Energy-Based Learning for Scene Graph Generation

Mohammed Suhail[1,2]   Abhay Mittal[4]   Behjat Siddiquie[4]   Chris Broaddus[4]
Jayan Eledath[4]   Gerard Medioni[4]   Leonid Sigal[1,2,3]

[1]University of British Columbia   [2]Vector Institute for AI   [3]Canada CIFAR AI Chair   [4] Amazon

suhail33@cs.ubc.ca   mrmittal@amazon.com   behjats@amazon.com   chrispb@amazon.com

eledathj@amazon.com   medioni@amazon.com   lsigal@cs.ubc.ca

Abstract

Traditional scene graph generation methods are trained using cross-entropy losses that treat objects and relationships as independent entities. Such a formulation, however, ignores the structure in the output space, in an inherently structured prediction problem. In this work, we introduce a novel energy-based learning framework for generating scene graphs. The proposed formulation allows for efficiently incorporating the structure of scene graphs in the output space. This additional constraint in the learning framework acts as an inductive bias and allows models to learn efficiently from a small number of labels. We use the proposed energy-based framework [1] to train existing state-of-the-art models and obtain a significant performance improvement, of up to 21% and 27%, on the Visual Genome [9] and GQA [5] benchmark datasets, respectively. Furthermore, we showcase the learning efficiency of the proposed framework by demonstrating superior performance in the zero- and few-shot settings where data is scarce.

1. Introduction

A scene graph is a graph-based representation of an image which encodes objects along with the relationships between them. Such a representation allows for a comprehensive understanding of images that is useful in several vision applications, including visual question answering [5, 22], image captioning [3, 28] and scene synthesis [4, 6].

A typical scene graph generation model comprises of the object detection network, which extracts object regions and corresponding features, and a message passing network with nodes initialized with these region features and edges accounting for the potential relations among them. The features are refined, through context aggregation, and then classified to produce both object (node) and relation (edge)

---

[1]Code and pre-trained models available at https://github.com/mods333/energy-based-scene-graph.
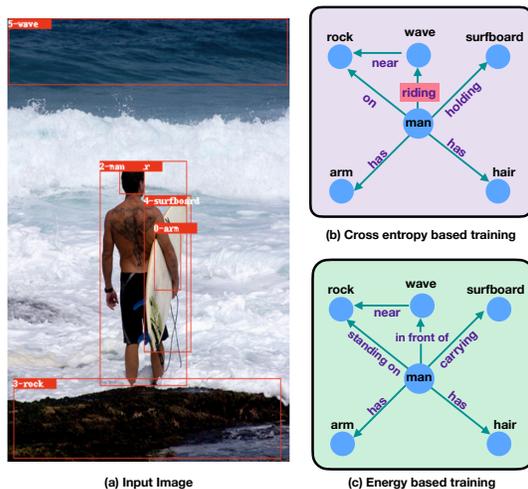

Figure 1. **Scene Graph Generation**: Figure shows scene graphs generated by a VCTree [22] model trained using conventional cross-entropy loss (purple) and our proposed energy-based framework (green). We make two crucial observations. First, the model trained using cross-entropy loss is incapable of consistent structural reasoning (`riding` is not possible given the rest of the graph). Second, the trained model tends to be biased, favoring more frequent relations (*e.g.*, `on`). Our proposed energy-based framework is designed, and able, to address these shortcomings.

labels. These networks are often trained end-to-end by minimizing individual cross-entropy losses on both sets of labels. A major drawback of such an approach is that quality of prediction (loss) is simply proportional to the number of correctly predicted labels and ignores the rich structure of the scene graph output space (*e.g.*, correlation or exclusion among object and relation label sets). In addition, the imbalance in the number of training samples for the relations results in dominant relations being heavily favored, leading to biased relation prediction at test time [21].

Figure 1 (b) illustrates the scene graph generated by a model [22] trained using the cross-entropy loss. Both the aforementioned drawbacks are apparent in the output. First,

the model predicts a relation `<man, riding, wave>`. A simple examination of the rest of the scene graph reveals that such a relationship is impossible given that the `man` is `on` a `rock` and `holding` a `surfboard`. Second, the model leans towards making generic relation predictions such as `<man, on, rock>` as opposed to more informative alternatives, *e.g.*, `<man, standing on, rock>`.

The origin of these issues can be identified by examining the likelihood term. Cross-entropy based training treats objects (O) and relationships (R) in a scene graph as independent entities. This amounts to factorizing the likelihood of a scene graph (SG), given an image (I), as the product of the likelihoods for the individual objects and relations:

$$\log p(SG|I) = \sum_{i \in O} \log p(o_i|I) + \sum_{j \in R} \log p(r_j|I). \quad (1)$$

Eq.(1) brings to light the underlying cause of the problem highlighted above. First, during loss computation, the loss for each relation term is independent of the relations predicted in the rest of the scene graph. Thus an incorrect relation such as `<man, riding, wave>` is penalized the same as `<man, behind, wave>` irrespective of the other relations (`<man, on, rock>`). However, using common sense reasoning, we can determine that `<man, riding, wave>` is highly improbable given `<man, carrying, surfboard>` and should be penalized heavily as opposed to a likely, albeit incorrect, relation `behind`. Second, due to the summation over individual relation terms, the model, in order to minimize the loss, is incentivized to predict relations which are more common in the training data.

While prior works have tried to address the issue of biased predictions [14, 21] in the context of scene graph generation, little progress has been made towards structured learning of scene graphs. In this work, we address both of these issues by proposing a novel generic loss formulation that incorporates the structure of scene graphs into the learning framework using an energy-based learning framework. This energy-based framework relies on graph message passing algorithm for energy computation, that is learned to model the joint conditional density of a scene graph, given an image. Such a formulation transforms the problem from maximizing sum of the individual likelihood terms to that of directly maximizing the joint likelihood of the objects and relations. Furthermore, this added structure acts as an inductive bias for the learning, allowing the model to efficiently learn relationship statistics from less data.

The proposed learning framework is general and hence can be used to train any off-the-shelf scene graph generation model. We experiment with various state-of-the-art models and demonstrate that our energy-based formulation achieves significant improvements in the performance over the corresponding models trained using the standard cross-entropy based formulation. We also demonstrate the enhanced generalization capability of models trained using our framework by evaluating zero shot relation retrieval performance. Finally, we demonstrate the ability of our energy-based framework to learn from lesser amounts of training data by demonstrating an improvement in relative performance when evaluating on few-shot relation triplets.

Figure 1 (c), shows the scene graph generated by the proposed method. The generated scene graph is more granular, predicting relations such as `<man, standing on, rock>` as opposed to the biased and generic variant `<man, on, rock>`. The model is also able to preclude improbable relations (*e.g.*, `<man, riding, wave>`) and instead predicts `in front of` between `man` and `wave`.

Contribution: Our main contribution is a novel energy-based framework for scene graph generation that allows for direct incorporation of structure into the learning. We also propose a novel message passing algorithm that is used for computing the energy of scene graph configurations. This message passing algorithm is generic and can be used for other applications such as learning graph embeddings. Finally, we demonstrate the efficacy of our proposed framework by applying it to multiple state-of-the-art models and evaluating performance on two benchmark datasets - Visual Genome [9] and GQA [5] - where we consistently outperform the cross-entropy based counterparts by up to 21% on Visual Genome and 27% on GQA.

2. Related Work

Scene Graph Generation: Scene Graph generation has become increasingly popular in the vision community [8, 12, 14, 17, 21, 26, 27, 29, 30, 31]. Early works, *e.g.*, [26, 27, 31], focused on improving context aggregation modules that facilitated learning better representations, thereby improving performance. Recent works, [8, 14, 32], identify the shortcomings of training using cross-entropy loss and propose handcrafted loss formulations to improve the performance. Our work can be considered as a generalization of the latter, where we allow the model to learn an approximate joint distribution over the scene graphs and images using an energy based formulation. Furthermore, to the best of our knowledge, our work is the first to incorporate structure in the output space, into the learning paradigm.

Energy Based Modeling: Energy Based Models(EBM) learn the unnormalized density of the data space, thereby avoiding the need to compute the partition function [10]. EBMs have recently sparked interest in generative modeling. One line of work uses energy models to learn the underlying data distribution and then implicitly generates samples from the learned distribution [1, 2]. In [19], they propose denoising score matching based training, where the model learns to generate data by denoising progressively in-

creasing noisy signals. While there has been an increase in the use of energy-based modeling for generative tasks, they have been relatively unexplored for discriminative tasks. In our work, we show how one can formulate a discriminative task of scene graph generation using an EBM framework.

3. Approach

We first provide an overview of current approaches for scene graph generation based on the standard cross entropy loss, followed by a description of our proposed energy-based learning framework along with the architecture used for energy computation.

3.1. Scene Graph Generation

Scene graph generation methods typically adopt a two stage framework. In the first stage, given an image $\mathcal{I}$, bounding boxes are obtained using a standard object detector such as Faster R-CNN [18]. Features corresponding to these regions are extracted using `RoIAlign` along with an initial distribution over object labels. In the next stage, these detections are used as inputs to predict the scene graphs. The bounding-box features along with the object label and the spatial coordinates of the bounding boxes are used to initialize a set of node features. These features are refined using architectures such as LSTM [31], TreeLSTM [22] or Graph Attention Networks [27], to incorporate contextual information. Object labels, $\mathbf{O}$, are then obtained by classifying the refined features. Relationship labels, $\mathbf{R}$, are obtained by extracting features from union of object bounding boxes, followed by state refinement using BiLSTMs [31] or BiTreeLSTMs [22] and subsequent classification.

These models are trained using standard cross-entropy loss on object and relation labels. Each object and relationship is considered in isolation when computing individual losses, which are then summed up to obtain the loss for the given image. Such a loss formulation ignores the fact that objects and relations in a scene graph are interdependent. Intuitively, incorporating such dependencies into the learning procedure should lead to an improvement in performance. However, it is not clear as to how one can exploit the rich structure in the output space. Most methods ([22, 26, 27, 31]) attempt to find a way around this by employing message passing algorithms in the input space that allow for aggregation of context information. However, this does not explicitly consider structure in the output space; neither in predictions, nor in the loss function used for learning. In this work, we propose a novel energy-based learning framework that allows scene graph generation models to be trained using a "loss" that explicitly incorporates structure in the output space.

3.2. Energy Based Modeling

Energy-based models [10] encode dependencies between variables by assigning a scalar energy value to an input configuration. Given a data point $\mathbf{x} \in \mathcal{X}$ with corresponding label $\mathbf{y} \in \mathcal{Y}$, let $E_\theta(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$ be a joint energy function. While energy models map inputs to unnormalized densities, we can define a probability distribution via the Boltzmann distribution, $p_\theta(\mathbf{x}, \mathbf{y}) = \frac{\exp(-E_\theta(\mathbf{x}, \mathbf{y}))}{Z(\theta)}$, where $Z(\theta) = \int \exp(-E_\theta(\mathbf{x}, \mathbf{y}))$ is referred to as the normalization constant or partition function. Computing the normalization constant, Z_θ for most parameterizations of the energy function is intractable. Therefore, learning the parameters θ using methods such as maximum likelihood is not straightforward. Most methods address this problem by rewriting the derivative of the log likelihood as $\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{p_\theta(\mathbf{x}', \mathbf{y}')}[\nabla_\theta E_\theta(\mathbf{x}, \mathbf{y}')] - \nabla_\theta E_\theta(\mathbf{x}, \mathbf{y})$, where the expectation is approximately estimated using MCMC methods that sample from the data distribution.

Unlike most prior works that train energy models for generative modeling, our focus is scene graph generation, a discriminative task. For such a task, we are only concerned with the relative energies of the various label configurations given an input $\mathbf{x}$. Training with a carefully crafted loss function circumvents the need for estimating the partition function or computing expectations. Therefore we can parametrize the energy function using an arbitrary neural network architecture. For a more detailed discussion on energy loss formulation refer to Section 2 in [11].

3.3. Energy Models for Scene Graphs Generation

We now describe our proposed energy-based learning framework for scene graph generation. In our formulation of the energy function, the data space $\mathcal{X}$ is the set of images $\mathcal{I} \in \mathbb{R}^{W \times H \times 3}$ and the label space $\mathcal{Y}$ is the set of scene graphs SG. The scene graph, SG, is defined by a tuple $(\mathbf{O}, \mathbf{R})$, where $\mathbf{O} \in \mathbb{R}^{n \times d}$ is the set of object labels and $\mathbf{R} \in \mathbb{R}^{n \times n \times d'}$ is the set of relationship labels; n is the number of objects in an image; d and d' is the total number of possible object and relation labels in the dataset.

A simple implementation of the joint energy function, would take an encoding of the image, $\mathcal{I}$, and a scene graph, SG, and produce an scalar energy value. However, there are a few challenges with this. First, a simple global CNN-based encoding of the image may fail to highlight, potentially small, regions relevant for scene graph prediction. Second, scene graph representation is variable in length (n is not fixed) and high dimensional. The second challenge can be addressed by pooling object $\mathbf{O}$ and relations $\mathbf{R}$ across n and $n \times n$ dimensions respectively. We propose a more sophisticated and effective scene graph refinement and gated pooling formulation in Section 3.4. To facilitate the former challenge of image encoding, we extract a graph
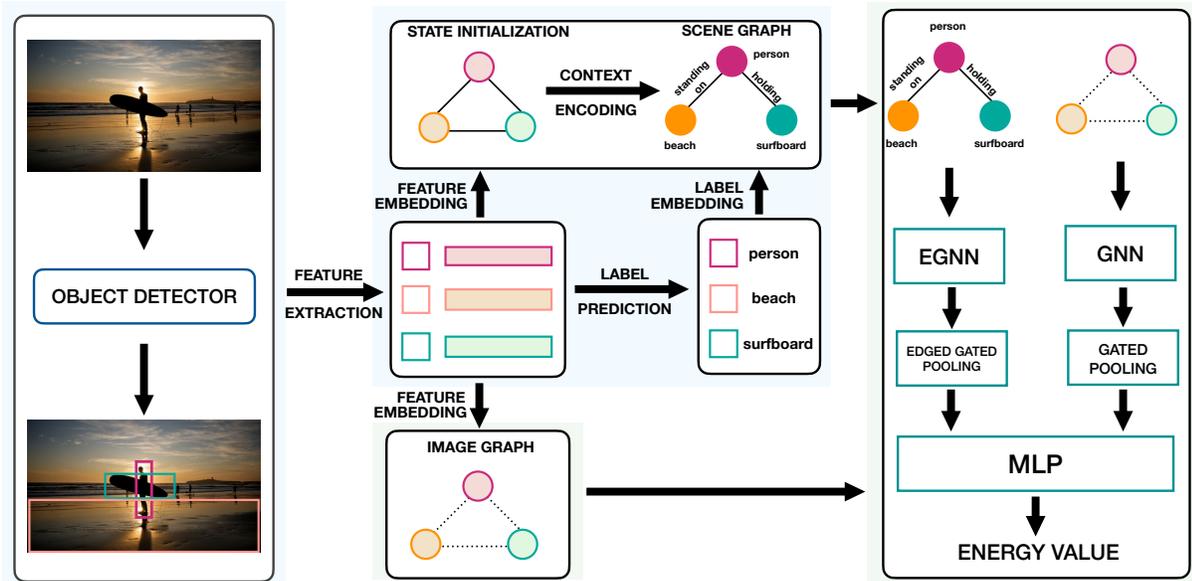
Figure 2. **Model Overview of the Energy-based Learning.** The region in light blue correspond to most traditional scene graph generation pipelines. The proposed energy-based learning framework is highlighted in light green. We initialize the image graph with the extracted object proposal features as the node states. We instantiate the scene graph with predictions from traditional pipeline (or ground truth annotation). The image graph and scene graph are fed into the energy model where they undergo state refinement using a Gated Graph Neural Network and a novel Edged Graph Neural Network, respectively. We then obtain vector representations of each graph using pooling layers. The representations are concatenated and passed as input to a multi-layer perceptron which predicts the energy of the joint input (image) - output (scene graph) configuration. The loss is computed from the energy values of the ground truth and predicted configuration.

based representation from the image. This representation is henceforth referred to as an image graph ($\mathsf{G}_{\mathcal{I}}$). The nodes of the image graph are instantiated using features extracted from the object bounding boxes.

Given a scene graph generation model M and an image $\mathcal{I}$, we predict a scene graph, G_{SG}^0 and compute the image graph $\mathsf{G}_{\mathcal{I}}$. The scene graph along with the image is provided as input to the energy model (E_θ) to compute the energy corresponding to the predicted configuration. Similarly, we compute the energy of the ground truth configuration using the ground truth scene graph (G_{SG}^+) and image graph ($\mathsf{G}_{\mathcal{I}}^+$) constructed from ground truth bounding boxes. These two energy values are then used to compute the energy loss

$$\mathcal{L}_e = E_\theta(\mathsf{G}_{\mathcal{I}}^+, \mathsf{G}_{SG}^+) - \min_{\mathsf{G}_{SG} \in SG} E_\theta(\mathsf{G}_{\mathcal{I}}, \mathsf{G}_{SG}). \quad (2)$$

Computing this loss requires solving an optimization problem to find a scene graph configuration that minimizes the energy value (second term in Eq.(2)). Starting from G_{SG}^0 we use Stochastic Gradient Langevine Dynamics (SGLD) [24] which approximately solves the optimization problem in an iterative manner:

$$\mathbf{O}^{\tau+1} = \mathbf{O}^\tau - \frac{\lambda}{2} \nabla_{\mathbf{O}} E_\theta(\mathsf{G}_{\mathcal{I}}, \mathsf{G}_{SG}^\tau) + \epsilon^\tau,$$

$$\mathbf{R}^{\tau+1} = \mathbf{R}^\tau - \frac{\lambda}{2} \nabla_{\mathbf{R}} E_\theta(\mathsf{G}_{\mathcal{I}}, \mathsf{G}_{SG}^\tau) + \epsilon^\tau \quad (3)$$

where $\mathbf{O}^t$ and $\mathbf{R}^t$ are the node and edge states in the scene graph and ϵ^t is sampled from a normal distribution $\mathcal{N}(0, \lambda)$.

Conceptually, the predicted G_{SG}^0 is used as an "initialization" to arrive at the low energy configuration through a series of steps defined by Eq.(3); each step similar to regular gradient descent with an added Gaussian noise. Differentiation through the optimization path guides parameter learning of the model M that generates G_{SG}^0 in the first place.

When training using the above loss, we observe that the energy values get arbitrarily large in magnitude leading to gradient overflow. We address this problem by adding an $L2$ regularization loss on the energy values:

$$\mathcal{L}_r = E_\theta(\mathsf{G}_{\mathcal{I}}^+, \mathsf{G}_{SG}^+)^2 + E_\theta(\mathsf{G}_{\mathcal{I}}, \mathsf{G}_{SG})^2. \quad (4)$$

Finally, since the space of scene graphs is very high dimensional, we need to restrict the search space of the energy model in order to stabilize the learning. This is done by incorporating the task loss used by the underlying scene graph generation model, $\mathcal{L}_t$ on the predicted output as an added regularization on the initial prediction. The total loss for training the scene graph generator and the the energy model is given by:

$$\mathcal{L}_{total} = \lambda_e \mathcal{L}_e + \lambda_r \mathcal{L}_r + \lambda_t \mathcal{L}_t, \quad (5)$$

where λ_e, λ_r and λ_t are the relative weights.

3.4. Energy Model Architecture

Given an image graph ($G_{\mathcal{I}}$) and a scene graph (G_{SG}), the energy model first refines the state representations using graph neural networks. We use a novel Edged Graph Neural Networks (EGNN) and Graph Neural Network [13] on the scene graph and image graph respectively to incorporate contextual information. This is followed by applying a pooling layer on each graph to obtain a vector representation summarizing the graph states. Finally, these two vectors are fed into a multi layer perceptron (MLP) to compute the energy value corresponding to the predicted scene graph configuration. We then repeat these operations for the ground truth scene graph and the input image. The energy model can be parametrized as:

$$E_\theta(G_{\mathcal{I}}, G_{SG}) = \texttt{MLP}\left[f(\texttt{EGNN}(G_{SG})); g(\texttt{GNN}(G_{\mathcal{I}}))\right], \quad (6)$$

where f and g are pooling functions.

Notation. We use $\mathbf{n}_i$ to represent the features of the ith node, which for G_{SG} corresponds to the ith object, initialized to the corresponding i^{th} row of matrix $\mathbf{O}$. For $G_{\mathcal{I}}$ $\mathbf{n}_i$ corresponds to the ith image region, initialized by the RoIAlign image features. We use $\mathbf{e}_{i \to j}$ to represent the feature of the directed edge from node i to j, initialized to $(i, j)^{th}$ column of $\mathbf{R}$. $\mathcal{N}_i$ denotes the neighbours of node i.

3.4.1 Edge Graph Neural Network

To allow for direct application of convolution operations on graph representations accommodating edge features such as scene graphs, we propose a variant of graph message passing algorithm. For each node $\mathbf{n}_i$, we aggregate the message from neighbouring node and edges by

$$\mathbf{m}_i^t = \alpha \, \mathbf{W}_{nn} \underbrace{\left(\sum_{j \in \mathcal{N}_i} \mathbf{n}_j^{t-1} \right)}_{\text{node to node message}} + (1-\alpha) \, \mathbf{W}_{en} \underbrace{\left(\sum_{j \in \mathcal{N}_i} \mathbf{e}_{j \to i}^{t-1} \right)}_{\text{edge to node message}},$$

$$(7)$$

where $\mathbf{W}_{nn}$ and $\mathbf{W}_{en}$ are the kernel matrices for node-to-node and node-to-edge communication and $0 \le \alpha \le 1$ is a hyper-parameter that controls the contribution of messages from edges and nodes. Similarly, the message passing for edges are given by

$$\mathbf{d}_{i \to j}^t = \mathbf{W}_{ee}[\mathbf{n}_i^{t-1} \| \mathbf{n}_j^{t-1}], \quad (8)$$

where $\mathbf{W}_{ee}$ is the kernel matrix for node-to-edge communication. Note that the message passing for edges is direction aware i.e. $\mathbf{d}_{i \to j} \ne \mathbf{d}_{j \to i}$. This is crucial as the relationship between two nodes change depending on direction of the edge for example <cat, has, tail> and <tail, on, cat>. The incoming messages are combined with the states using a gating mechanism [13].

3.4.2 Pooling Layer

We use gated pooling layers to generate vector representations of the two graphs. The pooling operation is given by

$$\mathbf{N} = \sum_k f_{gate}(\mathbf{n}_k) \odot \mathbf{n}_k \quad (9)$$

$$\mathbf{E} = \sum_{ij} g_{gate}(\mathbf{e}_{i \to j}) \odot \mathbf{e}_{i \to j} \quad (10)$$

where f_{gate} and g_{gate} are gating functions that map node and edge states to a scalar and $\odot$ represents element-wise multiplication. These two vectors are then passed through a linear layer, after concatenation, to obtain the final vector representation of the graph. In the image graph $G_{\mathcal{I}}$, since there are no edge features, we use the pooled node vector $\mathbf{N}$ as the vector representation of the graph.

4. Experiments

We present experimental results on two datasets: the Visual Genome dataset [9] and the GQA dataset [5].

Visual Genome: We use the pre-processed version of the dataset from [26]. The dataset consists of 108k images and contains 150 object categories and 50 predicate categories We use the original split with 70% of the images in the training set and the remaining 30% in the test set, with 5k images from the training set held out for validation [31].

GQA: The GQA dataset [5] is also constructed from images in the Visual Genome dataset. Starting from the scene graph annotations provided in Visual Genome, a normalization process is applied. This normalization process augments object and relation annotations and prunes inaccurate or unnatural relations. The resulting dataset contains a total of 1704 object categories, 311 relation categories We use the same $70 - 30$ split for the train and the test set, with 5k images in the validation set. Compared to Visual Genome, the GQA dataset has denser graphs with a larger number of object and relation categories.

4.1. Scene Graph Generation Models

The energy-based training introduced in this paper is generic and does not make any assumptions on the underlying scene graph generation model. This allows freedom in choosing the model architecture for image to scene graph mapping. On the Visual Genome dataset, we experiment with VCTree [22], Neural Motifs [31] and Iterative Message Passing [26]. We also experiment with VCTree-TDE [21], where the inference involves counterfactual reasoning. On the GQA dataset, we experiment with Transformers [23], instead of VCTree, as the larger number of object classes in the GQA dataset leads to considerably larger memory requirement in VCTree. The ability to experiment with different models demonstrates the versatility of our approach.

Dataset	Model	Method	Predicate Classification			Scene Graph Classification			Scene Graph Detection		
			mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100
Visual Genome	VCTree [22]	Cross Entropy	13.07	16.53	17.77	8.5	10.53	11.24	5.31	7.16	8.35
		EBM-Loss	**14.2**	**18.19**	**19.72**	**10.4**	**12.54**	**13.45**	**5.67**	**7.71**	**9.1**
	Motif [31]	Cross Entropy	12.45	15.71	16.8	6.95	8.85	9.05	5.07	6.91	8.12
		EBM-Loss	**14.17**	**18.02**	**19.53**	**8.18**	**10.22**	**10.98**	**5.66**	**7.72**	**9.27**
	IMP [26]	Cross Entropy	8.85	10.97	11.77	5.4	6.4	6.74	2.2	3.29	4.14
		EBM-Loss	**9.43**	**11.83**	**12.77**	**5.66**	**6.81**	**7.17**	**2.78**	**4.23**	**5.44**
	VCTree-TDE [21]	Cross Entropy	16.3	22.85	26.26	11.85	15.81	17.99	6.59	8.99	10.78
		EBM-Loss	**19.87**	**26.66**	**29.97**	**13.86**	**18.2**	**20.45**	**7.1**	**9.69**	**11.6**
GQA	Transformer [23]	Cross Entropy	1.17	2.48	3.69	.54	.97	1.29	-	-	-
		EBM-Loss	**1.28**	**2.94**	**4.71**	**.68**	**1.32**	**1.77**	-	-	-
	Motif [31]	Cross Entropy	.85	1.8	2.75	.42	.81	1.18	-	-	-
		EBM-Loss	**.94**	**2.1**	**3.19**	**.57**	**.9**	**1.26**	-	-	-
	IMP [26]	Cross Entropy	.5	.94	1.32	.28	.5	.65	-	-	-
		EBM-Loss	**.57**	**1.07**	**1.5**	**.34**	**.58**	**.76**	-	-	-

Table 1. **Quantitative Results.** We compare the proposed energy-based loss formulation against traditional cross-entropy loss using various state-of-the-art models. We report the mean Recall@K [22] under all three experimental setting.

4.2. Evaluation

Relationship Recall (RR). We use the **mean Recall@K (mR@K)** metric [22] to evaluate the performance of the scene graph generation models. We report the mean Recall@K instead of the Regular Recall@K (R@K) due to data imbalance that leads to reporting bias as pointed out in recent works [21]. The evaluation is performed under three settings, (1) Predicate Classification (**PredCls**): Predict the relationship labels, given the image, object bounding boxes and object labels. (2) Scene Graph Classification (**SGCls**): Predict the object and predicate labels, given the image and bounding boxes and (3) Scene Graph Detection (**SGDet**): Predict the scene graph from the image.

Zero-Shot Recall (zsR@K). Introduced in [15], zsR@K evaluates the ability to identify subject-predictae-object relation triplets that were not observed during training. We compute zsR@K for 3 settings: PredCls, SGCls and SGDet.

Few-Shot Recall. We introduce the few-shot Recall@K (fsR@K) metric that reports the Recall@K for relation triplets that occur a certain number of times in the training set. Unlike the conventional few shot metric, we use a range of values to generate the few shot triplet splits. Thus instead of splitting the triplets into 1-shot, 2-shot, *etc.*, we split them into groups of $1 - 5$-shot, $6 - 10$-shot, *etc.* with triplets in a $k_1 - k_2$-shot occurring between k_1 and k_2 times.

Sentence-to-Graph Retrieval (S2GR): Introduced in [21], S2GR was designed to address the inability of RR (Relationship Recall) and zsR (Zero-Shot Recall) to capture graph level coherence. In S2GR, the scene graph predicted for an image (obtained in the SGDet setting) is used as a semantic representation of the image. The task then is to retrieve images using their graph representation, with the image caption as a query. Note that retrieval task is based solely on the detected scene graph and no other visual information. As a consequence, any bias in the scene graph generation will result in a decrease in the S2GR performance. Similar to [21], we report Recall@20/100 on 1k/5k gallery.

4.3. Implementation Details

Detector. We pre-train a Faster R-CNN [18] with ResNeXt-101-FPN [16, 25] backbone. We obtain the weights of the pre-trained detector for Visual Genome from [20]. For GQA, we pre-train the object detector using standard Faster-RCNN settings. The object detector has 28 mAP on Visual Genome and 10 mAP on GQA.

Scene Graph Generator. The baseline models, trained with standard cross entropy loss, as well as our proposed framework, are trained using an identical setup. We use an SGD optimizer with an initial learning rate of 10^{-2}. For the Visual Genome models, we incorporate frequency bias [31] into the training and inference. We do not use the frequency bias on GQA dataset, due to high memory requirements.

Energy Model. In the sampling step of SGLD (Eq. 3), we set the number of iterations (τ) to 20 and the number of message passing iterations in EGNN and GNN to 3. We use a step size of 1 and clip the gradients within $[-0.01, 0.01]$. After every gradient update step, we normalize the node and edge states of the scene graph to the range of $[0, 1]$.

Sentence-to-Graph Retrieval. We use the same formulation for S2GR as previous work [21]. The problem is formulated as a graph matching problem where the image caption is converted to graph structure using [9]. The scene graphs and text graphs are mapped into a joint embedding space for retrieval using a Bilinear Attention Network [7].

Dataset	Model	Method	PredCls zsR@20/50	SGCls zsR@20/50	SGDet zsR@20/50
VG	VCTree	CE	1.43/4	.39/1.2	.19/.46
		EB	**2.25/5.36**	**.87/1.87**	**.21/.54**
	Motif	CE	1.28/3.56	.39/.83	0/.04
		EB	**2.07/4.87**	**.52/1.25**	**.11/.23**
	IMP	CE	12.17/17.66	2.09/3.3	.14/.39
		EB	**12.6/18.6**	**2.29/3.7**	**.16/.43**
	VCTree-TDE	CE	8.98/14.52	3.16/4.97	1.47/2.3
		EB	**9.58/15.14**	**4.18/6.38**	**1.62/2.68**
GQA	Transformer	CE	19.55/33.33	.94/1.83	-
		EB	**20.11/34.33**	**1.2/2.05**	-
	Motif	CE	17.74/30.61	1.27/2.16	-
		EB	**19.47/33.45**	**1.49/2.48**	-
	IMP	CE	15.58/27.6	1.02/1.88	-
		EB	**16.65/27.77**	**1.1/1.98**	-

Table 2. **Zero-shot Recall.** The zero shot recall performance comparison of model trained using cross-entropy (CE) and energy-based loss (EB) on the Visual Genome (VG) and GQA dataset.

5. Experimental Results

Quantitative Results: Table 1 compares performance of various state-of-the-art methods trained using cross-entropy and our energy-based loss on two datasets, Visual Genome and GQA. We observe that training using the proposed energy loss leads to a consistent improvement in the the mean Recall in all three tasks, for all the models. For the VCTree model we obtain a relative improvement of 12.7%, 22.3% and 5.6% on mR@20 for PredCls, SGCls and SGDet respectively. We obtain relative improvements of similar magnitudes with the Motif, IMP and VCTree-TDE models. On the GQA dataset, we present results of three models, Transformer, Motif and IMP. Similar to the Visual Genome dataset, we observe a consistent improvement in the mean Recall metric under PredCls and SGCls, with each of the models. We omit experiments on SGDet task due to low mAP of the underlying object detector. The proposed method leads to a relative improvement of 8.5% and 25.92% in the mR@20 metric for the PredCls and SG-Cls task when using the Transformer model for scene graph generation. The absolute performance on the GQA dataset is lower, compared to Visual Genome dataset due to larger number of object and relationship classes. Additionally, due to memory constraints, we omit the highly effective [31] frequency prior on the GQA dataset.

We provide additional results along with individual relation recall in the **supplementary material**. We observe that energy-based models obtain larger improvement in relations that have fewer training annotations, compared to their cross-entropy based counterparts.

Zero-Shot Recall: Table 2 reports zero-shot recall (zsR@20 and zsR@50) for all models. Similar to mR@K,

Few-Shot Recall@20					
$k_1 - k_2$ shot	1-5	6-10	11-15	16-20	20-25
C.E.	16.9	24.41	27.73	31.52	32.31
E.B.M.	**18.55**	**25.22**	**28.1**	**32.05**	**32.57**

Table 3. **Few-shot Recall@20.** Table compares the few short recall performance of a VCTree [22] model trained using cross-entropy and energy-based loss.

Sentence to Graph Retrieval							
Gallery Size		1000			5000		
		R@20	R@50	R@100	R@20	R@50	R@100
VCTree	CE	14	28.4	44.6	4.1	8.9	14.98
	EBM	**17.2**	**32.5**	**48.6**	**5**	**10.96**	**18.52**
Motif	CE	15.5	29.4	46.7	4.56	9.7	17
	EBM	**19.2**	**32**	**49.2**	**5.22**	**10.96**	**18.64**

Table 4. **Sentence to Graph Retrieval.** We compare the scene graph retrieval performance on gallery of 1000 and 5000 images.

	Predicate Classification		
Ablation	mR@20	mR@50	mR@100
0-steps	14.18	18.14	19.66
20-steps	*14.2*	*18.19*	*19.72*
40-steps	14.37	18.18	19.81
60-steps	14.41	18.23	19.79
120-steps	14.59	19.29	20
No-Image-Graph	14.19	18.05	19.55

Table 5. **Ablation.** We experiment with the number of optimization steps (τ) needed to estimate the energy loss and the effect of excluding image information in the energy model. All numbers were obtained using a VCTree [22] model.

we note consistent improvement on the zero-shot recall. We attribute this behaviour to our energy-based structure aware framework, which facilitates learning of models that are capable of performing global scene graph reasoning.

Few-shot Recall: In Section 1, we hypothesized that the energy based learning bakes an inductive bias into the learning, thereby allowing the models to learn from less amounts of data. To test this hypothesis, we measure the few-shot Recall@20 for the VCTree model. We train a scene graph detection model using the proposed energy-based model as well as the standard cross-entropy loss. The result, as shown in Table 3, demonstrate that our training framework is able to provide a significant boost in performance in few-shot scenarios, when less data is available. This shows that the energy formulation provides a data efficient method for learning scene graph generation.

Sentence-to-Graph Retrieval: Table 4 compares the results of sentence-to-graph retrieval experiments. We use VCTree and Motif as our scene graph generation architecture. For each model we observe relative improvements
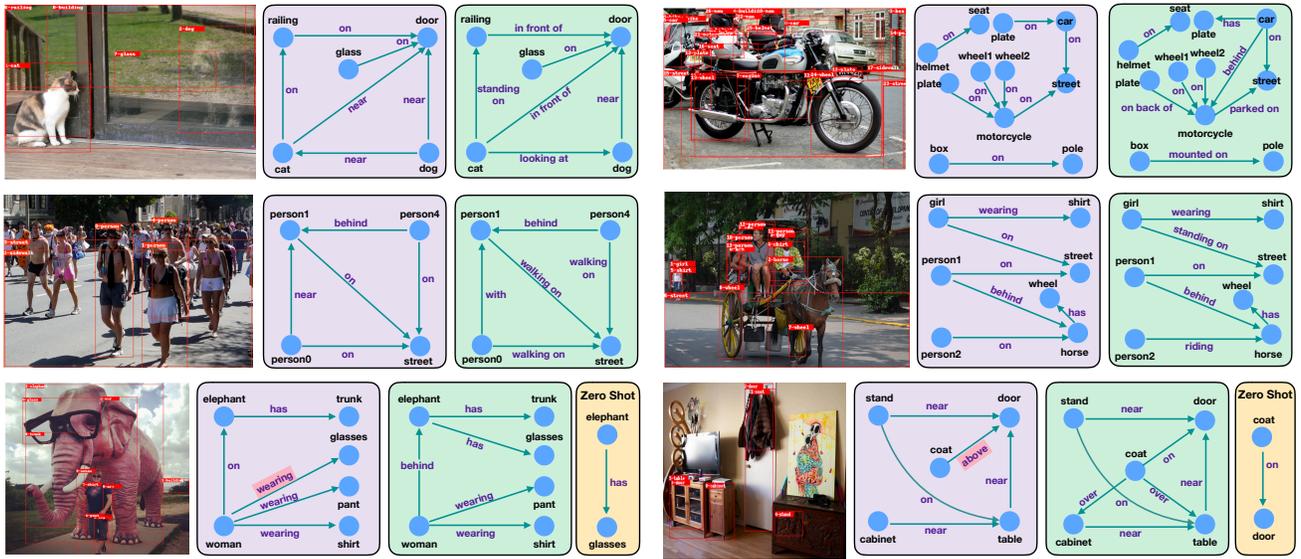
Figure 3. **Qualitative Results.** Visualizations of scene graphs generated by a VCTree [22] model trained using cross-entropy loss (in purple) and proposed energy-based loss (in green). The top two rows show visualization of relation retrieval. The bottom row shows zero-shot relation retrieval results with the zero shot triplet show in yellow.

ranging from 5%-23% in the retrieval recall compared to the scene graphs generated using corresponding baselines. This improvement can be attributed to the more coherent and informative scene graphs generated by our models.

Ablation Studies: We investigated the impact of optimization in Eq.(2) on the effectiveness of the energy model. We experimented with a different number of optimization steps while training a predicate classification model using VC-Tree. Note that increasing the number of iterations corresponds to a more precise minima. We also study the effect of removing the image graph input to the energy model to determine the effectiveness of modeling the energy density over the joint space of scene and image graph. Table 5 summarizes the mean recall@K for both of these experiments. We find that with an increase in the number of optimization steps in the energy loss, the mean recall almost consistently increases. Intuitively, this is expected as a larger number of optimization steps means that we have better chance of convergence when trying to find the minima in Eq.(2). This increase, however, comes with the added computation overhead and increase in training time. Similarly, we note that removing the image information from the joint modeling, there is a drop in performance as the model is now forced to learn from only the scene graph labels.

Qualitative Results: We visualize the qualitative results obtained from a VCTree model trained using the proposed energy-based framework as well as cross-entropy loss in Figure 3. The top two rows show results from the regular relation retrieval task. We observe that the models trained using our proposed framework can consistently

generate instructive relationships such as `mounted on`, `parked on`, `walking on`, `standing on` as opposed to the less informative and biased variant `on` generated by the baseline model. Similarly, in the top-left image, the energy-based training generates spatially informative relations such as `<cat, in front of, door>` instead of `<cat, near, door>` and `<cat, looking at, dog>` as opposed `<dog, near, cat>`. The bottom row shows results of zero-shot relation retrieval. In the first image, due to the triplets of `elephants` with `glasses` not being present in the training data, the baseline model predicts `<women, wearing, glasses>` whereas our method generates the accurate prediction `<elephant, has, glasses>`.

6. Conclusion

We present a novel model-agnostic energy-based learning framework for training scene graph generation models. Unlike cross-entropy based training, the proposed method embraces structure in the output space allowing the model to perform structure aware learning. We show that scene graph generation models can benefit from the proposed training framework by performing experiments on the Visual Genome and GQA datasets. We observe significant improvement in performance as compared to traditional cross-entropy based training. We also exhibit the generality and efficiency of our model through experiments in zero-shot and few-shot relationship settings. Finally, the proposed method does not make any assumptions on the underlying generation model and can be easily used with any model.

References

[1] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, pages 3608–3618, 2019. 2

[2] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020. 2

[3] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10323–10332, 2019. 1

[4] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[5] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5

[6] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018. 1

[7] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574, 2018. 6

[8] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Graph density-aware losses for novel compositions in scene graph generation. *arXiv preprint arXiv:2005.08230*, 2020. 2

[9] 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. *International journal of computer vision*, 123(1):32–73, 2017. 1, 2, 5, 6

[10] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 2, 3

[11] Yann LeCun and Fu Jie Huang. Loss functions for discriminative training of energy-based models. In *AIStats*, volume 6, page 34. Citeseer, 2005. 3

[12] Yikang Li, Wanli Ouyang, Zhou Bolei, Shi Jianping, Zhang Chao, and Xiaogang Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[13] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. In *4th International Conference on Learning Representations, ICLR*, 2016. 5

[14] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[15] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 852–869. Springer, 2016. 6

[16] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. 6

[17] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Advances in neural information processing systems*, pages 2171–2180, 2017. 2

[18] 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, 6

[19] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11918–11930, 2019. 2

[20] Kaihua Tang. A scene graph generation codebase in pytorch, 2020. https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch. 6

[21] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725, 2020. 1, 2, 5, 6

[22] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6619–6628, 2019. 1, 3, 5, 6, 7, 8

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 5, 6

[24] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011. 4

[25] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017. 6

[26] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5410–5419, 2017. 2, 3, 5, 6

[27] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018. 2, 3

[28] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019. 1

[29] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 606–623. Springer, 2020. 2

[30] Alireza Zareian, Zhecan Wang, Haoxuan You, and Shih-Fu Chang. Learning visual commonsense for robust scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 642–657. Springer, 2020. 2

[31] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018. 2, 3, 5, 6, 7

[32] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical co ntrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11535–11543, 2019. 2