This WACV 2023 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# GEMS: Scene Expansion using Generative Models of Graphs

Rishi Agarwal[1*] Tirupati Saketh Chandra[2*] Vaidehi Patil[3*]
Aniruddha Mahapatra[4*] Kuldeep Kulkarni[5] Vishwa Vinay[5]

Stanford University, USA[1] IIT Bombay, India[2] UNC Chapel Hill, USA[3]
Carnegie Mellon University, USA[4] Adobe Research, India[5]

rishia@stanford.edu tsaketh@iitb.ac.in vaidehi@cs.unc.edu
amahapat@andrew.cmu.edu {kulkulka, vvinay}@adobe.com

## Abstract

*Applications based on image retrieval require editing and associating in intermediate spaces that are representative of the high-level concepts like objects and their relationships rather than dense, pixel-level representations like RGB images or semantic-label maps. We focus on one such representation, scene graphs, and propose a novel scene expansion task where we enrich an input seed graph by adding new nodes (objects) and the corresponding relationships. To this end, we formulate scene graph expansion as a sequential prediction task involving multiple iterations of first predicting a new node and then predicting the set of relationships between the newly predicted node and previously chosen nodes in the graph. We propose and evaluate a sequencing strategy that retains the clustering patterns amongst nodes. In addition, we leverage external knowledge to train our graph generation model, enabling greater generalization of node predictions. Due to the inefficiency of existing maximum mean discrepancy (MMD) based metrics standard for graph generation problems, we design novel metrics that comprehensively evaluate different aspects of node and relation predictions. We conduct extensive experiments on Visual Genome and VRD datasets to evaluate the expanded scene graphs using the standard MMD based metrics, as well as our proposed metrics. We observe that the graphs generated by our method, GEMS, better represent the real distribution of the scene graphs compared with baseline methods like GraphRNN.*

## 1. Introduction

Creative photographers are gifted with the ability to imagine a set of concepts - objects and inter-object relationships - to capture in a photograph. However, they spend prohibitively expensive amount of time arriving at the kinds of scenes that contain a seed set of concepts they desire to be present in the photograph. Hence, it is desirable to empower them with recommendations of a wide variety of diverse and rich plausible scenes that contain these seed concepts. We wish to devise algorithms that can be leveraged to provide the user with effective recommendations of scenes that subsume the seed concepts while ensuring they are richer than the seeds represent themselves.

To this end, we express the seed concepts in the form of a scene graph [41, 19] and cast the task of producing more complete scenes as that of generating plausible novel scene graphs that contain the input seed graph. Specifically, we propose a novel *scene expansion* problem - given a seed graph, can we enhance it by the addition of objects so that the new graph corresponds to an enriched scene while satisfying the following requirements: (a) the proposed additions respects object co-occurrence patterns observed in the training set; (b) the enhanced scene graph is novel with respect to the existing collection of graphs; and (c) it is possible to generate diverse graph expansions for the same seed.

The space of generative models for unconditional generation of molecular graphs has received much attention [12, 33, 17, 32, 1] recently. Specifically, the auto-regressive models [10, 42] that have been shown to work well for molecular graph generation can potentially be repurposed for our task of scene graph expansion. However, the complexity of the graphs considered in these works tend to be several orders of magnitude smaller than scene graphs, in terms of the number of distinct types of nodes and relationship-edges. Moreover, these methods implicitly require the graphs to be connected, which is not necessarily a characteristic of the scene graphs that we deal with. In addition, scene graphs tend to be more diverse than molecular graphs. For these reasons, the above mentioned auto-regressive models that are proposed for graph generation cannot be used as-is for the scene graph expansion problem that we tackle.

[*] The first four authors contributed equally to this work and was done while authors were at Adobe Research.

Motivated by this, we design a novel auto-regressive graph expansion model, **GEMS** - Graph Expansion Model for Scenes, drawing inspiration from [42] that can generate graphs of various lengths (unlike [8, 33, 2]). We first flatten the scene graphs into sequences where each node in the sequence is connected by relationships with previous nodes in the sequence. Our proposed sequencing method tries to ensure that groups of objects connected in the scene graph occur close by in the resulting sequence, this ensures that the model learns an approximate notion of motifs [44]. Graph expansion then becomes a sequential prediction problem, where node generation precedes edge generation. Due to the imbalance in edge-types in scene graphs we use a class-rebalancing loss that helps produce higher quality graph expansions by avoiding predictions of degenerate edge-labels. Further, we incorporate external knowledge derived from the language domain for better generalization of node predictions to encourage generation of a diverse set of related node predictions. Our proposed method is then thoroughly evaluated using a set of standard metrics for the graph synthesis task (as in [10]). Where existing metrics do not provide a holistic view of scene graph expansion quality, we propose novel metrics specifically tailored for the task considered here. We summarize our key contributions below.

- We propose a novel scene expansion task that deals with enhancing a given seed graph by the addition of new objects and relationships so that the enhanced graph corresponds to an enriched scene.
- We design an auto-regressive model, GEMS, for conditional generation of scene graphs that generates nodes and edges hierarchically in a dependent manner.
- We propose a novel graph sequencing method (Cluster-Aware BFS) aimed at capturing object co-occurrences, and we subsequently illustrate the benefits of this method.
- To circumvent the drawbacks of traditional evaluation methodologies, we propose additional metrics to evaluate the generated scene graphs to capture the *coherence* of predicted edges and nodes.

Through extensive experiments on Visual Genome [21] and VRD [25] datasets, we show that our model outperforms the GraphRNN based baseline models comprehensively along most of the metrics, and is competitive with [9] that introduces complementary ideas to ours.

## 2. Related Work

The relevant literature is introduced from the following two aspects: (1) Scene Graph Extraction (2) Generative Models of Graphs.

### 2.1. Scene Graph Extraction

The standard task known as scene graph *generation* [34, 26] involves constructing a graph with nodes as objects and their attributes, with edges being relationships between them. This task involves producing a graph from image input, and is referred to as "extraction" in the rest of this paper. Broadly, scene graph extraction methods fall into two categories. First, referred to in this paper as **Internal Knowledge**, refers to the works [22, 5, 45, 20] where the features that are leveraged to produce the graph originate from only the image of interest. At a high level, scene graph extraction operates by detecting objects and their regions within the image followed by a relationship identification model sub-component that labels the connections between the objects. Subsequent works have attempted to address the issue arising from biased nature of the training data caused by the long-tail of infrequently occurring relationships [7, 35]. The second line of work in scene graph extraction leverages **External Knowledge** in the form of word embeddings [25] for the object and relationship class names as a prior from the language domain. Scene graph extraction methods that combine information internal to the image with external knowledge have shown increased levels of accuracy [43, 13].

Our work differs from these two lines of works in that we do not have access to the input image to extract the visual features. I.e., our input is in the form of a scene graph, with no access to the image modality. Hence, similar to the second line of work, we leverage external knowledge – as representations for nodes and edges, as well as a regularizer for the nodes and edge prediction tasks. We invoke state-of-the-art generative models of graphs, described next, to expand the given seed graph.

### 2.2. Generative Models of Graphs

Graphs are a powerful and natural representation of the data in many application settings. And, as with many other domains, generative models trained over a set of observed graphs have received much recent attention [14]. Most existing work considers molecular graphs, where sampling from a trained model allows the generation of novel molecules, the core objective of drug design. Variational Auto-Encoders (VAEs) are a popular method within this class of models [12, 33, 17, 32], and so are Generative Adversarial Networks (GANs) [1]. In the current paper, we consider scene graphs derived from images, where sparsity [6] needs to be specifically addressed since most object-object pairs do not have a relationship between them. In addition, scene graphs tend to be diverse, a characteristic they share with graphs from few other domains [24, 42].

The closest to our work is SceneGraphGen [9] and VarScene [37] that both introduce several complementary ideas. We differentiate our work on three main aspects: (1)

We define a custom sequencing function that tries to ensure that groups of objects connected in the scene graph occur close by in the sequence. (2) Similar to recent work on scene graph extraction from images, we show how the use of external information about object-object similarities can guide the model toward scene graphs that are more coherent. (3) Our exhaustive experimental evaluation also proposes novel metrics for the scene graph expansion task.

# 3. Problem Description and Model

We are given a collection of observed scene graphs $\mathbb{G} = \{G\}$ where each $G$ corresponds to an image and is represented by $G = (V, E)$ - a set of vertices $V \subseteq \mathbb{V}$ and directed, labelled edges $E \subseteq \{(u, e, v) | u, v \in \mathbb{V}, u \neq v, e \in \mathbb{E}\}$ that connect pairs of objects in $V$. Here, $\mathbb{V}$ is the set of distinct objects found in the collection of scene graphs and $\mathbb{E}$ is the set of unique relationships. Our objective is to take a graph $G_s \notin \mathbb{G}$ and expand it into $\hat{G}_s$ such that $G_s$ is a subgraph of $\hat{G}_s$. Drawing inspiration from [23, 42] we convert the graph into a sequence and transform this problem of graph expansion into sequential prediction. That is, under a node ordering $\pi \in \Pi$, a graph $G$ is flattened into a sequence $\mathcal{S}(G) = \{(v_i, \mathcal{E}_i)\}_{i=1}^{n}$, where $v_i \in \mathbb{V}$ indexes the $i^{th}$ node in the sequence induced by $\pi$. And $\mathcal{E}_i = \{\mathcal{E}_{i,j}\}_{j<i}$ is a list containing edge information for node $i$ with every node $j$ before it in the sequence. Since scene graphs are directed, we take each $\mathcal{E}_{i,j} = (e_{i,j}, e_{j,i})$ to be a pair of relationships - one in each direction - with $e_{i,j}$ denoting the relationship from $v_i$ to $v_j$.

## 3.1. Cluster-Aware BFS

Critical to being able to train graph generation models is the role of the process that converts a graph $G$ into a sequence $\mathcal{S}(G)$. GraphRNN [42] uses a breadth-first strategy (BFS) while GraphGen [10] uses a depth-first traversal. Both these options require the input graph to be fully connected. The input scene graphs in our context often contain disconnected components, these correspond to natural scenes where part of the image may not have relationships with objects in other parts. We have observed that strategies that artificially convert the given scene graphs into fully-connected graphs (e.g. with the help of dummy nodes and edges) exacerbates the problems due to the skewed distributions of objects and relationships. Additionally, as we might intuitively expect, some sets of objects co-occur frequently across the dataset of observed graphs. In an attempt to encourage the model to better handle clusters of objects that occur together, we devise a method that ensures that objects in the same cluster are close by in the sequence. For any given scene graph, we first identify its maximal connected subgraphs. We obtain the BFS sequence for each subgraph with a randomly chosen starting node. The sequence for the scene graph is obtained by concatenating subgraph sequences in random order. Randomizing across subgraphs before concatenation is aimed at introducing robustness with respect to the input seed graph.

As previously defined, $\mathcal{S}(G)$ can be thought of as a matrix where row $i$ holds information about $v_i$ and its relationships with the objects occurring previously in the sequence (in rows before $i$). We use the shorthand $\mathcal{S}_i$ for all information about the $i^{th}$ node in the sequence for graph $G$, and $\mathcal{S}_{<i}$ for all nodes and edges occurring before it. Similarly, $\mathcal{E}_i$ contains information about edges incident on the $i^{th}$ node, and $\mathcal{E}_{i,<j}$ denotes the edges between $v_i$ and all nodes upto $j$, i.e. $\{v_k\}_{k<j}$. A likelihood can now be defined for the sequence:

$$P(\mathcal{S}(G)) = \prod_{i=1}^{n_G} P(v_i|\mathcal{S}_{<i}) \times P(\mathcal{E}_i|\mathcal{S}_{<i}, v_i) \quad (1)$$

$$P(\mathcal{E}_i|\mathcal{S}_{<i}, v_i) = \prod_{j<i} P(\mathcal{E}_{i,j}|\mathcal{S}_{<i}, \mathcal{E}_{i,<j}, v_i, v_j) \quad (2)$$

## 3.2. Hierarchical Node and Edge Prediction

The expansion of graph sequence occurs in steps, we first predict a new node $\hat{v}_i$ and then a set of relationships between $\hat{v}_i$ and previous nodes in the sequence. In the current paper, both $P(v_i|\mathcal{S}_{<i})$ and $P(\mathcal{E}_{i,j}|\mathcal{S}_{<i}, \mathcal{E}_{i,<j}, v_i, v_j)$ are modeled separately by recurrent neural networks, given by $f_{node}$ and $f_{edge}$ respectively, with the corresponding parameters shared across different steps. Prediction of $\hat{v}_i$ is defined as:

$$\hat{v}_i \sim f_{node}(\mathcal{S}_{i-1}, h_{node}(\mathcal{S}_{<i})) \quad (3)$$

That is, the prediction of the $i^{th}$ node in the sequence depends on $\mathcal{S}_{i-1}$ and the hidden state of $f_{node}$ from the previous step. Correspondingly, the prediction of the new edge pair $(\hat{e}_{i,j}, \hat{e}_{j,i})$ is given by:

$$\hat{e}_{i,j} \sim f_{edge}(v_i, v_j, h_{edge}(\mathcal{S}_{<i}, \mathcal{E}_{i,<j})) \quad (4)$$

$$\hat{e}_{j,i} \sim f_{edge}(v_j, v_i, h_{edge}(\mathcal{S}_{<i}, \mathcal{E}_{i,<j}, \hat{e}_{i,j})) \quad (5)$$

Note that in our formulation we first predict $e_{i,j}$ and then $e_{j,i}$. For edge prediction, we explicitly provide $v_i$ and $v_j$ as inputs into $f_{edge}$ because the existence of an edge between two nodes, as well as its label, is more dependent on the local context (the nodes) than the rest of the graph. The objects and relationships are sampled from multinomial distributions, making our model a Dependent Multinomial Sequence Model (rather than Bernoulli sequences as in [42]). Our training objective is a combination of the losses computed on node and edge predictions:

$$\mathcal{L}(G) = \sum_{i \in V} l_{node}(p_{v_i}, p_{\hat{v}_i}) + \sum_{j<i;(v_i,e,v_j)\in E} l_{edge}(p_{e_{i,j}}, p_{\hat{e}_{i,j}}) + \sum_{j<i;(v_j,e,v_i)\in E} l_{edge}(p_{e_{j,i}}, p_{\hat{e}_{j,i}}) \quad (6)$$

We define the node prediction loss as:

$$l_{node}(p_{v_i}, p_{\hat{v}_i}) = H(p_{v_i}, p_{\hat{v}_i}) \tag{7}$$

Where, $H$ is the cross-entropy loss between a 1-hot encoding for the node label of $v_i$ ($p_{v_i}$) and the corresponding predicted probability $p_{\hat{v}_i}$ for $\hat{v}_i$ . Similarly, the edge prediction is defined as,

$$l_{edge}(p_{e_{i,j}}, p_{\hat{e}_{i,j}}) = \frac{1-\beta}{1-\beta^{N_e}} H(p_{e_{i,j}}, p_{\hat{e}_{i,j}}) \tag{8}$$

Where, $p_{e_{i,j}}$ is the 1-hot encoding of the ground-truth edge type, $p_{\hat{e}_{i,j}}$ is the is the probability distribution of predicted edge type between $v_i$ and $v_j$, and $N_e$ is the number of instances of this edge across the dataset. This loss is a class balanced loss [4] designed to tackle highly skewed distribution across relationship classes [35], so as to produce a model that is less prone to predicting degenerate edges.

### 3.3. External Knowledge

Cross-entropy is a very strict loss, in the sense that near misses (predicting a node that is semantically similar but not the ground-truth node) are not considered different from obvious errors. To encourage generalization of predicted node labels, we add an additional loss term $H(p_{\hat{v}_i}, q_i)$, where,

$$q_i = \min_q KL(q, p_{\hat{v}_i}) - E_{v \sim q}[f(v, v_i)] \tag{9}$$

$f(v, v_i)$ is the similarity between $v_i$ and $\hat{v}_i$ as obtained from external knowledge, and KL is Kullback-Leibler Divergence. $q_i$ is a proxy label that is dependent on both - the model prediction ($\hat{v}_i$) and the ground-truth ($v_i$). Intuitively, picking a $q_i \propto p_{\hat{v}_i} exp(f(v, v_i))$ provides us with a node that is similar (but different) to the ground-truth as captured by the similarity function $f$. By employing alternate functions (e.g. cosine similarity between word embeddings), we reduce the penalisation on the model by effective use of side information [16, 43]. The node prediction loss thus becomes:

$$l_{node}(p_{v_i}, p_{\hat{v}_i}) = (1-\alpha) H(p_{v_i}, p_{\hat{v}_i}) + \alpha H(p_{v_i}, q_i) \tag{10}$$

where $\alpha$ is a hyperparameter.

### 3.4. Inference

We convert the input seed graph $G_s$ into a sequence $\mathcal{S}(G_s)$. Using GEMS, we extend the sequence by progressively adding nodes and edges. To add a new node, we compute the distribution over node labels using the network $f_{node}$ and sample from this multinomial distribution. To add a relationship between nodes $v_i$ and $v_j$, we pick the most probable edge label between the two nodes as predicted by $f_{edge}$. In this way, the seed graph $G_s$ is sequentially expanded to provide $\hat{G}_s$ (an enhanced scene).

**Algorithm 1** Extraction of Seed Graph

```
Input: Scene Graph $G$
Parameter: k ← Number of seed graphs
Output:
  $S$ ← set of maximal connected components in $G$
  $seedgraph$ = []
  for $g \in S$ do
    $PR$ ← empty dictionary
    for $n \in nodes(g)$ do
      $PR[n] = PageRank(n)$ in $g$
    end for
    $subG$ ← set of all subgraphs of $g$
    $pr$ ← empty list
    for i in range(len($subG$)) do
      $pr[i]$ = $\frac{1}{|nodes(subG[i])|} \sum_{n \in nodes(subG[i])} PR[n]$
    end for
    $X \sim Normalize(pr)$
    $seedgraph[g]$ ← $k$ samples from $X$
  end for
  return $seedgraph$
```

## 4. Experiments

In this section, we provide empirical validation for the method described earlier. We begin by outlining the datasets and experiment design used.

### 4.1. Datasets

Our experiments use two publicly available standard datasets that have ground truth scene graph information. For **Visual Genome**, we utilize the preprocessed version from [40], containing 150 object classes and 50 relation classes. The dataset contains human-annotated scene graphs on $108,077$ images. Each scene graph on average contains 11.09 objects and 5.01 relationships. We use $70\%$ of the images for training and validation, and the remaining $30\%$ for testing. Similarly, the **Visual Relationship Dataset(VRD)** dataset contains 100 object classes and 70 relation types. The dataset includes 5000 images, of which we use $80\%$ of the images for training and validation, and the remaining $20\%$ are retained for testing.

### 4.2. Implementation Details

The two RNNs, $f_{node}$ and $f_{edge}$, are implemented as 4 layers of GRU cells. We use teacher forcing [39] during training, where the ground-truth of observed sequences (nodes & edges) are used, but model predictions are used during inference. Model fitting utilizes Stochastic Gradient Descent with Adam Optimizer and minibatches of size 32, with the learning rate set to $0.001$. $\alpha$ and $\beta$ are set to $0.2$ and $0.9999$ respectively in all experiments. We use pre-trained GloVe embeddings [29] as inputs into the model for node and edge labels. For each new predicted node, GEMS predicts edges with $k$-previous nodes denoted

|  |  | Visual Genome |  |  |  | VRD |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  |  | **GraphRNN** | **GraphRNN*** | **SceneGraphGen** | **GEMS** | **GraphRNN** | **GraphRNN**** | **SceneGraphGen** | **GEMS** |
| MMD | Degree (x$10^2$) ↓ | 47.47 | 16.44 | **6.97** | **2.11** | 10.94 | **7.94** | **4.44** | 9.91 |
|  | Clustering (x$10^2$) ↓ | 18.63 | 4.05 | **0.26** | **0.86** | 16.08 | **1.21** | **2.58** | 3.96 |
|  | NSPDK* (x$10^3$) ↓ | 22.60 | 5.10 | **0.73** | **1.21** | 6.62 | 6.57 | **2.85** | **4.09** |
|  | Node Label (x$10^4$) ↓ | 5.44 | **5.26** | 5.92 | **5.19** | 30.70 | 27.01 | **24.82** | **25.11** |
|  | Edge Label (x$10^2$) ↓ | 22.38 | 6.19 | **0.83** | **1.13** | 1.06 | **0.39** | **0.60** | 1.97 |
| Node Metrics | Count Reference | 11.09 |  |  |  | 7.01 |  |  |  |
|  | Count Predicted | 29.53 | 13.84 | 7.89 | 10.17 | 7.24 | 7.59 | 6.41 | 7.81 |
|  | $(Obj)_K$ (x$10^2$) ↑ | 83.7 | 86.9 | **93.1** | **92.9** | 87.7 | 85.9 | **93.1** | **91.1** |
| Edge Metrics | Count Reference | 5.01 |  |  |  | 7.11 |  |  |  |
|  | Count Predicted | 57.95 | 11.86 | 5.15 | 7.45 | 10.64 | 6.39 | 7.03 | 8.52 |
|  | MEP (x$10^2$) ↑ | 22.4 | 24.52 | **53.50** | **35.81** | 17.94 | 12.40 | **37.43** | **27.22** |
| Novelty (x$10^2$) ↑ |  | 12.26 | 57.59 | **87.37** | **75.75** | **20.22** | **25.39** | 9.48 | 12.98 |
| Diversity (x$10^2$) ↑ |  | 96.70 | **98.73** | 79.62 | **91.68** | 90.92 | **93.73** | 75.09 | **88.80** |

Table 1. Comparison of our method (GEMS) with the baselines [42] and [9] on Visual Genome [21] and VRD [25] datasets. For MMD metrics, lower is better (↓). For remaining metrics, larger is better (↑). GraphRNN* and GraphRNN** referes to GraphRNN with $max\_prev\_node = 6$ and 7 for Visual Genome and VRD respectively. **Note:** **red** and represents best and **blue** second best scores.

by $max\_prev\_node$. We set $max\_prev\_node$ empirically by choosing the value which covers $99^{th}$ percentile of all graphs in the dataset. Note that this leads to a loss of information (some relationships are ignored), and is an efficiency trade-off. The value of $max\_prev\_node$ for Visual Genome and VRD used are 6 and 7 respectively. More details are provided in the supplementary. The GEMS model is trained on the observed scene graphs in the training set - the strategy to construct seed graphs is described in Algorithm 1, and the experiments reported here used $k = 1$. The groundtruth used for evaluation are seed-graph and expanded-graph pairs derived from the test set, where the complete observed test scene graphs are taken to be the desired completions and seed graphs are derived from them again as in Algorithm 1.

## 4.3. Baselines

Since scene graph expansion is a novel task there are no prior baselines. For our purpose, we transform GraphRNN [42] to work for scene graphs containing bi-directional edge-relations between nodes. Another variant of GraphRNN is used as a baseline, GraphRNN* where $max\_prev\_node$ is set to 6 and 7 respectively for Visual Genome and VRD. We observe that having a smaller value of $max\_prev\_node$ helps the model generate nodes and edges whose count distributions match that in the reference training set. We also compare our method against SceneGraphGen [9], which, though mainly focuses on the unconditional generation of scene graphs, can be leveraged for scene graph expansion. More details of how we transform GraphRNN to work on scene graphs and SceneGraphGen's implementation are provided in the supplementary.

|  |  | **GraphRNN*** | **GraphRNN*<br>(w/ CBFS)** | **GEMS**<br>($\alpha = 0$) | **GEMS** |
|---|---|---|---|---|---|
| MMD | NSPDK* (x$10^3$) ↓ | 5.10 | **0.47** | 1.39 | 1.21 |
|  | Node Label (x$10^4$) ↓ | 5.26 | **5.14** | 5.16 | 5.19 |
|  | Edge Label (x$10^2$) ↓ | 6.19 | 1.70 | **1.07** | 1.13 |
| Node Metrics | Count Reference | 11.09 |  |  |  |
|  | Count Predicted | 13.84 | 9.70 | 9.23 | 10.17 |
|  | $(Obj)_K$ (x$10^2$) ↑ | 86.9 | 92.8 | 92.6 | **92.9** |
| Edge Metrics | Count Reference | 5.01 |  |  |  |
|  | Count Predicted | 11.86 | 2.74 | 6.84 | 7.45 |
|  | MEP (x$10^2$) ↑ | 25.52 | 19.9 | 35.40 | **35.81** |

Table 2. Evaluation of different components of our method on Visual Genome: GraphRNN with $max\_prev\_node = 6$ (GraphRNN*); GraphRNN* with Cluster-Aware Sequencing (GraphRNN* w/ CBFS); Our method without the use of external knowledge in the node loss (GEMS($\alpha = 0$)); The final model GEMS including all components.

## 4.4. Evaluation Protocol

Evaluation of generative models is a difficult task [36]. Current practice within the graph generation community is the use of Maximum Mean Discrepancy (MMD) as a way to characterise the performance of alternative models. Given two samples of graphs $\mathbf{G}_1 = \{G_{11}, G_{12}, ..., G_{1m}\} \sim \mathbb{G}$ and $\mathbf{G}_2 = \{G_{21}, G_{22}, ..., G_{2n}\} \sim \mathbb{G}$, the MMD between these two samples – $MMD(f(\mathbf{G}_1), f(\mathbf{G}_2))$ – is characterised by two factors: (1) a descriptor function, referred to as $f$, that returns a distribution of some chosen property over the set; and (2) a kernel function that computes the distance between the distributions. We consider three classes of descriptor functions capturing **Structural** (number of nodes, number of edges, node degree, clustering coefficient), **Label** (node and edge types) properties of the graphs, as well as **Sub-Graph Similarities** (referred to NSPDK from [3]).

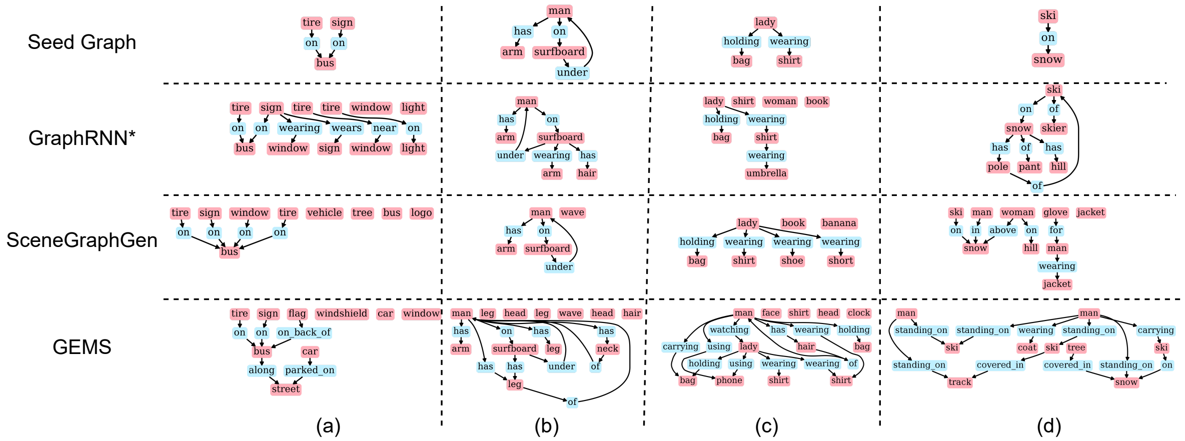

Figure 1. Comparison of expanded graphs generated by our model (GEMS) v/s baseline GraphRNN* (GraphRNN with $max\_prev\_node = 6$) and SceneGraphGen on Visual Genome seed graphs. Our model generates plausible relationships between objects, while GraphRNN* sometimes predicts irrelevant relationships.

The complexity of evaluating MMD is quadratic with respect to the number of samples in each set [11]. Computing MMD over subsets of the test set is one way to handle the computational cost of evaluation. Others [24, 9] propose alternative faster kernels to achieve the same purpose. The area of evaluation of graph generation models remains an open and active topic [27], our future work will look into the effects of these design choices. In the current work, we have followed the [10] and for Visual Genome we report the average value of MMD metrics calculated on $4$ independent test set splits. For VRD, we calculate MMD metrics on the entire test set. Our choice of kernel for MMD computation is the commonly used Gaussian kernel. In the results, we also report *Novelty*, as defined by [10], which computes the fraction of expanded scene graphs that are not sub-graph isomorphic to graphs in the training set. In the next section, we describe other metrics customized to the domain of scene graph generation.

In addition, similar to [9], we evaluate the quality of images generated from the expanded scene graphs between our method and the baselines using traditional metrics to evaluate quality of synthesized images, namely, Fréchet Inception Distance (FID) [15], Precision ($F_8$) and Recall ($F_{1/8}$) [30], and Inception Score (IS) [31]. The images are generated from the expanded scene graphs using pretrained models for Visual Genome dataset in sg2im [18] at a resolution of $64x64$. The generated images are compared against Ground-truth images from the Visual Genome dataset.

### 4.4.1 Metrics for Scene Completion

In this section, we introduce two new metrics to evaluate the output of scene graph generation methods. While we are utilizing them in a conditional setting, they are also valid for unconditional generation.

**Top-K Object Co-occurrence** $(Obj)_K$ The co-occurrence of a pair of objects in a set of graphs is calculated as the conditional probability of observing the pair in a scene graph given that one of the objects is present in the scene graph. We compare the co-occurrence of the $K$-most commonly observed pairs of objects in the test set with the co-occurrence of the corresponding pairs in the generated set of graphs as follows:

$$(Obj)_K = 1 - \frac{1}{K} \sum_{\substack{v_i, v_j \in \\ top_K(P_{test})}} \mid P_{test}[i,j] - P_{gen}[i,j] \mid \quad (11)$$

Here, $P_{test}$ ($P_{gen}$) is a matrix such that entry $(i,j)$ is the co-occurrence of the pair of objects $(v_i, v_j)$ in the test set (and generated set respectively). In combination with the other metrics, $(Obj)_K$ rewards a model that generates graphs containing coherent sets of objects.

**Modified Edge Precision (MEP)** is a metric inspired from modified n-gram precision [28] popularly used in NLP:

$$MEP = \underbrace{\min(1, exp^{(1-r/c)})}_{BrevityPenalty} \times \underbrace{\frac{\sum_{e \in G_E \cap (D_E \cup T_E)} 1}{\sum_{e \in G_E} 1}}_{EdgePrecision} \quad (12)$$

Here, $G_E$, $D_E$ and $T_E$ refer to the set of directed edges present in the generated graph $G$, the training set and the test set respectively. The variable $r$ is the average number of edges in the reference graphs, taken to be the set of test graphs containing the seed graph $G_s$ for which $G$ is the expansion. $c$ is the number of edges in the expanded scene graph $G$. Note that this metric brings information orthogonal to the others, as shuffling the edge labels in a scene graph would yield the same score on the remaining metrics.

In addition, we use an alteration to the Neighbourhood Sub-graph Pairwise Distance Kernel (NSPDK) based MMD metric. NSPDK computes the distance between two graphs by matching pairs of sub-graphs with different radii $r$ and distances $d$. Since the Node label MMD already does a

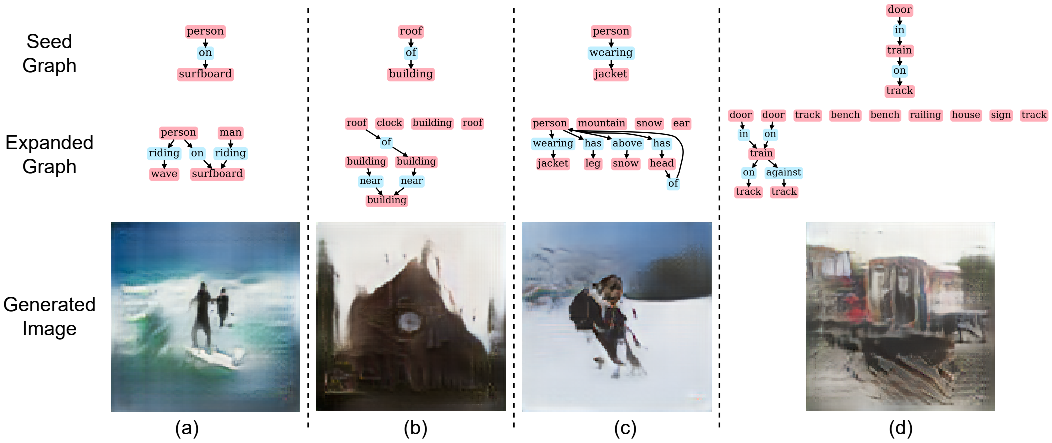

Figure 2. Examples of images generated by sg2im [18] using expanded scene graphs from seed graphs using our method (GEMS).

| | **FID** (↓) | **Inception** (↑) | **Precision** ($F_8$) (↑) | **Recall** ($F_{1/8}$) (↑) |
|---|---|---|---|---|
| GraphRNN* | 173.72 | 4.49 ± 0.06 | 0.031 | 0.185 |
| SceneGraphGen | **157.80** | **5.15 ± 0.135** | **0.040** | **0.235** |
| GEMS | **160.65** | **5.11 ± 0.1** | **0.045** | **0.240** |

Table 3. Comparison of the quality of images generated using sg2im [18] on the expanded scene graphs by different models for the same seed scene graph on Visual Genome dataset. **Note: red** represents best and **blue** represents second best scores.

node-level comparison, we exclude $(r, d) = (0, 0)$ and start from $(r, d) = (0, 1)$ instead. The altered metric, referred to as NSPDK* in the results, better captures the contribution of larger sub-graphs.

## 4.5. Main Results

Table 1 shows the comparison of graphs generated by GEMS and baselines methods in different metrics. For Visual Genome, our model outperforms the GraphRNN based baseline methods by a significant margin on all metrics demonstrating that graphs generated by our method are more meaningful and more closely resemble the observed scene graph distribution. For VRD our method shows comparable results to GraphRNN*, but outperforms GraphRNN in almost all metrics. We also provide SceneGraphGen results. A key contributor for SceneGraphGen's performance are the architectural advances that we do not use in our work. Still our method outperforms SceneGraphGen in terms Node Label and Degree MMD metrics for Visual Genome dataset, indicating better node predictions in the expanded scene graphs for Visual Genome. Figure 1 shows a qualitative comparison of graphs generated by GEMS and GraphRNN*, and SceneGraphGen. We note that GEMS is able to produce relationships amongst nodes that are more semantically meaningful. For e.g., we see relationships of the form $surfboard - wearing - arm$ and $snow - has - pole$ from GraphRNN*.

Table 3 shows a comparison of the quality of images generated using the pre-trained sg2im [18] on the expanded scene graphs by our model and the baselines for the same seed graph on Visual Genome dataset. Our model performs better in terms of Precision and Recall compared to all the baselines. For FID and Inception however, SceneGraphGen slightly outperforms GEMS. From an application perspective, it is important for our model to be able to produce novel and diverse, but plausible, scene graphs that can then be used as input conditioning for image synthesis models.

Figure 2 shows a few examples of seed graphs, the corresponding expanded scene graphs generated by our method (GEMS), and the images generated by sg2im [18] using the expanded scene graphs. We can see that from an abstract seed graphs like $roof - on - building$ or $person - wearing - jacket$, we are able to generate complete and meaningful scenes like a building having a large clock at the front or a person wearing jacket on a snowy mountain.

## 4.6. Ablation Study

### 4.6.1 Cluster-Aware BFS

From Table 2, it can be observed that Cluster-Aware BFS drastically improves the performance on NSPDK* and $(Obj)_k$ indicating that the generated graphs contain similar clusters of co-occurring objects as in the set of graphs in the training set. Figure 3 (last 2 rows) qualitatively shows the benefits of our Cluster-Aware BFS strategy for converting graphs into sequences. For e.g., (l) adds the cluster - $window, windshield, bus$ to the seed graph because these 3 objects would occur together in observed scenes. In (o) $flower, table$ are added to seed graph $vase - on - stand$ as $flower$ and $table$ occur together with $vase$ and $stand$.

### 4.6.2 Subject-Object Context & Class-Balancing Loss

Adding subject-object context to EdgeRNN for edge prediction enables the model to predict more meaningful relationships. Additionally, Class-balancing Loss is required to tackle the skewness in our scene graph dataset such that

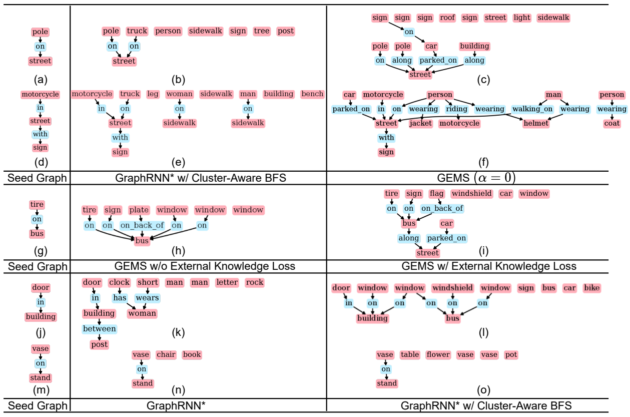
pole on street
(a)
pole truck person sidewalk sign tree post
on on
street
(b)
sign sign sign roof sign street light sidewalk
on
pole pole car building
on along parked_on along
street
(c)
motorcycle in street with sign
(d)
motorcycle truck leg woman sidewalk man building bench
in on on on
street sidewalk sidewalk
with
sign
(e)
car motorcycle person man person
parked_on in on wearing riding wearing walking_on wearing wearing
street jacket motorcycle helmet coat
with
sign
(f)
Seed Graph
GraphRNN* w/ Cluster-Aware BFS
GEMS (α = 0)
tire on bus
(g)
tire sign plate window window window
on on on_back_of on on
bus
(h)
tire sign flag windshield car window
on on on_back_of
bus car
along parked_on
street
(i)
Seed Graph
GEMS w/o External Knowledge Loss
GEMS w/ External Knowledge Loss
door in building
(j)
door clock short man man letter rock
in has wears
building woman
between
post
(k)
door window window windshield window sign bus car bike
in on on on on on
building bus
(l)
vase on stand
(m)
vase chair book
on
stand
(n)
vase table flower vase vase pot
on
stand
(o)
Seed Graph
GraphRNN*
GraphRNN* w/ Cluster-Aware BFS

Figure 3. The behavior brought by each component of our model in isolation (Middle and Right column results are without and with using a component respectively). $1^{st}$ two rows show the benefits of Subject-Object addition to edge prediction model, the use of GloVe embeddings and class-balancing edge loss. $3^{rd}$ depicts advantage of using external-knowledge for node prediction. The last two rows show the advantage of Cluster-Aware BFS.

model produces a more diverse set on relationships. The Edge Metrics in Table 2 show significant improvement in MEP and edge count compared to models without these components, thereby validating our hypothesis. Via our qualitative examples, we notice that the expanded graphs are not dominated by the $on$ and $with$ relations that occur very frequently in the Visual Genome dataset. Instead, a richer and more diverse set of edges (e.g. $parked\_on$, $wearing$, $riding$, $walking\_on$, $walking$) are generated.

### 4.6.3 External Knowledge Loss

Adding external knowledge helps in the addition of relatively similar but diverse nodes and not repeated node labels. This evident from the middle coloumn of Figure 3 . In (h) the model adds $window$ several times to $bus$, but in (i) the models adds both $window$ and $windshied$ (different node labels but similar in terms of glove embeddings used in the external-knowledge loss). From Table 2, it can be seen that adding external-knowledge loss improves $(Obj)_k$.

### 4.6.4 Multiple Outputs

At every step of GEMS, a node is first sampled from the multinomial distribution over node labels (output of $f_{node}$), and edges are added from previous sampled nodes to the new one (by $f_{edge}$). This process continues until an end-of-sequence node token is obtained. Our model can be used to generate $M$ alternative expansions of the same seed scene graph by invoking the sampling process multiple times with different seeds – Figure 4 provides qualitative examples.

For quantitative evaluation, we define a diversity metric, from a set of $M (= 3)$ expansions, we exclude the expansion that is sub-graph isomorphic to one of the other expansions,

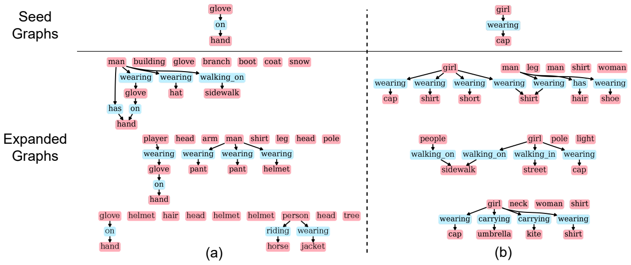
Seed Graphs
glove on hand
girl wearing cap
Expanded Graphs
man building glove branch boot coat snow
wearing wearing walking_on
glove hat sidewalk
has on
hand
player head arm man shirt leg head pole
wearing wearing wearing wearing
glove pant pant helmet
on
hand
glove helmet hair head helmet helmet person head tree
on riding wearing
hand horse jacket
(a)
girl man leg man shirt woman
wearing wearing wearing wearing wearing has wearing
cap shirt short shirt hair shoe
people girl pole light
walking_on walking_on walking_in wearing
sidewalk street cap
girl neck woman shirt
wearing carrying carrying wearing
cap umbrella kite shirt
(b)

Figure 4. Three different scene graph expansions produced by our model (GEMS) for the same input seed graph. GEMS not only adds diverse objects to the seed graph but also generates diverse visual scenarios.

and compute the percentage that remains. The results provided in Table 1 suggest a bias-variance trade-off. Across datasets, GraphRNN* has higher diversity, indicating that the predicted distribution over node labels at every step is flatter. Encouraging our GEMS model to produce diverse variations, while still respecting the training set distribution remains a topic for future work.

## 5. Discussion and Conclusion

In this paper, we considered the novel task of *scene graph expansion* – given an input seed scene graph, we enhance it by the addition of objects and relationships. The output, representing a more complex scene, is expected to respect co-occurrence patterns between objects and their relationships, while being novel with respect to the training set. Doing so automatically is enabled by the use of generative models of graphs, which provide a scalable mechanism to model real world graphs in multiple domains.

Our extensive experimental section illustrated that the standard MMD based evaluation does not highlight all behavioral characteristics of the models. In particular, we confirm the observation made by others that while models achieve satisfactory results on object-centric prediction tasks, modeling relationships is harder [38]. We propose new metrics specifically focussed on this aspect, and compare our models and baselines on the new metrics. However, evaluation of conditional and unconditional generation of scene graphs remains challenging. In particular, the design of metrics that capture the semantic plausibility of a generated scene graph is an important future direction.

From an application perspective, we have shown that the graph expansion mechanism allows us to generate candidate enriched scenes that can be provided as recommendations to creatives composing complex scenes. Leveraging external knowledge via embeddings from the linguistic domain allows our model to produce semantically realistic scene graph completions. Our autoregressive model also allows us to produce multiple diverse completions for the same input. Future work will aim to couple the graph and image domains, towards the end goal of scene synthesis.

# References

[1] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs, 2018.

[2] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *ArXiv*, abs/1805.11973, 2018.

[3] Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *ICML*, 2010.

[4] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[5] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE conference on computer vision and Pattern recognition*, pages 3076–3086, 2017.

[6] Hanjun Dai, Azade Nazi, Yujia Li, Bo Dai, and Dale Schuurmans. Scalable deep generative modeling for sparse graphs. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2302–2312. PMLR, 13–18 Jul 2020.

[7] Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, and Fei-Fei Li. Visual relationships as functions: Enabling few-shot scene graph prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[8] Shuangfei Fan and Bert Huang. Labeled graph generative adversarial networks. *arXiv preprint arXiv:1906.03220*, 2019.

[9] Sarthak Garg, Helisa Dhamo, Azade Farshad, Sabrina Musatian, Nassir Navab, and Federico Tombari. Unconditional scene graph generation. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[10] Nikhil Goyal, Harsh Vardhan Jain, and Sayan Ranu. Graphgen: A scalable approach to domain-agnostic labeled graph generation. WWW '20, New York, NY, USA, 2020. Association for Computing Machinery.

[11] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

[12] Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs. In *International conference on machine learning*, pages 2434–2444. PMLR, 2019.

[13] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation, 2020.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[16] Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1670–1679, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[17] Wengong Jin, Regina Barzilay, and Tommi S. Jaakkola. Junction tree variational autoencoder for molecular graph generation. *CoRR*, abs/1802.04364, 2018.

[18] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018.

[19] 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.

[20] Matthew Klawonn and Eric Heim. Generating triples with adversarial networks for scene graph construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[22] 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 international conference on computer vision*, pages 1261–1270, 2017.

[23] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

[24] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L Hamilton, David Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 4255–4265, 2019.

[25] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 852–869, Cham, 2016. Springer International Publishing.

[26] Yichao Lu, Himanshu Rai, Jason Chang, Boris Knyazev, Guangwei Yu, Shashank Shekhar, Graham W Taylor, and Maksims Volkovs. Context-aware scene graph generation with seq2seq transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15931–15941, 2021.

[27] Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions, 2021.

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine

translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[29] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[30] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in Neural Information Processing Systems*, 31, 2018.

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[32] Bidisha Samanta, Abir DE, Gourhari Jana, Pratim Kumar Chattaraj, Niloy Ganguly, and Manuel Gomez Rodriguez. Nevae: A deep generative model for molecular graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1110–1117, Jul. 2019.

[33] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, pages 412–422. Springer, 2018.

[34] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13936–13945, 2021.

[35] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3713–3722, 2020.

[36] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[37] Tathagat Verma, Abir De, Yateesh Agrawal, Vishwa Vinay, and Soumen Chakrabarti. Varscene: A deep generative model for realistic scene graph synthesis. In *International Conference on Machine Learning*, pages 22168–22183. PMLR, 2022.

[38] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2019.

[39] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

[40] 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 (CVPR)*, July 2017.

[41] Pengfei Xu, Xiaojun Chang, Ling Guo, Po-Yao Huang, Xiaojiang Chen, and Alexander G Hauptmann. A survey of scene graph: Generation and applications. *EasyChair Preprint*, (3385), 2020.

[42] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5708–5717. PMLR, 10–15 Jul 2018.

[43] Ruichi Yu, Ang Li, Vlad I. Morariu, and L. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1068–1076, 2017.

[44] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

[45] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship proposal networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5678–5686, 2017.