Supplementary Material for
Direct Visual Grounding by Directing Attention of Visual Tokens

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

Longin Jan Latecki
Temple University
Philadelphia, USA
latecki@temple.edu

1. Datasets

Below, we provide more information on dataset the creation steps and ground truth attention maps.

1.1. Line Tracing

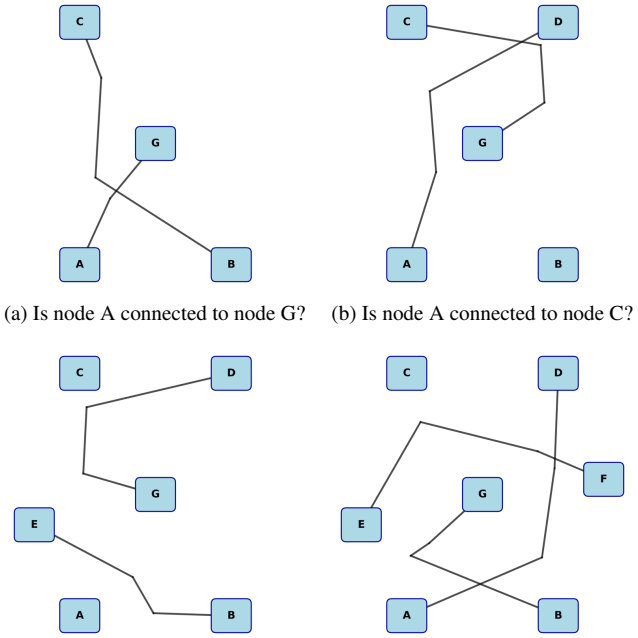
As shown in Fig.1, Graph nodes were drawn as labeled boxes (labels A–F) placed at fixed grid positions, with edges selected by sampling node pairs whose Euclidean distance exceeded a minimum threshold and then rendered as colored polygonal curves (line width set for clear visibility). Each vertex of these curves received a small random jitter to avoid overly regular layouts. To increase diversity without changing connectivity, every unique graph topology was rendered under simple augmentations (horizontal flip, vertical flip, and 90° rotation), producing multiple visual variants per structure.

1.2. Line Intersection

Each image was created by first laying out a 12 by 12 grid of potential vertex positions, then constructing two piecewise linear curves whose complexity depends on the target intersection count. For zero to two intersections, each curve had three points at the left, middle, and right positions, while for three to five intersections additional randomly jittered control points were added. The vertical positions of these points were sampled from predefined bands (low, mid, high) to steer how often the curves cross, and we computed exact line segment intersections, discarding and retrying any pair that did not match the desired count. Once a valid pair of red and blue curves was found, it was rendered as an image and paired with its correct intersection label. Figure 2 shows one representative image for each intersection count (0–5) to illustrate the variety in our dataset.

1.3. Grid Patch

Grid Patch is a synthetic dataset in which each image is rendered on a white background and divided into a 24×24 grid with gray overlay lines. The target patch, occupying one grid cell, is highlighted in red. The prompt asks for the grid coordinates of the red patch in the form (x, y), after defining the coordinate system: (0, 0) as the top-left and (23, 23) as the bottom-right cell. Specifically, x and y correspond to the column and row indices within the grid. The dataset contains 576 synthetic images, each paired with a question, with individual grid cells matching image patches and measuring 14×14 pixels.



(a) Is node A connected to node G? (b) Is node A connected to node C?

(c) Is node B connected to node E? (d) Is node E connected to node D?

Figure 1. Sample input images from the synthetically generated Line Tracing dataset, each paired with a Yes/No connectivity question.

1.4. PixMo-Points Subset

The PixMo-Points dataset is a large-scale resource for visual grounding based on real-world images. Human annotators

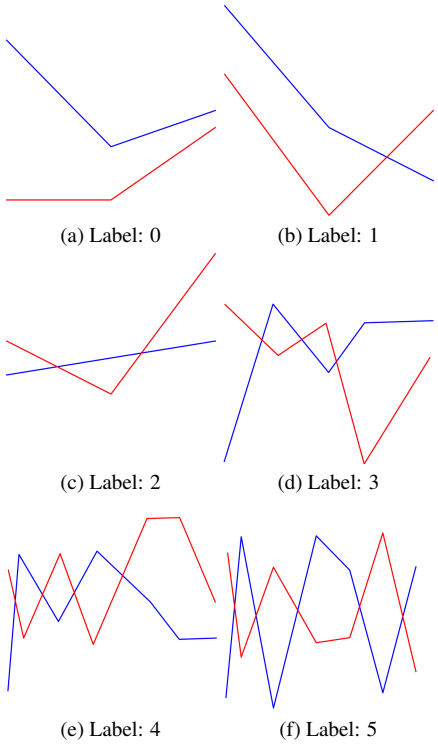


Figure 2. Sample images from the synthetically generated Line Intersection dataset with corresponding intersection counts between the red and blue lines (possible values: 0–5).

identified objects, provided textual descriptions, and pointed to the center of each object instance within the images. For our experiments, we used a subset of 1,500 images covering 150 randomly selected object categories. The task involves identifying the center of the referred object and predicting its coordinates (x, y), following the same format as used in the Grid Patch dataset.

1.5. Target Patch Selection

In this section, we explain how we select target patches for each dataset. Once we have target patches, we can generate ground truth attention maps as explained in the methodology section.

Line Tracing. We sampled points uniformly along each edge segment and assigned each sample to the nearest grid patch by comparing its coordinates to patch-center positions. We also ensured that the patches containing the edge's endpoints were always included.

Line Intersection. We treated each polygonal curve as a set of line segments, detected their crossings using orientation tests and exact intersection formulas, computed the intersection coordinates, and removed any repeated points so that each intersection is only counted once. The patches covering these unique points are the targets.

Grid Patch and PixMo-Points. We took the provided ground truth coordinates and directly mapped them to patch indices in the grid, using each coordinate pair as a target visual token.

RefCOCO. To find the centerline within the bounding box, we first check whether the box is taller or wider. The smaller of the two dimensions (width or height) is used as a reference to keep the axis well within the bounding box. If the box is taller, a vertical line is drawn, centered horizontally using an x-offset from the left edge. If the box is wider, a horizontal line is drawn, centered vertically using a y-offset from the top edge. In both cases, the line is shortened slightly on each end to avoid touching the edges directly. The target patches are defined as those traversed by this centerline.

2. Patch and PixMo-Points Performance

Table 1 reports results of all models on the Grid Patch and PixMo-Points datasets under base, NTP, and NTP+KLAL configurations. Here, in addition to accuracy (where predictions within three grid units of the ground truth count are correct), we also include the median distance errors.

3. Attention Analysis

This section examines how visual attention is distributed across the LLM's layers and how it transforms when projected from the vision encoder into the model's language embedding space.

3.1. Layer-wise Visual Attention

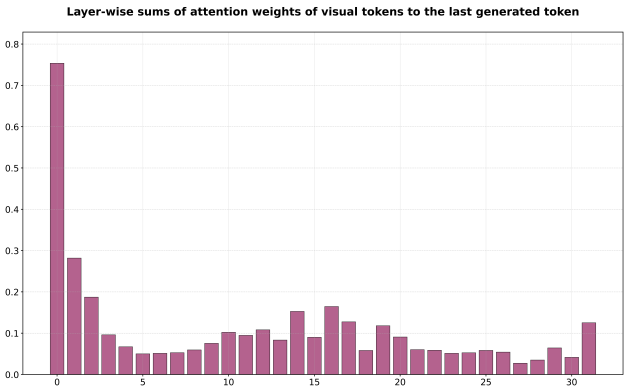
Fig. 3 compares the per-layer sum of attention weights to visual tokens for the LLava-v1.5 and Qwen2.5-VL base models. Across layers, the attention of the last answer token to visual tokens comprises about 10.93%(SD ± 12.66%) of the total attention to all tokens in LLava-v1.5, which is substantially higher than Qwen-2.5VL's average of 2.33% (SD ± 1.44%).

3.2. Vision Encoder vs. LLM Decoder Attention

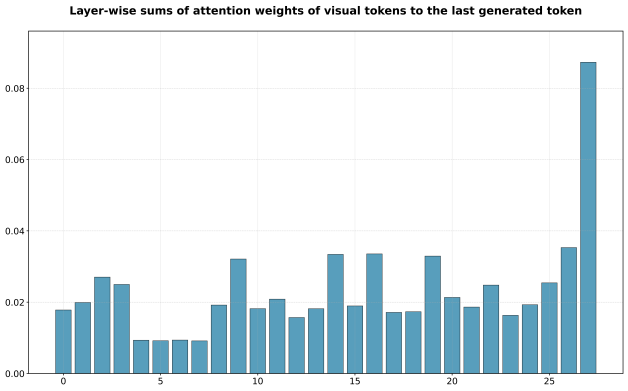
Fig. 4 shows that the Vision Encoder maintains partial focus on the lines, whereas the LLM loses that focus in LLava-v1.5.

Table 1. Performance comparison of models on Patch and Pixmo datasets

Method	Patch Dataset		Pixmo Dataset	
	Median Distance	**Accuracy**	**Median Distance**	**Accuracy**
LLaVA-v1.5-7B				
Base Model	8.06	10.42%	13.60	5.84%
NTP	5.39	20.41%	9.43	9.49%
NTP + KLAL	**3.61**	40.82%	7.14	16.52%
Qwen2.5-VL-7B-Instruct				
Base Model	11.05	6.12%	7.07	16.79%
NTP	6.08	28.57%	6.08	26.28%
NTP + KLAL	4.00	**44.90%**	**5.00**	**35.77%**
SOTA				
Molmo-7B-D	6.71	18.37%	7.13	21.53%
GPT-4o	4.26	38.78%	7.07	19.70%
Gemini-2.0 Flash	4.16	40.82%	7.21	18.98%

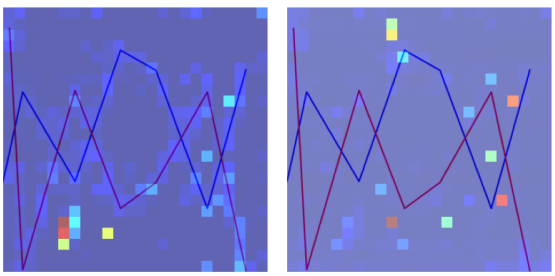


(a) LLava-v1.5-7B Base Model



(b) Qwen2.5-VL-7B-Instruct Base Model

Figure 3. Layer-wise visual attention analysis (sum of attention weights to visual tokens) for LLava-v1.5 and Qwen-2.5VL.



(a) Vision Encoder (b) LLM Decoder

4. Impact of Layer, Head, and Weighting Choices

We conduct ablation studies to assess the design choices in our KLAL loss. Table 2 compares strategies for aggregating attention signals across layers and heads. Applying the loss to each layer, with attention matrices averaged across heads, achieves the highest accuracy, whereas restricting supervision to specific layers or heads leads to a performance drop. For selecting the top 5 heads, we considered two measures: (i) the sum of attention weights assigned to visual tokens, and (ii) the sum of attention weights assigned to the top 20 patches in the ground truth attention maps. We then ranked heads according to both measures and selected the five that overlapped across the two rankings.

Table 3 examines the influence of the weighting hyperparameter λ in the combined loss $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NTP}} + \lambda \mathcal{L}_{\text{KLAL}}$. Performance peaks at $\lambda = 1.0$, indicating that balanced contributions from both terms are most effective.

Table 2. Ablation of design choices for attention supervision loss on Line Intersection for *Qwen2.5-VL-7B-Instr.* with **NTP + KLAL**. Unless otherwise noted, results are averaged across heads within each layer.

Aggregation strategy	**Accuracy (%)**
All layers	**70.23%**
Top 4 layers	68.73%
Middle layers (13–16)	63.00%
Last layer only	65.43%
Each head independently (all layers)	67.26%
Top 5 heads only (all layers)	64.84%

Table 3. Effect of varying the hyperparameter λ in the combined loss ($\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NTP}} + \lambda\mathcal{L}_{\text{KLAL}}$) on Line Intersection for *Qwen2.5-VL-7B-Instr.*

λ	**Accuracy (%)**
0.25	67.03%
0.50	68.98%
0.75	69.96%
1.00	**70.23%**
1.25	68.86%
1.50	67.77%

5. Implementation Configuration

We finetuned the LLava-v1.5-7B and Qwen2.5-VL-7B-Instruct VLMs using DeepSpeed on two NVIDIA A6000 GPUs (40 GB VRAM each). The vision encoder remained frozen except for a trainable projection layer, while Low-Rank Adaptation (LoRA) was applied to the language component: LLava-v1.5 employed a LoRA rank of 16 (after rank 8 proved insufficient), and Qwen2.5-VL used a LoRA rank of 8. Training was conducted with a per-GPU batch size of 32 (effective global batch size = 64). Optimization followed a cosine decay learning rate schedule (initial LR = 5×10^{-5}) with a linear warm-up over the first 3% of total training steps. For the REC task on RefCOCO, we finetuned Qwen2.5-VL-7B-Instruct for 400 steps under the same optimization setup described above.

For the commercial VLM baselines (GPT-4o and Gemini-2.0 Flash), we queried the public APIs using their default system prompts. We performed greedy decoding, did not apply top-k or top-p sampling, and left the temperature parameter at its default setting, ensuring that outputs were deterministic and directly comparable to those from our finetuned open-source models.