SpatialSense: An Adversarially Crowdsourced Benchmark for Spatial Relation Recognition

Kaiyu Yang
Princeton University
kaiyuy@cs.princeton.edu

Olga Russakovsky
Princeton University
olgarus@cs.princeton.edu

Jia Deng
Princeton University
jiadeng@cs.princeton.edu

Abstract

Understanding the spatial relations between objects in images is a surprisingly challenging task (Fig. 1). A chair may be “behind” a person even if it appears to the left of the person in the image (if the person is facing right). The predicate “on” entails very different spatial configurations in different contexts: a sweatshirt “on” a person versus a hat “on” a person versus a leash “on” a dog. Reasoning about spatial relations may further require understanding other objects in the scene: for example, whether two students are “next to” each other depends on if there is a third student between them. We introduce SpatialSense, a dataset specializing in spatial relation recognition which captures a broad spectrum of such challenges, allowing for proper benchmarking of computer vision techniques. SpatialSense is constructed through adversarial crowdsourcing, in which human annotators are tasked with finding spatial relations that are difficult to predict using simple cues such as 2D spatial configuration or language priors. Adversarial crowdsourcing significantly reduces dataset bias and samples much more interesting relations in the long tail compared to existing datasets. On SpatialSense, state-of-the-art recognition models perform comparably to simple baselines, suggesting that they rely on straightforward cues instead of fully reasoning about this complex task. The SpatialSense benchmark provides a path forward to advancing the spatial reasoning capabilities of computer vision systems.

1. Introduction

Visual understanding of space is essential for an intelligent agent. Such an understanding is the basis for describing scenes and referring to objects [7]: it is also the foundation required for tasks such as navigation and manipulation [26]. To understand space it is important to understand spatial relations, that is, how different spatial entities are configured relative to each other to compose a scene. Consider the following description: “Inside the living room, under the window next to the wall is a table, on top of which lies a vase with flowers in it”. The sentence may be put together any number of ways, but its meaning is determined by the objects (“room”, “window”, “table”, “vase”, “flowers”) and their spatial relations (“table in room”, “table under window”, “table next to wall”, “vase on table”, “flowers in vase”).

This raises the problem of spatial relation recognition: given two objects in a scene, what is their spatial relation? This problem is important, interesting, and challenging because the semantics of spatial relations are rich and complex. The spatial semantics between objects depend not
only on geometric properties such as location, pose, and shape, but also the frame of reference (e.g. “left to the car” can be relative to the observer or the car) and object-specific common sense knowledge (e.g. “hand over bed” does not imply physical contact while “blanket over bed” does).

**Contributions.** In this paper we introduce *SpatialSense*, a dataset specializing in spatial relation recognition. A key feature of the dataset is that it is constructed through adversarial crowdsourcing: a human annotator is asked to come up with adversarial examples to confuse a robot. This yields two desirable properties. First, the dataset includes negative examples as some human annotators are explicitly tasked with choosing spatial relations that are false. Second, the dataset focuses on questions that require more advanced reasoning and cannot be answered by simple spatial and language priors.

*SpatialSense* has 17,498 relations on 11,569 images.

Given two objects (names and bounding boxes), the task is to classify whether a particular spatial relation holds. We provide the object names and localizations to decouple object detection from spatial relation recognition, such that a successful relation recognition system can be directly placed on top of any object detection system. The dataset contains relations between 3,679 unique object classes, with 2,139 of these object classes appearing only once, providing a challenging long-tail distribution of concepts.

*SpatialSense* provides a rigorous testbed for spatial relation reasoning that is not easily amenable to simple priors. First, each predicate (“on”, “under”, etc.) has an equal number of positive and negative relations. Second, simple baselines using only 2D or language cues are significantly less effective on *SpatialSense* than on other existing spatial reasoning benchmarks. We evaluate multiple state-of-the-art visual relationship detection models on *SpatialSense*. Experimental results reveal that these models rely too much on dataset bias and now perform comparably to simple baselines. This demonstrates that adversarial crowdsourcing is effective for reducing dataset bias, and showcases that *SpatialSense* is an important step towards improving spatial reasoning capabilities of computer vision models.

2. Related Work

**Visual relationship recognition** Recognition of visual relations has recently emerged as a frontier of high-level computer vision moving beyond object recognition. Sadeghi & Farhadi [19] studied detecting visual phrases from images. A visual phrase can be a spatial relation (e.g. “person next to bicycle”). But their dataset contains only 17 unique visual phrases, 9 of which are spatial relations. This means that each spatial predicate only occurs with a small number of object categories: e.g. “next to” only occurs with “person”, “car”, and “bicycle”. Thus the dataset is unsuitable for evaluating a general understanding of “next to” that is agnostic to object categories.

Lu et al. [15] introduced the task of visual relationship detection — given an image, the algorithm predicts subject-predicate-object triplets as well as the object bounding boxes. They introduced the VRD dataset. VRD includes some spatial relations, but unlike our dataset, does not have negative examples. Thus evaluation using VRD has been based on Recall@K, which is ill-suited for spatial relations.

---

1The majority of images are RGB images from Flickr but 1,389 images are RGB-D from NYU Depth [20] which we include to make it possible to test the utility of depth information for spatial understanding.
because numerous valid spatial relations can hold in an image, making it difficult to choose the appropriate $K$. In addition, we will show in Sec.4 that the spatial relations in VRD are significantly easier to predict using simple priors. Visual Genome [11] is another dataset that is larger and has extensive annotations of visual relations. Similar to VRD, it covers a significant number of spatial relations but has no negative examples, and the spatial relations are easily predictable from simple priors.

The introduction of VRD and Visual Genome has spurred the development of new approaches for visual relationship detection. Successful methods typically build on top of an object detection module, and reason jointly over language and visual features [13, 27, 14, 5, 31, 28]. Multiple independent directions have been proven fruitful, including learning features that are agnostic to object categories [23], facilitating the interaction between object features and predicate features [23, 5, 13], and overcoming the scarcity of labeled data through weakly supervised learning [18, 28]. We benchmark some of the state-of-the-art approaches on our dataset and compare them with simple baselines based on language or 2D cues.

Peyre et al. introduced a dataset of unusual relations (UnRel) [13] sharing similar motivation to ours. To address the problem of missing annotations, the relations in UnRel are annotated exhaustively. Annotating every instance is made feasible by having a small predefined list of relations that are carefully designed to be unusual (such as “car under elephant”). This method is not scalable to a large number of relations. First, it is difficult to manually pick a large set of unusual relations. Second, for annotating every single instance, the amount of crowdsourcing efforts grows linearly with the number of relations. Our method samples interesting relations by encouraging crowd workers to discover them in images, and thus circumvents the scalability problem. As a result, UnRel has 76 unique relation triplets formed by 18 predicates while SpatialSense has 13,229 unique relations and 9 predicates.

**Dataset bias.** The bias in datasets can give models shortcuts allowing them to perform superficially good. Extensive research on dataset bias has been conducted in the context of visual question answering (VQA) [30, 6, 3, 9, 1]. Many VQA datasets suffer from language bias; the questions can be answered well simply by using language priors while ignoring images. Zhang et al. [30] balance the data for yes/no questions on abstract scenes. They show a question-image pair and ask the annotator to compose a new scene on which the answer for the question is different. Goyal et al. [6] applied the same idea to real images. Instead of asking annotators to create new scenes, they provide a few semantically similar images (measured by deep visual features) for the annotator to choose from.

Our adversarial crowdsourcing approach addresses the same issue (ensuring the input image is required to answer questions) in a substantially different way. Spatial relation recognition can be understood as a special case of VQA, where the questions are restricted to verifying spatial relations. In this sense, Zhang et al. [30] and Goyal et al. [6] ask the crowd to select hard images—images that defy the expected answers from language priors—with the questions fixed, whereas we ask the crowd to select hard questions with the images fixed. One potential advantage of selecting hard questions is that humans can easily compose new questions but cannot easily synthesize photorealistic images, and it can also be hard to find images that defy language priors—those images are by definition less common because language priors reflect common occurrences.

**Adversarial crowdsourcing.** Our approach for adversarial crowdsourcing is inspired by the “Beat the Machine” framework by Attenberg et al. [2], in which a crowd worker is challenged to find cases that will cause an AI system to fail. Adversarial crowdsourcing is related to active learning (e.g., [10, 22]) in that in both cases we seek difficult examples to improve learning. The key difference, however, is that in active learning it is the machine’s task to identify hard examples whereas in adversarial crowdsourcing it is on the human annotator.

### 3. Dataset Collection through Adversarial Crowdsourcing

A weakness observed in many datasets is a strong language bias, meaning that algorithms can perform well by exploiting language priors while ignoring the visual input [30, 6, 3, 9, 1]. In addition to language bias, there is another issue specific to spatial understanding. The real world is 3D but algorithms may exploit simple 2D cues without a true 3D understanding of space [8]. We address both issues in our adversarial crowdsourcing framework.

The data collection (Fig. 2) takes the form of annotators proposing spatial relations to make a robot fail. Given an image, the annotator comes up with a spatial relation by clicking on two objects, entering their names and selecting a predicate from a predefined list of 9 spatial predicates (above, behind, in, in front of, next to, on, to the left of, to the right of, under). Depending on whether we are collecting positive or negative examples, the proposed relations are required to be true or false. The robot then tries to guess the truthfulness of the relation using only object names and 2D coordinates (given by the clicks). The task is completed if the robot is wrong. Otherwise, the robot gives feedback about how the correct guess was made, and the annotator tries again. Additional crowdsourcing is used to verify the collected relations and annotate the object bounding boxes.

To reduce language bias and promote true 3D spatial understanding, we need the annotators to pick relations
that are difficult to predict given object names and 2D cues. The robot is therefore an ensemble of two models: a language-only model and a 2D-only model. The language-only model takes two object names along with the predicate, and outputs the probability that the relation holds. The object names are converted to word embeddings using Word2Vec [17], which are then encoded into a fixed-length feature vector by a gated recurrent unit (GRU) [4]. The one-hot encoding of the predicate is mapped to a vector of the same size by a linear layer. The three feature vectors are fused by element-wise addition, on top of which a 2-layer fully connected network outputs the probability. For the 2D-only model, linear layers map the object coordinates to feature vectors, and the prediction is made following the same procedure of the language-only model. The final output of the robot is the average of these two models. Initially, we train the robot on a dataset of 7,850 relations collected without adversarial crowdsourcing; During the collection of SpatialSense, we occasionally re-train the robot using all the then available data.

It is worth noting that we restrict the spatial predicate to a predefined list instead of letting human annotators enter free-form text. This is because the vocabulary of objects is vast and it would be cumbersome for the human annotator to choose from a long list of objects. In addition, a limited vocabulary of objects would restrict the annotator’s ability to pick the objects that form interesting spatial relations. In general, an image is rarely completely boring or predictable, so there will be an unusual or surprising spatial relation that beats the robot’s simple intuition. This setup thus provides an efficient way of obtaining spatial relations in the long tail.

We annotated 1,389 indoor images in NYU Depth [20] and 10,180 images in the wild from Flickr. When querying Flickr images, we use combinations of two keywords rather than a single keyword, following the approach adopted by COCO [15] to obtain images with diverse objects. Additionally, annotators can pick an image to annotate from a set of 8, so as to avoid images that do not have enough objects (e.g. a close up shot of a single foreground object). These techniques ensure that the images are complex scenes containing multiple objects necessary for relation reasoning.

We collected 4,342 relations on NYU images and 13,156 on Flickr images. Each relation consists of a spatial predicate, the names of two objects, and their bounding boxes in the image. There are equal number of positive and negative relations for each of the 9 predicates. 20% of the relations are reserved for testing and 15% for validation.

4. Analysis of the Dataset

The SpatialSense dataset has two key advantages compared to existing benchmarks. First, it contains positive
as well as negative relations. Second, it is constructed to be challenging; due to adversarial crowdsourcing, simple language and 2D priors are not enough to do well on this dataset. We now perform an in-depth analysis, comparing SpatialSense to VRD [16], Visual Genome [11] and a version of itself without adversarial crowdsourcing.

4.1. Comparison to Existing Datasets

Setup Since VRD and Visual Genome contain generic relations and no negative examples, we preprocess the data to make them comparable to our set: (1) Only the positive examples in SpatialSense are considered; the resulting dataset is referred to as SpatialSense-Positive; (2) We filter out non-spatial relations in VRD and Visual Genome; the resulting datasets are referred to as VRD-Spatial and VG-Spatial. In addition to discarding non-spatial relations, we also map the predicates in VRD and Visual Genome to their equivalents in our predefined list of 9 spatial predicates. For example, “rest on”, “park on” and “lying on” in VRD are all mapped to “on”. For Visual Genome, since there is no closed vocabulary, we examined the top-100 most frequent predicates to figure out the mapping. The detailed mapping is in the supplementary material.

Predicate distribution. Compared to VRD and VG, the predicate distributions in SpatialSense are less biased. Fig. 3 (Left) shows some frequent objects in VRD-Spatial, VG-Spatial and SpatialSense-Positive. For each object, it visualizes the distribution of predicates. For example, the top-left bar captures “something on table”, “something next to table”, etc.

For VG-Spatial, most of these objects are dominated by a single predicate, such as “something on table” or “something on street”. VRD-Spatial, which was annotated in-house rather than via crowdsourcing, looks more balanced; there are nevertheless a large number of “something on street” and “something under sky”. This supports our claim that many spatial relations in VRD and Visual Genome can be predicted without even looking at the image.

In contrast, SpatialSense-Positive has more balanced distributions of predicates, especially for “boring” objects. We call an object boring if its spatial relations to other objects are too predictable, such as table (“something on table”), sky (“something under sky”), and bed (“something on bed”). On the contrary, an object is “interesting” if its spatial relations are less predictable, for example, people and trees. An interesting observation is that SpatialSense is much more robust against boring objects. In existing datasets, the distributions of predicates are significantly biased for boring objects, such as “sky” and “table”. In SpatialSense, however, the distribution is modestly balanced even for “table”. Besides objects “on” tables, SpatialSense has plenty of examples of other relations such as objects “under” or “behind” tables. This property is desirable because it means less language bias. It is a result of adversarial crowdsourcing, in which annotators are forced to rely less on bias and focus more on unpredictable relations. There are almost always relations that are not easily predictable in any scene, and the key of adversarial crowdsourcing is to let annotators attend to these relations.

Fig. 3 (Right) shows the predicate distributions of the
top-50 objects in the three datasets. The figure for SpatialSense contains more varied colors, compared to existing datasets (especially to VG), which also illustrates that the relations in SpatialSense have diverse distributions of predicates.

2D Spatial distribution. SpatialSense is also less biased in 2D cues. Fig. 3 shows the distributions of 2D locations for the predicate “to the left/right of”. For each relation object₁-predicate-object₂, we take the 2D coordinate \((x_i, y_i)\) of an object to be the center of its bounding box, and compute the relative location normalized by the image size: \((\frac{x_i - x_2}{\text{image width}}, \frac{y_i - y_2}{\text{image height}})\). These are 2D points within \([-1, 1] \times [-1, 1]\) representing the 2D location of object₁ relative to object₂. For VRD-Spatial and VG-Spatial, the points for “to the left/right of” are almost separable. An algorithm can easily distinguish these two predicates from 2D cues alone. For SpatialSense-Positive, however, these two predicates have similar distribution, making it difficult to tell from 2D cues. This is because under our adversarial crowdsourcing framework, the annotators make extensive use of relative frame of references; a person standing to the left of a car can actually be on the right side of an image (when the car is facing towards the camera). Among other things, relative frame of references make SpatialSense less predictable from 2D cues and requires deeper spatial understanding.

Language and 2D baselines. To quantitatively show that SpatialSense is less biased in language and 2D locations, we examine the extent to which predicates can be determined from these cues. For each relation, given the two object names (bed, floor) and their bounding boxes, a model has to predict the correct predicate (on). We train/evaluate on the three datasets separately and compare accuracies. For the comparison to be fair, VRD-Spatial and VG-Spatial are randomly sampled to have the same size as SpatialSense-Positive. We adopt the official train/test split for VRD-Spatial and SpatialSense-Positive; for VG-Spatial, 20% of the images are reserved for testing.

The model architectures are similar to those used in data collection: For the language-only model, object names are encoded to fixed-length vectors using Word2Vec followed by a GRU. The two vectors are fused into one by element-wise addition, which is then classified by a 2-layer fully connected network. For the 2D-only model, bounding box coordinates are encoded by linear layers, and then fused and classified following the same procedure. The models are trained with cross-entropy loss. For VRD-Spatial and VG-Spatial, the experiments are replicated 100 times to smooth the effect of random sampling.

Table 1 shows the results of classifying predicates. SpatialSense-Positive has the lowest accuracies, which confirms that it is less predictable from language and 2D cues. We got similar results on VRD-Spatial using SVMs with RBF kernel, but they are impractical to be trained on VG-Spatial due to the large size. We also tried to encode object names simply by averaging the embedding of each word, but GRUs turned out to perform slightly better.

Table 1. SpatialSense has lower accuracies when predicting the spatial predicate from the object pair using language and 2D cues, which implies less bias. VRD-Spatial and VG-Spatial are randomly sampled to have the same size as SpatialSense-Positive; their experiments are replicated 100 times to produce the average accuracies and standard errors.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Language</th>
<th>2D Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRD-Spatial</td>
<td>66.9 ±0.1</td>
<td>59.6 ±0.1</td>
</tr>
<tr>
<td>VG-Spatial</td>
<td>76.0 ±0.1</td>
<td>65.3 ±0.1</td>
</tr>
<tr>
<td>SpatialSense-Positive</td>
<td>39.8</td>
<td>43.4</td>
</tr>
</tbody>
</table>

4.2. The Effect of Adversarial Crowdsourcing

It is adversarial crowdsourcing that reduces bias. We support this claim by comparing with a dataset constructed without adversarial crowdsourcing. The dataset is collected by annotators who propose positive spatial relations freely (without the need to beat a robot), and the negative relations are randomly generated. All relations go through human verification and the resulting dataset is referred to as SpatialNaive, which contains 3,015 images; 3,925 positive relations; and 3,925 negative relations. Just like SpatialSense, each predicate has equal number of positive and negative relations.

We quantify the amount of bias in SpatialSense and SpatialNaive by comparing the performance on the task of spatial relation recognition. Given two object names, their bounding boxes and a predicate, a model classifies whether or not the relation holds. Since SpatialNaive is smaller, we randomly sample a subset of SpatialSense, enforcing the two datasets to have exactly the same number of relations for each predicate. The model architectures are the same as those used for collecting data (described in section 3), except that now the 2D locations are represented by object
The results are summarized in Table 2. The language-only model and 2D-only model performs much worse on SpatialSense, which confirms the effectiveness of adversarial crowdsourcing to reduce dataset bias, especially the bias in language.

5. Baselines for Spatial Relation Recognition

Having verified that SpatialSense is an effective benchmark for spatial relation recognition, now we evaluate multiple baseline approaches on SpatialSense, including simple baselines based on language and 2D cues as well as state-of-the-art models for visual relationship detection. Experimental results reveal the difficulty for state-of-the-art models to go beyond simple priors and learn to reason about visual content: a simple baseline based on 2D cues performs competitively with state-of-the-art models. We also conduct a human evaluation on the testing data, quantifying the ambiguity in the task and setting an upper limit on the performance of algorithms.

Model architectures. The task is spatial relation recognition: given the image, two objects (their names and bounding boxes) and a spatial predicate, the model classifies whether the relation holds. Two simple baselines are evaluated: a language-only model and a 2D-only model. Their architectures are the same as in section 4.2. We also report the performance of combining their predictions by a weighted average. We evaluate two state-of-the-art models: deep relational network (DRNet) [5] and visual translation embedding network (VTransE) [27]. They were created for visual relationship detection but can be adapted to our task straightforwardly: First, object detectors are replaced by ground truth objects. Second, object names are encoded using word embeddings rather than one-hot encoding, since SpatialSense has unconstrained object categories. Third, for each relation subject-predicate-object, the model takes subject and object as input, and generates scores for all predicates; the score for that particular predicate is the final binary classification score. For details of the model architectures, please refer to the supplementary materials.

Implementation details. Object names are encoded to fixed-length vectors using Word2Vec followed by a GRU. When combining the language and 2D baselines by a weighted average, we find 80% from 2D and 20% from language to perform well (measured by validation accuracy).

For DRNet, we crop the union bounding box of the two objects, resize it to 280 × 280 and normalize the pixel values by the mean and standard deviation of all training images. During training, we then crop it with random size and aspect ratio, resize to 224 × 224 and apply color jittering; during testing, we simply take a 224 × 224 crop at the center. For VTransE, we resize the entire image rather than the union bounding box and others remain the same. The networks are trained using RMSProp [21]. The learning rate shrinks by a factor of 10 when the validation accuracy fails to increase in a consecutive 3 epochs. All models are trained for 25 epochs. For DRNet and VTransE, it typically takes 4 ~ 5 hours on a single GeForce GTX 1080 GPU.

Analyzing the results. Table 3 summarizes the testing accuracies. Due to the challenging nature of SpatialSense, the best models perform around 70%, which is quite low for a binary classification task. DRNet and VTransE are the best two models without ensemble, which confirms that models for visual relationship detection can be well adapted to our task. An interesting point is that the 2D baseline performs closely to state-of-the-art models, and is on a par with DRNet when combined with the language baseline. This suggests state-of-the-art models might learn to rely too much on 2D cues and fail to develop deeper visual reasoning capabilities.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Language</th>
<th>2D Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpatialNaive</td>
<td>69.2</td>
<td>71.3</td>
</tr>
<tr>
<td>SpatialSense</td>
<td>56.4 ± 0.1</td>
<td>65.2 ± 0.1</td>
</tr>
</tbody>
</table>

Table 2. SpatialSense has lower accuracies on spatial relation recognition than SpatialNaive — an ablation dataset constructed without adversarial crowdsourcing. The results confirm that adversarial crowdsourcing plays a crucial role in reducing bias. SpatialSense is randomly sampled to have the same size as SpatialNaive; its experiments are replicated 100 times to produce the average accuracies and standard errors.

We evaluate two state-of-the-art models: deep relational network (DRNet) [5] and visual translation embedding network (VTransE) [27]. They were created for visual relationship detection but can be adapted to our task straightforwardly: First, object detectors are replaced by ground truth objects. Second, object names are encoded using word embeddings rather than one hot encoding, since SpatialSense has unconstrained object categories. Third, for each relation subject-predicate-object, the model takes subject and object as input, and generates scores for all predicates; the score for that particular predicate is the final binary classification score. For details of the model architectures, please refer to the supplementary materials.

Implementation details. Object names are encoded to fixed-length vectors using Word2Vec followed by a GRU. When combining the language and 2D baselines by a weighted average, we find 80% from 2D and 20% from language to perform well (measured by validation accuracy).

For DRNet, we crop the union bounding box of the two objects, resize it to 280 × 280 and normalize the pixel values by the mean and standard deviation of all training images. During training, we then crop it with random size and aspect ratio, resize to 224 × 224 and apply color jittering; during testing, we simply take a 224 × 224 crop at the center. For VTransE, we resize the entire image rather than the union bounding box and others remain the same. The networks are trained using RMSProp [21]. The learning rate shrinks by a factor of 10 when the validation accuracy fails to increase in a consecutive 3 epochs. All models are trained for 25 epochs. For DRNet and VTransE, it typically takes 4 ~ 5 hours on a single GeForce GTX 1080 GPU.

Analyzing the results. Table 3 summarizes the testing accuracies. Due to the challenging nature of SpatialSense, the best models perform around 70%, which is quite low for a binary classification task. DRNet and VTransE are the best two models without ensemble, which confirms that models for visual relationship detection can be well adapted to our task. An interesting point is that the 2D baseline performs closely to state-of-the-art models, and is on a par with DRNet when combined with the language baseline. This suggests state-of-the-art models might learn to rely too much on 2D cues and fail to develop deeper visual reasoning capabilities.
Table 3. The testing accuracies of baseline methods on spatial relation recognition. The 2D baseline performs closely to state-of-the-art models, which suggests state-of-the-art models might learn to exploit simple priors and fail to develop deeper visual reasoning capabilities.

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall</th>
<th>above</th>
<th>behind</th>
<th>in</th>
<th>in front of</th>
<th>next to</th>
<th>on</th>
<th>to the left of</th>
<th>to the right of</th>
<th>under</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language-only</td>
<td>60.1</td>
<td>58.3</td>
<td>62.2</td>
<td>54.4</td>
<td>58.2</td>
<td>57.4</td>
<td>61.2</td>
<td>54.5</td>
<td>56.2</td>
<td>72.4</td>
</tr>
<tr>
<td>2D-only</td>
<td>68.8</td>
<td>58.0</td>
<td>66.9</td>
<td>70.7</td>
<td>63.1</td>
<td>62.0</td>
<td>76.0</td>
<td>66.3</td>
<td>74.7</td>
<td>67.9</td>
</tr>
<tr>
<td>Language + 2D</td>
<td>71.1</td>
<td>61.1</td>
<td>67.5</td>
<td>69.2</td>
<td>66.2</td>
<td>64.8</td>
<td>77.9</td>
<td>69.7</td>
<td>74.7</td>
<td>77.2</td>
</tr>
<tr>
<td>DRNet [5]</td>
<td>71.1</td>
<td>59.7</td>
<td>71.7</td>
<td>70.1</td>
<td>66.9</td>
<td>59.3</td>
<td>78.5</td>
<td>66.9</td>
<td>71.2</td>
<td>77.9</td>
</tr>
<tr>
<td>VTransE [27]</td>
<td>69.4</td>
<td>61.8</td>
<td>71.3</td>
<td>65.7</td>
<td>65.6</td>
<td>55.9</td>
<td>75.8</td>
<td>62.4</td>
<td>70.5</td>
<td>77.9</td>
</tr>
<tr>
<td>Human</td>
<td>94.6</td>
<td>90.00</td>
<td>96.26</td>
<td>95.0</td>
<td>95.78</td>
<td>94.5</td>
<td>95.7</td>
<td>88.8</td>
<td>93.2</td>
<td>94.14</td>
</tr>
</tbody>
</table>

Figure 6. Failing examples of the language and 2D baselines. The 2D baseline fails to consider relative frame of reference (e.g., “wall to the right of bike”) and depth information (e.g., “light lamp next to sofa” and “chair under cupboard”). The language baseline fails when a frequent spatial relation does not occur in a particular image (e.g., “cat on tree”), or a technically valid spatial relation is expressed in an unusual way (e.g., “blanket under paper”).

implies it makes similar predictions to a 2D-only model and supports our conjecture.

Some random failing examples for the language and 2D baselines are shown in Fig. 6. It is evident that models based solely on 2D cues struggle in scenarios that involves relative frame of reference (e.g., “wall to the right of bike” in the second row) or when depth information plays an important role (e.g., “light lamp next to sofa” and “chair under cupboard”). In contrast, language cues fall short when a frequent spatial relation does not occur in a particular input image (e.g., “cat on tree” in the first row), or a spatial relation is technically valid but people typically do not say it in that way (e.g., “blanket under paper”). These observations indicate neither language nor 2D cues is sufficient for spatial relation recognition. Beyond these simple cues, it is crucial to learn visual reasoning that elude current state-of-the-art. Our benchmark takes a step towards that goal by providing a more accurate gauge of a model’s visual reasoning ability.

Human evaluation. Recognizing spatial relations is inherently noisy; it is not always clear whether a spatial relation holds. In order to quantify the amount of ambiguity in the task and set an upper limit on the performance of algorithms, we conduct a human evaluation, in which annotators are asked to classify the spatial relations in the testing data of SpatialSense. Multiple human responses on the same relation is merged by majority vote. For quality control purpose, the annotators who answer “yes” more than 80% of the time are considered outliers, and their responses are excluded. We collect 10,205 predictions and the accuracies are in the last row of Table 3. Although not perfect, humans perform very well on this task, reaching an accuracy of 94.6%. The large gap between humans and algorithms provides a large room for future improvement on this benchmark.

6. Conclusion

We introduced the task of spatial relation recognition and SpatialSense, a dataset specializing in this task. It was constructed through adversarial crowdsourcing, which reduces dataset bias significantly, resulting in a more challenging and rigorous dataset. We benchmark multiple baselines on SpatialSense; a simple 2D baseline performs competitively with state-of-the-art models. This reveals that state-of-the-art models rely too much on dataset bias and proves SpatialSense to be an effective benchmark for spatial relation recognition.
References


