UNIVERSITY OF CALIFORNIA SAN DIEGO

Devil is in the Tails: Visual Relationship Recognition

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Electrical Engineering (Signal and Image Processing)

by

Alakh Himanshu Desai

Committee in charge:

Professor Nuno Vasconcelos, Chair
Professor Xiaolong Wang
Professor Pengtao Xie

2022

The Thesis of Alakh Himanshu Desai is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my lovely family who supported me and all the friends I made on this journey.

EPIGRAPH

Vision is the art of seeing
what is invisible to others.

Jonathan Swift

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Nuno Vasconcelos for his support as my research advisor. Through various ups and downs, his guidance has proved to be invaluable.

I would also like to acknowledge Subarna Tripathi and Gina Wu, for guiding and helping me throughout the research process. Their unwavering support helped me work learn and grow as a researcher.

Chapter 2, in part, contains material from A. Desai, TY Wu, S Tripathi, N Vasconcelos,"Learning of Visual Relations:The Devil is in the Tails", 2021 IEEE International Conference on Computer Vision (ICCV), Montreal, Canada, 2021. The thesis author was the primary investigator and author of this paper.

Chapter 3, in part is currently being prepared for submission for publication by A. Desai, TY Wu, S Tripathi, N Vasconcelos. The thesis author was the primary investigator and author of this material.

VITA

2019	*(Honors)* Bachelor of Technology in Electronics and Communication Engineering, International Institute of Information Technology Hyderabad
2020-2021	Graduate Teaching Assistant, Department of Electrical and Computer Engineering, University of California San Diego
2022-2022	Graduate Student Researcher, University of California San Diego
2022	Master of Science in Electrical Engineering (Signal and Image Processing), University of California San Diego

PUBLICATIONS

A. Desai, TY Wu, S Tripathi, N Vasconcelos,"Learning of Visual Relations:The Devil is in the Tails", 2021 IEEE International Conference on Computer Vision (ICCV), Montreal, Canada, 2021.

A. Desai, R. Chauhan and J. Sivaswamy, "Image Segmentation Using Hybrid Representations", 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), Iowa City, IA, USA, 2020.

D. J. Gaddipati, A. Desai, J. Sivaswamy and K. A. Vermeer, "Glaucoma Assessment from OCT images using Capsule Network", 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 2019.

ABSTRACT OF THE THESIS

Devil is in the Tails: Visual Relationship Recognition

by

Alakh Himanshu Desai

Master of Science in Electrical Engineering (Signal and Image Processing)

University of California San Diego, 2022

Professor Nuno Vasconcelos, Chair

Significant effort has been recently devoted to modelling visual relations. This has mostly addressed the design of architectures, typically by adding parameters and increasing model complexity. However, visual relation learning is a long-tailed problem, due to the combinatorial nature of joint reasoning about groups of objects. Increasing model complexity is, in general, ill-suited for long-tailed problems due to their tendency to over-fit. In this thesis, we explore an alternative hypothesis, denoted the Devil is in the Tails. Under this hypothesis, better performance is achieved by keeping the model simple but improving its ability to cope with long-tailed distributions. To test this hypothesis, we devise a new approach for training visual relationships models. This is based on an iterative decoupled training scheme, denoted

Decoupled Training for Devil in the Tails (DT2). DT2 employs a novel sampling approach, Alternating Class-Balanced Sampling (ACBS), to capture the interplay between the long-tailed entity and predicate distributions of visual relations. Results show that, with extremely simple architecture, DT2-ACBS significantly outperforms much more complex state-of-the-art methods on scene graph generation tasks. This suggests that the development of sophisticated models must be considered in tandem with the long-tailed nature of the problem.

Chapter 1

Scene Graph Generation

1.1 Introduction

Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding[2]. As with most computational tasks, we have attempted to achieve it by subdividing it into smaller tasks that can be solved with geometry, rules, or representation learning. There are several sub tasks in which deep learning models have surpassed the human visual system. These, however, belong to the category of perceptual tasks, such as image classification. In cognitive tasks, such as image description and question answering, however, computers lag far behind. ”Cognition is core to tasks that involve not just recognizing, but reasoning about our visual world.”[24] To achieve this, models must not only localize and detect the key objects in an image, but must also learn to understand the relationships between the detected objects. While deep learning models excel in learning representations for real world objects, the understanding of a scene is deeply embedded in the local and global relationships between the objects that constitute it. Therefore, understanding the relations in a scene is a major step in the direction of cognition which evades present day models. For example, “When asked *What vehicle is the person riding?*, computers will need to identify the objects in an image as well as the relationships riding(man, carriage) and pulling(horse, carriage) in order to answer correctly that *the person is riding a horse-drawn*

carriage."[24] To solve such cognitive tasks we can employ scene graphs.

A scene graph is a structured representation of an image, where nodes in a scene graph correspond to object bounding boxes and their categories; and the directed edges correspond to the pairwise relationships between objects. Scene graphs were first proposed by [22] as a data structure that describes the object instances in a scene and the relationships between these objects. A complete scene graph can represent the detailed semantics of a dataset of scenes, but not a single image or a video; moreover, it contains powerful representations that encode 2D/3D images [22], [1] and videos [31], [36] into their abstract semantic elements without restricting either the types and attributes of objects or the relationships between objects. Related research into scene graphs greatly promotes the understanding of various tasks such as vision, natural language, and their cross-domains.

The task of Scene Graph Generation is to generate a visually-grounded scene graph that most accurately correlates with an image. In the literature, this problem is formulated as a $< subject - predicate - object >$ tuple, wherein the subject and object are two of the detected objects in the scene and the "predicate" is the relationship that the "subject" imposes upon the "object". This formulation allows us to answer questions belonging to the "How is object A related to object B?" category or "What object is the object A *sitting* on?". Therefore, a graph containing as nodes, all the objects in a scene and as edges, the relationship between each pair of objects allows for a much higher level of cognition than a bag of objects approach. Scene graph generation is therefore a step in the direction of image understanding. Another aspect of scene graphs is that they create a bridge between the two most important modalities of cognition in general, visual information and natural language. A model that aims to learn the relationship between two image patches (each corresponding to a crop of the object), must in turn learn to understand this relation subject to the two labels of the detected objects. Therefore, the model must learn both visual relations and label relations as it generates the tuples for the scene.

1.2 Formulation

The inference of the visual relationships in a scene is usually formulated as a three stage process. The objects/entities in the scene are detected, classified, and the relationships between each pair of entities, in the form of predicates, are finally inferred. [21] formulated these stages with a *Scene Graph*. Let C and P be the set of entity and predicate classes, respectively. Each entity $e = (e^b, e^c) \in \mathscr{E}$ is composed by a bounding box $e^b \in R^4$ and a class label $e^c \in C$. A relation $r = (s, p, o)$ is a three-tuple, connecting a subject s and an object o identities ($s, o \in \mathscr{E}$), through a predicate $p \in P$. For example, *person-riding-bike* or *donkey-on-road*. The scene graph $G = (E, R)$ of an image I contains a set of entities $E = \{e_i\}_{i=1}^m$ and a set of relations $R = \{r_j\}_{j=1}^n$ extracted from the image. This can be further decomposed into 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 generation of a scene graph G from an image I is naturally mapped into the probabilistic model

$$Pr(G|I) = Pr(B|I)Pr(Y|B,I)Pr(R|B,Y,I), \tag{1.1}$$

where $Pr(B|I)$ is a bounding box prediction model, $Pr(Y|B,I)$ an entity class model and $Pr(R|B,Y,I)$ is a predicate class model.

As bounding box prediction has been widely studied in object detection [32], it is possible to simply adopt an off-the-shelf detector.

The literature divides the SGG task into three sub tasks based on the joint inference of the probabilistic model:

- **Predicate Classification (PredCls):** For this task the input for the model is the bounding boxes of the objects and their corresponding labels. The model must predict the predicate that best defines the relationship between the two objects.
- **Scene Graph Classification (SGCls):** For this task the input for the model is the bounding

boxes of the objects. The labels for the objects are not provided. The model must predict the label for each object and the predicate that best defines the relationship between them.

- **Scene Graph Detection (SGDet):** For this task the input for the model is the raw image. The model must localize the objects of interest, classify them and predict the predicate that best defines the relationship between them.

Now that we have understood what scene graph generation is, we can look into its applications and the inherent challenges associated with it.

1.3 Applications of Scene Graph Generation

Scene graphs provide a data structure that can be used to extract visual and textual information about a scene. This forms a useful representation for scene understanding tasks, that can now query the scene graph to answer semantic questions about the image. Therefore many cognition tasks, are essentially applications of scene graph generation.

1.3.1 Image generation

Apart from providing the relations between the various entities in an image, scene graphs also provide a relationship between the textual and visual modalities, that is, they provide a connection between the description of an image and the image itself. This feature of scene graphs can be exploited when asking a model to generate an image based on a description. While current models can generate impressive images for simple text descriptions, they fail to deliver when the complexity of the description increases. Scene graphs can simplify this problem, by breaking it down into simpler $< subject - predicate - object >$ tuples that the model can now use to generate the image piece-wise.

1.3.2 Image or video captioning

Image captioning methods that use RNN or LSTM based natural Language reasoning models to generate the natural language description or caption of the image cannot model the semantic relationships between the objects well. Therefore, the descriptions generated by such models are inaccurate. Scene graphs can help with alleviating these issues by capturing the semantic relationship between the objects in the image.

1.3.3 Cross-Modal Retrieval

Cross-Modal Retrieval is used for implementing a retrieval task across different modalities. such as image-text, video-text, and audio-text. Cross modal retrieval aims to find a common representation space, in which the samples from different modalities can be compared directly. The key to image-text cross-modal retrieval concerns learning a comprehensive and unified representation to represent the multi-modal data and scene graphs are the ideal choice in this context.

1.3.4 Visual Question Answering

VQA is also a multi-modal feature learning task. Scene graphs can extract the important information from a scene in the form of a graph, which enables scene graph-based VQA methods outperform traditional algorithms. With the help of scene graphs and graph networks that can encode the scene graph one can reason and answer the questions that require a common representation between text and images.

1.3.5 Image Understanding and Reasoning

The task of image understanding and reasoning is a very high level computer vision task. Unlike the low level tasks this cannot be performed by a machine with only pixel level information. This task is one of cognition and requires objects, relationships, attributes and other visual feature components to ”understand” the input image. A scene graph provides two

of the key input features required for this task. The localized and recognized objects and the relationship between each subject-object pair provides valuable information for a model that is attempting to solve the highly complex task of understanding a scene.

1.3.6 3D Scene Understanding

Constructing a 3D scene is very important for modeling complex indoor scenes and extracting useful information about the environment. As with the 2D scene graphs generated from 2D images, a scene graph can also be constructed from 3D scenes as a 3D scene graph, which can provide an accurate representation of the object relationships in 3D scenes. A 3D scene graph succinctly describes the environment by abstracting the objects and their relationships in 3D space in the form of graphs. To construct a 3D scene graph, it is necessary to locate the different objects, identify the elements, attributes, and relationships between the objects in 2D images, and then use all of this information to construct a 3D scene.

1.3.7 Human-Object / Human-Human Interaction

A subset of the generalized scene graph generation problem. the human-object interaction problem, is one where the subject is always a human. The human is the central entity in the scene and all the other detected objects are studied in relation to it. This allows the study of how humans interact with the surrounding and recognizing human actions. This task is very similar to scene graph generation but differs in one fundamental way in terms of implementation. While the task of recognizing the relationships between each subject-object pair in an image is $O(n^2)$ problem, the human interaction problem is a linear problem.

1.4 Challenges of Scene Graph Generation

When we consider this approach to Scene Graph Generation, we can observe the challenges that are a part of the problem description itself.

1.4.1 SGG is an ill-posed problem

The relationship between two objects in a scene derived from the real world is neither dis-ambiguous nor distinct. A relationship can be defined in various ways: a man might be *with* a child, *playing with* a child, *running with* a child, *playing soccer* with a child and many more ways, all of which are true albeit from different points of view or with different levels of detail. As the preceding example exhibits, there are multiple valid descriptions of a scene, which makes this an ill-posed problem.

1.4.2 Model Evaluation

Since many valid relations can and do exist between a pair of objects, comparing various models that attempt to solve this task is not a trivial matter. Also, while all the different versions are valid, not all of them are equal. Some interpretations of the relationship are clearly more valuable than others. For example, for an automated surveillance system, ”man 1 behind man 2” and ”man 1 holding man 2 at gunpoint” are two very different scenarios and a model that cannot differentiate between the two is not appropriate despite being correct about the relationship. Model evaluation must therefore consider the level of detail that a model provides over being vaguely correct at all times.

1.4.3 All relationships are not detectable

A pair of bounding boxes and their corresponding labels is not sufficient for a model to satisfactorily detect the true relationship between the objects. For example, two people far apart in an image might be waving at each other, but this would not be detected by a model which cannot consider the larger picture or is looking for local relationships alone. This is a highly non trivial problem to solve and is largely left unaddressed by the literature.

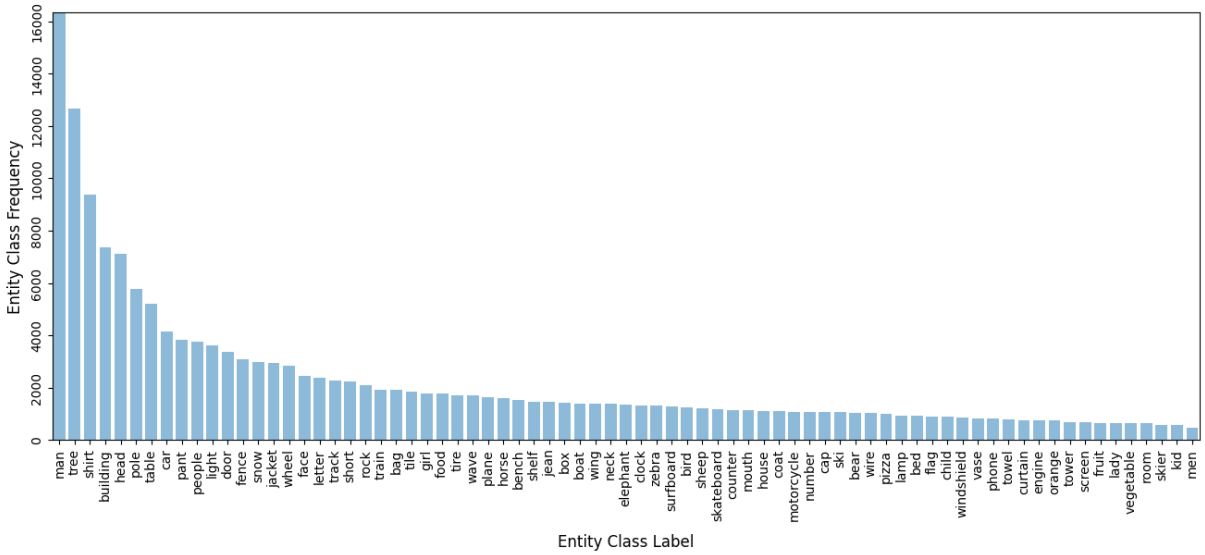

Figure 1.1. Entity Class Distribution

1.4.4 Long tailed nature of the problem

Entity Recognition

Entities in the wild are not class balanced. Datasets that are not curated to be well balanced, reflect this nature of the class distribution. The distribution of entity classes is what is referred to in statistics as a long-tail distribution. ”In ”long-tailed” distributions a high-frequency or high-amplitude population is followed by a low-frequency or low-amplitude population which gradually ”tails off” asymptotically. The events at the far end of the tail have a very low probability of occurrence. As a rule of thumb, for such population distributions the majority of occurrences (more than half, and where the Pareto principle applies, 80%) are accounted for by the first 20% of items in the distribution.” [41] In the case of entity recognition, while some classes dominate the datasets, a majority of the classes have very low representations in the world. This inherent bias results in most models trained on such data, being highly biased towards predicting high frequency, or **head** classes, since they increase the chances of the prediction being correct, and to ignore the low frequency **tail** classes. This would be acceptable if the high frequency classes represented the useful information that we are looking for. However, more

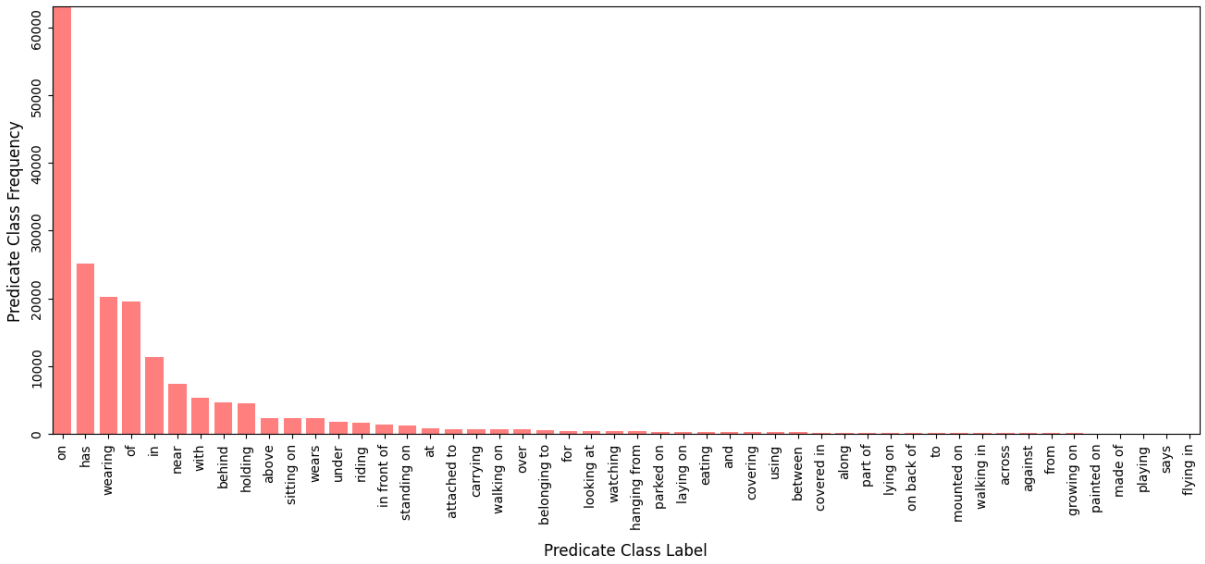

Figure 1.2. Predicate Class Distribution

often than not, the tail classes are much more useful. The long-tailed nature of entity recognition is a well researched topic and has provided valuable models which can learn to predict both the tail and the head classes. However, the fundamental concern with any of the single model methods is that improving the tail class performance for a model, necessarily degrades the head class performance. Most models attempt to keep this drop in performance as low as possible.

Predicate Classification

The long-tailed nature in the Scene Graph Generation task is not due to the long-tailed nature of the entity recognition sub-task alone. Predicate classification not only adds onto the long-tailed distribution, it also exaggerates the problem by being more skewed than the object distribution. Like with entity recognition, the tail end of the predicate distribution holds the most informative classes. A complex model, performing well on the overall data, is therefore not very valuable, because it overfits to the head class data.

Dual long-tailed nature of scene graph generation

The task of generating a scene graph is therefore to learn a distribution that is a joint distribution of two long-tailed distributions. This makes scene graph generation a joint long tailed

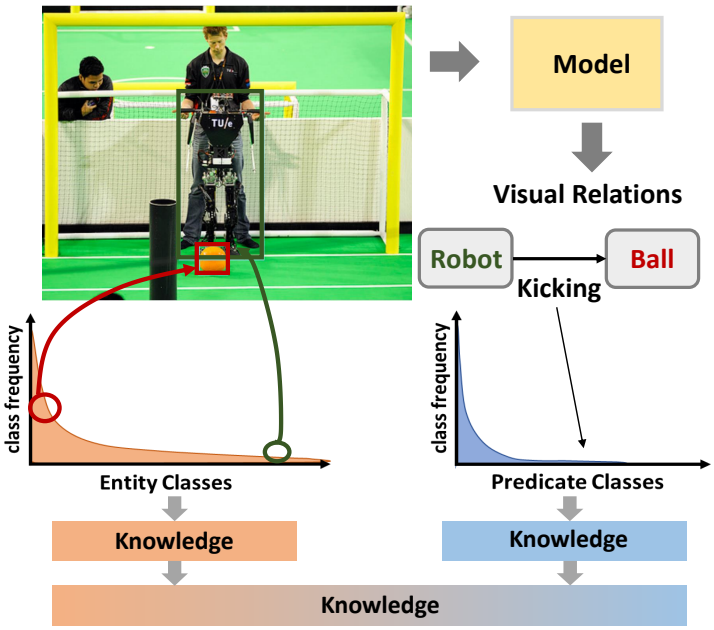

Figure 1.3. The learning process of visual relations need to consider the long-tailed nature of both entity and predicate class distributions.

problem. While long-tailed entity recognition has been addressed in the literature [29, 3, 8, 23], the imbalance becomes more prevalent for the SGG tasks, owing to the severe long-tailed nature of the predicate distribution.

Take Figure 1.3 for example. While the class of the subject ("ball") is popular, the class of the object ("robot") and the predicate ("kicking") can be infrequent, leading to the rare occurrence of the tuple "robot-kicking-ball". This shows that even when the entity class distribution is balanced, the imbalanced predicate class distribution can lead to a more imbalanced tuple distribution. Of course, such imbalance issues can be exacerbated if both entity classes and predicate classes are skewed (e.g. "tripod-mounted-on-donkey"). The combination of long-tailed entity and predicate classes makes SGG a more challenging problem.

While the long-tailed problem poses a great challenge to SGG tasks, it has not been well addressed in the SGG literature. Existing works [50, 47, 6, 34, 51] instead focused on designing more complex models, primarily by adding architectural enhancements that increase model size. While this has enabled encouraging performance under the Recall@k (R@k) metric, this

metric is biased toward the highly populated classes. This suggests that prior works may be over-fitting on popular predicate classes (e.g. *on/has*), but their performances could degrade on the less frequent classes (e.g. *eating/riding*). Such a bias towards the populated classes is problematic, because predicates lying in the tails often provide more informative depictions of scene content. The failure to predict tail classes could lead to a less informative scene graph , limiting the effectiveness of scene graphs for intended applications.

In this thesis, we explore the hypothesis that the Devil is in the tails. Under this hypothesis, visual relation learning is better addressed by a simple model of improved ability to cope with long-tailed distributions. As seen before, both distributions are heavily skewed, but with different magnitude. The imbalance in the predicate distribution is more severe than that in the entity distribution. To address this, we propose a new approach to visual relationship learning, based on a simpler architecture than those in the literature but a more sophisticated training procedure, denoted *Decoupled Training for Devil in the Tails (DT2)*.

Chapter 2

Visual Relationship Learning

2.1 Background and Related Work

2.1.1 Scene graph generation

Several works have addressed the generation of scene graphs for images [48, 46, 49, 17, 42, 45, 50, 47, 26, 12, 6, 34, 19, 51, 10]. Most approaches focus on either sophisticated architecture design or contextual feature fusion strategies, such as message passing and recurrent neural networks [50, 34], to optimize SGG performance on the Visual Genome dataset [24] under the Recall@K metric. While these approaches achieved gains for highly populated classes, underrepresented classes tend to have much poorer performance. Recently, [6, 33, 46, 40, 25] started to address the learning bias induced by the dataset statistics, by using a more suitable evaluation metric, mRecall@K, which averages recall values across classes. To address the dataset bias, TDE [33] employed causal inference in the prediction stage , whereas [40] used a pseudo-siamese network to extract balanced visual features, and PCPL [46] harnessed implicit correlations among predicate classes and used a complex graph encoding module consisting of a number of stacked encoders and attention heads. A concurrent work [25] introduces confidence-based gating with bi-level data resampling to mitigate the training bias. These methods considered, at most, the long-tailed distribution of either predicates or entities and do not disentangle the gains of sampling from those of complex architectures. For example, [46] proposed a contextual feature generator via graph encoding with 6 stacked encoders, each with

12 attention heads and a feed-forward network. We argue that long-tailed distributions should be considered for both entities and predicates and show that, when this is done, better results can be achieved with a much simpler architecture.

2.1.2 Long-tailed recognition

Prior work addresses the long-tailed issue in 3 directions: data re-sampling, cost-sensitive loss and transfer learning.

Data resampling

[15, 13, 54, 14, 11, 5] is a popular strategy to oversample tail (underrepresented) classes and undersample head (populated) classes. Oversampling is achieved either by duplicating samples or by synthesizing data [13, 54, 5]. While producing a more uniform training distribution, recent works [23, 52] argue that this strategy is unsuitable for deep representation learning like CNN. [23] decouples the representation learning from the classifier learning, adopting different sampling strategies in the two stages, whereas [52] proposes a two-stream model with a mixed sampling strategy. The proposed method lies in this direction, since we consider different distributions of entity and predicate classes, and adopt different sampling strategies for training different model components.

Cost-sensitive losses

[9, 8, 3, 28] assign different costs to the incorrect prediction of different samples, according to class frequency [8, 3] or difficulty [9, 28]. This is implemented by assigning higher weights or enforcing larger margins for classes with fewer samples. Weights can be proportional to inverse class frequency or effective number [8] and can be estimated by meta-learning [20]. This re-weighting strategy was recently applied to the scene graph literature [46] to overcome long-tailed distributions.

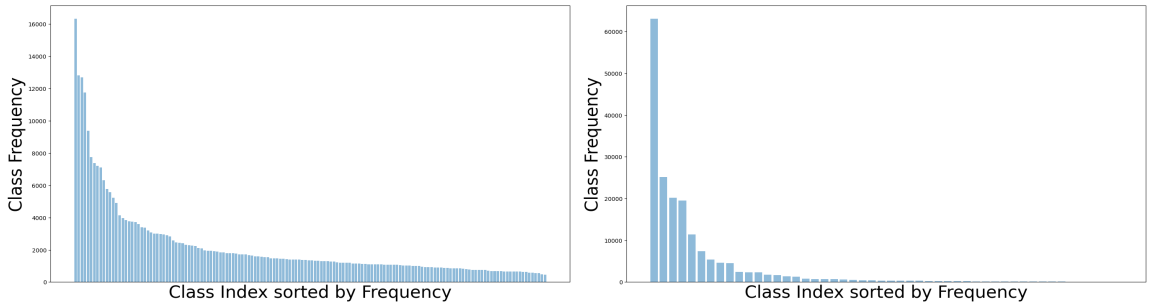

Figure 2.1. Object classes (left) and predicate classes (right) are both long-tailed distributed in Visual Genome (VG150).

Transfer learning

Transfer learning methods transfer information from head to tail classes. [38, 39] learns to predict few-shot model parameters from many-shot model parameters, and [29] proposes a meta-memory for knowledge sharing. [43] leverages a hierarchical classifier to share knowledge among classes. [44] learn an expert model for each class popularity, and combine them by knowledge distillation.

2.2 Dataset Overview

Visual Genome (VG) [24] is composed of 108k images across 75k object categories and 37k predicate categories, but 92% of the predicates have less than 10 instances. Following prior works, we use the original splits of the popular subset (i.e. VG150) for training and evaluation. It contains the most frequent 150 object classes and 50 predicate classes.The distribution remains highly long-tailed.

2.2.1 Long-tailed Nature

The long-tail nature of the problem is quite visible in the widely used Visual Genome [24] dataset. As shown in Figure 2.1, both the distribution of entity and predicate classes are long-tailed. For entities, the most populated class is $35\times$ larger than the least populated. For predicates, the former is $12,000\times$ larger than the latter ($5,000\times$ if the least frequent predicate class *flying*

in, is discarded). Note that this is much larger than the square of the ratio between entity classes $(1,225)$ suggested by the factorial nature of relationships.

2.3 Choosing Evaluation Metric

Since the ground-truth annotations of relationships are incomplete, it's improper to use simple accuracy as the metric. Therefore, Lu et al. transfer it to a retrieve-like problem in their work [30] and adopted *Recall*. The relationships are not only required to be correctly classified, but also required to have the highest score possible, so they can be retrieved from plenty of 'none' relationship pairs.

2.3.1 Recall@K (R@K)

This measures the average percentage of ground truth relation triplets that appear in the top K predictions and, like any average, is dominated by the most frequent relationship classes. Hence, it does not penalize solutions that simply ignore infrequent relationship classes. Focusing on designing ever more complex network architectures to optimize R@K performance, it is unclear whether all that is being accomplished is stronger overfitting to a few dominant classes (e.g. "on"). This is undesirable for two reasons.

1. The number of infrequent relations is much larger than that of dominant relationships.

2. While dominant relations include many obvious contextual relationships (e.g."car-has-wheels"), infrequent ones are potentially more informative (e.g. "monkey-playing-ball") of the scene content.

In summary, the long-tailed problem is exacerbated by the evaluation protocol, based on the Recall@K (R@K) measure and focus on optimizing R@K could lead to systems that are only capable of detecting a few relationships of relatively low information content.

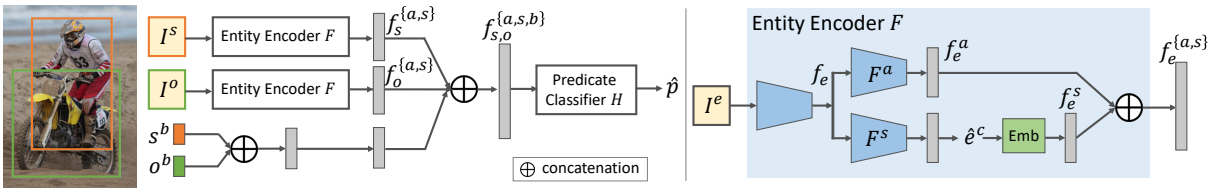

Figure 2.2. The model architecture of DT2 is composed of an entity encoder F (right) and a predicate classifier H.

2.3.2 Mean Recall@K (mR@K)

The main motivation of Mean Recall@K (mR@K) is that the VisualGenome [24] dataset is biased towards head predicates. If the 10 most frequent predicates are correctly classified, the accuracy would reach 90% even if the remaining 40 kinds of predicates are all wrong. This problem has been recognized in the recent literature, where some works [6, 33] have started to adopt the mRecall@K (mR@K) metric, which first averages the recall of triplets within the same predicate class and then averages the class recalls over all the predicate classes.

2.3.3 Category-wise mR@K

To zoom in more on the performance of classes with different popularity, we sort the 50 relation classes by their frequencies and divide them into 3 equal parts, head (16), body (17) and tail (17) and use the mR@K performance on these partitions for each SGG task.

2.4 Proposed Solutions

The solution we propose here leads from an alternative hypothesis: *Is the devil in the tails?* Or, in other words, can a simple model designed explicitly to cope with the long-tailed nature of visual relations outperform existing models, which are much more complex but ignore this property? To investigate this hypothesis, we introduce a solution that uses a model much simpler than recently proposed architectures, but is much more sophisticated in its use of sampling techniques that target the long-tailed nature of visual relationship.

2.4.1 Notations

For a relation tuple $r_j = (s_j, p_j, o_j)$ in image I, p_j is the ground truth predicate class, while $s_j = (s_j^b, s_j^c)$ and $o_j = (o_j^b, o_j^c)$ are the subject and object entities, composed of its associated bounding box coordinates (e.g. s_j^b) and ground truth entity class (e.g. s_j^c). The bounding boxes of an entity can be either the ground truth coordinates or the predictions from a detection model, depending on the task of interest (i.e. SGCls or SGDet). With the bounding boxes, the corresponding image patch I_j^s and I_j^o for the subject and object can be cropped from the image I.

In addition, we define ρ as a probability vector at the output of the softmax function with temperature τ, and its i^{th} entry is formulated as

$$\rho_i(f, \mathbf{W}, \tau) = \frac{\exp(\mathbf{w}_i^T f/\tau)}{\sum_k \exp(\mathbf{w}_k^T f/\tau)}\ , \tag{2.1}$$

where $f \in \mathscr{R}^d$ is a feature vector, $\mathbf{W} \in \mathscr{R}^{d\times k}$ is the matrix of k weight parameters $\mathbf{w}_k \in \mathscr{R}^d$.

2.4.2 Model architecture

Figure 2.2 summarizes the architecture of the *Decoupled Training for Devil in the Tails* (DT2) model. This combines an entity encoder F, as shown in the right part of Figure 2.2, and a predicate classifier H. DT2 takes the bounding box coordinates s_j^b, o_j^b [7] and the corresponding cropped image patches I_j^s and I_j^o as input. The entity encoder F is then applied to both I_j^s and I_j^o, to extract a pair of subject-object feature vectors $f_s^{\{a,s\}}, f_o^{\{a,s\}}$ that represent both the *appearance* and *semantics* of entities s_j and o_j. These are then concatenated with an embedding of the bounding box coordinates s_j^b and o_j^b, and fed to a predicate classifier H. Implementation details of the entity encoder and the predicate classifier are elaborated below.

Entity encoder

Entity encoder (F) first maps image patch I^e of entity e through a feature extractor, implemented with the first three convolutional blocks of a pretrained ResNet101 [16]. We use a

faster R-CNN pre-trained for object detection on Visual Genome under regular sampling (all images are sampled uniformly). The resulting feature vector f_e is then mapped to two feature vectors, f_e^s and f_e^a, that encode semantics and appearance information respectively, through two different branches sharing identical architecture. The semantic branch $F^s(\cdot;\theta)$ of parameter θ is implemented with a stack of convolution layers (the last convolutional block of ResNet101). Its output is then fed to a softmax layer that predicts the probability $\bar{e}^c \in [0,1]^C$ of the class of the entity e, i.e.

$$\bar{e}^c = \rho(F^s(f_e;\theta), \mathbf{W}^e, \tau = 1) \,, \tag{2.2}$$

where $\mathbf{W}^e$ is the matrix of the entity classifier weights and τ of ρ in (2.1) is set to 1. The one-hot encoding $\hat{e}^c$ can be generated by taking the *argmax* of $\bar{e}^c$, which is then mapped into a semantic feature vector $f_e^s \in R^{128}$ with a single fully connected layer.

While the semantic branch would be, in principle, sufficient to convey the entity identity to the remainder of the network, this does not suffice to infer visual relationships. For example, the detection of the "people" and "bike" entities in Figure 2.2 is not enough to infer whether the relationship is "person-standing by-bike" or "person-riding-bike". This problem is addressed by introducing the appearance branch $F^a(\cdot;\phi)$ of parameter ϕ, which outputs a feature vector $f_e^a \in R^{128}$ with no pre-defined semantics, simply encoding entity appearance. Finally, the feature vectors f_e^a and f_e^s are concatenated into a vector $f_e^{\{a,s\}} \in R^{256}$ that represents both the appearance and semantics of entity e.

Predicate classifier

Predicate classifier (H) takes the subject $f_s^{\{a,s\}}$ and object $f_o^{\{a,s\}}$ feature vectors as input. These vectors are then concatenated with an embedding of subject s^b and object o^b bounding boxes produced by a fully-connected layer, to create a joint encoding $f_{\{s,o\}}^{\{a,s,b\}} \in R^{520}$ of the semantics, appearance, and location of the subject-object patches I^s and I^o. The predicate

classifier H is implemented with a small feature extractor $H(.,\psi)$, consisting of three layers that perform dimension reduction. The input $f_{\{s,o\}}^{\{a,s,b\}} \in R^{520}$ is first transformed into a 256-dimension vector with a fully connected layer, followed by a batch normalization and a ReLU layer, the output of which is finally passed through a fully connected layer with a tanh non-linearity, to produce a final feature vector $f_{s,o} \in R^{128}$. This is fed to a softmax layer to produce the probability of the predicate class

$$\bar{p} = \rho(f_{s,o}, \mathbf{W}^p, \tau = 1) \tag{2.3}$$

where $\mathbf{W}^p$ is the weight matrix of the predicate classifier.

Model complexity is quite low for DT2, which has **10**$\times$ fewer trainable parameters than most of the recent approaches in the literature. For example, the SGCls model sizes of DT2, MOTIFS [50], VCTree [34] and TDE-MOTIFS [33] are **224 MB**, 1.68 GB, 1.65 GB and 2.1 GB respectively. This is by design, since our goal is to emphasize the importance of accounting for long tails during training, as is discussed next.

2.4.3 Training

DT2 is trained with standard cross-entropy losses targeted on entity and predicate classification. The former is defined as

$$L_{ent} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|E_i|}\sum_{e_k \in E_i} L_{ce}(e_k^c, \bar{e}_k^c) \tag{2.4}$$

where L_{ce} denotes the cross-entropy loss, $\bar{e}_k^c$ is the output probability prediction of (2.2) and e_k^c is the ground truth one-hot code of the k^{th} entity in the set E_i from image I_i. This is complemented by a predicate classification loss

$$L_{pred} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|R_i|}\sum_{r_k=(s_k,p_k,o_k)\in R_i} L_{ce}(p_k, \bar{p}_k) \tag{2.5}$$

where $\bar{p}_k$ is the output probability of (2.3) and p_k the ground truth one-hot code for the k^{th} predicate in the set R_i of visual relations in image I_i. Both (2.4) and (2.5) are important to guarantee that the network can learn from both entities and predicate relationships.

2.4.4 Sampling strategies

While encapsulating both semantics and appearance information, the proposed training loss in Sec. 2.4.3 requires a complementary sampling strategy tailored for long-tailed data. This long-tailed problem has been studied mostly in the object recognition literature, where an image patch is fed to a feature extractor with the parameter φ and the softmax layer ρ of (2.1) with weight matrix $\mathbf{W}$. A popular training strategy is to use different sampling strategies to train the two network components [23]. The intuition is that, because the bulk of the network parameters are in the feature extractor (φ), this should be learned with the largest possible amount of data. Hence, the entire network is first trained with **Standard Random Sampling (SRS)**, which samples images uniformly, independent of their class labels.

While this produces a good feature extractor, the resulting classifier usually overfits to the head classes, which are represented by many more images and have a larger weight on the cost function. The problem is addressed by fine-tuning the network on a balanced distribution, obtained with **Class Balanced Sampling (CBS)**. This consists of sampling uniformly over classes, rather than images, and guarantees that all classes are represented with equal frequencies. However, because images from tail classes are resampled more frequently than those of head classes, it carries some risk of overfitting to the former. To avoid overfitting, the fine-tuning is restricted to the weights $\mathbf{W}$ of the softmax layer. In summary, the network is trained in two stages. First, the parameters φ and $\mathbf{W}$ are jointly learned with SRS. Second, the feature extractor (φ) is fixed and the softmax layer parameters $\mathbf{W}$ are relearned with CBS.

2.4.5 Sampling for visual relationships

Similar to long-tailed object recognition, it is sensible to train a model for visual relations in two stages. In the first stage, the goal is to learn the parameters θ, ϕ, ψ of the feature extractors (see Sec. 2.4.2), which are the overwhelming majority of the network parameters. As in object recognition, the network should be trained with SRS. In the second stage, the goal is to fine-tune the softmax parameters $\mathbf{W}^e$ and $\mathbf{W}^p$ to avoid overfitting to head classes. However, unlike long-tailed object recognition, Figure 2.1 shows that predicates and entities can have very different distributions, which makes the learning of long-tailed visual relations a distinct problem. This indicates that two class-balanced sampling strategies are required to accommodate the distribution difference between predicate and entity classes.

A straightforward solution is to introduce a 2-step iterative training procedure, namely *entity-optimization step* (E-step) and *predicate-optimization step* (P-step), to optimize the weight of $\mathbf{W}^e$ and $\mathbf{W}^p$ respectively. In E-step, images are sampled from a distribution $\mathscr{P}_e$ that is uniform with respect to entity classes, which is denoted as Entity-CBS. While in P-step, they are sampled from a distribution $\mathscr{P}_p$ uniform with respect to predicate classes, denoted as Predicate-CBS. However, since the uniform sampling of $\mathscr{P}_p$ is not class-balanced for entity classes, P-step would lead to the overfitting of the entity classification parameters $\mathbf{W}^e$.

This is addressed by the novel sampling strategy, **Alternating CBS** (ACBS), tailored for long-tailed visual relations.

2.4.6 Alternating Class Balanced Sampling (ACBS)

ACBS contains a memory mechanism to maintain the entity predictions of the P-step, making sure that what was learned is not forgotten in the E-step. It is implemented with distillation [18] between the P-step and E-step and an auxiliary *teacher* entity classifier of weight matrix $\mathbf{W}^t$. The *teacher* entity classifier is inserted in parallel with the entity classifier of weight matrix $\mathbf{W}^e$ in (2.2), which is its *student*, and produces a second set of entity prediction

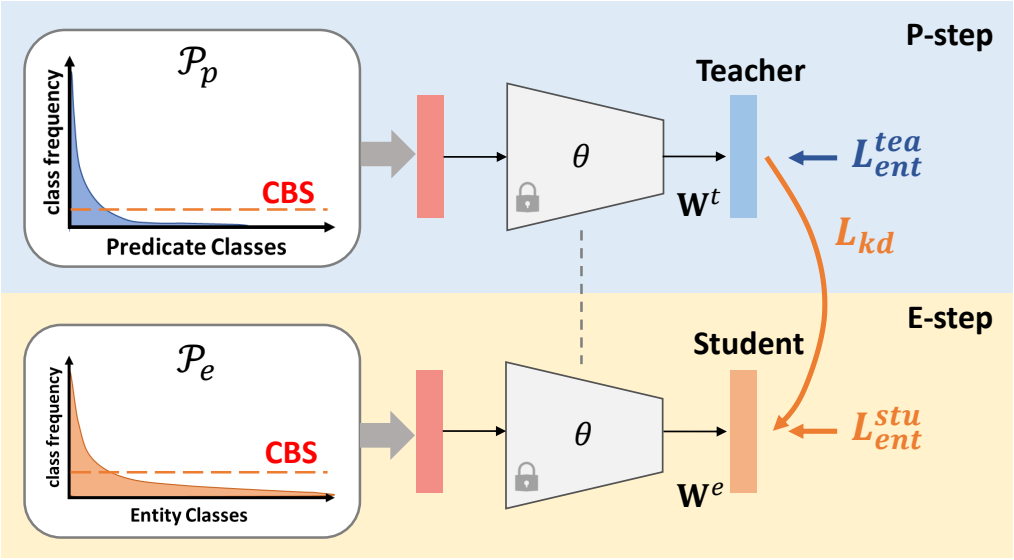

Figure 2.3. ACBS captures the interplay between the long-tailed distributions of entities and relations by implementing the knowledge distillation between P-step and E-step.

probabilities as

$$\vec{e}^t = \rho(F^s(f_e;\theta), \mathbf{W}^t, \tau = 1). \tag{2.6}$$

With the introduction of the teacher entity classifier, we rewrite (2.4) into L_{ent}^{stu} and L_{ent}^{tea}, where the former operates on $\bar{e}^c$ of (2.2) and the latter operates on $\vec{e}^t$. Furthermore, to distill knowledge from the teacher entity classifier, a Kullback-Leibler divergence (KL) loss (L_{kd}) is defined as

$$\mathrm{KL}(\rho(F^s(f_e;\theta), \mathbf{W}^e, \tau = \tau_s) || \rho(F^s(f_e;\theta), \mathbf{W}^t, \tau = \tau_s)), \tag{2.7}$$

where the two inputs to L_{kd} are the smooth version of (2.2) and (2.6) with temperature τ_s.

In summary, the P-step updates parameters $\mathbf{W}^p$ of the predicate classifier and $\mathbf{W}^t$ of the teacher with (2.5) and L_{ent}^{tea} respectively, while the student parameters $\mathbf{W}^e$ are kept fixed. In the E-step, $\mathbf{W}^p$ and $\mathbf{W}^t$ (teacher) are kept fixed, and $\mathbf{W}^e$ (student) is optimized with L_{ent}^{stu} and (2.7). This implements learning without forgetting [27] between the two steps, encouraging the student

Algorithm 1: Training procedure of ACBS

Input: Training dataset $\mathscr{D}$, predicate distribution $\mathscr{P}_p$, entity distribution $\mathscr{P}_e$, ACBS hyperparameters (α, β, τ_s), and model parameters (θ, ϕ, ψ).
Output: Model parameters ($\mathbf{W}^p$, $\mathbf{W}^e$).
while *Not convergence* **do**
  // P-Step
  $\mathscr{D}_p \leftarrow BalancedSample(\mathscr{D}, \mathscr{P}_p)$;
  while *batch in* $\mathscr{D}_p$ **do**
    $L_{total} \leftarrow L_{pred}$ (2.5) $+ \beta L_{ent}^{tea}$ (2.4);
    Minimize L_{total} with respect to $(\mathbf{W}^p, \mathbf{W}^t)$
  end
  // E-Step
  $\mathscr{D}_e \leftarrow BalancedSample(\mathscr{D}, \mathscr{P}_e)$;
  while *batch in* $\mathscr{D}_e$ **do**
    $L_{total} \leftarrow L_{ent}^{stu}$ (2.4) $+ \alpha L_{kd}$ (2.7);
    Minimize L_{total} with respect to $\mathbf{W}^e$
  end
end

classifier to mimic the predictions of the teacher classifier, and enabling the network to learn the new parameters for one distribution, e.g. $\mathbf{W}^e$, without forgetting the one, e.g. $\mathbf{W}^t$, previously learned for the other. The training procedure is detailed in Algorithm 1.

2.4.7 Implementation

DT2-ACBS is a two-stage training process. While SRS is adopted in the first stage when training the parameter of θ, ϕ and ψ, the proposed ACBS is adopted in the second stage to learn the classifiers. Apart from the differences in sampling strategies, both stages share a similar optimization scheme, where the Adam optimizer with initial learning rate 10^{-3} is adopted, with the learning rate decay of 0.5 for every 5 epochs. The batch size in the first stage is 256, while in the second stage, objects and predicates are sampled with 2 and 5 samples per class respectively. The hyperparameters α, β and τ_s are set to 0.5, 1 and 10 respectively using the validation set. For evaluation on SGG tasks, we adopt the protocol of [50, 34] to filter out the subject-object pairs that do not have a relationship.

Table 2.1. The result (mRecall@K) of SGG tasks (PredCls, SGCls, SGDet) compared to SOTA in scene graphs. Results for other methods are reported from the corresponding paper in general. † denotes our reproduced model with ResNet101-FPN backbone.

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
IMP+ [45]	-	9.8	10.5	-	5.8	6.0	-	3.8	4.4
FREQ [50]	8.3	13.0	16.0	5.1	7.2	8.5	4.5	6.1	7.1
MOTIFS [50]	10.8	14.0	15.3	6.3	7.7	8.2	4.2	5.7	6.6
MOTIFS [50]†	13.2	16.3	17.5	7.1	8.8	9.3	4.9	6.7	8.2
KERN [6]	-	17.7	19.2	-	9.4	10.0	-	6.4	7.3
VCTree [34]	14.0	17.9	19.4	8.2	10.1	10.8	5.2	6.9	8.0
GBNet [48]	-	22.1	24.0	-	12.7	13.4	-	7.1	8.5
TDE-MOTIFS-SUM [33]	18.5	25.5	29.1	9.8	13.1	14.9	5.8	8.2	9.8
TDE-MOTIFS-SUM [33]†	17.9	24.8	28.6	9.6	13.0	14.7	5.6	7.7	9.1
TDE-VCTree-SUM [33]	18.4	25.4	28.7	8.9	12.2	14.0	6.9	9.3	11.1
TDE-VCTree-GATE [33]	17.2	23.3	26.6	8.9	11.8	13.4	6.3	8.6	10.3
PCPL [46]	-	35.2	37.8	-	18.6	19.6	-	9.5	11.7
DT2-ACBS (ours)	**27.4**	**35.9**	**39.7**	**18.7**	**24.8**	**27.5**	**16.7**	**22.0**	**24.4**

2.5 Comparison to SOTA

To validate our hypothesis, we compare DT2-ACBS with the state-of-the-art methods on PredCls, SGCls and SGDet task on the popular subset VG150 of VG [24], under the mRecall@K metric. As shown in Table 2.1, compared baselines include 1) simple frequency-based method [50], 2) sophisticated architecture design for contextual representation learning [45, 6, 34, 48] and 3) recent works that tackle the long-tailed bias of predicate classes [33, 46]. Several observations can be made. First, DT2-ACBS outperforms all baselines in the first two groups by a large margin (mR@100 gain larger than 15.7%) on the PredCls task, where entity bounding boxes and categories are given. The baselines in the third group [33, 46], which address the long-tailed bias of the predicate distribution, are similar in spirit to DT2-ACBS. However, the latter relies on a simpler model design and a more sophisticated decoupled training scheme to overcome overfitting. This enables a 1.9% improvement on mR@100 (5% relative improvement), showing the efficacy of the proposed sampling mechanism for tackling the long-tailed problem in predicates distribution.

Next, when predicting both predicate and entity class given the ground truth bounding boxes (SGCls task), DT2-ACBS outperforms all existing methods by a larger mR@100 margin

Table 2.2. mR@100 on SGG tasks for head, body, tail classes. † denotes our reproduced models with ResNet101-FPN backbone.

Method	Predicate Classification			Scene Graph Classification			Scene Graph Detection		
	Head (16)	Body (17)	Tail (17)	Head (16)	Body (17)	Tail (17)	Head (16)	Body (17)	Tail (17)
MOTIFS [50]†	42.3	9.8	0.6	24.6	4.0	0.1	20.2	4.6	0.4
TDE-MOTIFS-SUM [33]†	**44.9**	35.8	6.1	**25.6**	15.8	3.3	22.2	5.6	0.1
DT2-ACBS (ours)	35.1	**45.2**	**38.6**	24.6	**29.1**	**28.6**	**22.3**	**26.7**	**24.0**

(1.9% on PredCls vs 7.9% on SGCls, equivalently relative improvement of 5% in PredCls vs 40% in SGCls). This significant improvement in SGCls performance can be ascribed to the decoupled training of ACBS, which better captures the interplay between the different distributions of entities and predicates.

Finally, we also ran DT2-ACBS on proposal boxes generated by a pre-trained Faster-RCNN for the SGDet task.Table 2.1 shows that DT2-ACBS outperforms existing methods by a significantly larger mR@100 margin of 12.7% ($>$ 100% relative improvement) on the SGDet task.

2.5.1 Class-wise performance analysis:

To study the performance of classes with different popularity, we sort the 50 relation classes by their frequencies and divide them into 3 equal parts, head (16), body (17) and tail (17). Table 2.2 presents the mR@100 performance on these partitions for each SGG task. As observed in prior long-tailed recognition work [29, 23], a performance drop in head classes is hard to avoid while improving tail class performance. The goal, instead, is to achieve the best balance among all the classes, which DT2-ACBS clearly does with notable improvements in the body and tail classes. It should also be noted that the drop in head performance can be deceiving, due to dataset construction problems like "wearing" and "wears" appearing as two different relationship classes. Most importantly, many VG150 tail categories (e.g. "standing on", "sitting on") are fine-grained versions of a head category ("on"). Some of the degradation in head class performance is just due to the predicates being pushed to the fine-grained classes, which is more informative. We notice that one of the high-frequency predicate classes *On* has a

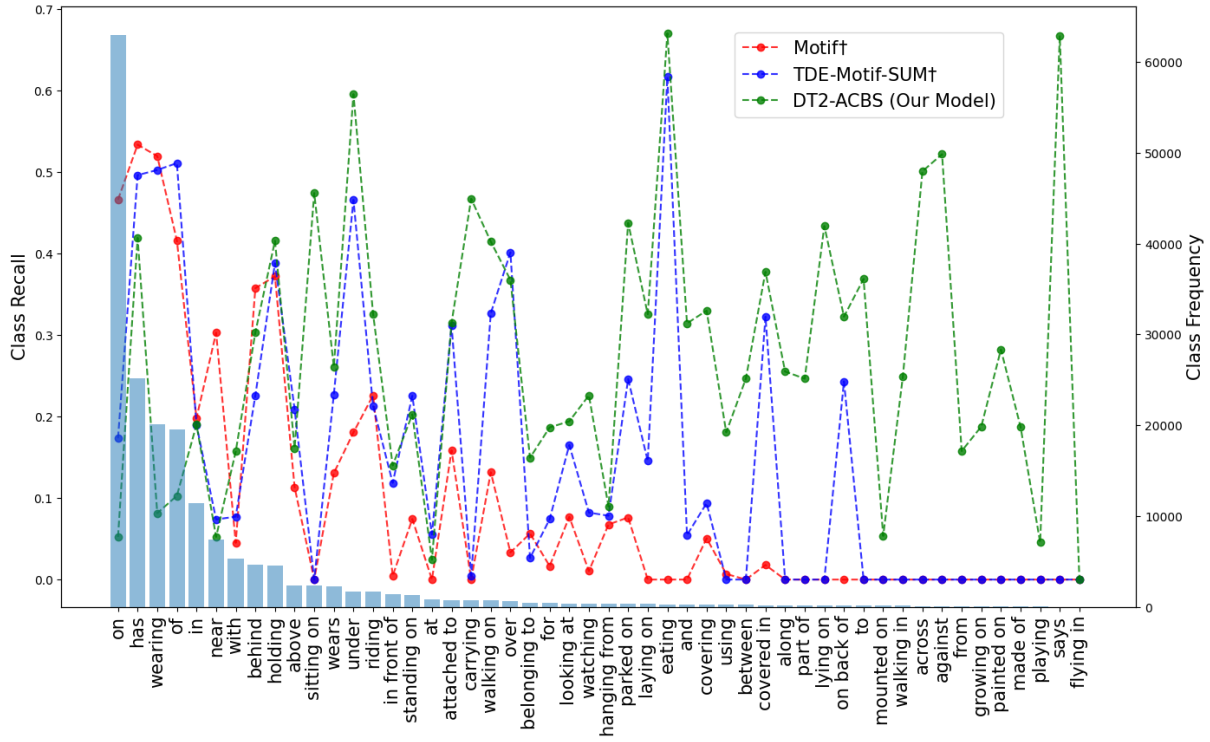

Figure 2.4. Comparisons of per class Recall@100 on SGCls. Classes are sorted in decreasing order of the number of samples.

low recall value (Figure 2.4) and observe that DT2-ACBS often instead predicts its fine-grained sub-categories, such as *standing on*, *sitting on*, *mounted on*. In particular, there are 41,620 ground truth instances of *On* predicate in the test set, and DT2-ACBS predicts *On*-subcategories 14,317 times on PredCls, which constitutes 34% incorrect predictions as per the metric. Overall, DT2-ACBS performs significantly better in body and tail classes on SGG tasks, and performs comparably on head classes for SGCls and SGDet, reaching the best balance across all the classes.

2.6 Ablation Studies

2.6.1 Ablation on Appearance Branch

The goal of appearance branch is to convey the image information *not* encoded in the entity labels but *relevant to predicate predictions.* Hence, appearance labels are difficult to define.

Table 2.3. Ablations of appearance branch. (subj, obj) Acc. denotes the accuracy of a pair of subject and object class.

Method	PredCls **mR@ 20 / 50 / 100**	SGCls **mR@ 20 / 50 / 100**	(subj, obj) **Acc**
w/o F^a	18.1 / 24.5 / 26.8	11.0 / 14.7 / 16.3	25.77
w/ F^a (ours)	**27.4 / 35.9 / 39.7**	**18.7 / 24.8 / 27.5**	26.26

Table 2.4. Ablations of ACBS with different teachers in SGCls.

Teacher	**mR@ 20 / 50 / 100**
E-step	15.2 / 20.2 / 22.0
P-step (ours)	**18.7 / 24.8 / 27.5**

We test the effectiveness of the appearance branch F^a by removing it and training the network with ACBS. Table 2.3 shows that entity classification accuracy remains similar, but PredCls and SGCls performance drops dramatically. Hence, the appearance branch contributes substantially to predicate classification. Note that the gains hold even when the ground truth entity labels are used (PredCls), confirming the argument that simply knowing entity classes is not enough for predicate prediction.

2.6.2 Ablations on Teacher

An intuitive experiment is to have E-step as the teacher rather than the P-step. In ACBS, $\mathbf{W}^e$ receives class-balanced *entity supervision,* so there is no risk of overfitting. The role of the teacher is to guarantee that the E-step update of $\mathbf{W}^e$ is not incompatible with the P-step update of $\mathbf{W}^p$. This distillation is exactly how ACBS fuses the knowledge learnt with different distributions. Using E-step as the teacher has weaker results, as shown in Table 2.4. According to our experiments, entity CBS is more important for the entity classification, and it should be the base of the entity classifier and not the other way around.

2.6.3 Ablations on sampling strategies

SGCls performance is affected by the intertwined entity and predicate distributions. In this section, we conduct ablation studies in Table 2.5 on 1) single-stage vs two-stage training and

Table 2.5. Ablations on different sampling strategies for SGCls.

Method	mR@20	mR@50	mR@100
Single Stage-SRS	6.4	9.6	11.2
Single Stage-Indep. CBS	8.5	11.2	12.4
DT2-Predicate-CBS	10.0	13.0	14.3
DT2-Indep. CBS	17.3	23.9	26.7
DT2-ACBS (ours)	**18.7**	**24.8**	**27.5**

2) different sampling schemes. The first half of the table shows the performances of single-stage training, where the representation and the classifier are learned together. This clearly underperforms the two-stage training, which is listed in the second half of the table, where we compare different sampling strategies in the second stage of DT2. For the predicate classifier, it can be trained based on either SRS or class-balanced sampling for predicates (Predicate-CBS). Since each relation comes with a subject and an object, it is possible to train the entity classifier with respect to Predicate-CBS, indicating the entity classifier can be trained based on SRS, Predicate-CBS or class-balanced sampling for entities (Entity-CBS). Note that the predicate classifier can not be trained with Entity-CBS, since an entity does not always belong to a visual relation tuple. From the second half of the table, we find that considering the distribution differences in predicates and entities is important, because DT2-Predicate CBS (i.e. Predicate-CBS for both entity and predicate classifier) does not perform as well as DT2-Indep. CBS (i.e. Entity-CBS for the entity classifier and Predicate-CBS for the predicate classifier). The observation that DT2-Indep. CBS already performs better than existing methods (Table 2.1) supports our claim that visual relations can be effectively modeled with a simple architecture if the long-tailed aspect of the problem is considered. Nevertheless, the proposed ACBS further improves the SGCls performance by distilling the knowledge between P-step and E-step (see Algorithm 1).

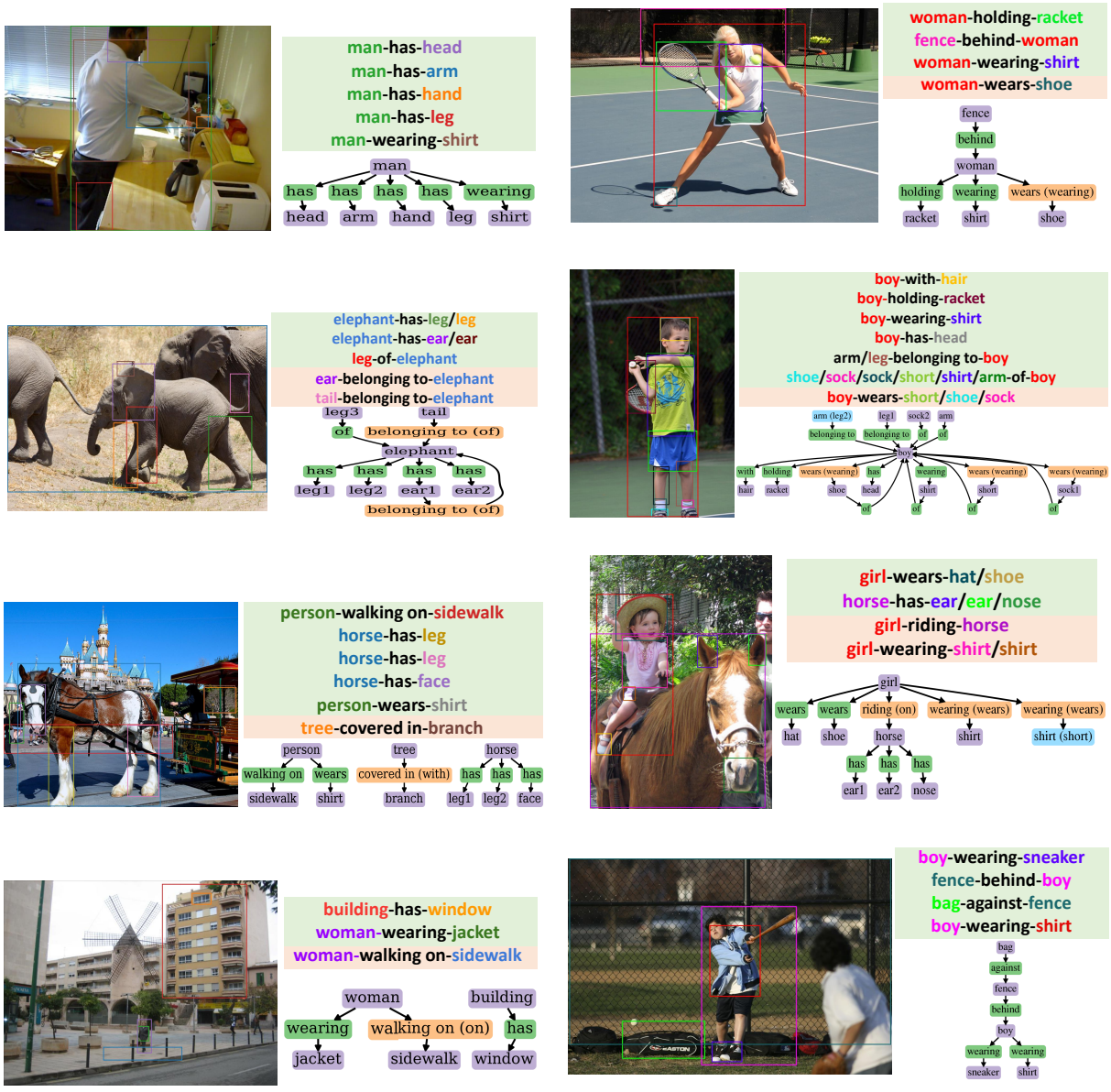

Figure 2.5. Qualitative results of PredCls and SGCls. Bounding box colors in image correspond to entities in triplets. Correct/incorrect predicates have green/orange background. In graphs, correct/incorrect entities are in purple/blue and predicates are in green/orange. Ground truth is in brackets.

2.7 Qualitative results

Figure 2.5 presents qualitative results of DT2-ACBS. In PredCls task, DT2-ACBS can correctly predict populated predicate classes (*has* & *wearing*) as well as non-populated predicate classes (*walking on*). Not only robust to long-tailed predicate classes, DT2-ACBS is also able to classify entities ranging from more populated classes (*boy* and *h*ead) to tail classes (*sneaker*, *r*acket and *s*ock).

We can observe that while the predicted predicates can be different from the ground truth, the relation can still be reasonable (e.g. a *subclass* or a *synonym* of the ground truth). For example, the predicted predicate "walking on" is actually a subclass of the ground truth predicate "on". These examples show that DT2-ACBS is able to predict more fine-grained predicates in tail classes and provide more exciting descriptions.

Chapter 2, in part, contains material from A. Desai, TY Wu, S Tripathi, N Vasconcelos,"Learning of Visual Relations:The Devil is in the Tails", 2021 IEEE International Conference on Computer Vision (ICCV), Montreal, Canada, 2021. The thesis author was the primary investigator and author of this paper.

Chapter 3

Future Work

While DT2 ACBS learns the interplay between the joint long tailed distribution of the entities and predicates, it's inference time is of the order of $O(n^2)$. This makes the use of this method prohibitive for real time scene graph generation. The model, however, emphasizes the importance of taking the long tailed nature of the distribution into consideration.

With this in mind we explore a DETR based model that provides an $O(n)$ inference time solution.

3.1 Scene Graph Generation with DETR

Transformers have changed the landscape of Deep Learning since their introduction in 2017 by [35]. Initially designed with sequence-to-sequence problems in mind, transformers have come a long way since their inception. Object detection is a task where a model learns to localize and classify the foreground objects from the background. Most deep learning approaches attempt to solve the task of object detection either as a classification problem or as a regression problem or both.

Facebook's DEtection TRansformer or DETR is "a method that views object detection as a direct set prediction problem."[4] Their approach removes the need for hand-designed components like a non-maximum suppression or anchor generation. The essence of the framework is a set-based global loss that forces unique predictions via bipartite matching and a transformer

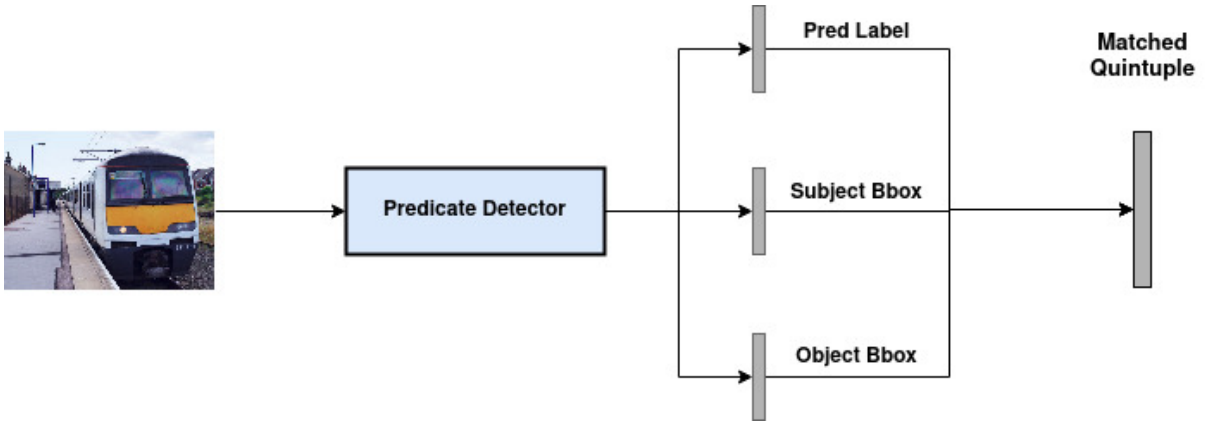

Figure 3.1. The model architecture DETR based basline

encoder-decoder architecture.

The task of scene graph generation involves three sub tasks: entity detection, entity recognition and predicate recognition. While entity recognition and predicate recognition are long-tailed problems, entity detection, that is, bounding box detection is not. Therefore, while SGG on the whole, is a joint long-tailed problem, entity detection combined with predicate recognition is long tailed only on the predicate distribution. We leverage this idea along with the understanding that training predicate recognition on entity labels does not allow the model to learn good "visual" features. We call the task of predicting the $<$*object bounding box, predicate label, subject bounding box*$>$ tuple **Predicate Detection** and we train a DETR model to perform this task.

3.2 Baseline model

To test our hypothesis we train the model on a task that is not a standard Scene Graph Generation Task in the literature. The model is trained to "detect" the relationship between the objects in the images. In essence it detects the objects and classifies the relationship between them, without classifying the objects themselves, previously defined as Predicate Detection. This is important because the problem remains a single long tailed problem, that of predicate classification combined with the simpler detection problem. Since this is not a standard task in

Table 3.1. The result (Recall@K, mRecall@K) of Predicate Detection.

Method	Predicate Detection					
	R@20	R@50	R@100	mR@20	mR@50	mR@100
DETR based baseline (ours)	32.7	46.1	55.9	17.2	18.0	18.1

the literature, a head to head comparison of various methods is not possible. We can however, consider this task to be close in meaning to the scene graph generation task if it did not have to predict the object labels.

As shown in Figure 3.1 the baseline model consists of a single branch, the predicate detection branch. It is trained to predict the bounding boxes of the subject and the object along with the predicate class label.

3.2.1 Training

DETR is known for taking very long amounts of time to converge and is notoriously resource-intensive. There are newer variants of DETR like the Deformable-DETR [53] and PnP-DETR [37] which converge much faster and are easier to train. We use one of these variants to conclude baseline experiments.

Predicate Detection: A PnP-DETR model with an encoder, decoder and three feed forward networks is trained with 200 queries to predict the <subject bounding box, object bounding box, predicate label> tuples given an image.

3.2.2 Results and Discussion

The results of the baseline support the hypothesis. The results of predicate detection show that the model learns the visual "definition" of a relationship. While previous models train the predicate classifier on both image features and the labels, the labels overpower the image features and the models can never truly converge to their potential. Also, since this is now long-tailed only in terms of the predicate distribution, this can be fine-tuned for improved mRecall performance, by fine tuning for the long tailed nature of the distribution.

The scene graph generation results show that even though the model was not trained with object labels at all it performed well in terms of recall.

Moreover, the results show another benefit of allowing the model to learn "visual relations" and that is the confidence of the models in its predictions. As can be seen in Table 3.1, the mean recall values do not drop as drastically from mR@100 to mR@20. This shows that the training is more stable and the model generalizes better.

This method learns to predict the nodes of the predicate detection in the form of a set and therefore can perform in $O(n)$ time. However, as can be seen in the results, the mean recall of any method not exclusively designed to learn the effect of the long-tailed distribution is always lagging far behind a method that takes that into consideration. The next steps in this direction is to incorporate a form of class balancing suitable to the DETR architecture that can help with modelling the underlying long-tailed distribution.

Chapter 3, in part is currently being prepared for submission for publication by A. Desai, TY Wu, S Tripathi, N Vasconcelos. The thesis author was the primary investigator and author of this material.

Chapter 4

Conclusion

Learning visual relations is inherently a long-tailed problem. Existing approaches have mostly proposed complex models to learn visual relations. However, complex models are ill-suited for long-tailed problems, due to their tendency to overfit. In this thesis, we consider the uniqueness of visual relations, where entities and relations have skewed distributions. We propose a simple model, namely DT2, along with an alternating sampling strategy (ACBS) to tackle the long-tailed visual relation problem. Extensive experiments on the benchmark VG150 dataset show that DT2-ACBS significantly outperforms the state-of-the-art methods of more complex architectures.

We then explore a set detection method for the SGG problem and use it to generate a model that learns relationships independently from the entities that constitute it, thereby decoupling the two long-tailed distributions. We train a model to prove the validity of this statement. Further research is required to incorporate the effect of the long tailed distribution in the new DETR based baseline.

Scene graph generation is a joint long tailed problem and has two primary modalities, one inherently stronger than the other. In order to design models that can "understand" a scene by looking at it, we must consider both these properties.

Bibliography

[1] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera, 2019.

[2] The British Machine Vision Association and Society for Pattern Recognition. What is computer vision? https://web.archive.org/web/20170216180225/http://www.bmva.org/visionoverview.

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[5] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002.

[6] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[7] Vincent S. Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[8] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *International Conference on Computer Vision (ICCV)*, 10 2017.

[10] Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationships as functions: Enabling few-shot scene graph prediction. *CoRR*, abs/1906.04876, 2019.

[11] Chris Drummond and Robert Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats oversampling. *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Datasets*, 01 2003.

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

[13] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[14] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. *Advances in Intelligent Computing*, 3644:878–887, 09 2005.

[15] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, Sep. 2009.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. *CoRR*, abs/1802.05451, 2018.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[19] Zih-Siou Hung, Arun Mallya, and Svetlana Lazebnik. Union visual translation embedding for visual relationship detection and scene graph generation. *CoRR*, abs/1905.11624, 2019.

[20] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[21] J. Johnson, R. Krishna, M. Stark, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 3668–3678, June 2015.

[22] 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 (CVPR)*, June 2015.

[23] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020.

[24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR*, abs/1602.07332, 2016.

[25] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, 2021.

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

[27] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

[28] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 07 2018.

[29] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors, 2016.

[31] Hang Qi, Yuanlu Xu, Tao Yuan, Tianfu Wu, and Song-Chun Zhu. Scene-centric joint parsing of cross-view videos. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 91–99. Curran Associates, Inc., 2015.

[33] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3713–3722. IEEE, 2020.

[34] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[36] Ruize Wang, Zhongyu Wei, Piji Li, Qi Zhang, and Xuanjing Huang. Storytelling from an image stream using scene graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9185–9192, Apr. 2020.

[37] Tao Wang, Li Yuan, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Pnp-detr: Towards efficient visual analysis with transformers, 2022.

[38] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision (ECCV)*, 2016.

[39] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[40] Bin Wen, Jie Luo, Xianglong Liu, and Lei Huang. Unbiased scene graph generation via rich and fair semantic extraction, 2020.

[41] Wikipedia. Long-tailed distribution. https://en.wikipedia.org/wiki/Long_tail.

[42] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 560–570. Curran Associates, Inc., 2018.

[43] Tz-Ying Wu, Pedro Morgado, Pei Wang, Chih-Hui Ho, and Nuno Vasconcelos. Solving long-tailed recognition with deep realistic taxonomic classifier. In *European Conference on Computer Vision (ECCV)*, 2020.

[44] Liuyu Xiang and G. Ding. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European Conference on Computer Vision (ECCV)*, 2020.

[45] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[46] Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. PCPL: predicate-correlation perception learning for unbiased scene graph generation. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, pages 265–273, 2020.

[47] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[48] 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)*, August 2020.

[49] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[50] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. *CoRR*, abs/1711.06640, 2017.

[51] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph generation. *CoRR*, abs/1903.02728, 2019.

[52] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. pages 1–8, 2020.

[53] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection, 2021.

[54] Yang Zou, Zhiding Yu, B.V.K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *European Conference on Computer Vision (ECCV)*, 2018.