Not All Relations are Equal: 
Mining Informative Labels for Scene Graph Generation

Arushi Goel\textsuperscript{1}, Basura Fernando\textsuperscript{2}, Frank Keller\textsuperscript{1}, and Hakan Bilen\textsuperscript{1}
\textsuperscript{1}School of Informatics, University of Edinburgh, UK 
\textsuperscript{2}CFAR, IHPC, A*STAR, Singapore.

Abstract

Scene graph generation (SGG) aims to capture a wide variety of interactions between pairs of objects, which is essential for full scene understanding. Existing SGG methods trained on the entire set of relations fail to acquire complex reasoning about visual and textual correlations due to various biases in training data. Learning on trivial relations that indicate generic spatial configuration like ‘on’ instead of informative relations such as ‘parked on’ does not enforce this complex reasoning, harming generalization. To address this problem, we propose a novel framework for SGG training that exploits relation labels based on their informativeness. Our model-agnostic training procedure imputes missing informative relations for less informative samples in the training data and trains a SGG model on the imputed labels along with existing annotations. We show that this approach can successfully be used in conjunction with state-of-the-art SGG methods and improves their performance significantly in multiple metrics on the standard Visual Genome benchmark. Furthermore, we obtain considerable improvements for unseen triplets in a more challenging zero-shot setting.

1. Introduction

In this paper, we look at a structured vision-language problem, scene graph generation \cite{26,56}, which aims to capture a wide variety of interactions between pairs of objects in images. SGG can be seen as a step towards comprehensive scene understanding and benefits several high-level visual tasks such as object detection/segmentation \cite{17,20,41}, image captioning \cite{1,18,22,30}, image/video retrieval \cite{16}, and visual question answering \cite{2,34}. In the literature, SGG is typically formulated as predicting a triplet of a localized subject-object pair connected by a relation (\textit{e.g.} person \textit{wear} shirt). Broadly, recent advances in SGG have been obtained by extracting local and global visual features in convolutional neural networks \cite{23,45,68} or graph neural networks \cite{31,56,61} combined with language embeddings \cite{36,39} or statistical priors \cite{65} for predicting relations between objects.

Despite the remarkable progress in this task, various factors (long-tail data distribution, language or reporting bias \cite{38}) in the established SGG benchmarks (\textit{e.g.} Visual Genome \cite{26}) have been shown to drive existing methods towards biased and inaccurate relation predictions \cite{49,50}. One major cause is that each subject-object pair is annotated with \textit{only one positive} relation, which typically depends on the annotator’s preference and is \textit{subjective}, while other plausible relations are treated as negative.\textsuperscript{1} For instance, a subject-object pair \textit{man–beach} in Fig. 1a is only annotated with one relation as \textit{man on beach}, even though other plausible relations are available such as \textit{standing on}. Hence, the models \cite{23,29,50,65} trained on this data become biased towards more frequently occurring labels, as reported in \cite{49}. To alleviate the biased training, Tang \textit{et al.} \cite{49} employ counterfactual causality to force the SGG model to base its predictions only on visual evidence rather than the data bias. Wang \textit{et al.} \cite{54} propose a semi-supervised technique that jointly imputes the missing labels of subject–object pairs with no annotated relations to obtain more balanced triplet distributions. Suhail \textit{et al.} \cite{47} propose an en-

\textsuperscript{1}Note that both training and test sets of the standard benchmarks are subject to the similar biases.
ergy based method that can learn to model the joint probability of triplets from few samples and thereby avoids generating biased graphs with inconsistent structure.

While the recent methods [8, 47, 49, 54] successfully tackle the bias towards more frequently occurring labels, this paper studies another type of bias related to label informativeness. It also manifests itself in missing annotations and has not been addressed in SGG before. In particular, we hypothesize that certain relation labels (implicit labels) are more informative than others (explicit labels) and training on implicit labels improves a model’s ability to reason over complex correlations in visual and textual data.

Our key intuition comes from prior computational and cognitive models [32] and recent work [10] that categorize relations into explicit (or spatial) and implicit, based on whether the relation defines the relative spatial configuration between the two objects implicitly or explicitly (e.g., *man standing on beach* vs. *man on beach* in Fig. 1a). Explicit relations are often easy to learn, e.g. from the spatial coordinates of subject–object pairs, thanks to their highly deterministic spatial arrangements, while implicit ones are often challenging due to the relative spatial variation and require deliberate reasoning. To test our hypothesis, we conduct experiments where we train a SGG model either only on explicit, or only implicit relations, and evaluate them on a test set including both types (i.e. zero-shot implicit or explicit relation classification), see Fig. 1b. Surprisingly, training only on implicit relations obtains good performance not only over implicit ones but also unseen explicit ones (only 2% lower in average training on explicit relations and 4% lower when trained on all labels), while training only on explicit relations performs poorly on implicit relations (where the performance drops to 0.1% from 24.3%).

In other words, training on implicit labels enables the model to better generalize to unseen explicit labels. However, due to partially annotated training data, many subject-object pairs are only labelled by explicit relations and their implicit relations are missing and obscured by the explicit ones.

Motivated by our analysis, we design a novel model-agnostic training procedure for SGG that jointly extracts more information from partially labeled data by mining the missing implicit labels, trains a SGG model on them and boosts its performance. In particular, our method involves a two stage training pipeline. The first stage trains a SGG model on a subset of training data including only annotated implicit relations, which allows the model to learn rich correlations in the data and encourages it to predict more informative implicit labels in the next stage. The second stage includes an alternating procedure that imputes missing implicit labels on the subset of samples annotated with explicit relations, followed by training on both the annotated and imputed labels, called *label refinement*. In this stage, a model is prone to confirm to its own (wrong) predictions to achieve a lower loss as observed in semi-supervised learning (e.g. [3, 51]). To prevent such overfitting, we regularize the model by a latent space augmentation strategy. We demonstrate that our method yields significant performance gains in the SGG task for the standard and zero-shot settings on the Visual Genome [26] when applied to several existing scene graph generation models.

In short, our contributions are as following. We identify a previously unexplored issue, missing informative labels in the standard SGG benchmark and address this through a model agnostic training procedure based on alternating label imputation and model training with effective regularization strategies. This method can be incorporated into state-of-the-art SGG models and boosts their performance by a significant margin.

2. Related work

Scene Graph Generation. SGG has been extensively studied in the past few years with the goal of better understanding the object relations in an image by either focusing on architecture designs [7, 29, 50, 56, 61, 65] or feature fusion methods [14, 19, 23, 28, 58, 63, 64, 68]. Recently, Tang et al. [49, 50] reported that the performance gains from these methods largely come from improved performance only on the head classes (frequently occurring relations) while the performance on most other relations is poor. They propose replacing the biased evaluation metric Recall@k with mean-Recall@k to assign equal importance to all labels.

The same authors [49] report that the bias in the data often drives SGG models to predict frequent labels and propose to use counterfactual causality i.e. measure the difference in predictions between the original scene and a counterfactual one to remove the effect of context bias and focus on the main visual effects of the relation. Chiou et al. [8] address the bias in scene graph generation by learning from positive and unlabeled object pairs. Suhail et al. [47] propose an energy based loss that learns the joint likelihood of object and relations instead of learning them individually. This helps to incorporate commonsense structure (man *riding horse* and man *feeding horse* occurring together are highly improbable) and in better context aggregation. Unlike [47, 49], our focus is on extracting more information from the training data through mining informative labels. In fact, we show that our model is orthogonal to theirs and boosts performance when incorporated to theirs. The most similar to ours, Wang et al. [54] propose a semi-supervised method that employs two deep networks, where the auxiliary one imputes missing labels of unlabeled pairs and self-trains on them and transfers its knowledge to the main network. Unlike [54], who treat all the labels equally, our method only imputes informative implicit labels. This is
crucial, as shown in Tab. 3, because, without such consideration, imputing labels cannot extract any substantial information from unlabeled samples leading to only minor gains. In addition, our framework is more efficient as it involves only a single network that jointly infers labels and trains on them, outperforming [54] significantly.

**Label Completion.** There is a rich body of work in the literature that focuses on learning from partial/missing labels in a multi-label learning setting where each image is labelled for multiple categories with some missing labels [5,6,9,15,21,35]. Common strategies to address this can be divided into two categories: 1) graph based methods [25,55] that exploit similarity between samples to predict missing labels, and 2) low rank matrix completion which extracts label correlations [6,15,57,60] to complete missing labels. There is another setting in which some instances miss all the labels, also called semi-supervised learning in multi-label classification [48,69]. In this setting, the classifier is trained for unseen data. While related to our setting, we look at the classification of relations conditioned on subject–object pairs (rather than on the image level), with each pair already labeled with one relation (rather than unlabeled images). Finally, we group the label set in two groups and treat them asymmetrically in our training.

**Semi-supervised learning.** Semi-supervised learning methods exploit unlabeled data via either pseudo-labeling or imputation with small amounts of labeled data [42,44] or by enforcing consistency regularization on the unlabeled data to produce consistent predictions over various perturbations of the same input [51,52] by applying several augmentation strategies such as Mixup [66], RandAugment [12], AutoAugment [11] or combine both pseudolabeling and consistency regularization [4,46]. Inspired from pseudo-labeling in semi-supervised learning, the main motivation of our method is to impute informative missing labels to improve generalization and learn complex features by relying on partially labeled data and still predict more accurate labels on the biased test set.

### 3. Methodology

#### 3.1. Revisiting SGG Pipeline

In SGG, we seek to localize and classify the objects in an image followed by labeling the visual relations between each pair of objects (or subject and object). Concretely, let $C$ and $P$ denote the object and relation classes respectively. Each subject or object $c = (e^b, e^c) \in \mathcal{E}$ consists of a bounding box $e^b \in \mathbb{R}^4$ and a class label $e^c \in C$. A relation tuple is a triplet of the form $r = (s, p, o)$ where the subject $s$ and the object $o (s, o \in \mathcal{E})$ are joined by the relation $p \in P$, e.g., *man wearing shirt*. Given an image $I$, we can then use a set of objects $E = \{e_i\}_{i=1}^m$ and a set of relations $R = \{r_j\}_{j=1}^n$, where $m$ and $n$ are the number of subject/objects and relation triplets in an image respectively, to define a scene graph $S = (E, R)$. A scene graph can also be written as a combination of a set of bounding boxes $B = \{e_i^b\}_{i=1}^m$, a set of class labels $Y = \{e_i^c\}_{i=1}^m$ and a set of relations $R$.

The SGG models can be decomposed as:

$$P(S|I) = P(B|I) P(Y|B, I) P(R|B, Y, I)$$  \hspace{1cm} (1)$$

where $P(B|I)$ is the object detector or bounding box prediction model, $P(Y|B, I)$ is an object class model and $P(R|B, Y, I)$ is a relation prediction model.

Existing methods [24,27,33,50,56,65,67,70] often employ a two-step process for the scene graph generation task. First, bounding-box proposals ($P(B|I)$) with class predictions and confidence scores ($P(Y|B, I)$) are extracted using off-the-shelf object detectors [17,41]. Then, a multimodal feature fusion model combines visual, language and spatial features to predict the relation for a given subject-object pair ($P(R|B, Y, I)$). Several methods adopt BiLSTMs [65], Bi-TreeLSTMs [50] or fully connected layers [24,67] to encode the co-occurrence between object pairs for relation prediction.

#### 3.2. Missing Relation Labels

Many visual relations are hypernyms, hyponyms, or synonyms [40,59] and hence are non-mutually exclusive. The standard SGG datasets (e.g. Visual Genome [26]) ignore this fact and only assume one annotated label per subject–object pair. Which one is assumed strongly depends on the annotator (manifesting as labeling or language/reporting bias [38]).

One way to circumvent this problem is to collect multiple labels for each triplet, which is however expensive and time consuming. Another potential solution is to use linguistic sources such as WordNet [37] or VerbNet [43] to automatically obtain the missing labels by exploiting the linguistic dependencies between relations. However, this is not trivial, as some of the relation and spatial vocabulary in the SGG datasets are not included in WordNet. Moreover, the context of relations in these language resources does not always allow the right inferences. For instance, in WordNet, *person riding horse* does not imply *person on horse* (no hyponymy relation), but this is the visual implication in the SGG datasets.

While one can use existing methods [8,54] to infer the missing labels, the estimated labels can be noisy and uninformative such that re-training a model on them may not improve the generalization performance. Here, inspired by previous work [10], we propose to group visual relations into two sets: explicit and implicit. Explicit relations encode spatial information between two objects such as ‘on’,
‘in front of’ or ‘under’ and are typically easy to learn, even only based on subject–object locations [10]. The implicit ones are normally verbs such as ‘riding’, ‘walking’ and for learning them the model has to find complex correlations in visual and textual data. In existing SGG datasets, some object pairs are annotated with implicit labels, while other pairs are labeled only with explicit ones and their implicit labels are missing. We propose to divide the set of predicate labels \( P \) into two sets, explicit and implicit and denote them by \( E \) and \( I \) respectively.

### 3.3. Proposed Method

In this section, we explain our proposed method for training the relation classifier \( f_o \) to implement \( P(R|B,Y,I) \). For each image \( I \), we assume that an object detector provides a set of candidate subject–object pairs \( \{(s−o)\} \) and each pair is represented by a \( d \)-dimensional joint embedding \( x \in \mathbb{R}^d \) including its visual, semantic and spatial features. Note that we apply our method to various existing SGG models which uses different object detector and joint embedding functions, and we provide these details in Sec. 4.

In particular, \( f_o \) is instantiated as a deep neural network parameterized by \( \theta \), takes in a joint feature embedding \( x \) for a subject–object pair \( s−o \) and outputs a softmax probability over \(|P|\) relations, i.e. \( f_o(x) : \mathbb{R}^d \rightarrow \mathbb{R}^{|P|} \). Our goal is to learn a relation classifier \( f_o \) that can correctly estimate the relation label of a subject–object pair in an unseen image.

Given a training set \( D \) with \(|D|\) samples, each including tuples of subject–object pairs \( s−o \) along with the relation label \( p \) and the joint feature embedding \( x \), which we denote with \( X = \{(s,p,o,x)\}^{|X|}_{i=1} \) with \(|X|\) tuples. We formulate the learning problem as minimization of two loss terms:

\[
\min_{\theta} \frac{1}{|D|} \sum_{i=1}^{|D|} \left( \mathcal{L}^{I}(X_i;\theta) + \mathcal{L}^{E}(X_i;\theta) \right) \tag{2}
\]

where \( \mathcal{L}^{I}(X_i;\theta) \) and \( \mathcal{L}^{E}(X_i;\theta) \) are the loss terms defined over implicit and explicit relations respectively.

For a given image \( I \) and its tuple \( X \), we pick the tuples whose relation is annotated only with implicit relation label (i.e. \( X^I = \{(s,p,o,x) | p \in \mathbb{I}\} \) ) and define the implicit loss term as:

\[
\mathcal{L}^{I}(X^I;\theta) = \frac{1}{|X^I|} \sum_{(s,p,o,x) \in X^I} \mathcal{L}_{CE}(f_o(x),p) \tag{3}
\]

where \( \mathcal{L}_{CE} \) is the cross-entropy loss. In other words, for the implicit relations, we follow the standard practice and compute its loss by using its ground-truth implicit relation label \( p \) which is an one-hot vector, as each subject–object pair is annotated with only one label.

Similarly, we formulate the explicit term \( \mathcal{L}^{E}(X^E;\theta) \) for the tuples with explicit labels:

\[
\mathcal{L}^{E}(X^E;\theta) = \frac{1}{|X^E|} \sum_{(s,p,o,x) \in X^E} \mathcal{L}_{KL}(f_o(x),\hat{p}) \tag{4}
\]

where \( \mathcal{L}_{KL} \) is the Kullback-Leibler divergence, \( X^E = \{(s,p,o,x) | p \in \mathbb{E}\} \) and \( \hat{p} \) is the imputed relation label for the subject–object pair, which is a vector with soft probabilities. Next, we discuss our method to obtain \( \hat{p} \).

#### Label imputation

The subject–object pairs that are annotated with explicit relations only can often be labelled with more informative implicit relation labels. For instance, the ground truth label may be \textit{person beside table}, but \textit{person eating at table} might also be correct in this case, and more informative. To impute the missing implicit relation for a subject–object pair, which is originally annotated with an explicit relation label, we follow a two-step procedure.

First, we take each subject–object pair annotated with an explicit label from \( X^E \) along with its joint embedding and impute its implicit label through the relation classifier \( f_o \) as:

\[
\hat{p} = \arg \max_{i \in \mathbb{I}} \left( \frac{\exp(f_o(x))}{\sum_{j \in \mathbb{E}} \exp(f_o(x))} \right) \tag{5}
\]

where \( f_i \) denotes its logit for the \( i \)-th relation class. In words, we compute softmax probabilities only over implicit relation labels and pick the highest scoring implicit label to obtain an one-hot vector \( \hat{p} \). Note that one can also obtain a soft probability vector over all implicit label classes, however, we empirically show that the former works better.

Second, as the subject–object pair is originally labelled with an explicit label \( p \), we also use this information and average the imputed label \( \hat{p} \) with its original label \( p \) as:

\[
\hat{p} = (p + \hat{p})/2. \tag{6}
\]
We call this step as **Label Refinement**. As \( \hat{p} \) includes equal probabilities (i.e., 0.5) for each of the explicit and implicit label, it is not an one-hot vector. Thus, we use KL divergence Eq. (4) to encourage the model to predict both the labels. Compared to standard cross-entropy loss, the KL divergence loss increases the model entropy by reducing over-confidence, resulting in smoother predictions. Most of the traditional methods for SGG trained with cross-entropy may get confused by inconsistent annotations, where same relation is labeled with less informative spatial relation in some images while more informative labels are used in some other images. Our loss function formulation and multilabel nature of the targets addresses this inconsistency, unlike previous work [47,50].

**Latent space augmentation.** Our training pipeline is illustrated in Fig. 2 which follows an alternating optimization and consists of two alternating steps where we employ the relation classifier \( f_\theta \) to impute the implicit labels and simultaneously optimize the classifier parameters \( \theta \). The main challenge here is that the model parameters can overfit to its own imputed labels quickly, resulting in a local optimum solution. This problem is notoriously known as confirmation bias that also occurs in many semi-supervised problems [3,46]. To prevent overfitting to the wrong imputed labels, existing solutions include applying various kinds of data augmentation including standard geometric and color transformations [13], their combinations [11,12,62] and also generating augmented version of samples [53,66].

As many SGG methods build on the feature space of an object detector, many augmentation strategies that are applied on raw pixels are not applicable to our case. Hence, we use Manifold Mixup [53] that generates augmented embeddings (in the manifold space rather than in the pixel space) by taking a convex combination of different pairs of embeddings (\( x \) and \( x' \)) and also their labels (\( p \) and \( p' \)):

\[
\tilde{x} = \lambda x + (1 - \lambda)x', \quad \tilde{p} = \lambda p + (1 - \lambda)p'
\]

where \( \lambda \) is sampled from a beta distribution, i.e. \( \lambda \sim \text{Beta}(\alpha, \alpha) \) with \( \alpha \) as a hyperparameter. Note that we apply this augmentation to the whole training set and allow mixing embeddings across samples from both the implicit and explicit set of relations. This augmentation acts as a regularizer and accounts for overfitting to the incorrect imputed labels while training.

### 3.4. Algorithm

In Algorithm 1, we detail our training pipeline. To obtain the initial parameters \( \theta_0 \), we first train our model on the tuples with only implicit labels as following (Line 2):

\[
\theta_0 = \arg\min_\theta \frac{1}{|D|} \sum_{i=1}^{|D|} L^I(X_i; \theta). \tag{8}
\]

The key intuition behind this is that model learned on only the implicit relations are more likely to produce confident predictions over implicit labels and hence not ‘distracted’ by explicit relation labels.

After the training on implicit relations, we iteratively impute implicit relation labels for the subject-object pairs annotated with the explicit relations (Line 5) and update the model parameters using Eq. (2) (Line 7). The model parameters optimized in Eq. (2) take as input the augmented versions of the sample and label pair (as in Eq. (7)) for both the implicit and explicit set of relation labels (Line 6).

**Algorithm 1** Our proposed optimization of the SGG model

1. **Input:** Training set \( D \) with \(|D|\) samples, each including a set of tuples \( X \) with \((s, p, o, x)\) with subject, object and relation label, joint embedding resp., Explicit and implicit relation sets \( E \) and \( I \) resp., relation classifier \( f_\theta \), \( T \) number of iterations, \( \eta \) learning rate.
2. **Initialize** \( \theta \) as in Eq. (8)
3. **for** \( t = 0, \ldots, T \) **do**
4. **Sample** a minibatch \( B = (X_1, \ldots, X_{|B|}) \sim D \)
5. **Impute** \( \hat{p} \): Impute implicit labels \( \hat{p}_t \) for \( B^E \) by using Eq. (5) and Eq. (6),
6. **Update** \( \theta \): \( \theta_{t+1} \leftarrow \theta_t + \eta \Delta_\theta \) where \( \Delta_\theta \) is the update for \( \theta \) obtained from Eq. (2),
7. **end for**
8. **return** \( \theta \)

### 4. Experiments

**Dataset and Evaluation Settings.** We evaluate our proposed method for scene graph generation on the Visual Genome (VG) [26] dataset. We use the pre-processed version of the VG dataset as proposed in [56]. The dataset consists of 108k images with 150 object categories and 50 relation categories. The training, test and validation split used in the experiments also follow previous work [47,49,56].

For evaluation on the Visual Genome dataset, we follow [56] and report performance on three settings: (1) **Predicate Classification (PredCls).** This task measures the accuracy of relation (also termed as predicate in literature) prediction when the ground truth object classes and boxes are given. It is not affected by the object detector accuracy. (2) **Scene Graph Classification (SGCls).** In this setting, we know the ground truth boxes and we have to predict the object classes and the relations between them. (3) **Scene Graph Detection (SGDet).** This is the most challenging setting and the models are used to predict object bounding boxes, object classes and the relations between them. We measure Mean Recall@K (mR@K) [49,50] to evaluate scene graph generation models. More recent work has
Scene Graph Classification

<table>
<thead>
<tr>
<th>Models</th>
<th>Method</th>
<th>Predicate Classification</th>
<th>Scene Graph Classification</th>
<th>Scene Graph Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mR@20</td>
<td>mR@50</td>
<td>mR@100</td>
</tr>
<tr>
<td>IMP [5, 56]</td>
<td>-</td>
<td>-</td>
<td>9.8</td>
<td>10.5</td>
</tr>
<tr>
<td>FREQ [50, 65]</td>
<td>-</td>
<td>8.3</td>
<td>13.0</td>
<td>16.0</td>
</tr>
<tr>
<td>Motifs [65]</td>
<td>-</td>
<td>10.8</td>
<td>14.0</td>
<td>15.3</td>
</tr>
<tr>
<td>KERN [7]</td>
<td>-</td>
<td>14.0</td>
<td>17.7</td>
<td>19.2</td>
</tr>
<tr>
<td>VCTree [50]</td>
<td>-</td>
<td>14.0</td>
<td>17.9</td>
<td>19.4</td>
</tr>
<tr>
<td>VCTree-TDE [54]</td>
<td>-</td>
<td>14.4</td>
<td>18.4</td>
<td>20.0</td>
</tr>
<tr>
<td>IMP [56]</td>
<td>Baseline</td>
<td>8.9</td>
<td>11.0</td>
<td>11.8</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>12.3</td>
<td>14.6</td>
<td>15.3</td>
</tr>
<tr>
<td>Motif-TDE-Sum [49, 65]</td>
<td>Baseline</td>
<td>17.9</td>
<td>24.8</td>
<td>28.7</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>21.3</td>
<td>27.1</td>
<td>29.7</td>
</tr>
<tr>
<td>VCTree [50]</td>
<td>Baseline</td>
<td>13.1</td>
<td>16.5</td>
<td>17.8</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>18.0</td>
<td>21.7</td>
<td>23.1</td>
</tr>
<tr>
<td>VCTree-EBM [47]</td>
<td>Baseline</td>
<td>14.2</td>
<td>18.2</td>
<td>19.7</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>21.0</td>
<td>24.9</td>
<td>26.5</td>
</tr>
<tr>
<td>VCTree-TDE [49]</td>
<td>Baseline</td>
<td>16.3</td>
<td>22.9</td>
<td>26.3</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>22.2</td>
<td>28.1</td>
<td>30.6</td>
</tr>
</tbody>
</table>

Table 1. Scene Graph Generation performance comparison on mean Recall@K [49] under all three experimental settings. We compare the results of our proposed framework (Ours) with the original model (Baseline) using different SGG architectures.

More specifically, the explicit relations are above, across, against, along, at, behind, between, in, in front of, near, on, over, under and the rest are treated as implicit relations.

**Model Generalization.** Our proposed framework has the flexibility to be trained with any scene graph generation model. Hence, we train with different model architectures to demonstrate the generalizability of our approach: Iterative Message Passing (IMP) [56], Neural-Motifs [65] and VCTree [50]. We also train with two other debiasing methods that build upon these models, Energy-based Modeling (EBM) [47], where the authors propose to train with an additional energy-based loss and Total Direct Effect (TDE), where counterfactual reasoning is used during inference.

**Explicit and Implicit Relation Labels.** For the Visual Genome dataset [26], Xu et al. [56] released a version of the dataset with 50 relations which are: above, across, against, along, and, at, attached to, behind, belonging to, between, carrying, covered in, covering, eating, flying in, for, from, growing on, hanging from, has, holding, in, in front of, laying on, looking at, lying on, made of, mounted on, near, of, on, on back of, over, painted on, parked on, part of, playing, riding, says, sitting on, standing on, to, under, using, walking in, walking on, watching, wearing, wears, with.

Inspired by [10], we divide the relation label space into a set of explicit and implicit relations. We define explicit relations when the spatial arrangement of objects are implied by the label itself e.g., “on”, “below”, “next to” and so on. For implicit relations, the spatial arrangement of objects is only indirectly implied, “riding”, “walking”, “holding” etc.

5. Results

**Quantitative Results.** Table 1 compares the performance of the state-of-the-art methods when trained with our proposed training framework on the Visual Genome dataset [26]. Our method consistently improves performance on all the three evaluation settings (PredCls, SGCls, SGDet) when trained with existing methods. With IMP [56] and Motif-TDE-Sum [49, 65], we obtain an absolute improve-

4https://groups.inf.ed.ac.uk/vico/research/NARE
In this section, we study different components of our method separately to validate their effectiveness. Table 3 evaluates the contribution of our proposed components. The results in the rows for the method Baseline show the effect of training with implicit relations on scene graph generation performance. If the model is trained with the explicit relations only, mean recall drops. There is also a marginal drop in performance compared to training with all relations when trained on a random subset of the relations. Even training on a subset of the data with only implicit relations, we can achieve at-par performance compared to training with all relations.

We also compare the results under settings when the model is trained on a particular subset of Train Label with different imputation strategies (results in rows with the method Ours). The relation labels are imputed on the holdout relation set with the top-1 (best) in the set of labels that it was trained on. Training and imputing with the random or explicit set of labels decreases performance compared to the baseline, with a significant drop when only explicit relations are considered. This shows the importance of learning with and imputing implicit relations: they provide useful information about interactions between object pairs not captured by explicit relations.

In Table 4, we study whether our method improves over the baselines mostly due to training with Latent Space Augmentation (Manifold Mixup) or combination of multiple components. Applying manifold mixup on either the baseline (row 2) or with only explicit ground-truth labels without imputation (row 4), does not provide significant gains. This indicates that mixup is effective only when applied to the imputed implicit labels. This hints that mixup can reduce overfitting to noisy imputations and allow for imputing more informative labels. We also show that the performance of soft imputed labels (row 7) is very similar to our proposed method (with hard labels, row 8). Using hard labels reduces noise in the predictions and encourages the model to learn from more high-confident predictions.

In Table 5, we show the results for training on relation subsets and our final method on the explicit and implicit set of relations separately. As discussed in Section 1, when the model is only trained on explicit relations, it fails to generalize to implicit relations. This is in contrast to training only on the implicit set of relations. This indicates that implicit relations are rich in information and perhaps learn complex and generalizable features. Our final method outperforms training on all relations (original model) and the subset of relations by a significant margin, showing the strength of mining informative labels for less informative samples.
Train Relations
Explicit
24.83/27.80
Implicit
20.10/22.89
22.89/25.34
24.47/28.79

Method localizes the baseline. For instance, in the bottom-left image, our method correctly identifies triplets with relations that were missed in the baseline, in relations such as sitting on, walking on, holding elephant of paper.

Moreover, our method also effectively identifies triplets with relations that were missed in the baseline. For instance, in the bottom-left image, our method localizes man holding paper correctly. Our method also corrects relations which are incorrectly predicted in the baseline, woman watching elephant as opposed to woman on elephant in the bottom-right image.

In Figure 4, we visualize the imputed implicit relations for the triplets annotated with explicit relations. In orange, we show the ground-truth scene graphs and the corresponding imputed scene graphs in green. It can be clearly observed from the ground-truth scene graphs that there is annotator bias towards spatial relations. Our label imputation strategy is able to find alternate and missing implicit relations for these triplets and exploit them during training. For instance, our method imputes important relations such as attached to, hanging from, sitting on which are more descriptive than their explicit counterpart on. This shows the importance of label imputation and generating descriptive scene graphs for comprehensive scene understanding.

**6. Conclusion**

We proposed a novel model-agnostic training framework for scene graph generation. We introduced the concept of label informativeness, which had not been explored in SGG before. A model trained on informative relations is able to model the visual and textual context better compared to training on simple spatial relations. We showed how to impute informative relations from the partially labeled data and jointly train with imputed and ground truth relations. We improved the performance across models and tasks, including in a zero-shot setting. One limitation of our approach is its limited ability in predicting relations with very few samples, which should be investigated in future work.

**Acknowledgement:** AG is supported by the Armeane Choksi Scholarship and HB is supported by the EPSRC programme grant Visual AI EP/T028572/1. This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-016). We thank the anonymous reviewers for their constructive feedback.
References


[40] Vignesh Ramanathan, Congcong Li, Jia Deng, Wei Han, Zhen Li, Kunlong Gu, Yang Song, Samy Bengio, Charles Rosenberg, and Li Fei-Fei. Learning semantic relationships for better action retrieval in images. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1100–1109, 2015. 3


