SelfGraphVQA: A Self-Supervised Graph Neural Network for Scene-based Question Answering

Bruno Souza*
University of Campinas
b234837@dac.unicamp.br

Marius Aasan
University of Oslo
mariuaas@uio.no

Helio Pedrini
University of Campinas
helio@ic.unicamp.br

Adín Ramírez Rivera
University of Oslo
adinr@uio.no

Abstract

The intersection of vision and language is of major interest due to the increased focus on seamless integration between recognition and reasoning. Scene graphs (SGs) have emerged as a useful tool for multimodal image analysis, showing impressive performance in tasks such as Visual Question Answering (VQA). In this work, we demonstrate that despite the effectiveness of scene graphs in VQA tasks, current methods that utilize idealized annotated scene graphs struggle to generalize when using predicted scene graphs extracted from images. To address this issue, we introduce the SelfGraphVQA framework. Our approach extracts a scene graph from an input image using a pre-trained scene graph generator and employs semantically-preserving augmentation with self-supervised techniques. This method improves the utilization of graph representations in VQA tasks by circumventing the need for costly and potentially biased annotated data. By creating alternative views of the extracted graphs through image augmentations, we can learn joint embeddings by optimizing the informational content in their representations using an un-normalized contrastive approach. As we work with SGs, we experiment with three distinct maximization strategies: node-wise, graph-wise, and permutation-equivariant regularization. We empirically showcase the effectiveness of the extracted scene graph for VQA and demonstrate that these approaches enhance overall performance by highlighting the significance of visual information. This offers a more practical solution for VQA tasks that rely on SGs for complex reasoning questions.

1. Introduction

The successful execution of Visual Question Answering (VQA) relies on a comprehensive understanding of the scene, including spatial interrelationships and reasoning inference capabilities [1, 14]. Incorporating scene graph (SG) representations in SG-VQA tasks has shown promising outcomes [13, 16, 18, 25, 32], providing concise representations of complex spatial and relational information.

Earlier investigations into SG-VQA demonstrated that successful models primarily rely on the utilization of manually annotated scene graphs for training [20, 21, 25], resulting in remarkably high levels of accuracy on the GQA dataset [14], surpassing human performance by a significant margin (see Table 1).

Despite the promising results, we argue that utilizing pre-annotated SGs in VQA is impractical in the real world due to its labor-intensive nature. Also, it permits a wide range of semantically corresponding SG [12] and when annotated it could potentially introduce questions-related biases, giving rise to concerns about its generalizability [2]. These issues may limit the model's ability to solve real-world problems beyond the dataset [23]. This is evident in a significant decline in accuracy, approximately 60% when models are confronted with automatically generated SGs. Additionally, studies assert that the main limitation in generalizing stems largely from linguistic correlations. [2, 17].

In this study, we address these challenges by extracting an SG from a given image using an unbiased, off-the-shelf scene graph generator [16], with the aim of removing any potential information leakage, as illustrated in Fig. 1's structure. Furthermore, our method employs semantically preserving augmentation, integrated with un-normalized contrastive framework, to further mitigate potential linguistic biases to enhance the visual cues translated as SG for VQA. We refer to it as the *SelfGraphVQA framework*, cf. Fig. 1.

Given its simplicity [7], our approach is trained using joint embeddings and a Siamese network architecture, inspired by the SimSiam model, which does not require negative samples [5, 9]. In this work, we explore three un-normalized contrastive approaches (node-wise, graph-wise, and regularization for permutation equivariance) and demonstrate its effectiveness by enhancing the visual information for the VQA task. A graph neural network (GNN)

*Work carried out as Guest Researcher at UiO.

To appear in Vision-and-Language Algorithmic Reasoning Workshop at ICCV 2023

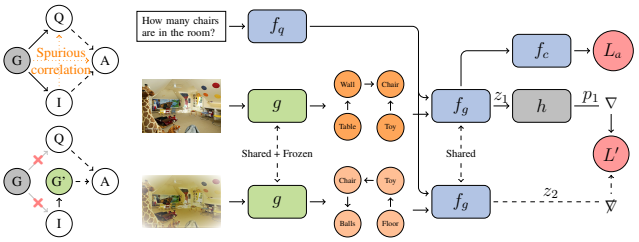


Figure 1: (Left) The statistical dependence of the task and the ideal graph, G. (Right) Our proposed framework removes data leakage by using the extracted SG G'. Our architecture comprises a question encoder f_q, a graph encoder f_g, and a classifier f_c. Two distinct views of one image are processed by the same pipeline. We use a frozen pre-trained SG generator g, and a prediction head h is applied through the top view with gradient backpropagation, while gradients are not propagated back from the lower view. We maximize the representation of the views using the similarity loss L'.

with a self-attention strategy (GAT) is employed to distill an SG representation relevant to the question by capturing visual interaction content among objects in the scene [7].

Our work differs from existing VQA models in three main aspects: (i) we generate as SG using a pre-trained, unbiased scene graph generator [16] in a more practical approach; (ii) we utilize un-normalized contrastive learning on the SG representation, along with augmentation, to eliminate any potential spurious correlations from annotated data and to heighten the visual information; and (iii) the use of a GAT encoder to enhance high-level semantic and spatial reasoning on the SG. We further investigate the behavior of visual enhancement when employing a more expressive language encoder, specifically BERT [15]. Importantly, our SelfGraphVQA framework does not require the costly pre-training strategy common to transformer-based models commonly used in vision-language tasks [8, 28, 32].

2. Related Work

Scene Graph and Visual Question Answering. Accurately assessment of VQA tasks, requiring a comprehensive understanding of visual perception and semantic reasoning, has gained substantial attention in the academic community, as these tasks holds significant practical value, particularly in enhancing accessibility for the visually impaired [4, 15, 19, 33, 34].

Several works have explored the information that SG representations may bring to VQA [20, 31], as opposed to the more data-hungry transformer-based visual language models [8, 19, 28]. However, existing SG-VQA approaches typically rely on idealized scene graphs and inherent dataset reasoning [20, 21]. Obtaining such annotations can be costly without an end-to-end pipeline. Moreover, even SoTA methods in SG-VQA exhibit limited generalization capabilities, potentially due to spurious correlations [2].

Table 1: Our experiments revealed a notable accuracy reduction in top-notch methods on the GQA dataset when transitioning from well-annotated to extracted scene graphs. We categorize methods by data type (e.g., annotated data or purely image-question extraction) and SGG usage. All methods are trained and validated uniformly, except for the test extracted configuration, trained on ideal data and validated on extracted SGG data.

Method	Eval. Data	Acc (%)
Human [14]	–	89.3
GraphVQA [20]	Annotated/SGG	94.8
LRTA [21]	Annotated/SGG	93.1
Lightweight [25]	Annotated/SGG	77.9
CRF [24]	Annotated	72.1
LXMERT [28]	Extracted	59.8
GraphVQA (original pre-trained on ideal)	**Test Extracted/SGG**	29.7
SelfGraphVQA (Local)	Extracted/SGG	51.5
SelfGraphVQA (Global)	Extracted/SGG	52.3
SelfGraphVQA (SelfSim)	Extracted/SGG	54.0

Self-Supervised Learning. Broadly speaking, recent advancements in self-supervised learning can be categorized into normalized [3, 6] and maximization representation learning [7, 11, 29]. Contrastive methods aim to bring embeddings of identically labelled images closer together while separating embeddings generated from different versions. In visual-language data, the prevailing approach for self-supervised learning involves pretraining a transformer-based model on a large dataset to solve pretext tasks before fine-tuning for downstream tasks [8, 27, 28, 32]. However, these methods can be computationally expensive and complex due to the use of negative samples and masking techniques. Modern un-normalized contrastive learning methods, e.g., BYOL [11] and SimSiam [7], use architectures inspired by reinforcement learning to maximize the informational content of the representations. In our proposal, we adopt a similarity maximization approach using a Siamese architecture for visual scene graph representation.

3. Methodology

We refer the reader to the appendix for the implementation details. We experiment with the maximization strategy with three independent and distinct similarity losses over either a localized node representation (i.e., object-wise), a global pooled graph representation (i.e., scene-wise), or a regularization node representation term to induce permutation equivariance. We denote the graph representations $z_i = f_g\big(g(x_i), f_q(q)\big)$, and the predictor's output vectors $p_i = h(z_i)$. Generally, the representations are maximized by minimizing the generic cosine distance D loss.

Local Similarity. To account for permutation invariance in the node representations, we compute cosine distances over all object pairs from the two views and use the maximally

similar node embedding pairs to compute the local loss by

$$L_\ell^*(p_1, z_2) = \frac{1}{O}\sum_i^O \arg\min_{z_{2,j}} D(p_{1,i}, z_{2,j}), \quad (1)$$

where O is the number of objects in the scene. Symmetrically, we compute $L_\ell^*(p_2, z_1)$, to obtain the overall local loss

$$L_\ell(z_1, z_2) = \frac{1}{2}\big(L_\ell^*(p_1, z_2) + L_\ell^*(p_2, z_1)\big). \quad (2)$$

Global Similarity. After obtaining a graph representation, we follow an approach similar to cosine similarity maximization for image classification [7, 11]. Along with the intuition that contrasting between global representations may enhance the visual cues, we assume that the global representation contains the full information about the scene. Similar to the local representation, we minimize the cosine distance, yielding a loss on the form

$$L_g(z_1, z_2) = \frac{1}{2}\big(D(p_1, z_2) + D(p_2, z_1)\big). \quad (3)$$

Regularization for Permutation Equivariance. We employ an *anchor*, where the SG of an unmodified image guides the SG of the augmented image, allowing us to obtain a more accurate representation of the original scene. Our assumption is that the local similarity loss decreases the global performance, while global similarity provides a contextual representation but loses local details. This technique aligns similar nodes and encourages regularization, making augmented scene representations closer to the original, thus mitigating permutation invariance in graph representations.

Denote the anchored representation by z_1, and the augmented representation by z_2. We determine intra-similarities of the anchors $s_{1,i} = \arg\min_{z_{1,j}} D(z_{1,i}, z_{1,j})$ and similarities of augmented views $s_{2,ij} = D(z_{2,i}, z_{2,j})$. We then compute cross-entropy (CE) between anchors and augmentations

$$J(z_1, z_2) = \mathrm{CE}(s_1, s_2), \quad (4)$$

which acts as a regularizer to constrain permutation equivariance for the augmentations in addition to the local loss. We combine these losses using

$$L_s(z_1, z_2) = L_\ell(z_1, z_2) + J(z_1, z_2), \quad (5)$$

which we refer to as a local self-similarity loss (SelfSim).

Distribution Link Representation Regularization. Similarly to the regularization for permutation equivariance, we apply link regularization *in conjunction with one of the other three similarity strategies*. The edges of the *anchor* SG guide the edges of the augmented SG. Denote the anchored edge score representation by r_1, and the augmented edge score representation by r_2. These scores characterize the relationship between the objects in the scene, and we aim to make the link distribution more robust to perturbation. *In this case, the scene graph generator [16] is trainable.* We compute the cross-entropy between the anchored edge scores and the augmented edge scores $J_e(r_1, r_2) = \mathrm{CE}(r_1, r_2)$, which acts as a regularizer to constrain the link prediction distribution, yielding

$$L_e(z_1, z_2) = L_\ell(z_1, z_2) + J_e(r_1, r_2). \quad (6)$$

All models utilizing this added link distribution regularizer are characterized by the inclusion of the term "link."

Overall Optimization Objective. Lastly, we outline the overall loss for optimizing the VQA objective. To identify the correct answer $a \in A$ given an example (x, q, A), where x represents the input image, and q is the associated question, we extract a point estimate of probabilities

$$p(a \mid x, q) = \sigma\left(\mathrm{logit}(x)\right), \quad (7)$$

where σ is the softmax function, and $\mathrm{logit}(x) = f(x, q)$ are the logits for all possible answers produced by our encoder. We calculate the cross-entropy loss for each instance,

$$L_{sup}(x) = \mathrm{CE}\left(p(a \mid x, q), a\right). \quad (8)$$

Our combined training loss is then given by

$$L(x) = \alpha L_{sup}(x) + \beta L'(z_1, z_2), \quad (9)$$

where L' can be any of the aforementioned similarity loss strategies: L_ℓ, L_g, or L_s, with or without L_e. The α and β are controlled hyperparameters that balance the contribution of the various components in the total loss.

4. Experiments and Ablations

We evaluate our framework on the GQA dataset [14]. Our study aims to establish a practical foundation for demonstrating the potential of SG along with an un-normalized contrasting approach to improve visual cues for VQA. Despite the noise data in the extracted SG, we demonstrate its effectiveness, Fig. 2, by highlighting the importance of further exploration. The utilization of non-idealized SG-VQA methods with un-normalized contrastive learning leads to improvements across all metrics, Table 2. Furthermore, our framework demonstrates faster convergence during training, approximately 20% faster in epochs compared to baselines. However, further investigation is required to validate them.

The un-normalized contrastive approach universally enhances results across question categories (Fig. 2), with specific types of approaches further improving the model's performance based on the query type.

Table 2: Results (%) on GQA by standard metrics.

Method	Binary (↑)	Open (↑)	Consist. (↑)	Validity (↑)	Plausab. (↑)	Distr. (↓)	Acc (↑)
Baseline	65.8	29.7	58.2	94.9	90.5	11.7	50.1
Baseline+BERT	68.0	32.2	62.6	95.0	90.9	7.7	53.8
Local	66.8	30.2	59.4	94.9	90.6	8.8	51.5
Global	67.7	30.8	62.5	94.9	90.6	6.7	52.3
SelfSim	**68.4**	31.3	**65.9**	94.9	90.7	**2.1**	54.0
Global+BERT+link	68.0	**33.0**	63.9	95.0	**91.2**	8.9	**54.5**
SelfSim+BERT+link	68.2	32.8	64.3	**95.0**	91.0	8.0	**54.5**

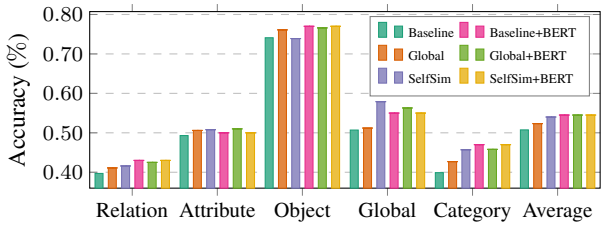


Figure 2: Accuracy on different question types.

Table 3: Change in accuracy under potentially disruptive augmentations and perturbations.

Question Type	Augmentation	Baseline	Global	Local	SelfSim
Relation	Flip	−1.6	−3.4	−3.2	−3.9
Attribute	Strong Color Jitter	+1.14	−3.7	−0.8	−1.2
Global	Gaussian Noise + Crop	−5.6	−7.7	−5.5	−8.1

Table 4: Results(%) of the Aug. Baseline and SelfSim.

Method	Binary	Open	Validity	Plausibility	Acc
Baseline Aug	65.1	28.7	94.6	90.1	50.1
SelfSim	68.4	31.3	94.9	90.7	54.0

We conducted ablations to demonstrate the functionality of our approach and carried out detailed observations that go beyond mere reliance on metrics using the GQA dataset.

Does the Scene Graph Really Matter? Through a perturbation study where images were augmented based on question types, we introduced disruptive noise such as image flipping to challenge the model's ability to answer spatial relational questions. The goal was to observe mistakes in the model's answers. The results, compared to the baseline (Table 3), showed greater variation in our model's performance, indicating that it pays more attention to visual information, whereas the baseline appears to rely on other sources of information.

Are Performance Gains Mainly Due to Augmentations? We compared our approach with the baseline architecture, training solely with data augmentation techniques to evaluate their influence on overall performance. Table 4 provides evidence that data augmentation techniques actually impair the performance of the architecture.

Are Our Models Less Biased? Our initial hypothesis was that current top-performing models might incorporate biases present in the questions into their weights. We conducted experiments to analyze this issue, introducing random noise to features in the scene graph while preserving its topology, and perturbing the language in up to 50% of the words in the questions. The results in Table 5 demonstrate that our approach relies less on linguistic features, prioritizing overall information and reducing linguistic bias.

Table 5: Sensitivity of accuracy (%) for bias question analyzes of SelfGraphVQA and SelfGraphVQABERT.

Setup	**Methods**			
Scene Graph + Question	Baseline	Local	Global	SelfSim
Noise + SG	16.2	16.6	28.6	26.6
Question + Noise	39.9	38.3	37.4	39.8
Noise + Noise	12.7	14.6	18.9	21.0
Question + Scene Graph	BERT Baseline	BERTGlobal+link	BERTSelfSim+link	
Noise + SG	21.0	23.2	24.5	
Question + Noise	42.4	41.8	42.8	
Noise + Noise	19.8	21.7	21.3	

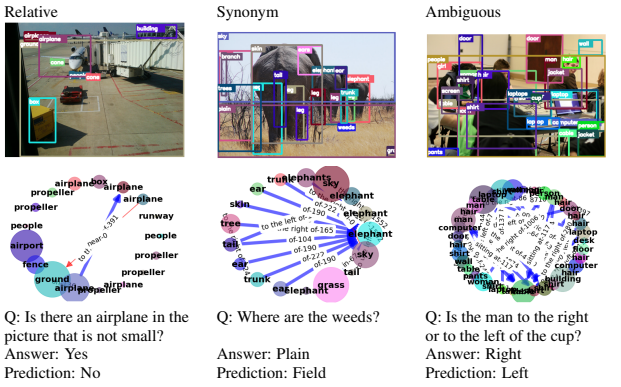


Figure 3: Examples to demonstrate the complexity of VQA.

Additionally, we explored visual enhancement, even when trained with a more expressive language module such as BERT. The experiments in Table 5 examine the impact of using BERT and its effect on enhancing visual information.

Examples. Given the wide range of acceptable answers, we argue that solely relying on standard evaluation metrics may not provide a fair comparison, thus presenting additional challenges to the field. Fig. 3 demonstrates the utility of SG for interpretability, as they enable a graphical analysis of objects and the overall composition of the scene.

5. Conclusions

Despite promising results in VQA tasks with idealized SG, our study revealed that models relying on manually annotated and expensive SG struggle with real-world data. To address this, we proposed SelfGraphVQA, a more practical SG-VQA framework that breaks the spurious correlation of annotated SG and learns to answer questions using extracted SG from a pre-trained SG generator. We employed un-normalized contrastive learning to maximize similar graph representations in different views. All approaches utilizing self-supervision showed improvement over their baselines. Overall, we demonstrated the effectiveness of extracted SG in VQA, underscoring the significance of continued exploration of the potential of SG for complex tasks. We also showed that self-supervision over the SG representation improved the results by enhancing the visual information within the task. We hope that this work raises awareness of the challenges of accentuating the role of the scene in answering questions from images.

Acknowledgements

This work was supported in part by the FAPESP (São Paulo Research Foundation) grant no. 2022/09849-8 and BEPE grant no. 2022/09849-8. The computations were performed in part on resources provided by Sigma2—the National Infrastructure for High-Performance Computing and Data Storage in Norway—through Project NN8104K. This work was funded in part by the Research Council of Norway, via the Centre for Research-based Innovation funding scheme (grant no. 309439), and Consortium Partners.

References

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4971–4980, 2018. 1

[2] Aishwarya Agrawal, Ivana Kajic, Emanuele Bugliarello, Elnaz Davoodi, Anita Gergely, Phil Blunsom, and Aida Nematzadeh. Reassessing evaluation practices in visual question answering: A case study on out-of-distribution generalization. In *Conf. European Ch. Assoc. Comput. Ling. (EACL)*, pages 1171–1196, 2023. 1, 2, 6

[3] Laurence Aitchison. InfoNCE is a variational autoencoder. *arXiv preprint arXiv:2107.02495*, 2021. 2

[4] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2612–2620, 2017. 2

[5] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 6, 1993. 1

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 15750–15758, 2021. 1, 2, 3, 7, 8

[8] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conf. Comput. Vis. (ECCV)*, pages 104–120. Springer, 2020. 2

[9] Quentin Garrido, Yubei Chen, Adrien Bardes, Laurent Najman, and Yann Lecun. On the duality between contrastive and non-contrastive self-supervised learning. *arXiv preprint arXiv:2206.02574*, 2022. 1

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 6904–6913, 2017. 6

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 33:21271–21284, 2020. 2, 3, 7, 8

[12] Xuanli He, Quan Hung Tran, Gholamreza Haffari, Walter Chang, Trung Bui, Zhe Lin, Franck Dernoncourt, and Nhan Dam. Scene graph modification based on natural language commands. *arXiv preprint arXiv:2010.02591*, 2020. 1

[13] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 10294–10303, 2019. 1

[14] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. *10.48550/arxiv.1902.09506*, 2019. 1, 2, 3, 6, 7

[15] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 31, 2018. 2, 8

[16] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Graph density-aware losses for novel compositions in scene graph generation. *arXiv preprint arXiv:2005.08230*, 2020. 1, 2, 3, 6, 7, 8

[17] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Inter. Conf. Mach. Learn. (ICML)*, pages 2873–2882. PMLR, 2018. 1

[18] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-Aware Graph Attention Network for Visual Question Answering. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 10313–10322, 2019. 1

[19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2

[20] Weixin Liang, Yanhao Jiang, and Zixuan Liu. GraghVQA: Language-guided graph neural networks for graph-based visual question answering. In *Wksp. Multimodal Artif. Intell. (ACLW)*, pages 79–86, Mexico City, Mexico, June 2021. Association for Computational Linguistics. 1, 2, 6, 7, 8

[21] Weixin Liang, Feiyang Niu, Aishwarya Reganti, Govind Thattai, and Gokhan Tur. LRTA: a transparent neural-symbolic reasoning framework with modular supervision for visual question answering. *arXiv preprint arXiv:2011.10731*, 2020. 1, 2, 6, 7, 8

[22] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, volume 32, 2019. 6

[23] Man Luo, Shailaja Keyur Sampat, Riley Tallman, Yankai Zeng, Manuha Vancha, Akarshan Sajja, and Chitta Baral. 'just because you are right, doesn't mean i am wrong': Overcoming a bottleneck in the development and evaluation of open-ended visual question answering (VQA) tasks. *arXiv preprint arXiv:2103.15022*, 2021. 1

[24] Binh X Nguyen, Tuong Do, Huy Tran, Erman Tjiputra, Quang D Tran, and Anh Nguyen. Coarse-to-fine reasoning for visual question answering. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4558–4566, 2022. 2, 6

[25] Sai Vidyaranya Nuthalapati, Ramraj Chandradevan, Eleonora Giunchiglia, Bowen Li, Maxime Kayser, Thomas Lukasiewicz, and Carl Yang. Lightweight visual question answering using scene graphs. In *ACM Inter. Conf. Inf. Knowl. Manag. (CIKM)*, pages 3353–3357, 2021. 1, 2

[26] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, pages 1532–1543, 2014. 7

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Inter. Conf. Mach. Learn. (ICML)*, pages 8748–8763. PMLR, 2021. 2

[28] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 2

[29] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Remi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *Wksp. Geom. Topol. Represent. Learn. (ICLRW)*, 2021. 2

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 7

[31] Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec. VQA-GNN: Reasoning with multimodal semantic graph for visual question answering. *arXiv preprint arXiv:2205.11501*, 2022. 2

[32] Zhecan Wang, Haoxuan You, Liunian Harold Li, Alireza Zareian, Suji Park, Yiqing Liang, Kai-Wei Chang, and Shih-Fu Chang. SGEITL: Scene graph enhanced image-text learning for visual commonsense reasoning. In *AAAI Conf. Artif. Intell. (AAAI)*, volume 36, pages 5914–5922, 2022. 1, 2

[33] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 16375–16387, 2022. 2

[34] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting visual representations in vision-language models. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 5579–5588, June 2021. 2

A. Datasets

We evaluate our SelfGraphVQA frameworks on the GQA dataset [14]. GQA is another large-scale effort (22M questions, each with one answer) that focuses on the compositionality of template-generated questions for real-world images. We use the official train/validation split of GQA.

Table A.1: Detailed statistics for the GQA dataset examined in our study compared to other possible statistics and the original paper dataset.

	Answers Candidates
Ours	1878
Alternative [2, 22, 24]	1533
Original [10, 14]	1878

The GQA dataset was selected for evaluation because it includes complex relational and spatial questions that require multiple reasoning skills, spatial understanding, and multi-step inference. These characteristics make it more challenging compared to previous visual question-answering datasets. Consequently, the GQA dataset is well-suited for evaluating the performance of scene graph models.

In contrast to prior studies [2, 24], our approach takes a simplistic approach by considering solely the ground truth distribution as a potential answer in the training dataset as a candidate for the answer distribution, without any filter techniques. Table A.1 provides detailed statistics for each dataset examined in our investigation.

Despite the substantial variations in the answering classes, we emphasize that our method proves to be effective and comparable to other existing approaches. In addition to the aforementioned points, this further highlights the fact that VQA is a complex and expansive challenge that lends itself to various approaches and needs continued exploration and refinement.

B. Baseline Architecture

Figure B.1 depicts the overall components of our baseline architecture. The unbiased pre-trained scene graph generator model utilized in this study originates from the research paper authored by Knyazev et al. [16]. Applying a density-normalized edge loss to the model, the authors contend that the model is aware of the graph density and, therefore, generalizes better even to rare compositions.

In our project, the frozen weights pre-trained Scene Graph Generator takes the image information and generates a scene graph representation. The Question Encoder receives the instructions and provides them to the GNN-based encoder. Each layer of the module pays attention to these instructions in order to update its hidden node states. The Classifier then takes the graph representation and the question vector concatenates them, and predicts the correct answer.

We use the similar architecture of the state-of-the-art graph-based GraphVQA model [20] and LRTA [21] over the GQA dataset as a baseline for our experiments, with some modifications in order to reduce the dependence on

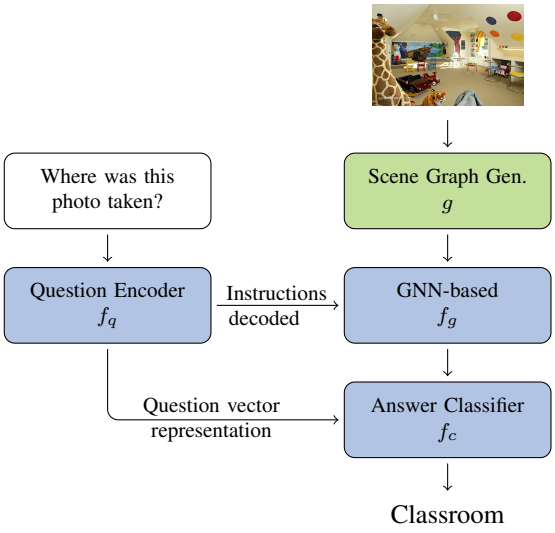


Figure B.1: The baseline architecture.

the annotated available data, as we aim to mitigate the limitations imposed by data availability and enhance the model's generalizability.

For practical purposes, the functional program instructions accompanying each question in the GQA dataset [14] are not necessarily available for inference on real-world data, so we train our decoder to decode the instructions from the question itself. These additional labels are processed by the reasoning module in the GraphVQA model which we explicitly omit in our baseline, as we are more interested in generalizability and real-world performance rather than expressively *solving* the GQA dataset.

In addition, we omit the pre-processing using the scene graph encoding module of the original GraphVQA, as the scene graph generation model g was selected to extract high quality SG-representations. Here, our f_g module is a graph attention network (e.g. GAT) [30].

In the GloVE embedding design, both the query encoder f_q and the graph encoder f_g designs are shared between the original baseline and our proposed modified model. Whereas in the BERT design, we only take the similarity of the graph encoder module f_g design, as our query encoder f_q and the language embedding is a BERT model. By adapting the similar SoTA architecture strategy to the specific design choices of each model, we aim to evaluate the performance and effectiveness of our proposed approach.

C. Architecture Details

Within this section, we aim to provide additional details regarding all components of our implementation approaches.

To ensure clarity and facilitate better comprehension, we have divided this section into two subsections: one discussing the utilization of GloVE word embedding along with a transformer-based model for the question encoder, and the other focusing on the application of BERT for word embedding and the question encoder.

Table C.1 provides a comprehensive overview of the two approaches employed in this study.

It is worth mentioning that the scene graph generator module has its weights frozen in all training approaches, except when we employ the Distribution Link Representation Regularization technique.

C.1. GloVE Word Embedding and Transformer-based Question Encoder

The images are fed through a pre-trained scene graph generator g from [16] work that generates scene graphs from images on the fly.

Except for the pooled graph-level representation (i.e., the module that feeds the classifier), which has a dimension size of 512, all node and edge features have dimension size 300.

The word embedding for the transformed-based query encoder module f_q has its initial weights initialized by using embeddings from GloVe [26]. Both hidden states and word embedding vectors have a dimension size of 300. The question representation is produced by the transformed-based question encoder.

Following [20, 21] work, we adopt a hierarchical sequence generation design, i.e., a transformer decoder model first parses the question into a sequence of M instruction vectors, $[i_1, i_2, \ldots, i_M]$. The i-th instruction vector will correspond exactly to the i-th execution step processed by the GNN encoder f_g module. In our experiments, we force M equals five. We note that SelfGraphVQA does not require any explicit supervision on how to solve the instruction step from the question, and we only supervise the final answer prediction.

For the un-normalized contrastive approach, the MLP prediction head h plays a crucial role in our model architecture. It comprises three fully connected layers, each followed by batch normalization and ReLU activation, except for the final layer. This setup ensures non-linearity and facilitates effective feature extraction. It is important to note that the MLP prediction head is exclusively utilized during the training phase and is subsequently discarded during inference, which aligns with prevailing practices in contemporary self-supervised training methods [7, 11].

The classification module f_c is another integral component of our model. It is designed as a two-layer MLP with a dropout rate of 0.2 and ELU activation.

As explained in Section 3, we independently apply the three self-supervised losses (i.e., local similarity, global similarity, and regularization for permutation equivariance) and compared performances. Our experimental choices

Table C.1: Detailed dimensions used in our study when employing the GloVE and BERT approaches.

Methods	Word dim.	Question dim	Node Dim	Link Dim	Graph dim
GloVE+Transf	300	300	300	300	512
BERT	756	512	512	512	512

were designed to minimize possible biases in the evaluation of our proposed framework.

Both anchored and augmented scene graphs along with the question ground on the scene feed our encoder model to infer a predicted answer. For a fair comparison, we train most of our model from scratch, except for the pre-trained scene graph generator g, whose weights are frozen.

C.2. BERT Word Embedding and Question Encoder

In this case, we employ the BERT model as our word embedding approach and the question encoder, as being a more expressive language model.

Once again, the images are fed through a pre-trained scene graph generator g from [16] work that generates scene graphs from images on the fly. In this particular case, all graph-level and node-level representations possess a dimension size of 512, encompassing both node and edge features. This configuration is deliberately chosen to ensure that the dimensions of the representations closely align with the dimension yielded by BERT word embedding, which is 756. By maintaining consistency in the dimensionality across different components, we aim to facilitate seamless integration and compatibility with BERT-based models.

The word embedding for the BERT query encoder f_q has its initial weights initialized by using embeddings from BERT [15]. Both hidden states and word embedding vectors have a dimension size of 512. The final question representation is derived by taking the average of all word embedding representations generated by BERT.

Following the same approach of [20, 21], we adopt a hierarchical sequence generation design, i.e., a transformer decoder module first parses the encoded question into a sequence of M instruction vectors, $[i_1, i_2, \ldots, i_M]$. The i-th instruction vector will correspond exactly to the i-th execution step processed by the GNN encoder f_g module. In our experiments, we force M equals five. We note that SelfGraphVQA does not require any explicit supervision on how to solve the instruction step from the question, and we only supervise the final answer prediction.

In this scenario, we employ two self-supervised loss techniques: global similarity and regularization for permutation equivariance. Additionally, we incorporate the Distribution Link Representation Regularization method overall approaches performed in this case. It is important to note that the Distribution Link Representation Regularization is jointly executed with one of the self-supervised loss techniques.

Table D.1: Training details for the GloVE and BERT approaches employed in our study.

Methods	Batch	Optimizer	lr	Epochs
GloVE+Transf	64	Adam	10^{-4}	50
BERT	32	Adam Belief	10^{-4}	50

As mentioned earlier, in this case, except for the object detector within the module, we have unfrozen the scene graph generator g weights, allowing it to be trainable and to learn the representation and classification during the training process, merely according to the prediction answers. We have made deliberate experimental choices to mitigate potential biases and ensure an unbiased evaluation of our proposed framework.

For the un-normalized contrastive training step, we employ the MLP prediction head h. It comprises three fully connected layers, each followed by batch normalization and ReLU activation, except for the final layer. This setup ensures non-linearity and facilitates effective feature extraction. It is important to note that the MLP prediction head is exclusively utilized during the training phase and is subsequently discarded during inference, which aligns with prevailing practices in contemporary self-supervised training methods [7, 11].

The classification module f_c is another integral component of our model. It is designed as a two-layer MLP with a dropout rate of 0.2 and ELU activation.

D. Training Details

In this section, we provide further elaboration on our training approaches. Likewise, we have divided this section into two subsections: one with the utilization of GloVE word embedding along with a transformer-based model for the question encoder, and the other focusing on the application of BERT for word embedding and the question encoder.

D.1. GloVe Word Embedding and Transformer-based Question Encoder

We train the models using the Adam optimizer with a learning rate of 10^{-4} and weight decay 10^{-4}. We apply a batch size of 64, and a a linear learning rate schedule using a factor of 10^{-1} for every 20 epochs. All models are trained for 50 epochs. We emphasize that during training the weights of the scene graph generator g are frozen, and do not receive weight updates.

D.2. BERT for Word Embedding and Question Encoder

We train the models using the Belief Adam optimizer with a learning rate of 10^{-4} and weight decay 10^{-4}. We apply a batch size of 32, and a linear learning rate schedule using a factor of 10^{-1} for every 10 epochs. All models are

Table E.1: Detailed self-supervised implementation in our study by approaches.

	SGG Methods	Baseline	Local Sim	Global Sim.	Self Sim
GloVE+Transf	Frozen SGG	✓	✓	✓	✓
	Link Regularizer				
BERT	Frozen SGG	✓		✓	
	Link Regularizer			✓	✓

trained for 50 epochs. It is worth noting that in these cases, the weights of the scene graph generator g are not frozen during training. This deliberate choice allows for continual updates and improvements, particularly in the edge representation, through the utilization of the Distribution Link Representation Regularization strategy.

E. Self-Supervised implementation details

Table E.1 provides a comprehensive overview of the approach adopted in our study. It is worth noting that our training process was conducted sequentially and iteratively, allowing us to evaluate the performance of each approach before deciding on the subsequent implementation choice.

For instance, upon observing that the Local Similarity approach exhibited comparatively lower performance, albeit surpassing the baseline, we made the decision to discontinue its implementation on further research (i.e. with the BERT module and link distribution regularization approach). This strategy narrowed down the training possibilities, enabling us to focus solely on the most promising experiments. Another noteworthy example pertains to the utilization of BERT as our word embedding and query encoder module. Upon observing its positive impact on results, we exclusively applied the link distribution regularization technique with this architecture.

F. Further Ablations

F.1. Further Discussion on Language Bias

We elaborate on additional experiments aimed at evaluating the model's robustness when trained with the BERT module. In this case, the experiments investigate the impact of using a more expressive language model, such as BERT, on language biases in the VQA task and whether it harms the enhancement of visual information. We evaluate both how the biases convey not ideal information when using a more expressive language model such as BERT, and how the self-supervised approaches perform for robustness.

In this particular experiment, we augmented the images using various semantically-preserving techniques including Gaussian blur, Gaussian noise, color jitter with adjustments to brightness, contrast, and hue, as well as random rotation of up to 45 degrees. As for the questions, a similar approach was employed by randomly replacing up to 50% of the words with other words.ù

In this context, we emphasize that our approach maintains the semantic integrity of the image content. Consequently, the underlying model retains its fundamental objective of accurately predicting the correct response, despite the heightened complexity introduced.

Table F.1: Sensitivity of accuracy (%) for bias analyzes of BERT module.

Setup	**Methods**		
Question + Scene Graph	BERT Baseline	BERTGlobal+link	BERTSelfSim+link
Noise + SG	21.0	23.2	24.5
Question + Noise	42.4	41.8	42.8
Noise + Noise	19.8	21.7	21.3

Table F.1 demonstrated that even when employing a more expressive language model in the GQA dataset, the self-supervised learning still enhances the visual information for the predicted answer. Precisely, the results presented indicate that our approaches exhibit greater resilience to noise while maintaining the importance of visual information for the task.

We emphasize that the findings of this study demonstrate that despite the integration of a more expressive language model, such as BERT, the self-supervised learning method remains effective in leveraging visual data to classifier the predicted answers. Nevertheless, it is crucial to highlight that in this particular scenario, the results indicate a possible influence of language biases inherent in the dataset when utilizing a more advanced language model. See the results when the perturbation is employed solely on the scene graph compared to the non-perturbed one, in Table F.1. Additionally, when analyzing the results with full perturbation, the findings indicate an enhanced level of robustness when the self-supervision technique is combined with the model.

F.2. Does SelfGraphVQA have a few-shot learning capability?

We trained SelfGraphVQA with varying percentages of labeled data and found comparable performance to the GQA dataset, suggesting that adding self-supervised contrastive loss improves model generalization. We wanted to evaluate the different models on subsets of the full dataset. We tested reducing the ground truth labeling requirements and compared the performance when using SelfGraphVQA as opposed to directly training a fully supervised classification network.

In this case, we trained our SelfGraphVQA varying the percentage of labeled data, (i.e., 20%, 50%, and 100% of data) and evaluated it on the test dataset. As demonstrated in Fig. F.1, our proposal performs comparably with half of the GQA dataset evaluated on standard metrics. This insinuates that adding self-supervised un-normalized contrastive loss improves the generalization of the model.

Table F.2 shows how our proposal performs with the standard metrics when trained with 50% of training data, and we see that the three approaches perform on par with

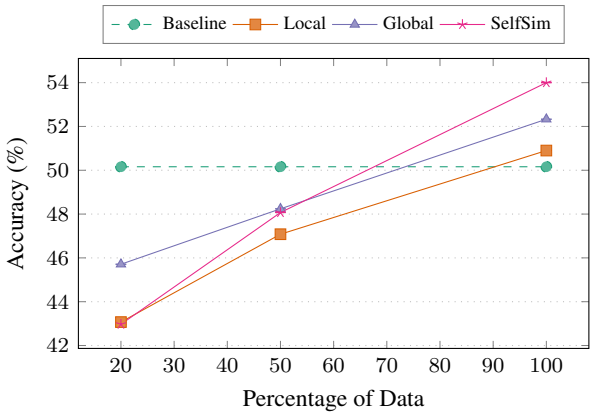


Figure F.1: Evaluation curve by percentage of data used in training on GQA dataset. The models obtain comparable results to baseline with 50% of the data. Note that we only illustrate the accuracy of the baseline trained on the full dataset for reference purposes.

Table F.2: Results (in %) evaluating by the standard metrics when training with 50% of GQA dataset.

Method	Binary	Open	Consistency	Validity	Plausibility	Accuracy
Global	63.5	27.6	54.1	94.8	90.1	48.2
Local	63.5	25.6	51.6	94.6	89.3	47.1
SelfSim	64.3	27.3	54.7	94.8	90.1	48.1

the baseline trained on the full dataset. In particular, the validity and plausibility metrics are consistent when compared to models trained on the full dataset.

Our intuition is that these metrics relate to linguistic bias and do not necessarily require large amounts of samples to converge, indicating that the model learns with little data what type of answer it should guess based on the type of question.

F.3. More Examples

We present additional examples to illustrate how scene graphs can contribute to the explainability of AI in the context of VQA, Figure F.2. These examples highlight that VQA remains an open area of research and that the performance of a model should be evaluated beyond standard metrics. These examples serve as a reminder that there is room for further exploration and improvement in the field of VQA, extending beyond conventional evaluation metrics.

All examples were predicted by the SelfSim framework. In the following discussion, the additional examples demonstrate both the problem of low agreement of VQA question answers due to ambiguity and the usefulness of scene graphs in providing more explainable AI for this task.

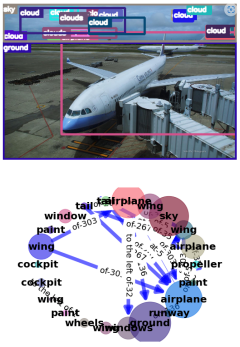
For instance in example 1, the model accurately predicts the answer, and the detection of the airplane in the scene graph is easily visualized. Conversely, in example 2, the model correctly do not detect the object mentioned in the question, leading to a correct answer of 'No'.

The benefits of using scene graphs for visual question answering become more evident in examples 3 and 4. In example 3, the model provides an objectively correct answer despite a different ground truth answer in the dataset. This discrepancy is explained by the scene graph, which highlights that the extracted object related to the question is 'flowers' rather than 'flowers'. In example 4, the model correctly classifies the link that relates the chair located to the right of the curtains in the scene graph, enabling the model to predict the correct answer.

In example 5, the acceptance of the model's answer 'liquid' as opposed to the ground truth 'beverage' is subjective and depends on the evaluator's opinion. This demonstrates that the model's response may fail to precisely evaluate the question, emphasizing the inherent challenges in VQA.

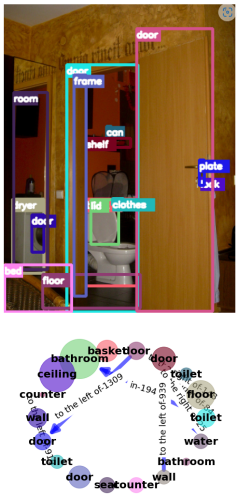
Overall, these examples highlight the potential benefits of incorporating scene graphs in visual question answering, offering insights into the model's reasoning and contributing to more interpretable AI systems.

(1) Correct



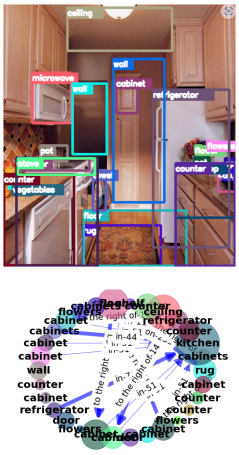
Q: What is the aircraft on the ground?
Answer: Airplane
Prediction: Airplane

(2) Correct



Q: Are there any parachutes or bags?
Answer: No
Prediction: No

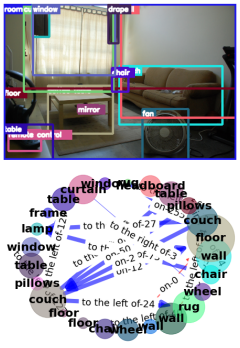
(3) SG explainable



Q: What is the white pot holding?

Answer: Flower
Prediction: Flowers

(4) SG explainable



Q: Which kind of furniture is right of the curtains?
Answer: Chair
Prediction: Chair

(5) Objectively Correct

Q: What is in the red glass??

Answer: Beverage
Prediction: Liquid

Figure F.2: Examples to demonstrate the complexity of VQA and the explainability of the scene graph. All example is predicted by the SelfSim framework.