Direct Visual Grounding by Directing Attention of Visual Tokens

Parsa Esmaeilkhani
Temple University
Philadelphia
parsa.esmaeilkhani@temple.edu

Longin Jan Latecki
Temple University
Philadelphia
latecki@temple.edu

Abstract

Vision Language Models (VLMs) mix visual tokens and text tokens. A puzzling issue is the fact that visual tokens most related to the query receive little to no attention in the final layers of the LLM module of VLMs from the answer tokens, where all tokens are treated equally, in particular, visual and language tokens in the LLM attention layers. This fact may result in wrong answers to visual questions, as our experimental results confirm. It appears that the standard next-token prediction (NTP) loss provides an insufficient signal for directing attention to visual tokens. We hypothesize that a more direct supervision of the attention of visual tokens to corresponding language tokens in the LLM module of VLMs will lead to improved performance on visual tasks. To demonstrate that this is indeed the case, we propose a novel loss function that directly supervises the attention of visual tokens. It directly grounds the answer language tokens in images by directing their attention to the relevant visual tokens. This is achieved by aligning the attention distribution of visual tokens to ground truth attention maps with KL divergence. The ground truth attention maps are obtained from task geometry in synthetic cases or from standard grounding annotations (e.g., bounding boxes or point annotations) in real images, and are used inside the LLM for attention supervision without requiring new labels. The obtained KL attention loss (KLAL) when combined with NTP encourages VLMs to attend to relevant visual tokens while generating answer tokens. This results in notable improvements across geometric tasks, pointing, and referring expression comprehension on both synthetic and real-world data, as demonstrated by our experiments. We also introduce a new dataset to evaluate the line tracing abilities of VLMs. Surprisingly, even commercial VLMs do not perform well on this task.

1. Introduction

Vision-Language Models (VLMs) have achieved remarkable success in various multimodal tasks, including image captioning, visual question answering (VQA), and image-text retrieval. Models like CLIP [40], Flamingo [2], Llava [29], MiniGPT4 [60], and Qwen-VL [3] have demonstrated the efficacy of aligning visual and textual modalities through contrastive and generative pretraining strategies. However, despite their impressive performance on general benchmarks, VLMs often struggle with tasks requiring intricate visual reasoning, such as spatial relations [20, 59], object counting [18, 38], and visual inference [13, 16, 50]. But more alarming is the fact that VLMs struggle with simple, low-level vision tasks like whether two lines intersect or two geometric primitives overlap or are close together [41]. This seems counterintuitive since VLMs excel in complex visual tasks, like a detailed description of image content [8, 14, 43].

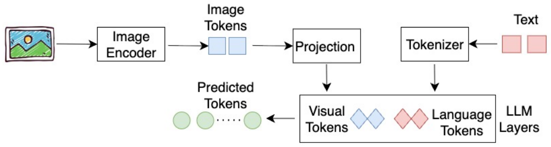


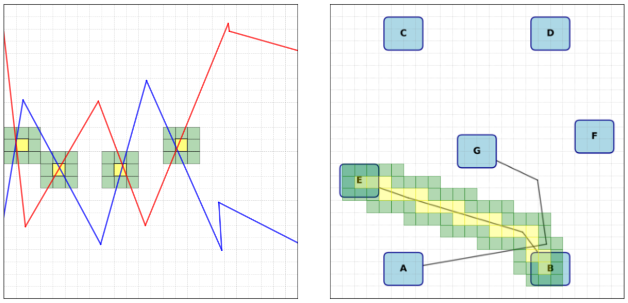
Figure 1. Processing flow in VLMs

Fig. 1 illustrates the processing flow of recent VLMs, like LLava-v1.5 [29] and Qwen2.5-VL [4]. The input image (or images) are first passed through a frozen visual encoder (typically a ViT, CLIP, or DINOv2) and then mapped with a projection layer (alignment module) to a language token embedding space. Finally, the image tokens are passed to an LLM, where they are mixed with special and language tokens. The mixing is performed in the attention layers of the VLM. We call the LLM tokens corresponding to the input image tokens visual tokens.

As visual tokens are transformed through the layers of an LLM, their embeddings change. The goal is to better align them with the embeddings of the language tokens to yield the desired answers. However, there is a danger that the information from visual tokens gets lost among all LLM's tokens after they are processed by the LLM layers. In an extreme case, the visual tokens are ignored and the LLM can even hallucinate an image description for an empty image, as was demonstrated in [30, 49]. Further evidence that visual tokens are often ignored among all LLM tokens is the fact that after removing half of the visual tokens, the VLM performance does not decrease [5]. As pointed out in [13], *The LLM's ability to use its vision representations is a limiting factor in VLM performance.*

So, the problem is the attention mixing of visual and language tokens in the LLM module of VLMs, where all tokens are treated equally, in particular, visual and language tokens. Indeed, answer tokens in state-of-the-art VLMs allocate only a small fraction of their attention to visual tokens (see Section 3.1 of the supplementary material). We call this problem the **problem of attention to visual tokens**. Intuitively, we would expect high attention of tokens representing language concepts to the corresponding regions in images, e.g., the language token "cat" should pay high attention to the tokens representing the image region of the cat. As our experimental results demonstrate, the standard next-token prediction (NTP) loss provides only a weak and indirect signal for directing attention to visual tokens. The issue is that LLMs have difficulty recognizing the special role of visual tokens in answering image-related questions and often rely on language priors instead [51].

The issue persists even if visual grounding is utilized. Visual grounding involves localizing a specific object (or a group of objects) in an image referred to with a natural language expression. This can be done with a bounding box containing the object or with an object mask pointing to the object location, e.g., Pix2Seq [6] and Kosmos-2 [37]. Many visual grounding approaches are able to accurately locate objects, e.g., with bounding boxes, but the question is whether they really know the location of these objects in images, i.e., do they know which visual tokens represent the corresponding object regions (ROIs) in the image. For example, the attention visualization experiments in [59] demonstrate that it is often not the case.



(a) Line Intersection task

(b) Is node E connected to node B?

Figure 2. Visualization of the ground truth map for the attention of answer tokens. Yellow highlights the target patches with the highest mass of the distribution, while green indicates the surrounding neighbors. The other patches have a fixed low value.

Our main hypothesis is that direct supervision of the attention of language tokens to corresponding visual tokens in the LLM module of VLMs will lead to improved performance on visual tasks. To demonstrate that this is indeed the case, we propose a novel training framework that directly supervises the attention maps of VLMs using a combination of Kullback-Leibler (KL) divergence loss in addition to the next-token prediction loss. The proposed KL attention loss (KLAL) directly grounds the answer text tokens in relevant visual tokens by increasing the attention of the answer tokens to the relevant visual tokens. By aligning the model's attention distributions with ground truth (GT) maps with KL divergence, the proposed KLAL strengthens the direct links between language and corresponding visual tokens. In our framework, the ground-truth (GT) attention maps could come from task geometry in the case of synthetic datasets, and from standard grounding annotations (e.g., bounding boxes or point annotations) in the case of real images, projected onto image patches with a smoothing function. This way, no new labels are required, while the maps provide explicit supervision at the token level. Our contribution is not in collecting new annotations, but in introducing a simple way to incorporate these GT attention maps into the LLM's training. Fig. 2 illustrates GT target attention maps for two tasks. For line intersection, the attention highlights patches with intersection points, while for line tracing, it follows the patches along the path connecting the queried nodes. In both cases, KLAL directs the answer tokens to focus on the relevant visual patches, improving both attention and answer quality as shown in Fig. 3.

Our approach is model-agnostic and can be seamlessly integrated into existing VLMs without architectural modifications. It is also simple to implement in that it does not require attaching any task-specific additional heads (e.g., for object localization or segmentation). As our experimental results demonstrate, the explicit visual grounding with KLAL not only improves the quality of VLM answers but also improves the deep embeddings of visual tokens.

Attention visualization has also been used to explain transformer inference, e.g., in [11] for a vision transformer (ViT). As argued in [28], it helps to ensure that LLMs provide correct and consistent information. So, the proposed KLAL can also be helpful in improving the interpretability of VLM responses. Our main contributions are as follows:

- Introduce an auxiliary loss to direct the attention of language answer tokens to visual tokens representing the relevant parts of the image.
- Provide clear experimental evidence that the improvement in the attention to relevant parts of the image contributes to the performance increase on visual tasks. This yields better explainability of the results, which is visible in the presented attention maps.
- Demonstrate that the deep embeddings of visual tokens improve, as tokens with high-norm concentrate in the parts

of the image most relevant to the correct answers.

- Introduce a new dataset to evaluate and finetune the performance of VLMs on a Line Tracing task that is essential for knowing which objects are connected in a given image.
- Evaluate the performance of open source and commercial VLMs on a variety of geometric and visual grounding tasks in order to determine their fundamental visual abilities. The specific tasks are counting the number of intersection points of lines, identifying which objects are connected by lines, pointing to object locations, and resolving referring expressions, where the model must map a natural language description to the correct object in the image.

2. Preliminaries

LLMs generate tokens in an autoregressive fashion using a decoder-only Transformer with causal masking [39]. At each step t, the model attends only to tokens at positions $< t$ and selects the most probable next token from its vocabulary. Then, it updates its weights by minimizing the next-token prediction loss. Our visual attention loss directly supervises the attention weights on visual tokens with respect to other query tokens.

2.1. Language Modeling

During training of VLMs, the entire sequence of input tokens typically follows the structure: text tokens corresponding to the system prompt ($\mathbf{X}_{\text{sys}}$), followed by visual tokens extracted from the image ($\mathbf{X}_V$), and finally text tokens representing the instruction or question provided after the image ($\mathbf{X}_{\text{instruct}}$). These segments are concatenated to form the full input context:

$$\mathbf{X} = \left[\mathbf{X}_{\text{sys}},\ \mathbf{X}_V,\ \mathbf{X}_{\text{instruct}}\right]$$

The image is processed through the pre-trained vision encoder and mapped to visual embedding tokens which are then mapped to language embedding space and mixed with language tokens. All of these tokens are passed through the LLM backbone, which then generates an answer sequence $\mathbf{X}_a = (x_1, \dots, x_{T_a})$ token-by-token under a left-to-right causal mask. The standard next-token prediction loss is:

$$L_{\text{NTP}}(\theta) = -\frac{1}{T_a} \sum_{i=1}^{T_a} \log p_\theta\big(x_i \mid \mathbf{X},\, \mathbf{X}_{a,<i}\big) \tag{1}$$

where θ denotes the model parameters and T_a is the length of the answer sequence. At each step i the model predicts x_i conditioned on the full context $\mathbf{X}$ and the previously generated answer tokens $\mathbf{X}_{a,<i}$.

2.2. Attention Block

The attention matrix of a single attention head h at layer l is given by:

$$\mathbf{A}^{(l,h)} = \operatorname{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{d_k}}\right) \tag{2}$$

where $\mathbf{Q}_h, \mathbf{K}_h \in \mathbb{R}^{n \times d_k}$ are the query and key matrices, and d_k is the head dimension. The softmax normalizes each row of the matrix such that the attention scores sum to 1. As each Transformer layer has H attention heads in parallel, the head outputs are concatenated along the feature dimension to form the layer's multi-head representation. The attention matrices from each layer are reused in our auxiliary visual attention loss which aims at encouraging the model to focus on semantically relevant image regions.

3. Methodology

The main idea of the proposed approach is to extend the training (text, image) pairs by adding target ground truth (GT) attention maps and utilizing a loss function to compare the attention of visual tokens (with respect to the text answer tokens) to the GT attention maps. We treat both attentions as distributions and compare them with KL divergence. So, we call our loss function KLAL. The GT attention maps are constructed automatically, i.e., no manual labeling is necessary. When used in addition to NTP, KLAL helps the model focus more on regions in the image that are decisive for the given task. It does so by increasing the attention on visual tokens corresponding to regions of interest during finetuning.

A first step is to compute the attention distribution over the visual tokens with respect to answer tokens. Let $\mathcal{S} = (\mathbf{X}, \mathbf{X}_a)$ denote one training sample containing the full input context (system + visual + instruction tokens) and the generated answer token sequence $\mathbf{X}_a$. The attention matrices $\mathbf{A}_h$ can be used to compute the attention distribution of the visual tokens to the specific answer tokens. We use the last generated answer, as it captures a summary of the model's focus and reflects how information has been aggregated across the preceding tokens. Let $\alpha^{(l,h)}$ be the submatrix of $\mathbf{A}^{(l,h)}$ representing the attention of all visual tokens to the last answer token for layer l and head h. This submatrix is a slice of the last row of the attention matrix. We normalize $\alpha^{(l,h)}$ to sum to one so that it represents a probability distribution. We average the normalized submatrices $\alpha^{(l,h)}$ across heads to obtain a single distribution:

$$Q_i^{(l)}(\mathcal{S}) = \frac{1}{H} \sum_{h=1}^{H} \alpha_i^{(l,h)}(\mathcal{S}), \quad i \in I_V, \tag{3}$$

where I_V denotes the set of indices corresponding to the visual tokens and l is the index of the LLM layers.

Let $P(\mathcal{S})$ be a GT target distribution over visual tokens I_V, which is defined at the end of this section. Its goal is to increase the model's focus on semantically important visual patches, i.e., the patches that are most relevant for obtaining correct answers.

We introduce a novel attention loss based on KL divergence (KLAL) to bring the predicted distribution $Q^{(l)}(\mathcal{S})$ at each layer l closer to constructed GT distribution $\mathcal{P}(\mathcal{S})$.

$$\begin{aligned}\mathcal{L}_{\text{KLAL}} &= \frac{1}{L}\sum_{l=1}^{L} D_{\text{KL}}\big(P(\mathcal{S}) \,\|\, Q^{(l)}(\mathcal{S})\big) \\ &= \frac{1}{L}\sum_{l=1}^{L}\sum_{i\in I_V} P_i(\mathcal{S}) \log\left(\frac{P_i(\mathcal{S})}{Q_i^{(l)}(\mathcal{S})}\right)\end{aligned} \tag{4}$$

where both distributions are defined over the set of visual tokens I_V and L denotes the total number of layers in LLM.

Finally, we combine the next-token prediction loss with our visual attention loss to optimize the VLM's parameters for both objectives: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NTP}} + \lambda\mathcal{L}_{\text{KLAL}}$. We empirically set λ to 1, which we found effective across all tasks (see Sec. 4 in supplementary material for ablations on λ and head/layer design choices). The target GT distribution $P(\mathcal{S})$ provides guidelines for LLM regarding which vision patches to focus on when answering the question in $\mathcal{S}$. $P(S) : I_V \to [0, 1]$ is defined as

$$P(\mathcal{S}) = \text{Normalize}\Big(\text{Smooth}\big(\mathbf{1}(I_P)\big)\Big), \tag{5}$$

where $P(\mathcal{S})$ is normalized to sum to one; $\text{Smooth}(\cdot)$ is any smoothing function (e.g., Gaussian); $\mathbf{1}(I_P)$ is an indicator equal to 1 for target patches $I_P \subseteq I_V$ and 0 otherwise.

The definition of the set of target patches I_P that induces GT maps is task-specific. For example, in the task of counting the number of intersection points between two polygonal curves, the patches containing the intersection points constitute the target patches. These are marked in yellow in Fig. 2(a), where the green patches are obtained by smoothing. In Fig. 2(b), the yellow patches trace the line connecting nodes E and D. They illustrate the target patches I_P for the answer "Yes" to the query "Is node E connected to node D?".

For real images, the GT map construction is based on existing annotations. For point annotations at object centers, I_P is the patch containing the point with light smoothing around it. For bounding-box annotations describing the referred object, we take the box's center line, vertical or horizontal depending on the box orientation, and mark the patches it traverses. Although I_P varies by task, our pipeline builds it automatically without manual labeling (Sec. 1.5 in supplementary material). For more complex visual tasks, without explicit annotations, GT maps can come from weakly supervised grounding methods [31, 45, 47], providing pseudo-labels at scale for our KLAL in real-world applications.

4. Experimental Evaluation

4.1. Datasets and Tasks

We evaluated our method on five datasets, each designed to test different aspects of spatial/geometric reasoning in VLMs. In the Line Intersection task, the model must count the number of intersections between lines, thereby testing its quantitative geometric understanding. In the Line Tracing task, it must determine whether two nodes are connected in a graph, assessing its capacity for tracing paths through visual structures. The two pointing tasks, one with synthetic images and one with real images, require the model to locate a target object and output its coordinates, evaluating the model's spatial localization and grounding capabilities. We also included a referring expression comprehension (REC) task using the well-established RefCOCO dataset [58], where the model must locate the object described in natural language by predicting the bounding box coordinates that contain it. For all datasets, we used an 80/20 train–test split, except for RefCOCO where we followed its standard splits.

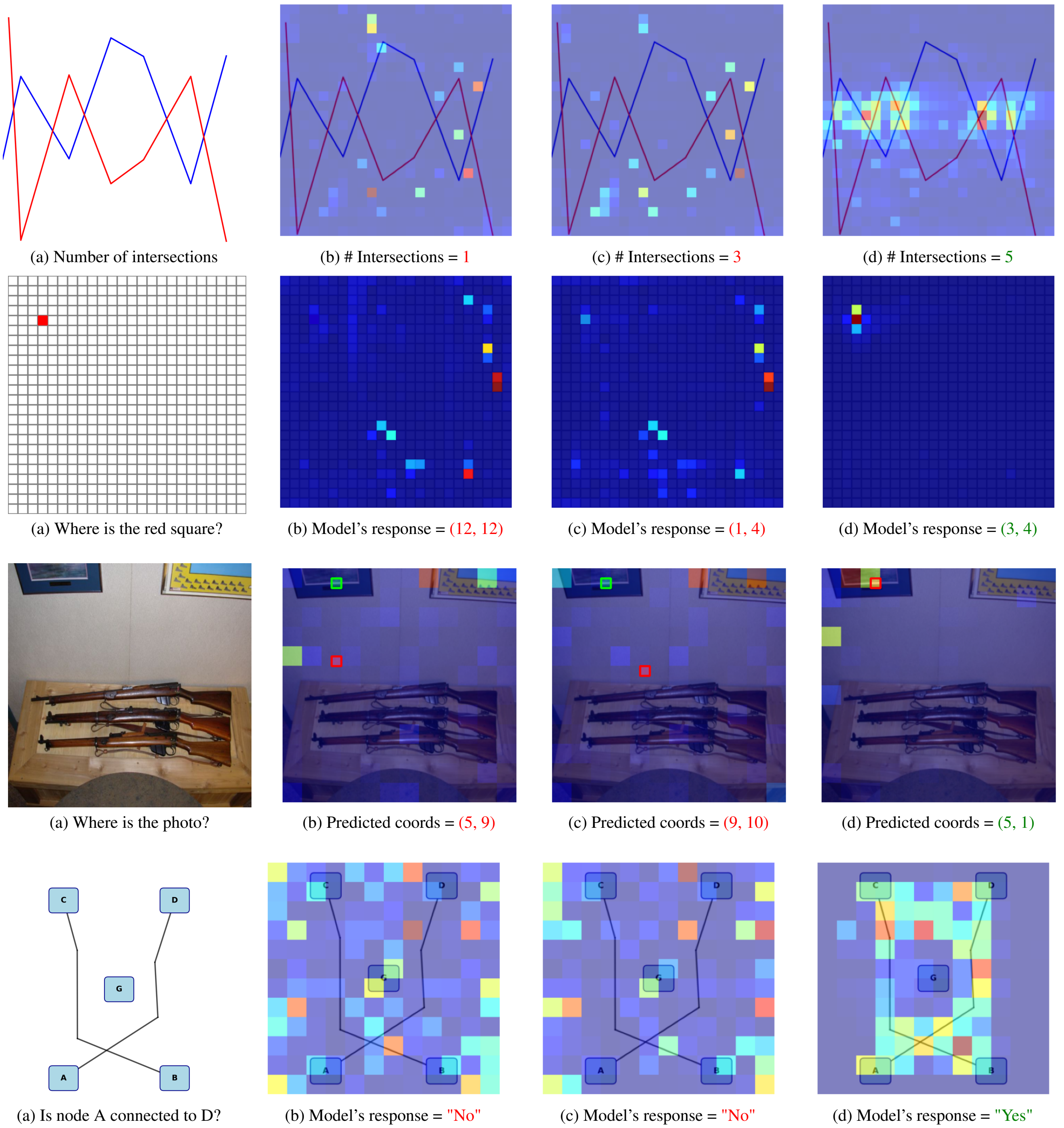
4.1.1. Geometric Datasets

Line Tracing: We constructed a synthetic dataset of graph images with complex topologies to challenge the line tracing abilities of VLMs. As illustrated in the last row of Fig. 3, the graphs consist of a central node and 3 to 6 other labeled nodes positioned around it. Some nodes are connected with polygonal curves with both short and long-range connections. Each graph contains 2 or 3 disjoint edges, ensuring that every node is connected to only one other node and forms distinct, disconnected pairs. So, we ensured the graphs are not cluttered. The dataset includes Yes/No questions asking whether two arbitrary nodes in a given graph are connected, resulting in 1,064 images and 5,360 question-answer pairs, with a balanced distribution of Yes and No answers.

Line Intersection: [41] introduced a geometric visual task to evaluate whether VLMs can count the number of intersections between two piecewise linear curves. The lines are colored in blue and red, with 0, 1, or 2 unique intersections. To increase difficulty, we extended the dataset by generating additional images where the number of intersections ranges from 3 to 5, e.g., see the first row in Fig. 3. So, the number of intersection points ranges from 0 to 5, with 200 images generated for each category, resulting in a total of 1,200 images labeled by the corresponding number of intersections. The accompanying question is: "How many times do the blue and red lines touch each other?"

4.1.2. Pointing Datasets

The pointing task evaluates the ability of VLMs to localize a target object We consider two variants: **Grid Patch:** Each image is divided into a 24×24 grid with gray overlay lines, and one target cell is highlighted in red. The prompt asks the model to output the grid coordinates of the red patch (see the second row of Fig. 3). **PixMo-Points:** A subset of 1,500 real-world images from PixMo-Points [12], spanning 150 object categories. Each sample provides a short textual prompt and a human-annotated point marking the object center. The model must output the corresponding (x, y) coordinates to the center of the referred object (see the third row of Fig. 3).



(a) Number of intersections (b) # Intersections = 1 (c) # Intersections = 3 (d) # Intersections = 5

(a) Where is the red square? (b) Model's response = (12, 12) (c) Model's response = (1, 4) (d) Model's response = (3, 4)

(a) Where is the photo? (b) Predicted coords = (5, 9) (c) Predicted coords = (9, 10) (d) Predicted coords = (5, 1)

(a) Is node A connected to D? (b) Model's response = "No" (c) Model's response = "No" (d) Model's response = "Yes"

Figure 3. Attention maps show the attention of the last answer token to visual tokens. The results in the first two rows are from LLava-v1.5, and the last two rows show results from Qwen2.5-VL. First column shows input images and tasks. Second column shows the attention maps of out-of-the-box models. Third column shows the NTP-finetuned model. Fourth column shows the attention maps of models finetuned with the proposed NTP+KLAL. The red text indicates wrong answers and the green the correct ones. In the third row, the green box indicates the ground truth patch, and the red box denotes the predicted patch. Not only are the answers produced by NTP + KLAL correct, but the attention maps are also dramatically improved and become much more interpretable.

4.2. Results and Analysis

On geometric and pointing tasks (Line Intersection, Line Tracing, and object pointing), we evaluated the LLava-v1.5-7B and Qwen2.5-VL-7B-Instruct models under three configurations: (1) base model (out-of-the-box), (2) finetuned with the next token prediction (NTP), and (3) finetuned with NTP combined with the proposed KLAL (starting from base model check point). For the REC task on RefCOCO, we focused on Qwen2.5-VL-7B-Instruct, the stronger of the two models, and evaluated it under the same configurations.

We compared the two models against both open-source and commercial state-of-the-art VLMs. Among the commercial baselines, we included GPT-4o [17] and Gemini-2.0 Flash [9]. For open-source baselines, we selected three models. GLaMM-FS-7B [42], specialized for grounding with region-level annotations, was evaluated only on the Line Intersection and Line Tracing tasks. Molmo-7B-D [12], trained with coordinate-level supervision including the PixMo-Points (superset of our dataset), and InstructBLIP-Vicuna-7B [10], a general-purpose instruction-tuned model, were both evaluated across all geometric and pointing tasks.

Task-specific accuracy metrics were used for evaluation. For the Line Intersection task, a response was considered correct if it exactly matched the ground truth number, which ranged from 0 to 5. For the Line Tracing task, a response was deemed correct if it matched the ground truth answer, either Yes or No. For the two pointing tasks (Grid Patch and PixMo-Points), a prediction was considered correct if the predicted coordinates were within 3 units of Euclidean distance from the ground truth coordinates; otherwise, it was considered incorrect. Final accuracy was computed as the ratio of correct predictions to the total number of test samples for each dataset.

Table 1. Accuracy on Line Intersection and Line Tracing tasks

Method	Line Intersection	Line Tracing
LLava-v1.5-7B		
Base Model	27.91%	50.00%
NTP	49.11%	46.76%
NTP + KLAL	55.68%	53.52%
Qwen2.5-VL-7B-Instr.		
Base Model	47.62%	49.62%
NTP	62.64%	53.82%
NTP + KLAL	**70.23%**	**62.21%**
SOTA		
Molmo-7B-D	41.07%	49.51%
GLaMM-FS-7B	27.50%	39.03%
InstructBLIP	36.67%	46.23%
GPT-4o	42.12%	55.34%
Gemini-2.0 Flash	56.41%	59.25%

Table 1 shows the out-of-the-box LLava-v1.5 and Qwen2.5-VL models performed above chance (16.7%) on the Line Intersection task but struggled on the more challenging Line Tracing dataset, where they achieved near-random accuracy of 50%. Finetuning with NTP led improved Qwen2.5-VL notably on both tasks. In contrast, LLava-v1.5 showed minimal gain on Line Tracing, likely due to its strong bias toward answering Yes. Adding KLAL to NTP resulted in significant gains: LLava-v1.5 improved by 6.6% on Line Intersection and 6.8% on Line Tracing, while Qwen2.5-VL improved by 7.6% and 8.4%, respectively. Although Gemini-2.0 Flash outperformed others, it still lagged behind Qwen2.5-VL with NTP + KLAL. This is notable since Gemini-2.0 Flash and GPT-4o excel on more complex tasks.

Table 2. Accuracy on Grid Patch and PixMo-Points datasets.

Method	Grid Patch	PixMo-Points
LLava-v1.5-7B		
Base Model	10.42%	5.84%
NTP	20.41%	9.49%
NTP + KLAL	40.82%	16.52%
Qwen2.5-VL-7B-Instr.		
Base Model	6.12%	16.79%
NTP	28.57%	26.28%
NTP + KLAL	**44.90%**	**35.77%**
SOTA		
Molmo-7B-D	18.37%	21.53%
InstructBLIP	6.44%	8.31%
GPT-4o	38.78%	19.70%
Gemini-2.0 Flash	40.82%	18.98%

Table 2 shows results on the Grid Patch and PixMo-Points pointing tasks. In their base forms, both LLava-v1.5 and Qwen2.5-VL exhibited low accuracy, particularly on PixMo-Points (chance level is below 1%, e.g., in Grid Patch a $24 \times 24 = 576$ grid gives random accuracy $1/576 \approx 0.17\%$). The NTP + KLAL combination led to substantial improvements over NTP alone. LLava-v1.5 improved by 20.4% on Grid Patch and 7.0% on PixMo-Points, while Qwen2.5-VL gained 16.3% and 9.5%, respectively. Molmo-7B-D outperformed all commercial SOTA models on PixMo-Points, likely due to its finetuning in coordinate-level supervision on images from the same dataset. (Note that Molmo-7B-D was evaluated in its native pixel-level coordinates; we then linearly mapped its predictions into our grid coordinate system.) Across both datasets, Qwen2.5-VL consistently outperformed LLava-v1.5 and surpassed all baseline models when finetuned with NTP + KLAL.

To assess generalization, we conducted cross-dataset transfer experiments between PixMo-Points and Grid Patch to examine whether learning on one dataset can positively transfer to the other in Table 3. Interestingly, when transferring from Grid Patch to PixMo-Points, NTP performed worse than the base model, indicating that NTP alone cannot generalize to unseen datasets without large-scale training data. However, KLAL + NTP enabled Qwen2.5-VL to transfer effectively, achieving much higher accuracy and, in some cases, reaching performance comparable to task-specialized baselines such as Molmo.

Table 3. Transfer accuracy between Pixmo-Points (P) and Grid Patch (G) datasets. P → G: trained on P, evaluated on G. G → P: trained on G, evaluated on P.

Method	P → G	G → P
Qwen2.5-VL-7B-Instr.		
Base Model	6.12%	16.79%
NTP	11.16%	16.05%
NTP + KLAL	**19.29%**	**21.98%**

To see whether our method scales to large, well-established visual grounding datasets, we evaluated KLAL on RefCOCO for the REC task. As shown in Table 4, while the base model and NTP already achieve strong accuracy, finetuning with KLAL + NTP yields further improvements across validation, testA, and testB splits. These results demonstrate that KLAL + NTP provides measurable gains even in high-performing settings and highlight its potential for broader application to real-world grounding benchmarks.

Table 4. Accuracy (IoU 0.5) of models finetuned on the RefCOCO training set and evaluated on RefCOCO validation and test splits.

Method	RefCOCO		
	val	testA	testB
Qwen2.5-VL-7B-Instr.			
Base Model	90.05%	93.75%	86.70%
NTP	90.65%	94.05%	86.90%
NTP + KLAL	**91.45%**	**94.65%**	**87.50%**

4.3. Visual Attention

Across all datasets, KLAL with NTP improved accuracy, and enhanced attention maps as shown in Fig. 3. We demonstrate this statistically in Fig. 4 on the task of counting the number of intersection points, where the target visual patch tokens are those containing the intersection points. The bar plots illustrate the ratio of the average attention of visual target tokens to the average attention of all visual tokens w.r.t the answer token. The values below one for the base models and the NTP finetuned models indicate that the influence of the target tokens on the answer is smaller than the influence of other visual tokens. **Only after finetuning with the proposed KLAL, we see that the average target token has higher influence on the answer than the average visual token.**

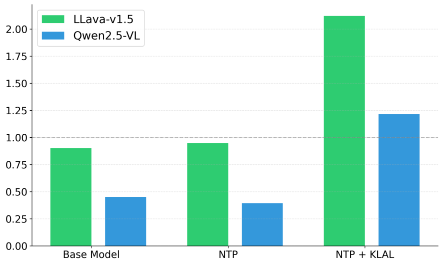


Figure 4. The bar plots illustrate the ratio of the average attention of target tokens w.r.t the answer token to the average attention of all visual tokens computed over all Line Intersection test images.

Interestingly, our attention-focused loss function also positively influences the deep embeddings of visual tokens. In the second row of Fig. 5, we replaced the attention of visual tokens to the answer token with the norm of the embeddings of visual tokens obtained from the last attention layer. As we can see, the proposed NTP + KLAL significantly increased the concentration of high-norm tokens at the line intersection points. This correlates with the attention to the answer token shown in the first row. Since high-norm tokens are more likely to be attended to, they are more likely to influence the answer. The higher concentration of high-norm visual tokens at the target regions is also confirmed by the results in Table 5. It demonstrates that using NTP alone has little impact on embedding magnitude, whereas adding KLAL increases the average norm by 6% for Qwen2.5-VL and by 19% for LLava-v1.5. This confirms that KLAL not only directs model attention but also strengthens the internal feature representations of the relevant visual tokens.

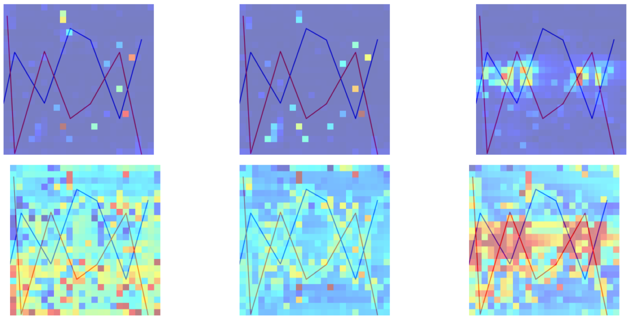
Model	NTP	KLAL+NTP
LLava-v1.5	1.05	**1.19**
Qwen2.5-VL	0.96	**1.06**

Table 5. Ratio of average embedding norm of target visual tokens after finetuning to the base model on the line intersection test set.

To summarize our experimental results clearly demonstrate our main hypothesis that direct supervision of the attention of language tokens to corresponding visual tokens in the LLM module of VLMs leads to improved performance on visual tasks. We demonstrated this on four simple but fundamental visual tasks as well as on the real-world RefCOCO benchmark. The details of the finetuning process are mentioned in Section 4 of the supplementary material.

5. Related Work

Large-scale VLMs such as CLIP [40] and Flamingo [2] align image and text via contrastive learning or frozen encoders with autoregressive LMs. We focus on instruction-tuned, autoregressive VLMs [3, 26, 29, 33, 60] that generate text



(a) Out-of-the-box (b) NTP (c) NTP + KLAL

Figure 5. The first row shows attention maps from the last output token of LLava-v1.5, while the second row presents a visualization derived from the magnitudes of visual token embeddings. Both types of visualizations reveal similar patterns, as enhanced attention through NTP + KLAL corresponds to increased focus of high-norm tokens on the target regions.

or special tokens as output. These VLMs have achieved impressive results on tasks ranging from image captioning [26, 52], image–text retrieval [40, 57], to visual question answering [2]. The successes largely stem from training on massive image–text corpora, which improve overall alignment between modalities. Yet scale alone does not guarantee faithful grounding: VLMs regularly hallucinate plausible but incorrect details when no image is provided [30] and can ignore up to half of the visual tokens without degrading performance [5], a symptom often traced to so-called "attention sinks" that absorb excessive weight despite low semantic relevance [21]. Moreover, existing benchmarks may not sufficiently test vision-centric abilities of VLMs [49, 50]. This shows that while larger and more diverse image–text datasets may boost benchmark scores, they do not ensure that generated text attends to the correct image regions.

To address these gaps, one approach integrates VLMs with visual grounding, mapping expressions to image regions by finetuning on grounding datasets and adding localization modules. Examples include LISA [24] with a '[SEG]' token and mask decoder, F-LMM [55] with lightweight mask refinement, and GLAMM [42] with an added grounding encoder and pixel decoder.

Another approach modifies attention only at inference, e.g., boosting visual token weights [30], redistributing mass from sink tokens [21], or pruning redundant tokens [5]. These reduce hallucination but leave representations unchanged and underperform finetuned models [22].

A third line of work supervises cross-modal attention during training. Inspired by attention–rationale alignment in NLP [53] and Grad-CAM in vision [44], recent VQA and grounding works add auxiliary losses that guide tokens toward annotated or automatically generated maps [36, 59]. Other approaches include attention regularization [32] and attention priors or multi-grained grounding for more relevant visual focus [15, 25]. These typically rely on object detectors, saliency maps, or external grounding models. Closely related to our goal, FastRM [46] and FiVL [1] analyze and encourage vision–language alignment by examining or leveraging attention patterns, with an emphasis on explainability and evaluation. Our KL Attention Loss (KLAL) follows this line of work; however, it differs by automatically deriving task-specific GT attention maps from underlying task properties or provided annotations and aligning visual-to-answer attention via layer-wise KL divergence.

Unlike approaches such as Pix2Seq [6, 37], which train LLMs to output bounding box coordinates (transferring visual knowledge into text), we directly link language tokens to relevant visual tokens. Our focus is not global embedding alignment as in CLIP [40], UNITER [7], or ViLBERT [34], but token-level grounding of answer tokens. Experiments show this is difficult with the standard NTP loss, as LLM attention layers seem to be unable to properly connect language and visual tokens when guided only by NTP.

While VLMs excel at general image understanding, they often struggle with geometric and relational reasoning on abstract structures such as line perception and connectivity [27, 41]. To address this, we introduce a synthetic Line Tracing dataset (Fig. 3) testing node connectivity via visual paths in complex graphs. Unlike chart-QA datasets [19, 35] or broader geometric benchmarks [23], it targets visual path perception, providing a benchmark for assessing ability to follow polygonal curves and infer topological connectivity.

6. Discussion and Conclusions

The proposed approach is inspired by direct supervision of visual attention in early childhood learning. As is well-established in psychology [48, 54], direct supervision during tasks like object identification in pictures is not just helpful, but it is crucial for child development. It transforms passive exposure into active learning. In particular, linking visual attention and word learning is essential [56]: *"moments in which a single object was visually dominant. If parents named the object during these moments of bottom-up selectivity, later forced-choice tests showed that infants learned the name, but did not when naming occurred during a less visually selective moment."* Therefore, we propose a novel loss function that directly supervises the attention of language tokens to corresponding vision tokens, and demonstrate experimentally its benefits.

Our approach is model-agnostic, integrates into existing VLMs without architectural changes, and requires no task-specific heads (e.g., for localization or segmentation). While we focused on visual tasks fundamental to spatial understanding, the proposed approach can be extended to other complex visual tasks where no annotations are provided.

7. Acknowledgment

This work was supported by NSF grant IIS-2331768.

References

[1] Estelle Aflalo, Gabriela Ben Melech Stan, Tiep Le, Man Luo, Shachar Rosenman, Sayak Paul, Shao-Yen Tseng, and Vasudev Lal. Fivl: A framework for improved vision-language alignment through the lens of training, evaluation and explainability, 2025. 8

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 23716–23736, 2022. 1, 7, 8

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. 1, 7

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1

[5] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024. 2, 8

[6] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *ICLR*, 2022. 2, 8

[7] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 8

[8] Kanzhi Cheng, Wenpo Song, Jiaxin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. Caparena: Benchmarking and analyzing detailed image captioning in the llm era. *arXiv preprint arXiv:2503.12329*, 2025. 1

[9] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 6

[10] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 6

[11] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 2

[12] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104, 2025. 4, 6

[13] Stephanie Fu, Tyler Bonnen, Devin Guillory, and Trevor Darrell. Hidden in plain sight: Vlms overlook their visual representations. *arXiv preprint arXiv:2506.08008*, 2025. 1, 2

[14] Roopal Garg, Andrea Burns, Burcu Karagol Ayan, Yonatan Bitton, Ceslee Montgomery, Yasumasa Onoe, Andrew Bunner, Ranjay Krishna, Jason Baldridge, and Radu Soricut. Imageinwords: Unlocking hyper-detailed image descriptions. *arXiv preprint arXiv:2405.02793*, 2024. 1

[15] Pingping Huang, Jianhui Huang, Yuqing Guo, Min Qiao, and Yong Zhu. Multi-grained attention with object-level grounding for visual question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3595–3600, 2019. 8

[16] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019. 1

[17] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 6

[18] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 1

[19] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017. 8

[20] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's" up" with vision-language models? investigating their struggle with spatial reasoning. *arXiv preprint arXiv:2310.19785*, 2023. 1

[21] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *ICLR*, 2025. 8

[22] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9339–9350, 2025. 8

[23] Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. Geomverse: A systematic evaluation of large models for geometric reasoning. *arXiv preprint arXiv:2312.12241*, 2023. 8

[24] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024. 8

[25] Thao Minh Le, Vuong Le, Sunil Gupta, Svetha Venkatesh, and Truyen Tran. Guiding visual question answering with attention priors. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 4381–4390, 2023. 8

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 7, 8

[27] Ruizhou Li and Haiyun Jiang. Graph-to-vision: Multi-graph understanding and reasoning using vision-language models. *arXiv preprint arXiv:2503.21435*, 2025. 8

[28] Zhenru Lin, Jiawen Tao, Yang Yuan, and Andrew Chi-Chih Yao. Existing llms are not self-consistent for simple tasks, 2025. 2

[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 1, 7

[30] Shi Liu, Kecheng Zheng, and Wei Chen. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision*, pages 125–140. Springer, 2024. 1, 8

[31] Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Dechao Meng, and Qingming Huang. Adaptive reconstruction network for weakly supervised referring expression grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2611–2620, 2019. 4

[32] Yibing Liu, Yangyang Guo, Jianhua Yin, Xuemeng Song, Weifeng Liu, Liqiang Nie, and Min Zhang. Answer questions with right image regions: A visual attention regularization approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(4):1–18, 2022. 8

[33] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024. 7

[34] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 8

[35] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 8

[36] Maria Parelli, Dimitrios Mallis, Markos Diomataris, and Vassilis Pitsikalis. Interpretable visual question answering via reasoning supervision. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2525–2529. IEEE, 2023. 8

[37] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. In *ICLR*, 2023. 2, 8

[38] Muhammad Fetrat Qharabagh, Mohammadreza Ghofrani, and Kimon Fountoulakis. Lvlm-count: Enhancing the counting ability of large vision-language models. *arXiv preprint arXiv:2412.00686*, 2024. 1

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 3

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1, 7, 8

[41] Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision*, pages 18–34, 2024. 1, 4, 8

[42] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018, 2024. 6, 8

[43] Noam Rotstein, David Bensaïd, Shaked Brody, Roy Ganz, and Ron Kimmel. Fusecap: Leveraging large language models for enriched fused image captions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5689–5700, 2024. 1

[44] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 8

[45] Tal Shaharabany and Lior Wolf. Similarity maps for self-training weakly-supervised phrase grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6925–6934, 2023. 4

[46] Gabriela Ben-Melech Stan, Estelle Aflalo, Man Luo, Shachar Rosenman, Tiep Le, Sayak Paul, Shao-Yen Tseng, and Vasudev Lal. Fastrm: An efficient and automatic explainability framework for multimodal generative models, 2025. 8

[47] Robin Strudel, Ivan Laptev, and Cordelia Schmid. Weakly-supervised segmentation of referring expressions, 2022. 4

[48] Michael Tomasello and Michael J Farrar. Joint attention and early language. *Child Development*, 57(6):1454–1463, 1986. 8

[49] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024. 1, 8

[50] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024. 1, 8

[51] Chenxi Wang, Xiang Chen, Ningyu Zhang, Bozhong Tian, Haoming Xu, Shumin Deng, and Huajun Chen. Mllm can see? dynamic correction decoding for hallucination mitigation. *arXiv preprint arXiv:2410.11779*, 2024. 2

[52] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021. 8

[53] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 11–20, 2019. 8

[54] David Wood, Jerome S Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100, 1976. 8

[55] Size Wu, Sheng Jin, Wenwei Zhang, Lumin Xu, Wentao Liu, Wei Li, and Chen Change Loy. F-lmm: Grounding frozen large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24710–24721, 2025. 8

[56] Chen Yu and Linda B Smith. Embodied attention and word learning by toddlers. *Cognition*, 125(2):244–262, 2012. 8

[57] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 8

[58] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pages 69–85. Springer, 2016. 4

[59] Yunan Zeng, Yan Huang, Jinjin Zhang, Zequn Jie, Zhenhua Chai, and Liang Wang. Investigating compositional challenges in vision-language models for visual grounding. In *CVPR*, 2024. 1, 2, 8

[60] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 7