How Well Do Vision Models Encode Diagram Attributes?

Anonymous ACL submission

Abstract

Research on understanding and generating diagrams has used vision models such as CLIP. However, it remains unclear whether these models accurately identify diagram attributes, such as node colors and shapes, along with edge colors and connection patterns. This study evaluates how well vision models recognize the diagram attributes by probing the model and retrieving diagrams using text queries. Experimental results showed that while vision models can recognize differences in node colors, shapes, and edge colors, they struggle to identify differences in edge connection patterns that play a pivotal role in the semantics of diagrams. Moreover, we revealed inadequate alignment between diagram attributes and language representations in the embedding space.

1 Introduction

Diagrams, as visual representations of organized information, play a crucial role in effective communication. By combining symbols such as shapes and text, diagrams masterfully convey complex information that might prove challenging to communicate through text alone. Hence, they are widely used in various fields, including business (Havemo, 2018), education (Kembhavi et al., 2016), and academic research (Purchase, 2014).

The widespread usage has attracted significant research interest aimed at understanding diagrams such as captioning (Hsu et al., 2021; Li et al., 2024), visual question answering (VQA) (Kahou et al., 2018; Chaudhry et al., 2019; Wang et al., 2024), and the automatic generation of diagrams based on text (Rodriguez et al., 2023; Belouadi et al., 2023; Zala et al., 2023). This research faces challenges, including understanding geometric shapes and evaluating alignment between text and diagrams. Addressing these challenges requires the development of models that accurately capture the attributes of diagrams and properly align them with language.

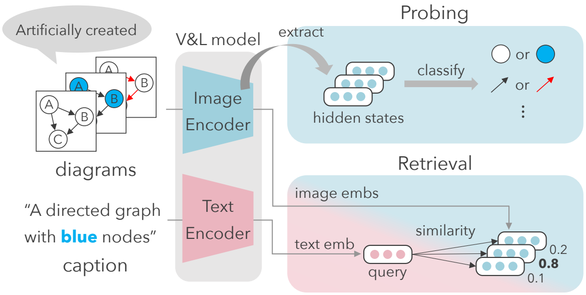

Figure 1: Overview of this study. We examined the extent to which vision models capture the diagram attributes by probing various layers of the vision models. We also investigated whether the diagrams correctly aligned with their textual descriptions through text-based image retrieval.

However, it remains unclear whether vision models capture the attributes of diagrams, such as nodes and edges, and align them with language. For example, diagram comprehension tasks often employ CLIP (Radford et al., 2021) as a visual encoder. In previous studies, the extent to which visual encoders, such as CLIP, can recognize image attributes (e.g., time and object location) has only been done for natural images (Zhang et al., 2024; Lewis et al., 2024). Therefore, the challenge of whether CLIP can adequately encode diagram features remains.

We investigated how well two widely used vision models (CLIP and BLIP (Li et al., 2022)) can capture the attributes of diagrams and align them with language. As shown in Figure 1, we artificially created directed graph-based diagrams as inputs to vision models to perform refined experiments on data with rigorously controlled distributions, which is difficult with manually generated data.[1] We used all layers of the vision models to ascertain whether differences in diagram attributes, such as node color and edge or connection patterns, are re-

[1] This dataset and our codes will be publicly available after this paper is accepted to the conference.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 564–575
August 11-16, 2024 ©2024 Association for Computational Linguistics

flected in feature representations. Furthermore, we conducted text-based image retrieval to examine whether the diagram attributes correctly correspond to their textual descriptions.

The experimental results revealed that the vision models capture attributes such as colors and shapes but not edge connection patterns. Additionally, we found that attributes such as node color are not correctly aligned with their textual descriptions. Our results indicate that models specialized for diagrams are essential for building a model that correctly understands diagrams and for accurately evaluating generated diagrams.

2 Experimental Design

We perform probing to examine how well the vision models recognize the attributes of diagrams, and we perform text-based diagram retrieval to examine whether the models align these attributes with language.

Diagrams are characterized by elements represented by symbols such as nodes or text, and the relationships between these elements (von Engelhardt, 2002; Kembhavi et al., 2016). These relationships are explicitly represented by connecting elements with arrows or enclosing multiple elements together. In other words, diagrams can be considered to have a structure similar to a graph.

2.1 Target Diagrams

We focus on diagrams that can be modeled using directed graphs and investigate whether vision models can recognize nodes and edges. In directed graph-based diagrams, nodes and edges have attributes such as color and shape, and differences in these attributes visually distinguish various information. In addition, the edge connection pattern plays a pivotal role in determining the semantics of a diagram.

We define four attributes for directed graph-based diagrams: **node color**, **node shape**, **edge color**, and **edge connection pattern**. We then create a dataset of directed graphs with three nodes and evaluate how well vision models recognize these attributes.

2.2 Dataset Construction

For each attribute, we define multiple values. Specifically, we prepare five values each for node color, node shape, and edge color, twenty-seven values for edge connection patterns (i.e., edge existence and direction), and ten values for node positions. We create 33,750 diagrams by taking the Cartesian product of these. See §A for details.

3 Probing

3.1 Experimental Settings

We conduct probing using classification models to investigate how well vision models recognize the attributes of diagrams. We construct a classification model to predict the value of a diagram (e.g., red node or blue node) using features extracted from vision models. Based on the performance of the classification models, we evaluate how well the vision models can capture the attributes of diagrams.

As features, we use the hidden states from all layers of the vision models, which are applied average pooling over the sequence, along with the output embeddings. We believe that examining all model layers makes it possible to analyze model characteristics that are difficult to understand by only analyzing the output embeddings. For example, we can conduct a detailed analysis of the model's internals, such as determining which layer acquires specific information and whether the acquired information is subsequently lost.

Probing Method Based on previous research (Heinzerling and Inui, 2024), we construct a regression-based classification model using partial least squares (PLS; Wold et al. (2001)) regression. PLS regression is a linear regression analysis method that employs dimensionality-reduced explanatory variables. Unlike principal component analysis (PCA; Pearson (1901)), PLS regression reduces dimensions by maximizing the covariance between explanatory variables and objective variables. This allows for the selective extraction of information from the explanatory variables by determining appropriate objective variables.

In PLS regression, we input the feature matrix $\boldsymbol{X} \in \mathbb{R}^{n \times h}$ of n samples and labels $\boldsymbol{y} \in \mathbb{R}^{n}$ corresponding to the diagram values as either 0 or 1 (e.g., red node or blue node), to obtain a function $f : \mathbb{R}^{h} \to \mathbb{R}$ (Equation 1).

$$f = \mathrm{PLSRegression}(\boldsymbol{X}, \boldsymbol{y}) \quad (1)$$

The function f takes the feature $\boldsymbol{x}_i$ of a diagram as input and returns a real value r_i.

The output of f is discretized into 0 or 1 using g (Equation 2) with a threshold of $\tau = 0.5$ to obtain

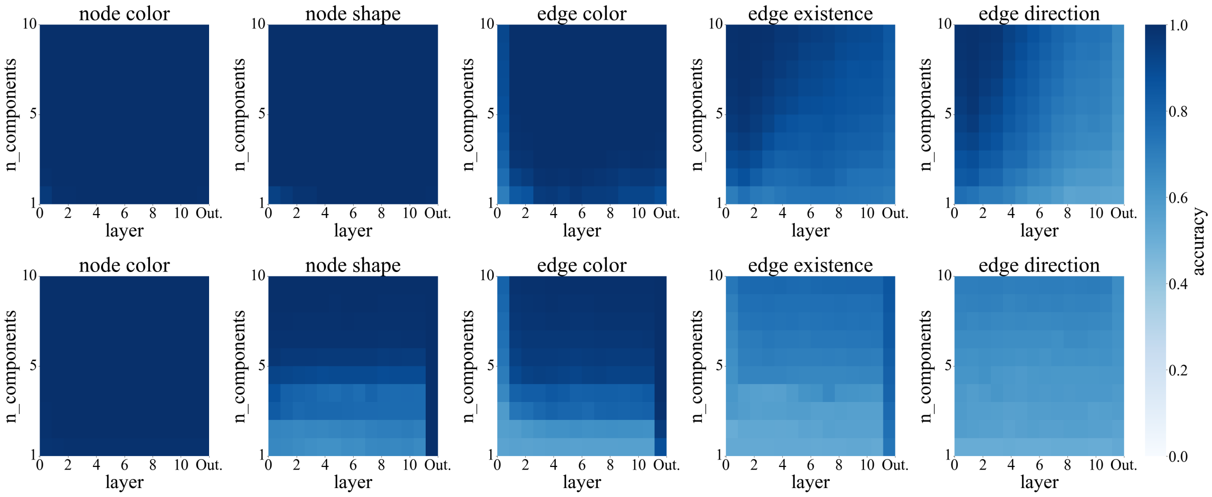

Figure 2: Probing results (Top: CLIP, Bottom: BLIP). The horizontal axis (layer) indicates the vision model's layers, and the vertical axis (n_components) represents the number of components after PLS regression dimensionality reduction. "Out." means output embeddings.

the predicted labels $\hat{y}_i$ (Equation 3).

$$g(r) = \begin{cases} 1 & (r \geq \tau) \\ 0 & (r < \tau) \end{cases} \tag{2}$$

$$\hat{y_i} = g(f(t_i)) \tag{3}$$

We then compute the accuracy between the predicted and ground truth labels to evaluate the performance of the classification model.

The aforementioned analysis is applied to all hidden states and output embeddings of the models. Additionally, by changing the number of components and conducting PLS regression, we analyze how many dimensions of a linear subspace the information on specific attributes is encoded.

Procedure For each attribute, we select two values. We train a model to classify between the two values using the features of diagrams from vision models as input. This classification model training is performed for all combinations of values. The average performance (i.e., accuracy) on the evaluation data for all trained models is regarded as the probing result for that attribute.

Dataset We prepare training and evaluation sets by splitting the subset of diagrams containing the two values into an 8:2 ratio. See §B.2 for hyperparameters.

Models We use CLIP (Radford et al., 2021) and BLIP (Li et al., 2022) as models to compute features of diagrams. Both models learn multimodal representations of images and language and are widely used as vision encoders.

3.2 Results

Figure 2 shows the results of probing.

Color and shape information is retained in most layers and output embeddings. Both models achieved high accuracy across most layers and embeddings for node color, node shape, and edge color. This suggests that both models capture information on these attributes in the early layers. Furthermore, achieving high accuracy with few components indicates that this information is retained in a low-dimensional subspace.

The information about edge connection patterns may not be retained in the output embeddings. Both models showed lower accuracy in the output embeddings for edge existence and direction than other attributes, suggesting that the information on these attributes might not be encoded in the output embeddings. Furthermore, the accuracy of the hidden states showed different trends for each model. BLIP consistently exhibited low accuracy across all layers, whereas CLIP achieved relatively high accuracy in the early layers, which then decreased in later layers. These results indicate that CLIP may lose information encoded in the early layers or encode it into complex, high-dimensional subspaces that are difficult to extract as the layers progress.

The linear layer may reduce the dimensions of the subspace retaining information BLIP achieved higher accuracy in classifying node shape and edge color using output embeddings with fewer components than using hidden states at the last layer. This result indicates that the linear projection used to compute the output embeddings from the hidden states might contribute to encoding the information on node shape and edge color into a lower-dimensional subspace.

4 Text-based Diagram Retrieval

	mAP@100				MRR@100			
	node		edge		node		edge	
	color	shape	color	conn.	color	shape	color	conn.
Rand.	.234	.234	.234	.362	.405	.405	.405	.550
CLIP	.868	.595	.513	.313	.907	.602	.685	.419
BLIP	.208	.206	.212	.394	.233	.241	.315	.485

Table 1: Results of text-based diagram retrieval. Scores for Rand. are chance rates. "conn." means edge connection patterns.

4.1 Experimental Settings

We perform text-based diagram retrieval to investigate whether the vision models properly align the diagram attributes with language.

We use the same set of diagrams $D = \{d_1, d_2, \ldots, d_{32750}\}$ described in §2 as the retrieval target and the caption c that describes the diagrams as the query. We use CLIP and BLIP as vision models. The diagrams and captions are fed into the vision model to obtain the diagram features $\boldsymbol{v}_{d_i}$, and the caption features $\boldsymbol{v}_c$. For each diagram, we compute the cosine similarity $\cos(\boldsymbol{v}_{d_i}, \boldsymbol{v}_c)$ with the caption, selecting the top 100 diagrams based on the highest similarity scores as the retrieval results.

Queries For queries, we create captions that describe the diagrams. Each caption specifies the value of diagrams (e.g., A directed graph with red nodes.). As described in §2, there are five values each for node color, node shape, and edge color. There are also three values for edge connection patterns: no edge, an edge directed forward (e.g., from node A to B), and an edge directed backward (e.g., from node B to A).

To ensure diversity, we use GPT-3.5 (OpenAI, 2022) to paraphrase and generate 10 captions for each value. We manually correct captions that are not properly paraphrased. See §C.1 for an example of captions.

Evaluation Metrics We evaluate retrieval results using mean average precision (mAP) (Everingham et al., 2010) and mean reciprocal rank (MRR) (Craswell, 2009) for each diagram attribute.

4.2 Results

Table 1 shows the results of retrieval.

CLIP generally aligns colors and shapes with language CLIP outperformed the chance rate across all metrics for node color, node shape, and edge color. However, the scores for edge connection patterns were comparable to the chance rate. These findings align broadly with the results from probing described in Section 3.2.

BLIP's performance was consistently at or below the chance rate across all attributes, suggesting a misalignment between the attributes and the language. Furthermore, the MRR scores for node color, node shape, and edge color underperformed relative to the chance rate. To understand the reason behind this, we analyzed the retrieved diagrams. We found that the top 100 retrieved diagrams excessively include those with a specific value (e.g., diagrams with white nodes). This indicates that there is a bias resulting in disproportionately high similarity for diagrams with specific values. See §C.2 for more details. Identifying the cause of this bias is a task for future work.

5 Conclusion

We conducted probing and text-based diagram retrieval experiments to investigate how well commonly used vision models recognize diagram attributes and align them appropriately with language. Our findings indicate that, while these models can identify differences in color and shape, they struggle with more semantic attributes such as edge connection patterns. Furthermore, we have also identified open issues related to language alignment, such as the effects of bias on specific diagrams.

Our next goal is to develop a model that is better capable of encoding diagram attributes, including edge connection patterns, into a unified embedding space. To accomplish this effectively, we plan to study sophisticated ways to train vision models with diagram datasets. Once we establish such a comprehensive vision encoder that is fully capable of diagram embeddings, we can use it as a solid basis to explore downstream diagram understanding tasks such as captioning and VQA and the automatic evaluation metrics for text-to-diagram generation.

References

Jonas Belouadi, Anne Lauscher, and Steffen Eger. 2023. Automatikz: Text-guided synthesis of scientific vector graphics with tikz. *CoRR*, abs/2310.00367.

Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi. 2019. LEAF-QA: Locate, encode & attend for figure question answering. *Proc. IEEE Workshop Appl. Comput. Vis.*, pages 3501–3510.

Nick Craswell. 2009. Mean reciprocal rank. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, page 1703. Springer US.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. 2010. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338.

Emelie Havemo. 2018. A visual perspective on value creation: Exploring patterns in business model diagrams. *European Management Journal*, 36(4):441–452.

Benjamin Heinzerling and Kentaro Inui. 2024. Monotonic representation of numeric properties in language models. *CoRR*, abs/2403.10381.

Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. 2021. SciCap: Generating captions for scientific figures. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3258–3264, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2018. Figureqa: An annotated figure dataset for visual reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 235–251. Springer.

Martha Lewis, Nihal Nayak, Peilin Yu, Jack Merullo, Qinan Yu, Stephen Bach, and Ellie Pavlick. 2024. Does CLIP bind concepts? probing compositionality in large image models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1487–1500, St. Julian's, Malta. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. 2024. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. *CoRR*, abs/2403.00231.

OpenAI. 2022. Introducing chatgpt. `https://www.openai.com/chatgpt`.

Karl Pearson. 1901. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Helen C Purchase. 2014. Twelve years of diagrams research. *Journal of Visual Languages & Computing*, 25(2):57–75.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Juan A. Rodriguez, David Vázquez, Issam H. Laradji, Marco Pedersoli, and Pau Rodríguez. 2023. Figgen: Text to scientific figure generation. In *The First Tiny Papers Track at ICLR 2023, Tiny Papers @ ICLR 2023, Kigali, Rwanda, May 5, 2023*. OpenReview.net.

Jörg von Engelhardt. 2002. *The Language of Graphics: A Framework for the Analysis of Syntax and Meaning in Maps, Charts and Diagrams*. Ph.D. thesis, University of Amsterdam.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. *CoRR*, abs/2402.14804.

Svante Wold, Michael Sjöström, and Lennart Eriksson. 2001. PLS-regression: a basic tool of chemometrics. *Chemometrics Intellig. Lab. Syst.*, 58(2):109–130.

Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. 2023. Diagrammergpt: Generating open-domain, open-platform diagrams via LLM planning. *CoRR*, abs/2310.12128.

Zhang, Zhang, Zhang, and Tresp. 2024. Can Vision-Language models be a good guesser? exploring VLMs for times and location reasoning. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, volume 0, pages 625–634.

A Dataset Details

Table 2 shows the values of each attribute of diagrams.

B Probing Details

B.1 Subset of Dataset

Table 3 shows the size of subsets of the dataset in probing.

B.2 Regression Model Training

We used the PLSRegression class from scikit-learn to train regression models. Table 4 shows the hyperparameters.

C Retrieval Details

C.1 Caption Examples

Table 5 shows examples of captions used for the retrieval task.

C.2 Examples of Actual Retrieval Results

For each model, the top 100 diagrams with the highest cosine similarities are shown in Figure 4, 5, 6, and 7. Figure 5 and 7 indicate a bias in BLIP's retrieval results.

attribute	values	number of values
node color	white, red, blue, green, yellow	5
node shape	circle, triangle, square, pentagon, hexagon	5
edge color	black, red, blue, green, yellow	5
edge connection pattern	(no edge, forward, backward) × 3 node pairs	27

Table 2: Variations in the values of each attribute.

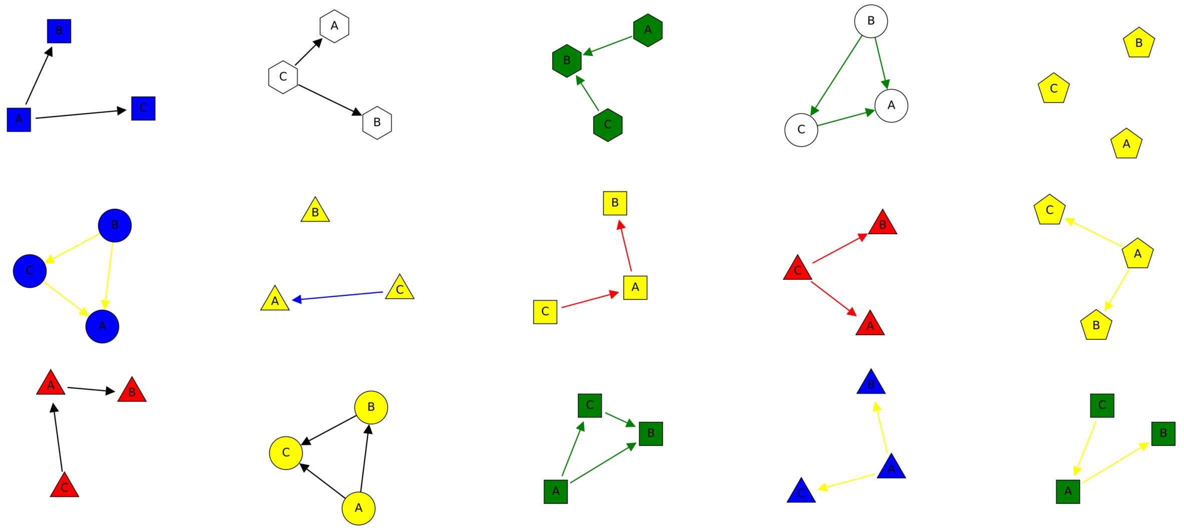

Figure 3: Diagrams included in our dataset.

node		edge	
color	shape	color	conn.
13,500	13,500	13,000†	22,500

Table 3: Subset size of the dataset in probing. † For edge color classification, we excluded data with no edges, resulting in fewer data samples than those of node color and node shape classification.

number of samples	80% of subset
scale	True
max_iter	500
tol	1e-06
copy	True

Table 4: Hyperparameters for PLS regression.

attribute=value	caption
node color=white	A directed graph with white nodes. A diagram featuring nodes in white. An image with nodes that are white. A graph where the nodes are in white color. In this graph the nodes are depicted in white. The diagram includes nodes colored white. The graph displays nodes that are white in color. The graph contains nodes that are white in color. The nodes in the graph are white. An illustration featuring white-colored nodes in the graph.
node shape=circle	A graph with circular nodes. A diagram featuring nodes that are circular in shape. In this graph the nodes are represented as circles. The graph includes nodes with a circular form. Circular nodes are present in the graph. Nodes within the graph are depicted as circles. A visual representation featuring circular nodes in the graph. On the graph nodes are displayed in a circular fashion. The nodes in the graph take on a circular appearance. The graph displays nodes that are circular in nature.
edge color=black	A directed graph with a black edge. A graph displaying a directed connection with a black edge. An image of a directed graph featuring one black edge. In this directed graph there is a single black edge. A diagram showing a directed link with a black arrow. The graph includes a black edge indicating direction. A visual representation of a directed relationship using a black edge. A single black edge signifies direction in the graph. The graph features a directed connection represented by a black edge. Within the directed graph there is a solitary black edge denoting direction.
edge direction=A → B	A directed graph with an edge stretched from A to B. A graph where there’s a directed edge extending from point A to point B. An edge pointing from A to B in a directed graph. In a directed graph there’s an edge connecting A to B. A graph displaying a directional connection from A to B. The graph has a directed link that runs from A to B. An arrow indicates the direction from A to B on the graph. A visual representation showing a directed path from A to B. The graph has a directed edge from A to B. There is an edge stretching from A to B in the diagram.

Table 5: Examples of captions used as a query for retrieval.

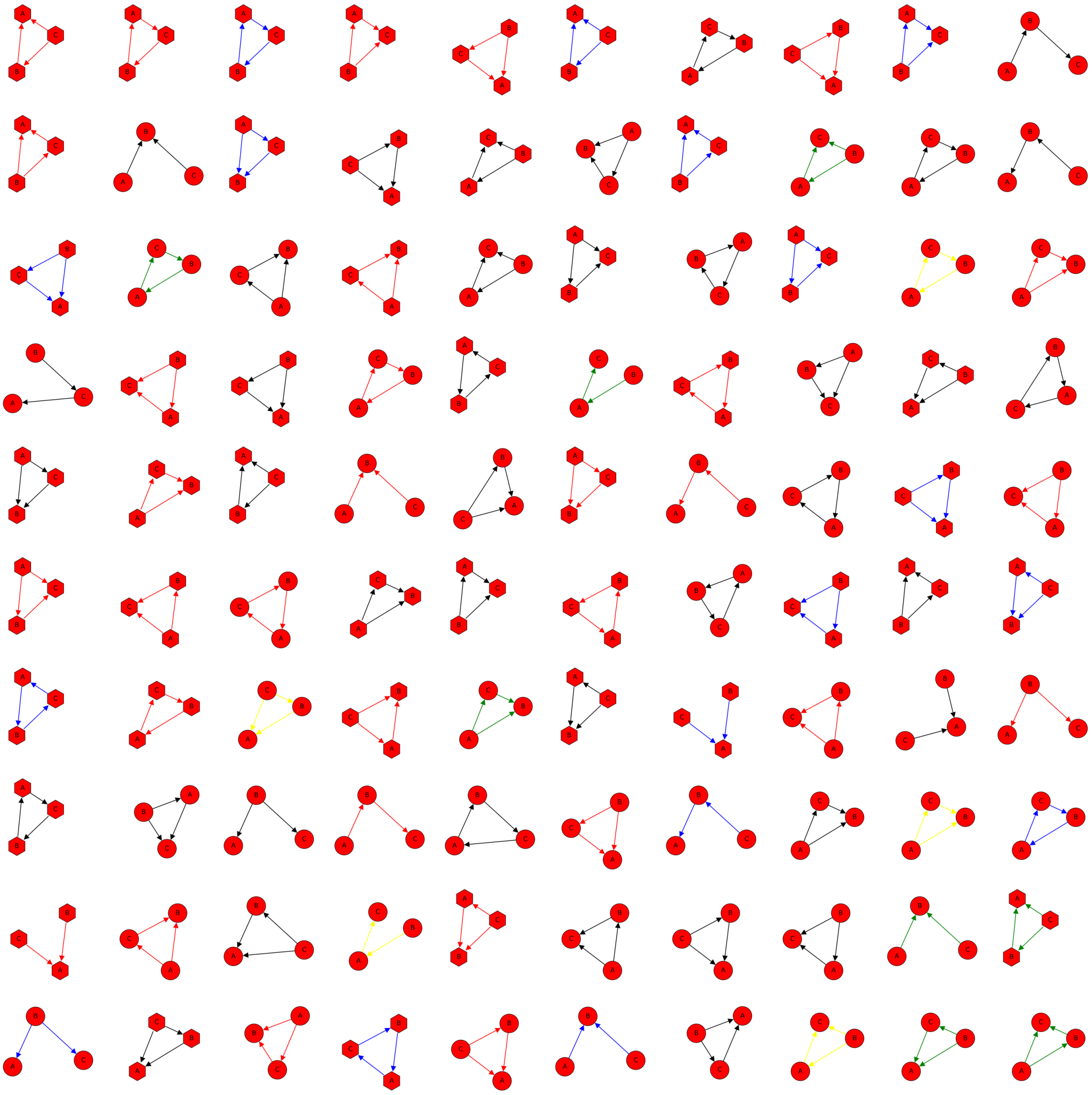

Figure 4: A retrieval result of CLIP for the caption “A directed graph with red nodes.”. All top 100 diagrams have red nodes, consistent with the caption description.

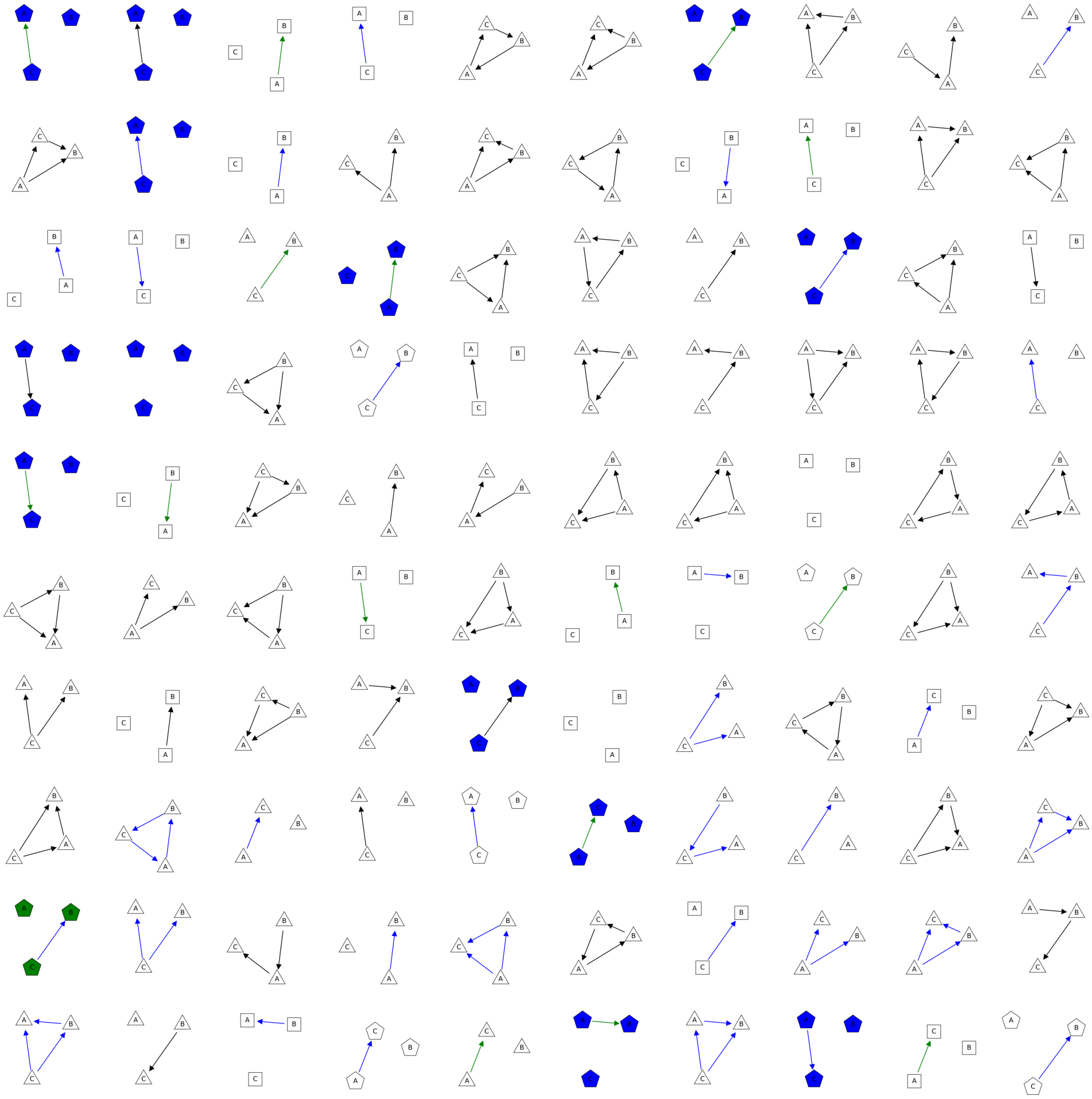

Figure 5: A retrieval result of BLIP for the caption “A directed graph with *red* nodes.”. None of the top 100 diagrams have red nodes; instead, they predominantly consist of white and blue nodes.

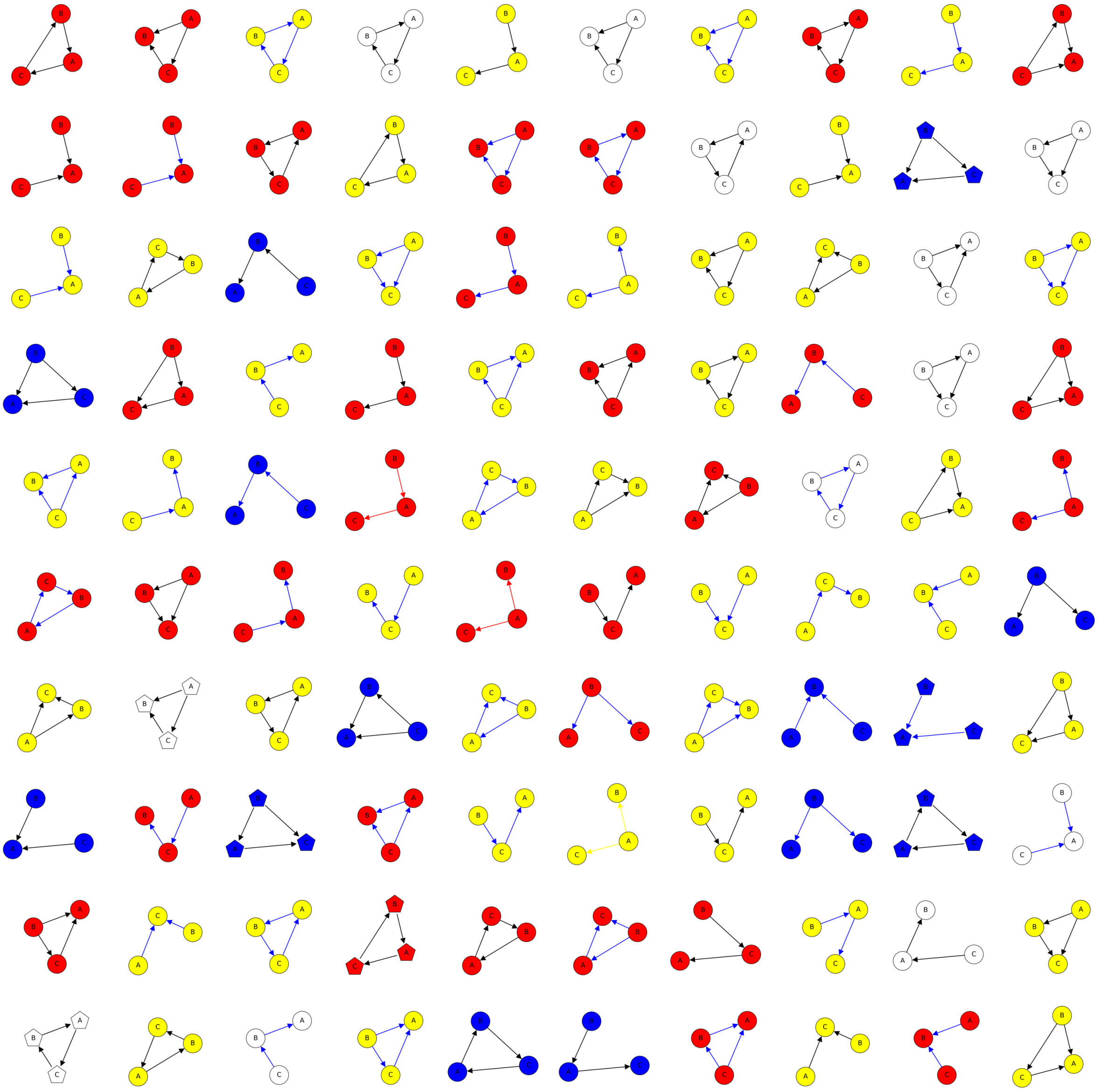

Figure 6: A retrieval result of CLIP for the caption “A directed graph with an edge stretched from A to B.”. This result includes diagrams with edges directed from A to B, diagrams with edges directed from B to A, and diagrams with no edge between A and B. Therefore, these results do not align with the content of the caption.

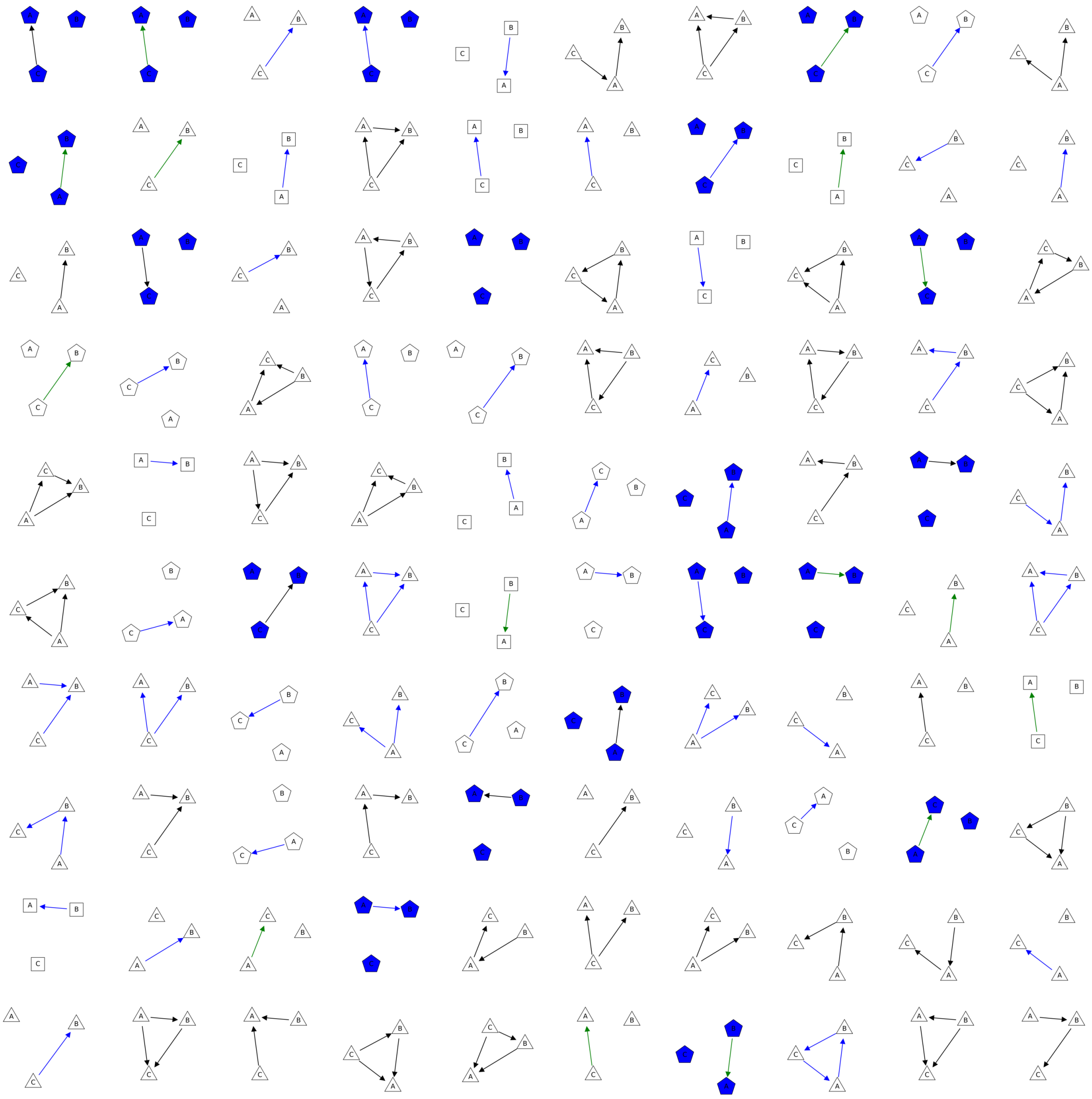

Figure 7: A retrieval result of BLIP for the caption “A directed graph with an edge stretched from A to B.”. This result includes diagrams with edges directed from A to B, diagrams with edges directed from B to A, and diagrams with no edge between A and B. Therefore, these results do not align with the content of the caption. Similar to the results in Figure 5, the top 100 diagrams consisted solely of white and blue nodes.