SGA: Scene Graph Alignment for Evaluation of Text-to-Image Generation

Qiyu Wu[1], Zilong Wu[1] and Yoshimasa Tsuruoka[1]

[1]*The University of Tokyo, Tokyo, Japan*

Abstract

The evaluation of text-to-image generation still lags behind the rapid advancement of generative AI. Although recent evaluation methods using visual language models (VLMs) have achieved good agreement with human annotation by predicting a global alignment score, interpretable results are still lacking. On the other hand, the latest efforts attempt to generate visual language questions based on scene graphs (SGs) to provide interpretability. However, the number of inferences required by the algorithms is proportional to the number of nodes and edges in the graph. In this work, we investigate the utilization of SGs and VLMs to achieve interpretable evaluation results with a constant number of VLM inferences. The proposed scene graph alignment (SGA) evaluates text-to-image generation from three perspectives: (1) node alignment; (2) edge alignment; (3) global alignment. The alignment of the nodes and edges is calculated by embedding-based similarity, and the global alignment is holistically determined by the VLM. Experiments demonstrate that SGA has a better agreement with human annotation compared to existing interpretable baselines, while with fewer inferences. Moreover, we also found that assembling SGA scores and global scores from multiple models could achieve better performance compared to those strong baselines.

Keywords

Evaluation of text-to-image generation, Scene graph alignment,

1. Introduction

Text-to-image generation has advanced rapidly with impressive generation quality. Representative generative AI models such as DALLE [3] and Stable Diffusion [4], can generate realistic images following users' textual input. However, the evaluation of text-to-image generation is still behind. The widely used embedding-based evaluation [5, 6], lacking interpretability and more crucially, performs like bag-of-words and is therefore likely to be unreliable [7]. Many recent studies [8, 9, 10, 2, 1, 11, 12] attempt to improve the evaluation of text-to-image generation using visual-language models (VLM) [13, 14], due to the better performance of visual-language representation.

There are two main types: single-score methods and fine-grained methods. Single-score methods holistically evaluate text-to-image generation, predicting a single score that measures the alignment between the texts and the generated image. VQAScore [1], as a representative method of this type, outputs the probability of "Yes" asking if the image aligns with the texts. In contrast, fine-grained methods focus more on interpretable results. For example, DSG [2] asks

---

HI-AI@KDD, Human-Interpretable AI Workshop at the KDD 2024, 26th of August 2024, Barcelona, Spain

qiyuw@g.ecc.u-tokyo.ac.jp (Q. Wu); zw2599@g.ecc.u-tokyo.ac.jp (Z. Wu);
yoshimasa-tsuruoka@g.ecc.u-tokyo.ac.jp (Y. Tsuruoka)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

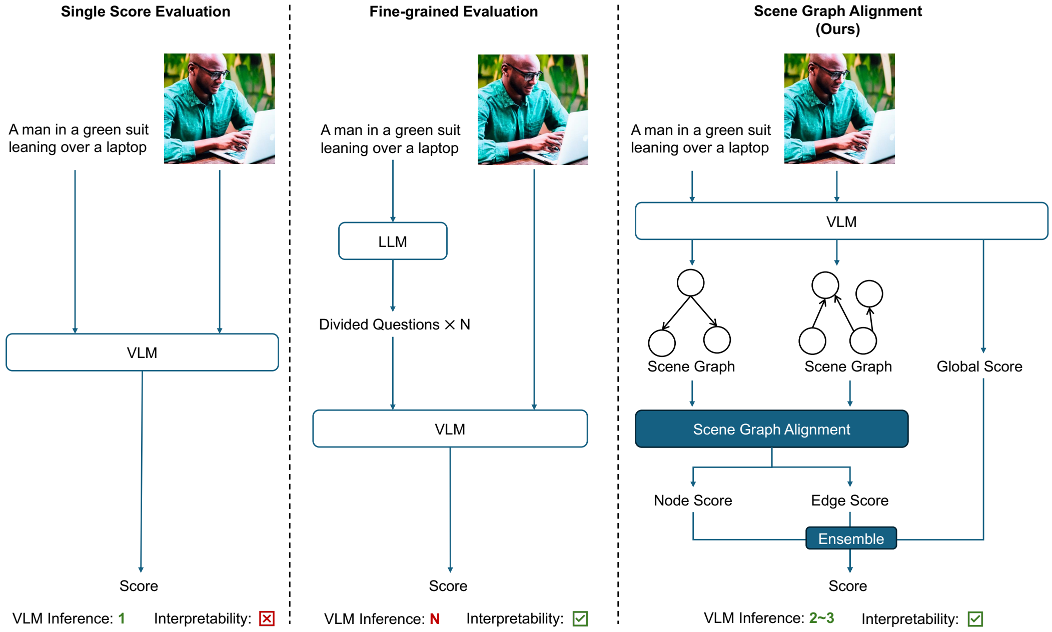


Figure 1: Evaluation of text-to-image generation utilizing VLMs. **Left:** direct using VLM to predict a single score but lack of interpretability, e.g., VQAScore [1]; **Middle:** using SG of the text to assist the VLM evaluation but it requires answering for each node and edge, e.g., DSG [2]; **Right (Ours):** generating SGs for the text and image separately and align the two SGs by embedding-based similarity.

multiple visual questions, which are generated based on the scene graph that represents the texts, to verify the existence of every node and edge. Figure 1 illustrates the difference between these two types of methods.

In general, single-score methods are simpler to use and perform better than fine-grained methods, i.e., have higher agreement with human annotation, referring to §4 for the detailed comparison. A fine-grained method usually sacrifices performance and efficiency as it includes multiple inference steps, which could induce more errors. In addition, it requires much more inferences that are proportional to the number of nodes and edges. Nevertheless, the interpretable evaluation results remain crucial for fine-grained evaluation needs.

To this end, we investigate a balance that preserves the advantages of the above-mentioned types of methods. Specifically, we propose a scene graph alignment (SGA) evaluation to provide both interpretability, and less inferences along with better performance. *Scene graph (SG)* is a graph that describes the scene in the image or text, with nodes representing the entities and edges representing the relation between two entities. By estimating the coverage of the text SG, the evaluation can be interpretable. For example, we can know whether a specific entity or relation in the text is correctly generated in the image. Unlike previous fine-grained methods that verify nodes and edges one by one, SGA generates SG with a single inference and then aligns the text SG and image SG by embedding-based similarity. By doing so, SGA only requires a constant number of inferences.

However, generating and aligning SGs for the evaluation of text-to-image generation is still non-trivial. Firstly, image usually contains more information than a textual description. This unbalance makes the generated SGs usually not comparable. In addition, the challenge also comes from the natural differences in the way information is conveyed between text and images, which result in discrepancies in the literal expression and graph structure of SGs. We propose language-conditioned SG generation through language generation by VLMs, and a similarity-based metric to assess the alignment between text and the generated image, by serializing nodes and edges into texts and then matching them by textual similarity. Finally, SGA evaluates text-to-image generation from three perspectives: (1) node alignment; (2) edge alignment; (3) global alignment. The global alignment is also predicted by the VLM, which can be achieved directly by editing the prompts as the previous single-score methods do [1].

Experiments show that the proposed SGA predicts alignment scores that correlate well with human annotation. Compared to the previous fine-grained method, SGA reduces the number of inferences to a constant, while achieves a better performance, which notably improves practicability of fine-grained text-to-image generation. Moreover, we also found that assembling SGA scores and global scores from multiple VLMs could achieve better performance compared to those strong baselines while maintaining the interpretability.

2. Scene Graph Generation with VLM

As dicsussed in §1, unlike conventional SG generators, we prompt VLMs to achieve generation of language-conditioned SGs to ensure accurate alignment afterward. This choice is inspired by the recent advance of large VLMs [13, 15, 16], which shows impressive generalized ability in most visual language understanding and generation. We generate two separate SGs for the text and the generated image, respectively. In particular, the language generation paradigm with VMLs is flexible enough to generate SGs along with other tasks in a single inference with a proper prompt. For example, we also ask the VLM to directly generate importance scores for aggregation [17], and a global alignment score [18] while generating the scene graph. We use `gpt-4o-2024-05-13`[1] as the VLM in this work. We manually design the instruction for the VLM in a zero-shot manner, which provides a template of output to make the results consistent for the following alignment. Refer to Figure 2 for the prompt of text SG generation and Figure 3 for the prompt of image SG generation.

Language-conditioned SG generation. To generate comparable SGs for text and image, especially when the image usually contains more information than the text, we propose language-conditioned SG generation, which is simple but effectively in our task. Specifically, we take text SG as input to the VLMs to make is understand the context and target when evaluating the generated image. Refer to the prompt in Figure 3 for details. The analysis with experiments will be presented in §E.

Importance scores. The VLM is prompted to assign an importance score to every node and edge in the SG for the text. This importance score is used §3.3 to aggregate all nodes and edges in an importance-weighed manner [17]. The importance scores are only assigned to the nodes and edges in the text SG, as the text-to-image evaluation is unidirectional and we only consider whether the information in the text is well aligned. All importance scores are normalized to

---

[1] https://platform.openai.com/docs/models/gpt-4o

sum up to 1 before being used for weighted aggregation. The effect of the use of importance scores will be presented in §E.1.

Global alignment score. We also prompt the VLM to predict a global alignment score between the given text and the image directly, as shown in Figure 3. This score is a direct evaluation by the VLM model, which is studied as an effective evaluation of text-to-image alignment [18, 1]. To achieve this, we input the generated SG for text into VLM, i.e., language-conditioned SG generation. This global score will be used together with the SG alignment scores to get a balanced overall score in §3.3.

Output example. Figure 4 shows an example of the output of the VLM. The VLM is prompted to include all the output in a JSON format for the following alignment. In actual implementation the *scene_graph_from_image* and *alignment_score* (global score) are generated together with one prompt in Figure 3, but *scene_graph_from_caption* is generated separately with prompt in Figure 2. In this example, we combine them together for better presentation.

3. Approximate Alignment of Scene Graphs

As demonstrated in Figure 4, existing VLMs are capable of generating SGs that represent texts or images well via our proposed pipeline. Especially when language-conditioned SG generation is applied, text SG and image SG are generally comparable. However, the alignment between two SGs can still be challenging, as we introduced in §1. We propose an approximate alignment approach to address the above issues, introduced in the following §B (preliminary), §3.1 (node score), §3.2 (edge score) and §3.3 (overall score).

3.1. Node Alignment Score

Each node in an SG consists of a *type* and possible multiple *attributes*. As mentioned above, directly aligning type names could be inaccurate due to the different literal expressions. Hence, we calculate the node similarity based two aspects: type name similarity and attribute similarity. Formally, given a node $n_i = (t_i, \mathcal{A}_i)$, the set of $\mathcal{A}_i$ consists of multiple pairs of $(key, value)$. We first serialize the attributes by concatenating the type name and all attributive values as inputs for calculating the attribute similarity:

$$SeN(n_i) = key_1, value_1, ..., key_k, value_k, ..., key_{|\mathcal{A}_i|}, value_{|\mathcal{A}_i|} \tag{1}$$

Next, the node alignment score between two nodes can be calculated by:

$$SGA^{node}(n_i^T, n_j^I) = \frac{Sim(t_i^T, t_j^I) + Sim(SeN(n_i^T), SeN(n_j^I))}{2} \tag{2}$$

where $n_i^T \in \mathcal{G}^T$ is the i_{th} node in the SG of the text and $n_j^I \in \mathcal{G}^I$ is the j_{th} node in the SG of the image. SeN is the function of serialization defined in Equation 1. Sim is the textual similarity function defined in Equation 8.

3.2. Edge Alignment Score

Similar to the alignment of the nodes, we also serialize the edge to calculate the similarity. Given a edge $r_i = (n_j, rl_i, n_k)$, it is serialized as follows:

$$SeR(r_i) = SeN(n_j), rl_i, SeN(n_k); \quad SeR_{rev}(r_i) = SeN(n_k), rl_i, SeN(n_j) \tag{3}$$

where n_j is the source node, n_k is the target node, and rl_i is the relationship. SeN is the serialization function for the nodes defined in Equation 1. Note that SeR and SeR^{rev} are distinguished because the edge is unidirectional, and hence the serialization is asymmetric. Next the alignment score between $r_i^T = (n_j^T, r_i^T, n_k^T)$ from the text and $r_i^I = (n_j^I, r_i^I, n_k^I)$ image can be calculated by:

$$SGA^{edge}(r_i^T, r_j^I) = \begin{cases} Sim(SeR(r_i^T), SeR_{rev}(r_j^I)) & \text{if } \textit{reverse} \\ Sim(SeR(r_i^T), SeR(r_j^I)) & \text{else} \end{cases} \tag{4}$$

reverse means the source and target nodes in the edge should be swapped, when the source node in the text is more likely to be matched with the target node in the image. Formally, *reverse* is True when:

$$SGA^{node}(n_j^T, n_j^T) + SGA^{node}(n_k^T, n_k^T) \leq SGA^{node}(n_j^T, n_k^T) + SGA^{node}(n_k^T, n_j^T)$$

where SGA^{node} is the node similarity function defined in Equation 2. This condition determines whether to swap the source and target nodes in the image graph by comparing the pairwise similarity between the four nodes. The source and target nodes should be swapped if the total node similarities are higher after swapping.

3.3. Overall Alignment Score

Nodes and edges matching. The overall alignment score between $\mathcal{G}^{\mathcal{T}}$ and G^I is a combination of the alignment scores of the nodes and the edges. However, since $\mathcal{G}^{\mathcal{T}}$ and G^I are generated separately, it is still difficult to find comparable counterparts to calculate the similarity. Matching corresponding nodes and edges in the other SG is a prerequisite to aggregate the scores. Besides, our target is matching nodes and edges with maximum similarity, while the total number of them could be different in the SGs. Specifically, each node (or edge) in one SG is matched with a counterpart in another SG, meanwhile the sum of the similarities of these matched nodes (or edges) is the maximum. This is a standard *weighted bipartite graph matching problem* [19], hence we directly use `scipy.optimize.linear_sum_assignment` [2] to obtain the matching results:

$$\mathcal{M}^{node}(n_i^T) = n_j^I \in \mathcal{N}^I; \quad \mathcal{M}^{edge}(r_i^T) = r_j^I \in \mathcal{R}^I \tag{5}$$

The cost matrix required, i.e., pair-wise similarities, for this problem can be calculated by our proposed alignment scoring function in Equation 1 and Equation 3.

---

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html

Aggregation and the overall SGAScore. After obtaining the matching results, for each node $n_i^T \in \mathcal{N}^T$ in the text SG, we sum up the alignment score of the node with its matching node in the image to calculate the aggregated scores. The aggregation for the edge alignment scores is the same. Combined with the global alignment score, the final overall alignment score can be calculated as:

$$\begin{aligned} SGA^{fg} = & \sum_{n_i^T \in \mathcal{N}^I} SGA^{node}(n_i^T, \mathcal{M}^{node}(n_i^T)) * w_i^{node} \\ & + \sum_{r_i^T \in \mathcal{R}^I} SGA^{edge}(r_i^T, \mathcal{M}^{edge}(r_i^T)) * w_i^{edge} \end{aligned} \tag{6}$$

$$SGAScore = \frac{2 \times SGA^{fg} \times SGA^{global}}{SGA^{fg} + SGA^{global}} \tag{7}$$

where SGA^{fg} indicate the *fine-grained* score that is the average of weighted node similarities and edge similarities. w_i^{node} and w_i^{edge} are the importance scores predicted by the VLM, as introduced in §2, which are assigned to each node and edge in G^T for weighted aggregation. SGA^{global} is the VLM-based score that directly assesses the alignment between text and image, introduced in §2. Finally as defined in Equation 7, the $SGAScore$ is the harmonic mean of SGA^{fg} and SGA^{global} to achieve a balance for the evaluation of text-to-image generation.

4. Evaluation and Discussion

4.1. SGA Scores Correlates Well to Human Annotation

Table 1 demonstrates the experimental results on TIFA160 and Pick-a-Pic. SGA is compared with baselines with three different levels of interpretability: fine-grained, single-score with or without textual reasoning process. Refer to §D for the details of baselines.

SGA performs competitively with fine-grained baselines and single-score baseline with textual interpretability. According to the results, SGA achieves a competitive level of agreement with human, compared to fine-grained baselines: TIFA and DSG. SGA outperforms the baselines in pairwise accuracy, Pearson coefficient on TIFA160, and accuracy on Pick-a-Pic. As for interpretability, these fine-grained baselines provide divided questions that verify specific entities and relationships, which can also be measured by the node and edge scores from our SGA. On the other hand, for single-score baselines with reasoning process by VLMs, such as VIEScore and GPT4V-Eval, the performance are generally competitive. SGA performs the best in pairwise accuracy on TIFA160, GPT4V-Eval performs the best in Pearson and Kendall coefficients, and VIEScore outperforms the others on Pick-a-Pic, while differences are not significant. GPT4V-Eval and VIEScore share a very similar idea of asking the VLM, such as GPT4V, to evaluate the alignment between the text and the generated image by outputting a single score. Along with the score, the reasoning process of the VLM is also available. The interpretability is still limited as there is no structure in the textural reasoning process, which could be harder to interpret than divided questions or graph-based similarities.

Method	Interpretability	# of VLM inferences	TIFA 160			Pick-a-Pic
			Acc.	ρ	τ	Acc.
Fine-grained baselines						
TIFA [20]	Divided Questions	N*	60.4	49.3	38.1	-
DSG [2]	Divided Questions	N*	54.3	55.6	**45.4**	70.0
SGAScore (Ours)	Graph-based	2	**64.5**	**58.2**	40.5	**76.0**
Single-score baselines, w/ reasoning process by VLMs						
VIEScore [18]	Textual	1	63.9	61.2	47.4	**78.0**
GPT4V-Eval [21]	Textual	1	64.0	**58.9**	**46.8**	74.0
SGAScore (Ours)	Graph-based	2	**64.5**	58.2	40.5	76.0
Single-score baselines						
CLIPScore [5]	None	1	55.8	29.6	19.9	76.0
BLIPv2Score [22]	None	1	57.5	35.6	23.3	80.0
VQAScore [1]	None	1	71.2	66.2	51.9	**84.0**
VQAScore (Re-run)	None	1	**71.5**	**66.4**	**52.6**	83.0
SGAScore (Ours)	Graph-based	2	64.5	58.2	40.5	76.0

Table 1

The comparison of evaluation performance. The Pairwise Acc. / Pearson ρ / Kendall τ are reported. Higher scores are better. Part of the results except ours are from [1], which is also the basic of our evaluation programs. *: N is the number of questions that verifies every entites and relationships mentioned in the text, refer to §D for details.

SGA underperforms VQAScore. SGA outperforms embedding-based metrics CLIPScore and BLIPScore. However, the strongest baseline VQAScore [1] outperforms SGAScore by a clear margin. VQAScore is achieved by training a bidirectional VQA model and take the likelihood as the score, which is simple and well correlated to human judgement, but without interpretability. It should be harder to improve fine-grained methods than single-score ones, as the methods are usually more sophisticated. Hence it is difficult to control the accuracy of each step, e.g., question generation, scene graph generation, or question answering.

5. Conclusion

In this work, we investigate the utilization of VLM along with SGs, to propose a fine-grained evaluation method for text-to-image generation. Our proposed scene graph alignment score (SGAScore) generates SGs for the text and image separately and aligns them using embedding-based similarity in an approximate manner. By doing so, SGAScore can provide interpretable evaluation results with only two inferences by VLM, making it an alternative to existing fine-grained baselines that usually generate and answer multiple divided visual questions. Experiments and analysis demonstrate that SGAScore achieves a competitive level of agreement with human annotation. However, although fine-grained methods have an advantage due to interpretable evaluation, single-score baselines still outperform fine-grained methods, including the proposed SGAScore.

References

[1] Z. Lin, D. Pathak, B. Li, J. Li, X. Xia, G. Neubig, P. Zhang, D. Ramanan, Evaluating text-to-visual generation with image-to-text generation, arXiv preprint arXiv:2404.01291v2 (2024).

[2] J. Cho, Y. Hu, J. M. Baldridge, R. Garg, P. Anderson, R. Krishna, M. Bansal, J. Pont-Tuset, S. Wang, Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation, in: The Twelfth International Conference on Learning Representations, 2024.

[3] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever, Zero-shot text-to-image generation, in: International conference on machine learning, Pmlr, 2021, pp. 8821–8831.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 10684–10695.

[5] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, Y. Choi, Clipscore: A reference-free evaluation metric for image captioning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 7514–7528.

[6] J. Li, D. Li, C. Xiong, S. Hoi, Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, in: International conference on machine learning, PMLR, 2022, pp. 12888–12900.

[7] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, J. Zou, When and why vision-language models behave like bags-of-words, and what to do about it?, in: The Eleventh International Conference on Learning Representations, 2022.

[8] Z. Zhou, M. Shi, H. Caesar, Vlprompt: Vision-language prompting for panoptic scene graph generation, arXiv preprint arXiv:2311.16492v2 (2023).

[9] X. Wu, Y. Hao, K. Sun, Y. Chen, F. Zhu, R. Zhao, H. Li, Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis, arXiv preprint arXiv:2306.09341v2 (2023).

[10] T. Gupta, A. Kembhavi, Visual programming: Compositional visual reasoning without training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14953–14962.

[11] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, Y. Dong, Imagereward: Learning and evaluating human preferences for text-to-image generation, Advances in Neural Information Processing Systems 36 (2024).

[12] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, O. Levy, Pick-a-pic: An open dataset of user preferences for text-to-image generation, Advances in Neural Information Processing Systems 36 (2023) 36652–36663.

[13] H. Liu, C. Li, Q. Wu, Y. J. Lee, Visual instruction tuning, Advances in neural information processing systems 36 (2024).

[14] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, et al., Pali: A jointly-scaled multilingual language-image model, in: The Eleventh International Conference on Learning Representations, 2022.

[15] H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, Y. Sun,

et al., Deepseek-vl: towards real-world vision-language understanding, arXiv preprint arXiv:2403.05525v2 (2024).

[16] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, Q. Ye, F. Wei, Grounding multimodal large language models to the world, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=lLmqxkfSIw.

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with BERT, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[18] M. Ku, D. Jiang, C. Wei, X. Yue, W. Chen, Viescore: Towards explainable metrics for conditional image synthesis evaluation, arXiv preprint arXiv:2312.14867v2 (2023).

[19] Wikipedia contributors, Assignment problem — Wikipedia, the free encyclopedia, 2024. URL: https://en.wikipedia.org/w/index.php?title=Assignment_problem&oldid=1199647159, [Online; accessed 30-May-2024].

[20] Y. Hu, B. Liu, J. Kasai, Y. Wang, M. Ostendorf, R. Krishna, N. A. Smith, Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 20406–20417.

[21] X. Zhang, Y. Lu, W. Wang, A. Yan, J. Yan, L. Qin, H. Wang, X. Yan, W. Y. Wang, L. R. Petzold, Gpt-4v (ision) as a generalist evaluator for vision-language tasks, arXiv preprint arXiv:2311.01361v1 (2023).

[22] J. Li, D. Li, S. Savarese, S. Hoi, Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, in: International conference on machine learning, PMLR, 2023, pp. 19730–19742.

[23] C. Lu, R. Krishna, M. Bernstein, L. Fei-Fei, Visual relationship detection with language priors, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, 2016, pp. 852–869.

[24] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al., Visual genome: Connecting language and vision using crowdsourced dense image annotations, International journal of computer vision 123 (2017) 32–73.

[25] R. Zellers, M. Yatskar, S. Thomson, Y. Choi, Neural motifs: Scene graph parsing with global context, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5831–5840.

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

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

[28] Q. Li, C. Ji, S. Guo, Z. Liang, L. Wang, J. Li, Multi-modal knowledge graph transformer framework for multi-modal entity alignment, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 987–999. doi:`10.18653/v1/2023.findings-emnlp.70`.

[29] Z. Chen, J. Chen, W. Zhang, L. Guo, Y. Fang, Y. Huang, Y. Zhang, Y. Geng, J. Z. Pan, W. Song, et al., Meaformer: Multi-modal entity alignment transformer for meta modality

hybrid, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 3317–3327.

[30] Z. Lin, Z. Zhang, M. Wang, Y. Shi, X. Wu, Y. Zheng, Multi-modal contrastive representation learning for entity alignment, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 2572–2584. URL: https://aclanthology.org/2022.coling-1.227.

[31] T. Gao, X. Yao, D. Chen, SimCSE: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 6894–6910. URL: https://aclanthology.org/2021.emnlp-main.552. doi:`10.18653/v1/2021.emnlp-main.552`.

[32] Q. Wu, C. Tao, T. Shen, C. Xu, X. Geng, D. Jiang, PCL: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 12052–12066. URL: https://aclanthology.org/2022.emnlp-main.826. doi:`10.18653/v1/2022.emnlp-main.826`.

[33] Y. Xie, Q. Wu, W. Chen, T. Wang, Stable contrastive learning for self-supervised sentence embeddings with pseudo-siamese mutual learning, IEEE/ACM Transactions on Audio, Speech, and Language Processing 30 (2022) 3046–3059.

[34] K. Zhao, Q. Wu, X.-Q. Cai, Y. Tsuruoka, Leveraging multi-lingual positive instances in contrastive learning to improve sentence embedding, in: Y. Graham, M. Purver (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 976–991.

[35] Z. Miao, Q. Wu, K. Zhao, Z. Wu, Y. Tsuruoka, Enhancing cross-lingual sentence embedding for low-resource languages with word alignment, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 3225–3236. URL: https://aclanthology.org/2024.findings-naacl.204.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[37] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al., Photorealistic text-to-image diffusion models with deep language understanding, Advances in neural information processing systems 35 (2022) 36479–36494.

[38] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, B. Hutchinson, W. Han, Z. Parekh, X. Li, H. Zhang, J. Baldridge, Y. Wu, Scaling autoregressive models for content-rich text-to-image generation, Trans. Mach. Learn. Res. (2022).

[39] J. Cho, A. Zala, M. Bansal, Dall-eval: Probing the reasoning skills and social biases of text-

to-image generation models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3043–3054.

[40] D. Deutsch, G. Foster, M. Freitag, Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.

[41] H. Wu, J. Mao, Y. Zhang, Y. Jiang, L. Li, W. Sun, W.-Y. Ma, Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6609–6618.

A. Related Work

A.1. Evaluation of Text-to-image Generation

Current automatic evaluation for text-to-image generation can be approximately classified into two types: single-score methods and fine-grained methods. Single-score methods such as CLIPScore [5] and BLIPScore [6] calculate embedding-based similarity between text and generated image. Those metrics behave like bag-of-words [7] and is therefore likely to be unreliable in complex scenarios. More recent studies utilize VLMs to further improve the evaluation, such as VQAScore [1] and VIEScore [18], which input text and image into the VLM and obtain a overall score. Fine-grained methods focus more on interpretable results. DSG [2], as a representative work of this type, generates visual language questions for every node and edge in the SG generated by the text. DSG verifies fine-grained elements but requires inferences for every question, and performance is worse than most single-score metrics. In addition, there are also metrics such as ImageReward [11], PickScore [12], and HPSv2 [9] that rely on human evaluators. Although these methods provide valuable insights, they are often expensive and difficult to reproduce. Hence, this line of work is out of the scope in this work.

A.2. SG Generation and Alignment

SG generation is an important area for learning visual-language knowledge. It generates graphs where each object is a node, and the relationships between objects are the edges, thus generating SGs can represent images at a detailed level and bridge the gap between images and semantics. Starting from the work by Lu et al. [23] and Visual Genome [24], most SG generation methods follow the pipeline of object, relationship, and attribute detection. For example, Neural Motifs [25] leverages global context and motifs to improve SG generation by integrating vision and language effectively. Graph R-CNN [26] employs region proposal network to accurately identify relationships between objects. Tang et al. [27] propose SGDet and unifies the evaluation of SG generation. In our work, we use SG as a structural representation of the text and image, to achieve interpretable evaluation of text-to-image generation.

SG alignment, as a sub-goal in this work, measures whether the entities and relationships in the text are well generated in the image. This could be challenging, as the structure of the SG and the literal expression of the nodes and edges varies, which means that there could be multiple correct SGs for an identical image. As a result, conventional SG evaluation generally uses the top-K recall [27] as the metric so that the most correct predictions can be counted. A related task is multimodal knowledge graph alignment; recent methods [28, 29, 30] typically require to learn a common embedding space for all modalities, where similar entities in the knowledge graph have similar embeddings. In this work, we propose to serialize nodes and edges and calculate embedding-based similarity, as above-mentioned solutions are not practical due to the requirement of efficiency and open-domain semantics.

Text Scene Graph Generation

```
Given a caption {TEXT} that describes an image, generate a scene graph for this caption. Follow these guidelines:

1.Text-Based Scene Graph:
Identify key entities (objects) and their attributes (properties) mentioned in the text.
Structure nodes (entities) and edges (relationships) based on described interactions and spatial relationships.
Assign an importance score to each node and edge based on their relevance in the text.

2.Output Format:
Output the scene graph in JSON format with nodes and edges clearly defined.
Output the thoughts and reasoning behind the scene graph generation process. Note that the scene graph will be used for later
alignment analysis with an image-based scene graph.

3.Example Output Structure:
{
  "scene_graph_from_caption": {
   "thoughts": "entity_type_1 was detected as it was mentioned in the text. attribute_1 was excluded as it was too abstract
      to be used for later alignment analysis.",
    "nodes": [
      {“Id”: 1, “type”: “entity_type_1”, “attributes”, {“attribute_1”: “value_1”}, "importance_score": 0.9},
      {“id”: 2, “type”: “entity_type_2”, “attributes”: {“attribute_2”: “value_2”}, “importance_score”: 0.8},
      {"id": 3, "type": "entity_type_3", "attributes": {}, "importance_score": 0.6}
    ],
    "edges": [
      {"source": 1, "target": 2, "relationship": "relationship_type_1", "importance_score": 0.7},
      {"source": 1, "target": 3, "relationship": "relationship_type_2", "importance_score": 0.5}
    ]
  }
}

Important Notes:
Ensure all mentioned entities and their relationships are accurately identified and described.
Assign appropriate importance scores based on relevance in the text.
Do not include additional text or explanations in the output.
```

Figure 2: Text Scene Graph Generation

B. Preliminary

Task definition. The text SG and image SG are defined as $\mathcal{G}^T$ and $\mathcal{G}^I$, respectively. A graph $\mathcal{G}$ consists of a pair set of $(\mathcal{N}, \mathcal{R})$ with nodes $\mathcal{N}$ and edges $\mathcal{R}$. Each node $n_i \in \mathcal{N}$ contains a type name t_i and a set of attributes $\mathcal{A}_i$, i.e., $n_i = (t_i, \mathcal{A}_i)$. Each edge $r_i \in \mathcal{R}$ contains two nodes and a relation, i.e., $r_i = (n_j, rl_i, n_k)$. Our goal is to calculate the alignment score $SGAScore(\mathcal{G}^T, \mathcal{G}^I)$, which is defined as *a unidirectional score that measures how much of the information in the text is covered in the image*. Note that we should only consider the unidirectional completeness of the text because the information contained in the image is usually more than that in the text.

Textual similarity on serialized nodes and edges. As the SGs of text and image are generated separately, leading to potential differences in structures and literal expression, we serialize the edges and nodes to calculate the textual similarity for measuring the semantic alignment. Serialization of nodes and edges will be introduced in §3.1 and §3.2, respectively.

Then we calculate the similarity using a textual embedding model. Text embedding models[31, 32, 33, 34, 35] encode texts into dense vectors. Semantic similarity can be easily obtained by calculating the cosine distance between the vectors. We use `qiyuw/pcl-roberta-base`[3] as the embedding model in this work. All settings are kept as the default introduced in the original PCL paper [32]. Formally, the similarity between serialized nodes or between serialized edges can be calculated as:

$$Sim(\cdot, \cdot) \in [0, 1] \tag{8}$$

C. Benchmarks and Metrics

Benchmarks. Following previous studies [2, 1], we evaluate the performance of text-to-image evaluator by assessing its agreement with human annotation. The following benchmarks are chosen:

- **TIFA160** [20] contains 160 text prompts from four sources: MSCOCO captions [36], DrawBench [37], PartiPrompts [38], and PaintSkill [39]. Each prompt is input into five text-to-image models, generating 800 image-text pairs. Additionally, these pairs are labeled using a 1-5 Likert scale for human evaluation in the DSG paper [2], which is used in this work for assessing the agreement with human.
- **Pick-a-Pic** [12] is a dataset including labels of human preference for text-to-image generative models. We use a clean subset selected by Lin et al. [1], as the original dataset contains many improper contents and incorrect labels. The selected dataset contains 100 prompts for 200 generated images for binary accuracy.

Metrics. We use classical **Pearson** and **Kendall** coefficients to assess the agreement with the human annotation. The benchmarks provide human annotation such as 1-5 Likert scale for each text-image pairs, these coefficients measure the correlation between the human and model judgements. In particular, as advocated by Deutsch et al. [40] and Lin et al. [1], pairwise accuracy is a more reliable metric to assess the model's agreement with human. Specifically, the pairwise accuracy determines whether the model and human agree with pairwise ranking for each text-image pair scores, higher, lower, or ties to another text-image pair. Refer to the original papers [40, 1] for a detailed analysis. In this work, we use **pairwise accuracy** as the main metric for assessing the evaluator.

D. Baselines

We category baselines into single-score methods and fine-grained methods for fair comparison in the following sections. For baselines, we follow the implementation in the original papers, e.g., the prompts for the VLMs. We fix the random seed as 42. The evaluation program is implemented based on the codebase[4] of VQAScore [1].

---

[3] https://huggingface.co/qiyuw/pcl-roberta-base

[4] https://github.com/linzhiqiu/t2v_metrics

Single-score baselines. Single-score methods holistically evaluate the generated image, i.e., giving an overall score directly. We choose two widely used baselines of **CLIPScore** [5], **BLIPScore** [22] in this category. These embedding-based methods encode text and image into a shared space and calculate the distance as the score. Another strong baseline is **VQAScore** [1], instead of encoding text and image into embeddings, it achieves this in a visual question-answer manner, i.e. asking a VLM to answer whether the generated image represents the information in the text. The probability of generating "Yes" is taken as the score. Most single-score methods are not interpretable in nature, except **VIEScore** [18] and **GPT4V-Eval** [21], which generate an explanation along with a textual score by VLMs such as GPT-4V. Single-score baselines are easy to use and generally perform well in terms of human agreement, however, the interpretability is lacked or limited.

Fine-grained baselines. Fine-grained methods divide the evaluation into multiple steps, e.g. visual questions. By answering those fine-grained questions, the method can provide interpretable results, including the specific existence of entities or relationships. For example, **TIFA** [20] generates multiple questions for the text by an LLM, e.g., "Is there a moon?", and returns the accuracy of answering all the questions by VLMs as the score. More relevant to our work, **DSG** [2] has a similar pipeline but constrains the generation of questions with SG, to obtain more reasonable questions and question-answer logic. As the pipeline includes multiple questions generation and answering, these fine-grained methods usually scarify the accuracy of agreement with human annotation. Nevertheless, the interpretable results could provide insight into the evaluation, making these baselines another important line of work.

E. Implementation Analysis

Before comparing it with previous works, we investigate the best practice of SGA in this section. There are two implementation decisions to make: the SG generation model and the use of language-conditioned SG generation. The decision will be discussed in the following paragraphs. As conventional SG generators are not capable of generating a global score and importance scores as VLMs, we calculate the fine-grained score SGA^{fg} defined in Equation 6 and do not use importance scores, i.e., averaging the scores of nodes and edges, for a fair comparison in this section.

Choice of SG generation model. VLMs are flexible in generating SGs along with other tasks in a single inference, such as predicting a global alignment score or giving a reasoning process, as we argued in §2. Even so, it is still necessary to study the use of conventional SG generators for both text and image, because our proposed approximate alignment is applicable to any other SG generator. Hence, we conduct an evaluation of different combinations of SG generators. For text SG, we use SGParser[5] provided by Wu et al. [41]. SGParser takes a single sentence and parses it into a SG. For image SG, we choose the widely used SGDet [27][6], which takes a single image and outputs an SG. For SGDet and SGParser, we use the default settings.

---

[5] https://github.com/vacancy/SceneGraphParser

[6] https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch

The performance of different SG generators on TIFA160 is shown in Table 2. It can be observed that language-conditioned GPT-4o generally performs better than SGDet as the image SG generator. The possible reason is the capability of GPT-4o to conduct the language-conditioned SG generator, which can mitigate the imbalanced amount of information for text and image. In contrast, the performance of SGParser and GPT-4o is comparable to that of the text SG generator. It can be expected that the SG generation for a relatively short sentence is easier than that for image, and traditional parser should perform well on this task. As a result, we choose **GPT-4o** as image SG generator and **SGParser** as text SG generator for SGA. The reason we choose SGParser as it is much easier and faster than GPT-4o, given the comparable performance.

Effect of language-conditioned SG generation. To further verify the necessity of language-conditioned generation, we evaluate the performance of GPT-4o without text SG as input, i.e., only input the image into GPT-4o and prompt it to generate SG. Refer to §2 for detailed implementation of this. The results are in Table2. It shows that no matter whether using SGParser or GPT-4o as text SG generator, the performance will drop largely. This indicates that language-conditioned SG generation is essential when applying SGA for the evaluation of text-to-image generation.

Image SG Generator	Text SG Generator	
	SGParser [41]	GPT-4o
SGDet [27]	53.4/12.5/14.9	47.6/3.0/2.9
GPT-4o w/o language-conditioned	53.3/21.0/14.8	53.6/20.8/15.3
GPT-4o w/ language conditioned	60.5/42.1/30.1	60.3/43.6/32.1

Table 2

Comparing different combinations of SG generators on TIFA160. The Pairwise Acc. / Pearson ρ / Kendall τ are reported, in which Pairwise Acc. is the main metric as we discussed in §C. Higher scores are better.

E.1. Detailed Analysis

The balance between fine-grained and global scores. In Equation 7 we combine the fine-grained score SGA^{fg} and global score SGA^{global} by harmonic mean. To further investigate the balance between the fine-grained and global scores, we set a hyperparameter β, to assess the effects of different values of β on the combined performance of SGA^{fg} and SGA^{global} by adjusting their ratios. Formally, we calculate SGA^{β} as follows,

$$SGA^{\beta} = (1 - \beta) \times SGA^{fg} + \beta \times SGA^{global} \tag{9}$$

Figure 5 shows the performance of SGA^{β} when varying the value of β. The best β is between 0.5 and 0.9, and neither removal of SGA^{fg} nor SGA^{global}, that is, when β is 1.0 or 0.0, can lead to a significant drop in all metrics. The phenomenon suggests that fine-grained and global score evaluate text-to-image from different perspectives, and the combination of them is overall helpful. Moreover, $SGAScore$ defined in Equation 7, which is the harmonic mean of SGA^{fg} and SGA^{global}, still performs the best no matter the value of $beta$. Consequently, the simple but effective harmonic mean is recommended to achieve the balance between fine-grained and global evaluation in practice.

Ablation study Table 3 demonstrates the ablation study of different components of SGAScore. We first evaluate the performance when removing the use of importance scores defined in §2. Surprisingly, the performance does not drop significantly after removing importance scores, which indicates that simply averaging all entities and relationships mentioned in the text is how humans make judgements. In addition, we evaluate the performance of single scores in SGAScore. The results show that the global score performs relatively well, but the performance still drops by 9.9 in pairwise accuracy. In contrast, if we only use node or edge similarity, the performance drops largely. On the other hand, we also evaluate the performance when removing the each score, and observe that removing global score will greatly degrade the performance, and the degradation is narrower when removing node or edge scores. Overall, the results indicate that all three types of scores are important for SGAScore. Edge and node scores are useful not only for interpretability but also for the agreement with human annotation. In particular, the global score seems to be more essential as it can solely perform well and the removal of it leads to larger degradation. The possible reason is that the global score evaluates the generated image from a global perspective, which is a strong complement to the fine-grained scores.

Method	Pearson	Kendall	Acc.
SGAScore	58.2	40.5	64.5
-w/o Importance Score	58.1 ↓ 0.1	40.5 ↓ 0	64.5 ↓ 0
-Only Node Score	44.5 ↓ 13.7	30.7 ↓ 9.8	58.4 ↓ 6.1
-Only Edges Score	38.4 ↓ 19.8	30.3 ↓ 10.2	59.9 ↓ 4.6
-Only Global Score	57.5 ↓ 0.7	44.9 ↑ 4.4	54.6 ↓ 9.9
-w/o Nodes	56.1 ↓ 2.1	39.5 ↓ 1.0	64.1 ↓ 0.4
-w/o Edges	57.0 ↓ 1.2	39.5 ↓ 1.0	63.1 ↓ 1.4
-w/o Global Score	44.4 ↓ 13.8	32.5 ↓ 8.0	60.6 ↓ 3.9

Table 3
Ablation study of SGA. The Pairwise Acc. / Pearson ρ / Kendall τ on TIFA160 are reported. Higher scores are better.

F. Composite score of SGAScore and VQAScore.

Although fine-grained methods performs worse than VQAScore, we hypothesize that fine-grained methods and single-score methods evaluate text-to-image generation from different perspectives implicitly. Accordingly, we attempt to combine SGAScore and VQAScore and achieve a better performance than VQAScore. Figure 6 shows the performance of composite score with varying ratio of SGAScore and VQAScore. The optimal performance occurs when SGAScore and VQAScore are combined in a 3:7 ratio, which is reported in Table 1. Besides, when the ration is between 5:5 and 9:1, the composite score can perform competitively with VQAScore. This is a minor but interesting finding, but we want to note that it should be used with caution in practice, as it makes the composition of the score more complex and possibly weakens the interpretability.

Image Scene Graph Generation

Given an image, generate a scene graph from the image. Your task is to evaluate the quality of the image generation by creating a scene graph based on the image content. Do not intentionally generate results that match the text. Instead, focus on accurately detecting and describing all objects, their attributes, and relationships in the image.

1.Step-by-Step Process:
Identify each entity or relationship in the image. For every entity or relationship in the following caption-based scene graph, evaluate its visibility and clarity in the image, and decide whether to include it in the output scene graph. If an entity or relationship is not perfectly visible in the image, exclude it from the output scene graph.

Caption-based scene graph:
{TEXT_SCENE_GRAPH}

In addition, evaluate the overall alignment degree between the image and the caption, and output an alignment score.

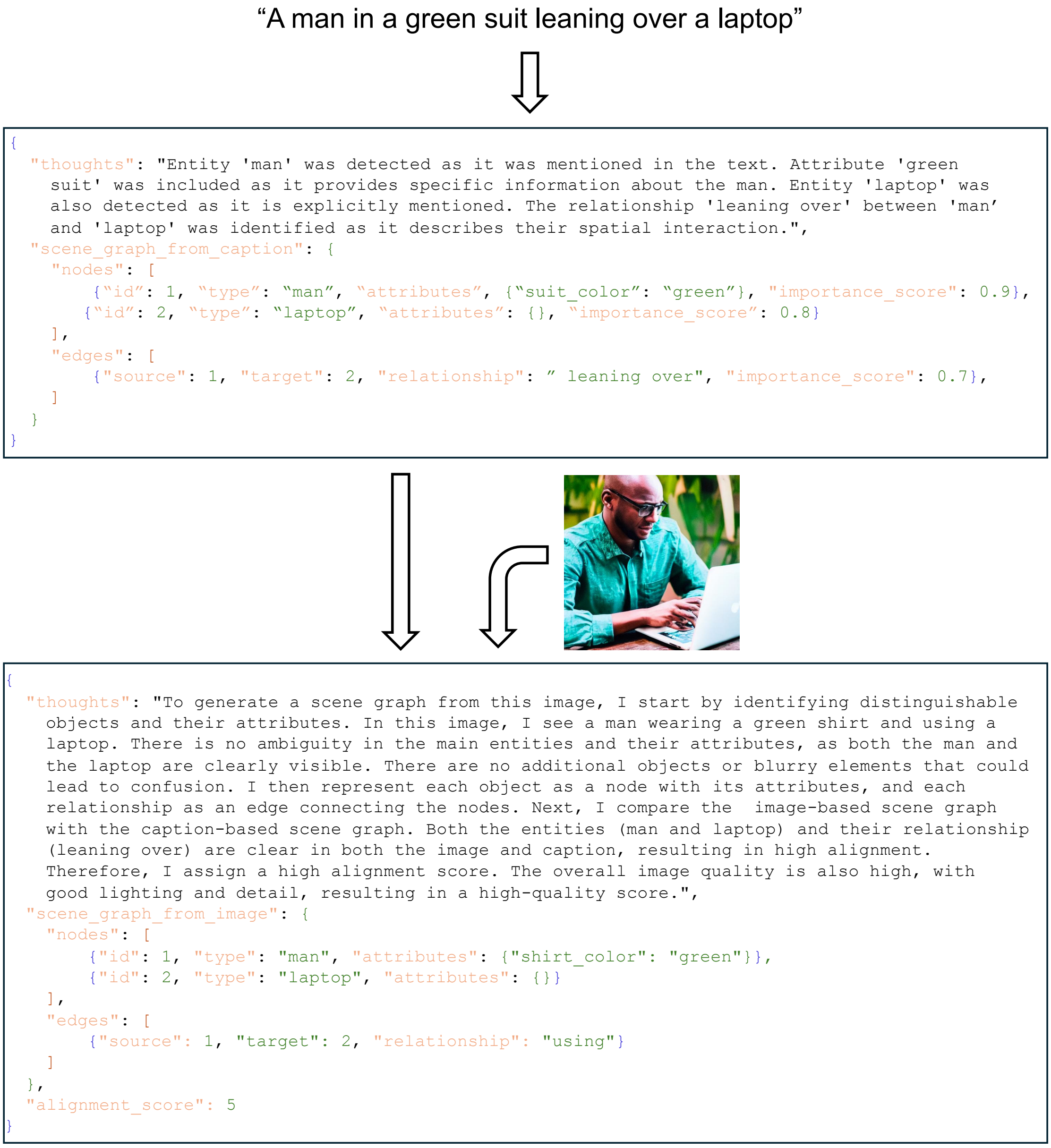
2.Output Format:
Output the scene graph in JSON format with nodes and edges clearly defined.
Output the thoughts and reasoning behind the scene graph generation process. Note that your task is to evaluate the quality of image by generating a scene graph from it. Do not intentionally generate results that match the text. Instead, focus on accurately detecting and describing all objects, their attributes, and relationships in the image.
Each node and edge should include a source indicator (image) and have corresponding identifiers where matches are suspected.
Output a score between 1-5 to determine the degree of alignment between the image-based and caption-based scene graphs.

Example Output Structure:

```
{
  "thoughts": "To generate a scene graph from an image, I start by identifying all distinguishable
    objects and their attributes. In this image, I see [Entity1] with [Attribute1] and [Entity2]
    with [Attribute2]. There is also a blurry shape that might be [Entity3], but its features are
    not clear enough to be included. I then evaluate the visibility and clarity of these elements,
    ensuring that only perfectly visible objects and relationships are included. Since the blurry
    shape isn't distinct, it is excluded from the scene graph. I construct the scene graph by
    representing each object as a node with its attributes and each relationship as an edge
    connecting the nodes. Next, I compare the image-based scene graph with a caption-based scene
    graph, looking for matches and discrepancies. For instance, if the caption describes [Entity1]
    with [Attribute1] next to [Entity2] with [Attribute2], but also mentions [Entity3], which
    isn’t clearly visible in the image, the alignment would be lower. I assign an alignment
    score based on how well the two graphs match, with a score of 2 indicating fair alignment due
    to some discrepancies.",
  "scene_graph_from_image": {
    "nodes": [
      {"id": 1, "type": "entity_type_1", "attributes": {"attribute_1": "value_1"}},
      {"id": 3, "type": "entity_type_3", "attributes": {}}
    ],
    "edges": [
      {"source": 1, "target": 3, "relationship": "relationship_type_2"}
    ]
  }
  "alignment_score": 4
}
```

Important Notes:
If the information is not clearly visible or is too abstract in the image, it should be considered not visible.
Ensure the generated scene graph is based solely on the image content, the caption-based scene graph is just for alignment the textual description.
Assign a score between 1-5 to indicate the degree of alignment between the image-based and caption-based scene graphs (1: poor alignment, 5: perfect alignment).

Figure 3: Image Scene Graph Generation



“A man in a green suit leaning over a laptop”

```
{
  "thoughts": "Entity 'man' was detected as it was mentioned in the text. Attribute 'green
    suit' was included as it provides specific information about the man. Entity 'laptop' was
    also detected as it is explicitly mentioned. The relationship 'leaning over' between 'man’
    and 'laptop' was identified as it describes their spatial interaction.",
  "scene_graph_from_caption": {
    "nodes": [
        {“id”: 1, “type”: “man”, “attributes”, {“suit_color”: “green”}, "importance_score": 0.9},
       {“id”: 2, “type”: “laptop”, “attributes”: {}, “importance_score”: 0.8}
    ],
    "edges": [
        {"source": 1, "target": 2, "relationship": ” leaning over", "importance_score": 0.7},
    ]
  }
}
```

```
{
  "thoughts": "To generate a scene graph from this image, I start by identifying distinguishable
    objects and their attributes. In this image, I see a man wearing a green shirt and using a
    laptop. There is no ambiguity in the main entities and their attributes, as both the man and
    the laptop are clearly visible. There are no additional objects or blurry elements that could
    lead to confusion. I then represent each object as a node with its attributes, and each
    relationship as an edge connecting the nodes. Next, I compare the  image-based scene graph
    with the caption-based scene graph. Both the entities (man and laptop) and their relationship
    (leaning over) are clear in both the image and caption, resulting in high alignment.
    Therefore, I assign a high alignment score. The overall image quality is also high, with
    good lighting and detail, resulting in a high-quality score.",
  "scene_graph_from_image": {
    "nodes": [
        {"id": 1, "type": "man", "attributes": {"shirt_color": "green"}},
        {"id": 2, "type": "laptop", "attributes": {}}
    ],
    "edges": [
        {"source": 1, "target": 2, "relationship": "using"}
    ]
  },
  "alignment_score": 5
}
```

Figure 4: An example of language-conditioned SG generation with the VLM. The example is from from TIFA160 [2], in which the image is generated by `stable-diffusion-v1-5`.

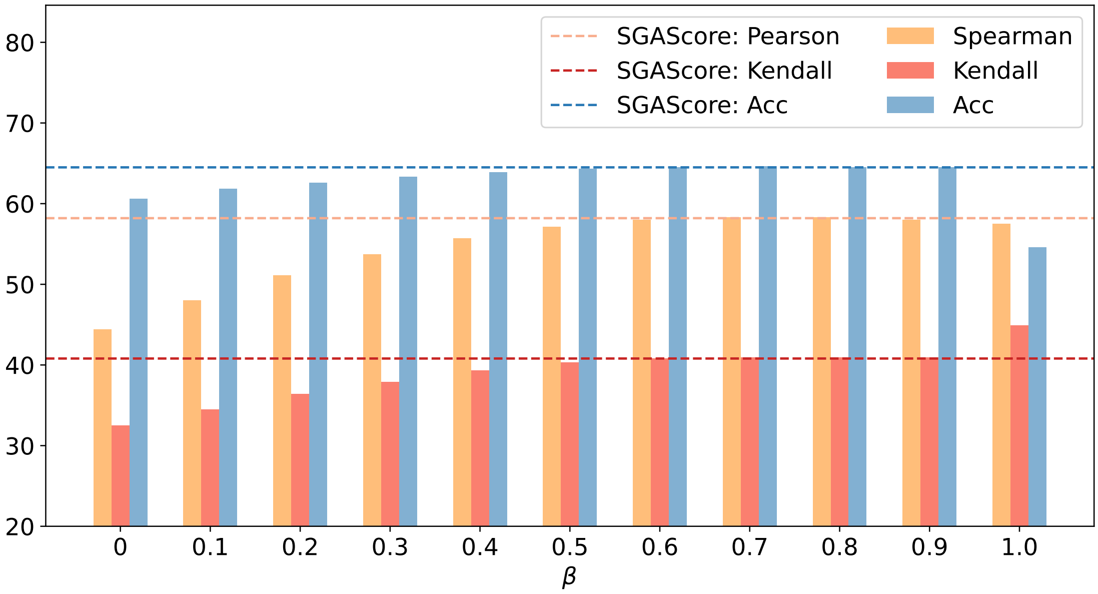


Figure 5: Analysis on the ratio between fine-grained score and global scores. The x-axis is value of β introduced in Equation 9, which means the proportion of the global score when calculating the overall score. The dotted lines are performance of SGAScore, which is calculated by harmonic mean. The y-axis are metrics measuring the evaluation performance. The Pairwise Acc. / Pearson ρ / Kendall τ on TIFA160 are reported. Higher scores are better.

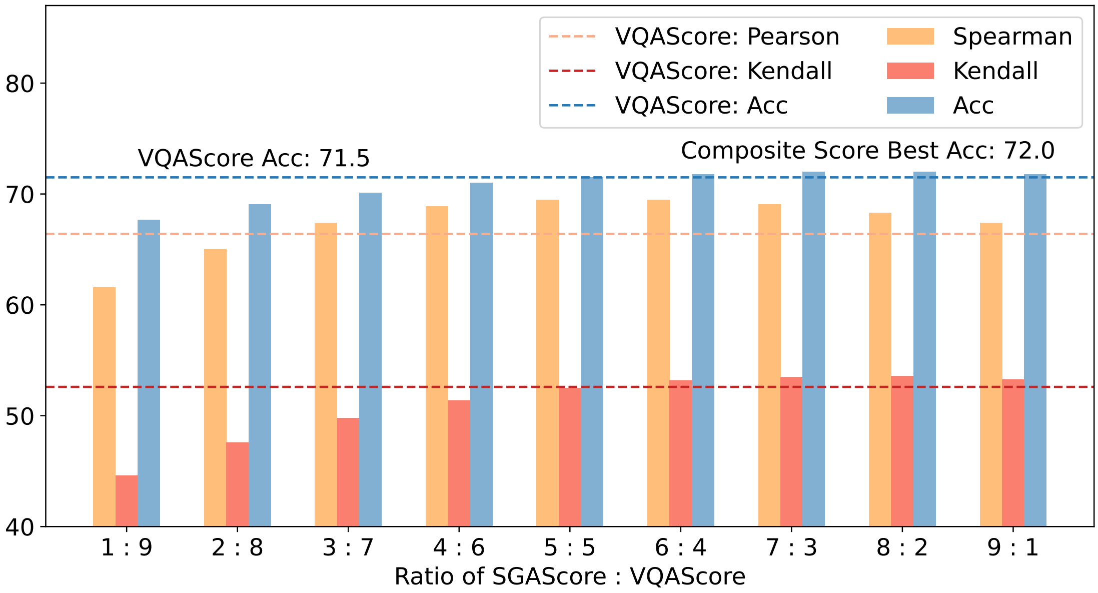


Figure 6: Analysis on the combination of SGAScore and VQAScore. The dotted lines are performance of VQAScore. The x-axis is the ratio of SGAScore to VQAScore. The y-axis are metrics measuring the evaluation performance. The Pairwise Acc. / Pearson ρ / Kendall τ on TIFA160 are reported. Higher scores are better.