Indoor Scene Parsing with Instance Segmentation, Semantic Labeling and Support Relationship Inference

[1,3]Wei Zhuo , [2]Mathieu Salzmann , [1,3]Xuming He , and [1,3]Miaomiao Liu

[1] Australian National University, Canberra, Australia
[2] CVLab, EPFL, Switzerland
[3] Data61, CSIRO, Canberra, Australia
wei.zhuo@anu.edu.au, mathieu.salzmann@epfl.ch {xuming.he, miaomiao.liu}@data61.csiro.au

Abstract

Over the years, indoor scene parsing has attracted a growing interest in the computer vision community. Existing methods have typically focused on diverse subtasks of this challenging problem. In particular, while some of them aim at segmenting the image into regions, such as object or surface instances, others aim at inferring the semantic labels of given regions, or their support relationships. These different tasks are typically treated as separate ones. However, they bear strong connections: good regions should respect the semantic labels; support can only be defined for meaningful regions; support relationships strongly depend on semantics. In this paper, we therefore introduce an approach to jointly segment the instances and infer their semantic labels and support relationships from a single input image. By exploiting a hierarchical segmentation, we formulate our problem as that of jointly finding the regions in the hierarchy that correspond to instances and estimating their class labels and pairwise support relationships. We express this via a Markov Random Field, which allows us to further encode links between the different types of variables. Inference in this model can be done exactly via integer linear programming, and we learn its parameters in a structural SVM framework. Our experiments on NYUv2 demonstrate the benefits of reasoning jointly about all these subtasks of indoor scene parsing.

1. Introduction

Indoor scene understanding is one of the core challenges in computer vision. It aims at providing detailed information about the objects in a scene, such as their type and how they interact with each other. Such a level of understanding could have a high impact in many applications, such as personal robotics, where, to be able to interact with objects, one needs to reason about their semantics and how they are placed relative to each other.

In essence, indoor scene parsing is a complex problem that consists of multiple subtasks, such as segmenting the scene into meaningful regions [2, 7, 18], such as object or surface instances, predicting semantic labels for every pixel in the scene [16, 4, 22] and reasoning about the support relationships of different regions [11, 6, 19, 15]. In the literature, with the exception of [20] that jointly reasons about regions and semantics, existing approaches typically tackle these subtasks independently. These subtasks, however, truly are strongly connected. For instance, the support relationship of two regions is highly correlated with their semantics; reasoning about support can be facilitated by using semantically meaningful regions. By addressing these tasks separately, or sequentially, existing methods cannot leverage the full collective power of all these dependencies.

In this paper, we therefore introduce an approach to jointly segment the instances and infer their semantic labels and support relationships in an indoor scene from a single input image. To this end, we exploit a hierarchical segmentation and formulate our problem as that of finding the regions corresponding to instances in this hierarchy, while simultaneously predicting a semantic label for each such region and the support relationship between any pair of such regions. We jointly express these subtasks in a single Markov Random Field (MRF). This allows us to effectively encode the dependencies between them, thus leveraging all the connections underlying our overall problem.

We perform inference in the resulting MRF exactly by formulating it as an integer linear programming problem. To cope with the size of this problem, we propose to make use of a regressor trained to predict the overlap of each region with a ground-truth instance to effectively prune the region candidates. Thanks to the efficiency of this reduced inference strategy, we can learn the parameters of our model using structural Support Vector Machines (SVM). To this end, we design a loss function that reflects the multi-task nature of our indoor scene parsing formalism.

We demonstrate the effectiveness of our approach on the NYUv2 dataset [19]. Our experiments evidence that accounting for the dependencies between regions, their semantics and their support helps improving the prediction of the corresponding variables, with a particularly high impact on support relationships.

2. Related Work

Indoor scene understanding has been an important research focus in the computer vision community. As discussed above, this challenging problem consists of multiple subtasks. In particular, here, we tackle the tasks of instance segmentation, semantic labeling and support relationships prediction. We therefore focus the discussion below to the methods that have proposed to address these tasks.

Segmenting an image into regions has attracted a huge interest over the years [1, 2, 3, 18]. A complete review of this literature goes beyond the scope of this paper. Here, we briefly discuss the ones that have been used for indoor scene understanding. In this context, the most direct approach consists of using standard over-segmentation methods, such as SLIC [1], Mean-Shift [3] and normalized-cut [18]. In [14], multiple such over-segmentations were employed jointly for monocular normal estimation. By contrast, many approaches favor exploiting hierarchical segmentations [1, 2, 7, 9]. While some works then select specific levels in this hierarchy [23, 17], others aim to automatically find the best active regions in it, e.g., that fit the image contours [9], or whose pixel intensities follow a Gaussian distribution [10]. Segmentation, however, often acts as a pre-processing step to later perform some other task.

In particular, semantic segmentation methods have often relied on pre-defined image regions [17, 19, 7, 21]. The motivation behind this was both computational cost and robustness to noise. Indeed, early approaches to semantic segmentation often relied on MRFs, in which inference can be expensive when working at pixel level. Furthermore, working with regions allows one to regularize the predictions spatially. With the recent advent of Deep Learning, and progress in efficient inference methods [12], many approaches now work directly at the level of pixels [16, 4, 22].

By contrast, when it comes to estimating support relationships, the notion of regions remains necessary. The idea of estimating support was introduced in [19], where a hierarchical segmentation was used to predict support from below, from behind or no support between pairs of regions. In this context, [6] predicts the height and extent of surfaces that can support objects or people. In [11] , instead of 2D segments, support is defined between 3D boxes. More recently, [15] proposed to make use of object classes and physical stability to reason about support relationships between regions. All these methods make use of an RGBD image as input. By contrast, here, we aim to predict support from a single, standard RGB image.

More importantly, most of the methods discussed above tackle a single subtask of the challenging indoor scene understanding problem. The only exceptions we are aware of are [20], which jointly selects active regions in a hierarchy and predicts their semantic label, and [19], which jointly reason about semantics and support relationships. Both of these works, however, also makes use of RGBD as input. By contrast, here, we aim to jointly segment the object or surface instances and infer their semantics and their support relationships from a single RGB image. To the best of our knowledge, our work constitutes the first attempt at considering all three subtasks together.

3. Our Approach

Our goal is to jointly solve three sub-problems of indoor scene understanding, i.e., instance segmentation, semantic labeling and support relationship prediction, so as to account for their dependencies. To this end, we make use of a segmentation hierarchy, obtained by the method of [7]. Our problem then translates to that of selecting the regions that best match ground-truth instances in this hierarchy, predicting their semantic label and their pairwise support relationships. We express this as inference in an MRF with three types of nodes: region selection ones, semantic label ones and support relationships ones. The edges in the model encode the dependencies between these variables.

More specifically, let us assume to be given a hierarchy of R regions forming a tree. To select the active regions in this tree, we define a set of binary variables $A = \{a_i\}_{i=1}^{R}$, $a_i \in \{0, 1\}$. Furthermore, let $M = \{M_i\}_{i=1}^{R}$, $M_i \in \{1, \ldots, K\}$ be the set of semantic labeling variables defining the class to which a region belongs, for K semantic classes. We then define an additional set of variables to model the support relationships between any two regions. To this end, let S_{ij} denote the type of support that region j provides to region i. Following [19], we consider three different cases: No support ($S_{ij} = 0$); j supports i from below ($S_{ij} = 1$); j supports i from behind ($S_{ij} = 2$). Note that we will often refer jointly to the latter two types as positive support, as opposed to the first type that corresponds to negative support. Furthermore, we introduce a hidden region to model the fact that some regions may be supported by a region that is not visible in the image. Altogether, the support variables can be expressed as $S = \{S_{ij}\}_{i=1,j=0}^{R}$, $S_{ij} \in \{0, 1, 2\}$, where $j = 0$ corresponds to support by the hidden region.

We then formulate the problem of jointly inferring these three types of variables as that of maximizing the function

$$E(A,M,S) = \sum_{i=1}^{R} \phi_a(a_i) + \sum_{i=1}^{R} \phi_{ma}(M_i,a_i) + \phi_{tree}(A) + \sum_{i=1}^{R}\sum_{j=0}^{R} \phi_s(S_{ij}) + \sum_{i=1}^{R}\sum_{j=0}^{R} \phi_{sa}(S_{ij},a_i,a_j) \tag{1}$$

with respect to A, M and S, which can equivalently be converted to minimizing an MRF energy. The function relies on several potentials, which we discuss below.

The first term $\phi_r(a_i)$ is a unary potential encoding the probability that region i is active. We define this potential as $\phi_r(a_i) = w_a^T f_i^a [a_i = 1]$, where $[\cdot]$ is the indicator function, thus setting this potential to zero when $a_i = 0$. The vector f_i^a is a feature vector defined in Section 3.3, and w_a is the corresponding parameter vector to be learned from data.

The potential $\phi_{ma}(M_i,a_i)$ encodes the probability of predicting a particular semantic label for region i if the region is active. Simultaneously, it assigns a fixed cost to inactive regions. This can be expressed as

$$\phi_{ma}(M_i,a_i) = \begin{cases} 0 & a_i = 0 \\ w_{ma:M_i}^T f_i^{ma} & a_i = 1 \end{cases} \tag{2}$$

where f_i^{ma} is a feature vector, which, as described in Section 3.3, links semantics and support relationships. The vector $w_{ma:M_i}$ contains the parameters corresponding to each class M_i and will be learned from data.

The potential $\phi_{tree}(A)$ enforces constraints on the set of active regions. For the segmentation to be valid, every pixel in the image should be covered by a single region. This is achieved by making sure that only one region is selected in every path from the root of the segmentation hierarchy to a leaf node. To this end, we thus define $\phi_{tree}(A) = \sum_{\gamma \in \mathit{\Gamma}} -\infty[1 \neq \sum_{i\in\gamma}[a_i = 1]]$, where $\mathit{\Gamma}$ is the set of all root-to-leaf paths in the tree.

The unary potential $\phi_s(S_{ij})$ encodes the probability of a support variable to belong to either of the three classes. We write this potential as

$$\phi_s(S_{ij}) = w_{s:S_{ij}}^T f_{ij}^s \ , \tag{3}$$

where f_{ij}^s is a feature vector, which, as described in Section 3.3, links support types and semantics. The parameter vector $w_{s:S_{ij}}$ for each class S_{ij} will also be learned.

Finally, $\phi_{sa}(S_{ij},a_i,a_j)$ is a higher-order potential encoding the dependencies between the support variables and the region selection ones. We define this potential as

$$\phi_{sa}(S_{ij},a_i,a_j) = w_{sa} \begin{cases} w_b^T f_{ij}^{sa}, & S_{ij} \neq 0 \wedge (a_i = 0 \vee a_j = 0)] \\ w_c^T f_{ij}^{sa}, & S_{ij} \neq 0, a_i = 1, a_j = 1 \\ 0, & otherwise \ , \end{cases} \tag{4}$$

where f_{ij}^{sa} is a feature vector on a pair of regions, as described in Section 3.3. The vector w_b contains the parameters corresponding to the scenario where we predict a positive relationships even though either region is inactive, and w_c is the parameter vector for the case where both regions are active and we predict a positive relationship. Typically, we would like to penalize the first case and favor the second one. Other cases are assigned a fixed cost of zero.

3.1. Inference

To perform exact inference in our model, we propose to re-write it as an integer linear program (ILP). To this end, let $a \in \mathcal{B}^{2R+1}$ be a vector of binary variables representing the states of A, where $a_{i,1} = 1$ encodes the fact that region i is active, while $a_{i,0} = 1$ corresponds to an inactive region i. Here, we add an extra variable $a_{0,1} = 1$ corresponding to the hidden region and forcing it to always be active. Furthermore, $m = \{m_{i,k}\}$, $1 \leq i \leq R$, $0 \leq k \leq K$, denotes binary variables encoding the pairwise state space of M and A, where $m_{i,0}$ represents the case where $a_i = 0$ for an arbitrary M_i, and $m_{i,k\neq 0}$ corresponds to the pairwise state $a_i = 1$ and $M_i = k$. Additionally, let $s = \{s_{i,j,t\in\{0,1,2\}}\}$ encode the state of the support relationship variables, and z the triplet states corresponding to the higher-order term $\phi_{sa}(S_{ij},a_i,a_j)$, where $z_{i,j,l}$, $l \in \{1,2,3\}$, corresponds to the three cases in Eq. 4.

Inference in our model can then be re-written as the binary linear program

$$\underset{a,m,s,z}{\text{argmax}} \quad \sum_{i=1}^{R} \theta_i^a a_{i,1} + \sum_{i=1}^{R}\sum_{k=0}^{K} \theta_{i,k}^m m_{i,k} + \sum_{i=1}^{R}\sum_{j=0}^{R}\sum_{t=0}^{2} \theta_{i,j,t}^s s_{i,j,t} + \sum_{i=1}^{R}\sum_{j=0}^{R}\sum_{l=1}^{3} \theta_{i,j,k}^{sa} z_{i,j,l} \tag{5}$$

subject to

$$\begin{aligned} & a_{i,l},\ m_{i,u},\ s_{i,j,t}\ z_{i,j,l} \in \{0,1\} \quad \forall i,l,j,t,u,v \\ & a_{0,1} = 1\ , \end{aligned} \tag{6}$$

$$a_{i,0} + a_{i,1} = 1, \quad \forall i \tag{7}$$

$$\textstyle\sum_{k=0}^{K} m_{i,k} = 1, \quad \forall i \tag{8}$$

$$m_{i,0} = a_{i,0}, \quad \forall i \tag{9}$$

$$\textstyle\sum_{t=0}^{2} s_{i,j,t} = 1, \quad \forall i,j \tag{10}$$

$$\textstyle\sum_{l=1}^{3} z_{i,j,l} = 1, \quad \forall i,j \tag{11}$$

$$\textstyle\sum_{i\in\gamma} a_{i,1} = 1, \quad \forall \gamma \in \mathit{\Gamma} \tag{12}$$

$$\textstyle\sum_{t\in\{1,2\}}\sum_{j=0}^{R} s_{i,j,t} \geq a_{i,1}, \quad \forall i \tag{13}$$

$$\textstyle\sum_{t\in\{1,2\}}(s_{i,0,t} + s_{i,j,t}) \leq a_{i,1}, \quad \forall i, j \neq 0 \tag{14}$$

$$s_{i,0,1} \geq m_{i,1}, \quad \forall i \tag{15}$$

$$z_{i,j,2} = s_{i,j,0}, \quad \forall i,j \tag{16}$$

$$z_{i,j,3} \leq \sum_{t=1}^{2} s_{i,j,t}, \quad \forall i,j \tag{17}$$

$$z_{i,j,3} \leq a_{i,1}, \quad \forall i,j \tag{18}$$

$$z_{i,j,3} \leq a_{j,1}, \quad \forall i,j \tag{19}$$

$$z_{i,j,3} \geq \sum_{t=1}^{2} s_{i,j,t} + a_{i,1} + a_{j,1} - 2, \quad \forall i,j, \tag{20}$$

where the θ:s encode the different potentials described above. The constraints can be interpreted as follows: Eqs. 7 – 11 enforce the binary variables to correspond to valid predictions. Eq. 12 enforces the tree constraints on the region selection variables. Eq. 13 forces a region to be supported by at least one region when it is active. This constraint encodes the fact that there is no floating region in the real world. Eq. 14 prevents a region to be supported by the hidden region if there is a region in the scene that can support it. Eq. 15 forces a region to be supported by the hidden region if its semantic label is ground (semantic class 1 in our case). Eq. 16 – 20 enforce the binary variables z to correspond to one of the three cases in Eq. 4. To solve this ILP, we make use of Gurobi.

Speeding up inference. While Gurobi is very efficient, it remains too slow for us to handle our typical hierarchies, which contain roughly 200 regions. To address this issue, we therefore propose to first prune the regions. This procedure follows two steps. First, we remove the regions that contain less than 625 pixels, which, based on our statistics, are unlikely to correspond to object instances. Second, we exploit a regressor trained to predict the Intersection over Union (IoU) between a region in the hierarchy and a ground-truth instance. To this end, we make use of a neural network with three fully-connected layers, intertwined with ReLU activation, batch normalization, and dropout. This network is depicted by Fig. 1. We use deep features in conjunction with hand-crafted geometric ones as input to this shallow IoU regression network. See Section 3.3 for more detail about these features. We train this network using the square loss between the true IoU and the predicted one. To this end, we use batches of size 256, a learning rate of 10^{-3} and a momentum of 0.95. The dropout rate was set to 0.5. We also subsample the data so as to have a roughly balanced training set. To this end, we discretize the IoU interval $[0,1]$ into 10 bins, and subsample the data such that each bin contains roughly the same number of samples. At test time, we keep the 80 regions with highest predicted IoU that satisfy the constraint that each root-to-leaf path in the segmentation tree contains at least one region. In practice, this pruning yields less than 1% decrease in oracle weighted coverage, while greatly reducing the number of regions.

After pruning, we then train a two-class support classifier on the remaining regions to predict positive or negative support. We make use of this classifier to prune support pairs. To this end, we threshold the classifier score so as to obtain a high recall of positive support. In practice, we achieve 94% recall, while reducing from 5600 to 1100 pairs.

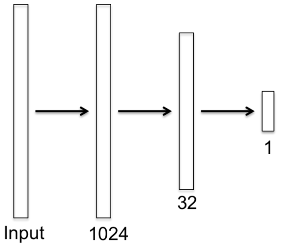

Figure 1. **Architecture of our IoU regressor.** We make use of a network with three fully-connected layers to predict the IoU between a candidate region and a ground-truth instance. We perform ReLU activation, batch normalization and dropout after the first and second layers.

Given the features, the pruning process for pairs takes 3s per image on average and that for regions 0.2s on average. Inference then takes 0.2s per image on average.

3.2. Learning

Given training data, we aim to learn the parameters of our model. One of the challenges of learning comes from the fact that, typically, the ground-truth instances that we seek to predict do not appear in our hierarchical segmentation. To reflect what will happen at test time, however, we would like to learn our model using the noisy segments from the hierarchies obtained from the training images. To this end, following [20], we rely on an *oracle segmentation*. Below, we first explain how these oracle segmentations are obtained, and then discuss our learning algorithm.

3.2.1 Oracle Segmentation

The goal of oracle segmentation is to find among the regions in a noisy hierarchical segmentation those that best match ground-truth instances and correspond to a valid tree cut, i.e., cover the image without redundancy. To this end, we make use of the ILP formulation of [20]. This formulation relies on two kinds of binary variables. The first ones are equivalent to our region selection variables $a = \{a_{i,l}\}$, $1 \leq i \leq R$, $l \in \{0,1\}$, discussed above. The second kind of variables encode the mapping between ground-truth instances and segments in the hierarchy. Let us denote these variables as $o \in \mathcal{B}^{G \times R}$, with G the number of ground-truth instances.

An oracle segmentation can then be obtained by solving the optimization problem

$$\underset{a,o}{\operatorname{argmin}} \quad \sum_{g=1}^{G} \sum_{i=1}^{R} \quad \theta^{o}_{g,i} o_{g,i} \tag{21}$$

subject to

$$a_{i,l}, o_{g,k} \in \{0,1\}, \quad \forall i,l,g,k, \tag{22}$$

$$a_{i,0} + a_{i,1} = 1, \quad \forall i, \tag{23}$$

$$\sum_{i \in \gamma} a_{i,1} = 1, \quad \forall \gamma \in \mathit{\Gamma} \tag{24}$$

$$o_{g,i} \leq a_{i,1}, \quad \forall g, i, \tag{25}$$

$$\sum_{i=1}^{R} o_{g,i} = 1, \quad \forall g, \tag{26}$$

$$\begin{aligned} & o_{g,i} + a_{j,1} \leq 1, \quad \forall g, i, j, \\ & \text{if} \quad \mathrm{IoU}(r_g, r_j) > \mathrm{IoU}(r_g, r_i) \end{aligned} \tag{27}$$

where $\mathrm{IoU}(\cdot,\cdot)$ denotes the intersection over union between two regions, and $\theta_{g,i} = \frac{|L_{r_g}|}{L}(\mathrm{IoU}(r_g, r_s) - \mathrm{IoU}(r_g, r_i))$ encodes the amount of weighted coverage lost by selecting region i instead of s, which corresponds to the best possible match for ground-truth region g. Most constraints simply force the solution to be valid, with the Eq. 27 guaranteeing that, among the regions that are active, the best one is assigned to a ground-truth region.

3.2.2 Learning via Structural SVM

We now turn to the learning problem per say. To this end, let $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(N)}, y^{(N)})\}$ be a set of pairs of images and labels, where $y^{(n)} = \{A^{(n)}, M^{(n)}, S^{(n)}\}$ comprises the best selection of segments from the segmentation tree, obtained using the oracle segmentation described above, the corresponding semantic labels, taken as the dominant label in each region, and support relationships, described in Section 4, for image i.

Our goal is to learn the weights in our MRF. The energy in this MRF can be equivalently written as $w^T\phi(x, y)$, where w concatenates all the weights we seek to learn, and, with a slight abuse of notation, $\phi(x, y) = [\phi_a, \phi_{ma}, \phi_s, \phi_{sa}]$ concatenates the corresponding features, so as to compute the different potentials. Following a margin re-scaling structural SVM formulation, learning the weights can be expressed as the optimization problem

$$\begin{aligned} & \min_{w, \epsilon \leq 0} \frac{1}{2} w^T w + \frac{\lambda}{N} \sum_{n}^{N} \epsilon_n \\ & s.t. \;\; w^T[\phi(x^{(n)}, y^{(n)}) - \phi(x^{(n)}, y)] \leq \triangle(y, y^{(n)}) - \epsilon_n, \; \forall y \end{aligned}$$

where $\triangle(y, y^{(n)})$ returns the loss of an arbitrary prediction y compared to the best configuration.

Here, to reflect the nature of our problem, where we aim to predict different types of variables jointly, we design the multi-task loss

$$\begin{aligned} \triangle\,(y, y^{(n)}) &= \frac{w^{ls}_{sup}}{Q} \sum_{i=1}^{R} \sum_{j=0}^{R} 1[S_{ij} \neq S^*_{ij}] \\ &+ w^{ls}_r \frac{1}{L} \sum_{g \in G} L_{r_g} \left(\max_{i \in A^{(n)}} \mathrm{IoU}(r_g, r^{(n)}_i) \right) \\ &- w^{ls}_r \frac{1}{L} \sum_{g \in G} L_{r_g} \left(\max_{i \in \hat{A}} \mathrm{IoU}(r_g, r_i) \right) \;, \end{aligned} \tag{28}$$

where $\hat{A}$ is the active set of A, that is, the set of regions such that $a_i = 1$, and similarly for $\hat{A}^{(n)}$ w.r.t. $A^{(n)}$. L_{r_g} is the number of pixels in region g, L is the number of pixels in all the ground-truth regions in an image, and Q is the number of active pairs in $\hat{A}$. Here, we use $w^{ls}_r = 1, w^{ls}_{sup} = 0.5$.

Loss-augmented Inference. An important step in structural SVM learning consists of performing loss-augmented inference to find predictions that have a high loss, but correspond to a low energy (or rather a high score in our maximization formulation). This can be expressed as solving

$$y^* = \operatorname*{argmax}_{\hat{y} \in \mathsf{y}} \triangle(\hat{y}, y^{(n)}) + w^T\phi(x, \hat{y}) \;. \tag{29}$$

Translating this into an ILP then yields the problem

$$\begin{aligned} \operatorname*{argmax}_{a,m,s,o} \quad & \sum_{i=1}^{R} \theta^a_i a_{i,1} + \sum_{i=1}^{R} \sum_{k=0}^{K} \theta^m_{i,k} m_{i,k} \\ & + \sum_{i=1}^{R} \sum_{j=0}^{R} \sum_{t=0}^{2} \theta^s_{i,j,t} s_{i,j,t} \\ & + \sum_{i=1}^{R} \sum_{j=0}^{R} \sum_{l=1}^{3} \theta^{sa}_{i,j,k} z_{i,j,l} \\ & + \sum_{g=1}^{G} \sum_{i=1}^{R} \quad \theta^o_{g,i} o_{g,i} \\ & + \sum_{i=1}^{R} \sum_{j=0}^{R} \sum_{t=0}^{2} \theta^{sl}_{i,j,t} s_{i,j,t} \end{aligned} \tag{30}$$

subject to the constraints of (5) and (21). Here, $\theta^o_{g,i}$ encodes the loss on the regions and is defined as in (21), $\theta^{sl}_{i,j,t}$ encodes the hamming loss on support relationships. It can thus be written as

$$\theta^{sl}_{i,j,t} = \frac{1}{Q}, s.t \quad t \neq S^*_{ij} \quad \forall t \in \{1, 2, 3\} \;. \tag{31}$$

To learn our model, we use the BCFW solver of [13]. Loss-augmented inference takes 1s per image on average.

3.3. Features

As discussed above, the IoU regressor, the support classifier and the potentials of Eq. 1 rely on different types of features. Here, we describe these feature vectors.

The IoU regressor relies on four types of features as input, which we refer to as **Conv5-SP**, **Pb-SP**, **Ext-Pb-SP** and **RGeo**. **Conv5-SP** is obtained from spatially pooled [8] features coming from the conv5 layer of the FCN-32s model of [16] fine-tuned on NYUv2 to predict semantics using RGB and HHA as input. HHAs were obtained from depth prediction using the method of [5]. **Pb-SP** and **Ext-Pb-SP** are derived from the semantic probability maps of the FCN-32s model mentioned above, using spatial pooling on each region and on a bounding box of 1.25 the region's extent around it, respectively. **RGeo** corresponds to the geometry features used in [19].

The support classifier relies on two types of features. The first concatenates **Pb-SP**, **Ext-Pb-SP** and **RGeo** for both regions. The second, denoted as **PGeo**, includes the containment, geometry and horizontal features of [19] computed on pairs of regions.

The feature vector f_i^a is obtained by concatenating two types of features, which we refer to as **RF** and **RGeo**. **RF** corresponds to the feature map after the second batch normalization module in the 3-layer neural network described in Section 3.1. It encodes the connection between the IoU regressor and the selection of the region.

The feature vector f_i^{ma} contains five types of features, denoted by **RGeo**, **Pb-SP**, **Ext-Pb-SP**, **Pb** and **Hm**. The first three have been described above. **Pb** is defined as the average over the region pixels of the K-dimensional semantic probability vectors obtained by the same FCN-32s as above. **Hm** aims to incorporate dependencies between semantics and support relationships. To this end, for region i, this feature is obtained by averaging over all the other regions j the probability of each support class between i and j, obtained by our SVM support classifier.

The feature vector f_{ij}^s is formed by two feature types, **Ps** and **Pm**. **Ps** is directly taken as the probabilities predicted by our support classifier. **Pm** aims at modeling dependencies between support and semantics. It concatenates the semantic features **Pb** discussed above for both regions.

The feature vector f_{ij}^{sa} concatenates **RGeo** and **RF** features for both regions, as well as the corresponding IoUs predicted by our 3-layer neural network. It further includes the feature **PGeo** described above.

The running time for feature extraction on regions and pairs are 14s and 2.7s per image on average, respectively.

4. Experimental Evaluation

We evaluate our model on the NYUv2 dataset, which provides RGB images and their corresponding depth maps. Note that, here, we do *not* use these depth maps. The dataset contains 749 images for training and 654 for testing.

The ground-truth regions, i.e., object or surface instances, and corresponding semantics are provided by [19]. The semantics include four classes: ground, structure, props and objects. Ground-truth support relationships were defined by [20] on the ground-truth regions. Based on the strategy of [20], we map these ground-truth support relationships to our segmentation hierarchy as follows: Any pair in which both regions have an IoU with ground-truth regions greater than 0.25 is assigned the corresponding ground-truth type. The other regions are assigned the *no support* label. If, at the end of this procedure, a region is not supported by any other region, we define it as being supported by the hidden one.

4.1. Evaluation Metrics

Since we predict three different types of variables, we need different metrics to evaluate them. Here, we use:

Instance segmentation accuracy. To evaluate our segmentation results, we make use of the maximum weighted coverage, defined over ground-truth regions $\mathcal{G}$ and predicted regions $\mathcal{R}$ as

$$\text{Coverage}_w(\mathcal{G},\mathcal{R}) = \frac{1}{|\mathcal{I}|}\sum_{j=1}^{|\mathcal{G}|} |r_j^G| \max_{1,\ldots,|R|} \text{IoU}(r_j^G, r_i^R)$$

where $|\mathcal{I}|$ is the number of pixels in the whole set of ground-truth regions, which may be less than the total number of pixels in the image, and $|r_j^G|$ is number of pixels in ground-truth region j.

Semantic labeling accuracy. To evaluate the predicted semantics, we make use of the standard average accuracy computed over all the pixels and per-class accuracy, where averaging is done over the classes.

Support relationship accuracy. For the support relationships, we evaluate the precision and recall of the positive support types on pairs not containing the hidden region. These values are defined as

$$\text{precision} = \frac{\text{\# true positive predictions}}{\text{\# positive predictions}}, \quad (32)$$

$$\text{recall} = \frac{\text{\# true positive predictions}}{\text{\# of positive samples}}. \quad (33)$$

4.2. Experimental Results

We now present our results on NYUv2. Since our model addresses multiple tasks, as a first experiment, we evaluate the influence of several of its components via an ablation study. To this end, we compare our complete model (**Ours**) with the following baselines:

Basic: This baseline only performs instance segmentation and includes the region unary and tree constraints of Eq. 1.

Ours-NS: This model jointly predicts the region selection variables and the semantics. However, it does not account for the support relationships. This model consists of the first three terms in Eq. 1.

Ours-ND: This model also infers the three kinds of variables. It contains all the terms in Eq. 1, but does not leverage the features that link support and semantics, i.e., **Hm** and **Ps** in Section 3.3. In essence, while predicting all variables, this baseline only models limited dependencies between them. In addition to these baselines, we also report the support predictions obtained with the linear SVM support classifier (**SC**) discussed in Section 3.3, which, among others, makes use of features encoding information about the region IoU with ground-truth and the semantics.

Model	W. Cov	Sem Avg Acc	Sem Per-Cls Acc	Support Precision	Support Recall
Basic	58.9	-	-	-	-
SC	-	-	-	44.8	39.0
Ours-NS	59.3	73.0	72.0	-	-
Ours-ND	59.3	73.3	72.2	47.0	41.9
Ours	59.4	73.2	72.1	47.6	43.1
Ours(GtSem)	60.1	-	-	48.2	45.0

Table 1. **Evaluation on NYUv2.** We compare our approach to several baselines, mostly corresponding to different components of our complete model. Note that some of these baselines do not predict all variable types, and can thus only be evaluated on some metrics. These results demonstrate the importance of jointly inferring multiple variable types, in particular on the quality of the support relationships.

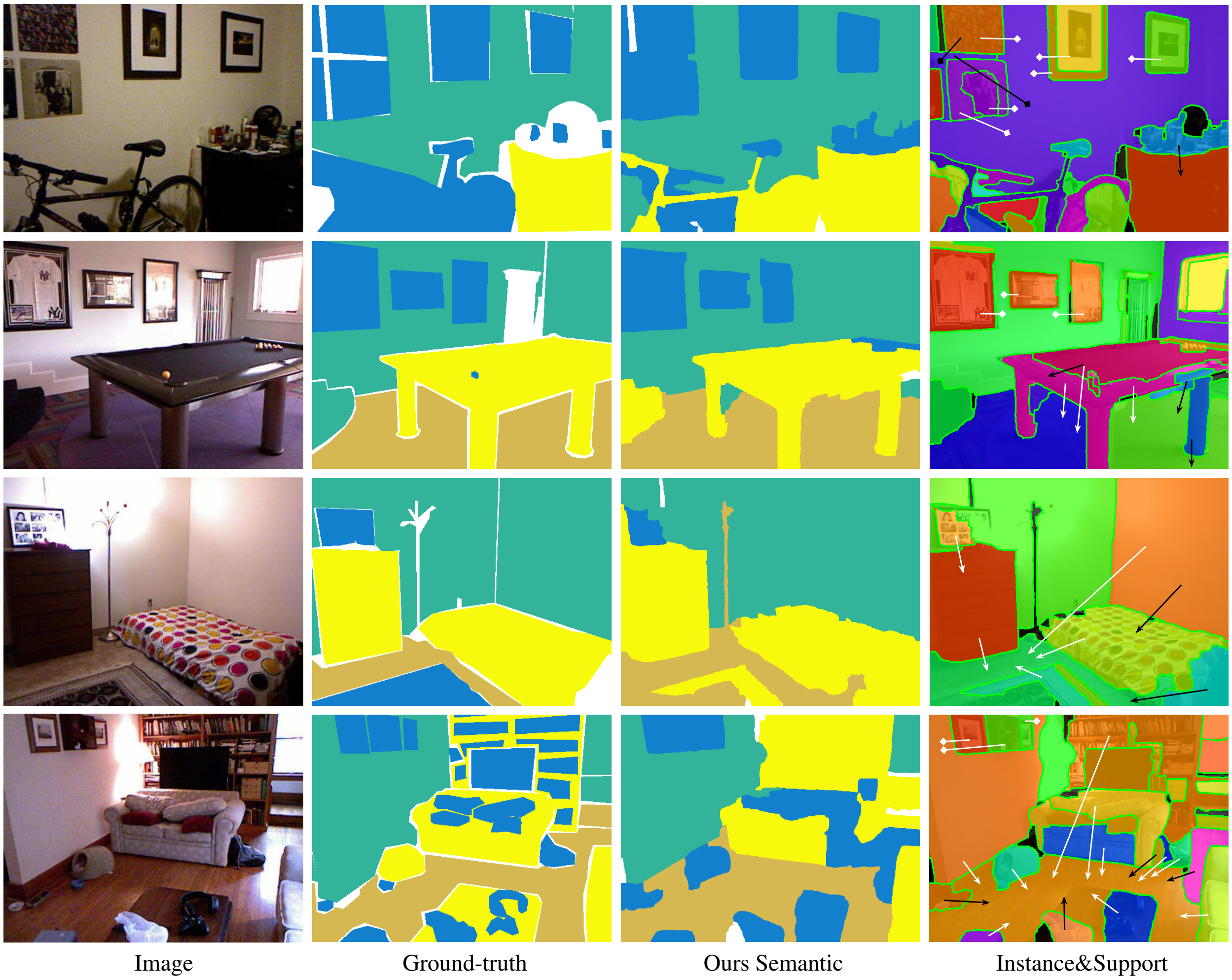

Figure 2. **Qualitative evaluation of our results.** We show the input image, the ground-truth semantics, the semantics predicted by our approach, and our regions and support predictions. We show the correct relationships in white and the incorrect ones in black. Support from below is indicated by an arrow head and from behind by a diamond one. Note that our semantics match the ground-truth ones quite closely. Furthermore, our regions typically correspond to semantically-meaningful portions of the scene, that is, complete object or surface instances, and our support corresponds to correct relationships. (Best viewed in color.)

The results of our method and of these baselines are provided in Table 1. Note that some baselines do not predict all the variables, and can thus not be evaluated according to all the metrics. These results show that (i) jointly predicting regions and semantics improves the quality of the segments; (ii) predicting all three types of variables yields a significant boost to the support quality compared to our support classifier; (iii) modeling the dependencies between the different variable types further improves the support predictions, particularly in terms of recall. Altogether, we believe that these

Model	Oracle W.Cov	W. Cov	Sem Avg Acc	Sem Per-Cls Acc	Support Precision	Support Recall
Basic	68.8	61.1	-	-	-	-
SC	-	-	-	-	48.3	37.9
Ours-NS	68.8	62.8	74.8	73.7	-	-
Ours	68.8	62.7	75.3	74.3	49.5	38.6
[20]	70.6	62.5	-	-	-	-
[19]	-	-	-	-	54.5	-

Table 2. **Evaluation on NYUv2 RGBD.** We compare our approach to several baselines corresponding to different components of our complete model and to the state-of-the-art methods [20, 19]. Note that, while our oracle weighted coverage is lower than that of [20], we achieve higher weighted coverage, thus showing the impact of accounting for the dependencies between multiple tasks.

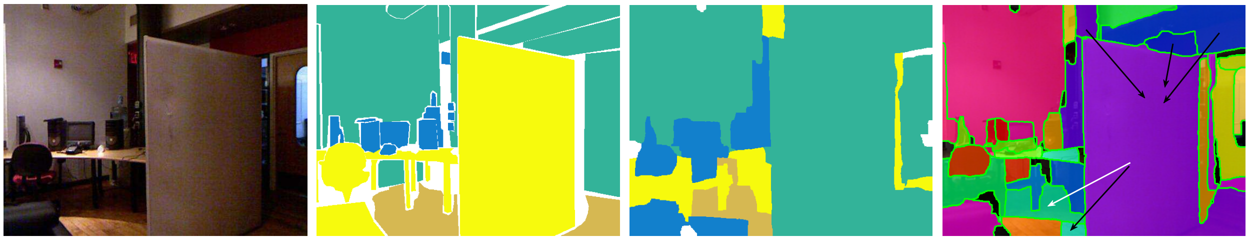

Figure 3. **Failure case.** Here, our support relationships are affected by a wrong semantic labeling.

results demonstrate the benefits of jointly inferring regions, semantics and support relationships.

To further evidence the impact of semantics, we performed an experiment where we used the ground-truth ones in our model. This model is denoted as **Ours(GtSem)**. This resulted in a 3.1% relative improvement on recall, thus showing that better semantics yield better support.

In Fig. 2, we provide some qualitative results obtained with our approach. Note that the semantic labels we predict closely match the ground-truth ones. Note also that, while they contain some degree of over-segmentation, the regions we produce typically still remain reasonably large, with a clear semantic meaning. Our method is also able to predict accurate support relationships, even in the presence of many different objects, as in the last row of the figure. In Fig. 3, we show a typical failure case of our approach. We have observed that such failures mostly occur when a region is over-segmented, or assigned to the wrong semantic category. Note that this again indicates the dependencies between these different subtasks of indoor scene parsing.

Comparison with RGBD-based methods. As mentioned in Section 2, existing methods that predict support relationships all work with RGBD images as input. To compare against these methods, we slightly modified our approach to exploit RGBD.In particular, we generated the hierarchy using ground-truth depth, and employed ground-truth depth to extract our features, except for the semantic probability ones. The results in Table 2 show again that our model benefits from solving multiple tasks. Note that, despite the fact that the oracle performance obtained from our segmentation hierarchy is lower than that of [20], the segmentation obtained by our method has a higher weighted coverage. In other words, since the gap between our weighted coverage and the oracle one is significantly smaller than for [20], i.e., 5.5% vs 8.1%, our model essentially selects better regions than [20]. The comparison to [19] for support prediction should be taken with caution, since the regions are different. We believe that this comparison shows that both method perform similarly, with our approach providing additional information about the scene. Note that we expect that exploiting depth more thoroughly than done here could give our approach a bigger boost.

5. Conclusion

We have introduced an approach to jointly segmenting the instances in an image and predicting their semantic labels and support relationships. To the best of our knowledge, this constitutes the first attempt at jointly tackling these three subtasks of indoor scene understanding. Our experiments have demonstrated that jointly reasoning about these three tasks is in general beneficial, and particularly so for support relationships. Indoor scene understanding, however, is not limited to these three tasks. One can, for example, also aim to predict depth, surface normals and object affordances. Ultimately, we believe that all these problems should be tackled jointly to better leverage their dependencies. This will be the focus of our future research.

6. Acknowledgements

The first author is supported by the Chinese Scholarship Council and Data61/CSIRO. The authors also thank NVIDIA for providing them a Tesla k40c graphics card.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[2] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–335, 2014.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[4] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[5] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[6] R. Guo and D. Hoiem. Support surface prediction in indoor scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2144–2151, 2013.

[7] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571, 2013.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[9] D. Hoiem, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. *International Journal of Computer Vision*, 91(3):328–346, 2011.

[10] S. X. Hu, C. K. Williams, and S. Todorovic. Tree-cut for probabilistic image segmentation. *arXiv preprint arXiv:1506.03852*, 2015.

[11] Z. Jia, A. Gallagher, A. Saxena, and T. Chen. 3d-based reasoning with blocks, support, and stability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2013.

[12] V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Adv. Neural Inf. Process. Syst*, 2011.

[13] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *ICML*, 2013.

[14] L. Ladickỳ, B. Zeisl, and M. Pollefeys. Discriminatively trained dense surface normal estimation. In *European Conference on Computer Vision*, pages 468–484. Springer, 2014.

[15] W. Liao, M. Y. Yang, H. Ackermann, and B. Rosenhahn. On support relations and semantic scene graphs. *arXiv preprint arXiv:1609.05834*, 2016.

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[17] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2759–2766. IEEE, 2012.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[19] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[20] N. Silberman, D. Sontag, and R. Fergus. Instance segmentation of indoor scenes using a coverage loss. In *European Conference on Computer Vision*, pages 616–631. Springer, 2014.

[21] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *European conference on computer vision*, pages 352–365. Springer, 2010.

[22] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[23] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 614–622, 2015.