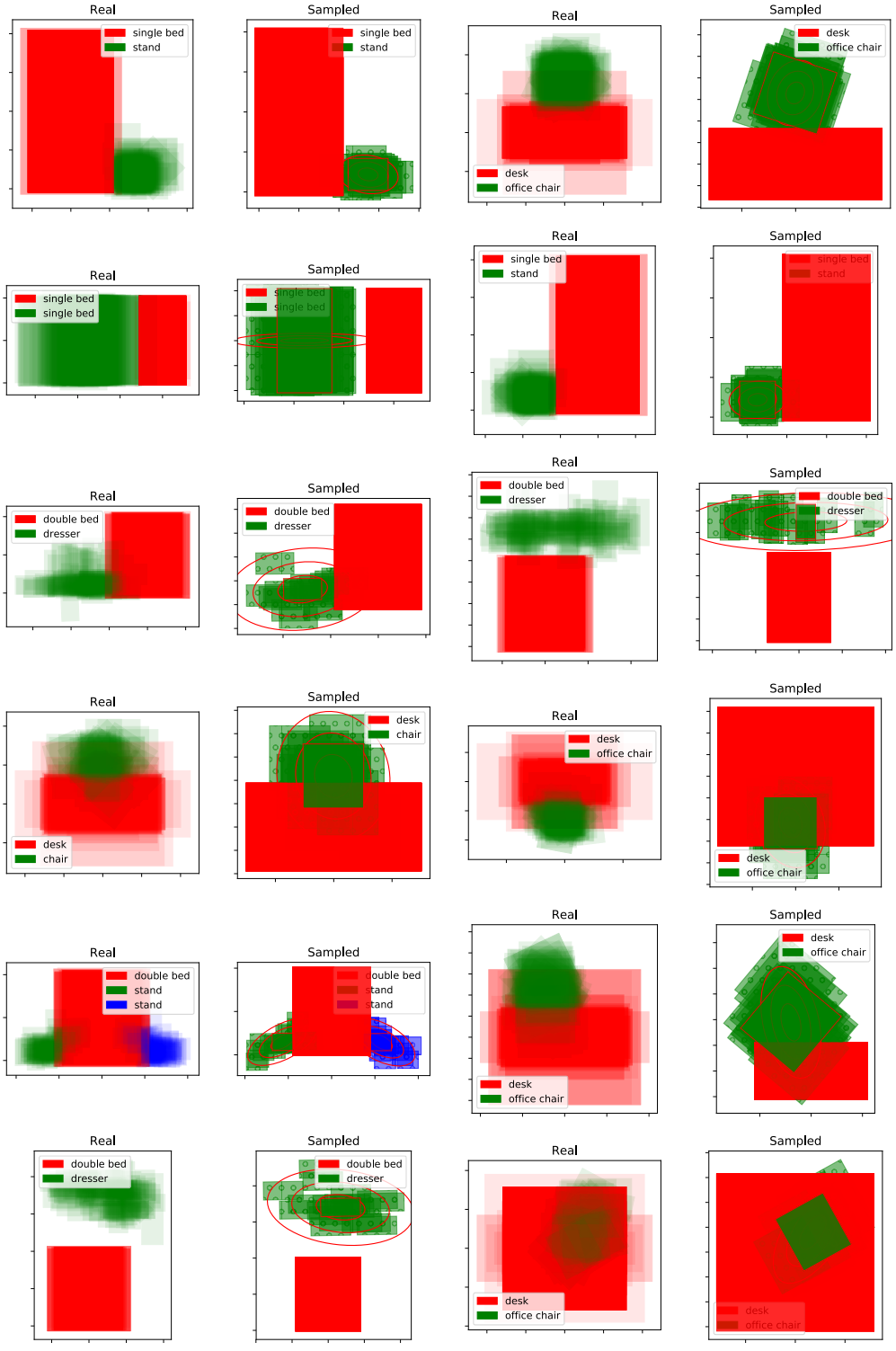


Figure B.3: Motifs for small rooms learned with full covariance matrices. Real occurences and samples

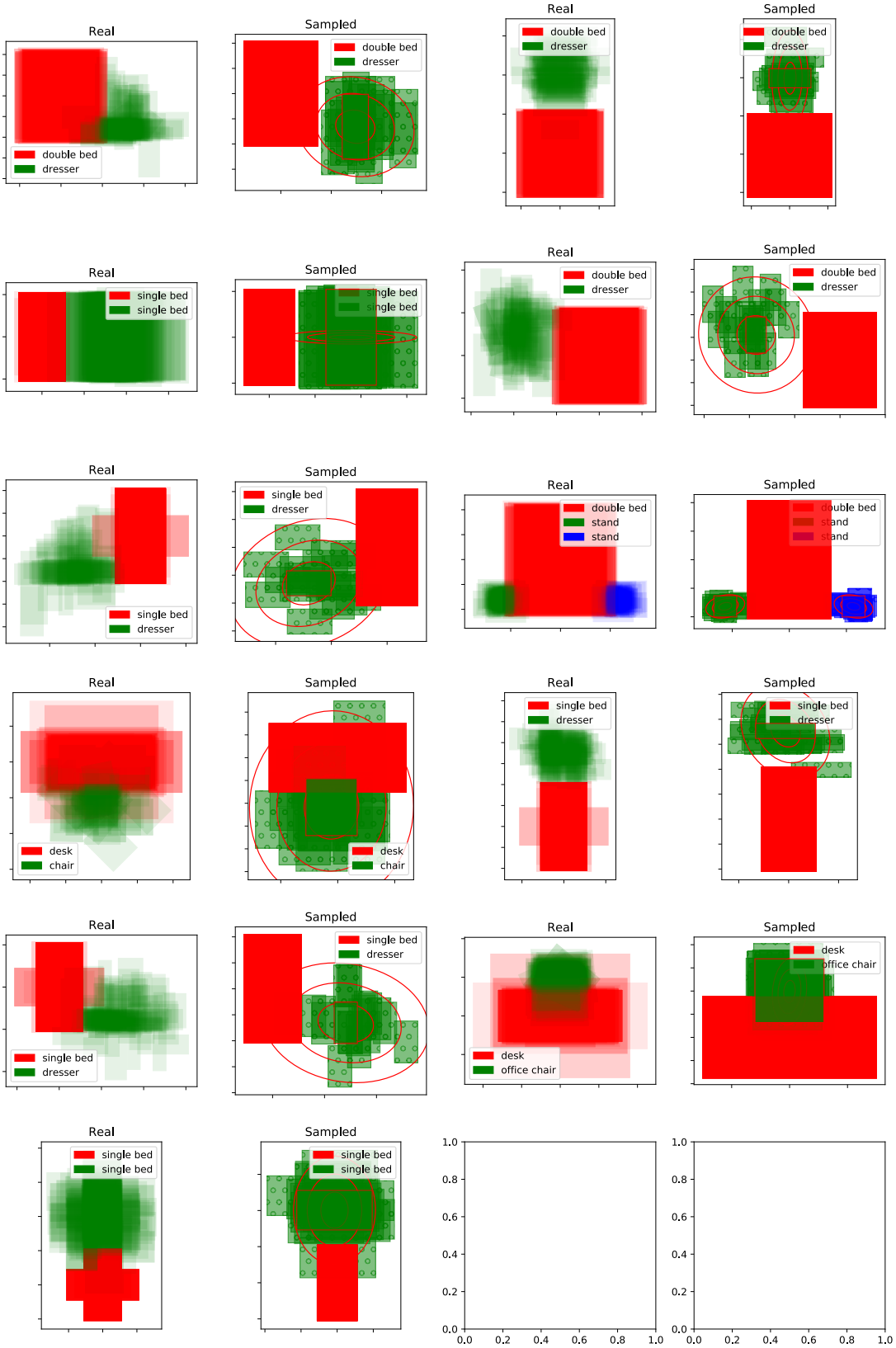


Figure B.4: Motifs for small rooms learned with full covariance matrices. Real occurences and samples

B.4 Fitting Rotations to Model

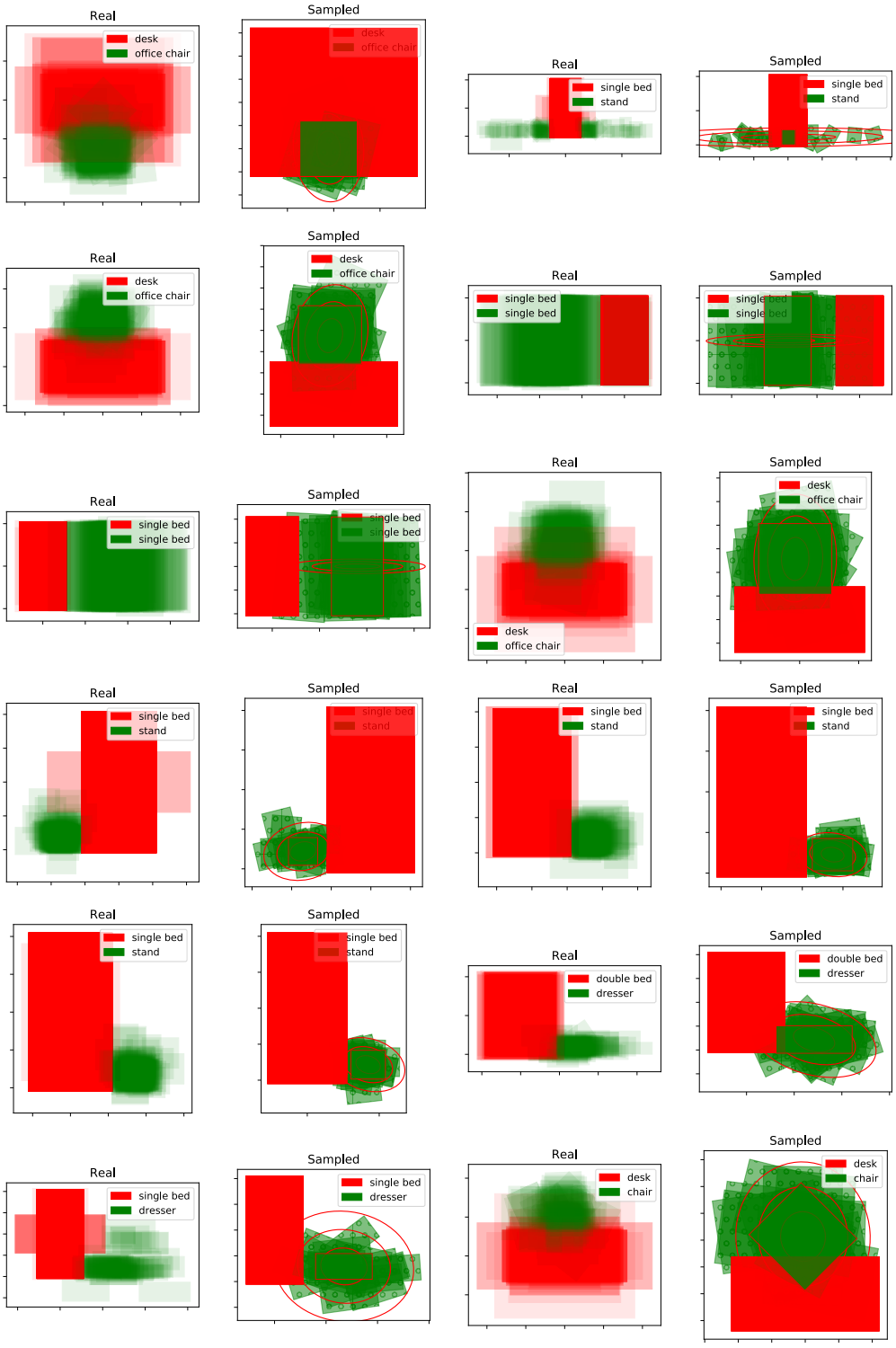


Figure B.5: Motifs for small rooms learned fitting rotations and with full covariance matrices. Real occurences and samples

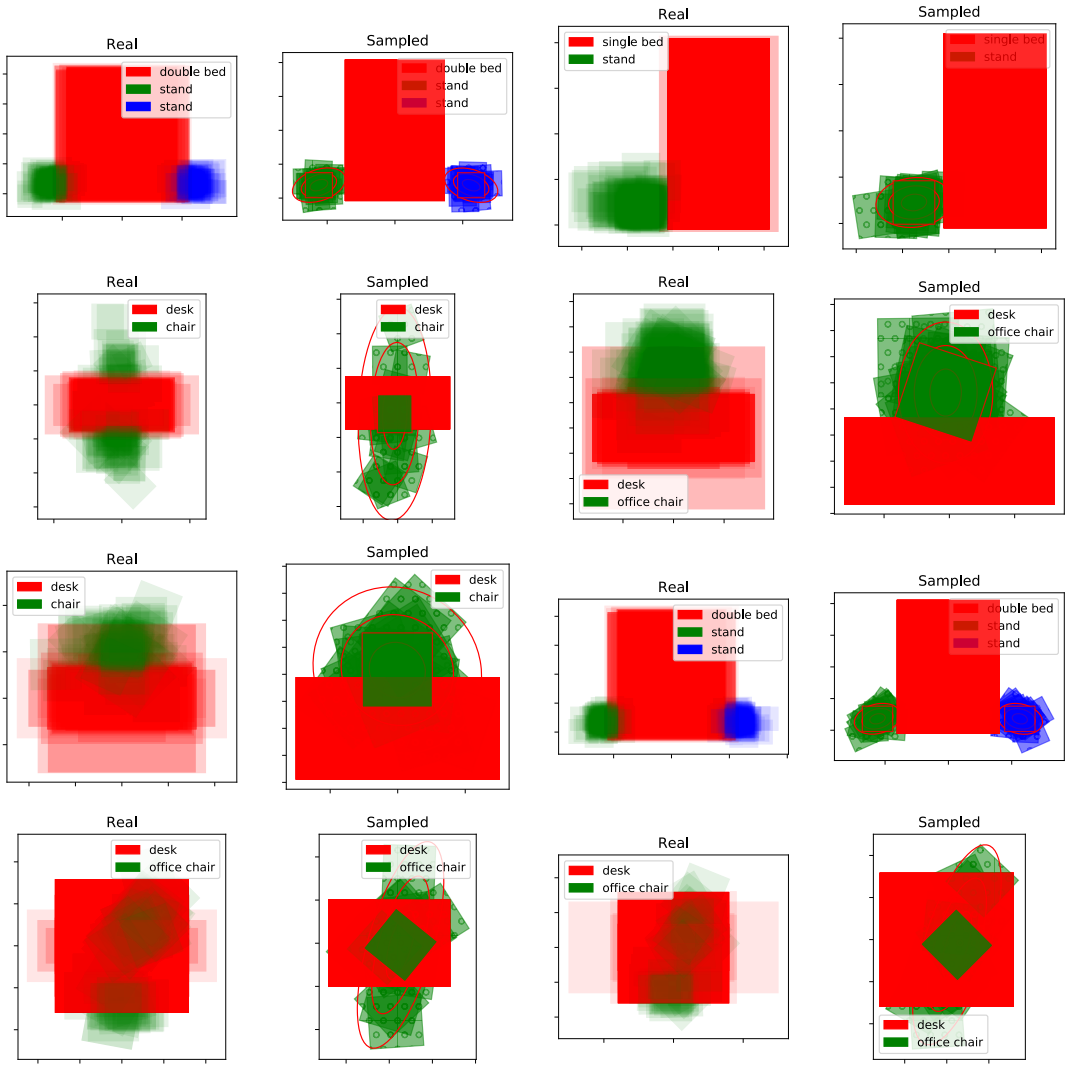


Figure B.6: Motifs for small rooms learned fitting rotations and with full covariance matrices. Real occurences and samples

Appendix C

Sampling new scenes

C.1 Samples $\mathcal{M}_1$

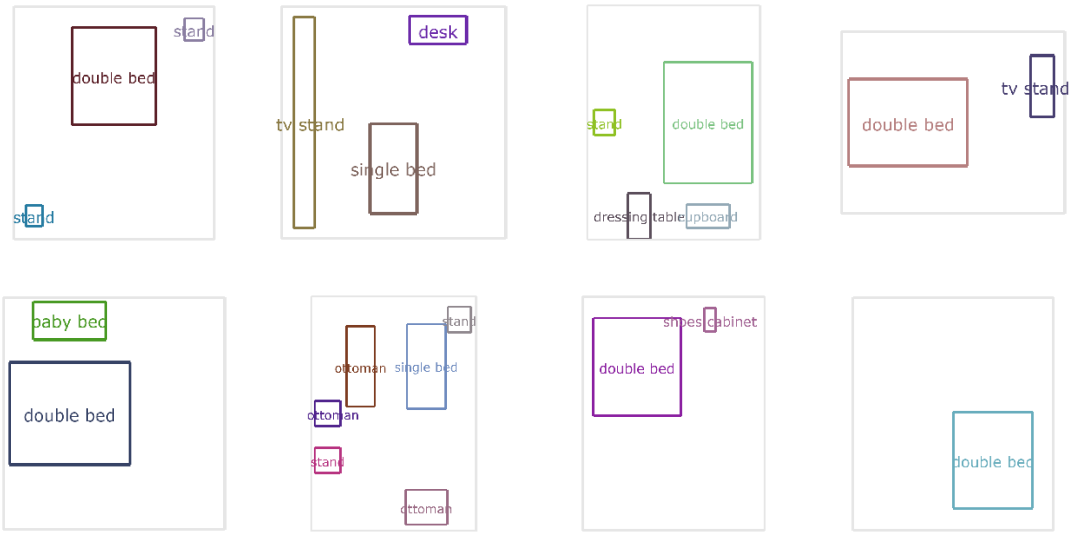


Figure C.1: Valid scenes small rooms

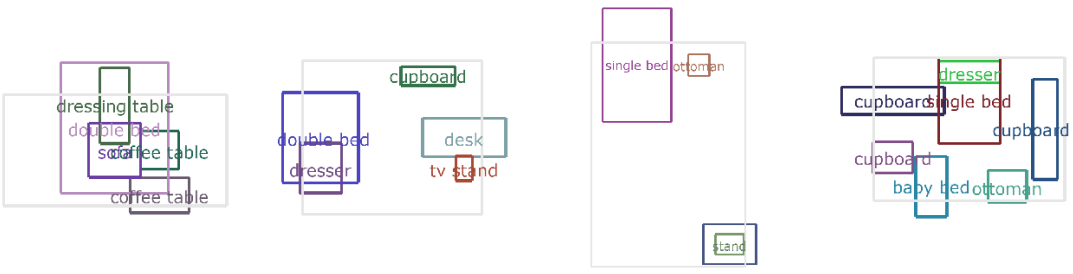


Figure C.2: Invalid scenes small rooms

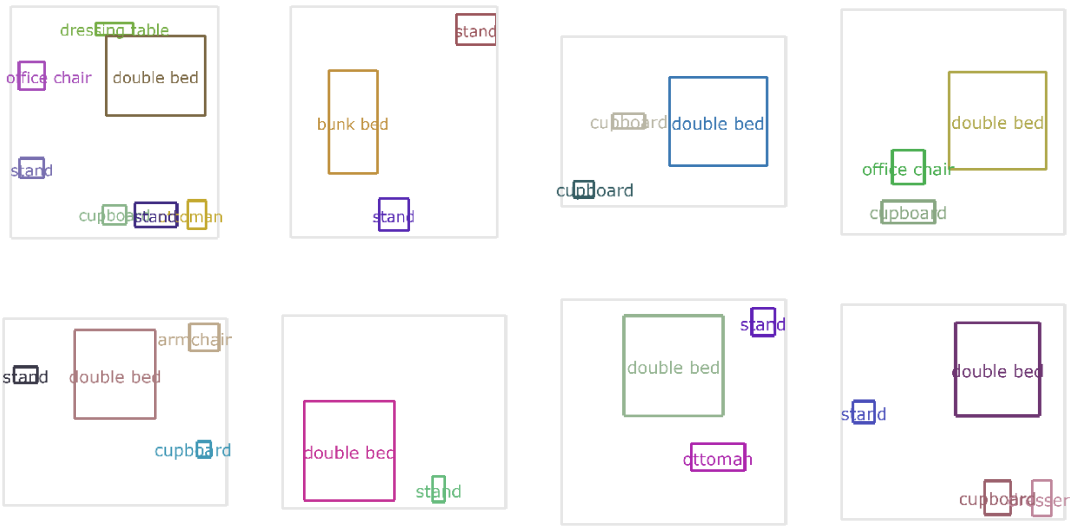


Figure C.3: Valid scenes medium rooms

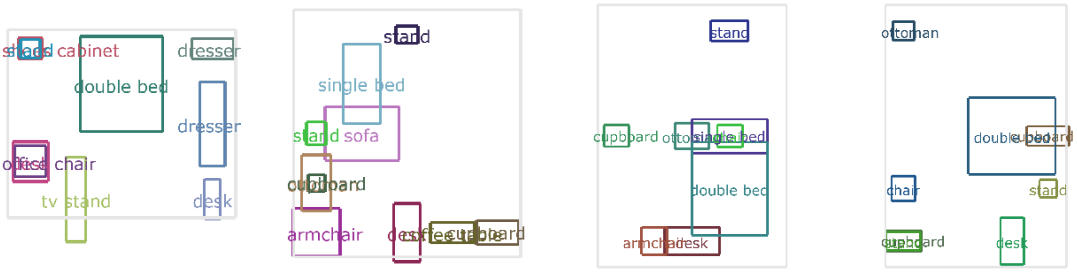


Figure C.4: Invalid scenes medium rooms

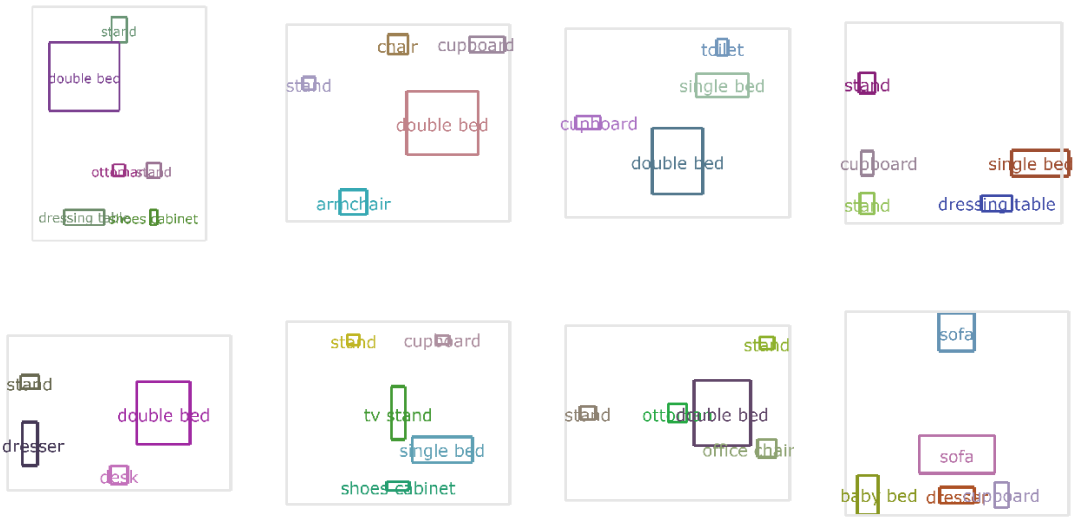


Figure C.5: Valid scenes big rooms

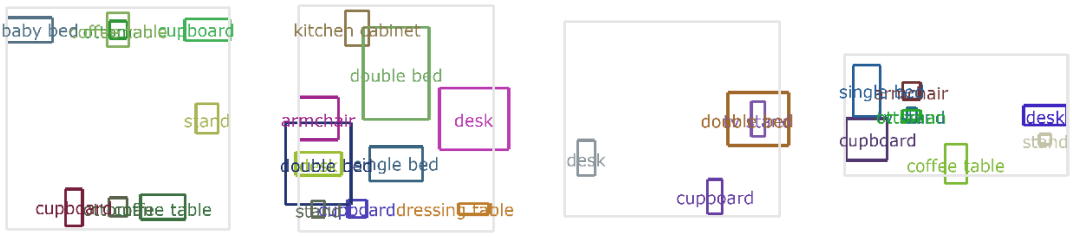


Figure C.6: Invalid scenes big rooms

C.2 Samples $\mathcal{M}_2$

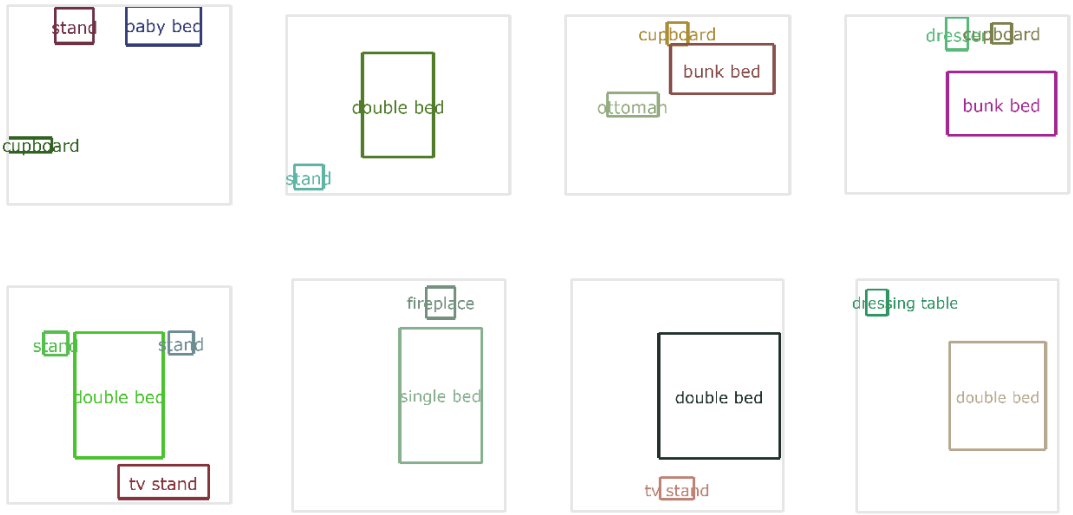


Figure C.7: Valid scenes small rooms

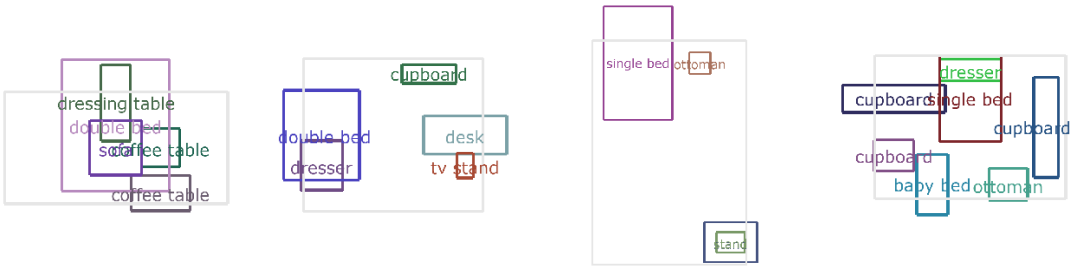


Figure C.8: Invalid scenes small rooms

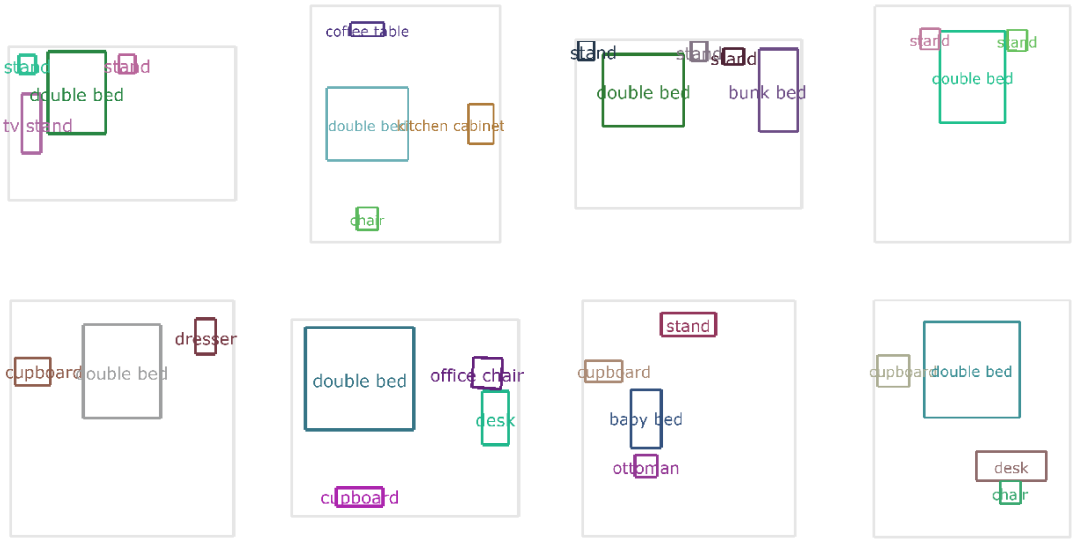


Figure C.9: Valid scenes medium rooms

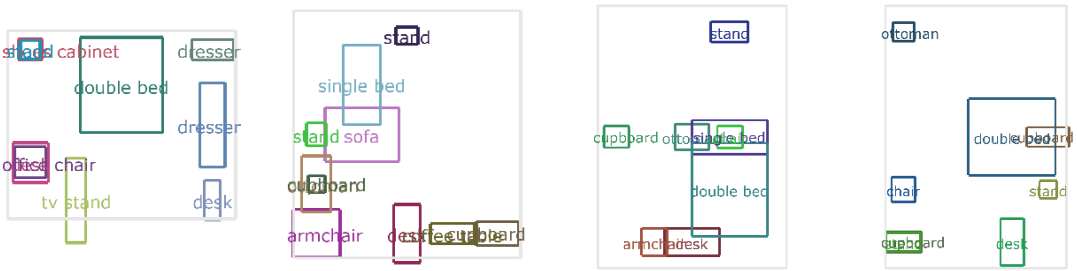


Figure C.10: Invalid scenes medium rooms

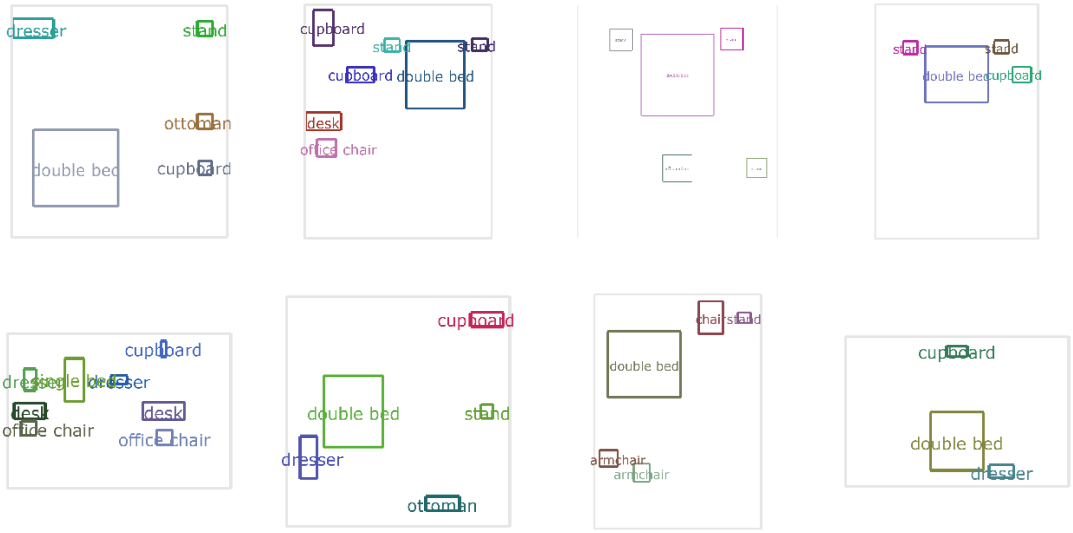


Figure C.11: Valid scenes big rooms

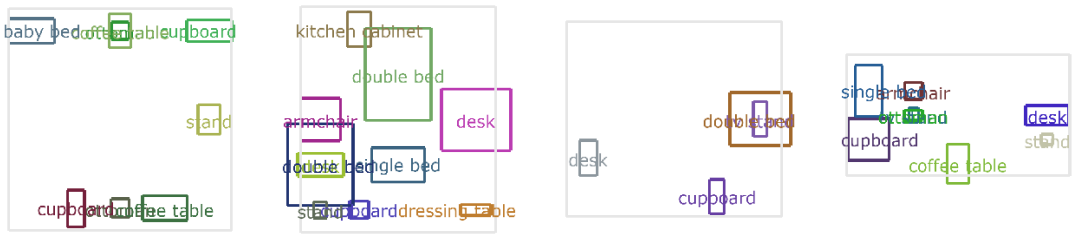


Figure C.12: Invalid scenes big rooms

Appendix D

Probability of scenes based on object count

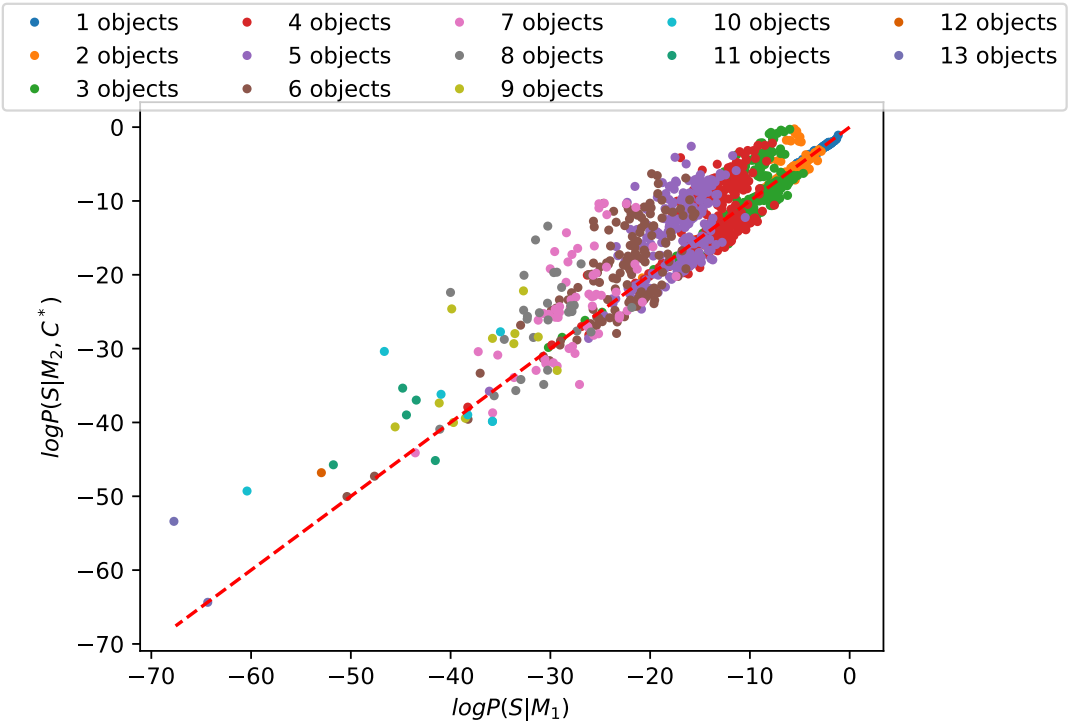


Figure D.1: Probability of small Rooms labeled by object count

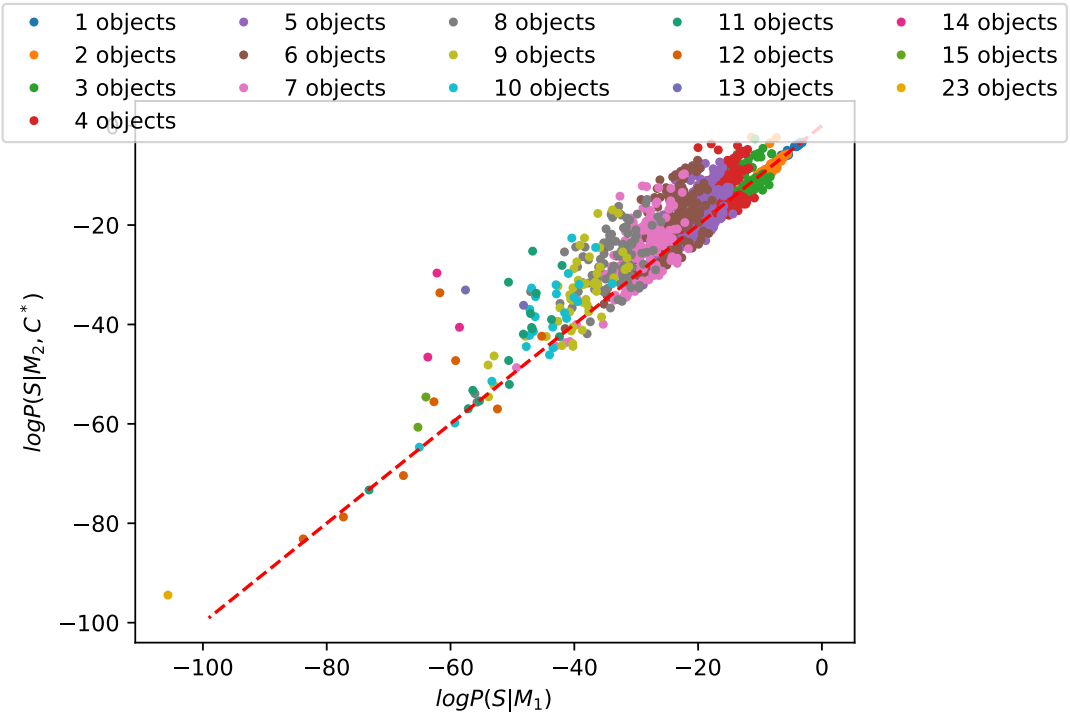


Figure D.2: Probability of Medium Rooms labeled by object count

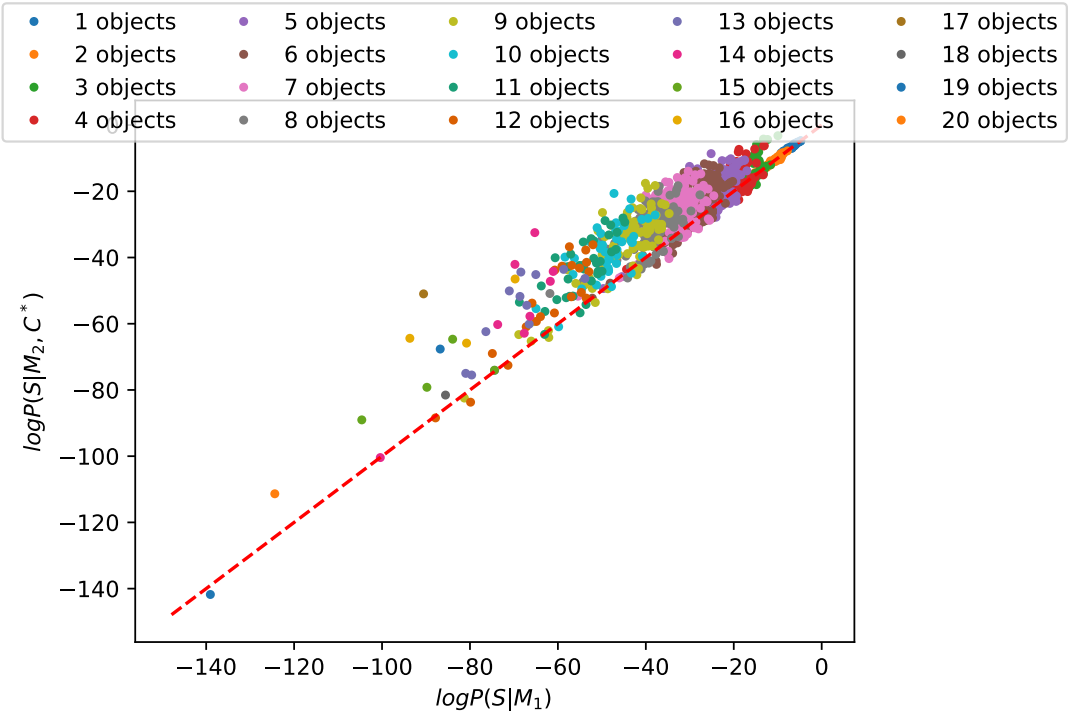


Figure D.3: Probability of Big Rooms labeled by object count

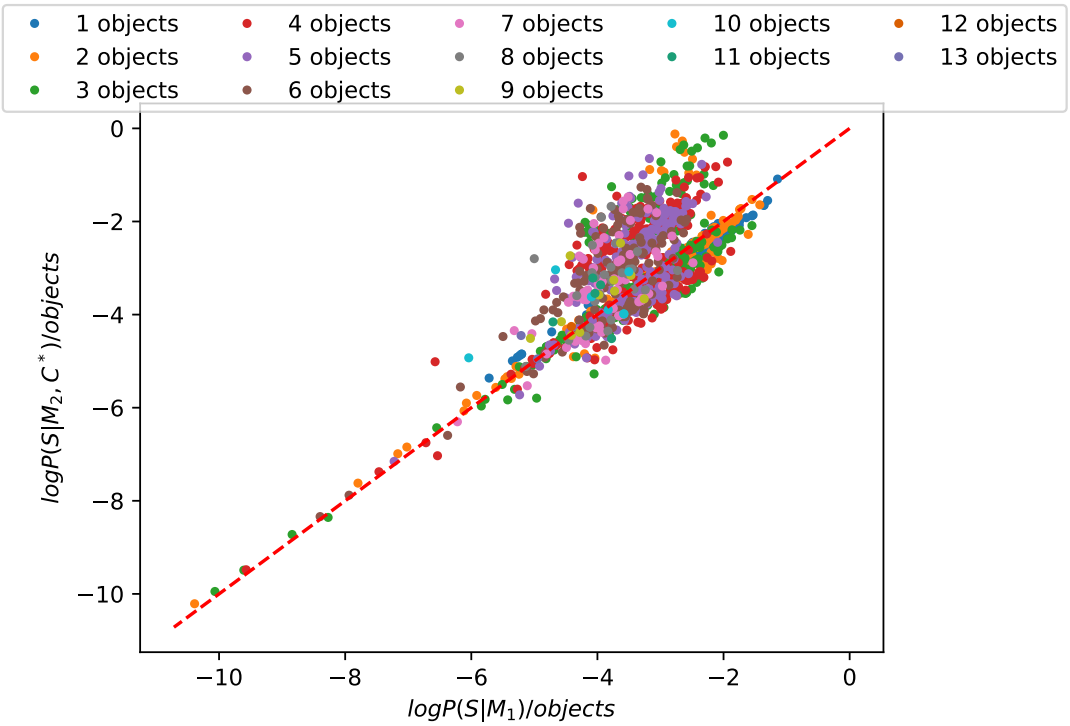


Figure D.4: Probability of small Rooms normalized by object count labeled by object count

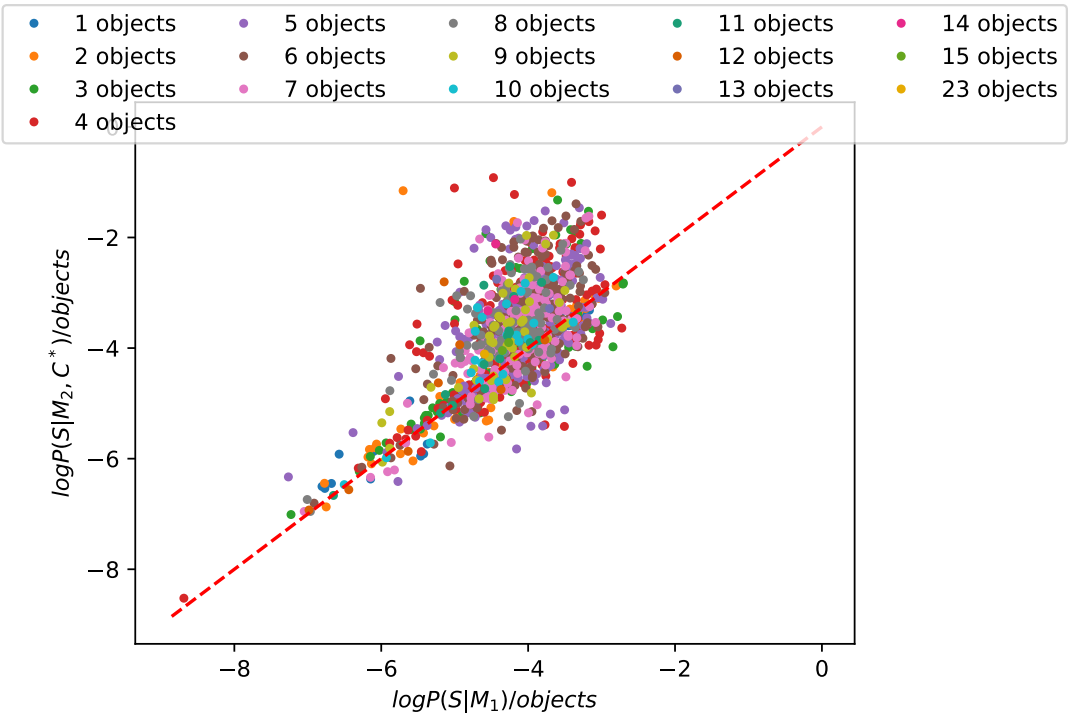


Figure D.5: Probability of Medium Rooms normalized by object count labeled by object count

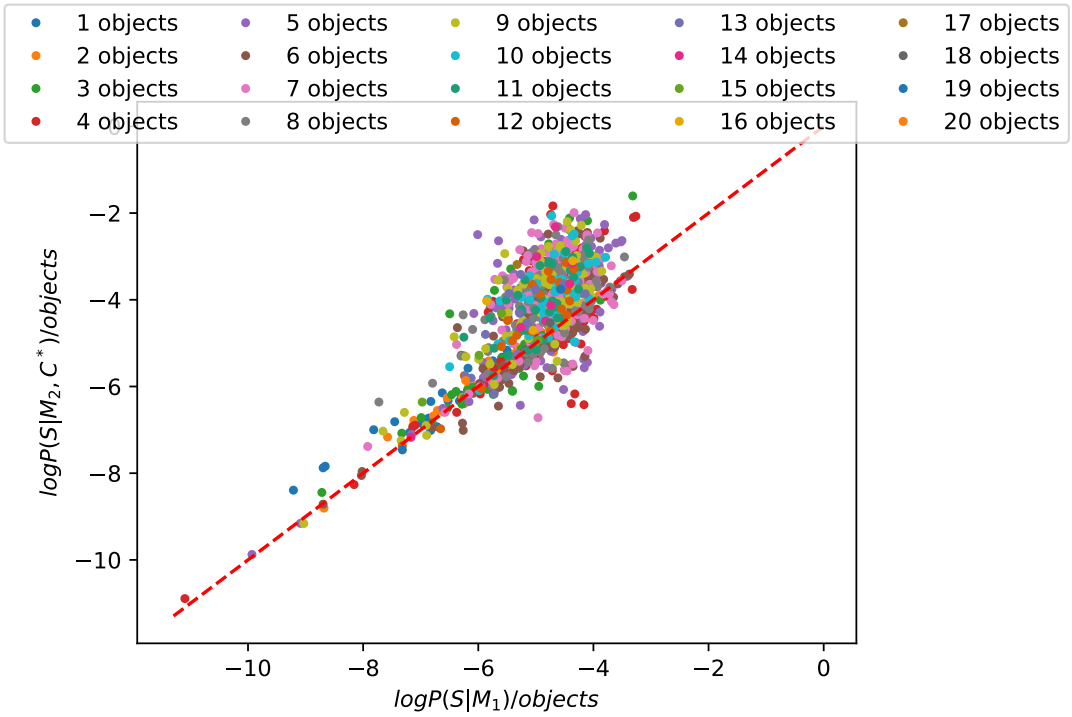


Figure D.6: Probability of Big Rooms normalized by object count labeled by object count

Bibliography

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (TOG)*, 31(6):135, 2012.

Walter R Gilks and Pascal Wild. Adaptive rejection sampling for gibbs sampling. *Applied Statistics*, pages 337–348, 1992.

Ankur Handa, Viorica Pătrăucean, Simon Stent, and Roberto Cipolla. Scenenet: An annotated model generator for indoor scene understanding. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 5737–5743. IEEE, 2016.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Paul Henderson and Vittorio Ferrari. A generative model of 3d object layouts in apartments. *arXiv preprint arXiv:1711.10939*, 2017.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986, 1984.

Tianqiang Liu, Siddhartha Chaudhuri, Vladimir G Kim, Qixing Huang, Niloy J Mitra, and Thomas Funkhouser. Creating consistent scene graphs using a probabilistic grammar. *ACM Transactions on Graphics (TOG)*, 33(6):211, 2014.

Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. In *ACM transactions on graphics (TOG)*, volume 30, page 87. ACM, 2011.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5899–5908, 2018.

Carl Edward Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems 12*, pages 554–560. MIT Press, 2000.

Sylvia Richardson and Peter J Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792, 1997.

Nicolas Rondan. Informatics research proposal: Learning scene arrangements. 2018.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

John W Tukey. *Exploratory data analysis*, volume 2. Reading, Mass., 1977.

Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220, 2016.

Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)*, 37(4):70, 2018.

Ken Xu, James Stewart, and Eugene Fiume. Constraint-based automatic placement for scene composition. In *Graphics Interface*, volume 2, page 4, 2002.

Michael Ying Yang, Wentong Liao, Hanno Ackermann, and Bodo Rosenhahn. On support relations and semantic scene graphs. *ISPRS journal of photogrammetry and remote sensing*, 131:15–25, 2017.

Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. Make it home: automatic optimization of furniture arrangement. 2011.

Yibiao Zhao and Song-Chun Zhu. Scene parsing by integrating function, geometry and appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3119–3126, 2013.