# SynthRel0: Towards a Diagnostic Dataset for Relational Representation Learning

Daniel Dorda , Moin Nabi

{daniel.dorda, m.nabi}@sap.com
SAP ML Research, Berlin

## Abstract

*This work analyses the sources of complexity in scene graph proposal problems, and develops a mathematical framework for efficiently designing synthetic relationship models. An entropy based metric is proposed for measuring the ambiguity of relational datasets. Using these tools, a first approximation to a synthetic dataset is given, and experiments with a simple baseline are performed to show how the difficulty of the proposed task changes with varying dataset parameters, like missing annotation ratio and feature granularity. These experiments illuminate the desirable qualities of future synthetic relationship datasets.*

## 1. Introduction

Scene understanding is a fundamental problem of computer vision which can be posed in a variety of ways. Datasets such as Places [22] assign scene-level labels to images, and pose "holistic scene understanding" as a classification task. This problem can be solved by the CNN based techniques which spearheaded the deep learning revolution.

However, when scenes become complex, this holistic formulation loses its validity. A singular label is bound to be ambiguous and poorly defined, and cannot capture the subtle interactions between multiple objects.

Scene graph proposal (SGP) is a significantly harder task. Its goal is to detect objects and label the relationships between them [7]. The difficulty can be partially attributed to the fact that SGP confounds many different problems, and the approaches to solving them are far from clear.

For example, recent works question the representation learning power of graph convolutions [8, 17], which are employed by many models [19, 11]. Sadly, with current relational data, validating these claims would be arduous.

Additionally, the reliance on RCNNs [12] for object detection could cause problems if it is shown that RCNN outputs do not contain enough information to solve the relationship proposal problem.

Furthermore, the datasets which currently exist introduce further unwanted complications. This is because of biases [20] and missing annotations [15].

Biases cause models to overfit to simple positional and linguistic priors. Work has been done to create datasets without such bias, but it is hindered by the cost of gathering and annotating rare, unbiased data [10, 20].

Missing annotations complicate evaluation by making it impossible to tell the difference between false positive proposals and un-annotated ones, and can cause problems during model training [15].

It is currently unfeasible to address these problems by collecting larger natural image datasets. We believe that appropriately designed synthetic datasets address these shortcomings. They give us access to richly labelled data, and allow us to control for bias. Furthermore, by adjusting the model's inputs and the way relationships are defined, we can monitor when the model fails to perform, and diagnose the aforementioned problems.

Unfortunately, designing good synthetic relationships is challenging. It is easy to inadvertently introduce biases, or make the task too easy. Crucially, failing to capture the nuances of real relationship dynamics will lead to datasets which aren't able to diagnose any real problems.

For this reason, we develop a mathematical framework which helps with defining a synthetic relationship dataset, and measuring its complexity using conditional entropy.

We illustrate how these methods work on a toy dataset, SynthRel0. Composed of simple objects which interact in one type of relationship, SynthRel0 is shown to be too easy to solve. However it allows us demonstrate the strengths of synthetic data. We document model behaviour under held-out annotations, illustrate techniques for making future datasets more challenging, and showcase how these changes affect the conditional entropy of the relationships.

## 2. Related Work

### Relationship Datasets

Most dominant datasets across the scene graph literature fall into the same category of general relationship proposal. These datasets include Visual Genome [4], Visual Relationships [7] and Open Images [5]. They annotate the strongest relationships across a limited number of dominant objects in the scene.

This provides a great resource for learning common relationships for downstream tasks such as VQA, scene captioning, and image retrieval [14, 6, 3]. However, it introduces exploitable biases, allowing models to improve their classification score without learning meaningful relationship dynamics.

For example, in Visual Genome, 89% of relationships involving a table contain the predicate "on" [20]. Thus, a lazy network will learn to predict that everything interacting with a table is on it, which is a fantastic loss-minimising strategy when training with biased data.

Missing annotations, common in popular datasets [15], cause further problems. During evaluation, it becomes impossible to discriminate between false positives and unannotated correct detections. By including negative annotations, such as in Open Images, it is possible to mitigate this problem. However, this increases the cost of obtaining annotated data, and limits the number of training samples available.

Some recent works highlight a growing need for a different style of relationship data. For example, Unusual Relations [10] focuses on creating a test set which features atypical relationship triplets, enabling the generalisation capabilities of the network to be evaluated. Yang et al. propose SpatialSense, a dataset of adversarial examples [20], which facilitates diagnosing the aforementioned lazy network problem. They evaluate several SOTA models against simple baselines, and show that complex models fail to capture much information beyond positional and linguistic priors.

However, collecting thoroughly annotated, real data comes with high overheads. Therefore, their work is limited by only containing 9 predicate classes, which are adequate for coarsely describing spatial relationships, but do not take into account the more complex functional relationships, or the long tail distribution of real relationships.

Synthetic datasets are able to overcome these problems. Fully annotated data is a given, and meta-annotations (such as whether scenes contain rare or non-trivial relationships) allow for diagnosing particular weaknesses. With an appropriate synthetic generator, we are also able to freely control the number of object and predicate classes, and analyse how models perform as these factors vary.

### Synthetic Datasets

Synthetic datasets have had a significant impact in other fields, such as VQA, where it is hard to gather large, richly annotated, unbiased datasets of natural images. Early datasets such as DAQUAR [9] suffered from not controlling for question conditional bias, low question variance, and not having appropriate functional frameworks for expressing their generation process.

The CLEVR [2] dataset tackled these issues and became widely adopted across the field. It gave a robust framework for scene and question generation using a functional-programming style formulation.

However, its focus on question answering makes it ill suited to SGP, as CLEVR defines only five relationship types, four spatial and one colour dependent.

Therefore, evaluating a relational reasoning module on CLEVR [13] does not reflect on the performance of the module on the complicated relationship dynamics observed in natural images. This shows a lack of theoretical foundations necessary for expanding and modifying synthetic scene datasets.

### Scene Graph Proposal

Single relationship proposal techniques [7, 10] serve as a conceptual predecessor of scene graph generation. However, objects in a scene are not isolated, and there exists mutual information between relationships. This led to the logical extension that by sharing information between objects, the initial proposals for both objects and relationships can be improved.

Xu et al. [18] used an RNN architecture to iteratively refine the scene graph.

An improvement over this approach was realised by employing graph convolutional networks [1]. GCNs provided tools which were more effective at distributing information across graph structures, such as graph attention [16]. Many papers focus on ways to implement these techniques.

Graph-RCNN [19] uses novel relationship proposal based on object priors, followed by graphical attention which refines the initial estimates. Qi et al. [11] also utilise graphical attention to focus on effectively diffusing information through the scene graph.

There are several noteworthy similarities between these models. First, all the models share the common RCNN backbone, used for object detection and feature extraction.

Additionally, all the models use supervised training. This means they are vulnerable to missing, erroneous or sparse annotation data. Unfortunately, many works opts to ignore rare annotations, due to the noisiness of the labels on datasets without a closed vocabulary, such as Visual Genome, or due to a lack of an adequate number of training samples in other cases.

## 3. Dataset Design

### 3.1. Mathematics of Scene Graphs

In this section, we develop a mathematical notation for describing the scene graph proposal problem. First, let's describe the variables of a scene in plain language, to gain a general understanding of the problem before introducing mathematical notation.

We define an abstract scene as a set of objects, described by features. Objects form pairwise relationships with each other, and the type of relationship is determined by the features of the directly interacting objects, as well as the features of the nearby objects, which contribute to a wider scene context.

We can say the scene *S* contains *N* objects, and each object *i* has a corresponding $f_i$ such that the feature set of *S* is $F = \{f_1, ..., f_N\}$.

Further, we say that there exists a mapping from *F* to *R*, the set of relationships in scene *S*, i.e. $p_r : F \mapsto R$. The sets *F* and *R* define the vertices and edges of the scene graph, respectively.

Note that these are prescribed qualities, which nonetheless conform to the intuition behind the problem. However, several of these inoffensive assertions require further discussion. How exactly can the features $f_i$, and the relationships in *R* represented?

First, let's consider the features. The de-facto standard is to use the output of an RCNN [12], which for each object gives a triplet $(c, \mathbf{x}, f_v)$, corresponding to the class, position and CNN output layer activations.

We generalise object features as a set of correlated, yet orthogonal representations, i.e. $f_i = \{f_i^1, ..., f_i^K\}$. This representation corresponds to the intuition that features such as class, pose, position, size, or appearance are covariant, yet distinct, and could be disentangled, given the right feature extractor. Additionally, we can define the set of distributions from which the features $f^k$ are sampled as $P_f = \{p(f^1), ..., p(f^K)\}$.

In practice, it is difficult to derive this representation. Firstly, in natural images, the set of relevant features is not known. Further, the image pixels of an object are simply one sample from one element of $P_f$, from which we are required to infer much higher level features. RCNN networks give us just one way of mapping pixel values to the set of features $(c, \mathbf{x}, f_v)$, with no indication of how well suited this feature set is to solving the SGP problem.

The first power of synthetic datasets lies in being able to define all the relevant features of generated objects by specifying $P_f$. These features can be sampled from arbitrarily complex distributions, but knowing them during training allows us to skip the feature detection stage of most scene graph proposal networks. This isolates the scene graph proposal stage, leading to faster training. Additionally, it lets us determine exactly how informative each element of $P_f$ is during relationship inference, allowing a well annotated dataset to be used for diagnosing model robustness.

Having developed the above representation for a scene's objects, we now consider the relationship proposal mechanism. For simplicity, we assume that $R = \{r_{ij} \forall i, j \in \{1, ..., N\}\}$ where $r_{ij} \in \mathbb{R}^d$ encodes the relationship between a pair of objects *i* and *j*.

This assumption has limitations, such as being unable to express relationships which involve multiple objects. Such relationships, e.g. "Alice, Bob and Charlie are playing football", are expressed naturally by scene hypergraphs. However, the field is not refined enough to merit considering such esoteric models, making this assumption valid.

Next, we can formulate the relationship proposal mapping $p_r : F \mapsto R$, and reason about the strata of relationship complexity.

For example, models which predict relationships by considering the pairwise interaction between objects [10, 7], assume that the function $p_r$ can be factorised into a series of functions $p_r' : f_i, f_j \mapsto r_{ij}$ which operate on object pairs, i.e. $R = \{p_r'(o_i, o_j)\ \forall o \in O\}$.

However, it is easy to observe that this assumption is faulty, since real relationships are influenced by scene context, and $p_r$ is likely to operate on the more than just two feature vectors at a time.

Additionally, we are able to consider how informative the elements $f_i^k$ of each $f_i$ are to relationship proposal. For example, one of the implications of the work of Yang et al. [20], is that 2D position and language priors are either far more informative than visual activations, or the functions which map these features to relationships are far easier to learn than the mappings between activations and relationships. This kind of reasoning is closely tied to the entropy based metrics put forward in Section 3.4.

During the design of synthetic datasets, we propose a custom function $p_r$. In the synthetic universe, we are able to control how many types of relationships are possible, and how they are defined by the interactions of each objects' features.

### 3.2. Synthetic Relation Datasets

Following the formulation in the previous section, to create a complete synthetic universe we need to specify, implicitly or explicitly:

- The form of object features *f*
- A sampling strategy $P_f$
- A relationship proposal function $p_r$

To guide the design of the above parameters, we identify factors contributing to the complexity of SGP on natural images.

The following properties were identified:

1. Distinctions between functional and geometric relationships
2. Ambiguous relationships
3. Missing annotations
4. Conditional dependence between elements of $P_f$
5. Adversarial examples
6. Different forms of the features in $f$

The above points are explained in detail below.

We propose that the difference between functional and geometric relationships, such as `dog-wears-hat` vs `dog-under-hat`, arises when multiple $p_r$ are applied to the same objects, but operate on different elements $f$. For example, consider a fictional object whose $f = \{\mathbf{x}, c\}$, a combination of a position vector and a colour. We can define two $p_r$, a geometric $p_{r1} : \mathbf{x} \mapsto r$ and a colouric $p_{r2} : c \mapsto r$. Thus, relations like `on, below` are controlled by $p_{r1}$, whereas `is-the-same-colour-as` is controlled strictly by $p_{r2}$. This fits the intuition behind real relationships, where geometric relationships depend only on the position of two objects, whereas other relationships consider more esoteric object features. With multiple $p_r$, we could emulate a wealth of functional relationships.

Ambiguity exists in relationships, where a pair of objects can be annotated with multiple correct relationships. Some works deal with multiple predicate labels for an object pair by randomly sampling one predicate [21]. However, it would be insightful to observe how models behave in an ambiguous environment, and what strategies can be developed to cope with it. In Section 3.4, entropy is explored as a measure of ambiguity, and a method for increasing dataset ambiguity is given.

Missing annotations can be easily incorporated into synthetic datasets, however, care needs to be taken to ensure that relationships are removed in a similar way they would be in real datasets. In real data, only the weakest relationships are left unannotated. Thus, synthetic data needs to be able to order the relationships in a scene by a metric of strength, and when removing them, remove the weakest ones first. Failure to do so can lead to unexpected model behaviour, as discussed in Section 4.1.

Two kinds of feature dependence need to be considered, intra- and inter-object. To illustrate the significance of each of the former effects, consider a real scene with two objects. Given that object A is red, it is more likely to be an apple than a cucumber. Given that A is a red cucumber, does that increase the probability of the other object being an art student? One strategy when sampling synthetic features is to make them as independent of each other as possible, to minimise biases which lazy models will exploit. However, we argue that this naive approach would limit the informativeness of object relationships, most of which are defined by the dependencies between objects. Further work is required to propose appropriate rules for conditionally sampling object features.

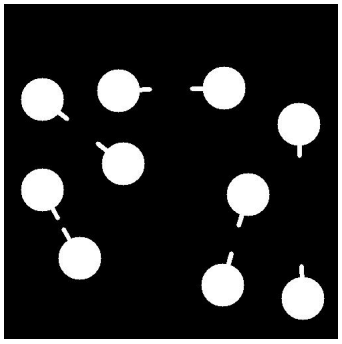

Figure 1. A visualisation of a SynthRel0 scene

Adversarial examples come from relationships which are easy to misclassify. For example, SpatialSense [20] contains abundant spatially adversarial examples, where it is easy to falsely infer the relationship between two objects based only on their position. Correctly identifying the relationship requires reasoning about about the remaining features of the object, which are usually much less significant to the relationship.

Finally, real data is likely to be described by a number of data formats, such as one-hot encodings, real valued continuous or limited-domain tensors. Synthetic models need to be able to effectively leverage various data types, which ought to be provided by a well-designed synthetic dataset.

It is worth mentioning that badly designed synthetic datasets would not be useful either for complex diagnostic tasks, or for the testing of new ideas. However, when designed with care, they could prove to be a useful tool for the evaluation of theoretical capabilities of new models. This could be ensured by designing an adequately complex $p_r$, and presenting the model with uniquely sampled data, designed to test characteristics such as few-shot potential, generalizability, and robustness to missing annotations or adversarial examples.

### 3.3. SynthRel0

SynthRel0 is a toy dataset which serves as the first approximation to a complete synthetic relationship dataset. While it is missing many of the desirable features identified in Section 3.2, we demonstrate it to highlight challenges inherent in relationship dataset design. When evaluating the dataset on our benchmarks, we used 12k training and 3k val scenes.

Each scene in SynthRel0 is composed of 10 objects. This

constraint allows for our simple baseline, which is incapable of abusing this fact, to be tested quickly and efficiently.

Each object's features $f$ are given by position and orientation, i.e. $f = \{\mathbf{x}, \theta\}$. These objects can be visualised, as in Figure 1, as circles with a beak. According to the chosen relationship proposal function $p_r$, two objects are in a relationship if their beaks point directly at each other. This function, defined below, chooses for each object one relationship, based on the minimum of a cost function proportional to perpendicular distance and angular affinity.

$$r_{ij} = \arg\min_j [d(f_i, \mathbf{x}_j) \cdot a(\theta_i, \theta_j)]$$

$$d = (\mathbf{x}_i - \mathbf{x}_j) - \left( ((\mathbf{x}_i - \mathbf{x}_j) \cdot \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \end{pmatrix}) \cdot \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \end{pmatrix} \right)$$

$$a = c + cos(\theta_i - \theta_j)$$

Where $d$ and $a$ are distance and angular affinity functions, respectively, and the constant $c = 2$. Due to feature sampling technique selected, each object has a partner whose cost is exactly 0.

The simplicity of this dataset allows us to test techniques for making synthetic relationship data more complicated, while also revealing pitfalls which future datasets need to avoid.

For example, without due care given to the generation process, it is easy to introduce biases, such as when assigning object IDs. During our first approach, we assigned them sequentially. This introduced a bias in the ground truth adjacency matrix, and models trained on the biased dataset performed far better than ones trained on a dataset where this bias was eliminated.

### 3.4. Measuring dataset complexity

One measure of the complexity of a system is its unpredictability. It is also interesting to measure how this unpredictability changes as we obtain more information. These quantities are elegantly expressed by conditional information entropy.

$$H(X|y) = -\sum_x p(x|y) \log p(x|y)$$

Thus, the entropy of a relationship conditional on some features tells us how inherently unpredictable this relationship is. For a relationship $r_{ij}$ and a set of objects' features $F$, we can define several quantities of interest.

$H(r_{ij})$ gives the prior entropy, which ought to be high, as relationships ought to be unpredictable without looking at object features.

$H(r_{ij}|f_i, f_j)$ describes how unpredictable a relationship is based on the two implicated objects. A high entropy can indicate that a relationship is ambiguous, or that some informative features have not been considered.

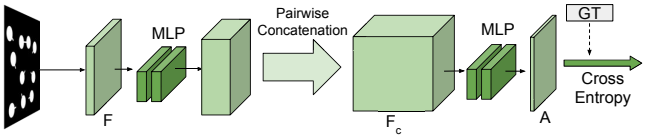


Figure 2. Baseline model, composed of two MLPs and trained with cross entropy loss

$H(r_{ij}|f_i^k, f_j^k)$ is the entropy given only partial object features. If this value is low, it means that the conditioning features are highly informative. Informative features make the problem solvable, but also lead to overfitting issues.

$H(r_{ij}|F)$ is the entropy given all object features. If this value is non-zero, it means the problem is inherently ambiguous, and cannot be solved perfectly.

These quantities are easy to measure on synthetic data, with hand-crafted statistics. To demonstrate, we will explore how entropy changes on SynthRel0 as the features are made more ambiguous.

For each scene $p(r_{ij}) = 0.1$. This allows us to calculate entropy as $H(R) = 0.469$, which is low, due to the low cardinality of the set of possible relationships and the high prior probability of no relationship. The prior entropy of real data is higher, because the relationship set is larger, and relationships are more equiprobable without given features.

To calculate $H(R_{ij}|[f_i, f_j])$ on SR0, we can use the relationships proposal function $p_r$. We must approximate the calculation over a continuous $f$ by discretizing the feature space.

Each position can be assigned to one of $n$ sectors, and each orientation to one of $m$ nominal directions. We can estimate the probability $p(r_{ij}|[f_i, f_j])$ using a frequentist approach, by finding out, for a given $[f_i, f_j]$, the fraction of times a relationship between the two objects occurs.

It is trivial to observe that as the granularity of the discretization decreases, $H(R_{ij}|[f_i, f_j])$ goes to zero, because the relationship is completely specified by the given features.

Further work is needed to determine how to scale this method to larger, more elaborate datasets, since the discretization schemes used here are specific to the simple features of SynthRel0.

## 4. Experiments

We ran experiments on a simple benchmark, varying the fraction of annotations which are removed during training, $\gamma$, and the complexity of the dataset. Two ways of increasing complexity were tested: increasing ambiguity by modifying the input using the bucketing scheme described in Section 3.4, and adding a non-informative feature vector to $f$ which the network must learn to ignore. We use orientation bucketing, with 4 buckets, meaning that every value of $\theta$ is rounded to the nearest of four cardinal orientations.

Model performance was evaluated using Recall@1. We use this strict metric due to the simplicity of the problem, since a more forgiving metric would fail to be discriminative enough.

The benchmark used was a network of two MLPs, trained using cross entropy loss on binary predictions between object pairs. The input to the model is a tensor of features $F$. We directly use the positions and orientations of the objects to make this tensor, since it would be trivial to train a detection network on the objects in the image. Skipping the object detector increases computational efficiency and focuses on graph proposal instead.

### 4.1. Results

Based on the recall curves in Figure 3, we can make several interesting observations.

On the uncorrupted data, the model performs well, reaching >99% R@1. As $\gamma$ is increased, the model converges more slowly, but the final recall is almost unaffected. With 90% of the annotations removed, only a 10% drop in performance is observed.

This indicates that as predicted, the task is too simple. To make it more difficult, the feature set of each object can be expanded, and the number of possible relationships and complexity of the proposal functions can be increased.

Additionally, we see how removing relationships at random does not make the task more difficult in a predictable manner. We believe this is due to the fact that SynthRel0 has no means of ranking the strength of relationships in the scene. Due to this, there are no discernible differences between removed and retained relationships, which causes a properly tuned model to simply propose all feasible relationships as positives.

Bucketing makes the problem more difficult. The performance is worse, but far less than anticipated. At $\gamma = 0$, achieving R@1 of 87%. Interestingly, performance decreases dramatically as $\gamma$ is increased.

This shows that, as predicted, increasing the entropy of the task makes it more difficult, and the model is unable to mimic the performance it achieved previously. However, it reveals an unforeseen dependence on $\gamma$, which highlights the fact that synthetic relational dataset design is a complicated problem with many unknown interactions between the models and data.

Finally, when uninformative features are added, the model learns to ignore them extremely quickly, and performance is identical to that on the uncorrupted data. This means that simple schemes for adding non-informative features are not effective, and more complicated modes of feature obfuscation should be pursued.

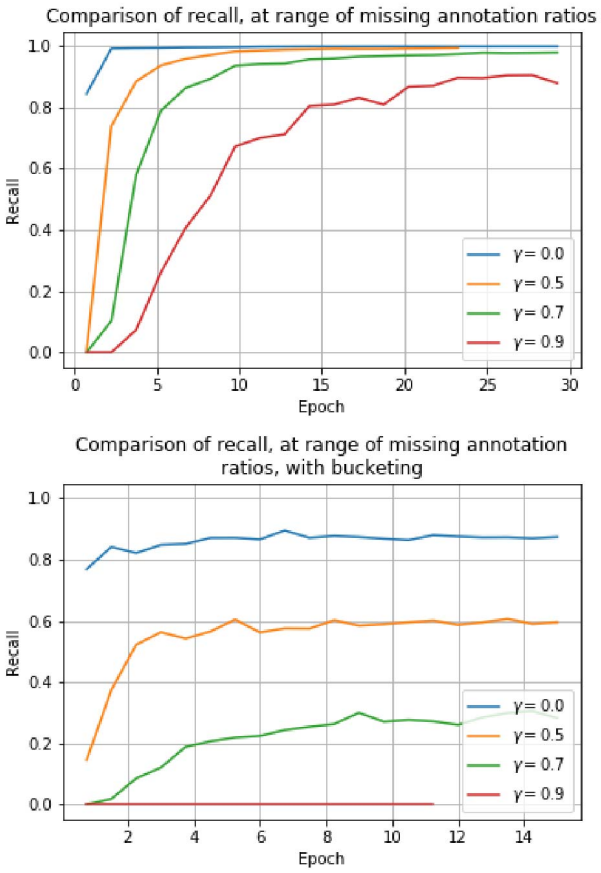


Figure 3. Recall curves for the uncorrupted data (top) and the bucketed data (bottom), show that performance decreases as conditional relationship entropy increases

## 5. Conclusion

Development of synthetic data for scene graph recognition is an open problem that poses many challenges.

It is not clear what the characteristics of natural relationships are, and emulating them artificially is even harder.

Our mathematical framework enables structured reasoning about the problem, and with its help many relationship datasets can be developed. Further work can refine the general class of relationship models proposed here, to better express the complexity of real data.

Likewise, a detailed analysis of the factors contributing to the complexity of real data can inform the design of more complex relationship proposal functions. We recommend studying how changing features and proposal functions affects the behaviour of current models.

There is a demand for better diagnostic data, and we believe the answer is synthetic. We hope this work serves as a stepping stone towards a powerful dataset, which highlights the weaknesses of current models and moves future research down new, unexplored avenues.

## References


[1] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[2] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890, 2016.

[3] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[4] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[5] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018.

[6] X. Li and S. Jiang. Know more say less: Image captioning based on scene graphs. *IEEE Transactions on Multimedia*, 21:2117–2130, 2019.

[7] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016.

[8] T. Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

[9] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. *CoRR*, abs/1505.01121, 2015.

[10] J. Peyre, I. Laptev, C. Schmid, and J. Sivic. Weakly-supervised learning of visual relations. In *ICCV*, 2017.

[11] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo. Attentive relational networks for mapping images to scene graphs. *ArXiv*, abs/1811.10696, 2018.

[12] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[13] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. W. Battaglia, and T. P. Lillicrap. A simple neural network module for relational reasoning. *CoRR*, abs/1706.01427, 2017.

[14] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2017.

[15] S. Tripathi, A. Bhiwandiwalla, A. Bastidas, and H. Tang. Heuristics for image generation from scene graphs. 2019.

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[17] F. Wu, T. Zhang, A. H. S. Jr., C. Fifty, T. Yu, and K. Q. Weinberger. Simplifying graph convolutional networks. *CoRR*, abs/1902.07153, 2019.

[18] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. *CoRR*, abs/1701.02426, 2017.

[19] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.

[20] K. Yang, O. Russakovsky, and J. Deng. Spatialsense: An adversarially crowdsourced benchmark for spatial relation recognition, 2019.

[21] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. *CoRR*, abs/1711.06640, 2017.

[22] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.