Visual Relationships as Functions: Enabling Few-Shot Scene Graph Prediction

Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, Li Fei-Fei
Stanford University
{apoorvad, anarc, ranjaykrishna, msb, feifeili}@cs.stanford.edu

Abstract

Scene graph prediction — classifying the set of objects and predicates in a visual scene — requires substantial training data. The long-tailed distribution of relationships can be an obstacle for such approaches, however, as they can only be trained on the small set of predicates that carry sufficient labels. We introduce the first scene graph prediction model that supports few-shot learning of predicates, enabling scene graph approaches to generalize to a set of new predicates. First, we introduce a new model of predicates as functions that operate on object features or image locations. Next, we define a scene graph model where these functions are trained as message passing protocols within a new graph convolution framework. We train the framework with a frequently occurring set of predicates and show that our approach outperforms those that use the same amount of supervision by 1.78 *at recall@*50 *and performs on par with other scene graph models. Next, we extract object representations generated by the trained predicate functions to train few-shot predicate classifiers on rare predicates with as few as* 1 *labeled example. When compared to strong baselines like transfer learning from existing state-of-the-art representations, we show improved* 5*-shot performance by* 4.16 *recall@*1*. Finally, we show that our predicate functions generate interpretable visualizations, enabling the first interpretable scene graph model.*

1. Introduction

Scene graph prediction takes as input an image of a visual scene, and returns as output a set of relationships denoted as <subject - predicate - object>, such as <woman - drinking - coffee> and <coffee - on - table>. The goal is for these models to classify a large number of relationships for each image. However, due to the complexity of the task and uneven distribution of training relationship instances in the world and in training data, existing scene graph models are only performant with the most popular relationships (predicates). These existing models can be broadly divided into two approaches. The first approach detects the objects and then recognizes their pairwise relationships [8, 38, 39, 57]. The second approach jointly infers the objects and their relationships [33, 35, 55] based on object proposals. Both approaches treat relationship prediction as a multiclass predicate classification problem, given two object features. Such a formulation produces reasonable results as objects are a good indicator of relationships [58]. However, since the resulting object representations are utilized for both object as well as predicate classification, they confound the information required for both tasks. The representations, are therefore, not generalizable and can not be used to train the vast majority of less-frequently occurring predicates.

We present a new scene graph model that formulates predicates as functions, resulting in a scene graph model who's object representations can be used for few-shot predicate prediction. Instead of using the object representations to predict predicates, we instead treat predicates as two individual functions: a forward function that transforms the subject representation into the object, and an inverse function that transforms the object representation back into the subject. We further introduce a new graph convolution framework that uses these functions as localized message passing protocols between object nodes [26]. To further ensure that the object representations are disentangled from encoding specific information about a predicate, we divide each forward and inverse function into two components: a spatial component that transforms attention over the image space [29] and a semantic component that operates over the object features [59]. Within each graph convolution step, each pair of object representations score the functions by checking which of them agree with the difference between their representations. These scores are then used to weight the transformations performed by the functions and used to update the object representations. After multiple iterations, the object representations are classified into object categories and the function weights that remain

above a threshold result in a detected relationship.

By treating predicates as functions between object representations, our model is able to learn a meaningful embedding space that can be used for transfer learning of new few-shot predicate categories. For example, the forward function for `riding` learns to move the spatial attention to look below the subject to find the object and to move to a semantic location where rideable objects like `car`, `skateboard`, and `bike` can be found. We use the object representations generated by these functions to train few-shot predicate classifiers such as `driving` with as few as 1 labeled example.

Through our experiments on Visual Genome [30], a dataset containing visual relationship data, we show that the object representations generated by the predicate functions result in meaningful features that can be used to enable few-shot scene graph prediction, exceeding existing transfer learning approaches by 4.16 at recall@1 with 5 labelled examples. We further justify our design decisions by demonstrating that our scene graph model performs on par with existing state-of-the-art models and even outperforms models that also do not utilize external knowledge bases [18], linguistic priors [39, 58] or rely on complicated pre- and post-processing heuristics [58, 6]. We run ablations where we remove the semantic or spatial components of our functions and demonstrate that both components lead to increased performance but the semantic component is responsible for most of the performance. Finally, since our predicates are transformation functions, we can visualize them individually, enabling the first interpretable scene graph model.

2. Related work

Scene graphs were introduced as a formal representation for visual information [25, 30] in a form widely used in knowledge bases [19, 7, 61]. Each scene graph encodes objects as nodes connected together by pairwise relationships as edges. Scene graphs have led to many state of the art models in image captioning [1], image retrieval [25, 48], visual question answering [24], relationship modeling [29], and image generation [23]. Given its versatile utility, the task of scene graph prediction has resulted in a series of publications [30, 8, 37, 33, 35, 41, 55, 58, 56, 22] that have explored reinforcement learning [37], structured prediction [28, 9, 51], utilizing object attributes [11, 43], sequential prediction [41], and graph-based [55, 34, 56] approaches. However, all of these approaches have classified predicates using object features, confounding the object features with predicate information that prevents their utility when used to train new few-shot predicate categories.

Predicates and relationships. The strategy of decomposing relationships into their corresponding objects and predicates has been recognized in other works [34, 56] but we generalize existing methods by treating predicates as functions, implemented as general neural network modules. Recent work on referring relationships showed that predicates can be learned as spatial transformations in visual attention [29]. We extend this idea to formulate predicates as message passing semantic and spatial functions in a graph convolution framework. This framework generalizes existing work [34, 56] where relationships are usually treated as latent representations instead of functions. It also generalizes papers that have restricted these functions to linear transformations [5, 59].

Graph convolutions. Modeling graphical data has historically been challenging, especially when dealing with large amounts of data [53, 4, 60]. Traditional methods have relied on Laplacian regularization through label propagation [60], manifold regularization [4], or learning embeddings [53]. Recently, operators on local neighborhoods of nodes have become popular with their ability to scale to larger amounts of data and parallelizable computation [17, 44]. Inspired by these Laplacian-based, local operations, graph convolutions [26] have become the de facto choice when dealing with graphical data [26, 46, 36, 21, 10, 42]. Graph convolutions have recently been combined with RCNN [16] to perform scene graph detection [56, 23]. Unlike most graph convolution methods, which assume a known graph structure, our framework doesn't make any prior assumptions to limit the types of relationships between any two object nodes, i.e. we don't use relationship proposals to limit the possible edges. Instead, we learn to score the predicate functions between the nodes, strengthening the correct relationships and weakening the incorrect ones over multiple iterations.

Few-shot prediction. While graph-based learning typically requires large amounts of training data, we extend work in few-shot prediction, to show how the object representations learned using predicate functions can be further used to transfer to rare predicates. The few-shot literature is broadly divided into two main frameworks. The first strategy learns a classifier for a set of frequent categories and then uses them to learn the few-shot categories [27, 52, 50, 14]. The second strategy learns invariances or decompositions that enable few-shot classification [12, 13, 32, 49, 40, 6]. Our framework more closely resembles the first framework because we use the object representations learned using the frequent predicates to identify few-shot relationships with rare predicates.

Modular neural networks have been successful in numerous machine learning applications [3, 31, 54, 2, 24]. Typically, their utility has focused on the ability to train individual components and then jointly fine-tune them. Our paper focuses on a complementary ability of such networks: our functions are trained together and then used to learn additional predicates without retraining the entire model.

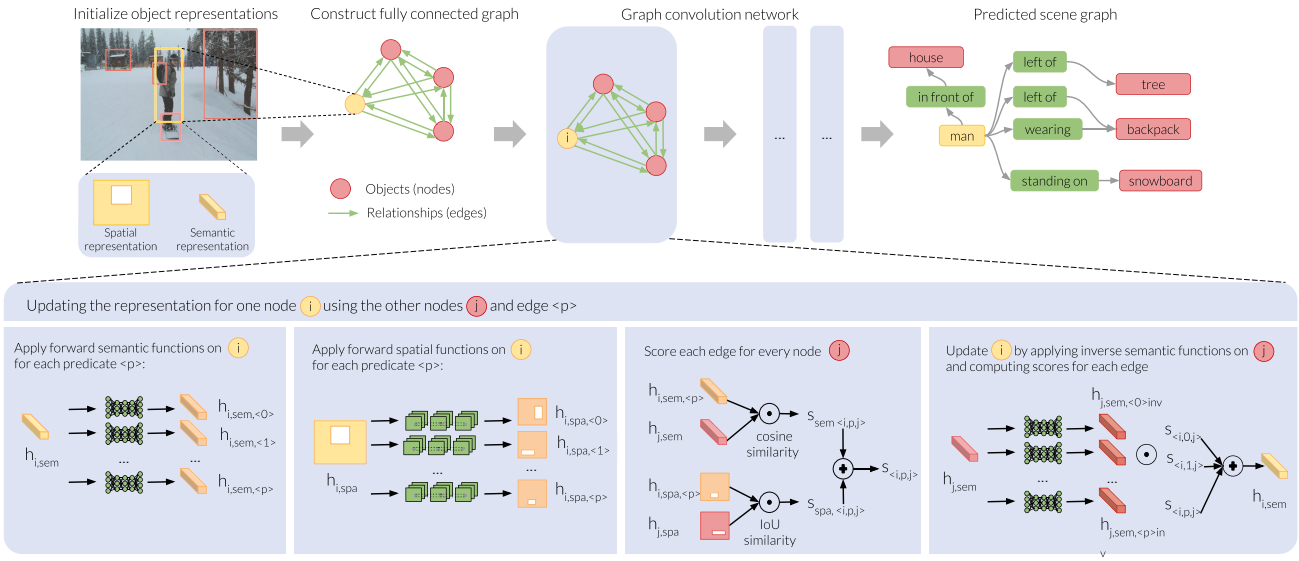


Figure 1. We introduce a scene graph approach that formulates predicates as learned functions, which result in an embedding space for objects that is effective for few-shot. Our formulation treats predicates as learned semantic and spatial functions, which are trained within a graph convolution network. First, we extract bounding box proposals from an input image and represent objects as semantic features and spatial attentions. Next, we construct a fully connected graph where object representations form the nodes and the predicate functions act as edges. Here we show how one node, the `person`'s representation is updated within one graph convolution step.

3. Graph convolution framework with predicate functions

In this section, we describe our graph convolution framework (Figure 1) and the predicate functions.

Problem formulation. Our goal is to learn effective predicate functions whose transformations result in effective object embeddings. We will use these functions for the task of scene graph generation in a graph convolution framework. Formally, the input to our model is an image I from which we extract a set of bounding box proposals $B = \{b_1, b_2, \dots b_n\}$ using a region proposal network [45]. From these bounding boxes, we extract initial object features $H^0 = \{h_1^0, h_2^0, \dots h_n^0\}$. These boxes and features are sent to our graph convolution framework. The final output of our model is a scene graph denoted as $G = \{\mathcal{V}, \mathcal{E}, \mathcal{P}\}$ with nodes (objects) $v_i \in \mathcal{V}$, and labeled edges (relationships) $e_{ijp} =< v_i, p, v_j > \in \mathcal{E}$, where $p \in \mathcal{P}$ is one of $|\mathcal{P}|$ predicate categories.

Traditional graph convolutional network. Our model is primarily motivated as an extension to graph convolutional networks that operate on local graph neighborhoods [10, 47, 26]. These methods can be understood as simple message passing frameworks [15]:

$$m_i^{t+1} = \sum_{j \in N(i)} M(h_i^t, h_j^t, e_{ij}), \qquad h_i^{t+1} = U(h_i^t, m_i^{t+1}) \tag{1}$$

where h_i^t is a hidden representation of node v_i in the t^{th} iteration, M and U are respectively aggregation and vertex update functions that accumulate information from the other nodes. $N(i)$ is the set of neighbors of i in the graph.

Our graph convolutional network. Similar to previous work [47] which used multiple edge categories, we expand the above formulation to support multiple edge types, i.e. given two nodes v_i and v_j, an edge exists from v_i to v_j for all $|\mathcal{P}|$ predicate categories. Unlike previous work where edges are an input [47], we initialize a fully connected graph, i.e. all objects are connected to all other objects by all predicate edges. If after the graph messages are passed, predicate p is scored above a hyperparameter threshold, then that relationship $< v_i, p, v_j >$ is part of the generated scene graph. The updated equations are then,

$$m_i^{t+1} = \sum_{p \in \mathcal{P}} \sum_{j \neq i} M_p(h_i^t, h_j^t, e_{ijp}), \tag{2}$$

$$h_i^{t+1} = U(h_i^t, m_i^{t+1}) = \sigma(W_0 h_i^t + m_i^{t+1}) \tag{3}$$

where $M_p(\cdot)$ are learned message functions between two nodes for the predicate p, which we will detail later in this section. Note that this formula is a generalized version of the exact representation used in the previous work [47], where $M_p(h_i^t, h_j^t, e_{ijp}) = \frac{1}{c_{i,p}} W_p h_j^t$ if $(v_i, p, v_j) \in \mathcal{E}$ and 0 otherwise, and σ is the sigmoid activation. Here, $c_{i,p}$ is a normalizing constant for the edge (i, j) as defined in previous work [47].

Node hidden representations. With the overall update step for each node defined, we now explain the hidden object

representation h_i^t. Traditionally, object nodes in graph models are defined as being a D-dimensional representation of the node $h_i \in \mathcal{R}^D$ [10, 47, 26]. However, in our case, we want these hidden representations to encode both the semantic information for each object proposal as well as its spatial location in the image. These two components will be separately utilized by the semantic and spatial predicate functions. Instead of asking our model to learn to represent both of these pieces of information, we built invariances into our representation such that it knows to encode them both explicitly. Specifically, we define each hidden representation as a tuple of two entries: $h_i^t = (h_{i,sem}^t, h_{i,spa}^t)$ — a semantic object feature $h_{i,sem}^t \in \mathcal{R}^D$ and a spatial attention map over the image $h_{i,spa} \in \mathcal{R}^{L \times L}$. In practice, we extract $h_{i,sem}^0$ from the penultimate layer in ResNet-50 [20] and set $h_{i,spa}$ as a $L \times L$ mask with 1 for the pixels within the object proposal and 0 outside.

With the semantic and spatial separation, we can rewrite equation 3:

$$m_i^{t+1} = (m_{i,sem}^{t+1}, m_{i,spa}),$$
$$m_{i,sem}^{t+1} = \sum_{p \in \mathcal{P}} \sum_{j \neq i} M_{sem}(h_{i,sem}^t, h_{j,sem}^t, e_{ijp}) \quad (4)$$

Note that $m_{i,spa}$ does not get updated because we fix the object masks for each object.

Predicate functions. To define $M_{sem}(\cdot)$, we introduce the semantic ($f_{sem,p}$) and spatial ($f_{spa,p}$) predicate functions for predicate p. Semantic functions are multi-layer perceptrons (MLP) while spatial functions are convolution layers, each with 6 layers and ReLU activations. Previous work on multi-graph convolutions [47] assumed that they had a priori information about the structure of the graph, i.e. which edges exist between any two nodes. In our case, we are attempting to perform both node classification as well as edge prediction simultaneously. Without knowing which edges actually exist in the graph, we would be adding a lot of noise if we allowed every predicate to equally influence another node. To circumvent this issue, we first calculate a score for each predicate p:

$$s_p(h_i^t, h_j^t) = \alpha s_{p,sem}(h_{i,sem}^t, h_{j,sem}^t) + (1-\alpha) s_{p,spa}(h_{i,spa}, h_{j,spa}), \quad (5)$$

$$s_{p,sem}(h_{i,sem}^t, h_{j,sem}^t) = \cos\left[f_{sem,p}(h_{i,sem}^t), h_{j,sem}^t\right], \quad (6)$$

$$s_{p,spa}(h_{i,spa}, h_{j,spa}) = \text{IoU}\left[f_{spa,p}(h_{i,spa}), h_{j,spa}\right], \quad (7)$$

where $\alpha \in [0, 1]$ is a hyperparameter, $\cos(\cdot)$ is the cosine distance function, and $\text{IoU}(\cdot)$ is the differentiable intersection over union function that measures the similarity between two soft heatmaps. This gives us a score for how likely the node v_i believes that the edge $< v_i, p, v_j >$ exists. Similar to recent work [29], $f_{spa,p}(\cdot)$ shifts the spatial attention from $h_{i,spa}$ to where it thinks node v_j should be. It encodes the spatial properties of the predicate we are learning and ignores the object features. To complement the spatial predicate function, we use $f_{sem,p}(\cdot)$ to transform $h_{i,sem}^t$. This shifted representation is what the model expects to be similar to $h_{j,sem}^t$. By using both the spatial and semantic score in our update of h_i, the two representations interact with one another. So, even though these components are separate, they create a cohesive score for each predicate. This score is used to weight how much node v_j will influence node v_i through a predicate p in the update in equation 3. We can now define:

$$M_{sem}(h_{i,sem}^t, h_{j,sem}^t, e_{ijp}) = s_p^l(h_i^t, h_j^t) f_{sem,p^{-1}}^{-1}(h_{j,sem}^t) \quad (8)$$

$f_{p^{-1}}(\cdot)$ represents the backward predicate function from `object` back to the `subject`. For example, given the relationship <`person - riding - snowboard`>, our model not only learns how to transform `person` using the function `riding`, but also how to transform `snowboard` to `person` by using the inverse predicate `riding`$^{-1}$. Learning both the forward and backward functions per predicate allows us to pass messages in both directions even though our predicates are directed edges.

Hidden representation update. We now define $U_{sem}(\cdot)$ that accumulate the messages passed by the semantic predicate functions to update the semantic object representation:

$$U_{sem}(h_{i,sem}^t, m_{i,sem}^{t+1}) = W_0 h_{i,sem}^t + \frac{1}{|\mathcal{P}|(|\mathcal{V}|-1)} m_{i,sem}^{t+1} \quad (9)$$

$$h_i^{t+1} = (U_{sem}(h_{i,sem}^t, m_{i,sem}^{t+1}), h_{i,spa}) \quad (10)$$

where W_0 is learned weight. The spatial representation does not get updated because the spatial location of an object does not move.

Scene graph output. Finally, we predict the categories of each node using $v_i = g(h_i)$, where g is an MLP that generates a probability distribution over all the possible object categories. Each possible relationship e_{ijp} is output as a relationship only if $s_p^T(h_i^T, h_j^T) * s_{p^{-1}}^{-T}(h_j^T, h_i^T) > \tau$ where T the total number of iterations in the model and τ a threshold hyperparameter.

4. Few-shot predicate framework

With our semantic ($f_{sem,p}$) and spatial ($f_{spa,p}$) predicate functions trained for the frequent predicates $p \in \mathcal{P}$, we now utilize these functions to create object representations to train few-shot predicates. We design few-shot predicate classifiers to be MLPs with 2 layers with ReLU activations

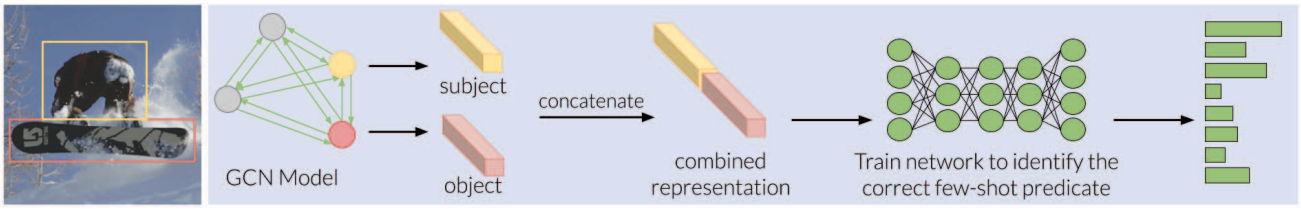


Figure 2. Overview of our few-shot training framework. We use the learned predicate function from the graph convolution framework to generate embeddings and attention masks for the object representations. These representations are used to train few-shot predicate classifiers.

between layers. We assume that rare predicates are $p' \in \mathcal{P}'$ and only have k examples each.

The intuition behind our k-shot training scheme lies in the modularity of predicates and their shared semantic and spatial components. By decomposing the predicate representations from the object in the graph convolutions, we create an representation space that supports predicate transformations. We will show in our experiments that our embeddings space places semantically similar objects that participate in similar relationships together. Now, when training with few examples of rare predicates, such as `driving`, we can rely on the semantic embeddings for objects that were clustered by `riding`.

We pass all k labelled examples of a predicate pair of objects $< v_i, p', v_j >$ through the learned predicate functions and extract the hidden representations $(h_{i,sem}, h_{i,spa})$ and $(h_{j,sem}, h_{j,spa})$ from the final graph convolution layer. We concatenate these transformations along the channel dimension and feed them as an input to the few-shot classifiers. We train the k-shot classifiers by minimizing the cross-entropy loss against the k labelled examples amongst $|\mathcal{P}'|$ rare categories.

5. Experiments

We begin our evaluation by first describing the dataset, evaluation metrics, and baselines. Our first experiment studies our graph convolution framework and compares our scene graph prediction performance against existing state-of-the-art methods. Our second experiment tests the utility of our approach on our main objective of enabling few-shot scene graph prediction. Finally, our third experiment showcases interpretable visualizations by visualizing the predicate transformations.

Dataset: We use the Visual Genome [30] dataset for training, validation and testing. To benchmark against existing scene graph approaches, we use the commonly used subset of 150 object and 50 predicate categories [55, 58, 56]. We use publicly available pre-processed splits of train and test data, and sample a validation set from the training set [58]. The training, validation, and test sets contain $36,662$ and $2,794$ and $15,983$ images, respectively.

Evaluation metrics: For scene graph prediction, we use three evaluation tasks, all of which are evaluated at recall@50 and recall@100. (1) `PredCls` predicts predicate categories, given ground truth bounding boxes and object classes, (2) `SGCls` predicts predicate and object categories given ground truth bounding boxes, and (3) `SGGen` detects object locations, categories and predicate categories. Metrics based on recall require ranking predictions. For `PredCls` this means a simple ranking of predicted predicates by score. For `SGCls` this means ranking subject-predicate-object tuples by a product of subject, object, and predicate scores. For `SGGen` this means a similar product as SGCls, but tuples without correct subject or object localizations are not counted as correct. We refer readers to previous work that defined these metrics for further reading [39].

For few-shot prediction, we report recall@1 and recall@50 on the task of `PredCls`. We vary the number of labeled examples available for training few-shot predicate classifiers from $k \in [1, 2, 3, 4, 5]$. We also report recall@1 in addition to the traditional recall@50 because each image only has a few instances of rare predicates in the test set.

Baselines: We classify existing methods into two categories. The first category includes other scene graph approaches that, like our approach, only utilizes Visual Genome's data as supervision. This includes Iterative Message Passing (`IMP`) [55], Multi-level scene Description Network (`MSDN`) [35], `ViP-CNN` [33], `MotifNet-freq` [58]. The second category includes models such as `Factorizable Net` [34], `KB-GAN` [18] and `MotifNet` [58], which use linguistic priors in the form of word vectors or external information from knowledge bases while `MotifNet` also deploys a custom trained object detector, class-conditioned non-maximum suppression, and heuristically removes all object pairs that do not overlap. While not comparable, we report their numbers for clarity.

5.1. Scene graph prediction

We report scene graph prediction numbers on Visual Genome [30] in Table 1. This experiment is meant to serve as a benchmark against existing scene graph approaches. We outperform existing models that only use

Table 1. We perform on par with all existing state-of-the-art scene graph approaches and even outperform other methods that only utilize Visual Genome's data as supervision. We also report ablations by separating the contribution of the semantic and the spatial components.

		SG GEN		SG CLS		PRED CLS	
	Metric	recall@50	recall@100	recall@50	recall@100	recall@50	recall@100
vision only	IMP [55]	06.40	08.00	20.60	22.40	40.80	45.20
	MSDN [35]	07.00	09.10	27.60	29.90	53.20	57.90
	MotifNet-freq [58]	06.90	09.10	23.80	27.20	41.80	48.80
	Graph R-CNN [56]	11.40	**13.70**	**29.60**	**31.60**	54.20	**59.10**
	Our full model	**13.18**	13.45	23.71	24.66	**56.65**	57.21
external	Factorizable Net [34]	13.06	16.47	-	-	-	-
	KB-GAN [18]	13.65	17.57	-	-	-	-
	MotifNet [58]	**27.20**	**30.30**	35.80	36.50	**65.20**	**67.10**
	PI-SG [22]	-	-	**36.50**	**38.80**	65.10	66.90
Ablation	Our spatial only	02.05	02.32	03.92	04.54	04.19	04.50
	Our semantic only	12.92	12.39	23.35	24.00	56.02	56.67
	Our full model	**13.18**	**13.45**	**23.71**	**24.66**	**56.65**	**57.21**

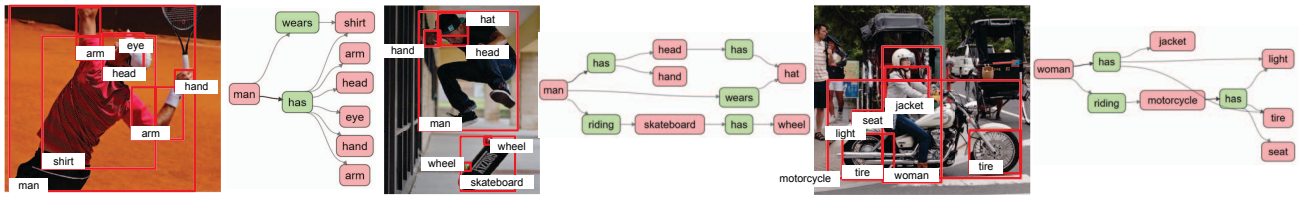


Figure 3. Example scene graphs predicted by our graph convolution fully-trained model.

Visual Genome supervision for `SGGen` and `PredCls` by 1.78 and 1.82 recall@50, respectfully. But we fall short on recall@100. As we move from recall@50 to recall@100, models are evaluated on their top 100 predictions instead of their top 50. Unlike other models that perform a multi-class classifiction of predicates for every object pair, we assign binary scores to each possible predicate between an object pair individaully. Therefore, we can report that no relationship exists between a pair of objects. While this design decision allows us to separate learning predicates transformations and object representations, it penalizes our model for not guessing relationships for every single object pair, thereby, reducing our recall@100 scores. We also notice that since our model doesn't utilize the object categories to make relationship predictions, it performs worse for the task of `SGCls`, which presents models with ground truth object locations.

We also report ablations of our model trained using only the semantic or spatial functions. We observe that different ablations of the model perform better on certain types of predicates. The spatial model performs well on predicates that have a clear spatial or location-based aspect, such as `above` and `under`. The semantic model performs better on non-spatial predicates such as `has` and `holding`. Our full model outperforms the individual semantic-only and spatial-only models as predicates can utilize both components. We visualize some scene graphs generated by our network in Figure 3.

5.2. Few-shot prediction

Our second experiment studies how well we perform few-shot scene graph prediction with limited examples per predicate. Our approach requires two sets of predicates, a set of frequently occurring predicates and a second set of rare predicates with only k examples. we split the usual 50 predicates typically used in Visual Genome, and place the 25 most predicates with the most training examples into the first set and place the remaining 25 predicates into the second set. In our experiments, we train the predicate functions and the graph convolution framework using the predicates in the first set. Next, we use them to train k-shot classifiers for the rare predicates in the second set by utilizing the representations generated by the pretrained predicate functions. We iterate over $k \in [1, 2, 3, 4, 5]$.

For a rigorous comparison, we choose to compare our method against `MotifNet` [58], which outperforms all existing scene graph approaches and uses linguistic priors from word embeddings and heuristic post-processing to generate high-quality scene graphs. Specifically, we report two different training variants of MotifNet: `MotifNet-Baseline`, which is initialized with random weights and trained only using k labelled examples

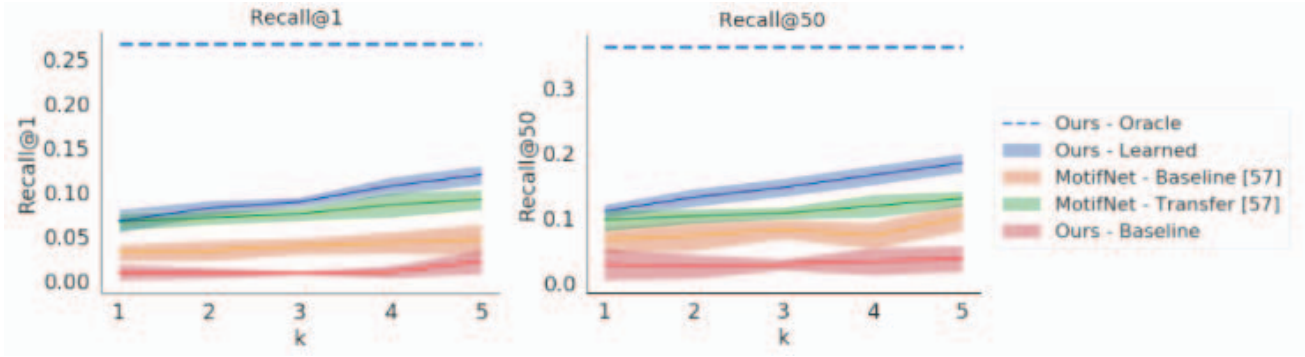


Figure 4. We show Recall@1 and Recall@50 results on k-shot predicates. We outperform strong baselines like transfer learning on MotifNet [58], which also relies on linguistic priors.

and `MotifNet-Transfer`, which is first trained on the frequent predicates and then finetuned on the k few-shot predicates. We also compare against `Ours-Baseline`, which trains our graph convolution framework on the k few-shot predicates and `Ours-Oracle`, which reports the upper bound performance when trained with all of Visual Genome.

Results in Figure 4 outline that our method performs better than all baseline comparisons for all values of k. We find that our learned classifiers are similar in performance to `MotifNet-Transfer` when $k = 1$. This is likely because `MotifNet-Transfer` also has access to additional information available from word embeddings. The improvements seen by our approach increase as k increases to $k = 5$, where we outperform the baselines by 3.26 recall@50. Eventually, as more labels becomes available, the Neural Motif model outperforms our model for values of $k \geq 10$.

5.3. Interpretable predicate transformation visualizations

Our final experiment showcases another utility of treating predicates as functions. Once trained, these functions can be individually visualized and qualitatively evaluated. Figure 5(left and middle) shows examples of transforming spatial attention from four instances of `person`, `horse`, `boy`, and `banana` in four images. We see that `above` and `standing on` moves attention below the person `looking` moves attention left towards the direction the `horse` is looking. `wearing` highlights the center of the `boy`. Figure 5(right) shows semantic transformations applied to the embedding representation space of objects. We see that `riding` transforms the embedding to a space that contains objects like `wave`, `skateboard`, `bike` and `horse`. Notice that unlike linguistic word embeddings, which are trained to place words found in similar contexts together, our embedding space represents the types of visual relationships that objects participate. We include more visualizations in our appendix.

6. Conclusion

We introduced the first scene graph prediction model that treats predicates as functions and generates object representations that can effectively enable few-shot learning. We treat predicates as neural network transformations between object representations. The functions disentangle the object representations from storing predicate information, and instead generates an embedding space with objects that embed similar relationships close together. Our representations outperform existing methods for few-shot predicate prediction, a valuable task since most predicates occur infrequently. Also, our graph convolution network, which trains the predicate functions, performs on par with existing scene graph prediction state-of-the-art models. Finally, the predicate functions result in interpretable visualizations, allowing us to visualize the spatial and semantic transformations learned for each predicate.

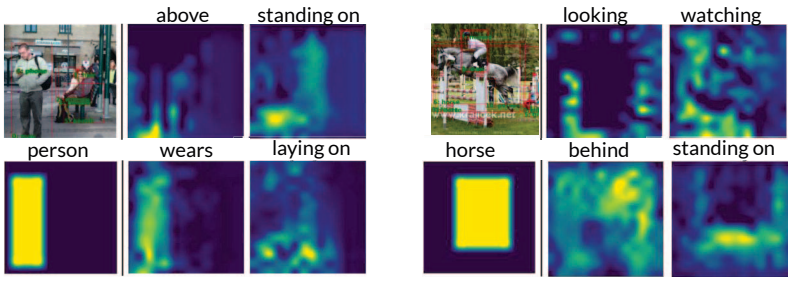
Acknowledgements We thank Iro Armeni, Suraj Nair, Vincent Chen, and Eric Li for their helpful comments. This work was partially funded by the Brown Institute of Media Innovation and by Toyota Research Institute (TRI) but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016. 2

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*, 2016. 2

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2016. 2



subject	predicate	closest objects
girl	riding	wave, skateboard, bike, horse
man	wears	shirt, jacket, hat, cap, helmet
person	has	hair, head, face, arm, ear
dog	laying on	bed, beach, bench, desk, table
child	walking on	street, sidewalk, snow, beach
boy	sitting on	bench, bed, desk, chair, toilet
umbrella	covering	kid, people, skier, person, guy

Figure 5. (left, middle) Spatial transformations learned by our model applied to object masks in images. (right) Semantic transformations applied to the average object category embedding; we show the nearest neighboring object categories to the transformed subject.

[4] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006. 2

[5] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013. 2

[6] Vincent S Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. *arXiv preprint arXiv:1904.11622*, 2019. 2

[7] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 423. Association for Computational Linguistics, 2004. 2

[8] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3298–3308. IEEE, 2017. 1, 2

[9] Chaitanya Desai, Deva Ramanan, and Charless C Fowlkes. Discriminative models for multi-class object layout. *International journal of computer vision*, 95(1):1–12, 2011. 2

[10] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015. 2, 3, 4

[11] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009. 2

[12] Li Fe-Fei et al. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1134–1141. IEEE, 2003. 2

[13] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 2

[14] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017. 2

[15] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017. 3

[16] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[17] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016. 2

[18] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. *arXiv preprint arXiv:1904.00560*, 2019. 2, 5, 6

[19] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics, 2005. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 4

[21] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015. 2

[22] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems*, pages 7211–7221, 2018. 2, 6

[23] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. *arXiv preprint arXiv:1804.01622*, 2018. 2

[24] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. *arXiv preprint arXiv:1705.03633*, 2017. 2

[25] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015. 2

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1, 2, 3, 4

[27] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015. 2

[28] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 2

[29] Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. Referring relationships. In *Computer Vision and Pattern Recognition*, 2018. 1, 2, 4

[30] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 2, 5

[31] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016. 2

[32] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33, 2011. 2

[33] Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao'Ou Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 7244–7253. IEEE, 2017. 1, 2, 5

[34] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *European Conference on Computer Vision*, pages 346–363. Springer, 2018. 2, 5, 6

[35] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1261–1270, 2017. 1, 2, 5, 6

[36] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015. 2

[37] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4408–4417. IEEE, 2017. 2

[38] Wentong Liao, Lin Shuai, Bodo Rosenhahn, and Michael Ying Yang. Natural language guided visual relationship detection. *arXiv preprint arXiv:1711.06032*, 2017. 1

[39] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, pages 852–869. Springer, 2016. 1, 2, 5

[40] Akshay Mehrotra and Ambedkar Dukkipati. Generative adversarial residual pairwise networks for one shot learning. *arXiv preprint arXiv:1703.08033*, 2017. 2

[41] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Advances in Neural Information Processing Systems*, pages 2168–2177, 2017. 2

[42] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016. 2

[43] Devi Parikh and Kristen Grauman. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 503–510. IEEE, 2011. 2

[44] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014. 2

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[46] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 2

[47] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1703.06103*, 2017. 3, 4

[48] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. 2

[49] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2

[50] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *Advances in Neural Information Processing Systems*, pages 2255–2265, 2017. 2

[51] Zhuowen Tu and Xiang Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1744–1757, 2010. 2

[52] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 2

[53] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. 2

[54] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406, 2016. 2

[55] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017. 1, 2, 5, 6

[56] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. *arXiv preprint arXiv:1808.00191*, 2018. 2, 5, 6

[57] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. *arXiv preprint arXiv:1707.09423*, 2017. 1

[58] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. *arXiv preprint arXiv:1711.06640*, 2017. 1, 2, 5, 6, 7

[59] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, volume 1, page 5, 2017. 1, 2

[60] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004. 2

[61] Guodong Zhou, Min Zhang, DongHong Ji, and Qiaoming Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007. 2