YOLO-SG: An Efficient Framework for Scene Graph Generation

Shui Jie
School of Computer Science
Peking University
SCHOOL ID HERE <!!!>

Aleksei Lobanov
School of Computer Science
Peking University
SCHOOL ID HERE <!!!>

Lawrence Leroy Chieng Tze Yao
School of Computer Science
Peking University
SCHOOL ID HERE <!!!>

Charles Young
School of Computer Science
Peking University
2401112106

Abstract

Scene graph generation (SGG) is the task of detecting object pairs and their relations in a visual medium, widely used for captioning, generation, and visual question answering. 2D scene graph generation is a subtask that focuses on generating a 2D graph given an image. While the development of models capable of performing 2D SGG has improved in both accuracy and speed, the computational complexity of the problem and the inherently long-tailed distribution of large, available datasets have led to generation speed and accuracy less than ideal for real-time use. Mainstream approaches focus on two-stage generation, where object detection is performed first, followed by a series of comparisons for relation inference. However, these have an inherent drawback where the computational complexity of detecting n objects and their relationships is n^2. More recent models have utilized encoder-decoder structures to reduce generation into a 1-stage problem. Unfortunately, the computation required for these architectures is still too high. Additionally, the model bias caused by long-tailed data distributions remains a key problem in both approaches. In this work, we propose YOLO-SG, a novel SGG framework capable of operating in real-time by decoupling object detection and relation detection and by performing relationship inference with multiple detection models in parallel. Our proposal seeks to both alleviate the effects of the long-tailed distribution problem and perform high-speed inference. Preliminary experiments on the Visual Genome 1.2 dataset demonstrate that YOLO-SG can achieve competitive performance with state-of-the-art models while maintaining high inference speed.

1 Introduction

In recent decades, object detection problems have become an increasingly popular subject in research literature[32] as the rise of deep learning[13] has led to many breakthroughs in the field. In this work, we leverage this rapid advancement in object detection to improve upon a downstream task: scene graph generation (SGG)[10]. SGG encompasses the set of tasks that focus on detecting object pairs and their relationships in visual media, creating scene graphs that can then be used for applications such as image captioning[7, 6], image generation[26], and visual question answering[31]. Two significant barriers stand in the way of higher performances for SGG tasks: The long-tailed distribution problem [15, 3] and the quadratic increase in candidate triplets [15, 3, 29]. When collecting data regarding a subject in an image and its relation to other objects, a small number of predicate terms often appear significantly more than all other terms. For example, 'on' can be used to

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

describe the relation between almost any subject-object pair where the subject is 'above' the object, and hence will appear significantly more than predicates such as 'mounted' or 'reaching'. Despite many attempts to alleviate the effect this uneven distribution has on the final results[14, 5, 18, 8], the prediction bias that arises from the uneven data distribution remains an important problem to be solved. The second barrier is the quadratic increase in computation required proportional to the number of detected objects. Mainstream approaches first obtain a list of objects in an image, and then infer the predicate (if one exists) for every possible subject-object pair [29, 10]. Unfortunately, the number of parameters required in the second stage scales quadratically with the number of objects found, as n objects would have n^2 potential predicates. Recent works have proposed a one-stage approach that leverages transformers to generate a set of triplets using object information[3, 16]. However, the large number of parameters required in a transformer architecture also slows down the model inference times, preventing the use of these approaches in real-time applications. This paper seeks to address the aforementioned long-tailed data distribution by leveraging the recent advancements in object detection accuracy. To do this, we first reformulate the SGG task into an object detection and classification problem, where each relationship triplet is represented as three objects: the subject, the object, and the relationship predicate. By formulating the problem in this way, we can divide the training data into balanced subsets, where every subset consists of predicate classes with a similar number of instances. This allows us to train multiple detection models, each specialized in detecting a subset of predicates. More importantly, we propose this approach as a solution to the long-tailed distribution problem, as these models will have significantly less bias than a single model trained on all predicate instances. Specifically, we propose a new framework, YOLO-SG, that leverages the YOLOv11 object detection model[21] to detect objects and predicates in parallel. YOLO-SG consists of four main components: object detection, predicate detection, object clustering, and predicate inference. The object and predicate detection module is composed of two sets of YOLOv11 models[21]. The first set is trained to detect and label objects, while the second set is trained to detect and label predicates. Additionally, we divide the training data such that every subset consists of 4 predicate classes, where the number of instances of each class in the subset is balanced. By doing so, we obtain multiple unbiased models that detect and label their own set of predicate classes instead of a single heavily biased model that predicts all predicate classes at once. The predicate inference module takes the output of the first stage and obtains a list of objects for each predicate using a simple clustering algorithm. It then passes every possible pair of objects in this list through a lightweight multilayer perceptron which is trained on the same subset of predicate classes. This then returns confidence scores for that subset of predicates. We then return the triplet with the highest confidence score of the detected predicate as the final output. Since we only use the object labels and positions as inputs as opposed to visual features, we can significantly reduce the number of trained parameters compared to methods that rely on convolutional layers or transformer architectures. We evaluate our model on the Visual Genome 1.2 dataset[12] and demonstrate that our model can achieve competitive performance with state-of-the-art models while maintaining real-time inference speeds.

2 Team Members

Shui Jie

As the overall project manager of the team, I am responsible for the overall work of the projects, including different iterations of goals and work distribution. I am responsible for designing the overall pipeline, training the YOLO model, and data analysis on individual model performance on different sets of objects and relationships. I also took part in coding the APIs to integrate my models into the complete pipeline.

Aleksei Lobanov

My name is Aleksei Lobanov, I am 23 years old, and I did my undergrad at Shenzhen MSU-BIT University, studying Applied Mathematics, where I did research on Formal Languages. At PKU, I study at the School of CS, NEEC Lab under Professor Luo Guojie. My research interests include programming languages, compiler development, and other systems programming-related topics.

This project was the first time I had to do anything ML-related, it was a big learning experience for me. I had two big tasks: pre-processing the dataset to try to improve the class imbalance within the predicates, and developing a small neural network that tries to assemble the final output of object-predicate-subject triplets. I found working on the neural network particularly challenging, since I had to learn everything from scratch, and I have to thank my amazing teammates for helping me throughout the semester.

Lawrence Leroy Chieng Tze Yao

Charles Young

My name's Charles Young, I'm a 23-year-old who went to undergrad at UC San Diego for a bachelor's in Mathematics and Computer Science. I study under Professor Tao Xie, and my research interests include mutation testing, metaheuristics, and developer tool use. I was responsible for writing the report and researching related works to improve our framework. I was able to learn a lot about the current state of the art in scene graph generation and object detection, and I'm grateful to my teammates for their hard work and dedication to the project.

3 Background and Related Work

Scene Graph Generation

Since beginning in 2015 with Johnson et als. work[10], many works have been published improving upon and expanding the scope of SGG[3, 4, 29, 17, 11, 16, 20, 9, 14, 5, 8, 26, 18, 24, 2, 28]. Mainstream works often focus on the development of a two-stage scene graph generation approach composed of an object detection stage and a contextual reasoning stage [29**?** , 11]. This two-stage approach the is foundation for most modern SGG methods[29, 4, 17, 5, 8]. The first stage is often completed using pre-trained detection models[1], while contextual reasoning is done using a variety of methods, from convolutional neural networks[29] to transformers[3]. To address the quadratic growth of computation relative to object count common in two-stage models, recent works have proposed several one-stage approaches to the problem[3, 16]. These models leverage transformer architectures to generate a set of triplets using object information and have been shown to be competitive with two-stage models in terms of performance. However, the large number of parameters required in a transformer architecture also slows down the model inference times, preventing the use of these approaches in real-time applications[3, 18].

Long-tailed Distribution Problem Several works have been proposed to mitigate the long-tail bias introduced by the data[24, 30, 2, 9]. These works can be split into two categories: approaches that used biased training data paired with extra learning techniques or approaches that attempt to remove bias during training. In the first category, TDE[24] leverages a causal graph to infer the effect of long-tail bias. CogTree[30] builds a cognitive structure that distinguishes course-to-fine relationships based on biased predictions. DLFE[2] uses Dynamic Label Frequency Estimation to recover unbiased probabilities and reduce reporting bias. In the second category, The second category of works often approaches the problem by preprocessing training data to remove bias. PCPl[28] utilizes the correlation between predicate classes to adjust the loss weights. BGNN[17] uses a bi-level data resampling strategy with a confidence-aware bitartite graph to reduce bias. One particular work that heavily motivated our approach is the Context Knowledge Network (CKN) proposed by Jin et al. [9], which only uses the object labels and bounds as an input to a multilayer perceptron with the number of possible predicates as the output. Despite the lack of visual information, the authors were still able to achieve competitive high-speed performance with state-of-the-art models, whilst also reducing model bias. This work motivated our approach with the use of a lightweight MLP similar to CKN, as Jin et al. demonstrated potential in the ability of machine learning models to extract relationship information from non-visual contexts[9]. However, our approach differs as we use more than one model, with each model specializing in obtaining confidence scores for only a subset of predicates.

You Only Look Once (YOLO)

YOLO is a popular object detection model that has been widely used in many computer vision tasks due to its ability to perform single-pass, real-time inferencing. The first version of YOLO proposed by Redmon et al. in 2016 allowed for end-to-end training, as well as object bounding box proposal and object classification in a single forward pass[21]. More importantly, the model was able to achieve this at a rate of 45 frames per second, more than enough for real-time applications. Since then, multiple versions of YOLO have been introduced by various different authors, each of which attempt to improve upon the original whilst following its core design philosophy: open source, end-to-end, and one-shot[25]. These newer versions often apply novel techniques to achieve greater overall performance, measured by inference speed, training cost, and inference accuracy[25, 22, 23], For our framework, we chose to use the YOLOv11 model due to its higher inference speed compared to YOLOv8 due to the lower number of trained parameters[27]. However, it is still possible that performance could be improved using an older version, but this will be left as potential future work.

4 Method

5 Object Detection

Our proposed framework, YOLO-SG, decouples the process of object and predicate detection by framing both subproblems as standard object detection tasks. Rather than inferring predicates between every pair of detected objects in a computationally expensive second stage, we directly detect "predicate entities" in parallel alongside object detection. By modeling predicates as distinct objects, we leverage YOLOv11[27] for both object and predicate recognition. This approach reduces the complexity of pairwise comparisons from n^2 operations to a more manageable parallel detection task.

6 Clustering Algorithm

We chose to use an Intersection-Over-Union (IoU) based clustering algorithm to associate detected objects with their corresponding predicates due to its simplicity and low time complexity. Given n objects, the algorithm has a worst-case complexity of $O(n \log n)$. For each predicate entity detected, we calculate the IoU of objects that overlap with the predicate bounds, grouping all objects that have an IOU larger than 0.5 with the predicate bounding box. Due to time constraints, we did not perform experiments to determine the optimal IOU threshold, and this is left as potential future work.

6.1 Preprocssing

Since we need the neural network to interpret words and letters meaningful, we first convert object and and predicate labels into vectors using GLOVE[19], a word-to-vector library commonly used by other works on SGG[14].

We use Visual Genome 1.2, a dataset composed of more than 108 thousand images and 2.3 million relationship predicates [12] as our training and test set. To treat predicates as objects, we assign each annotated predicate to a bounding box that encapsulates both the subject and object bounding boxes. For example, given a pair of objects $A, B \in \mathbb{R}^2$ with predicate r in Visual Genome 1.2, we create a new bounding box R that minimally encloses both A and B. such that Our dataset are separated into wo In order to represent relationships and objects rather than edges between objects, we must add a corresponding predicate bounding box for each relationship triplet. We define the predicate bounding box R as the minimal bounding box that encapsulates both the subject and object bounding boxes. Given a pair of object bounding positions $A, B \in \mathbb{R}^2$ with predicate r, we define the predicate bounding box R as

$$R = (\min(A[1], B[1]), \max(A[2], B[2]))$$

This ensures that each predicate is represented as a single entity in the image which encompasses the subject and object. For example, given a 'man' with bounding box $A = [0$ We use the original scene graph annotations to produce a secondary dataset of predicate bounding boxes, each labeled with its corresponding predicate. We now have two datasets which are composed of the same images but different bounding boxes and labels: one has the bounds and labels for objects, while the other has the bounds and labels of predicates.

Data Splitting We split the dataset as follows. We first sort the predicates found in the dataset by the number of instances associated with each predicate. Next, we separate a number of predicates into their own subset. We determine how many predicates to place into the set by calculating the ratio between the number of a instances containing the most frequently occuring predicate with the number of instances containing the least frequently occuring predicate. It follows that the larger the resulting number, the more unbalanced the dataset. For every subset, we select the largest number of predicates ordered by frequency such that the ratio does not exceed a chosen threshold.

Metric

Cleaning One significant issue with the Visual Genome dataset is that it contains a large number of object and predicate labels which are semantically equivalent to one another (for example, 'under' and 'below'). In a recent release, the maintainers of the dataset has provided us with a collection of synonym sets that group similar objects and predicates into the same class, with one term assigned by group. After converting all synonymous terms to their respective group representative term, we remove objects which rarely appear along with any associated triplets.

6.2 Model training

Object detection models Here we arrive at the long-tailed distribution problem. For objects, we train a YOLO model on objects Object YOLO: Trained on the object dataset, this model detects all objects in a given image. predicate YOLO: Trained on the predicate dataset, this model detects all predicate entities R in the same image. Each model is trained using conventional object detection procedures and loss functions, taking advantage of YOLO's efficient architecture. By isolating objects and predicates into their own detection tasks, we allow each model to specialize and minimize biases that often arise when both tasks are intertwined.

6.3 Inference

We start with weights pro the YOLO v11 model on the COCO dataset and evaluate it on the CoCo Dataset on 5 NVIDIA 4090Ti graphics cards. Training took 25 hours, with a maximum epoch of 300 and a batch size of 8. We evaluate our model using mean Recall@50.

Parallel detection For a given test image, we pass it through both the Object YOLO and the predicate YOLO models independently and in parallel. The Object YOLO output gives us a set of detected objects with corresponding bounding boxes and class labels. The predicate YOLO output provides a set of detected "predicate boxes," each with a predicted predicate label.

Clustering and association we need to determine which objects are involved in each detected predicate. We apply a clustering algorithm to group each predicate bounding box R with candidate object bounding boxes that it encloses or overlaps. This gives us one or more candidate subject-object pairs for each predicate detection.

Triplet selection using MLP scoring After obtaining two sets of detections—one for objects and one for predicates—we need the determine the subject and object associated with each triplet. For every predicate bounding box and label, we collect a list of detected objects have have a large overlap within the predicate box. To select the most like subject-object pair associated with the label, we use a lightweight MLP classifier trained to score the plausibility of a triplet (subject, predicate, object). For each predicate detection, we run the MLP once with every potential object pair that can be formed by objects in the list, selecting the triplet which returns the highest confidence score.

7 Evaluation

7.1 Evaluation Metrics

The task of scene graph generation can be divided into several subtasks, each with varying levels of difficulty. Predicate classification (PredCLS)[3, 9, 5] predicts predicates given the subject and object labels and bounds. Scene graph classification (SGCLS)[3, 5] predicts the triplet labels given the bounding boxes. Scene graph detection (SGDET)[3**?**], also referred to simply as Scene graph generation (SGGen)[9, 5], predicts the labels and bounding boxes of all three elements. To evaluate our model, we follow the common standard set by previous works[] and adopt the evaluation metrics recall@K (R@K) and mean recall@K (mR@K). Given a single image, Recall@k is defined as the fraction of ground truth triplets which can be found in the top k triplet predictions. Two triplets are considered equivalent when all three elements (subject, predicate, and object) have been labelled correctly and both object and subject bounding boxes have IoU > 0.5 compared to the ground truth bounding boxes.

7.2 dataset

7.3 setting

7.4 implementation

7.5 results

8 Conclusion

9 Submission of papers to NeurIPS 2024

Please read the instructions below carefully and follow them faithfully.

9.1 Style

Papers to be submitted to NeurIPS 2024 must be prepared according to the instructions presented here. Papers may only be up to **nine** pages long, including figures. Additional pages *containing only acknowledgments and references* are allowed. Papers that exceed the page limit will not be reviewed, or in any other way considered for presentation at the conference.

The margins in 2024 are the same as those in previous years.

Authors are required to use the NeurIPS LaTeX style files obtainable at the NeurIPS website as indicated below. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

9.2 Retrieval of style files

The style files for NeurIPS and other conference information are available on the website at

`http://www.neurips.cc/`

The file `neurips_2024.pdf` contains these instructions and illustrates the various formatting requirements your NeurIPS paper must satisfy.

The only supported style file for NeurIPS 2024 is `neurips_2024.sty`, rewritten for LaTeX 2_ε. **Previous style files for LaTeX 2.09, Microsoft Word, and RTF are no longer supported!**

The LaTeX style file contains three optional arguments: `final`, which creates a camera-ready copy, `preprint`, which creates a preprint for submission to, e.g., arXiv, and `nonatbib`, which will not load the `natbib` package for you in case of package clash.

Preprint option If you wish to post a preprint of your work online, e.g., on arXiv, using the NeurIPS style, please use the `preprint` option. This will create a nonanonymized version of your work with the text "Preprint. Work in progress." in the footer. This version may be distributed as you see fit, as long as you do not say which conference it was submitted to. Please **do not** use the `final` option, which should **only** be used for papers accepted to NeurIPS.

At submission time, please omit the `final` and `preprint` options. This will anonymize your submission and add line numbers to aid review. Please do *not* refer to these line numbers in your paper as they will be removed during generation of camera-ready copies.

The file `neurips_2024.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in Sections 10, 11, and 12 below.

10 General formatting instructions

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing (leading) of 11 points. Times New Roman is the preferred typeface throughout, and will be selected for you by default. Paragraphs are separated by ½ line space (5.5 points), with no indentation.

The paper title should be 17 point, initial caps/lower case, bold, centered between two horizontal rules. The top rule should be 4 points thick and the bottom rule should be 1 point thick. Allow ¼ inch space above and below the title to rules. All pages should start at 1 inch (6 picas) from the top of the page.

For the final version, authors' names are set in boldface, and each name is centered above the corresponding address. The lead author's name is to be listed first (left-most), and the co-authors' names (if different address) are set to follow. If there is only one co-author, list both author and co-author side by side.

Please pay special attention to the instructions in Section 12 regarding figures, tables, acknowledgments, and references.

11 Headings: first level

All headings should be lower case (except for first word and proper nouns), flush left, and bold.

First-level headings should be in 12-point type.

11.1 Headings: second level

Second-level headings should be in 10-point type.

11.1.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

12 Citations, figures, tables, references

These instructions apply to everyone.

12.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

```
http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf
```

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

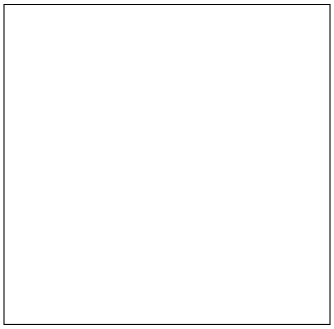

Figure 1: Sample figure caption.

Hasselmo, et al. (1995) investigated...

If you wish to load the `natbib` package with options, you may add the following before loading the `neurips_2024` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{neurips_2024}
```

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]," not "In our previous work [4]." If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form "A. Anonymous" and include a copy of the anonymized paper in the supplementary material.

12.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

Note that footnotes are properly typeset *after* punctuation marks.[2]

12.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

12.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

[1] Sample of the first footnote.

[2] As in this example.

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 1.

12.5 Math

Note that display math in bare TeX commands will not create correct line numbers for submission. Please use LaTeX (or AMSTeX) commands for unnumbered display math. (You really shouldn't be using $$ anyway; see `https://tex.stackexchange.com/questions/503/why-is-preferable-to` and `https://tex.stackexchange.com/questions/40492/what-are-the-differences-between-align-equation-and-displaymath` for more information.)

12.6 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

13 Preparing PDF files

Please prepare submission files with paper size "US Letter," and not, for example, "A4."

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

  ```
  \usepackage{amsfonts}
  ```

  followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

  ```
  \newcommand{\RR}{I\!\!R} %real numbers
  \newcommand{\Nat}{I\!\!N} %natural numbers
  \newcommand{\CC}{I\!\!\!\!C} %complex numbers
  ```

  Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

13.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

Acknowledgments and Disclosure of Funding

Use unnumbered first level headings for the acknowledgments. All acknowledgments go at the end of the paper before the list of references. Moreover, you are required to declare funding (financial activities supporting the submitted work) and competing interests (related financial activities outside the submitted work). More information about this disclosure can be found at: `https://neurips.cc/Conferences/2024/PaperInformation/FundingDisclosure`.

Do **not** include this section in the anonymized submission, only in the final paper. You can use the `ack` environment provided in the style file to automatically hide this section in the anonymized submission.

References

References follow the acknowledgments in the camera-ready paper. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

References

[1] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *European conference on computer vision* (2020), Springer, pp. 213–229.

[2] Chiou, M.-J., Ding, H., Yan, H., Wang, C., Zimmermann, R., and Feng, J. Recovering the unbiased scene graphs from the biased ones. In *Proceedings of the 29th ACM International Conference on Multimedia* (2021), pp. 1581–1590.

[3] Cong, Y., Yang, M. Y., and Rosenhahn, B. Reltr: Relation transformer for scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 9 (2023), 11169–11183.

[4] Desai, A., Wu, T.-Y., Tripathi, S., and Vasconcelos, N. Learning of visual relations: The devil is in the tails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 15404–15413.

[5] Dornadula, A., Narcomey, A., Krishna, R., Bernstein, M., and Li, F.-F. Visual relationships as functions: Enabling few-shot scene graph prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2019), pp. 0–0.

[6] Gao, L., Wang, B., and Wang, W. Image captioning with scene-graph based semantic concepts. In *Proceedings of the 2018 10th international conference on machine learning and computing* (2018), pp. 225–229.

[7] GU, J., JOTY, S., CAI, J., ZHAO, H., YANG, X., AND WANG, G. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 10323–10332.

[8] GU, J., ZHAO, H., LIN, Z., LI, S., CAI, J., AND LING, M. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 1969–1978.

[9] JIN, T., GUO, F., MENG, Q., ZHU, S., XI, X., WANG, W., MU, Z., AND SONG, W. Fast contextual scene graph generation with unbiased context augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 6302–6311.

[10] JOHNSON, J., KRISHNA, R., STARK, M., LI, L.-J., SHAMMA, D., BERNSTEIN, M., AND FEI-FEI, L. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3668–3678.

[11] JUNG, D., KIM, S., KIM, W. H., AND CHO, M. Devil's on the edges: Selective quad attention for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 18664–18674.

[12] KRISHNA, R., ZHU, Y., GROTH, O., JOHNSON, J., HATA, K., KRAVITZ, J., CHEN, S., KALANTIDIS, Y., LI, L.-J., SHAMMA, D. A., ET AL. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision 123* (2017), 32–73.

[13] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep learning. *nature 521*, 7553 (2015), 436–444.

[14] LEE, C.-W., FANG, W., YEH, C.-K., AND WANG, Y.-C. F. Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 1576–1585.

[15] LI, H., ZHU, G., ZHANG, L., JIANG, Y., DANG, Y., HOU, H., SHEN, P., ZHAO, X., SHAH, S. A. A., AND BENNAMOUN, M. Scene graph generation: A comprehensive survey. *Neurocomputing 566* (2024), 127052.

[16] LI, R., ZHANG, S., AND HE, X. Sgtr: End-to-end scene graph generation with transformer. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 19486–19496.

[17] LI, R., ZHANG, S., WAN, B., AND HE, X. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 11109–11119.

[18] LIANG, X., LEE, L., AND XING, E. P. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 848–857.

[19] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.

[20] PLUMMER, B. A., SHIH, K. J., LI, Y., XU, K., LAZEBNIK, S., SCLAROFF, S., AND SAENKO, K. Revisiting image-language networks for open-ended phrase detection. *IEEE transactions on pattern analysis and machine intelligence 44*, 4 (2020), 2155–2167.

[21] REDMON, J. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016).

[22] REDMON, J., AND FARHADI, A. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 7263–7271.

[23] SOHAN, M., SAI RAM, T., REDDY, R., AND VENKATA, C. A review on yolov8 and its advancements. In *International Conference on Data Intelligence and Cognitive Informatics* (2024), Springer, pp. 529–545.

[24] Tang, K., Niu, Y., Huang, J., Shi, J., and Zhang, H. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 3716–3725.

[25] Terven, J., Córdova-Esparza, D.-M., and Romero-González, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction 5*, 4 (2023), 1680–1716.

[26] Tripathi, S., Bhiwandiwalla, A., Bastidas, A., and Tang, H. Using scene graph context to improve image generation. *arXiv preprint arXiv:1901.03762* (2019).

[27] Ultralytics. Ultralytics yolo11 open-sourced, 2024.

[28] Yan, S., Shen, C., Jin, Z., Huang, J., Jiang, R., Chen, Y., and Hua, X.-S. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In *Proceedings of the 28th ACM international conference on multimedia* (2020), pp. 265–273.

[29] Yang, J., Lu, J., Lee, S., Batra, D., and Parikh, D. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 670–685.

[30] Yu, J., Chai, Y., Wang, Y., Hu, Y., and Wu, Q. Cogtree: Cognition tree loss for unbiased scene graph generation. *arXiv preprint arXiv:2009.07526* (2020).

[31] Zhang, C., Chao, W.-L., and Xuan, D. An empirical study on leveraging scene graphs for visual question answering. *arXiv preprint arXiv:1907.12133* (2019).

[32] Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. Object detection in 20 years: A survey. *Proceedings of the IEEE 111*, 3 (2023), 257–276.

A Appendix / supplemental material

Optionally include supplemental material (complete proofs, additional experiments and plots) in appendix. All such materials **SHOULD be included in the main submission.**

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: **[TODO]**

    Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.