

Signals Are Present, Predictions Fail: Probing Diagram Representations in Large Vision-Language Models

Haruto Yoshida¹, Keito Kudo¹, Yoichi Aoki¹, Ryota Tanaka²,
Itsumi Saito¹, Keisuke Sakaguchi¹, Kentaro Inui¹

¹Tohoku University, ²Human Informatics Labs., NTT, Inc.

{yoshida.haruto.p1, keito.kudo.q4, youichi.aoki.p2}@dc.tohoku.ac.jp,
ryota.tanaka@ntt.com, {itsumi.saito, keisuke.sakaguchi}@tohoku.ac.jp,
kentaro.inui@mbzuai.ac.ae

Abstract

Large vision-language models (LVLMs) demonstrate strong performance on diagram understanding benchmarks, yet they still struggle to understand relationships between elements, particularly those represented by directed edges between nodes. To investigate the causes of this limitation, we probe the internal representations of LVLMs using a synthetic diagram dataset based on directed graphs. Our linear probing experiments reveal that signals about nodes and edges are linearly decodable from the hidden states of LVLMs, even for aspects that LVLMs struggle with in VQA, such as edge direction and connectivity between nodes. These results reveal a gap between internal signals and model outputs. We further perform causal interventions on the model’s hidden states, showing that, for some aspects, the signals identified through probing causally contribute to the model’s final predictions. These findings also hold on flowcharts, not only on synthetic diagrams.

1 Introduction

Diagrams convey complex information through visual layouts that people can grasp quickly, making them an important communication medium across various domains, including academic research (Hsu et al., 2021; Kohn and Aladjem, 2006) and business (Jošt et al., 2017; Kernbach and Eppler, 2010). Accordingly, various systems for diagram understanding have been studied (Kembhavi et al., 2016; Li et al., 2025).

Large vision-language models (LVLMs) (Bai et al., 2025a,b; Liu et al., 2024; Team, 2025) have shown strong performance on diagram understanding benchmarks (Zhang et al., 2025; Lu et al., 2024). However, prior studies report that LVLMs still struggle to understand certain visual elements in diagrams, such as relational information indicated by arrows and lines (Hou et al., 2025; Zhu et al., 2025; Pan et al., 2024). To better understand

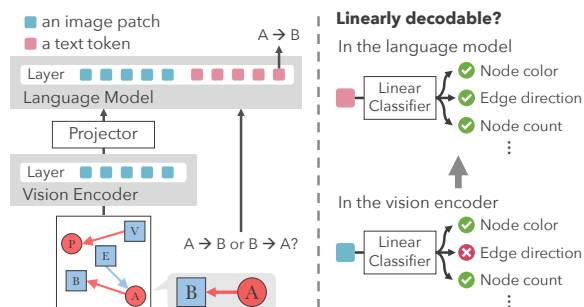


Figure 1: Overview of this study. We analyze the internal representations of LVLMs using linear probing and causal interventions on a synthetic diagram dataset. Even for aspects that LVLMs fail to answer in the VQA setting, the corresponding signals remain linearly decodable from their hidden states, revealing a gap between internal signals and model outputs.

the causes of these limitations, it is important to examine how LVLMs internally represent diagram information, yet how they represent basic visual elements such as nodes, edges, and global structure remains underexplored.

To fill this gap, we comprehensively investigate how diagram-specific visual elements are internally distinguished in LVLMs. We focus on whether information about nodes, edges, and global structure is represented internally in the vision encoder and the language model. In contrast to prior analyses that focus on natural images (Gandelsman et al., 2024; Tao et al., 2024), we target diagrams, which rely on symbolic representations to illustrate structured relationships. To enable controlled analysis, we construct a synthetic diagram dataset. We apply probing (Alain and Bengio, 2017) to examine the linear decodability of node, edge, and global information, and causal interventions (Zhang and Nanda, 2024) to assess whether the detected signals contribute to model predictions.

Our contributions are threefold. First, we introduce a synthetic diagram dataset with controllable

Category	Aspect a	Label set $\mathcal{Y}$
Single	Node Color	{red, green, yellow, blue, brown, orange, pink, purple}
	Node Shape	{circle, square, pentagon, hexagon, septagon}
	In-deg.	{0, 1, 2, 3, 4}
	Out-deg.	{0, 1, 2, 3, 4}
Multiple	Edge Color	{red, green, yellow, blue, brown, orange, pink, purple}
	Edge Style	{solid, dashed}
	Edge Exist.	{exist, not exist}
	Edge Direct.	{forward, backward}
	Multi-hop Path	{exist, not exist}
Global	Node Count	{1, 2, 3, 4, 5}
	Edge Count	{1, 2, 3, 4, 5}

Table 1: The 3 categories, 11 aspects, and their corresponding label sets $\mathcal{Y}$.

visual elements, enabling systematic analysis while reducing the influence of domain-specific knowledge. Second, our probing analysis shows that signals about nodes and edges are linearly decodable from the hidden states of either the vision encoder or the language model. Importantly, these signals remain linearly decodable even for aspects that the model struggles with in VQA, revealing a gap between internal signals and model outputs. This suggests that the difficulty may lie not in the absence of relevant signals, but in the model’s ability to use them. Third, our causal interventions show that, for several aspects, the signals identified through probing causally contribute to the model’s predictions. We further confirm that these findings hold on flowcharts, indicating that they are not specific to our synthetic diagrams. These findings provide insight into how diagram information is represented and used within LVLMs.

2 Task and Model Setup

As a testbed for our analysis, we evaluate the VQA task on synthetically created simple diagram images (graphs). This VQA task can be solved as a classification task. As shown in Figure 1, the LVLM takes as input a synthetically created simple diagram image x and a question q , and outputs the answer $y \in \mathcal{Y}$ (where $\mathcal{Y}$ is the set of answer labels) to the question as text.

In our experiments, we primarily focus on Qwen3-VL-8B-Instruct (Bai et al., 2025a) as the main target of analysis. In addition, we conduct the same experiments on Qwen2.5-VL 7B (Bai et al., 2025b), LLaVA1.5 7B (Liu et al., 2024), and Gemma3-4B-IT (Team, 2025). These results are

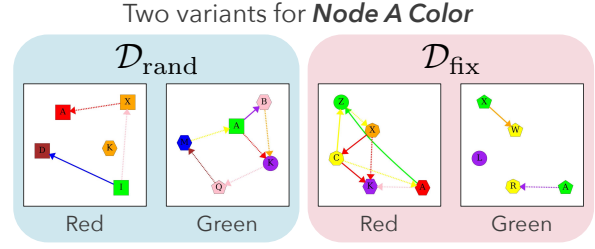


Figure 2: Examples of synthetic diagrams. Each diagram contains five nodes, and we control evaluation aspects such as node color, shape, and edge connectivity. We provide two variants: $\mathcal{D}_{\text{rand}}$, which uses random node layouts (left part), and $\mathcal{D}_{\text{fix}}$, which uses fixed layouts (right part).

reported in Appendix D and E.

3 Dataset Construction

To investigate how LVLMs perceive basic visual elements in diagrams, we construct a synthetic dataset of directed-graph diagrams. The diagrams consist of nodes and edges and serve as a simplified approximation of real-world diagrams, retaining only their core visual elements. This setup allows precise control over evaluation factors such as node color and edge direction. Moreover, synthetic data helps reduce biases and shortcuts inherited from natural-image and natural-language datasets (Kafle and Kanan, 2017; Gururangan et al., 2018; Dancette et al., 2021; Healey et al., 2024), enabling fine-grained analysis and evaluation.

3.1 Diagram Specification

Following prior work on diagram understanding (Zhu et al., 2025; Sartori et al., 2025), each diagram satisfies the following properties: (1) Nodes and edges are colored using one of eight colors: red, green, yellow, blue, brown, orange, pink, or purple. (2) Each diagram contains 5 nodes. (3) Node shapes are one of circle, square, pentagon, hexagon, or septagon. (4) Each node is labeled with a single alphabetic character as an identifier. (5) Edge styles are either solid or dashed. Further details are provided in Appendix B.1.

3.2 Evaluation Aspects

We define 11 aspects as basic visual information in diagrams, as shown in Table 1. These aspects are based on visual information emphasized in prior works (Silva et al., 2011; Pan et al., 2024; Sterzik et al., 2024; Tseng et al., 2025; Hou et al., 2025; Zhu et al., 2025).

Diagram understanding fundamentally requires composing multiple visual components, primarily through binding visual features with objects and aggregating information across components. These operations apply at different spatial scopes, based on which we classify the 11 aspects into the following three categories:

- **Single:** Visual information localized around a single node.
- **Multiple:** Combinations of multiple localized pieces of visual information. Since edges are defined between two nodes, their recognition requires a multi-step process: first identifying the endpoint nodes, then recognizing the edge attributes between them.
- **Global:** Visual information that cannot be determined without considering all nodes and edges in the diagram.

In analyzing each aspect, we focus on attributes of the node with identifier A (node A) and the edge connecting nodes A and B (edge AB) as representatives¹. For example, in the Node Color aspect, we evaluate whether the color of node A is represented (i.e., linearly decodable) in the model. The Node Count aspect is an exception: multiple nodes share the identifier A and the task is to count them, since the total number of nodes is fixed at five.

3.3 Dataset Definition

For each aspect a , we define a dataset $\mathcal{D}$ to verify whether it can be recognized as follows:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N. \quad (1)$$

Here, x_i is the i -th diagram image and y_i is its label. Specifically, this is a dataset containing labels y_i for each aspect as shown in Table 1.

In this study, we train probes to analyze the internal representations of an LVLM. To train and evaluate these probes, for each aspect, we construct two variants with different node layouts:

- $\mathcal{D}_{\text{rand}}$: For each diagram in the dataset, the node layout is determined independently at random.
- $\mathcal{D}_{\text{fix}}$: For all diagrams in the dataset, nodes with each identifier are placed at the same spatial position.

¹The choice of identifiers is arbitrary, and the same analysis can be applied to others such as node C.

Figure 2 shows examples of each variant. For each aspect, both $\mathcal{D}_{\text{rand}}$ and $\mathcal{D}_{\text{fix}}$ consist of 100 data points per classification class.

Training dataset for probing. We construct the diagram dataset with random node layouts, $\mathcal{D}_{\text{train}}$, as the base for the probe training dataset. In pilot experiments, we observed that the probe achieved near-perfect accuracy on certain patches by exploiting shortcuts, such as predicting based on colors present in the input patch without identifying the target node or edge (see Appendix D.1 for details). To prevent this, we prepare an additional dataset $\mathcal{D}_{\text{rand}}^{\perp}$, which consists of diagrams in which the target node or edge does not exist, with the gold label $y = \perp$ (e.g., the question asks about node A, but node A does not exist in the image). This forces the probe to verify target existence before predicting its attributes, which resolved the shortcut issue in our experiments. $\mathcal{D}_{\text{rand}}^{\perp}$ contains the same number of samples as $\mathcal{D}_{\text{rand}}$. Using $\mathcal{D}_{\text{rand}}^{\perp}$, we define the probe training dataset as $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{rand}} \cup \mathcal{D}_{\text{rand}}^{\perp}$.

Evaluation dataset for probing. We prepare the evaluation dataset by constructing M variants of $\mathcal{D}_{\text{fix}}$. Formally, denoting the j -th evaluation subset as $\mathcal{D}_{\text{fix}}^j$, the entire evaluation dataset is expressed as $\mathcal{D}_{\text{test}} = \{\mathcal{D}_{\text{fix}}^0, \mathcal{D}_{\text{fix}}^1, \dots, \mathcal{D}_{\text{fix}}^{M-1}\}$. The fixed node layout, which is not seen during probe training, allows us to analyze the model’s internal representations in a controlled manner, and evaluating across M variants ensures that the observed behaviors are not artifacts of a specific spatial configuration.

4 Preliminary Experiment: VQA

As a preliminary experiment, we evaluate the VQA performance of LVLMs on synthetic diagrams to confirm above-chance-level performance. Specifically, we assess whether the models can correctly predict the gold label y given a diagram x and question q in a simple VQA setting. The question q includes the label set $\mathcal{Y}$ as answer choices. For example, for the Node Color aspect, the question is “What color is node A?” with choices $\mathcal{Y} = \{\text{red}, \text{green}, \dots\}$. The same input format is used in the probing and causal intervention experiments. The full list of questions for each aspect is provided in Table 6.

4.1 Evaluation Metric

We define the final score as the mean accuracy across all evaluation data subsets. Formally, it is

	Node Color	Node Shape	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Chance level	12.5	20.0	20.0	20.0	12.5	50.0	50.0	50.0	50.0	20.0	20.0
Qwen3-VL 8B	95.5	78.4	53.7	40.6	80.6	85.7	81.2	53.9	55.9	78.7	83.8

Table 2: Acc_{VQA} for Qwen3-VL 8B. Accuracy exceeded chance level for all aspects, but Edge Direction and Multi-hop Path remained near chance, while other aspects achieved substantially higher accuracy.

defined as:

$$\text{Acc}_{\text{VQA}}(q, \mathcal{D}_{\text{test}}) = \text{mean}_{d_{\text{fix}} \in \mathcal{D}_{\text{test}}} \text{acc}_{\text{VQA}}(q, d_{\text{fix}}). \quad (2)$$

Here, acc_{VQA} is the accuracy of the LVLM on the VQA task for dataset $d_{\text{fix}} \in \mathcal{D}_{\text{test}}$, and $\text{mean}_{s \in \mathcal{S}}(\cdot)$ denotes the element-wise arithmetic mean of the values associated with each $s \in \mathcal{S}$. We judge a prediction as correct if the LVLM’s output text contains the label y as a substring.

4.2 Experimental Results

Table 2 summarizes the Acc_{VQA} of Qwen3-VL 8B. The model achieved high accuracy on aspects such as Node Color and Edge Style, while accuracy on Edge Direction and Multi-hop Path remained near chance level. This suggests that Qwen3-VL recognizes basic visual attributes but struggles with relational properties that require identifying edge directions or tracing multi-step paths. See Appendix A for additional discussion.

5 Probing

We investigate whether visual information in diagrams is linearly encoded within LVLMs. To this end, we perform linear probing on the hidden states of each layer and patch in both the vision encoder and the language model, and examine whether there are hidden states from which such signals can be captured. Linear probing is an effective method for identifying which parts of a model contain signals corresponding to specific information (Alain and Bengio, 2017).

5.1 Probe Definition

Let $\mathbf{h}_{l,t}$ denote the hidden state at layer l and position t of the vision encoder or language model given diagram x and question q . The probe f that predicts the VQA task answer or $\perp$ from the hidden state is defined as a single-layer linear classifier:

$$\hat{y} = f(\mathbf{h}_{l,t}) = \arg \max_{\mathcal{Y} \cup \{\perp\}} \mathbf{W} \mathbf{h}_{l,t} + \mathbf{b}, \quad (3)$$

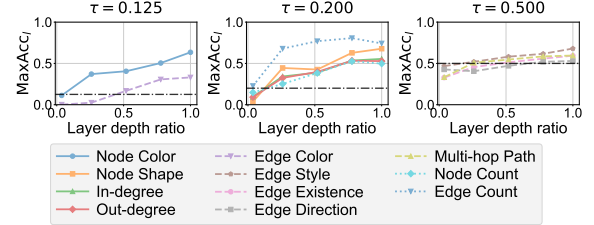


Figure 3: MaxAcc_l per layer in the vision encoder of Qwen3-VL 8B. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . Accuracy for Single- and Global-category aspects rose with depth, whereas Multiple-category aspects stayed near the threshold throughout.

where $\hat{y} \in \mathcal{Y} \cup \{\perp\}$ denotes the predicted answer label, $\mathbf{W} \in \mathbb{R}^{(|\mathcal{Y}|+1) \times |\mathbf{h}_{l,t}|}$ is the weight matrix and $\mathbf{b} \in \mathbb{R}^{(|\mathcal{Y}|+1)}$ is the bias term.

We train probes for three components using the training dataset $\mathcal{D}_{\text{train}}$ described in § 3.3: (i) the vision encoder and (ii) the image-input part of the language model, for which we train a per-layer probe f_l from image-patch hidden states $\mathbf{h}_{l,t}$ at all patch positions t ; and (iii) the text-input part of the language model, for which we train a separate probe $f_{l,t}$ for each layer l and each token position t in question q . Training details, including hyperparameters and how we address shortcut learning of probes, are provided in Appendix D.

5.2 Evaluation Metric

Let $\text{acc}_{l,t}(d_{\text{fix}})$ denote the accuracy of probe f (the match rate between $\hat{y}$ and y) when using the hidden state at layer l and position t on evaluation data subset $d_{\text{fix}} \in \mathcal{D}_{\text{test}}$. The final evaluation score is the mean accuracy across all subsets:

$$\text{Acc}_{l,t}(\mathcal{D}_{\text{test}}) = \text{mean}_{d_{\text{fix}} \in \mathcal{D}_{\text{test}}} \text{acc}_{l,t}(d_{\text{fix}}). \quad (4)$$

To obtain a per-layer score of how linearly encoded the target information is, we define the maximum probing accuracy MaxAcc_l at layer l :

$$\text{MaxAcc}_l(\mathcal{D}_{\text{test}}) = \text{mean}_{d_{\text{fix}} \in \mathcal{D}_{\text{test}}} (\max_t \text{acc}_{l,t}(d_{\text{fix}})). \quad (5)$$

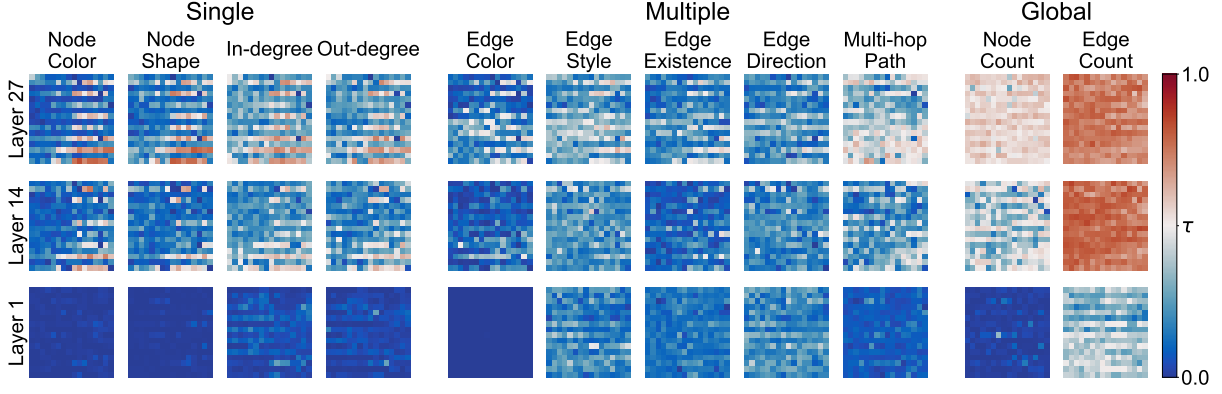


Figure 4: Position-wise accuracy in the vision encoder of Qwen3-VL 8B. Each heatmap shows accuracy by patch position for a specific layer and aspect. The evaluation diagrams use the node layout shown in the right part of Figure 2. Single-category aspects localized near node positions (bottom-right), Global-category aspects spread across background patches, and Multiple-category aspects showed few high-accuracy patches.

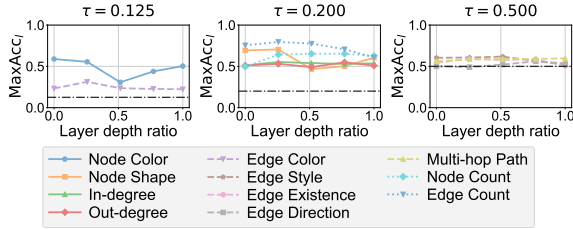


Figure 5: MaxAcc_l per layer in the image-input part of the language model in Qwen3-VL 8B. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . Compared with the final vision encoder layer, accuracy showed little change overall.

That is, MaxAcc_l takes the maximum over positions and averages across subsets. We introduce a threshold $\tau = 1/|\mathcal{Y}|$ (chance-level accuracy excluding $\perp$), and consider probing successful when $\text{Acc}_{l,t} > \tau$.

5.3 Experimental Results

Signals for Single (node) and Global (counting) aspects become linearly decodable in the vision encoder. Figure 3 reports MaxAcc_l at each layer of the vision encoder. For aspects in the Single and Global categories, accuracy increased with depth, suggesting that the linear decodability of signals for these aspects increases in deeper layers. In contrast, accuracies for aspects in the Multiple category remained lower than those for the other categories even in the final layers, suggesting that signals for these aspects are more difficult to decode linearly from individual hidden states of the vision encoder.

Signals for Single (node) aspects are localized near node positions, while those for Global (counting) aspects are broadly distributed across patch positions. Figure 4 visualizes $\text{acc}_{l,t}(d_{\text{fix}})$ as heatmaps across vision encoder layers and patch positions. We use the subset d_{fix} as the examples shown in the right part of Figure 2 (i.e., diagrams sharing the same node layout). For aspects in the Single category, high-accuracy patches were concentrated around the target node A in the bottom-right region. For aspects in the Global category, accuracy exceeded the threshold at almost all positions in later layers, suggesting that signals for these aspects are broadly distributed across patch positions, including background regions. This high accuracy on background patches suggests that hidden states corresponding to background positions may serve as storage or aggregation points for global information. Prior work on register tokens (Darcet et al., 2024) in Vision Transformers (Dosovitskiy et al., 2021) reports that global information is concentrated in a small subset (around 2%) of background tokens. In contrast, our results suggest that signals for aspects in the Global category are distributed across nearly all background patches in our diagram setting.

Signals remain linearly decodable in the image-input part of the language model. Figure 5 shows MaxAcc_l across language model layers. MaxAcc_l at layer 0 was nearly identical to that at the final vision encoder layer, suggesting that linearly decodable signals are largely preserved through the adapter. MaxAcc_l then changed only slightly across language model layers for all as-

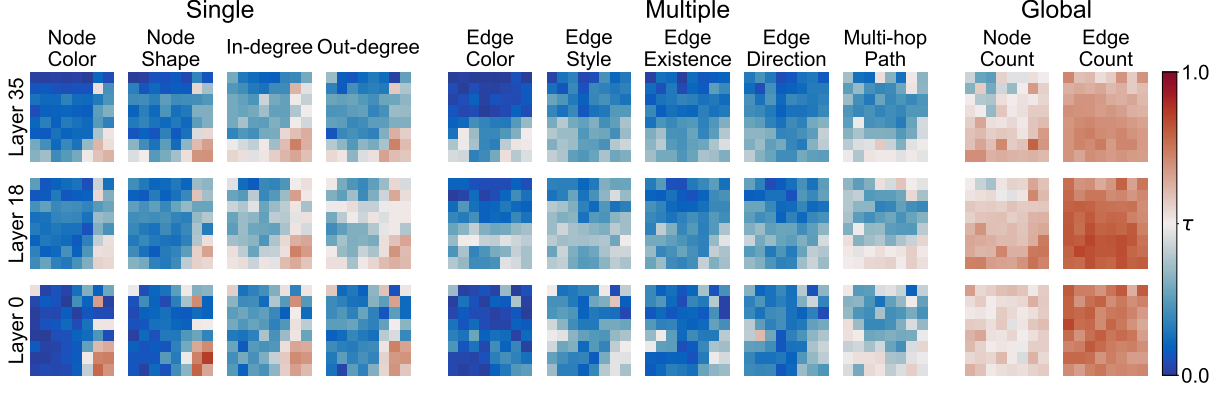


Figure 6: $acc_{l,t}(d_{fix})$ in the language model of Qwen3-VL 8B for a representative subset d_{fix} . The node layout of the evaluation diagrams is the same as that of the diagrams in the right part of Figure 2. The spatial distribution of high-accuracy patches remained largely unchanged across language model layers.

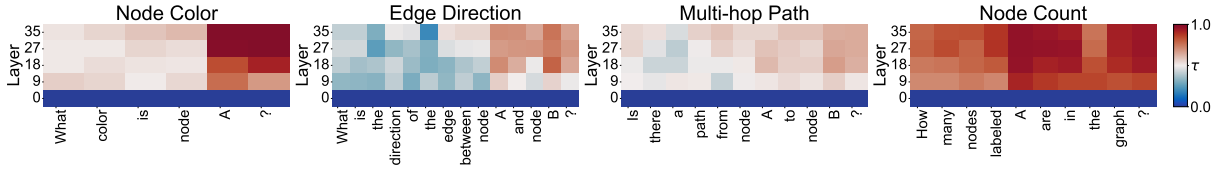


Figure 7: Probing results for text tokens in Qwen3-VL 8B. The x- and y-axes indicate token position in the question and model layer, respectively. Accuracy increased sharply at tokens specifying the target node or edge.

pects, indicating that these signals remain linearly decodable throughout image-input processing. The position-wise analysis (Figure 6) also showed little change across layers in the number or spatial distribution of high-accuracy positions.

Aspect-specific signals become linearly decodable through text conditioning. Figure 7 shows $acc_{l,t}$ across layers and token positions in the text-input part of the language model. Results for aspects other than Node Color, Edge Direction, Multi-hop Path, and Node Count are provided in Appendix D.3. For most aspects, accuracy increased sharply at tokens specifying the target nodes or edges. This suggests that the language model selectively aggregates information from image-position hidden states into text-position hidden states according to the target specified by the text. In contrast, aspects in the Global category showed relatively high accuracy from the first text token, suggesting that signals for these aspects are already linearly decodable before explicit target specification. For Multi-hop Path, although accuracy was relatively low compared with other aspects, it slightly exceeded the threshold across a wide range of layers in the text-input part, suggesting that signals for this aspect are linearly decodable, albeit weakly.

In summary, these results show that signals related to node and edge information are linearly decodable from hidden states in either the vision encoder or the language model. The layers and positions at which these signals are most linearly decodable vary across aspects. In particular, for Edge Direction and Multi-hop Path, the corresponding signals remain linearly decodable despite low VQA accuracy, revealing a gap between internal signals and model outputs.

6 Causal Intervention

In § 5, we found that signals for various diagram aspects are linearly decodable from the hidden states of the vision encoder and the language model. However, it remains unclear whether these signals causally contribute to the model’s predictions. To test this, we selectively corrupt hidden states with high probing accuracy in both the vision encoder and the language model. We then measure the resulting change in VQA performance.

6.1 Method

As shown in Figure 8, our procedure is to: (1) identify the target layers and positions, (2) replace the selected hidden states with an uninformative hidden state averaged over the evaluation dataset,

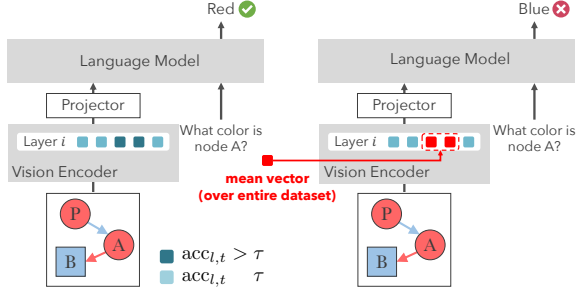


Figure 8: Overview of the causal intervention. Hidden states at patch positions whose probing accuracy exceeds the threshold are replaced with the uninformative hidden state computed over all images and positions in the evaluation dataset.

and (3) compare VQA accuracy (§ 4.1) between interventions on high-probing-accuracy positions and random positions.

Target layers. We apply interventions at early, middle, and late layers of both the vision encoder (layers 1, 14, and 27) and the language model (layers 0, 18, and 35)².

Target positions. We intervene at positions where the probing accuracy exceeds the threshold τ . We replace all such target positions simultaneously in a single intervention. For dataset $d_{\text{fix}} \in \mathcal{D}_{\text{test}}$ and layer l , the set of target positions for intervention is defined as:

$$\mathcal{S}_{d_{\text{fix}}}^l = \{t \in \{1, \dots, T\} \mid \text{acc}_{l,t}(d_{\text{fix}}) > \tau\}, \quad (6)$$

where T is the total number of positions.

Intervention operation. Following prior work (Neo et al., 2025), we use the dataset-mean hidden state as an uninformative reference. For each $d_{\text{fix}} \in \mathcal{D}_{\text{test}}$, we compute $\mu_{l,d_{\text{fix}}}$ at layer l as:

$$\mu_{l,d_{\text{fix}}} = \text{mean}_{x_i \in d_{\text{fix}}, t \in \{1, \dots, T\}} h_{i,l,t}. \quad (7)$$

By averaging over all images and positions, the mean hidden state retains the general activation statistics of the layer but removes information specific to any particular image or position. For each input x_i , we replace the hidden states at intervention target positions $t \in \mathcal{S}_{d_{\text{fix}}}^l$ with this vector:

$$\tilde{h}_{i,l,t} = \begin{cases} \mu_{l,d_{\text{fix}}} & (t \in \mathcal{S}_{d_{\text{fix}}}^l), \\ h_{i,l,t} & (t \notin \mathcal{S}_{d_{\text{fix}}}^l). \end{cases} \quad (8)$$

We compute Acc_{VQA} under this intervention setting. We refer to this setting as *accuracy-based*.

²We exclude vision encoder layer 0, which contains only patch embeddings, and language model layer 36, whose hidden states, except at the final position, do not affect the output.

Control experiment. To verify that the intervention effect truly comes from targeting high-probing-accuracy positions, we conduct a control experiment that intervenes at randomly selected positions instead. Specifically, we sample a set $\mathcal{R}_{d_{\text{fix}}}^l \subset \{1, \dots, T\}$ with $|\mathcal{R}_{d_{\text{fix}}}^l| = |\mathcal{S}_{d_{\text{fix}}}^l|$ and perform replacement as:

$$\tilde{h}_{i,l,t} = \begin{cases} \mu_{l,d_{\text{fix}}} & (t \in \mathcal{R}_{d_{\text{fix}}}^l), \\ h_{i,l,t} & (t \notin \mathcal{R}_{d_{\text{fix}}}^l). \end{cases} \quad (9)$$

We compute Acc_{VQA} under this intervention setting. We refer to this setting as *controlled*.

Evaluation criteria. We compute the accuracy difference as *accuracy-based* – *controlled*. Because both settings use the same mean-vector replacement procedure, taking the difference controls for generic effects of replacement, allowing us to assess the contribution of signals at the targeted positions. Therefore, a negative difference indicates that these signals causally contribute to the model’s predictions.

6.2 Results

Table 3 reports the accuracy difference between the *accuracy-based* and *controlled* settings at each layer of Qwen3-VL 8B. For the Single category and the Edge Color, Edge Style, and Node Count aspects, we observed a relatively large drop in accuracy compared with the *controlled* setting. This indicates that, for these aspects, the hidden states of both the vision encoder and the language model contain representations that contribute to the model’s predictions. For Edge Direction and Multi-hop Path, the model’s VQA accuracy was already close to chance level (§ 4.2), and thus, little change is expected regardless of the intervention. Exceptionally, for Edge Count, probing accuracy exceeded τ at all positions in most layers, so all patches are replaced with $\mu_{l,d_{\text{fix}}}$ in both settings, making the two conditions equivalent (± 0.0); thus, we cannot determine its causal contribution under our intervention design.

7 Generalization to Flowcharts

To examine whether our findings generalize beyond synthetic graphs, we conduct VQA, probing, and causal intervention experiments with Qwen3-VL 8B on flowcharts. We construct a dataset for each aspect in § 3.2 by editing and extending Mermaid code sampled from FlowVQA (Singh et al.,

	Layer	Node Color	Node Shape	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Vision encoder	1	-6.0	± 0.0	± 0.0	± 0.0	± 0.0	-1.1	± 0.0	± 0.0	± 0.0	-5.5	-6.0
	14	-77.4	-46.7	-24.0	-3.6	-4.8	-0.8	-0.8	+0.2	-0.8	-24.4	± 0.0
	27	-78.2	-55.7	-17.5	-11.1	-32.9	-8.3	+0.1	-0.6	+0.6	-5.8	± 0.0
Language model	0	-74.5	-52.0	-19.7	-8.5	-28.8	-9.1	-2.5	-0.5	-0.3	-3.7	± 0.0
	18	-16.5	-10.4	-3.9	+0.1	-30.7	-4.5	-0.5	+1.8	-1.8	-0.8	± 0.0
	35	-0.1	+0.0	± 0.0	+0.1	+0.0	± 0.0	± 0.0	-0.2	+0.1	+0.1	± 0.0

Table 3: Accuracy difference between the *accuracy-based* and *controlled* settings (*accuracy-based* – *controlled*). More negative values (**bold**) indicate a larger accuracy decrease under the *accuracy-based* intervention than under the *controlled* intervention. For Edge Count, probing accuracy exceeded τ at all positions in most layers, making the *accuracy-based* and *controlled* settings equivalent (± 0.0), precluding causal assessment for this aspect.

2024). We exclude Node Shape because varying node shapes changes the image size, preventing consistent patch-wise evaluation across images. Unlike the synthetic graphs, these flowcharts contain characteristics commonly found in real-world flowcharts, such as textual labels and conventional layouts. We follow the experimental setup used for the synthetic graphs, with modifications where necessary for the flowchart setting. Detailed setup and results are provided in Appendix F.

VQA trends are largely consistent with those observed on synthetic graphs. Table 4 reports VQA accuracy for all aspects. Accuracy on Out-degree, Edge Existence, Edge Direction, and Multi-hop Path remained near chance level, while accuracy on the other six aspects was clearly above chance level. This is largely consistent with the results on synthetic graphs, except that Out-degree and Edge Existence exceeded chance level there.

Signals for all aspects are linearly decodable within the model. Figures 9 and 10 show MaxAcc_l per layer in the vision encoder and the language model, respectively. Signals for all aspects, including aspects in the Multiple category that the model struggles with in VQA, were linearly decodable from these hidden states. Consistent with the results on synthetic graphs, this reveals a similar gap between internal signals and model outputs on flowcharts.

Signals identified through probing causally contribute to the model’s predictions on flowcharts. Table 5 reports the difference in VQA accuracy between the *accuracy-based* and *controlled* settings. For the six aspects with above-chance VQA accuracy, intervening on patches with high probing accuracy decreased VQA accuracy for at least

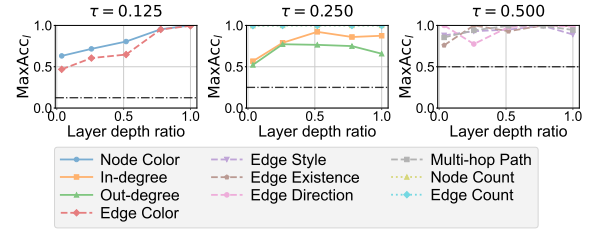


Figure 9: MaxAcc_l per layer in the vision encoder of Qwen3-VL 8B on flowcharts. Axes and the threshold line follow Figure 3. All aspects exceeded the threshold, including Multiple-category aspects.

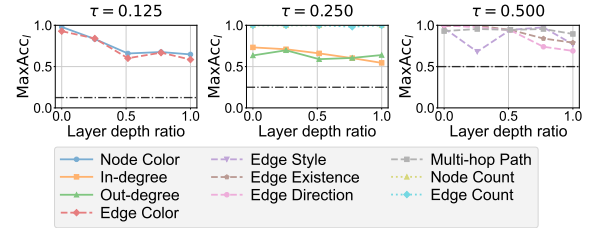


Figure 10: MaxAcc_l per layer in the image-input part of the language model in Qwen3-VL 8B on flowcharts. Axes and the threshold line follow Figure 3. All aspects maintained accuracy above the threshold.

one layer. This indicates that the signals identified through probing causally contribute to the model’s predictions on flowcharts, showing that our finding is not limited to synthetic graphs. For the remaining aspects, VQA accuracy was already close to chance level, leaving little room for further decreases.

8 Related Work

Diagram understanding. Prior work on diagram understanding has addressed tasks such as VQA (Lu et al., 2024; Wang et al., 2024) and image captioning (Hsu et al., 2021; Huang et al., 2023). While classical approaches relied on rule-based methods such as pattern matching and predefined schemas (Blostein, 1996; Song et al., 2017),

	Node Color	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Chance level	12.5	25.0	25.0	12.5	50.0	50.0	50.0	50.0	25.0	25.0
Qwen3-VL 8B	95.3	46.8	25.0	78.6	98.5	53.0	50.0	50.0	45.8	45.5

Table 4: Acc_{VQA} for Qwen3-VL 8B on flowcharts.

	Layer	Node Color	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Vision encoder	1	+0.6	± 0.0	± 0.0	-7.4	-23.5	-5.0	-0.5	± 0.0	-3.0	-19.2
	14	-76.8	-0.5	± 0.0	-69.5	+5.0	-4.5	± 0.0	± 0.0	-32.5	+1.3
	27	-75.4	-0.8	± 0.0	-58.6	-1.0	± 0.0	± 0.0	-1.5	-23.3	-1.5
Language model	0	-44.0	-17.0	± 0.0	+2.4	-47.5	-17.5	± 0.0	± 0.0	-15.5	-2.5
	18	-29.8	-14.8	± 0.0	-26.9	-37.0	+11.5	± 0.0	+0.5	-7.3	-2.5
	35	-0.1	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	+0.3	+0.3

Table 5: Causal intervention results on flowcharts, where more negative values indicate a larger decrease in VQA accuracy relative to the *controlled* setting.

LVLm-based approaches have recently become prevalent. Understanding diagrams requires recognizing visual elements such as symbols and shapes, as well as capturing their relationships. Studies (Pan et al., 2024; Zhu et al., 2025; Hou et al., 2025) show that LVLms adequately recognize individual elements but struggle to capture their relationships. However, the underlying cause of this limitation remains underexplored. Our work investigates this limitation from the perspective of internal representations by analyzing how LVLms represent diagram elements and their relationships.

Internal analysis of LVLms. Prior work has examined the internal representations of LVLms to understand how they process visual information. Gandelsman et al. (2024) analyzed how natural images are represented within CLIP (Radford et al., 2021), whose vision encoder is widely used in LVLms. They found that multiple attention heads capture specific properties such as color, shape, and number, and that visual information is localized to patches corresponding to specific image regions. Tao et al. (2024) examined how global and local information from natural images is represented in the language model of LVLms. They found that middle layers encode global information, while later layers encode local information. In contrast, we analyze internal representations for diagrams, where information is conveyed symbolically through text and shapes and structurally through relationships such as edge connectivity.

Gap between internal signals and outputs. Prior work has shown a gap between information

detectable within LVLms and their outputs. Projecting internal representations of LVLms into the language model’s vocabulary space reveals that objects present in the image receive higher confidence than hallucinated ones, even when the model’s outputs contain hallucinations (Jiang et al., 2025). Similarly, another study found that linear probes on hidden states substantially outperform generation-based predictions of object presence, with this gap primarily originating from the language model’s output head (Li et al., 2026). Both studies focus on object presence in natural images. We show that a similar gap arises for relational aspects of diagrams, such as edge direction and the existence of multi-hop paths.

9 Conclusion

We constructed a synthetic diagram dataset and applied probing and causal interventions to investigate how LVLms represent visual information in diagrams. Our probing analysis shows that signals about nodes and edges are linearly decodable from the hidden states of either the vision encoder or the language model. Notably, these signals remain linearly decodable even for aspects that the model struggles with in VQA, revealing a gap between internal signals and model outputs. Causal interventions further show that, for several aspects, the signals identified through probing causally contribute to the model’s predictions. These findings also hold on flowcharts, indicating that they are not specific to our synthetic diagrams. These findings provide insight into how LVLms represent and use diagram information.

Limitations

Coverage of real-world diagrams. We use synthetic diagrams to minimize the influence of dataset biases and shortcut learning that can arise from natural-language and natural-image data. Although using a synthetic dataset enables well-controlled and fine-grained analysis, our setup does not necessarily capture all complex situations found in real-world diagrams. To partly address this, we additionally evaluate on flowcharts derived from FlowVQA (Singh et al., 2024) (§ 7), which contain features characteristic of real-world flowcharts, such as textual labels and conventional layouts. This evaluation is nevertheless limited in two respects. First, the flowcharts are still produced by controlled editing and augmentation of node and edge attributes, and are therefore not fully natural. Second, flowcharts are only one of many diagram formats, and diagrams in other domains may rely on different visual conventions. We therefore do not claim that our findings extend to real-world diagrams in general, and evaluating a wider range of diagram formats remains future work.

In addition, we focus on basic visual information, such as node color and shape, and edge direction. Although complex information may be represented by combining these cues, how the model represents composite visual patterns and higher-order structures remains unexamined.

Probing methods. Probing has been widely used to analyze internal representations in language models (Conneau et al., 2018; Tenney et al., 2019; Campbell et al., 2023; Li et al., 2023), but its validity as an interpretability method has also been questioned (Liu et al., 2023; Burns et al., 2023). Our analysis relies on linear probes and therefore measures only linear decodability. A lack of linear decodability does not imply that the corresponding signal is absent; it may be encoded nonlinearly or distributed across multiple positions. Future work could examine these possibilities using a broader range of probing methods (Gurnee et al., 2023; Bricken et al., 2023).

Ethics Statement

We do not anticipate any significant ethical concerns because (i) we conducted no human-subject experiments, and (ii) our tasks involve only simple, synthetically generated diagrams and do not address ethically sensitive topics.

Acknowledgments

This work was supported by AMED Grant Number JP26wm0625405, JST PRESTO Grant Number JPMJFR242S, JST SPRING Grant Number JPMJSP2114, JST BOOST Grant Numbers JPMJBY24D2 and JPMJBS2421, and JSPS KAKENHI Grant Numbers JP24K20829, JP25K03175, and JP25KJ0615, and the Nakajima Foundation. We thank the members of the Tohoku NLP Group for their advice and support.

References

- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery.
- Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes.
- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhi-fang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-VL technical report.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-VL technical report.
- B Berlin and P Kay. 1999. *Basic Color Terms: Their Universality and Evolution*. The David Hume Series. Cambridge University Press, Cambridge, England.
- Dorothea Blostein. 1996. General diagram-recognition methodologies. In *Graphics Recognition Methods and Applications*, pages 106–122, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh*

International Conference on Learning Representations.

- James Campbell, Phillip Guo, and Richard Ren. 2023. Localizing lying in llama: Understanding instructed dishonesty on true-false questions through prompting, probing, and patching. In *Socially Responsible Language Modelling Research*.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $\$&!#^*$ vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.
- Corentin Dancette, Rémi Cadène, Damien Teney, and Matthieu Cord. 2021. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1554–1563.
- Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2024. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. 2024. Interpreting CLIP’s image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*.
- Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Jennifer Healey, Laurie Byrum, Md Nadeem Akhtar, and Moumita Sinha. 2024. Evaluating nuanced bias in large language model free response answers. In *Natural Language Processing and Information Systems*, pages 378–391, Cham. Springer Nature Switzerland.
- Yifan Hou, Buse Giledereli, Yilei Tu, and Mrinmaya Sachan. 2025. Do vision-language models really understand visual language? In *Forty-second International Conference on Machine Learning*.
- 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.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Chieh-Yang Huang, Ting-Yao Hsu, Ryan Rossi, Ani Nenkova, Sungchul Kim, Gromit Yeuk-Yin Chan, Eunyee Koh, Clyde Lee Giles, and Ting-Hao ‘ken-neth’ Huang. 2023. Summaries as captions: Generating figure captions for scientific documents with automated text summarization. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 80–92, Prague, Czechia. Association for Computational Linguistics.
- Nicholas Jiang, Anish Kachinthaya, Suzanne Petryk, and Yossi Gandelsman. 2025. Interpreting and editing vision-language representations to mitigate hallucinations. In *The Thirteenth International Conference on Learning Representations*.
- Gregor Jošt, Jernej Huber, Marjan Heričko, and Gregor Polančič. 2017. Improving cognitive effectiveness of business process diagrams with opacity-driven graphical highlights. *Decision Support Systems*, 103:58–69.
- Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1983–1991.
- Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision – ECCV 2016*, pages 235–251, Cham. Springer International Publishing.
- Sebastian Kernbach and Martin J Eppler. 2010. The use of visualization in the context of business strategies: An experimental evaluation. In *2010 14th International Conference Information Visualisation*, pages 349–354.
- Kurt W Kohn and Mirit I Aladjem. 2006. Circuit diagrams for biological networks. *Molecular Systems Biology*, 2(1):MSB4100044.
- Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*.

- Xue Li, Yiyun Sun, Wei Cheng, Yinglun Zhu, and Haifeng Chen. 2025. Chain-of-region: Visual language models need details for diagram analysis. In *The Thirteenth International Conference on Learning Representations*.
- Yifan Li, Kun Zhou, Xin Zhao, Lei Fang, and Jirong Wen. 2026. Analyzing and mitigating object hallucination: A training bias perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(8):6636–6643.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26286–26296.
- Kevin Liu, Stephen Casper, Dylan Hadfield-Menell, and Jacob Andreas. 2023. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4797, Singapore. Association for Computational Linguistics.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*.
- Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. 2025. Towards interpreting visual information processing in vision-language models. In *The Thirteenth International Conference on Learning Representations*.
- OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card.
- Huitong Pan, Qi Zhang, Cornelia Caragea, Edward C Dragut, and Longin Jan Latecki. 2024. FlowLearn: Evaluating large vision-language models on flowchart understanding. In *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 73–80. IOS Press.
- 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*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Camilo Chacón Sartori, Christian Blum, and Filippo Bistaffa. 2025. VisGraphVar: A benchmark generator for assessing variability in graph analysis using large vision-language models. *IEEE Access*, 13:21788–21810.
- Samuel Silva, Beatriz Sousa Santos, and Joaquim Madeira. 2011. Using color in visualization: A survey. *Computers & Graphics*, 35(2):320–333.
- Shubhankar Singh, Purvi Chaurasia, Yerram Varun, Pranshu Pandya, Vatsal Gupta, Vivek Gupta, and Dan Roth. 2024. FlowVQA: Mapping multimodal logic in visual question answering with flowcharts. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1330–1350, Bangkok, Thailand. Association for Computational Linguistics.
- Dan Song, Dongming Wang, and Xiaoyu Chen. 2017. Retrieving geometric information from images: the case of hand-drawn diagrams. *Data Mining and Knowledge Discovery*, 31(4):934–971.
- Anna Sterzik, Nils Lichtenberg, Jana Wilms, Michael Krone, Douglas W Cunningham, and Kai Lawonn. 2024. Perception of line attributes for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):1041–1051.
- Mingxu Tao, Quzhe Huang, Kun Xu, Liwei Chen, Yansong Feng, and Dongyan Zhao. 2024. Probing multimodal large language models for global and local semantic representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13050–13056, Torino, Italia. ELRA and ICCL.
- Gemma Team. 2025. Gemma 3 technical report.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.
- Chin Tseng, Arran Zeyu Wang, Ghulam Jilani Quadri, and Danielle Albers Szafrir. 2025. Shape it up: An empirically grounded approach for designing shape palettes. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):349–359.
- Shaowei Wang, Lingling Zhang, Longji Zhu, Tao Qin, Kim-Hui Yap, Xinyu Zhang, and Jun Liu. 2024. CoG-DQA: Chain-of-guiding learning with large language models for diagram question answering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13969–13979.
- Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.
- Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng

Li. 2025. MATHVERSE: Does your multi-modal LLM truly see the diagrams in visual math problems? In *Computer Vision – ECCV 2024*, pages 169–186, Cham. Springer Nature Switzerland.

Yingjie Zhu, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. 2025. Benchmarking and improving large vision-language models for fundamental visual graph understanding and reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30678–30701, Vienna, Austria. Association for Computational Linguistics.

A Details of VQA Experiment

Evaluation setup. Table 6 shows the questions used to evaluate each aspect in our experiments.

Additional discussion. Table 2 shows that accuracy varies substantially within the Multiple category: Edge Color, Edge Style, and Edge Existence are answered with high accuracy, whereas Edge Direction and Multi-hop Path remain near chance. This trend suggests a gradation in the degree and type of integration required. Edge Color and Edge Style can be determined by inspecting any local part of the edge, and Edge Existence can often be determined by detecting whether a visible connection between the queried nodes is present. In contrast, Edge Direction requires identifying the two queried endpoint nodes and then locating and interpreting the arrowhead between them. Multi-hop Path further requires composing directional relations across multiple edges. These latter aspects impose stronger demands on compositional integration, which may contribute to the observed accuracy gap.

B Dataset

B.1 Detailed Dataset Specification

We select node and edge colors from the 11 Basic Color Terms of Berlin and Kay (1999), excluding white, black, and gray, which leaves eight colors. For node shapes, we use simple geometric figures without strong semantic associations, including circles and regular polygons. We exclude equilateral triangles because they resemble arrowheads and may therefore blur the distinction between nodes and edges. For Multi-hop Path, we exclude direct edges between nodes A and B and place exactly one intermediate node between them, so a positive label always corresponds to a two-edge path.

C Preliminary Experiment: VQA

Table 7 reports the VQA accuracy of all models evaluated in this study. Across all models, Node Color achieved relatively high accuracy, while In-degree and Out-degree were consistently lower than the other aspects in the Single category, approaching chance level for LLaVA1.5 7B. Within the Multiple category, Edge Color and Edge Existence were answered above chance by most models, whereas Edge Direction stayed near chance for all models and Multi-hop Path for most models. These trends are consistent with

the results for Qwen3-VL reported in § 4.2 and suggest that recognizing edge directions and multi-step relational structure is difficult across models.

D Probing

D.1 Shortcut Learning of Probes

In the probing experiments, we observed shortcuts across all aspects where the probe predicted answers based solely on superficial features of the input patches. These shortcuts resulted in 100% accuracy on patches in certain regions. For example, for Node Color, the probe achieved 100% accuracy on patches containing colored nodes by predicting the local patch color without determining whether the patch corresponds to node A. Beyond Node Color, we also confirmed shortcuts that exploited information such as patch shape and position.

These shortcuts do not solve the intended probing task. To correctly classify Node Color, the probe must first identify node A and then predict its color. However, shortcut-based predictions skip the identification of node A. These shortcuts arise because superficial patch features are correlated with the target labels under the training distribution.

To address this problem, as described in § 3.3, we add diagrams without node A as dummy data to the training dataset. This approach ensures that when the input does not contain the information necessary for prediction, the probe outputs the dummy class (N/A), thereby avoiding falsely high accuracy through shortcuts.

D.2 Probe Training Hyperparameters

We used linear classifiers for probe training, training separately for each aspect. The hyperparameters differed only in batch size between image-part probing and text-part probing. Table 8 shows the hyperparameters used for each type of probing.

For hyperparameter search, we used Optuna (Akiba et al., 2019) with TPESampler to perform three hyperparameter search trials. The search targets were batch size and learning rate, where batch size was selected categorically from the candidates shown in Table 8, and learning rate was searched on a log scale ranging from 0.0001 to 0.01. For the evaluation metric, we split 10% of the training data as validation data and used accuracy on the validation data. We did not use early stopping during training; instead, we evaluated performance on the validation data at each epoch and saved the model

Aspect	Question
Node Color	What color is node A?
Node Shape	What shape is node A?
In-degree	What is the in-degree of node A?
Out-degree	What is the out-degree of node A?
Edge Color	What color is the edge between node A and node B?
Edge Style	What style is the edge between node A and node B?
Edge Existence	Does an edge exist between node A and node B?
Edge Direction	What is the direction of the edge between node A and node B?
Multi-hop Path	Is there a path from node A to node B?
Node Count	How many nodes labeled A are in the graph?
Edge Count	How many edges are in the graph?

Table 6: Questions used to evaluate each aspect.

	Node Color	Node Shape	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Chance level	12.5	20.0	20.0	20.0	12.5	50.0	50.0	50.0	50.0	20.0	20.0
Qwen3-VL 8B	95.5	78.4	53.7	40.6	80.6	85.7	81.2	53.9	55.9	78.7	83.8
Qwen2.5-VL 7B	86.6	72.6	40.4	33.8	61.0	73.3	65.5	49.7	53.3	93.8	38.7
LLaVA1.5 7B	40.1	31.3	22.1	24.9	33.2	47.3	54.4	50.0	50.0	20.0	26.1
Gemma3-4B-IT	68.7	51.8	33.9	35.9	53.2	67.0	61.8	49.5	50.0	30.9	20.1

Table 7: VQA accuracy of all models evaluated in this study. Across all models, Edge Direction remained near chance and Multi-hop Path was difficult for most models, consistent with the pattern observed for Qwen3-VL in Table 2.

Parameter	Image Part	Text Part
Epochs	1000	1000
Batch size *	{1000, 2000, 4000, 8000, 16000}	{10, 20, 40, 80, 160}
Learning rate *	[0.0001, 0.01] (log scale)	[0.0001, 0.01] (log scale)
Optimizer	AdamW	AdamW
betas	[0.9, 0.999]	[0.9, 0.999]
eps	1e-8	1e-8
weight decay	0.01	0.01

Table 8: Hyperparameters used for probe training. Batch size and learning rate were subject to hyperparameter search using Optuna. Optimizer parameters use PyTorch’s AdamW default values. *Subject to hyperparameter search

that achieved the highest accuracy.

D.3 Additional Results

In this section, we present additional probing analyses beyond the Qwen3-VL 8B results reported in the main text. These include detailed results for Qwen3-VL 8B, robustness checks with different node layouts, larger visual elements, and 7-node graphs, as well as results for additional models (Qwen2.5-VL 7B, LLaVA1.5 7B, Gemma3-4B-IT).

Qwen3-VL 8B (detailed probing results). Figures 11 and 12 present image probing results with two additional intermediate layers that were omitted in Figures 4 and 6. As described in § 5.3, Single category information is encoded near node

positions, while Global category information is distributed across background regions. Figure 13 presents text probing results for aspects omitted from Figure 7. As described in § 5.3, accuracy sharply increased at token positions that specify the target nodes or edges.

Qwen3-VL 8B (different node layouts). Figures 15 and 16 show results with different node layouts (see Figure 14). Whereas the evaluation diagrams in the main text place the target node (node A) at the bottom right, here node A is placed on the left side. The distribution of patches with high accuracy for node information changed according to the position of node A. This positional shift indicates that the decoded signals track the location of the target node rather than a fixed image region.

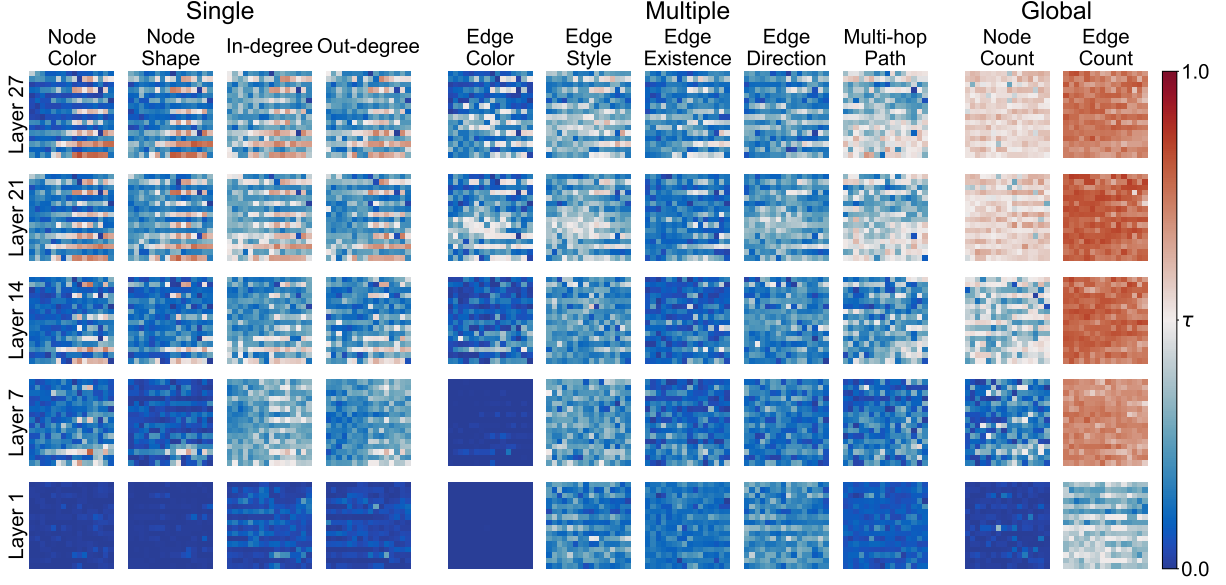


Figure 11: Accuracy for each hidden state in the vision encoder of Qwen3-VL 8B. Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were consistent with Figure 4.

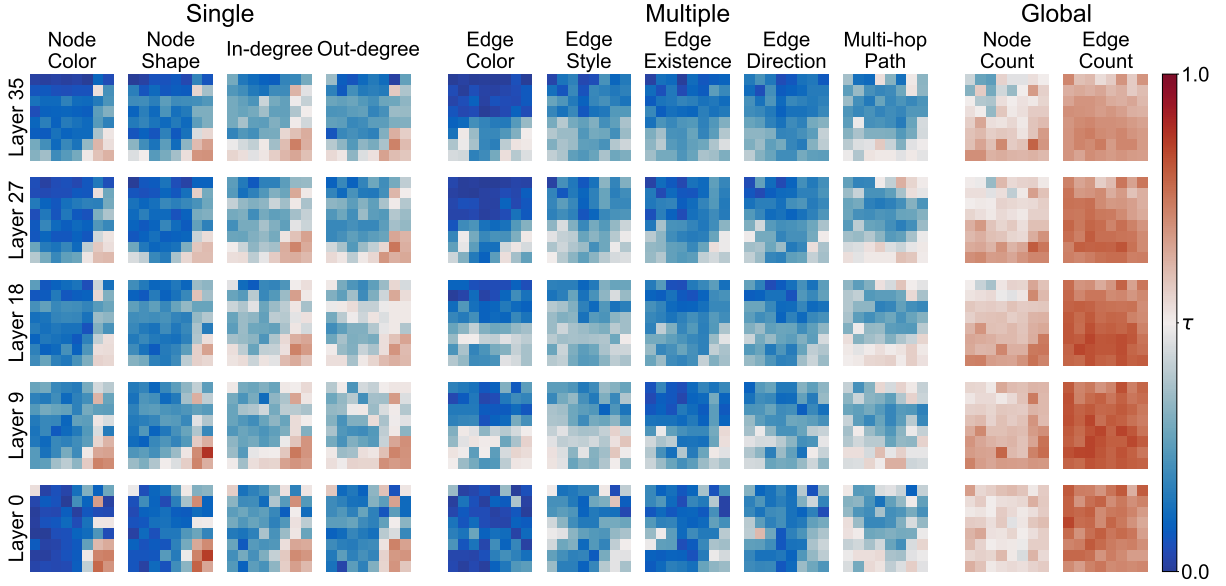


Figure 12: Accuracy for each hidden state in the image-input part of the Qwen3-VL 8B language model. Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were consistent with Figure 6.

Qwen3-VL 8B (larger nodes and edges). Figures 18 and 19 show probing results when the sizes of nodes and edges are increased (see Figure 17 for example diagrams). The results exhibit the same trends as the default-size setting reported in § 5.3, indicating that the observed patterns are robust to this change in visual-element size.

Qwen3-VL 8B (7-node graphs). We report probing results on diagrams containing 7 nodes instead of 5 (see Figure 20 for example diagrams). Fig-

ures 21 and 22 show the layer-wise results, Figures 23 and 24 show the patch-wise results, and Figure 25 shows the text-token results. The overall accuracy is slightly lower and fewer patches exceed the threshold than in the 5-node setting, possibly reflecting the increased task difficulty from the larger number of nodes. Nonetheless, the results exhibit the same trends as the default 5-node setting reported in § 5.3, indicating that the observed encoding patterns are robust to moderate increases

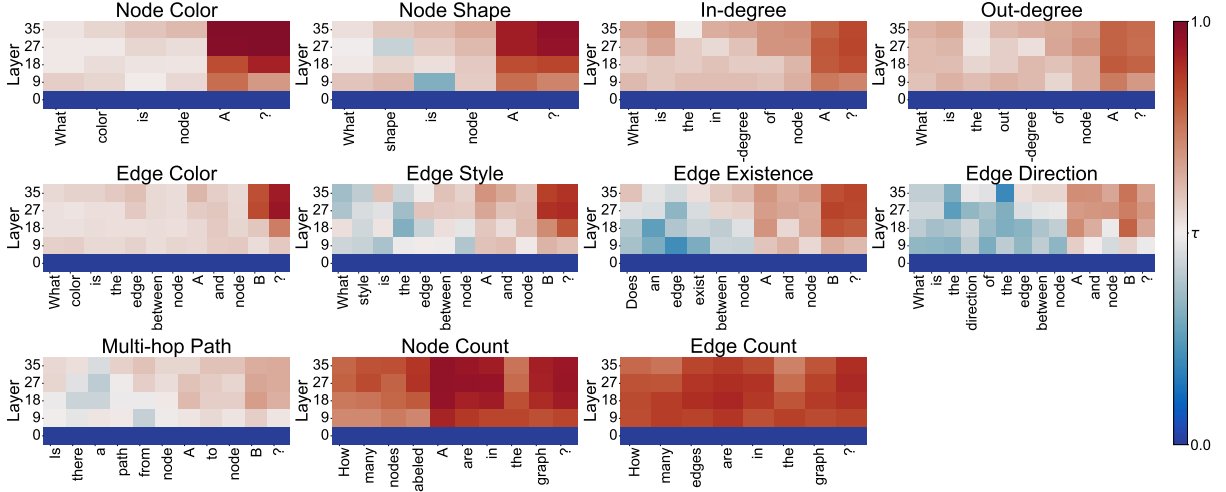


Figure 13: Probing results for text tokens in Qwen3-VL 8B. The x-axis shows token position in the question, and the y-axis shows layer position. Trends were consistent with Figure 7.

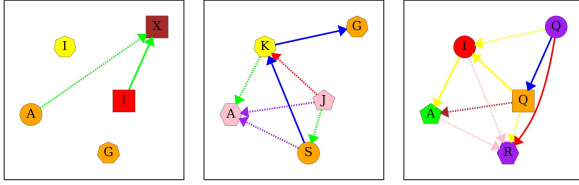


Figure 14: Example diagrams with layouts distinct from those in Figure 2.

in graph size.

Qwen2.5-VL 7B. Figures 26 and 27 show the layer-wise probing results, and Figures 28 and 29 show the patch-wise results for Qwen2.5-VL 7B. The overall trends are similar to Qwen3-VL, with node information showing high accuracy while edge relationships remained near chance level. In the patch-wise analysis, patches corresponding to node positions showed higher accuracy for aspects in the Single category, while aspects in the Global category showed high accuracy across a wide range of positions, including background regions.

Figure 30 shows the probing results for text tokens. Similar to Qwen3-VL, we observed a trend where accuracy sharply increased at token positions that specify nodes or edges.

LLaVA1.5 7B. Figures 31 and 32 show the layer-wise probing results, and Figures 33 and 34 show the patch-wise results for LLaVA1.5 7B. The overall trends are similar to Qwen3-VL and Qwen2.5-VL, but a distinctive difference was observed in the patch-wise analysis. In LLaVA1.5, background patches showed higher accuracy than patches corresponding to node positions. The background of our

diagrams is uniformly white and carries almost no information of its own. The corresponding hidden states may therefore be available to store information from other positions, as discussed for aspects in the Global category in § 5.3. In LLaVA1.5 this tendency extends to aspects in the Single category, whose signals are localized near node positions in the other models.

Figure 35 shows the probing results for text tokens. Similar to other models, we observed information aggregation at token positions that specify target nodes or edges.

Gemma3-4B-IT. Figures 36 and 37 show the layer-wise probing results, and Figures 38 and 39 show the patch-wise results for Gemma3-4B-IT. The overall trends are similar to Qwen3-VL and Qwen2.5-VL, but accuracy was lower for some node information and edge relationships compared to other models.

Figure 40 shows the probing results for text tokens. As with the other models, we observe an aggregation of information at the token positions that specify the target nodes or edges; however, the effect is weaker than in the other models.

E Additional Results for Causal Intervention

In § 6, we reported the causal intervention results as the accuracy difference between the *accuracy-based* and *controlled* settings for Qwen3-VL 8B. Here, we additionally report the absolute accuracies for the *accuracy-based*, *controlled*, and *clean* settings, where *clean* denotes the accuracy obtained

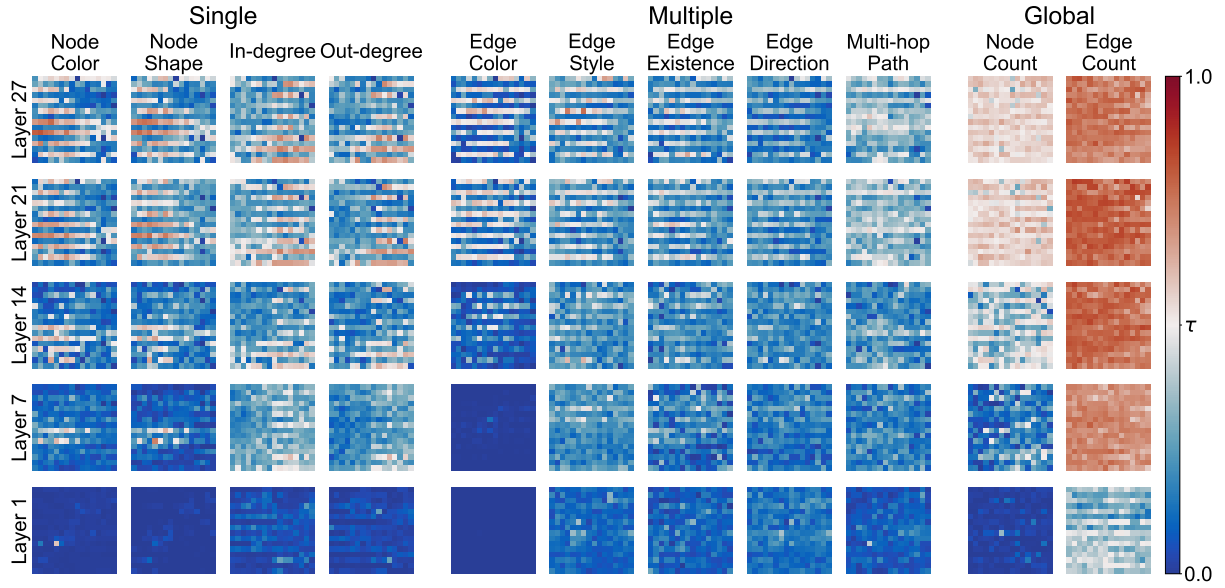


Figure 15: Accuracy for each hidden state in the vision encoder of Qwen3-VL 8B (node A placed on the left side). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. High-accuracy patches for Single-category aspects localized on the left side, following node A’s position.

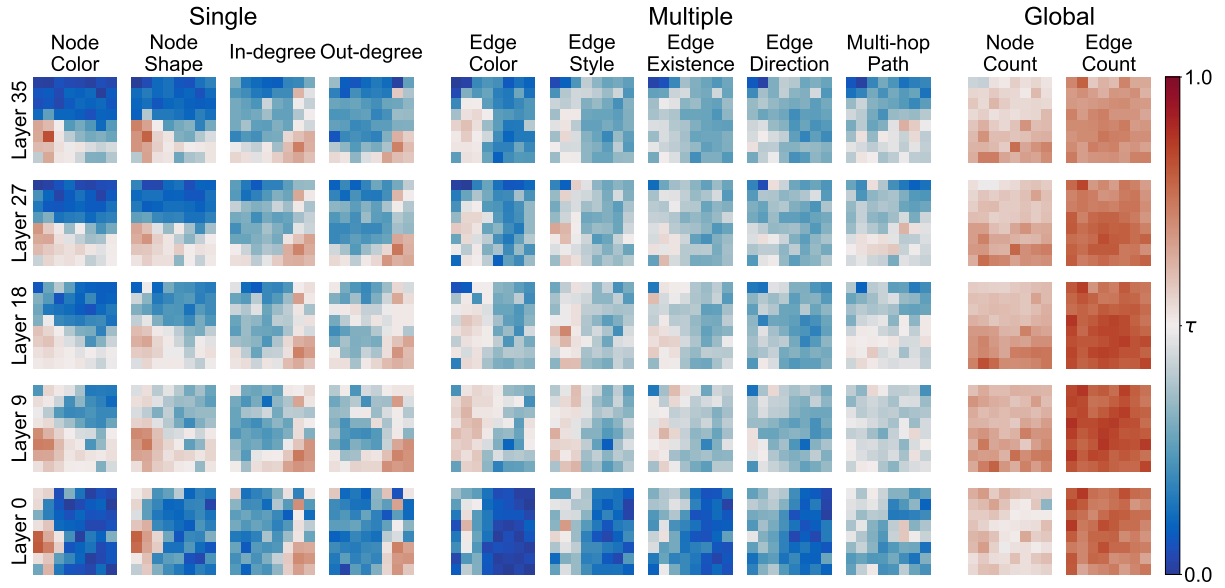


Figure 16: Accuracy for each hidden state in the image-input part of the Qwen3-VL 8B language model (node A placed on the left side). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. High-accuracy patches for Single-category aspects localized on the left side, following node A’s position.

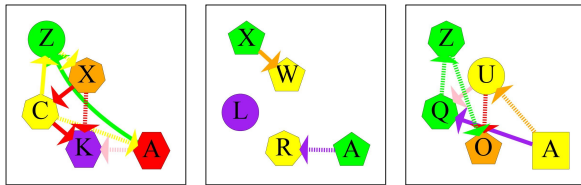


Figure 17: Example diagrams with larger nodes and edges compared to those in Figure 2.

without any intervention, and extend the analysis to Qwen2.5-VL 7B, LLaVA1.5 7B, and Gemma3-4B-IT.

Selective intervention for Edge Count. As described in § 6, for Edge Count the probing accuracy exceeds the threshold at all positions in most layers, so the *accuracy-based* and *controlled* settings coincide and the causal contribution cannot be assessed. We therefore apply a selective intervention that

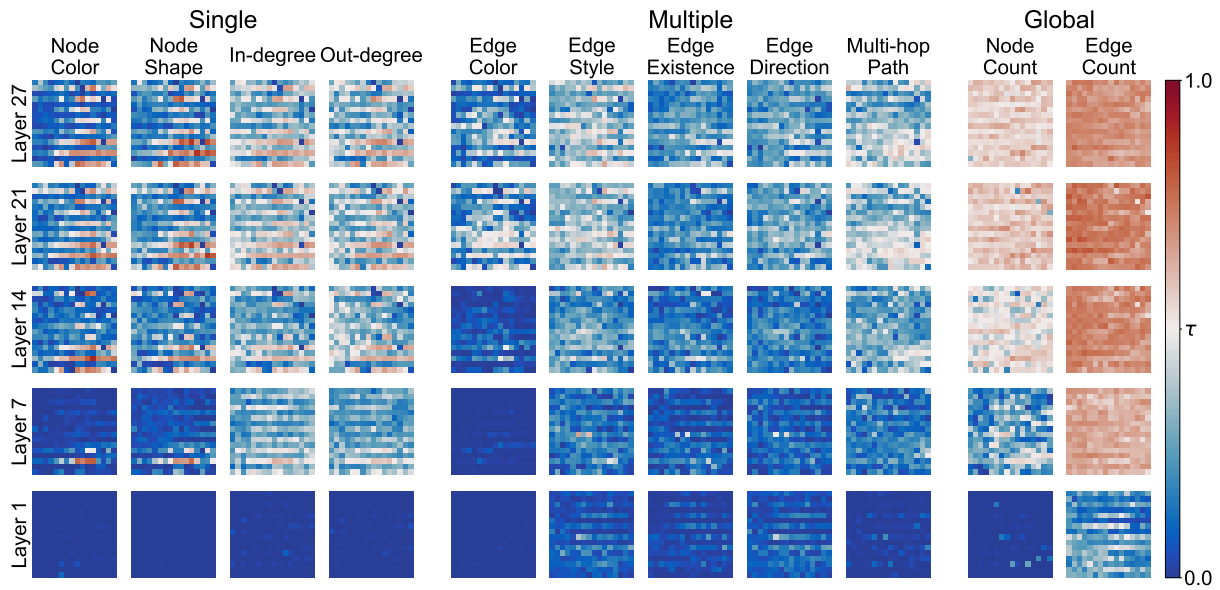


Figure 18: Accuracy for each hidden state in the vision encoder of Qwen3-VL 8B (larger nodes and edges). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were consistent with Figure 4.

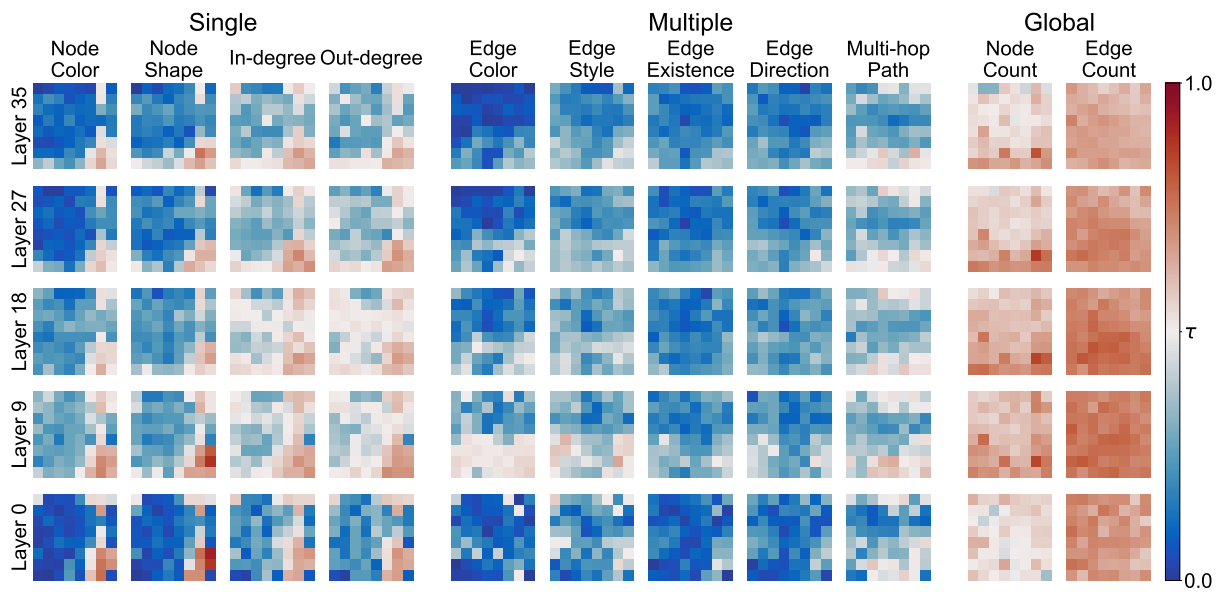


Figure 19: Accuracy for each hidden state in the image-input part of the Qwen3-VL 8B language model (larger nodes and edges). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were consistent with Figure 6.

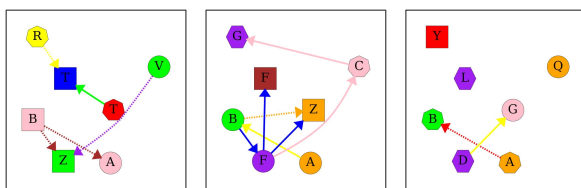


Figure 20: Example diagrams with 7 nodes compared to those in Figure 2.

targets only the top 50% of patches by probing accuracy. Table 9 reports the results. Intervening on patches with high probing accuracy lowered VQA accuracy by 14.9 to 21.7 points compared with intervening on the same number of randomly selected patches. The only exception is the final layer of the language model; this exception is consistent with the late-layer trend observed for the other aspects in the main text. These results confirm that, for

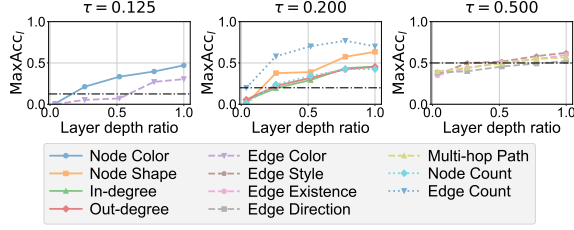


Figure 21: MaxAcc_l per layer in the vision encoder of Qwen3-VL 8B (7-node graphs). Axes and the threshold line follow Figure 3. Trends were similar to the 5-node setting (Figure 3), although the overall accuracy was slightly lower.

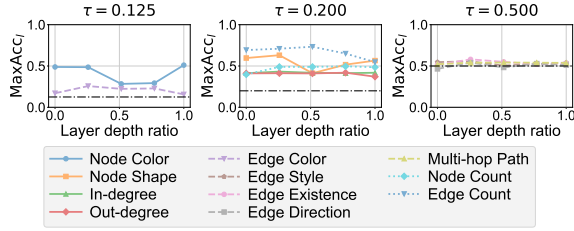


Figure 22: MaxAcc_l per layer in the language model of Qwen3-VL 8B (7-node graphs). Axes and the threshold line follow Figure 5. Trends were similar to the 5-node setting (Figure 5), although the overall accuracy was slightly lower.

Edge Count as well, the signals detected by probing causally contribute to the model’s predictions.

Qwen3-VL 8B. Figure 41 shows the *clean*, *accuracy-based*, and *controlled* accuracy at each intervention layer for Qwen3-VL 8B. The results are discussed in detail in § 6.

Qwen2.5-VL 7B. Table 10 and Figure 42 report the results for Qwen2.5-VL 7B. Similar to Qwen3-VL, the intervention caused a large drop in accuracy for aspects such as Node Color, Node Shape, Edge Color, and Node Count, indicating that the signals detected by probing in both the vision encoder and the language model causally contribute to the model’s predictions.

LLaVA1.5 7B. Table 11 and Figure 43 report the results for LLaVA1.5 7B. Although LLaVA1.5 has lower baseline VQA performance than the other models, we still observed an accuracy drop after the intervention for aspects such as Node Color, Node Shape, and Edge Color.

Gemma3-4B-IT. Table 12 and Figure 44 report the results for Gemma3-4B-IT. We observe a drop in performance after the intervention for multiple aspects, including Node Color, Node Shape, and

Edge Color, indicating that linearly decodable information in both the vision encoder and the language model contributes to VQA inference.

Across all four models, replacing hidden states at positions with high probing accuracy with the dataset-wide mean reduced accuracy for Node Color, Node Shape, and Edge Color in both the vision encoder and the language model. Qwen3-VL and Qwen2.5-VL additionally showed drops for Node Count. These results indicate that the signals identified by probing are not merely decodable but contribute causally to VQA predictions. In contrast, aspects that are poorly captured by linear probes, such as Edge Direction, Edge Existence, and Multi-hop Path, showed little to no change after intervention. For these aspects, VQA accuracy is already close to chance level, leaving little room for further decreases, so the intervention is not informative about their causal contribution.

F Experiments on Data Closer to Real-World Diagrams

To examine whether the trends generalize beyond synthetic graphs, we conduct VQA and probing experiments on flowchart images derived from FlowVQA (Singh et al., 2024), using Qwen3-VL 8B as the target model. Although the dataset remains controlled because node and edge attributes are edited and augmented, the flowcharts contain features characteristic of real-world flowcharts, such as text labels and canonical layouts.

F.1 Evaluation Aspects and Class Configuration

We analyze 10 of the 11 aspects defined in § 3.2 of the main text, excluding Node Shape (§ 7).

The class configuration for each aspect is shown in Table 13. While the configuration largely matches that of the synthetic data, it differs in the following respects:

- In-degree / Out-degree: 5 classes (0–4) in the synthetic setting → 4 classes (1–4) in the flowchart setting. Except for the start and end nodes, every node in a flowchart is connected by at least one incoming and one outgoing edge, so a degree of zero does not arise.
- Node Count: in the synthetic setting, every diagram contains exactly five nodes, so this aspect asks how many nodes share the identifier A (5 classes, 1–5). In the flowchart setting, the

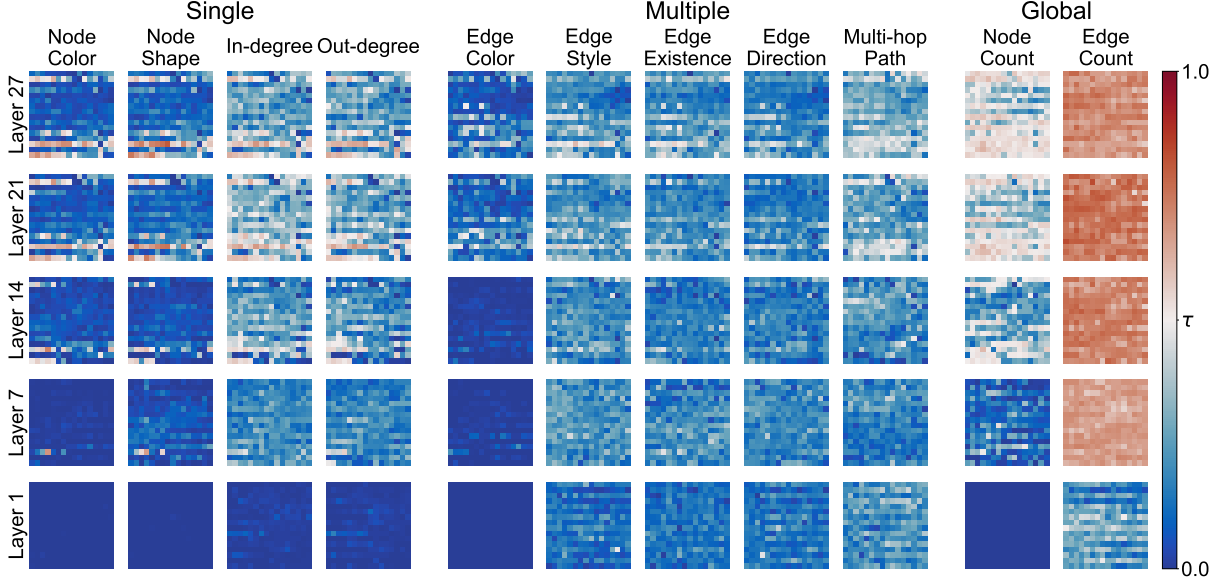


Figure 23: Accuracy for each hidden state in the vision encoder of Qwen3-VL 8B (7-node graphs). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were consistent with the 5-node setting (Figure 4), although fewer patches exceeded the threshold.

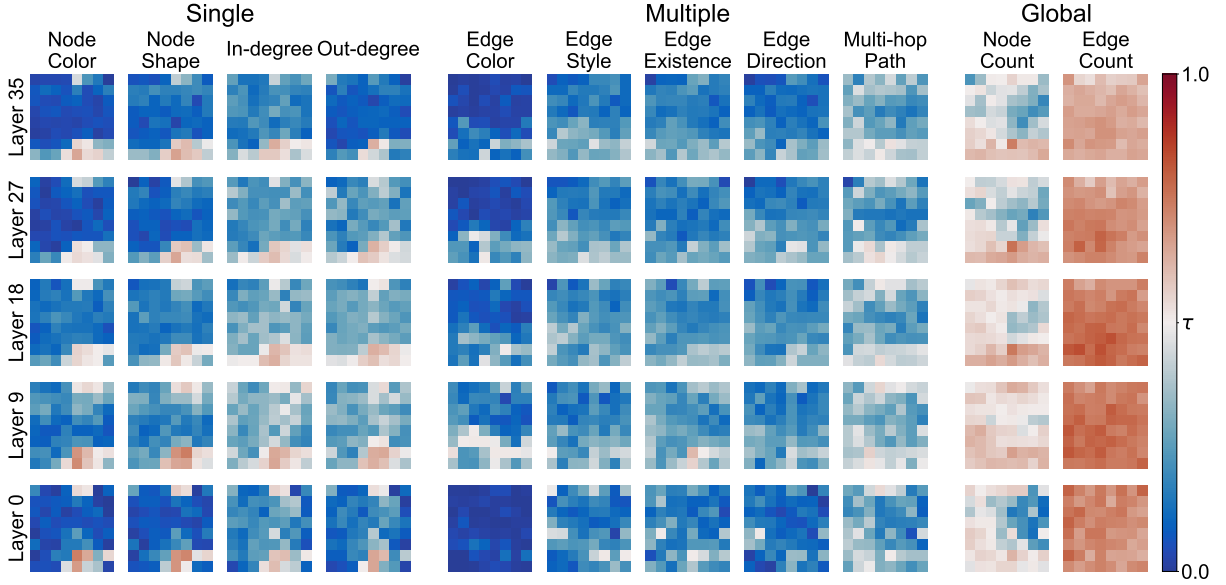


Figure 24: Accuracy for each hidden state in the image-input part of the Qwen3-VL 8B language model (7-node graphs). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were consistent with the 5-node setting (Figure 6), although fewer patches exceeded the threshold.

number of nodes varies across flowcharts, so it asks for the total number of nodes (4 classes, 7–10).

- **Edge Count:** 5 classes (1–5) in the synthetic setting $\rightarrow$ 4 classes (7–10) in the flowchart setting. We adjusted the range to preserve the distribution of the original FlowVQA data while matching the synthetic data in complexity.

F.2 Dataset Construction

We construct the dataset in three steps: (1) sampling base flowcharts from FlowVQA, (2) editing and data augmentation, and (3) Mermaid rendering.

Step 1: Sampling base flowcharts. To keep the graph complexity comparable to the synthetic setting (§ 3), we restrict the source to FlowVQA samples with at most 10 nodes. From this pool, we independently sample base flowcharts for each as-

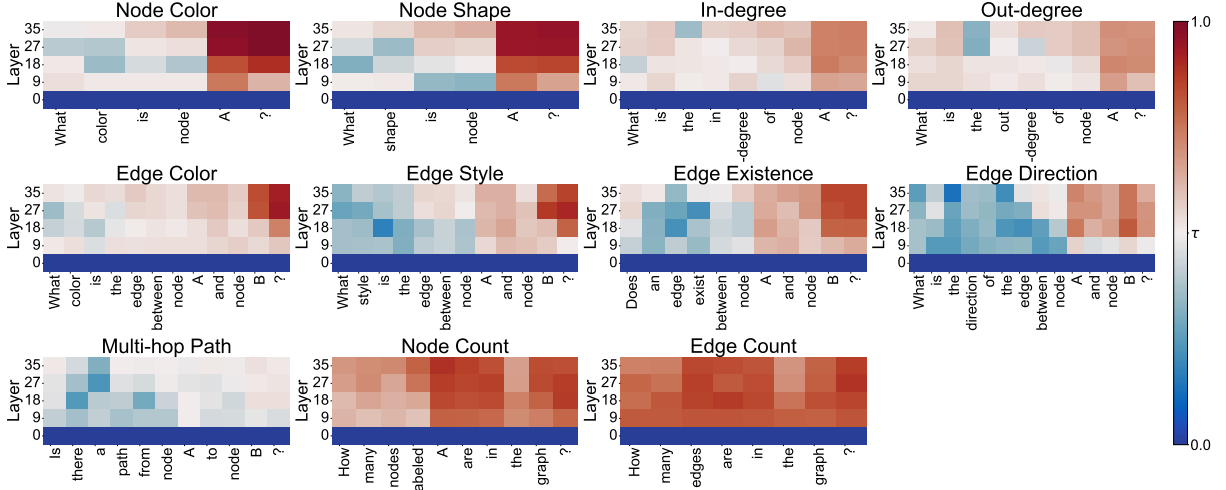


Figure 25: Probing results for text tokens in Qwen3-VL 8B (7-node graphs). The x-axis shows token position in the question, and the y-axis shows layer position. Trends were consistent with the 5-node setting (Figure 7).

Component	Layer	Accuracy-based (%)	Controlled (%)	Diff.
Vision encoder	1	25.6	42.4	-16.8
	14	51.6	66.5	-14.9
	27	50.4	72.1	-21.7
Language model	0	30.7	51.8	-21.2
	18	54.8	75.8	-20.9
	35	83.9	83.8	+0.1

Table 9: VQA accuracy for Edge Count under the selective (top-50%) intervention on Qwen3-VL 8B, comparing the *accuracy-based* setting (intervening on the highest-probing-accuracy patches) with the *controlled* setting (intervening on the same number of randomly selected patches). Values are averaged over the evaluation seeds.

	Layer	Node Color	Node Shape	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Vision encoder	1	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	-0.9
	16	-48.8	-34.9	-7.1	-5.3	± 0.0	-0.1	± 0.0	± 0.0	+0.3	-29.6	± 0.0
	32	-48.4	-38.0	-9.6	-7.2	-14.3	-1.3	-1.2	-0.1	+0.4	-3.2	± 0.0
Language model	0	-60.0	-48.2	-13.2	-6.9	-25.9	-6.7	-2.3	+0.3	+1.1	-3.1	± 0.0
	14	-44.8	-36.8	-9.5	-3.9	-27.4	-1.9	-1.3	+0.2	+1.4	-0.7	± 0.0
	27	± 0.0	-0.1	± 0.0	+0.1	-0.2	-0.1	-0.2	+0.1	-0.3	-0.1	± 0.0

Table 10: Accuracy difference between *accuracy-based* and *controlled* settings (*accuracy-based* - *controlled*) for Qwen2.5-VL 7B. More negative values (**bold**) indicate a larger accuracy decrease under the *accuracy-based* intervention than under the *controlled* intervention.

pect.

For aspects that analyze a specific node or node pair (the Single and Multiple categories), we sample 10 flowcharts for training and one for evaluation, all sharing a common target node (or node pair). Aspects in the Single category target the node with the label “Return True”, while aspects in the Multiple category target the node pair (“Return True”, “End”). We choose these as targets because they are the labels that most frequently appear across FlowVQA flowcharts, allowing us to

fix a consistent target across aspects while retaining a sufficient number of candidate flowcharts.

In contrast, for aspects in the Global category (Node Count, Edge Count), which analyze the diagram as a whole, the classification target is the total number of nodes (or edges) in the flowchart. We therefore sample 10 training and 1 evaluation flowcharts for each class, where each class corresponds to a specific node count (or edge count). For example, for Node Count, we sample flowcharts with 7, 8, 9, and 10 nodes for each respective class.

	Layer	Node Color	Node Shape	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Vision encoder	1	-2.4	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	+0.9
	12	-15.1	-5.5	-1.4	-0.7	-0.2	-0.1	-0.2	± 0.0	± 0.0	± 0.0	± 0.0
	23	-8.8	-1.6	± 0.0	-0.6	-4.2	+0.4	+0.4	± 0.0	± 0.0	± 0.0	± 0.0
Language model	0	-12.8	-5.7	± 0.0	+0.1	-3.9	-0.5	+0.3	± 0.0	± 0.0	± 0.0	± 0.0
	16	-7.9	-2.3	+0.0	+0.4	-3.6	-0.1	-0.8	± 0.0	± 0.0	± 0.0	± 0.0
	31	-0.1	+0.1	-0.2	+0.4	+0.1	-0.3	± 0.0	± 0.0	± 0.0	± 0.0	+0.0

Table 11: Accuracy difference between *accuracy-based* and *controlled* settings (*accuracy-based* – *controlled*) for LLaVA1.5 7B. More negative values (**bold**) indicate a larger accuracy decrease under the *accuracy-based* intervention than under the *controlled* intervention.

	Layer	Node Color	Node Shape	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Vision encoder	1	± 0.0	± 0.0	± 0.0	± 0.0	+0.1	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0
	14	-39.6	-18.6	-4.3	-6.7	-3.1	-0.9	+0.2	+0.3	± 0.0	+0.3	-3.3
	27	-30.2	-15.6	-1.8	-4.3	+0.5	± 0.8	-0.1	+0.1	± 0.0	-5.3	-0.1
Language model	0	-34.9	-25.1	-1.0	-3.0	-28.8	-5.7	-2.3	+0.3	-0.8	+0.1	-0.9
	17	-17.9	-3.9	-0.6	-0.3	-3.0	± 0.0	± 0.0	-0.1	± 0.0	± 0.0	± 0.0
	33	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	+0.1	± 0.0	± 0.0	± 0.0	± 0.0

Table 12: Accuracy difference between *accuracy-based* and *controlled* settings (*accuracy-based* – *controlled*) for Gemma3-4B-IT. More negative values (**bold**) indicate a larger accuracy decrease under the *accuracy-based* intervention than under the *controlled* intervention.

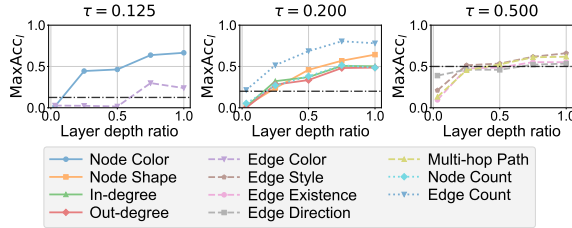


Figure 26: MaxAcc_l per layer in the vision encoder of Qwen2.5-VL 7B. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . Trends were similar to Qwen3-VL (Figure 3): Single- and Global-category aspects rose with depth, while Multiple-category aspects stayed near the threshold.

We restrict evaluation to a single base flowchart (one per class for aspects in the Global category) because computing patch-wise accuracy requires a fixed node layout within each evaluation set. Using flowcharts with random node layouts for training and a fixed layout for evaluation also aligns with the synthetic setting.

Step 2: Editing and data augmentation. For each base flowchart, we generate samples for each class by editing the attributes of the target node (or pair), following the edit operations in Table 14. Aspects in the Global category require no such edit

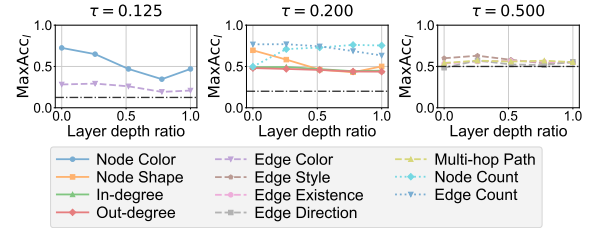


Figure 27: MaxAcc_l per layer in the language model of Qwen2.5-VL 7B. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . Trends were similar to Qwen3-VL (Figure 5).

because their classes are determined at sampling time. We then apply the following augmentations to obtain 100 samples per class:

- Common: randomize the colors of non-target nodes, edge colors, and edge styles.
- Training only: shuffle the labels of non-target nodes and randomize their shapes.

We apply label shuffling and shape randomization only to the training data to prevent the probe from exploiting non-target node text labels or shapes as shortcuts.

Step 3: Mermaid rendering. We render the edited Mermaid code as PNG images using

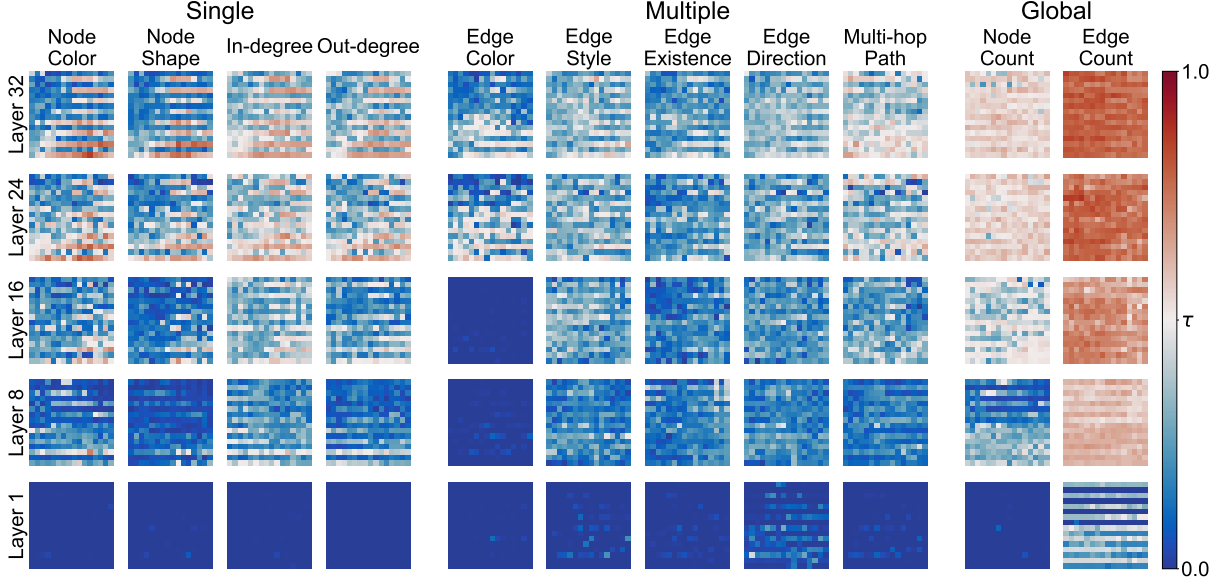


Figure 28: Accuracy for each hidden state in the vision encoder of Qwen2.5-VL 7B. Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were similar to Qwen3-VL (Figure 4).

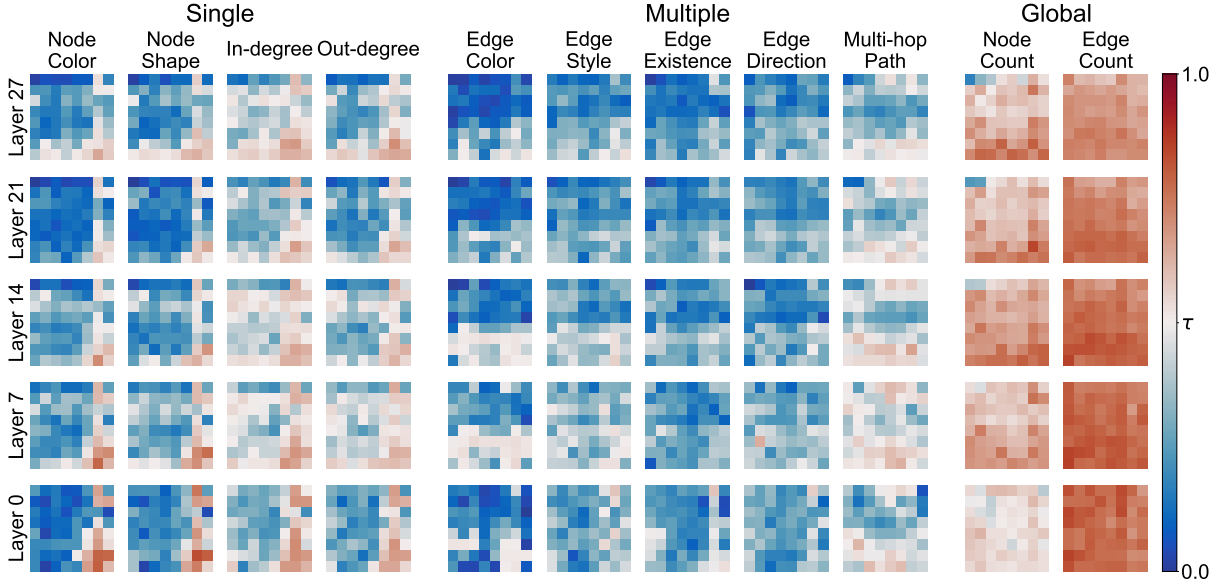


Figure 29: Accuracy for each hidden state in the image-input part of the Qwen2.5-VL 7B language model. Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were similar to Qwen3-VL (Figure 6).

mermaid-cli and provide them directly to Qwen3-VL without resizing or other preprocessing. Figure 45 shows examples of the rendered flowcharts.

F.3 Preliminary Experiment: VQA

We design our own multiple-choice questions for this analysis rather than using those provided in FlowVQA. The questions follow the same format as in § 4 of the main text, explicitly specifying the target node (or pair) for each aspect. The full list of

questions for each aspect is provided in Table 15. As in the synthetic setting, we evaluate VQA using accuracy.

F.4 Probing

Setup. We use the same single-layer linear classifier as in § 5 of the main text, training separate probes on the hidden states of the vision encoder, the image-input part of the language model, and the text-input part of the language model. The probe

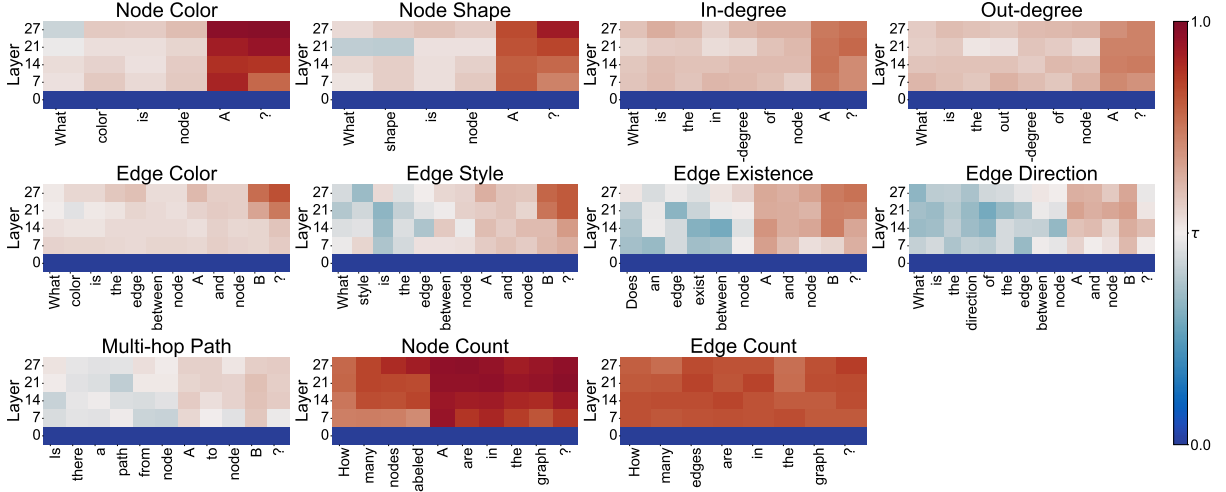


Figure 30: Probing results for text tokens in Qwen2.5-VL 7B. The x-axis shows token position in the question, and the y-axis shows layer position. Trends were similar to Qwen3-VL (Figure 7).

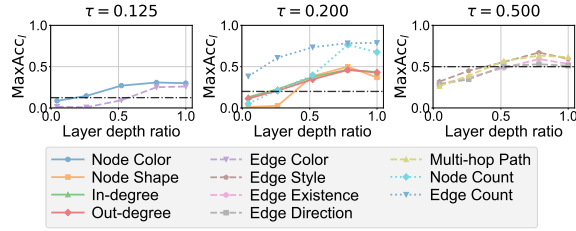


Figure 31: MaxAcc_l per layer in the vision encoder of LLaVA1.5 7B. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . Trends were similar to Qwen3-VL (Figure 3).

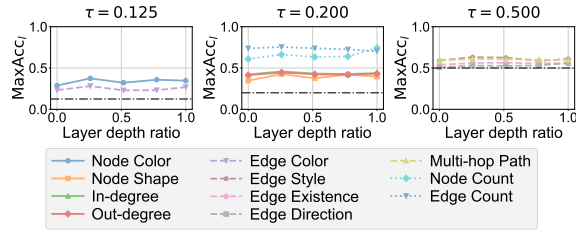


Figure 32: MaxAcc_l per layer in the language model of LLaVA1.5 7B. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . Trends were similar to Qwen3-VL (Figure 5).

architecture and training procedure are identical to those used on the synthetic data. We modify two hyperparameters of the vision encoder probe, as listed in Table 16: we reduce the maximum number of epochs to 200, which we found sufficient for convergence, and shift the batch-size search range to larger values to accommodate the increased num-

Category	Aspect a	Label set $\mathcal{Y}$
Single	Node Color	{red, green, yellow, blue, brown, orange, pink, purple}
	In-deg.	{1, 2, 3, 4}
	Out-deg.	{1, 2, 3, 4}
Multiple	Edge Color	{red, green, yellow, blue, brown, orange, pink, purple}
	Edge Style	{solid, dotted}
	Edge Direct.	{forward, backward}
	Edge Exist.	{exist, not exist}
	Multi-hop Path	{exist, not exist}
Global	Node Count	{7, 8, 9, 10}
	Edge Count	{7, 8, 9, 10}

Table 13: Class configuration of the 10 aspects analyzed in the flowchart setting.

ber of patches caused by the higher input resolution. As in the synthetic setting, we perform a layer-wise analysis that tracks accuracy across layers and a position-wise analysis that examines how accuracy is distributed across patch positions (for the vision encoder and the image-input part) or token positions (for the text-input part).

All aspects, including those in the Multiple category, are linearly decodable in the vision encoder. Figure 46 shows MaxAcc_l at each layer of the vision encoder, and Figure 47 shows the patch-wise accuracy distribution. For all aspects, accuracy exceeded the threshold within the vision encoder, indicating that the information for every aspect is linearly decodable from a single hidden state already within the vision encoder. In the patch-wise distribution, aspects in the Single category showed high-accuracy patches concentrated around

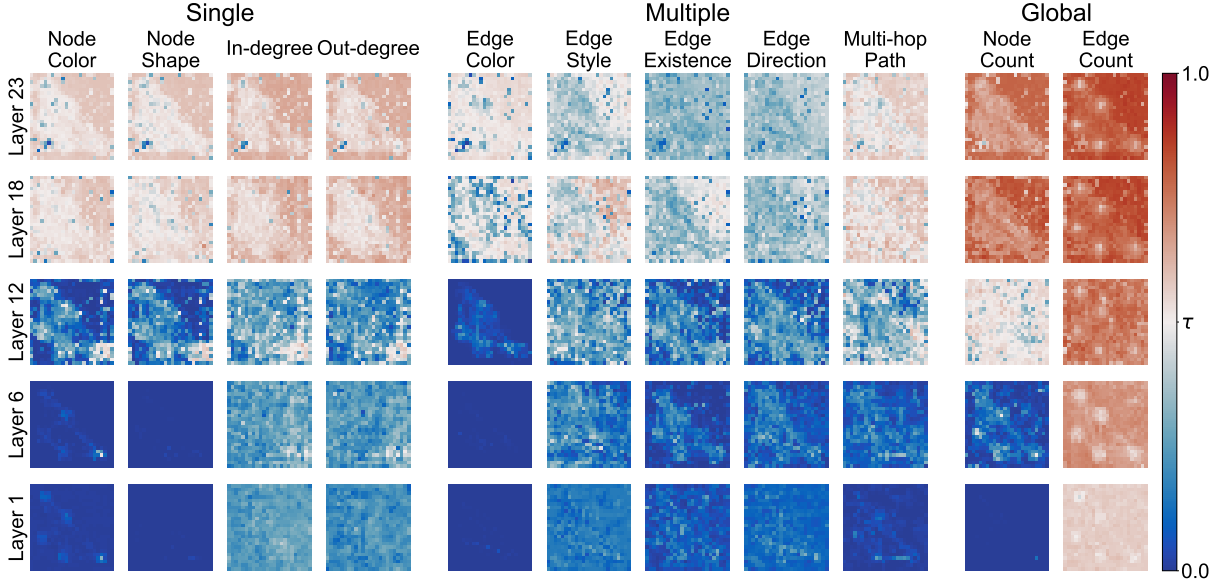


Figure 33: Accuracy for each hidden state in the vision encoder of LLaVA1.5 7B. Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Unlike Qwen3-VL (Figure 4), background patches showed higher accuracy than patches corresponding to node positions.

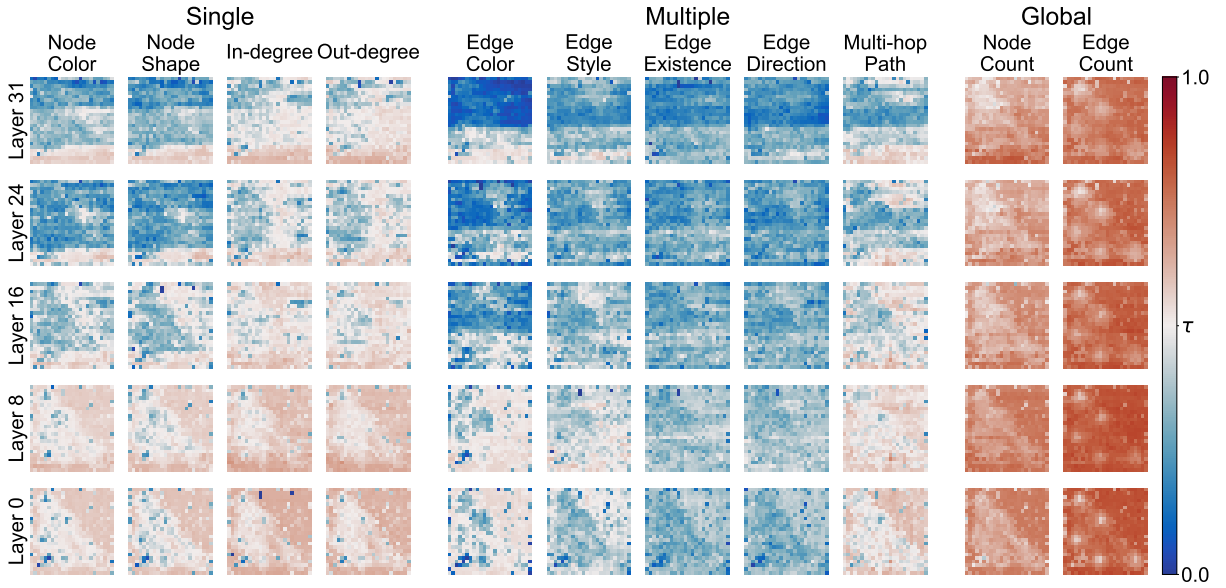


Figure 34: Accuracy for each hidden state in the image-input part of the LLaVA1.5 7B language model. Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Unlike Qwen3-VL (Figure 6), background patches showed higher accuracy than patches corresponding to node positions.

the target node, while aspects in the Global category showed high accuracy across a broad range of patches, including background patches. These spatial patterns for the two categories are consistent with the synthetic results (§ 5.3). In contrast, aspects in the Multiple category showed a different trend from the synthetic data: while most aspects in the Multiple category did not exceed the threshold within the vision encoder in the synthetic setting,

in this setting all aspects in the Multiple category exceeded the threshold.

Linearly decodable signals are preserved in the image-input part of the language model. Figure 48 shows MaxAcc_l across language model layers, and Figure 49 shows the patch-wise accuracy distribution. All aspects maintained accuracy above the threshold throughout the image-input part, and the patch-wise distribution showed little

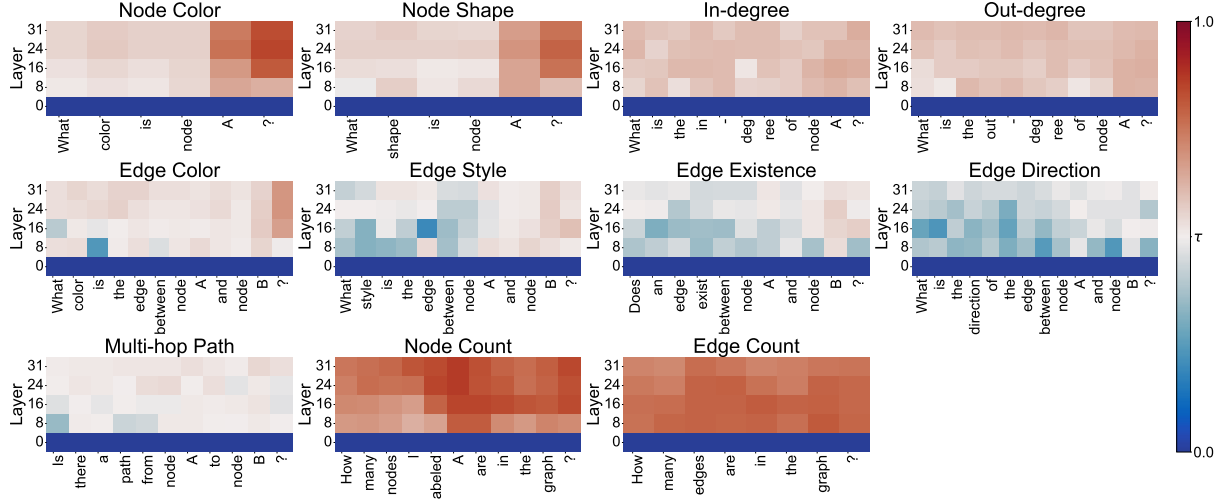


Figure 35: Probing results for text tokens in LLaVA1.5 7B. The x-axis shows token position in the question, and the y-axis shows layer position. Trends were broadly similar to Qwen3-VL (Figure 7), though less pronounced.

Aspect	Edit operation
Node Color	Randomly select the color of the target node from 8 colors.
In-deg.	Add or delete in-edges to set the in-degree of the target node to one of 1–4.
Out-deg.	Add or delete out-edges to set the out-degree of the target node to one of 1–4.
Edge Color	Randomly select the color of the edge between the target pair from 8 colors.
Edge Style	Change the edge between the target pair to solid or dotted.
Edge Direct.	Reverse the direction of the edge between the target pair.
Edge Exist.	Add or delete the edge between the target pair.
Multi-hop Path	Edit intermediate edges to change the reachability between the target pair.
Node Count	No edits (the class is determined at sampling time).
Edge Count	No edits (the class is determined at sampling time).

Table 14: Edit operations for each aspect.

Aspect	Question
Node Color	What color is the ‘Return True’ node?
In-deg.	What is the in-degree of the ‘Return True’ node?
Out-deg.	What is the out-degree of the ‘Return True’ node?
Edge Color	What color is the edge between the ‘Return True’ node and the ‘End’ node?
Edge Style	What style is the edge between the ‘Return True’ node and the ‘End’ node?
Edge Direct.	What is the direction of the edge between the ‘Return True’ node and the ‘End’ node?
Edge Exist.	Does an edge exist between the ‘Return True’ node and the ‘End’ node?
Multi-hop Path	Is there a path from the ‘Return True’ node to the ‘End’ node?
Node Count	How many nodes are in the graph?
Edge Count	How many edges are in the graph?

Table 15: Questions used to evaluate each aspect in the flowchart setting. “Answer with only one word from the choices.” and “ (Choices: <comma-separated-choices>)” are appended to every question.

change from the vision encoder result. These results indicate that the linearly decodable signals observed in the vision encoder were preserved through the image-input part, consistent with the synthetic results (§ 5.3).

Signals for Edge Direction and Edge Existence are confined to intermediate text-input layers. Figure 50 shows $acc_{l,t}$ across layers and token positions in the text-input part

of the language model. Similar to the synthetic results (§ 5.3), accuracy was particularly high at token positions that specify the target node or pair. However, for Edge Direction and Edge Existence, high accuracy appeared only in intermediate layers and did not propagate to the final layer. This suggests that, although signals corresponding to these aspects exist within the model, they are not propagated to the final layer; this differs from the synthetic results, where these

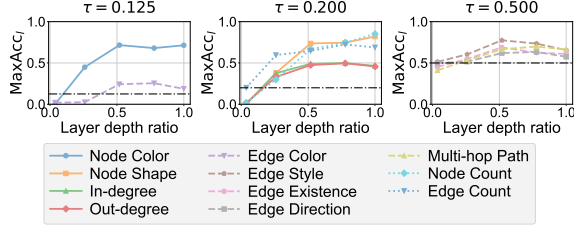


Figure 36: MaxAcc_l per layer in the vision encoder of Gemma3-4B-IT. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . Trends were similar to Qwen3-VL (Figure 3), though accuracy was lower for some aspects.

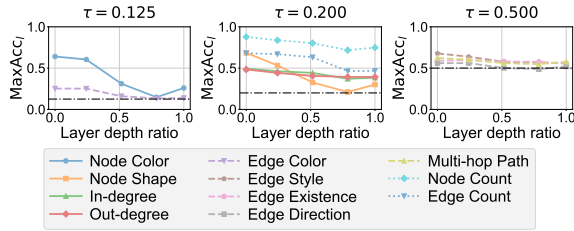


Figure 37: MaxAcc_l per layer in the language model of Gemma3-4B-IT. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . Trends were similar to Qwen3-VL (Figure 5), though accuracy was lower for some aspects.

Parameter	Vision encoder probe
Epochs	200
Batch size*	{4000, 8000, 16000, 32000, 64000}

Table 16: Hyperparameters of probe training in the flowchart setting that differ from the synthetic setting (Table 8). All other hyperparameters are identical to those of the synthetic setting. *Subject to hyperparameter search.

aspects maintained high accuracy through the final layer.

Possible causes of the differences from the synthetic results. The results in this setting differ from those on the synthetic data in two respects: (1) aspects in the Multiple category became linearly decodable already within the vision encoder, and (2) in the text-input part of the language model, the signals for Edge Direction and Edge Existence remained in intermediate layers and did not propagate to the final layer. Several factors may explain these differences, including the complexity of text labels in FlowVQA flowcharts, the higher input im-

age resolution, and the diversity of node and edge layouts.

G Analysis of LLaVA Training Data

Prior studies report that LVLMs struggle to understand relational information indicated by arrows and lines (Hou et al., 2025; Zhu et al., 2025; Pan et al., 2024). Consistent with these findings, our VQA experiments show that all models achieved near-chance accuracy on Edge Direction (§ 4.2, Appendix C). One possible explanation is that text describing arrow directions is underrepresented in the training data. To investigate this hypothesis, we analyze the publicly available pre-training and instruction tuning datasets of LLaVA1.5.

G.1 Setup

We quantify how many samples mention arrow directions in the pre-training and instruction tuning datasets of LLaVA1.5. First, we filter samples whose text contains the substring “arrow.” We lowercase the text before matching. We then classify whether each filtered sample describes an arrow direction (e.g., pointing left or upward) using gpt-oss-120b (OpenAI, 2025) as an LLM judge. Note that we analyze only the text portion of each sample, without referring to the associated images.

G.2 Results

The substring “arrow” appears in 1,108 of 558,128 pre-training samples (0.20%) and in 4,408 of 665,298 instruction tuning samples (0.66%). Among these, only 161 pre-training samples (0.03% of all pre-training samples) and 646 instruction tuning samples (0.10%) were judged as describing an arrow direction. These results suggest that the scarcity of arrow direction descriptions in the training data may be one factor contributing to the near-chance Edge Direction accuracy. However, since our analysis targets only the text portion and does not account for visual information in the images, further investigation is needed to draw stronger conclusions. Additionally, this analysis covers only LLaVA1.5; whether similar trends hold for the training datasets of other models remains an open question.

H Fine-Tuned Model Analysis

To investigate whether training on diagram data changes how LVLMs represent visual information,

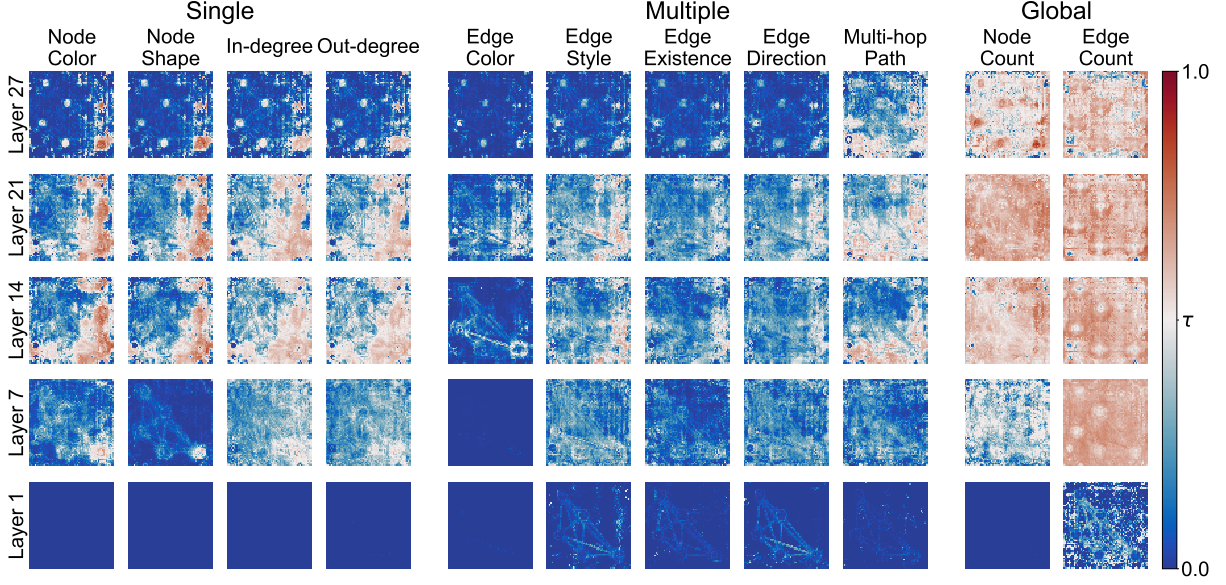


Figure 38: Accuracy for each hidden state in the vision encoder of Gemma3-4B-IT. Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were similar to Qwen3-VL (Figure 4), though accuracy was lower for some aspects.

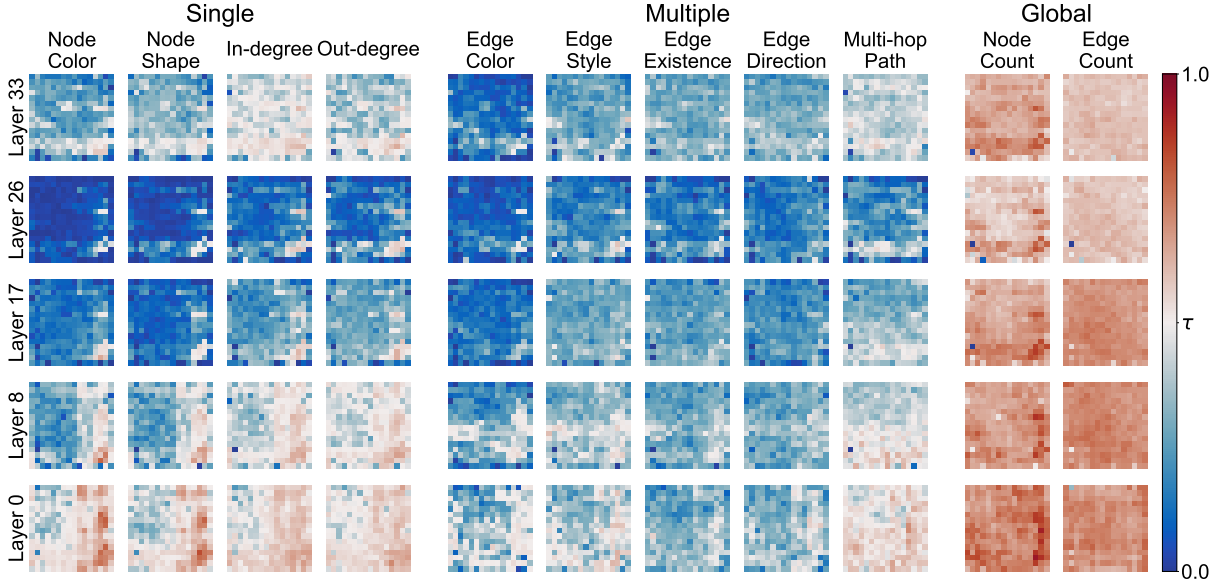


Figure 39: Accuracy for each hidden state in the image-input part of the Gemma3-4B-IT language model. Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect. Spatial patterns were similar to Qwen3-VL (Figure 6), though accuracy was lower for some aspects.

we fine-tune Qwen3-VL 8B on our synthetic diagram dataset and analyze the resulting changes in VQA accuracy and internal representations.

H.1 Training Setup

We fine-tune Qwen3-VL 8B using LoRA (Hu et al., 2022) in two configurations: (1) updating only the language model (FT w/L), and (2) updating both the vision encoder and the language model (FT w/VL). The input format follows the VQA setting

described in § 2: each training sample consists of a diagram image and a question, and the model is trained to output the corresponding label as text.

We combine $\mathcal{D}_{\text{rand}}$ from all 11 aspects into a single training set of 4,900 samples. These diagrams are generated with a different random seed from those used for probe training, so the fine-tuning data and the probe training data do not overlap. The LoRA and training hyperparameters are sum-

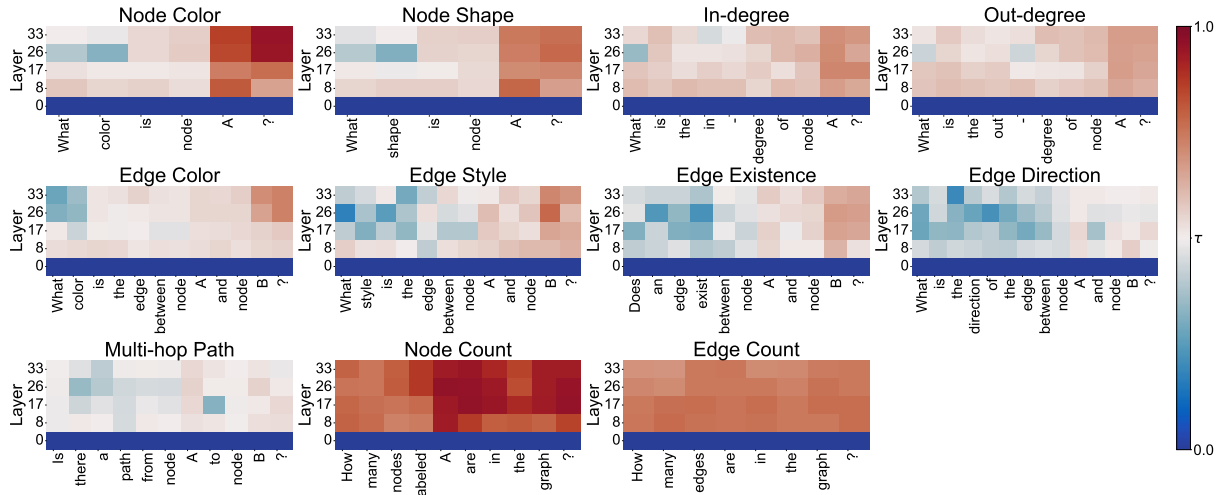


Figure 40: Probing results for text tokens in Gemma3-4B-IT. The x-axis shows token position in the question, and the y-axis shows layer position. Trends were broadly similar to Qwen3-VL (Figure 7), though less pronounced.

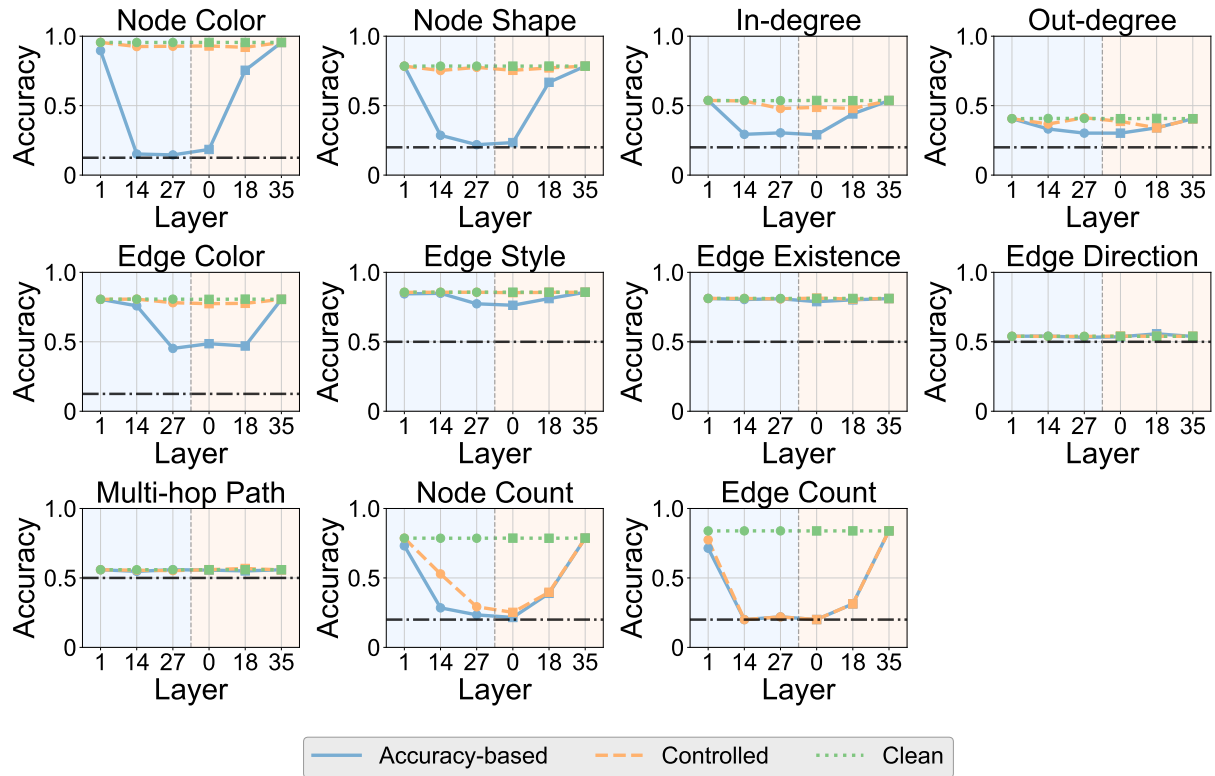


Figure 41: *Clean*, *accuracy-based*, and *controlled* accuracy at each intervention layer for Qwen3-VL 8B. The background colors of the plots indicate the results for the vision encoder and the language model, respectively.

marized in Table 17.

H.2 VQA Accuracy after Fine-Tuning

We evaluate the VQA accuracy of the fine-tuned models using the same protocol as in § 4. Table 18 reports the results. Fine-tuning improved accuracy on almost all aspects, with the exception of Node Color and Node Count under FT w/VL.

In particular, accuracy improved on all aspects

even when only the language model was fine-tuned without updating the vision encoder (FT w/L). Since the vision encoder is frozen in this configuration, this suggests that the improvement does not come from newly acquired signals in the vision encoder, but rather that the language model became able to use signals that were already present.

Comparing the two configurations, aspects in

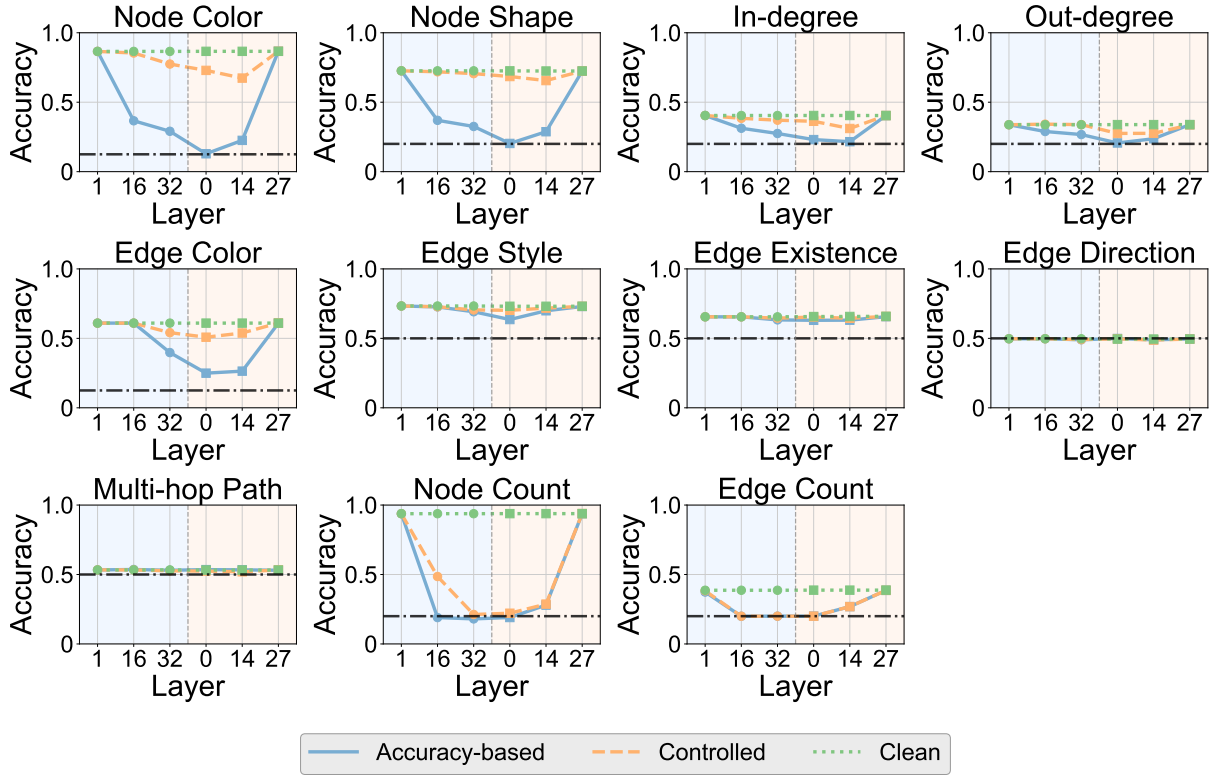


Figure 42: *Clean*, *accuracy-based*, and *controlled* accuracy at each intervention layer for Qwen2.5-VL 7B. The background colors of the plots indicate the results for the vision encoder and the language model, respectively.

Parameter	FT w/L	FT w/VL
Epochs	10	10
Batch size	8	8
Learning rate	1e-4	1e-4
Optimizer	AdamW	AdamW
betas	[0.9, 0.999]	[0.9, 0.999]
eps	1e-8	1e-8
weight decay	0.01	0.01
LoRA rank	4	4
LoRA alpha	16	16
Target modules	q_proj, v_proj	qkv, q_proj, v_proj

Table 17: Hyperparameters for LoRA fine-tuning.

the Single and Global categories achieved higher accuracy when only the language model was fine-tuned (FT w/L), with the exception of Out-degree. In contrast, aspects in the Multiple category consistently achieved higher accuracy when both the vision encoder and language model were fine-tuned (FT w/VL). This suggests that updating the vision encoder is beneficial mainly for aspects whose signals are not yet strongly present in the pre-trained vision encoder, such as those in the Multiple category.

H.3 Changes in Internal Representations

We apply the same probing analysis as in § 5 to the fine-tuned models and compare the results with those of the pre-trained model reported in § 5.3. Since FT w/L does not update the vision encoder, its vision encoder representations are identical to those of the pre-trained model; we therefore omit the corresponding figures and report only the language model results for this configuration. In both configurations, the layer-wise trends and the spatial distribution of high-accuracy patches remained largely unchanged compared with the pre-trained model (Figures 51–58). This suggests that, despite the improvement in VQA accuracy (Appendix H.2), the linear decodability measured by our probes remained largely unchanged after fine-tuning with LoRA.

This gap between the improved VQA accuracy and the unchanged linear decodability suggests that much of the required signal was already present in the pre-trained model. That is, fine-tuning may contribute mainly by making signals that already exist internally usable at inference time, since we would expect a corresponding change in linear decodability if the improvement came mainly from acquiring new information. This suggests that training the

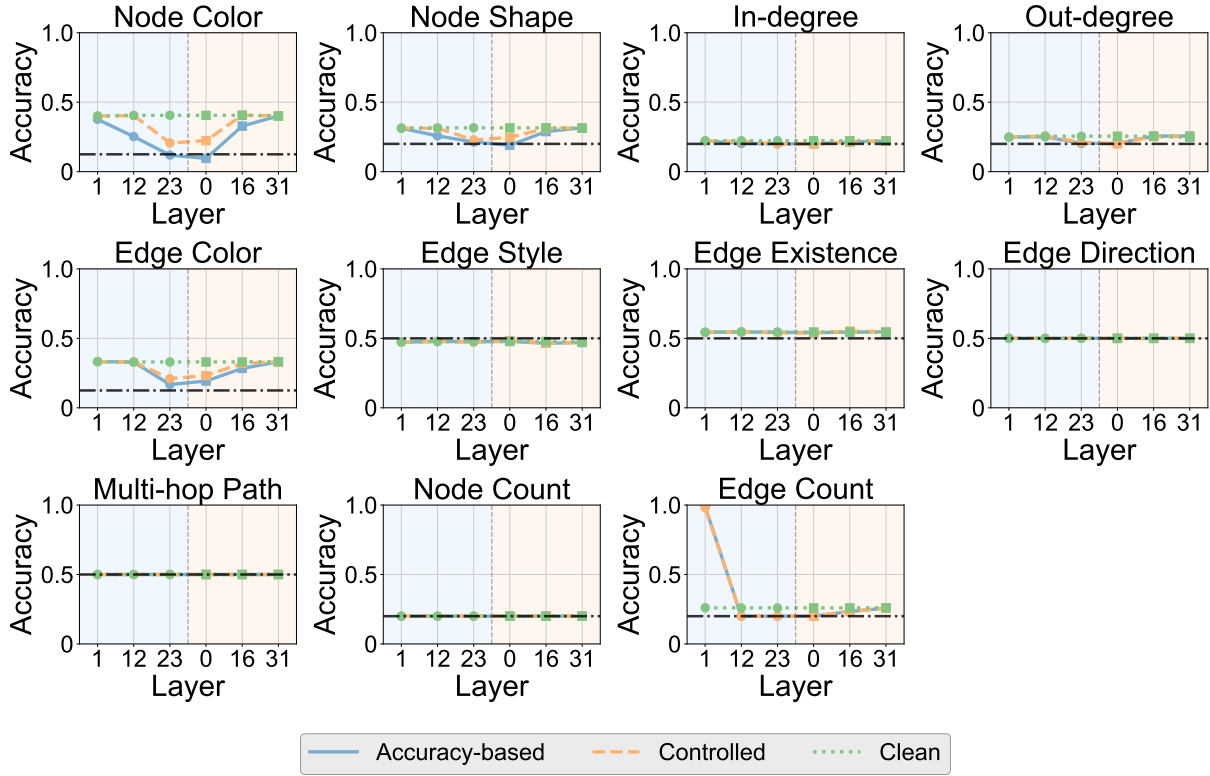


Figure 43: *Clean*, *accuracy-based*, and *controlled* accuracy at each intervention layer for LLaVA1.5 7B. The background colors of the plots indicate the results for the vision encoder and the language model, respectively.

	Node Color	Node Shape	In-deg.	Out-deg.	Edge Color	Edge Style	Edge Exist.	Edge Direct.	Multi-hop Path	Node Count	Edge Count
Baseline	95.5	78.4	53.7	40.6	80.6	85.7	81.2	53.9	55.9	78.7	83.8
FT w/L	99.0	86.8	71.1	66.0	90.6	91.1	86.1	70.8	59.1	79.9	88.3
FT w/VL	93.5	85.0	63.8	68.3	91.6	91.5	87.8	72.7	68.0	77.3	87.8
Chance level	12.5	20.0	20.0	20.0	12.5	50.0	50.0	50.0	50.0	20.0	20.0

Table 18: VQA accuracy of Qwen3-VL 8B before and after fine-tuning. Baseline denotes the pre-trained model without fine-tuning. FT w/L denotes fine-tuning only the language model, and FT w/VL denotes fine-tuning both the vision encoder and the language model. FT w/L improved accuracy on all aspects, while FT w/VL slightly decreased Node Color and Node Count. Single- and Global-category aspects benefited more from FT w/L, while Multiple-category aspects benefited more from FT w/VL.

language model to better exploit signals that are already present internally is a promising direction for improving diagram understanding. It remains possible, however, that probing simply fails to capture the changes. If LoRA updates primarily adjust nonlinear transformations or later decoding stages rather than the linear subspace, or if the low rank (rank 4) is insufficient to alter the global structure of the representation space, the representations may change without this being visible to probing. Disentangling these factors requires further investigation, which we leave as future work.

I Computational Resources

We used NVIDIA RTX 6000 Ada, RTX PRO 6000 Blackwell Max-Q, and H100 GPUs in this study.

J Usage of AI Assistants

For writing this paper and developing the experimental code, we used AI assistants (e.g., ChatGPT and GitHub Copilot). Their use was limited to tasks such as code completion, translation, text editing, and table formatting, and all technical content is based solely on the authors’ ideas.

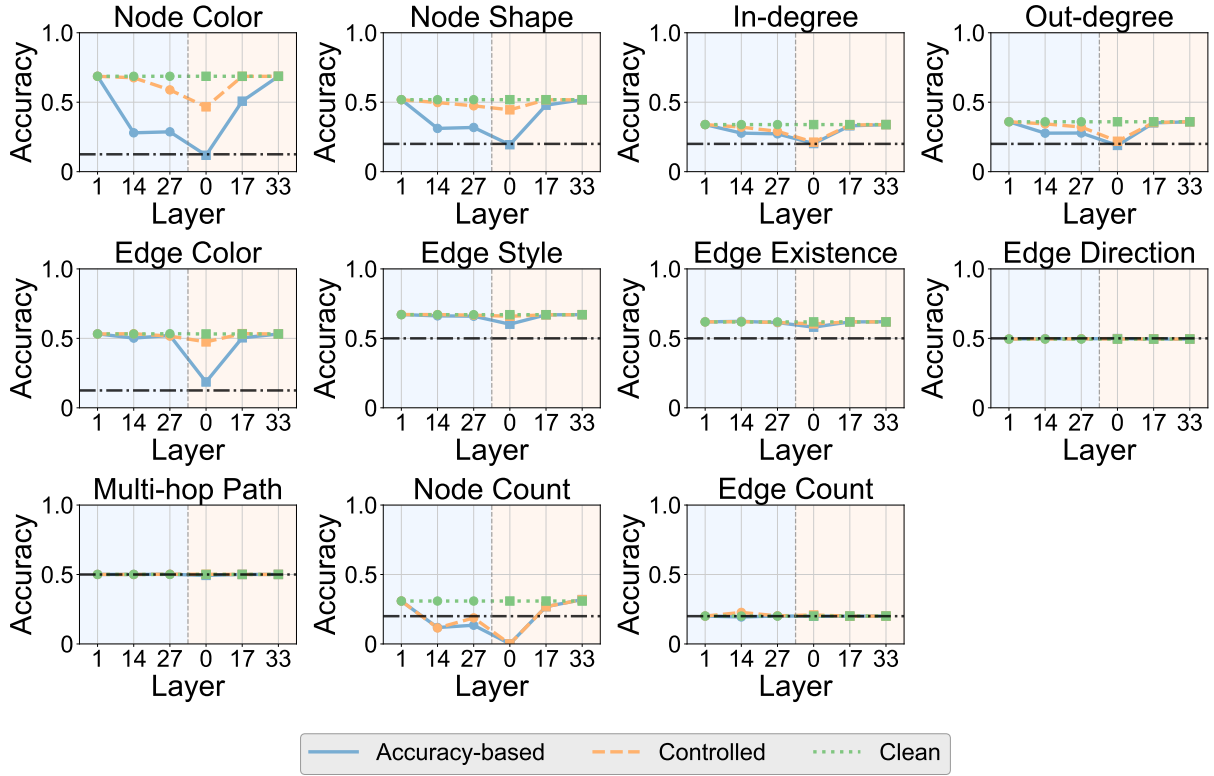


Figure 44: *Clean*, *accuracy-based*, and *controlled* accuracy at each intervention layer for Gemma3-4B-IT. The background colors of the plots indicate the results for the vision encoder and the language model, respectively.

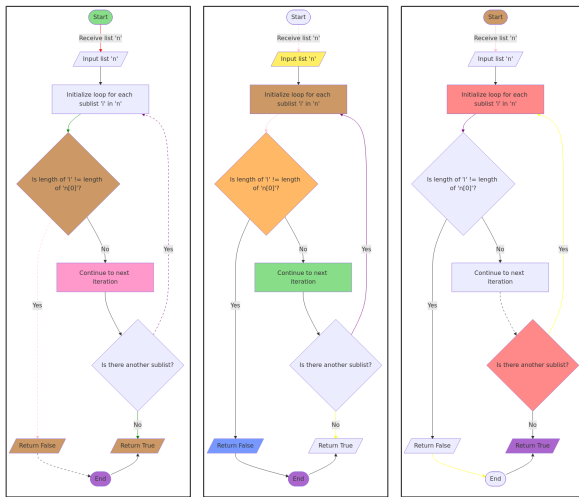


Figure 45: Examples of flowcharts used for evaluation. The three samples share the same node layout, while node colors, edge colors, and edge styles are randomized.

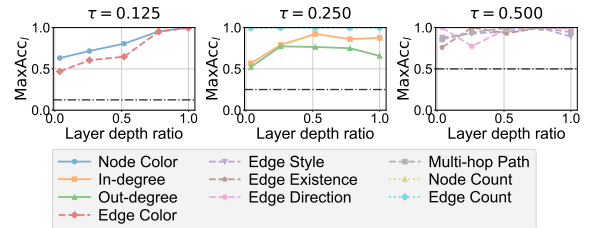


Figure 46: MaxAcc_l per layer in the vision encoder of Qwen3-VL 8B on flowcharts. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . All aspects exceeded the threshold within the vision encoder, including Multiple-category aspects whose accuracy did not reach the threshold in the synthetic setting.

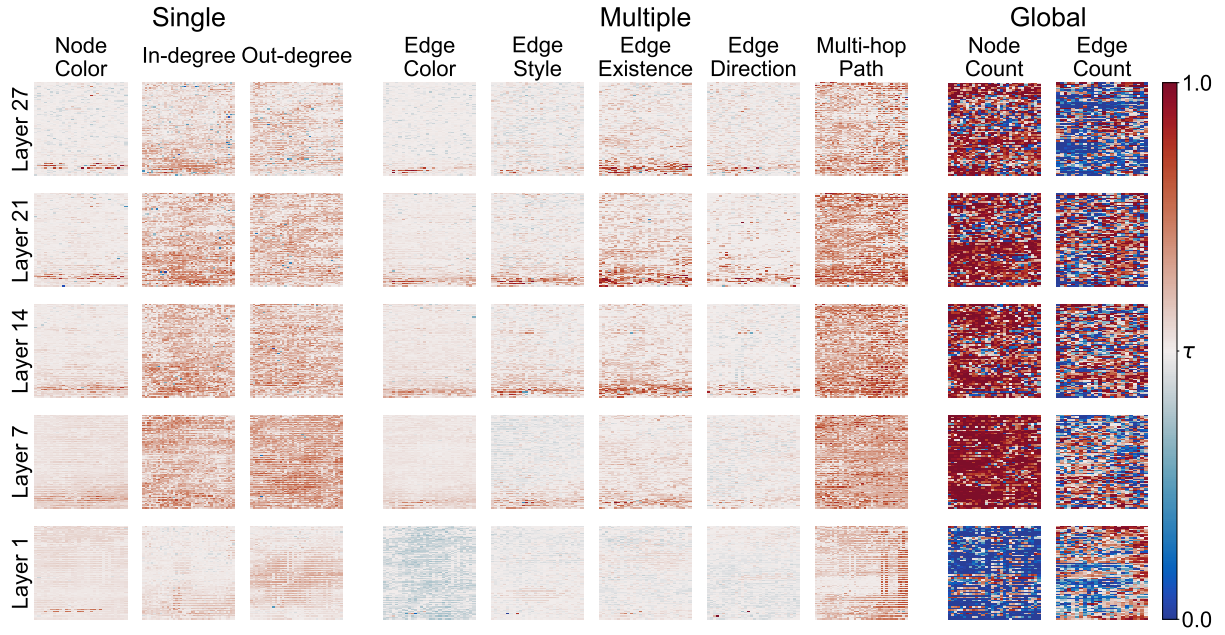


Figure 47: $acc_{l,t}(d_{fix})$ in the vision encoder of Qwen3-VL 8B on flowcharts, for a representative subset d_{fix} . Each heatmap shows accuracy by patch position for a specific layer and aspect. Single-category aspects localized near the target node, Global-category aspects spread across background patches, and Multiple-category aspects also showed high-accuracy patches, unlike in the synthetic setting.

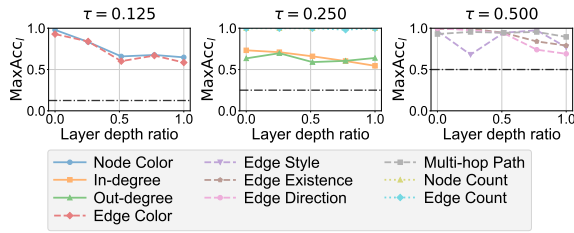


Figure 48: $MaxAcc_l$ per layer in the image-input part of the language model in Qwen3-VL 8B on flowcharts. The x-axis shows relative layer position (0 is the input layer, 1 is the final layer), and the y-axis shows accuracy. The black dash-dot line in each panel indicates the threshold τ . All aspects maintained accuracy above the threshold.

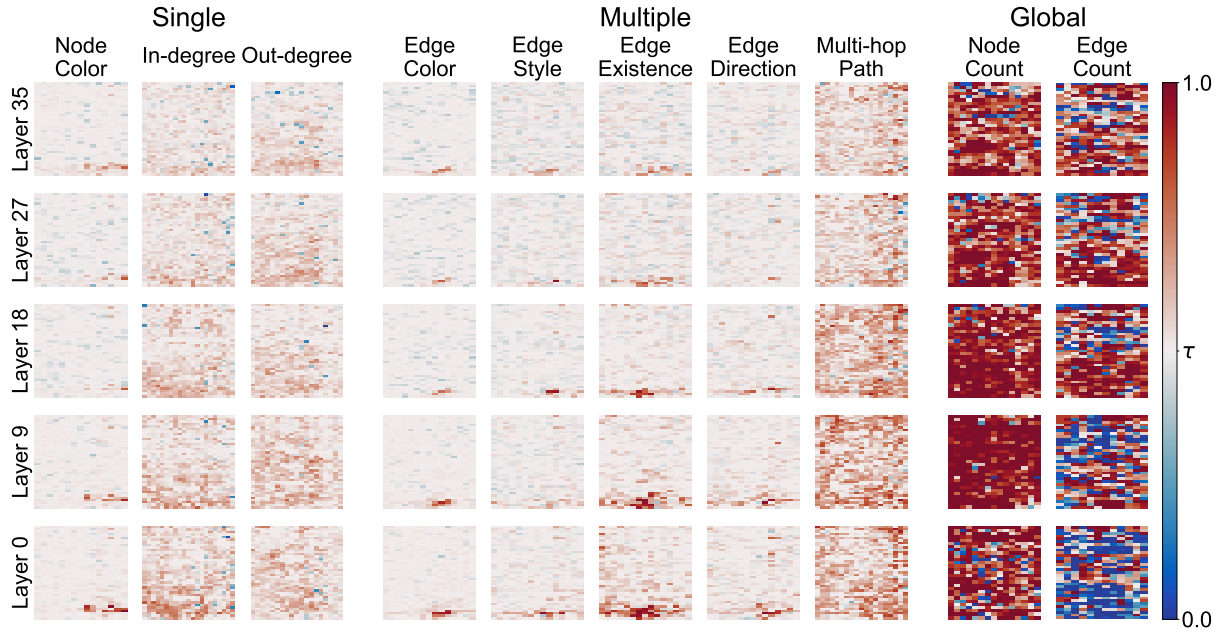


Figure 49: $acc_{l,t}(d_{fix})$ in the image-input part of the language model in Qwen3-VL 8B on flowcharts, for a representative subset d_{fix} . The spatial distribution of high-accuracy patches was largely unchanged from the vision encoder result.

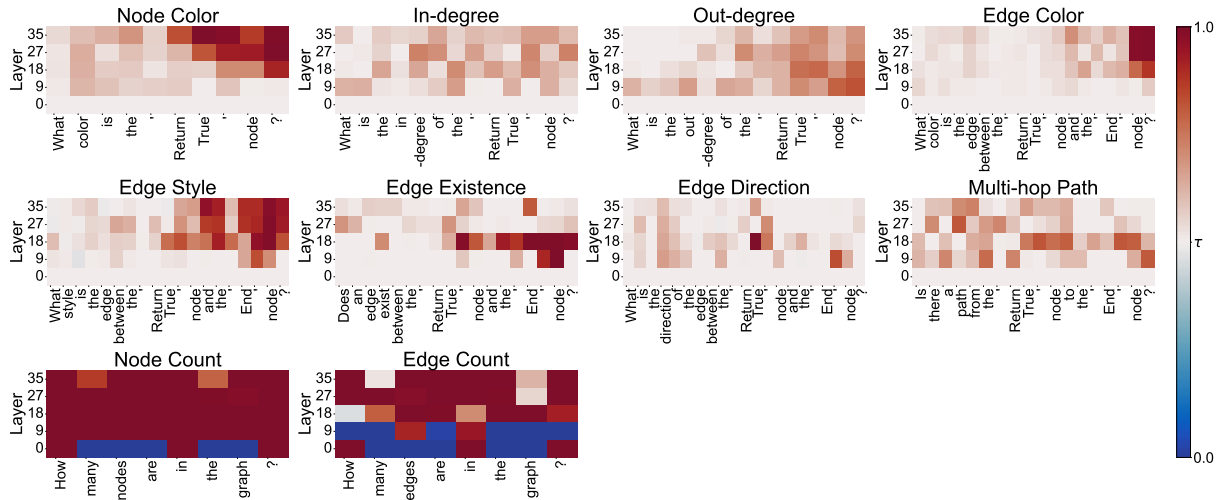


Figure 50: Probing results for text tokens in Qwen3-VL 8B on flowcharts. The x-axis shows token position in the question, and the y-axis shows layer position. For Edge Direction and Edge Existence, high accuracy appeared only in intermediate layers and did not propagate to the final layer.

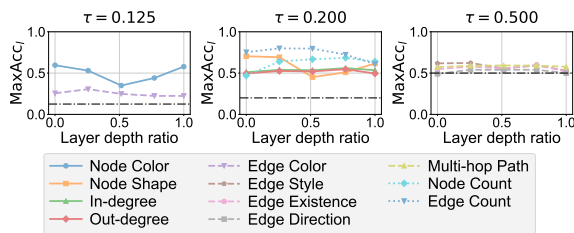


Figure 51: $MaxAcc_l$ per layer in the language model of Qwen3-VL 8B (FT w/L). Axes and the threshold line follow Figure 5. Trends were largely unchanged from the pre-trained model (Figure 5).

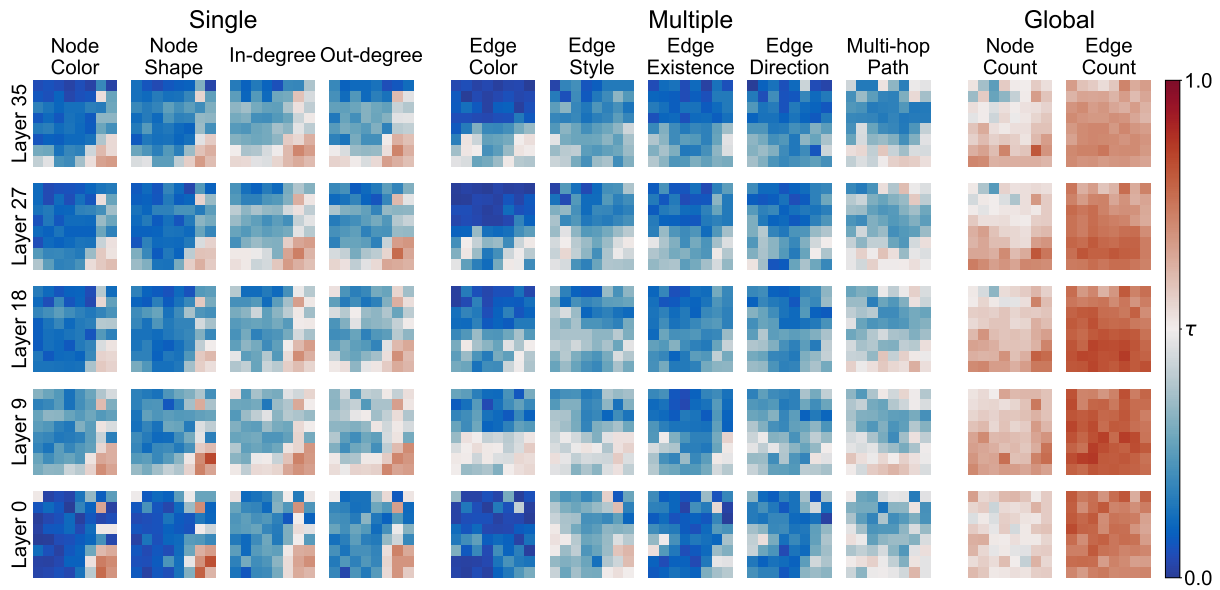


Figure 52: Accuracy for each hidden state in the image-input part of the Qwen3-VL 8B language model (FT w/L). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect.

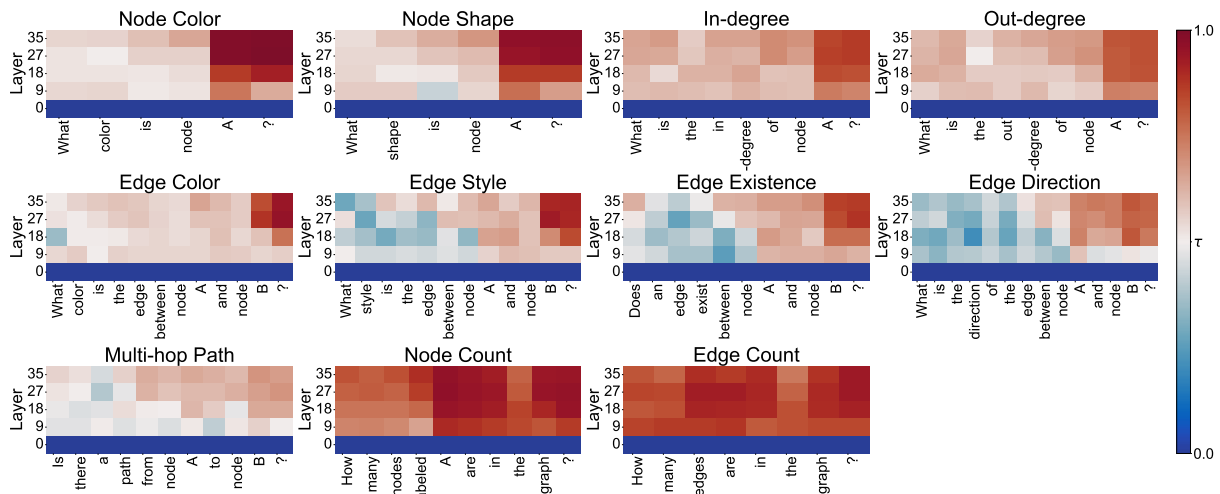


Figure 53: Probing results for text tokens in Qwen3-VL 8B (FT w/L). The x-axis shows token position in the question, and the y-axis shows layer position.

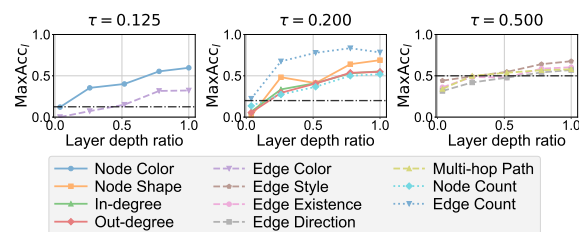


Figure 54: MaxAcc_l per layer in the vision encoder of Qwen3-VL 8B (FT w/VL). Axes and the threshold line follow Figure 3. Trends were largely unchanged from the pre-trained model (Figure 3).

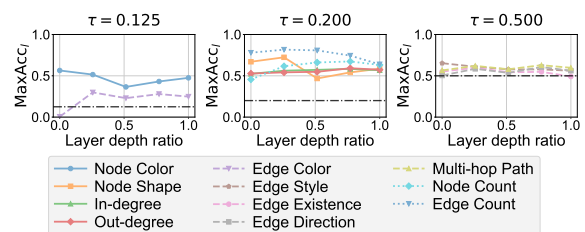


Figure 55: MaxAcc_l per layer in the language model of Qwen3-VL 8B (FT w/VL). Axes and the threshold line follow Figure 5. Trends were largely unchanged from the pre-trained model (Figure 5).

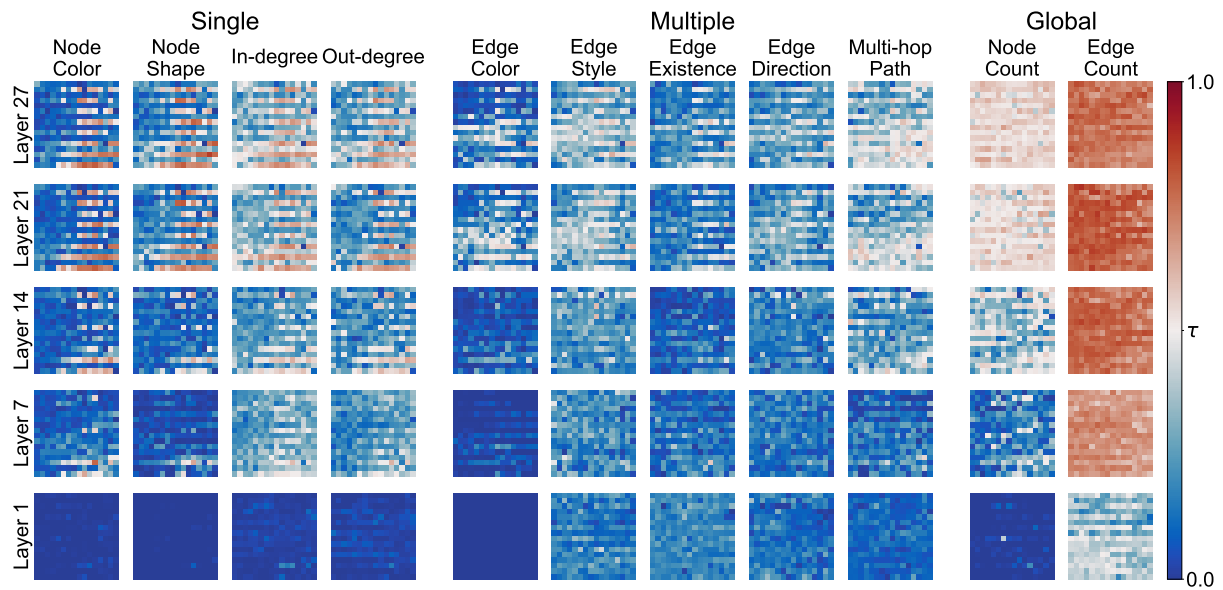


Figure 56: Accuracy for each hidden state in the vision encoder of Qwen3-VL 8B (FT w/VL). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect.

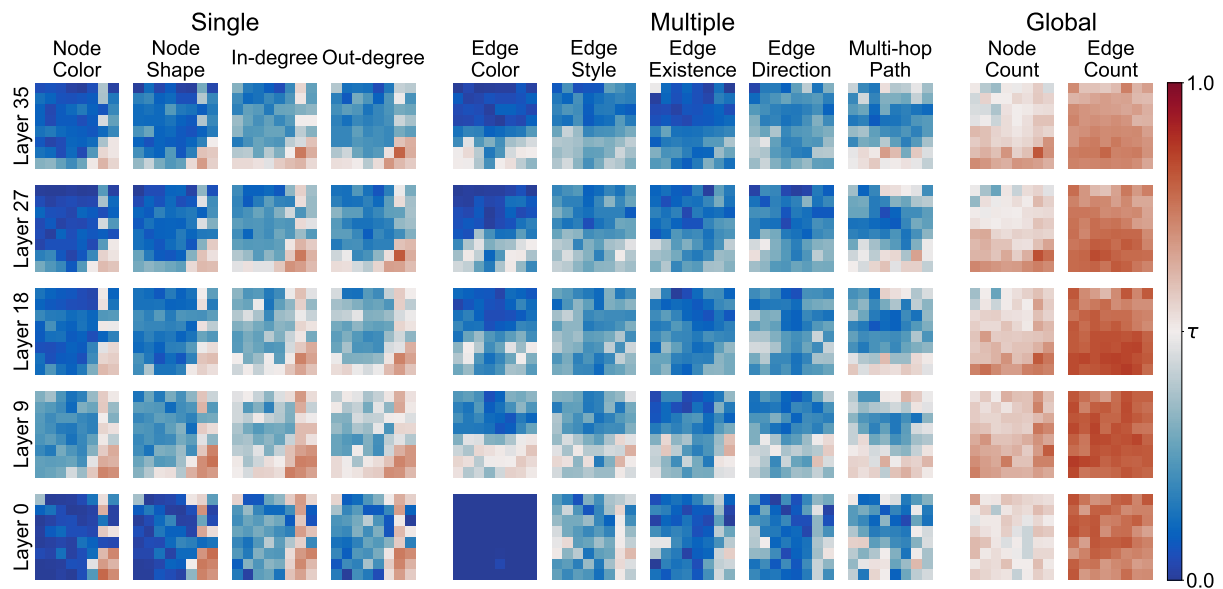


Figure 57: Accuracy for each hidden state in the image-input part of the Qwen3-VL 8B language model (FT w/VL). Each heatmap shows the accuracy of individual hidden states for a specific layer and aspect.

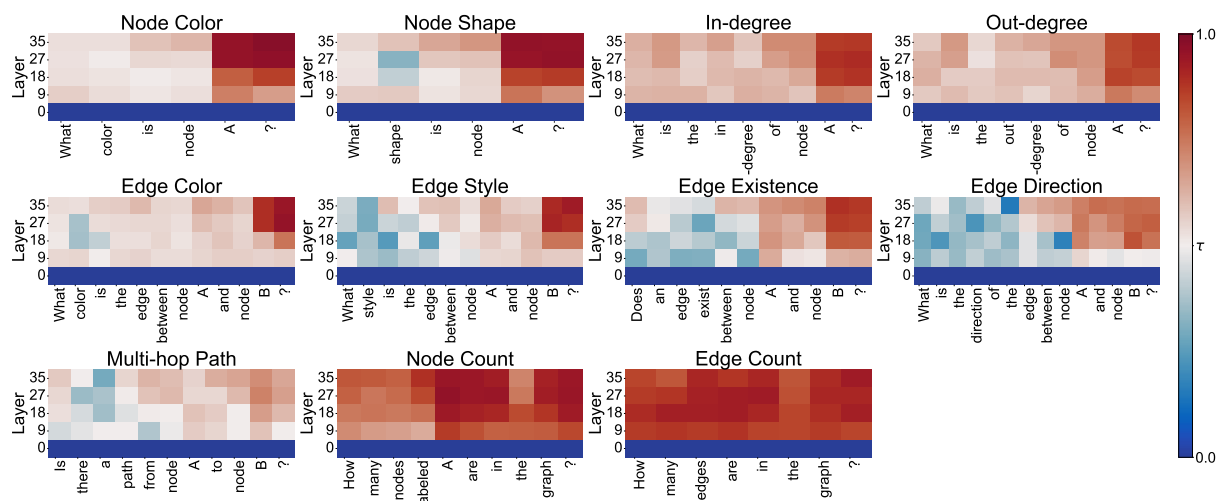


Figure 58: Probing results for text tokens in Qwen3-VL 8B (FT w/VL). The x-axis shows token position in the question, and the y-axis shows layer position.