THE DUAL MECHANISMS OF SPATIAL VARIABLE BINDING IN VISION–LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many multimodal tasks, such as image captioning and visual question answering, require vision–language models (VLMs) to bind objects with their properties and spatial relations. Yet it remains unclear where and how such associations are computed within VLMs. In this work, we show that VLMs rely on two concurrent mechanisms to represent spatial variable binding. In the language model backbone, intermediate layers represent content-independent spatial relations on top of visual tokens corresponding to objects. However, this mechanism plays only a secondary role in shaping model predictions. Instead, the dominant source of spatial information originates in the vision encoder, whose representations encode the layout of objects and are directly exploited by the language model backbone. Notably, this spatial signal is distributed globally across visual tokens, extending beyond object regions into surrounding background areas. We validate the generalization of our findings to complex natural images from the COCO dataset, where globally amplifying the vision-derived spatial representations across all image tokens corrects spatial variable binding failures across models of various sizes. Together, our results clarify how spatial variable binding is computed within VLMs and highlight the central role of vision encoders in enabling it.

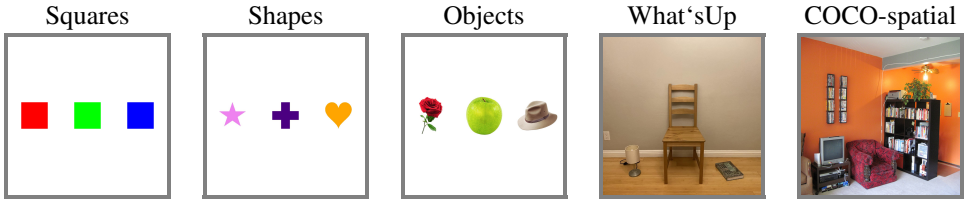
1 INTRODUCTION

Spatial variable binding, the ability to associate objects with their properties and their relations to other elements, is a fundamental component of multimodal reasoning (Cheng et al., 2024; Chen et al., 2021; Zheng et al., 2025; Chen et al., 2024; Goetting et al., 2024; Wen et al., 2025). Tasks such as image captioning, visual question answering, and visual navigation require vision–language models (VLMs) to bind objects to spatial relations and reason about their relative arrangement within the scene. Although recent advances have made VLMs highly capable across many tasks, spatial variable binding remains a significant challenge (Campbell et al., 2024; Chen et al., 2025; Wen et al., 2024; Kamath et al., 2023). In this work, we investigate the internal mechanisms underlying spatial variable binding in VLMs and show that the mechanisms identified in controlled settings also account for model behavior in complex natural scenes.

In language models (LMs), recent studies have shown that associating entities with their attributes (often referred to as variable binding) is done by forming symbolic representations for each entity in context. This representation encodes content-independent information about the order of entities in context, and allows the model to distinguish between entities and retrieve entity-specific information when needed (Dai et al., 2024; Prakash et al., 2024; 2025). Follow-up studies suggest that VLMs similarly rely on ordering information when reasoning about objects in images (Assouel et al., 2025; Kang et al., 2026). However, while these findings establish the use of ordering-based representations during multimodal reasoning, they do not reveal their origin. In particular, it remains unclear whether such representations are constructed within the LM backbone, inherited from the vision encoder, or emerge from interactions between the two. Identifying the source of these representations is essential for understanding spatial variable binding failures in VLMs and potentially developing principled methods to diagnose and correct them.

We show that the ordering-based representations underlying spatial variable binding in VLMs arise from two concurrent sources: The vision encoder represents the global layout of objects in the image, encoding ordering information that is directly projected into the embedding space of the LM

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



"The <property> of the <object type> to the <direction> of <reference object> is _____"

Figure 1: **Experimental Settings.** We study the internal mechanism responsible for spatial variable binding across three synthetic settings (Squares, Shapes, Objects) and one natural controlled settings (What'sUp). We further show that our findings generalize to complex natural scenes from the COCO dataset with complex scenes and objects, where a simple intervention corrects spatial binding failures.

backbone. Then, the LM backbone can further augment these representations by forming ordering information over object-associated visual tokens. While the vision encoder provides the primary source of ordering information, the LM-side mechanism plays a secondary role, augmenting spatial ordering when that information in the vision embeddings is degraded or completely removed. Notably, the ordering information provided by the vision encoder is distributed globally across visual tokens, extending beyond object regions into surrounding background areas. This is in contrast to the LM backbone, where information is formed locally over object-related tokens, similar to the process found in language processing (Prakash et al., 2025; Assouel et al., 2025).

We establish these results through a series of controlled interchange intervention experiments (Vig et al., 2020; Meng et al., 2022; Geiger et al., 2022) using carefully constructed counterfactual samples across three synthetic datasets and one controlled naturalistic dataset (Kamath et al., 2023). We validate our findings across five transformer-based VLMs from three model families (Qwen, Gemma, and Pixtral), showing that they rely on similar mechanisms to solve spatial variable binding. Finally, we demonstrate that this mechanism extends to natural images: a simple global intervention that amplifies vision-derived ordering information across all image tokens corrects up to 55% of previously incorrect predictions on complex, in-the-wild COCO scenes with diverse objects and spatial relations (Lin et al., 2014; Kamath et al., 2023) (see Fig. 12).

Overall, our results elucidate the mechanisms underlying spatial association in VLMs, connect binding mechanisms identified in LMs to multimodal reasoning, and demonstrate the central role of vision encoders in supporting spatial variable binding in VLMs. All code and data supporting this study will be made publicly available upon publication.

2 Related Works

Benchmarking Spatial Reasoning Many benchmarks have been proposed to evaluate spatial reasoning in VLMs, including synthetic datasets designed to test compositional relations and naturalistic benchmarks derived from real images and human annotations (Kamath et al., 2023; Chen et al., 2024; Gholami et al., 2025; Ma et al., 2025). These benchmarks have revealed that spatial reasoning remains challenging for current VLMs, particularly as relational complexity increases or when reasoning must generalize beyond object-centric cues.

Understanding Inner Working of VLMs A growing body of work has investigated the internal workings of VLMs, examining cross-modal attention patterns, token alignment, representational geometry, and information flow between vision encoders and LM backbones (Nikankin et al., 2025; Qin et al., 2025; Assouel et al., 2025; Jiang et al., 2025; Neo et al., 2025; Jiang et al., 2024; Kaduri et al., 2025; Rott Shaham et al., 2024; Cohen et al., 2026). These studies have provided valuable insights into how visual and textual information is integrated and how multimodal representations emerge. Our work builds on this line of research by focusing specifically on the mechanisms underlying spatial association and reasoning.

Symbolic Representation in LMs & VLMs Several studies have shown that LMs implement variable binding by forming content-independent ordering identifiers, over important tokens (Gur-Arieh et al., 2025; Prakash et al., 2025; Dai et al., 2024; Prakash et al., 2024; Feng & Steinhardt, 2024; Davies et al., 2023). These mechanisms underlying variable binding are fundamental to

many in-context reasoning tasks and have been presented as evidence of symbol-like processing in neural networks. Recent work has extended this line of inquiry to VLMs, investigating similar symbolic representations (Kang et al., 2026; Assouel et al., 2025; Saravanan et al., 2025; Hasani et al., 2025). Our findings relate to this body of work while revealing an important distinction in the multimodal setting. We show that although VLMs form symbolic representations that encode ordering information within the LM backbone, they only act as a supporting mechanism. Instead, symbolic representations originating from the vision encoder play a dominant role.

3 Experimental Setup

3.1 Data

We study spatial variable binding across increasing visual complexity, from synthetic and controlled settings to complex natural images (Fig. 1). The controlled settings are necessary to enable causal analysis, which requires constructing original-counterfactual input pairs that differ in exactly one factor. The naturalistic settings then provide evidence that the findings apply to real-world conditions.

Synthetic Settings We consider three synthetic settings: *Squares*, *Shapes*, and *Objects*. In all settings, three equal-sized items are placed at fixed, equal distances and arranged either horizontally or vertically. The settings differ only in visual content: *Squares* uses colored squares, *Shapes* uses colored geometric shapes, and *Objects* uses real-world object images. All images are generated programmatically from predefined collections of colors, shapes, and objects (see full list at App. A), enabling precise control over attributes and spatial configuration. For each image, we query the VLM to predict an attribute (color, shape, or object name) of a target entity (square, shape, or object) based on its relative position to a reference entity. For our intervention experiments (see Sec. 5), we sample 100 clean–counterfactual image pairs from each setting and report 95% bootstrap confidence intervals; we validate the statistical reliability of this sample size in App. C.4. Importantly, we further validate that our results generalize to images with real-world backgrounds (App. C.11) and to diagonal spatial arrangements (App. C.5).

Naturalistic Setting For mechanistic analysis, we use the control subset of What'sUp (Kamath et al., 2023) which consists of images containing pairs of objects: a central reference object (e.g., a chair) and a secondary object positioned adjacent to it on either side. To adapt this dataset to our intervention experiments (Sec. 5), which require images containing three objects, we construct composite scenes by merging pairs of What'sUp images into a single image with a shared reference object and two surrounding objects, resulting in 1074 images. This construction supports only horizontal spatial relations. In Sec. 5.4, we evaluate generalization to natural images from COCO-spatial (Kamath et al., 2023; Lin et al., 2014), curated by filtering the COCO validation set for images that exhibit spatial relations. This results in 2687 images with complex scenes containing multiple objects, varied spatial arrangements, and realistic backgrounds. We also report generalization of these results to the on/under and front/behind relations using the original What'sUp images.

3.2 Models

The mechanistic study is performed with five transformer-based VLMs from three model families: `Qwen2-VL-7B-Instruct`, `Qwen2-VL-2B-Instruct`, `Qwen3-VL-8B-Instruct`, `Gemma-3-4b-it`, and `Pixtral-12B`. These models differ in scale, training data, and vision-encoder design (in particular, the positional encoding of the vision encoder; see App. C.6.2), allowing us to assess whether the identified mechanisms generalize across diverse VLM designs. The behavioral performance of the models on the spatial variable binding tasks is summarized in App. A.1 (Table 2). In the main text we present results for `Qwen2-VL-7B-Instruct`; results for all other models are reported in App. C.1 and App. C.3. For causal experiments, we use examples where the model answers correctly, ensuring the identified mechanisms reflect genuine behavior rather than noise. Nevertheless, we show generalization of the findings to failure cases in Sec. 5.4.

3.3 Methods

Representation Probing A probe (Belinkov, 2022) is typically a lightweight classifier trained to predict a target property y from an internal representation $\mathbf{h} \in \mathbb{R}^d$ extracted from the model. In its

simplest form, a linear probe takes the form $\hat{y} = \mathbf{W}\mathbf{h} + \mathbf{b}$, where $\mathbf{W}$ and $\mathbf{b}$ are learned parameters. Successful probing that generalizes to test cases indicates that the target information is linearly decodable from $\mathbf{h}$, suggesting that it is present in the model's representation.

Causal Mediation Analysis Interchange intervention is a technique for testing causal relationships between a model's internal representations and its behavior (Vig et al., 2020; Meng et al., 2022). Let $f(\cdot)$ denote the model, and let $\mathbf{h}_\ell(x)$ denote the internal representation at component or layer ℓ produced during a forward pass on input x. Given an *original* input x and a corresponding *counterfactual* input x', an interchange intervention replaces $\mathbf{h}_\ell(x)$ with $\mathbf{h}_\ell(x')$ while keeping the remainder of the computation unchanged (*i.e.*, *activation patching*). This yields an intervened output $\hat{y}_{\text{int}} = f\big(x;\, \mathbf{h}_\ell \leftarrow \mathbf{h}_\ell(x')\big)$. If the intervened output $\hat{y}_{\text{int}}$ matches the target outcome associated with the counterfactual input, this provides evidence that the intervened representation $\mathbf{h}_\ell$ plays a causal role in the model's computation. We quantify this effect using *interchange intervention accuracy* (IIA) (Geiger et al., 2022). Rather than measuring binary agreement with the intervention target as in the original IIA, we average probability assigned to the target intervention outcome token.

4 Preliminaries

4.1 Ordering Information in LMs

Variable binding is the process of associating attributes with their respective entities. For instance, given a red, green, and blue square, the process of assigning the color feature to each square involves variable binding. Recent works have investigated the underlying mechanisms that enable variable binding in LMs (Prakash et al., 2025; Dai et al., 2024). A common finding is the reliance of LMs on content-independent ordering representations to bind entities to their attributes. These representations encode the order of entities independently of their content, and align this ordering with a corresponding order of attributes in the context, enabling the model to retrieve entity-specific information when queried. For instance, given a prompt such as "*Box A contains an apple, box B contains a banana, and box C contains cherries. What is the color of the fruit in box B?*", the model uses ordering information corresponding to the second entity (*Box B*) to retrieve the attributes of the corresponding second fruit (*banana*). Prior analyses show that such ordering information is formed in intermediate layers on top of entity-related token positions and is subsequently transferred to the final token position, to retrieve the relevant property (e.g., the color "*yellow*" of the banana).

4.2 Ordering Information in VLMs

In VLMs, spatial variable binding is a fundamental capability central to many spatial reasoning tasks. Recently, Assouel et al. (2025) demonstrates that when VLMs are queried about objects within images, they retrieve symbolic representations that encode their ordering information. While this work establishes the existence of such ordering representations for visual variable binding tasks, it does not characterize the source of it. Specifically, the work mainly shows that for variable binding, the model uses this symbolic representation at the last token position, and that it comes from visual tokens; however, it remains unclear *where and how* VLMs create ordering information. In particular, it remains unclear whether ordering representations are constructed within the LM backbone, inherited directly from the vision encoder, or emerge through interactions between the two components. In this work, we show that the source of this is two-fold: the vision encoder encodes positional information to capture spatial layout, while the LM backbone further enhances this information.

5 Experiments

5.1 VLMs Use Ordering Information for Spatial Variable Binding

We begin by establishing the presence of ordering representations that the model uses at the final token position, across a range of tasks, as described in Sec. 3.1, by conducting interchange intervention experiments at the final token position. Specifically, we separately patch the residual stream vector of each layer at the last token position with that from a counterfactual sample (see Fig. 2) featuring different square colors in the image and a different directional query.

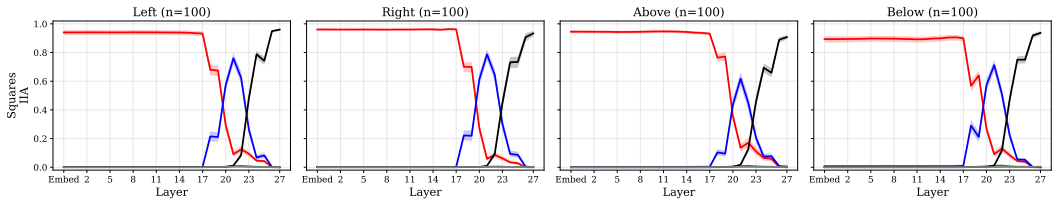


Figure 3: **Last Token Position Patching Results**: The output of the clean run remains unchanged up to layer 19. From layer 20 through layer 22, the output changes to the color of the clean square corresponding to the correct position of the counterfactual run, suggesting that these layers patch the order of the target square. Beyond layer 22, the model predicts the counterfactual answer, indicating that layers after 22 encode the final answer. Shaded areas are 95% bootstrap confidence intervals.

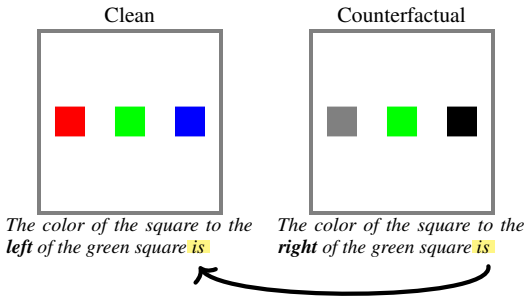


Figure 2: **Last-Token Position Patching.** The residual stream at the final token ("is") of the counterfactual run is patched into the clean run. Clean answer: **Red**. If ordering information is transferred, the output becomes **Blue** (the clean square at the counterfactual's ordinal position); if attribute information is transferred, it becomes **Black** (the counterfactual answer).

If the model uses the ordinal position of the queried square to retrieve its color, we expect this ordering information to be transferred from the counterfactual run to the clean run. Concretely, in the counterfactual run corresponding to Fig. 2, if the model encodes a representation indicating that the third object is the correct one, patching this representation into the clean run should cause the model to select the object with the same ordinal position in the clean image. As a result, the output should change from the color of the originally selected object (the leftmost square; **Red**) to the color of the square in the clean image corresponding to the third position (**Blue**), which matches the correct position in the counterfactual run. In contrast, if the model directly transfers attribute information rather than ordering information, patching the final-token representation from the counterfactual run should cause the output to change directly to the correct answer of the counterfactual image itself (e.g., **Black** in Fig. 2), independent of the clean image. This would indicate that color information and not order is being transferred by the intervention.

Figure 3 shows that the final output initially corresponds to the color of the correct square of the clean input between layers **0-19**, indicating that the intervention on these layers has no effect. It then switches to selecting the color of the square in the clean image corresponding to the ordinal position of the correct square in the counterfactual image **20-22**, indicating that ordering information is being transferred. At layers **23–27**, the model switches to the color of the counterfactual correct square, indicating that correct color attribute information is being transferred directly.

This pattern suggests that the model first computes an ordering representation of the correct square and then, in later layers, retrieves its corresponding color information. For clarity, we present results for the *Squares* setting for the Qwen model. Consistent behavior is observed across all settings and models (App. C.1). Results are averaged over 100 clean–counterfactual pairs per setting, with 95% bootstrap confidence intervals (App. C.4). For convenience, the plots use the color scheme from the example in Fig. 2; however, the specific colors, shapes, and object categories vary across data points, as well as the queried directions.

5.2 Is Vision Encoder the Source of Ordering Representations?

While the previous subsection established the presence of ordering representations, it did not fully characterize them. In particular, a key open question concerns where these ordering representations are generated within VLMs: are they produced in the LM backbone, the vision encoder, or through interactions between the two components? Identifying the source of ordering representations is

important not only for understanding how they arise, but also because it yields predictions that we test on natural images (Sec. 5.4).

5.2.1 Probing Visual Embeddings

We begin by asking whether ordering representations originate in the vision encoder. To address this question, we train linear probes on visual token embeddings, extracted immediately after projection from the vision encoder, to predict the ordinal position of each object in the image. Successful probe performance indicates that ordering information is linearly decodable from the visual embedding.

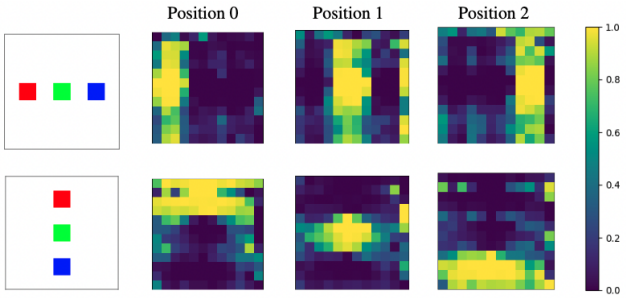


Figure 4: **Position information in visual embeddings extends beyond object regions.** We train linear probes on top of object tokens of the visual embedding to predict their order in the image. At test time, these probes generalize to background tokens, producing strips-like pattern aligned with object position, suggesting that background tokens encode order information of nearby objects. See App. C.9 for more complex arrangements.

For each image setting described in Section 3.1, we train two linear probes, one for horizontal and one for vertical configurations, to classify embeddings as one of three spatial positions. Training data is constructed using token embeddings extracted from 90 images; the embeddings that correspond to the three objects in an image serve as positive examples for their respective positions. In the *Squares*, *Shapes*, and *Objects* datasets, identifying the visual tokens associated with each object is straightforward due to their fixed spatial layouts. In contrast, objects in the *What'sUp* dataset vary in size and are not aligned to fixed positions. To handle this, we preprocess *What'sUp* images by extracting bounding boxes for each object and use their spatial coordinates to identify the corresponding visual tokens (see Fig. 63).

The resulting classifiers achieve nearly perfect accuracy on a test set of 30 images. In addition to classifying object-token representations, we also apply the probes to all other tokens in the image to assess whether ordering information representations are present in them. Surprisingly, we find that the information about the object ordering is not confined to tokens corresponding to the objects themselves, but is instead distributed across multiple background tokens, as shown in Fig. 4. Notably, probe predictions exhibit a strip-like spatial pattern aligned with object position, indicating that positional information is distributed coherently across background tokens rather than localized to object regions. In App. C.6.2 we trace at which vision-encoder layer these strips emerge and relate the pattern to the encoder's positional encoding. This behavior contrasts with prior work on LMs (Prakash et al., 2024; 2025; Dai et al., 2024), which shows that such representations are typically localized to a single token or a small set of adjacent tokens. In the following subsections, we demonstrate that the information contained in these strips is causally relevant for producing the correct final output.

5.2.2 Causal Intervention on Vision Tokens

Although the probing results demonstrate the presence of ordering information in visual representations, they do not by themselves establish a causal relationship with the model's final output. We therefore perform interchange intervention experiments to test whether systematically manipulating these representations leads to corresponding changes in the model's predictions.

As illustrated in Fig. 5, we intervene on visual token embeddings by swapping the left and right strips of the clean image with those from a counterfactual image. Specifically, we replace the embeddings of the left strip in the clean image with the right-strip embeddings from the counterfactual image, and vice versa. The intervention is designed such that both patched regions contain squares of the same color, ensuring that color values are preserved and that only ordering information is altered. In addition to intervening on the visual token embeddings, we also separately patch the residual stream vectors of each layer of the LM backbone.

If ordering information encoded by the vision encoder is causally relevant, this intervention should cause the model's final output on the clean image to switch from the left square (red) to the right

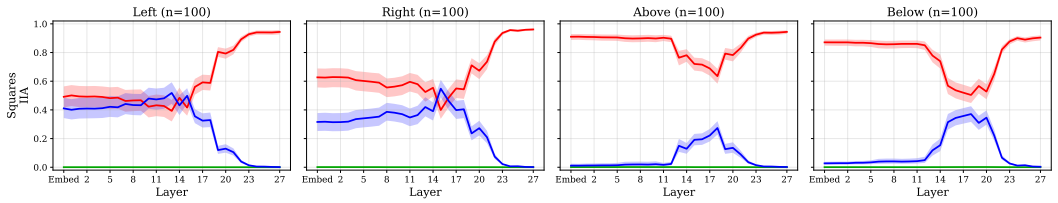


Figure 6: **Object patching**. Interchange interventions applied only to visual tokens corresponding to the squares. Patching square tokens at any layer does not reliably change the model's final output relative to the clean run, indicating that square-localized tokens alone do not carry sufficient ordering information to determine the model's prediction.

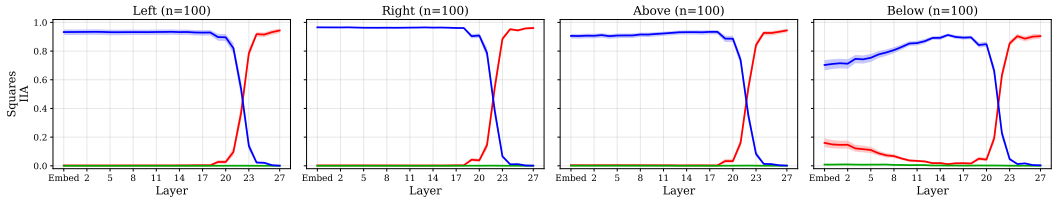


Figure 7: **Strip patching**. Interchange interventions applied to both square tokens and their background strips immediately switch the model's output to the incorrect color in the clean image, consistent with the transferred ordering information. This demonstrates that ordering information is distributed across strip-aligned visual tokens and is causally relevant for reasoning.

square (blue), reflecting the swapped ordering information. As shown in Fig. 7, we observe the expected change in the final output when either the visual token embeddings or the residual stream vectors up to layer 24 are patched. In contrast, patching only the square-localized visual tokens, without the surrounding strip, does not reliably induce this change (Fig. 6). These indicate that ordering information encoded by the vision encoder is not localized to object tokens, but is distributed across background visual tokens.

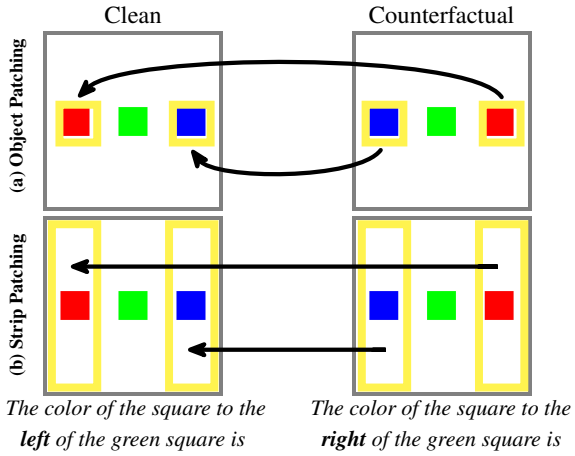


Figure 5: **Causal intervention on visual token embeddings.** The visual tokens of the left and right squares are swapped (left↔right) from the counterfactual run into the clean run, either (a) for the square tokens only or (b) together with their background strips. If the patched tokens carry ordering information, the output switches to the square on the opposite side of the queried reference.

We observe similar behavior in other settings, including Shapes, Objects, and What'sUp, which also generalizes to all models, as described in App. C.3. All results are averaged over 100 clean–counterfactual pairs (shaded: 95% bootstrap CIs). Moreover, the original correct square's logit overtakes the intervened one around layer 23, indicating that by then all required information, both ordering and color, has been transferred to the last token (Sec. 4.2).

5.3 LM Backbone Enhances Ordering Information

In the previous sections, we showed that visual token embeddings produced by the vision encoder and projector module already encode object ordering information. We now ask what role, if any, the LM backbone plays in spatial variable binding. Namely, does the LM backbone merely consume ordering information from the vision components, or can it generate such information when needed?

5.3.1 Removing Ordering Information from Vision Embeddings

To isolate the contribution of the LM backbone, we first ablate ordering information from the visual token embeddings using the interventions illustrated in Fig. 8. Specifically, we apply two modifications: (1) we replace the embedding of each square with an embedding of the same-colored square taken from an image in which that square appears alone in the middle of the image, thereby preserving color information while removing relative ordering; and (2) we replace all background-token embeddings with those from an empty image, eliminating ordering information encoded in background regions. Together, these interventions effectively remove ordering information originating from the vision encoder, while preserving object colors. We do not apply this intervention to the What'sUp dataset, as constructing a precise intervention is challenging for natural images.

As shown in Table 6 (App. C.2), removing ordering information from the vision embeddings leads to a substantial drop in performance across datasets, confirming the central role of vision-derived ordering information. However, performance remains above chance (33.3%), suggesting that the LM backbone can compensate for the missing ordering signals.

5.3.2 Do LMs Backup Ordering Information?

To test whether the LM backbone generates its own ordering information, we perform an interchange intervention experiment analogous to that described in Section 5.2.2, but under the ablated-vision setting described above (e.g., the LM gets images with no ordering information from the vision encoder). In contrast to earlier experiments, here we patch only the square-associated visual tokens, rather than the entire strips.

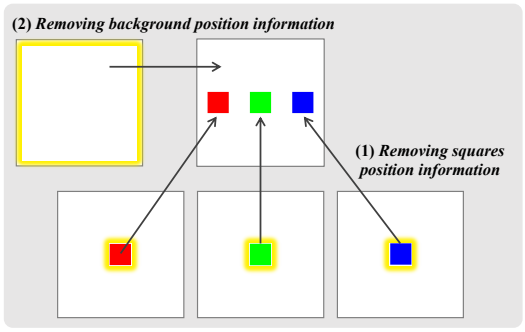


Figure 8: **Ordering information is ablated from vision embeddings** by replacing square embeddings with same-colored isolated middle squares and background embeddings with those from an empty image, preserving color while removing position cues.

If the LM backbone generates ordering information independently, then patching square tokens from a counterfactual run should transfer this order information and induce a change in the final output. If not, the model's prediction should remain unchanged. The results in Fig. 9 show a flip in model prediction according to order information at middle layers. This indicates that, when vision-derived ordering information is removed, the LM backbone forms its own ordering representations in middle layers, and that patching these representations switches the prediction according to the patched order. See App. C.3 for other settings and models.

Together, these results indicate that VLMs rely on two sources of ordering information: a primary signal originating from the vision encoder and projector module, and a secondary signal generated within the LM backbone. While the vision-derived signal dominates when present, the LM backbone can partially reconstruct ordering information and act as a backup mechanism. Supporting evidence for this interpretation is provided in Figs. 7, 23, 22, where we show that vertically arranged objects strengthen ordering signals in the same intermediate LM layers, indicating the LM backbone enhances the ordering information provided by the vision embedding in these cases. We further find that the LM-backbone ordering information is predominantly encoded by relative position, whereas the vision encoder encodes both absolute and relative position (App. C.6.2).

5.4 Validation on Natural Images

Sections 5.1–5.3 characterize the mechanism on controlled synthetic images. In this subsection we check whether it generalizes to natural images. Specifically, we test three hypotheses that follow from the synthetic-setting results: (i) the vision encoder's ordering representations should be decodable from natural images globally; (ii) strengthening the vision-encoder ordering information should correct spatial binding failures; and (iii) if the ordering information is content-independent, it should transfer across object types and spatial relations.

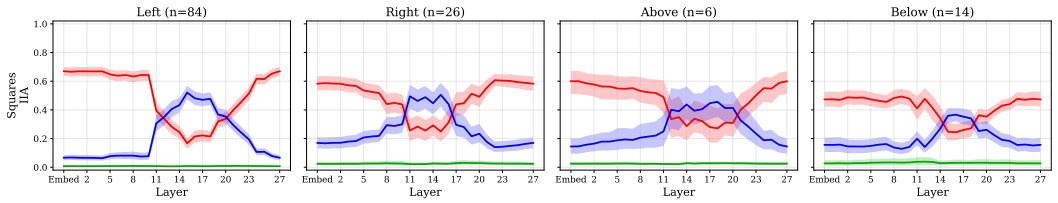


Figure 9: **Patching results after removing ordering information from vision encoder.** The model's output switches only in intermediate layers within the range of 11-20, indicating that the LM backbone generates ordering information even when vision-derived ordering representations are absent. Panel titles give the per-direction n, which is capped by the accuracy of the ablated model.

We test this on COCO-spatial (Sec. 3.1; examples in Fig. 12), where the model states the relation (left/right or above/below) between a query and a reference object. Since the position probes of Sec. 5.2.1 are trained to capture ordering information, adding their directions to the visual embeddings should enhance the vision-encoder ordering signal. Concretely, for each visual token t we set $\text{emb}_t \leftarrow \text{emb}_t + \alpha_i \cdot \text{probe}_i$, where probe_i is the probe for direction i and $\alpha_i \in [1, 15]$. The intervention is applied to all visual tokens and probe directions, so neither object locations nor the queried direction is needed at test time; probes are trained once on annotated object tokens (App. A.4).

Table 1: Amplifying ordering representation corrects failures.

	Gemma		Qwen	
Intervention	**Acc.**	**% Corr. Fail.**	**Acc.**	**% Corr. Fail.**
None	0.53	–	0.60	–
Random amp.	0.57	9.6%	0.65	13.1%
Ordering amp.	**0.72**	**40.2%**	**0.82**	**54.5%**

Table 1 reports `Gemma-3-4b-it` and `Qwen2-VL-2B-Instruct` (the `7B` variant is near ceiling here), using a per-item α; fixed-α results and all other models are in App. C.7. Ordering directions correct 40% and 54% of failures versus 10% and 13% for random directions, so the gains are specific to ordering information, which confirms that vision-encoder ordering information also dominates in complex natural images. This experiment aims to test generalization to natural settings, where causal interventions are hard to design, not to reach state-of-the-art accuracy. Still, chain-of-thought (CoT) prompting is no substitute: it raises `Qwen2-VL-2B`'s accuracy but lowers `Gemma-3-4b`'s, and breaks about three times as many previously correct items as amplification (App. C.7.5). Amplification costs at most 5 points on seven non-spatial VQA benchmarks (App. C.7.6).

We conduct additional experiments to check the generalization of these results to further natural-image settings. (i) *Transferability*: position probes trained on object images, as in Sec. 5.2.1, recover the strip-like ordering patterns on What'sUp images (App. C.10) and on synthetic objects over real backgrounds (App. C.11). This indicates that the ordering information captured by probe directions is transferable across object and image settings. (ii) *Correcting more failures*: amplification also corrects 43–55% of failures on the What'sUp composites (App. C.7.2) and 53–68% for additional object relations such as "on/under" and "in front/behind" on the original, unstitched What'sUp images (examples in Fig. 43; App. C.7.3). This indicates that the findings also hold for more complex object relations, including depth. The gains persist when α is selected on a validation split and evaluated on held-out failures (App. A.4). (iii) *Content independence*: a probe trained on What'sUp corrects COCO-spatial failures as well as an in-domain probe (48% vs. 47%), and a horizontal probe corrects vertical failures nearly as well as a vertical one (33% vs. 35%; App. C.7.4), indicating a shared ordering code rather than dataset- or axis-specific features. Together, these results indicate that the mechanism identified in the controlled synthetic settings also governs spatial variable binding in complex natural scenes.

6 Conclusions

We show that spatial variable binding in VLMs relies on ordering representations arising from two complementary sources: a primary signal encoded by the vision encoder and a secondary mechanism formed within the LM backbone. By isolating and intervening on these representations, we show that the same mechanism governs natural images, where amplifying them corrects binding failures. Our results highlight that mechanistic understanding can improve model performance.

AI USE STATEMENT

In this work, we used generative AI tools to polish the text of the manuscript, for example by fixing typos and grammatical errors. All AI-assisted text was reviewed by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies the internal mechanisms of publicly released, open-weight vision–language models using programmatically generated synthetic images and two existing public datasets, What'sUp (Kamath et al., 2023) and COCO (Lin et al., 2014). It involves no human subjects, no collection of personal data, and no annotation beyond the labels distributed with these datasets. The interventions we describe modify a model's internal activations only to amplify spatial ordering information that the model already computes; they add no new capabilities and are unlikely to enable misuse beyond that of the underlying models. We discuss broader impacts in App. B.2 and limitations in App. B.1.

REPRODUCIBILITY STATEMENT

The construction of all datasets is described in Sec. 3.1 and App. A, including the full lists of colors, shapes, and objects used in the synthetic settings (App. A.2), the exact prompt templates (App. A.3), and the image-tokenization parameters and strip widths for each model (Table 4). All five models are publicly available and are identified by their release names in Sec. 3.2. The probing and interchange-intervention procedures are specified in Sec. 3.3; the sample sizes, bootstrap protocol, and a comparison of sample sizes in App. C.4; and the amplification protocol, probe-training details, and the selection of the amplification coefficient, including a held-out validation of that choice, in App. A.4. Compute requirements are given in App. B.3. All code and data supporting this study will be made publicly available upon publication.

REFERENCES

Rim Assouel, Declan Campbell, Yoshua Bengio, and Taylor Webb. Visual symbolic mechanisms: Emergent symbol processing in vision language models. *arXiv preprint arXiv:2506.15871*, 2025.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Declan Campbell, Sunayana Rane, Tyler Giallanza, Camillo Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven Frankland, Tom Griffiths, Jonathan D Cohen, et al. Understanding the limits of vision language models through the lens of the binding problem. *Advances in Neural Information Processing Systems*, 37:113436–113460, 2024.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 513–523, 2021.

Shiqi Chen, Tongyao Zhu, Ruochen Zhou, Jinghan Zhang, Siyang Gao, Juan Carlos Niebles, Mor Geva, Junxian He, Jiajun Wu, and Manling Li. Why is spatial reasoning hard for vlms? an attention mechanism perspective on focus areas, 2025. URL https://arxiv.org/abs/2503.01773.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

Ido Cohen, Daniela Gottesman, Mor Geva, and Raja Giryes. Performance gap in entity knowledge extraction across modalities in vision language models, 2026. URL https://arxiv.org/abs/2412.14133.

Qin Dai, Benjamin Heinzerling, and Kentaro Inui. Representational analysis of binding in language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17468–17493, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.967. URL https://aclanthology.org/2024.emnlp-main.967/.

Xander Davies, Max Nadeau, Nikhil Prakash, Tamar Rott Shaham, and David Bau. Discovering variable binding circuitry with desiderata. *arXiv preprint arXiv:2307.03637*, 2023.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *International Conference on Learning Representations*, 2024.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pp. 7324–7338. PMLR, 2022.

Mohsen Gholami, Ahmad Rezaei, Zhou Weimin, Sitong Mao, Shunbo Zhou, Yong Zhang, and Mohammad Akbari. Spatial reasoning with vision-language models in ego-centric multi-view scenes. *arXiv preprint arXiv:2509.06266*, 2025.

Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. End-to-end navigation with vision language models: Transforming spatial reasoning into question-answering. *arXiv preprint arXiv:2411.05755*, 2024.

Yoav Gur-Arieh, Mor Geva, and Atticus Geiger. Mixing mechanisms: How language models retrieve bound entities in-context. *arXiv preprint arXiv:2510.06182*, 2025.

Hosein Hasani, Amirmohammad Izadi, Fatemeh Askari, Mobin Bagherian, Sadegh Mohammadian, Mohammad Izadi, and Mahdieh Soleymani Baghshah. Uncovering grounding ids: How external cues shape multimodal binding. *arXiv preprint arXiv:2509.24072*, 2025.

Nick Jiang, Anish Kachinthaya, Suzie Petryk, and Yossi Gandelsman. Interpreting and editing vision-language representations to mitigate hallucinations, 2024. URL https://arxiv.org/abs/2410.02762.

Nick Jiang, Amil Dravid, Alexei Efros, and Yossi Gandelsman. Vision transformers don't need trained registers. *arXiv preprint arXiv:2506.08010*, 2025.

Omri Kaduri, Shai Bagon, and Tali Dekel. What's in the image? a deep-dive into the vision of vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14549–14558, June 2025.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? Investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Raphi Kang, Hongqiao Chen, Georgia Gkioxari, and Pietro Perona. Linear mechanisms for spatiotemporal reasoning in vision language models. *arXiv preprint arXiv:2601.12626*, 2026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso de Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6924–6934, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models, 2025. URL https://arxiv.org/abs/2410.07149.

Yaniv Nikankin, Dana Arad, Yossi Gandelsman, and Yonatan Belinkov. Same task, different circuits: Disentangling modality-specific mechanisms in vlms. *arXiv preprint arXiv:2506.09047*, 2025.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations*, 2024.

Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Rott Shaham, David Bau, and Atticus Geiger. Language models use lookbacks to track beliefs, 2025. URL https://arxiv.org/abs/2505.14685.

Yulu Qin, Dheeraj Varghese, Adam Dahlgren Lindström, Lucia Donatelli, Kanishka Misra, and Najoung Kim. Vision-and-language training helps deploy taxonomic knowledge but does not fundamentally alter it. *arXiv preprint arXiv:2507.13328*, 2025.

Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Forty-first International Conference on Machine Learning*, 2024.

Darshana Saravanan, Makarand Tapaswi, and Vineet Gandhi. Investigating mechanisms for in-context vision language binding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4852–4856, 2025.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Chuan Wen, Dinesh Jayaraman, and Yang Gao. Can transformers capture spatial relations between objects?, 2024. URL https://arxiv.org/abs/2403.00729.

Congcong Wen, Yisiyuan Huang, Hao Huang, Yanjia Huang, Shuaihang Yuan, Yu Hao, Hui Lin, Yu-Shen Liu, and Yi Fang. Zero-shot object navigation with vision-language models reasoning. In *International Conference on Pattern Recognition*, pp. 389–404. Springer, 2025.

Xu Zheng, Zihao Dongfang, Lutao Jiang, Boyuan Zheng, Yulong Guo, Zhenquan Zhang, Giuliano Albanese, Runyi Yang, Mengjiao Ma, Zixin Zhang, et al. Multimodal spatial reasoning in the large model era: A survey and benchmarks. *arXiv preprint arXiv:2510.25760*, 2025.

Appendix

We bring here additional results and experimental details.

A Data

A.1 Models' Accuracy

Table 2 reports the accuracy of all five models on the studied tasks; Table 3 breaks the synthetic settings down by queried direction.

Table 2: Accuracy of all five models on the studied tasks.

	Squares	Shapes	Objects	What'sUp
Qwen2-VL-7B-Instruct	1.00	1.00	1.00	0.73
Gemma-3-4b-it	0.98	0.99	0.99	0.62
Pixtral-12B	0.55	0.55	0.63	0.44
Qwen2-VL-2B-Instruct	1.00	1.00	0.96	0.66
Qwen3-VL-8B-Instruct	1.00	1.00	1.00	0.88

	Squares				Shapes				Objects			
	L	R	A	B	L	R	A	B	L	R	A	B
Qwen2-VL-7B-Instruct	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Qwen2-VL-2B-Instruct	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.96	0.99	0.95	0.92
Qwen3-VL-8B-Instruct	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Gemma-3-4b-it	1.00	1.00	0.93	0.97	1.00	1.00	0.97	0.97	1.00	1.00	0.95	1.00
Pixtral-12B	0.59	0.57	0.50	0.54	0.25	0.57	0.68	0.69	0.61	0.57	0.67	0.67

Table 3: Full behavioral performance, according to queried directions (Left, Right, Above, Below)

A.2 Image Settings

Each of our settings consists of images constructed from the following possible objects:

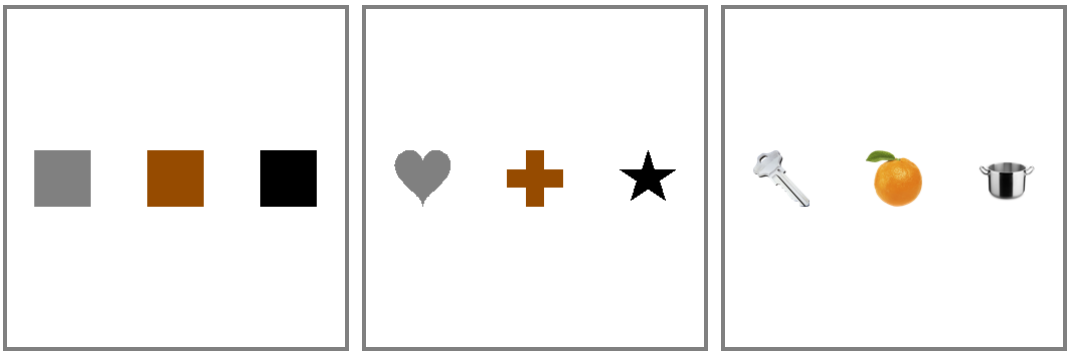
Squares:
`red`, `green`, `blue`, `black`, `brown`, `gray`

Shapes:
`red square`, `green circle`, `blue triangle`, `black star`, `brown cross`, `gray heart`

Objects:
`orange`, `apple`, `pot`, `bell`, `hat`, `rose`, `key`, `bomb`

For each model, we report the following parameters in Table 4.

- **# Image tokens**: total number of image tokens in an image, as a square grid of `num tokens` × `num tokens` tokens.
- **# Pixels / Token**: number of image pixels per token, as `num pixels` × `num pixels` pixels per token.
- **# Tokens / Object**: the number of tokens each object in the image occupies.
- **Object spacing**: the horizontal/vertical distance, in tokens, between adjacent objects in an image.
- **Strip width**: the strip width, in tokens, used for interventions such as those in Fig. 7.



(a) Square setting (b) Shapes setting (c) Objects setting

Figure 10: Example images of the Square, Shapes, and Objects settings.

Model	# Image tokens	# Pixels / Token	# Tokens / Object	Object spacing	Strip width
Qwen2-VL-7B-Instruct	12×12	28×28	2×2	2	4
Qwen2-VL-2B-Instruct	12×12	28×28	2×2	2	4
Qwen3-VL-8B-Instruct	12×12	32×32	2×2	2	4
Gemma-3-4b-it	16×16	56×56	3×3	3	3
Pixtral-12B	15×15	16×16	2×2	2	2

Table 4: Image tokenization and spatial layout parameters for each model. Qwen2-VL-2B shares the 7B vision encoder (14-px patches merged 2×2), and Qwen3-VL-8B uses 16-px patches merged 2×2, so both produce the same 12×12 grid as Qwen2-VL-7B. Pixtral's encoder has a fixed 16-px patch stride, so a 240-px canvas yields a 15×15 grid.

A.3 Prompts

Generation prompt tokens are added to the end of each prompt to signal the beginning of the assistant response.

System prompt (all settings):
`You are a helpful assistant. You respond in one token.`

User prompt templates:
Each `<direction>` $\in$ {`left`, `right`, `above`, `below`} is mapped to a preposition:

`{left` $\mapsto$ `"to the left of", right` $\mapsto$ `"to the right of", above` $\mapsto$ `"above", below` $\mapsto$ `"below"}`

Squares:
`The color of the square <preposition> the <middle color> square is`
Shapes:
`The color of the shape <preposition> the <middle color> <middle shape> is`
Objects:
`The object <preposition> the <middle object> is a(n)`
What's Up (Objects):
`The object on the floor <preposition> the <middle object> is a(n)`

Two-choice prompt templates: For the templates below, `<option 1>` and `<option 2>` denote the two answer options: object labels for What'sUp and relation labels for COCO. Each example is evaluated under both option orderings: `<option 1> or <option 2>` and `<option 2> or <option 1>`.

What'sUp (Objects):

```
Which object is <preposition> the <middle object>:
<option 1> or <option 2>?  Answer with one word.
```

COCO-spatial, horizontal (Gemma):

```
Is the <subject object> to the <option 1> or to the
<option 2> of the <reference object>?  Answer with one
word.
```

COCO-spatial, vertical (Gemma):

```
Is the <subject object> <option 1> or <option 2> the
<reference object>?  Answer with one word.
```

A.4 NATURAL-IMAGE VALIDATION – EXPERIMENTAL SETUP

Following the dataset construction described in Sec. 3.1, we evaluate the ordering-amplification intervention from Sec. 5.4 on the What'sUp composites and COCO-spatial examples.

For What'sUp, each composite image contains 3 objects, yielding two two-option forced-choice questions; querying the object to the `left` and to the `right` of the middle reference object. For example, a possible query for the leftmost image in Fig 11 is *Which object is to the left of the chair: a book or a lamp? Answer with one word.* and the other query is Which object is to the right of the chair: a book or a lamp? Answer with one word.



Figure 11: **Examples for images from What'sUp control (Kamath et al., 2023)**

For COCO-spatial, each pair of objects serves as subject–reference for querying, and is converted into a binary spatial-relation question: `left`/`right` for horizontal relations and `above`/`below` for vertical relations. This formulation reduces dependence on open-ended object naming or recognition. For example, a possible query for the leftmost image in Fig 12 is *Is the dog to the left or to the right of the newspaper? Answer with one word.* and the other query is Is the newspaper to the left or to the right of the dog? Answer with one word.



Figure 12: **Examples for images from COCO-spatial (Kamath et al., 2023; Lin et al., 2014)**

To control for answer-option order bias, each query is evaluated twice, once with the answer options A and B in the query, and once with the reverse order: option B and A (for example *Is <subject> to the **left** or to the **right** of <reference>* and then again with reverse answer options: *Is <subject> to the **right** or to the **left** of <reference>*) An example is counted as correct only if both queries are answered correctly; otherwise, it is treated as a failure. Under random guessing, this strict criterion yields a 25% chance-level accuracy.

Amplification directions are obtained from the position probes described in Sec. 5.2.1. For What'sUp, baseline-correct data consists of composite images for which both directional examples, `left` and `right`, are correct under the strict two-ordering protocol. We train a three-class `left`/`middle`/`right` position probe using visual-token embeddings from the left side-object, central reference-object, and right side-object strips. For COCO-spatial, we train separate two-class probes for horizontal and vertical relations using baseline-correct subject–reference examples. The original COCO bounding boxes of the subject and reference objects are mapped to visual tokens, whose embeddings are mean-pooled within each box to obtain object-level representations for probe

training. Probes are trained as linear classifiers with AdamW (lr=10^{-3}, weight_decay=10^{-4}) for 200 epochs.

As described in Sec. 5.4, the intervention amplifies all the learned ordering directions in the visual embeddings. For each failure example, we test only the queries that were incorrect before amplification (e.g., asking if the subject is *to the left or to the right* of the reference, and not about whether it is *to the right or to the left* if the second was answered correctly). The example is counted as corrected only if all previously failed query(s) become correct under the intervention. If the model originally failed under both queries, the same amplification coefficient α must correct both.

We use consistent notation across all amplification tables in Sec. C.7. PROBE and RANDOM denote the fixed-α setting, where a single coefficient is selected across all examples. PROBE* and RANDOM* denote the per-example setting, where each example may use the coefficient that yields its best correction outcome, serving as an upper bound on intervention performance. We report post-intervention accuracy and corrected-failure rate, i.e., the fraction of baseline failures corrected by the intervention.

Held-out selection of the amplification coefficient. The fixed-α results above select the single best α on the same failures on which it is evaluated. To verify that this does not inflate the reported fix rates, we repeat the experiment with a validation/test split of the baseline failures (seed 0; approximately 70%/30%): α is chosen only on the validation failures and the fix rate is then measured on the held-out failures. Probe training is unaffected, as probes are trained only on baseline-correct examples. Table 5 shows that the held-out fix rates closely track the all-failures results of Tables 11 and 12 and remain far above the random-direction control, whose held-out fix rate drops to 0–19%.

Table 5: Held-out generalization of the amplification coefficient. α is selected on validation failures only (seed 0, ≈70/30 split) and evaluated on the held-out failures; "fix" is the fraction of held-out failures corrected. Gemma-3 COCO-spatial rows come from a run on a 394-item subset of COCO-spatial (baseline $185/394 = 47.0\%$), so their failure counts differ from Table 11.

				Probe		Random	
Dataset	Model	Axis	$n_{\text{val}}/n_{\text{test}}$	α	Held-out fix	α	Held-out fix
COCO-spatial	Qwen2-VL-2B	horizontal	75/31	8	32.3% (10/31)	15	16.1% (5/31)
COCO-spatial	Qwen2-VL-2B	vertical	49/21	7	38.1% (8/21)	12	19.0% (4/21)
COCO-spatial	Gemma-3-4b	horizontal	97/41	14	22.0% (9/41)	15	17.1% (7/41)
COCO-spatial	Gemma-3-4b	vertical	50/21	15	28.6% (6/21)	10	0.0% (0/21)
What'sUp	Qwen2-VL-2B	horizontal	78/33	13	45.5% (15/33)	15	0.0% (0/33)
What'sUp	Gemma-3-4b	horizontal	310/134	7	58.2% (78/134)	13	12.7% (17/134)

B Additional Discussions

B.1 Limitations.

This work focuses on a small set of spatial relations as a minimal setting that enables rigorous causal analysis. The mechanistic analysis covers left/right and above/below, and extends to diagonal arrangements in App. C.5; the amplification experiment additionally covers the on/under and front/behind relations of What'sUp (App. C.7.3). More complex configurations, such as "between" or hierarchical relations, and a mechanistic account of depth relations such as front/behind, are left for future work. The mechanistic analysis is further limited to controlled and composite settings, as causal interchange interventions require paired inputs that differ in exactly one factor — a condition that cannot currently be satisfied with unmodified natural images. Nevertheless, the amplification experiment of Sec. 5.4 extends to complex natural images from COCO-spatial. Several of the additional analyses are run on a single model: the probe-transfer experiments (App. C.7.4) on `Qwen2-VL-2B` and the diagonal analysis on `Qwen2-VL-7B`. Finally, while we validate our findings across five open-weight VLMs, results on closed-source models or architectures with fundamentally different vision encoders remain unexplored. We view these as important directions for future work.

B.2 Broader Impact Statement

This work underscores the critical role of the vision encoder in multimodal systems, challenging the prevailing focus on the language backbone. By demonstrating that the dominant symbolic representations for spatial variable binding originate in the visual components, we highlight that future advances in VLMs must prioritize the training of robust vision encoders to ensure accurate spatial layout encoding. Furthermore, we show that the identified mechanisms account for model behavior on natural images, where amplifying vision-derived spatial signals corrects spatial binding failures without additional training or fine-tuning.

Methodologically, our discovery that spatial information is diffused across multiple visual tokens, extending even into background regions, signals a necessary paradigm shift for interpretability research. Standard analysis techniques that focus on single tokens are insufficient for capturing these distributed representations. As the field moves toward reasoning models that generate increasingly long sequences of tokens, developing interpretability methods capable of analyzing information distributed across multiple tokens will be essential for diagnosing and enhancing model behaviors.

B.3 Compute Resources

All the experiments were carried out on two 80GB NVIDIA A100 GPUs.

C ADDITIONAL RESULTS

C.1 LAST TOKEN POSITION PATCHING

In addition to the results described in Section 5.1, we report the results of the last-token interchange intervention experiment on all settings (Squares, Shapes, Objects, and What'sUp) and all models: `Qwen2-VL-7B-Instruct` (Fig. 13), `Gemma-3-4b-it` (Fig. 14), `Qwen2-VL-2B-Instruct` (Fig. 15), `Qwen3-VL-8B-Instruct` (Fig. 16), and `Pixtral-12B` (Fig. 17). All curves are means over n=100 clean–counterfactual pairs with 95% bootstrap confidence bands, except Pixtral, for which the original n=50 runs are shown (its lower task accuracy, Table 2, does not admit 100 pairs on which the model is correct for both the clean and the counterfactual prompt). The same three-phase pattern — clean answer, then the answer at the counterfactual's ordinal position, then the counterfactual answer — appears in every model and setting, with the transition shifted to the model's own late layers (e.g., layers 18–21 of 28 for Qwen2-VL-2B, layers 22–25 of 36 for Qwen3-VL-8B).

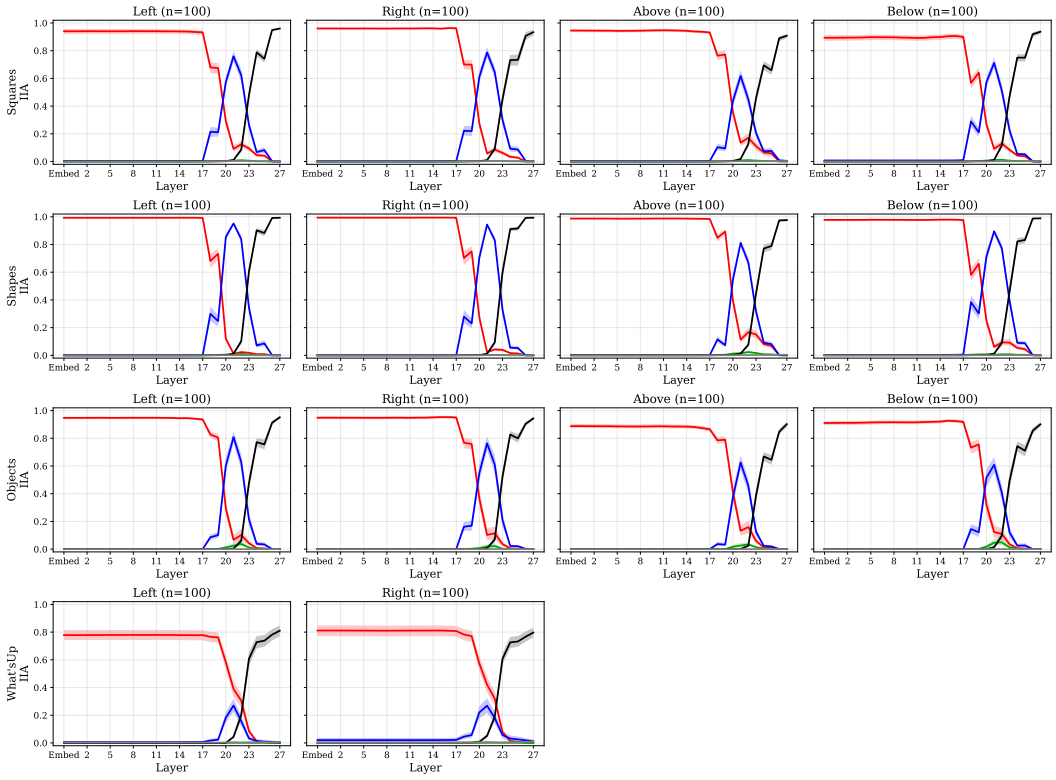


Figure 13: Final token interchange intervention experiments with the `Qwen2-VL-7B-Instruct` model.

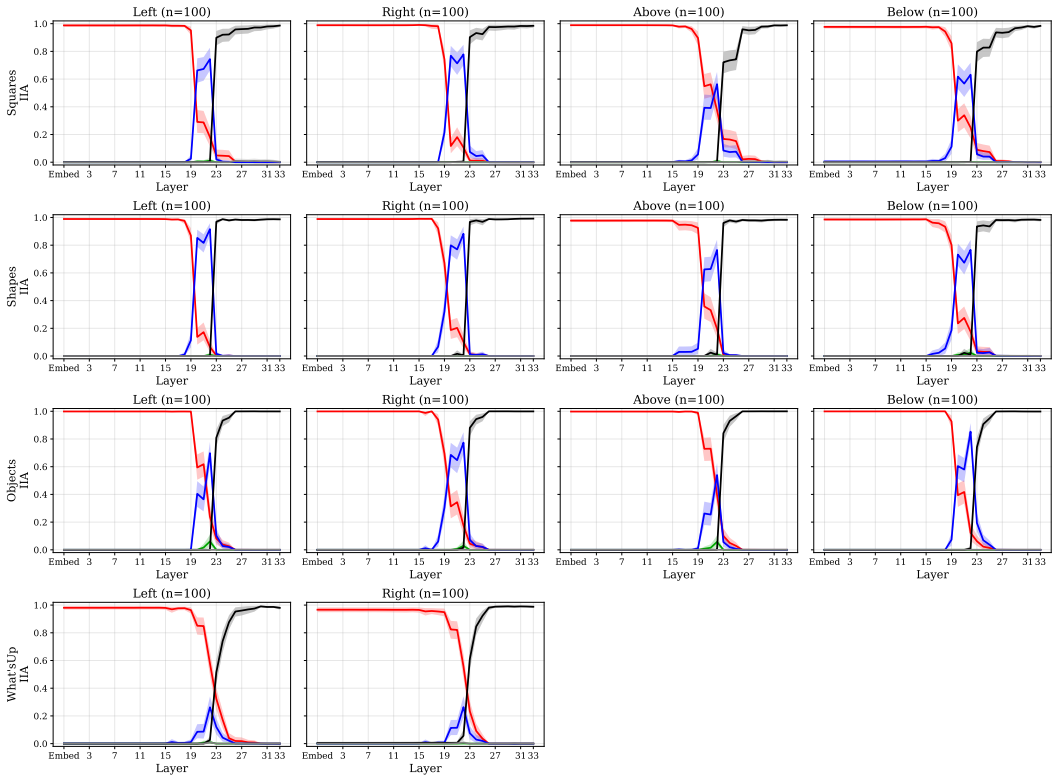


Figure 14: Final token interchange intervention experiments with the `Gemma-3-4b-it` model.

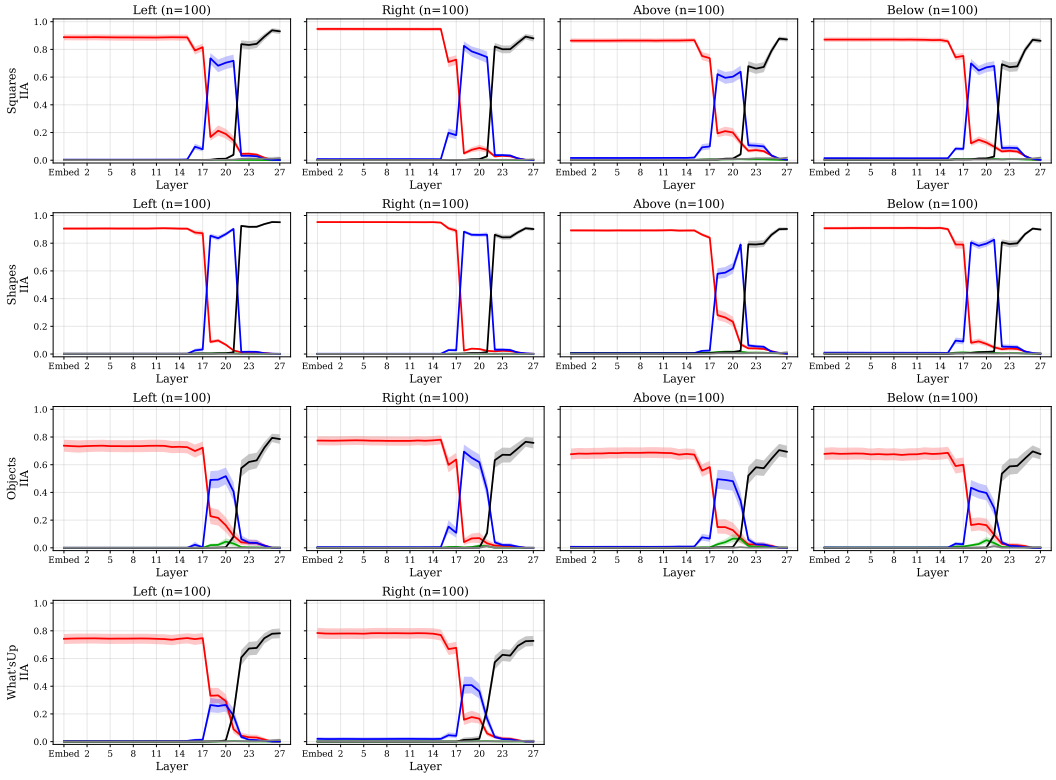


Figure 15: Final token interchange intervention experiments with the `Qwen2-VL-2B-Instruct` model.

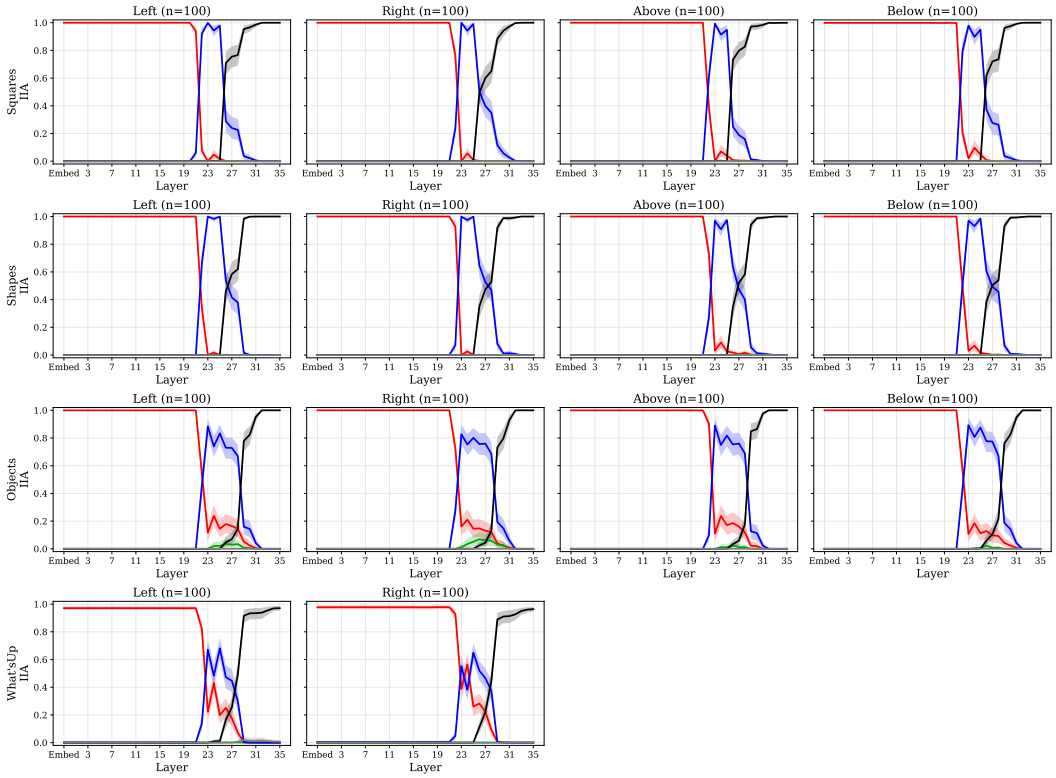


Figure 16: Final token interchange intervention experiments with the `Qwen3-VL-8B-Instruct` model.

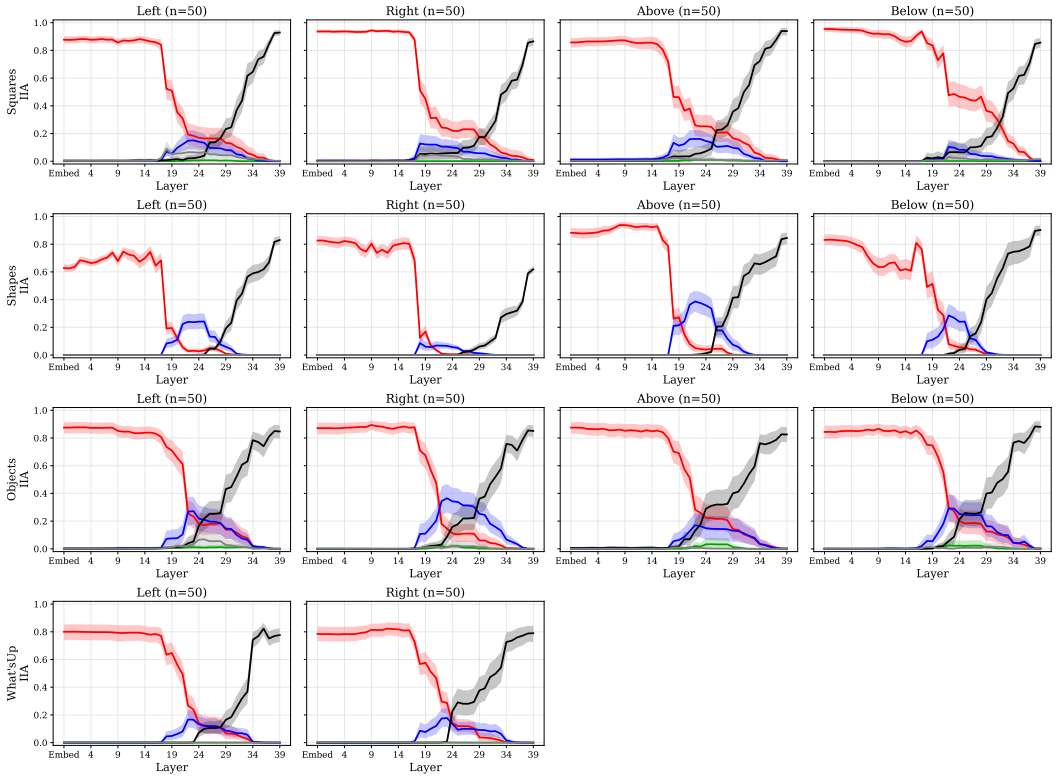


Figure 17: Final token interchange intervention experiments with the `Pixtral-12B` model (n=50 pairs; see text).

C.2 Removing Ordering-Information from Visual Embeddings Impact on Behavioral Performance

We report the behavioral performance of the VLMs after removing the ordering information encoded by the vision encoder. We observe a significant drop, demonstrating its vital role in the spatial reasoning capabilities of VLMs.

Table 6: Behavioral performance after removing spatial information from vision embeddings (chance is 0.33). Pixtral is evaluated in bf16.

	Squares	Shapes	Objects
Qwen2-VL-7B-Instruct	0.60	0.62	0.43
Qwen2-VL-2B-Instruct	0.62	0.51	0.19
Gemma-3-4b-it	0.57	0.64	0.54
Pixtral-12B	0.41	0.78	0.42

C.3 Vision Token Patching

In this subsection, we present the results of the interchange intervention experiment described in Section 5.2.2. We report results for interventions that patch both the square and strip visual tokens for each model. The findings are shown in Fig. 19, Fig. 21, Fig. 18, and Fig. 20. In both cases, the results suggest that ordering information is not sufficiently encoded in the square visual token embeddings. Figs. 24–29 report the same two interventions for `Qwen2-VL-2B-Instruct`, `Qwen3-VL-8B-Instruct`, and `Pixtral-12B`. Strip patching flips the prediction in all three models while square-only patching does not, but the layer at which the flip appears differs across models, and comparing it with the vision-ablated experiment of Sec. 5.3.2 distinguishes two cases.

In `Qwen3-VL-8B-Instruct`, patching the projected visual embeddings alone has no effect, and the flip appears from LM layer 1 onward (Fig. 27). This does not indicate that the LM backbone constructs the ordering information itself: when the vision-derived ordering information is ablated (Fig. 30, bottom rows), the LM backbone's own ordering representation flips the prediction only in layers 12–23, well after the layer-1 onset of the strip-patching flip. The ordering information therefore still originates in the vision encoder; it enters the LM backbone through its first layers rather than through the input embeddings alone, consistent with Qwen3-VL's DeepStack design, which injects intermediate vision-encoder features into the early LM layers in addition to the projected embeddings.

In `Pixtral-12B`, by contrast, neither the projected embeddings nor the first seven LM layers carry a transferable ordering signal: strip patching flips the prediction only in LM layers 8–17 (Fig. 29; cleanly for the horizontal arrangements and for Shapes, weakly for the remaining vertical arrangements), and this is the same window in which square-token patching flips the prediction after the vision-side ablation (Fig. 31). In Pixtral the LM backbone thus appears to construct the ordering information itself, with little contribution from the vision encoder. This may also explain Pixtral's low accuracy on the synthetic tasks (Table 2): without a vision-derived ordering signal the model has to rely on the weaker LM-side mechanism, and ablating the vision-side ordering information does not consistently reduce its accuracy (Table 6).

All curves are means over n=100 pairs with 95% bootstrap confidence bands (Pixtral: original n=50 runs).

In Fig. 23 and Fig. 22, we present additional patching results after removing vision encoder-generated ordering information from visual embeddings using the procedure from Section 5.3.1. Output switching occurs in intermediate layers, corroborating our finding that the LM backbone produces its own ordering information. Figs. 30 and 31 show the same experiment for the remaining models. Under the ablation, the per-direction n (panel titles) is capped by the accuracy of the ablated model, so several cells have $n \ll 100$ and correspondingly wide confidence bands; cells with no valid pairs are left empty.

Results are omitted when sufficient clean/counterfactual input pairs cannot be generated.

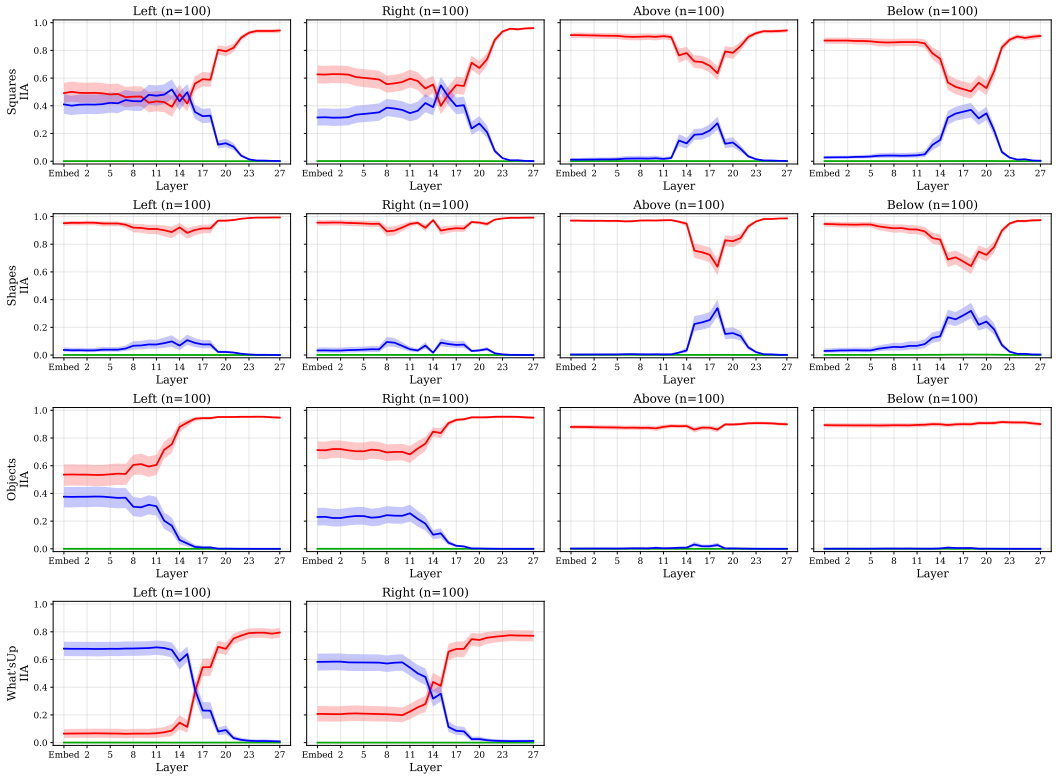


Figure 18: Visual token embeddings of square tokens interchange intervention experiment on `Qwen2-VL-7B-Instruct` model.

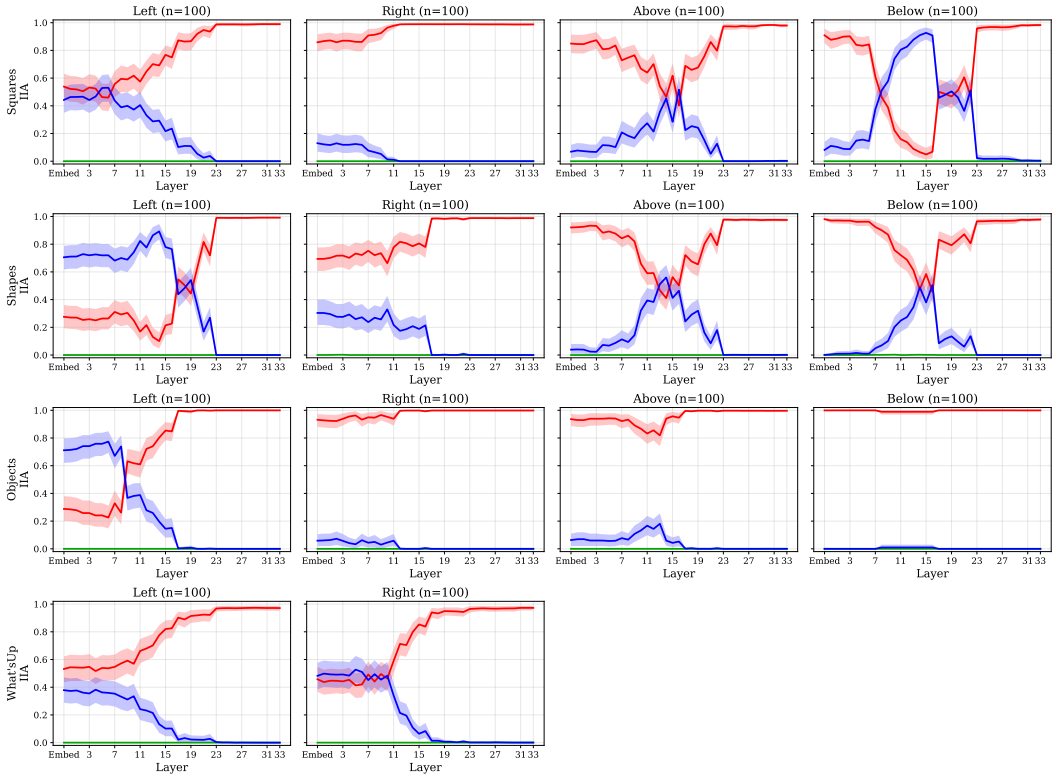


Figure 19: Visual token embeddings of square tokens interchange intervention experiment on `Gemma-3-4b-it` model.

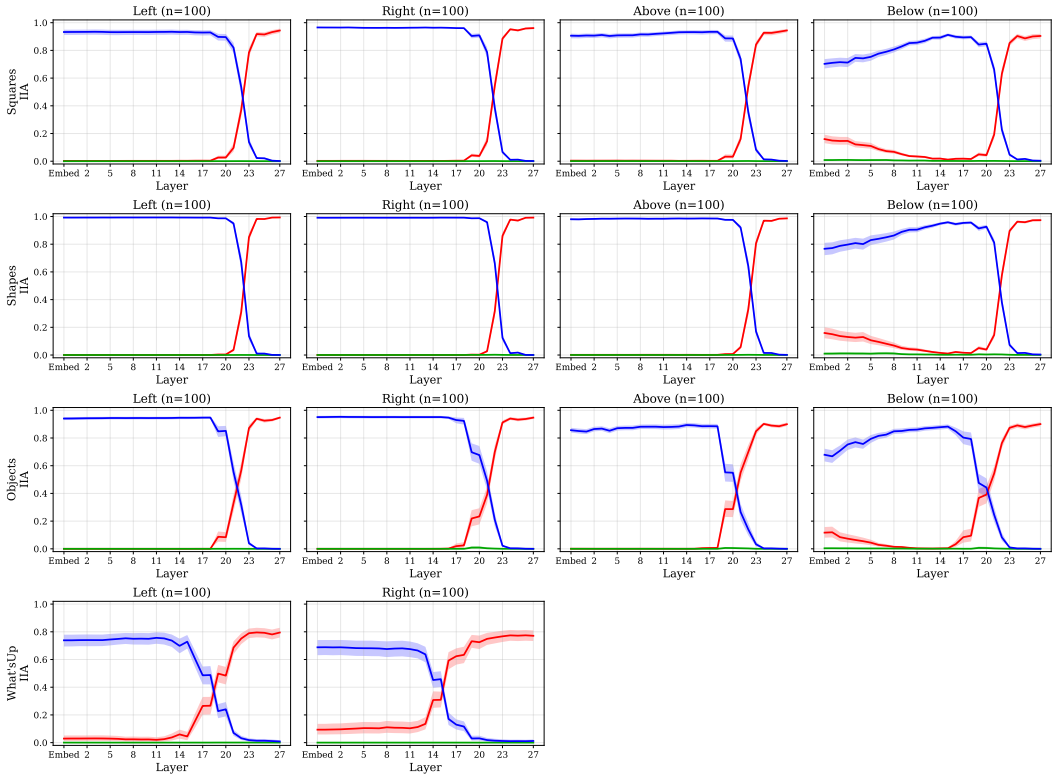


Figure 20: Visual token embeddings of strip tokens interchange intervention experiment on `Qwen2-VL-7B-Instruct` model.

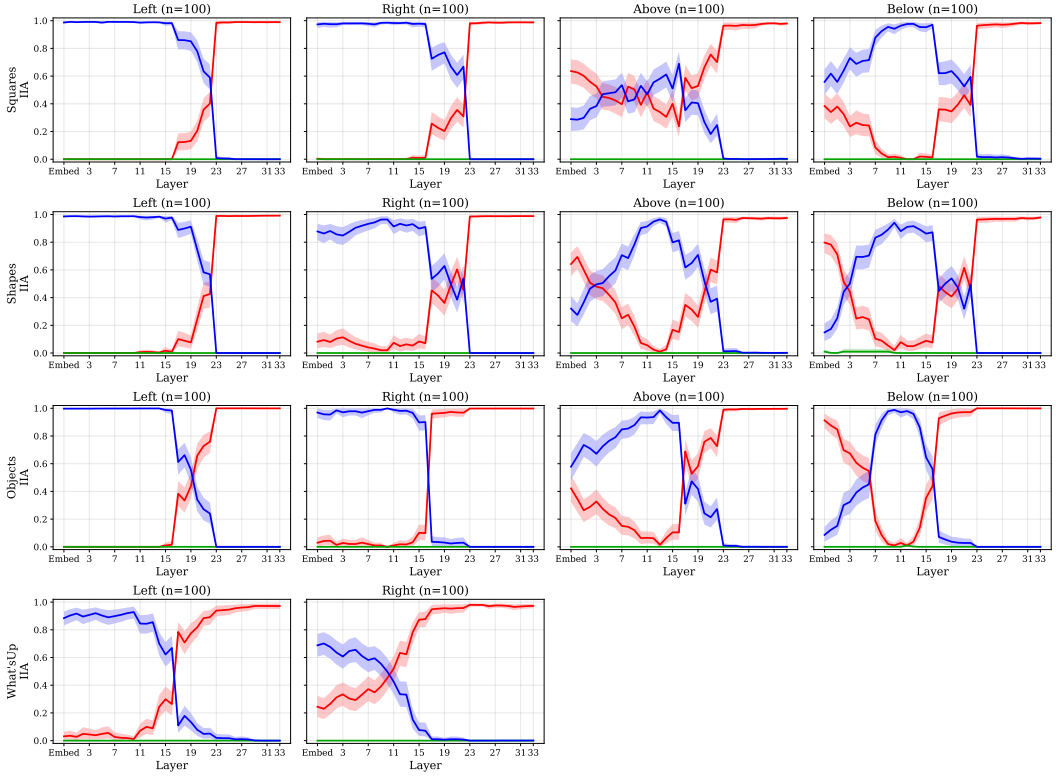


Figure 21: Visual token embeddings of strip tokens interchange intervention experiment on `Gemma-3-4b-it` model.

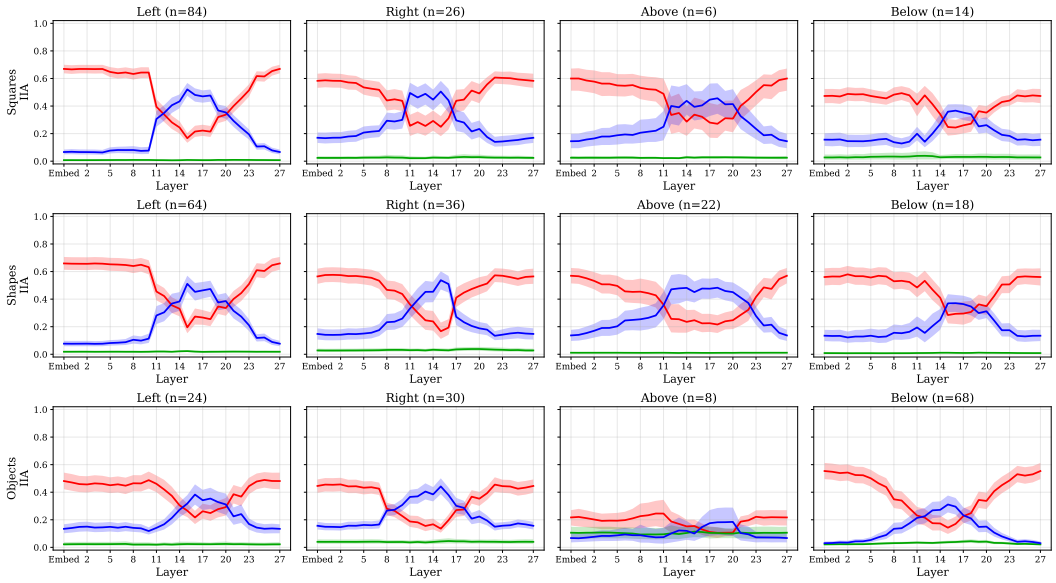


Figure 22: Square-token patching after removing vision-derived ordering information, `Qwen2-VL-7B-Instruct`.

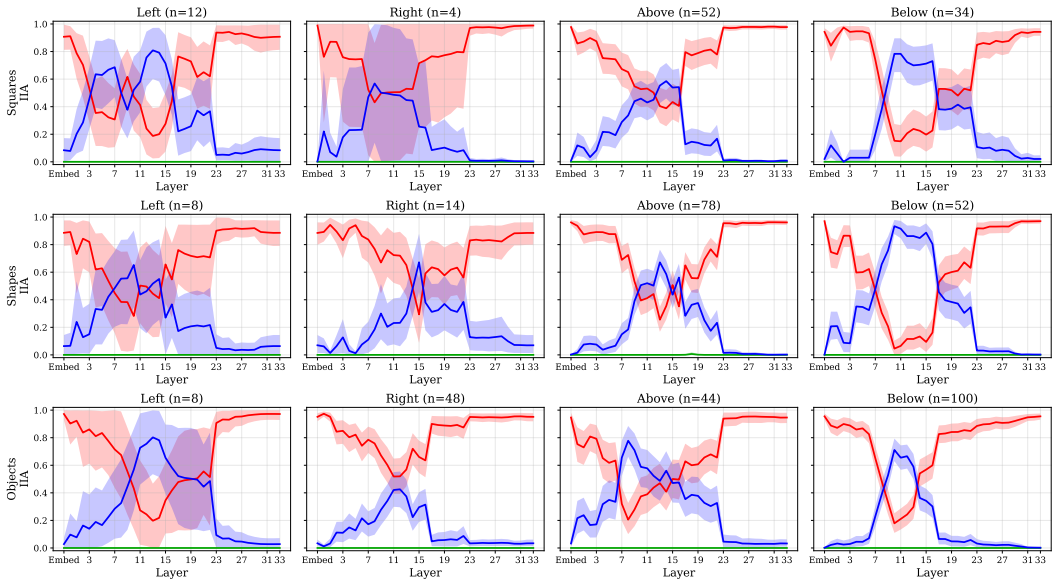


Figure 23: Square-token patching after removing vision-derived ordering information, `Gemma-3-4b-it`.

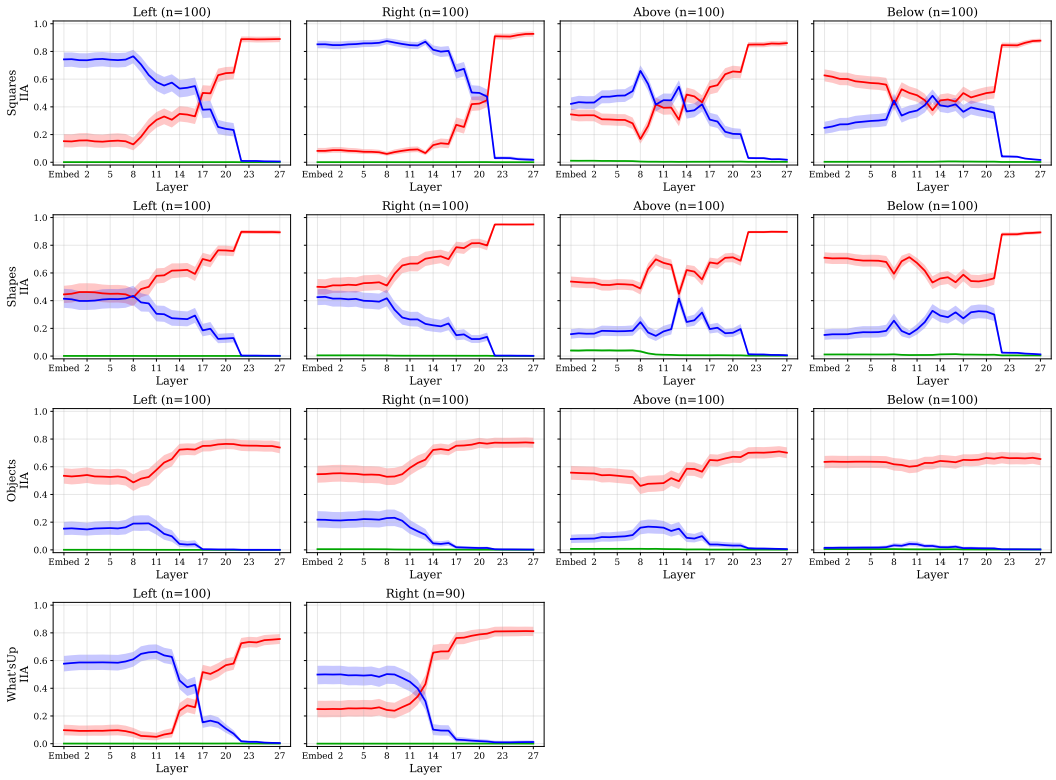


Figure 24: Visual token embeddings of square (object) tokens interchange intervention experiment on the `Qwen2-VL-2B-Instruct` model.

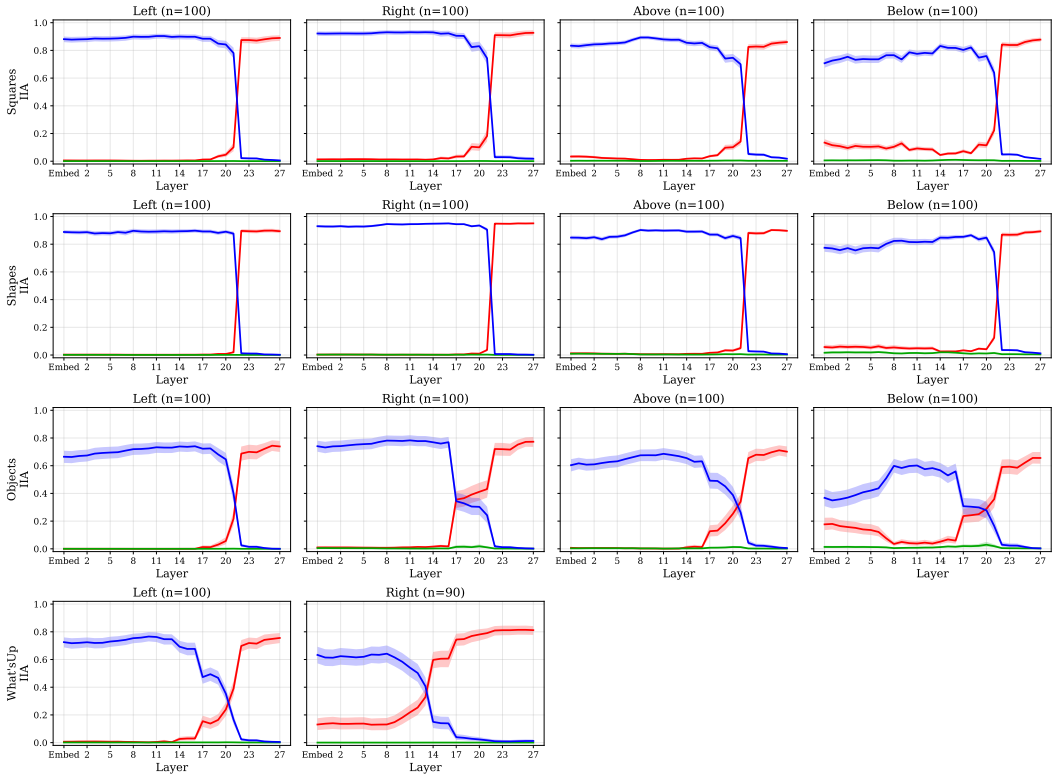


Figure 25: Visual token embeddings of strip tokens interchange intervention experiment on the `Qwen2-VL-2B-Instruct` model.

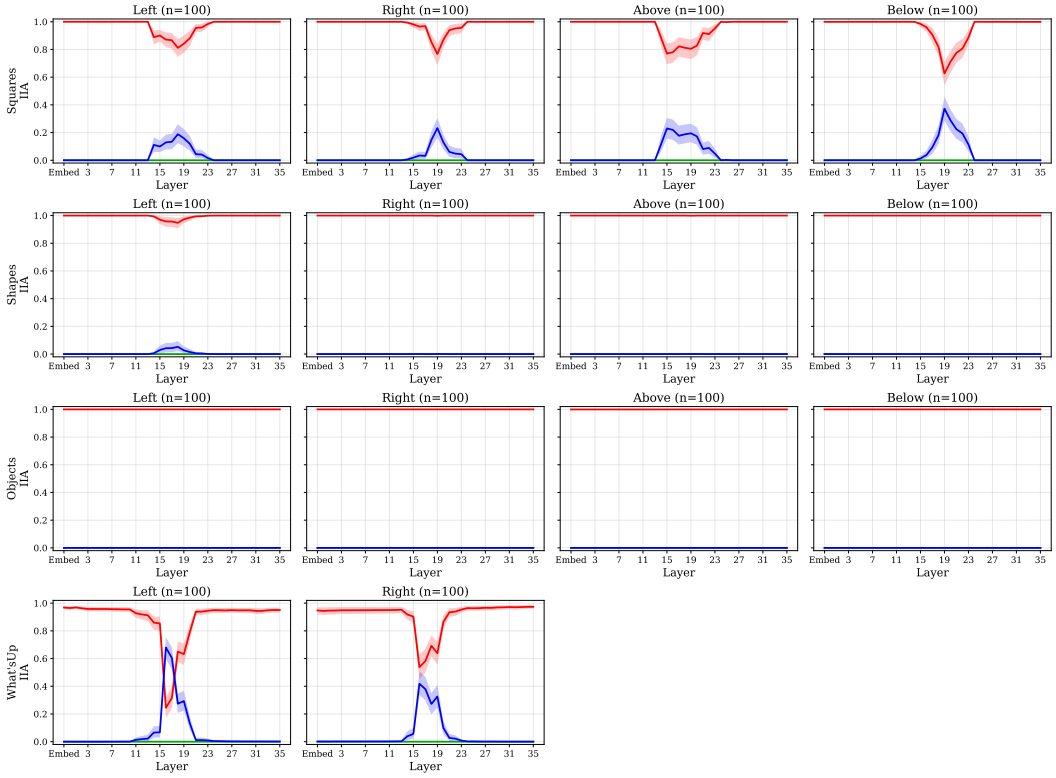


Figure 26: Visual token embeddings of square (object) tokens interchange intervention experiment on the `Qwen3-VL-8B-Instruct` model.

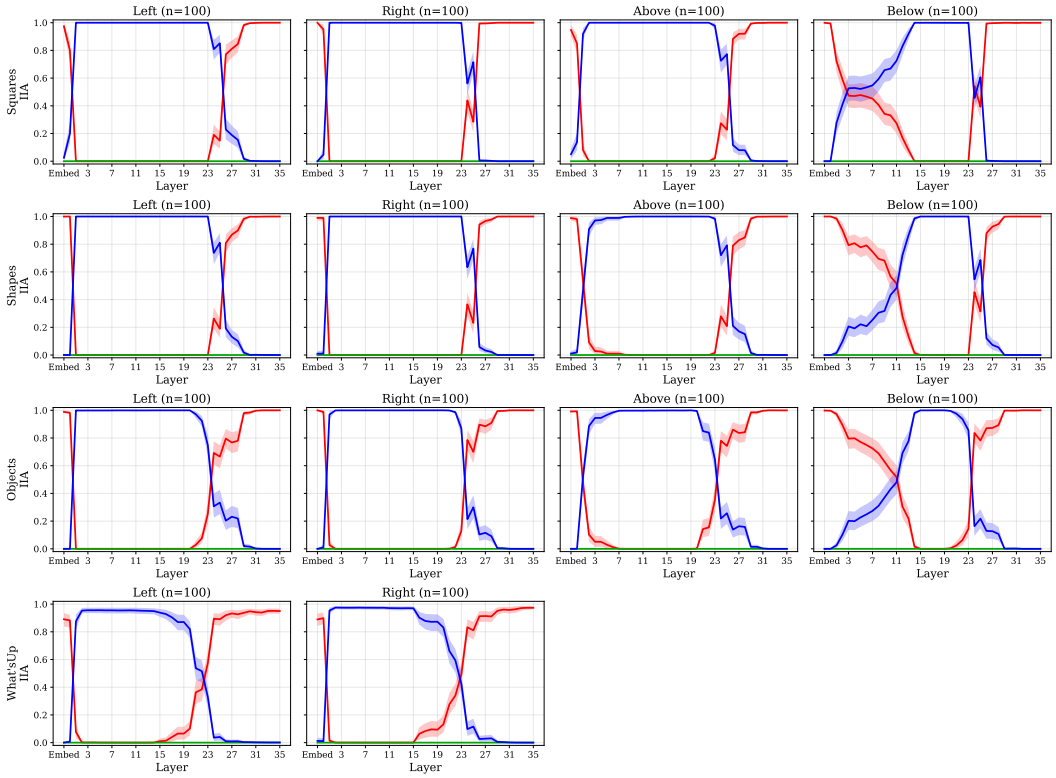


Figure 27: Visual token embeddings of strip tokens interchange intervention experiment on the `Qwen3-VL-8B-Instruct` model.

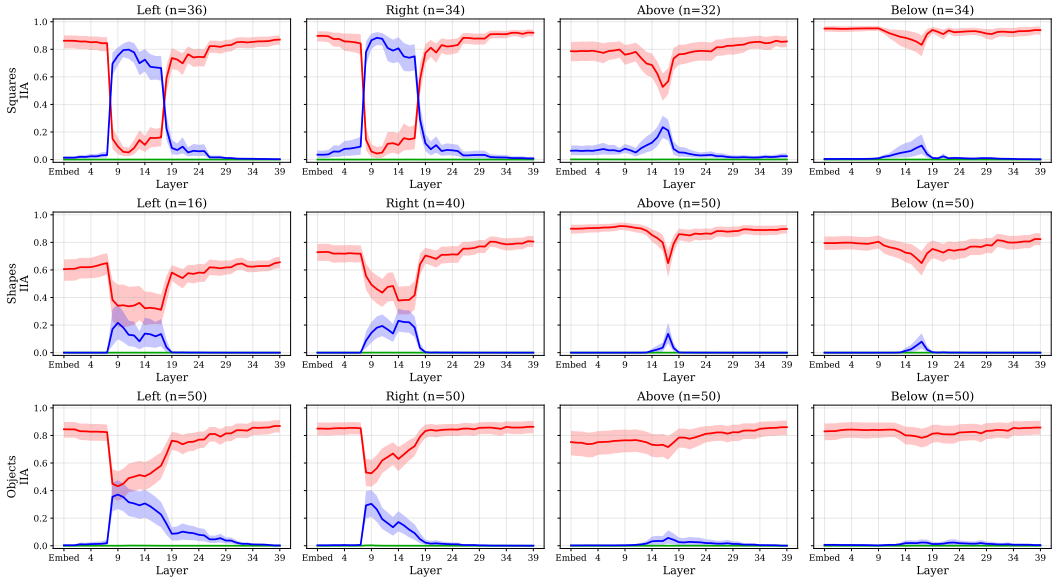


Figure 28: Visual token embeddings of square (object) tokens interchange intervention experiment on the `Pixtral-12B` model (n=50).

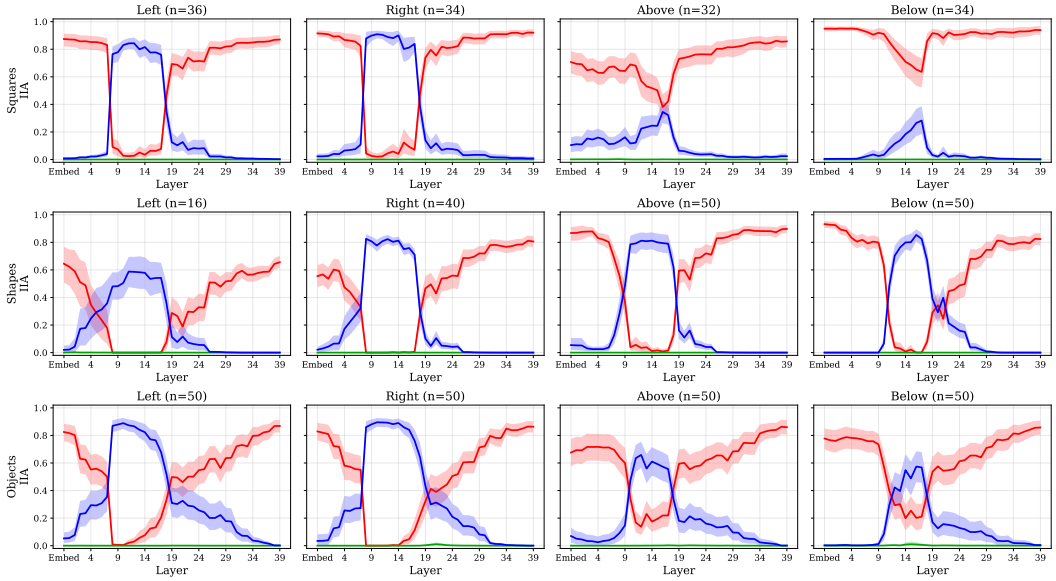


Figure 29: Visual token embeddings of strip tokens interchange intervention experiment on the `Pixtral-12B` model (n=50).

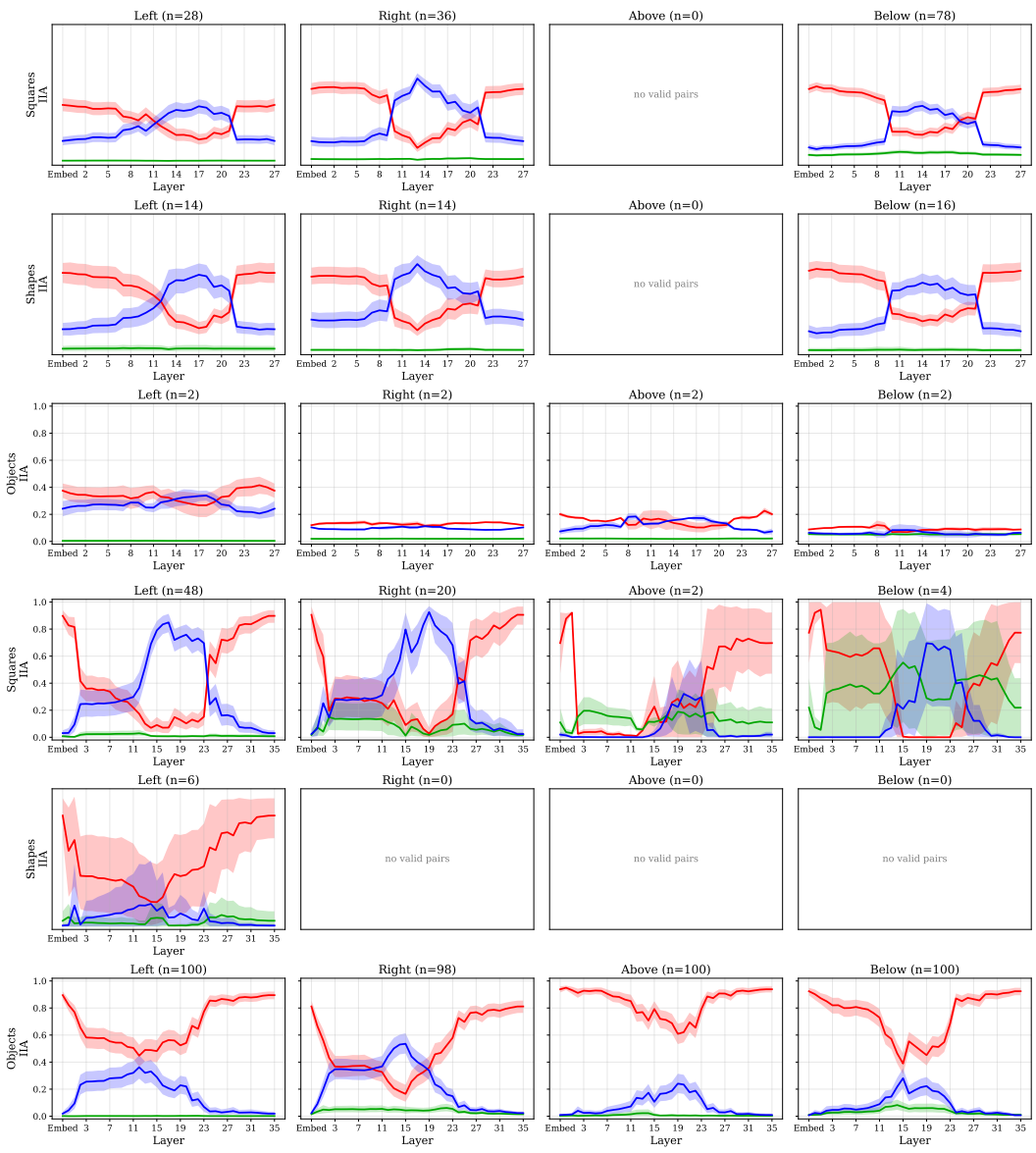


Figure 30: Square-token patching after removing vision-derived ordering information for `Qwen2-VL-2B-Instruct` (top three rows) and `Qwen3-VL-8B-Instruct` (bottom three rows). Empty panels are directions for which the ablated model answers no clean–counterfactual pair correctly.

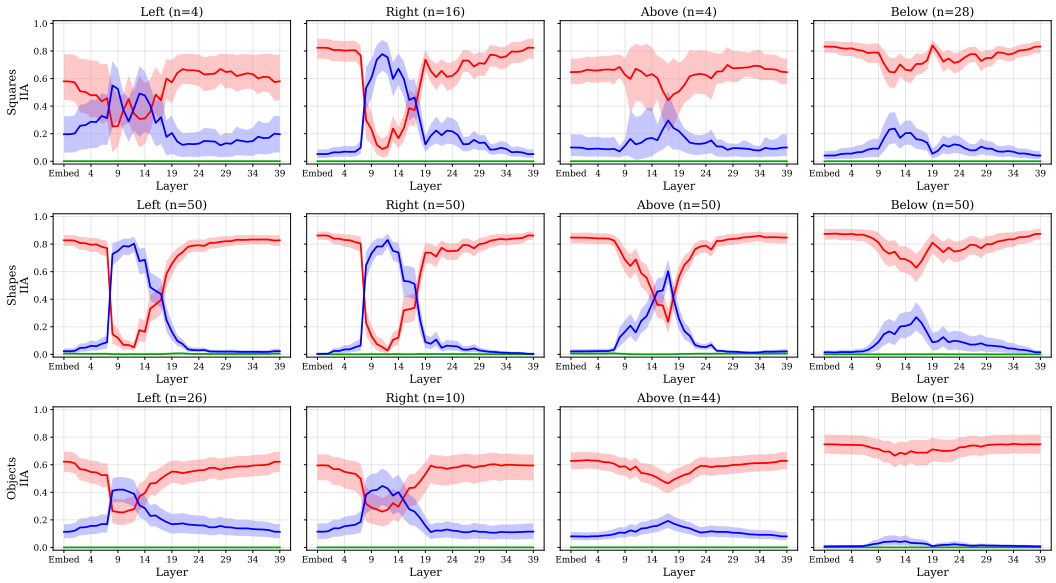


Figure 31: Square-token patching after removing vision-derived ordering information, `Pixtral-12B` (n=50 runs; the per-direction n in the panel titles is capped by the accuracy of the ablated model and is very small for several cells). The flip occurs in LM layers 8–17, the same window as the strip-patching flip in Fig. 29.

C.4 Sample Size and Statistical Significance

All interchange-intervention curves in this paper use n=100 clean–counterfactual pairs per direction and setting, with 95% percentile-bootstrap confidence bands (B=2000 resamples over pairs). An earlier version of these experiments used n=50 pairs. Because pairs are kept only when the model answers *both* the clean and the counterfactual prompt correctly, the achievable n is bounded by model accuracy; the ordering-ablation experiment of Sec. 5.3.2 and several Pixtral cells therefore have $n < 100$ (reported in panel titles).

To check that the sample size does not change the conclusions, we compared the n=50 and n=100 estimates cell by cell. For each (experiment, model, setting, direction) we take the peak-layer effect, $P(\text{counterfactual answer}) - P(\text{clean answer})$ at the layer where it is maximal, and test whether it differs between the two sample sizes with a two-sided permutation test (10^4 shuffles). Where the n=100 pairs are a superset of the n=50 pairs, we test the original 50 against the 50 added pairs; otherwise the two samples are treated as independent draws. Table 7 summarizes the 132 direction cells (119 testable; 13 ablation cells admitted no new pairs). The median absolute change in the peak effect is 0.010; 5 of 119 tests reach $p < 0.05$, matching the ≈ 6 expected by chance, none survives Benjamini–Hochberg correction, and 129 of 132 cells have overlapping 95% confidence intervals. The n=100 results thus reproduce the n=50 results and merely tighten the intervals.

Table 7: n=50 vs. n=100 peak-layer effects. $|\Delta|$: absolute change in the peak effect; $p < .05$: cells with an uncorrected permutation $p < 0.05$; $q < .05$: cells significant after Benjamini–Hochberg correction across all 119 tests; overlap: cells whose 95% CIs overlap.

Experiment	Model	Cells (tested)	Median $\|\Delta\|$	Max $\|\Delta\|$	$p < .05$ / $q < .05$	CI overlap
Last-token patching (Fig. 3)	Qwen2-VL-7B	14 (14)	0.009	0.018	1 / 0	14/14
	Gemma-3-4b	14 (14)	0.003	0.013	0 / 0	13/14
Object patching (Fig. 6)	Qwen2-VL-7B	14 (14)	0.044	0.190	0 / 0	14/14
	Gemma-3-4b	14 (14)	0.061	0.234	2 / 0	13/14
Strip patching (Fig. 7)	Qwen2-VL-7B	14 (14)	0.005	0.041	1 / 0	13/14
	Gemma-3-4b	14 (14)	0.012	0.119	0 / 0	14/14
Ablated-vision patching (Fig. 9)	Qwen2-VL-7B	12 (5)	0.000	0.027	0 / 0	12/12
	Gemma-3-4b	12 (6)	0.001	0.140	1 / 0	12/12
Absolute vs. relative (App. C.6.2)	Qwen2-VL-7B	12 (12)	0.007	0.079	0 / 0	12/12
	Gemma-3-4b	12 (12)	0.032	0.117	0 / 0	12/12
All		132 (119)	0.010	0.234	5 / 0	129/132

C.5 Diagonal Arrangements

The main text studies axis-aligned relations (left/right, above/below). To test whether the same mechanism handles relations that are not aligned with the token grid, we construct two additional Squares settings in which the three squares lie on the principal (top-left to bottom-right) or the secondary (top-right to bottom-left) diagonal of the image, and query them with compound relations ("above and to the left of", "below and to the right of", "above and to the right of", "below and to the left of"). Strips are defined analogously to the axis-aligned case as diagonal bands (the set of tokens with a fixed value of row+column, or row−column, that pass through a square). We run all four causal experiments on `Qwen2-VL-7B-Instruct` with n=100 pairs (Fig. 32) and train the position probes of Sec. 5.2.1 on the diagonal settings (Fig. 33).

The results mirror the axis-aligned settings. Last-token patching shows the same three-phase pattern with an ordering phase around layers 20–22; strip patching flips the prediction from the vision embedding through layer ≈20; square-only patching has a weak effect, as it does for the axis-aligned Squares setting on this model (Fig. 6); and, after ablating the vision-derived ordering information, square-token patching flips the prediction in middle layers, with n reduced to 18–46 by the accuracy of the ablated model. The diagonal probes reach 100% test accuracy, and their predictions again spread along diagonal strips through the squares. The ordering mechanism is therefore not tied to the horizontal and vertical axes of the token grid.

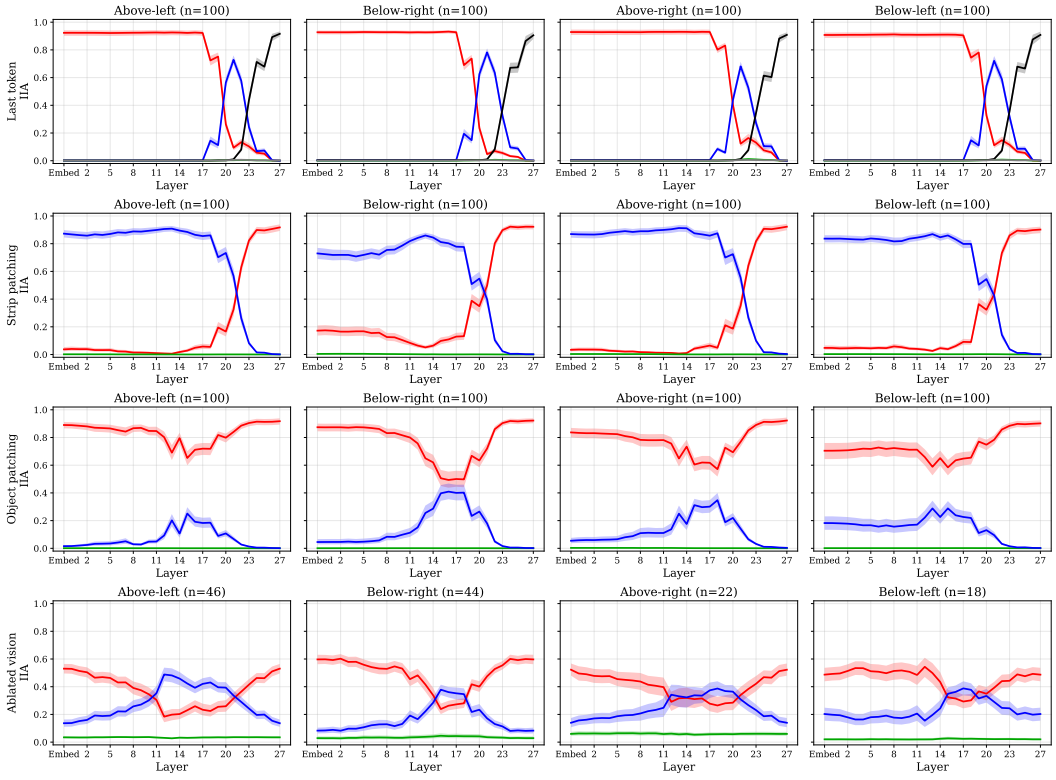


Figure 32: **Diagonal arrangements, `Qwen2-VL-7B-Instruct`** (n=100 pairs, 95% bootstrap CIs). Rows: last-token patching (colors as in Fig. 3), strip patching, square patching, and square patching after the vision-ordering ablation (colors as in Fig. 7). Columns: the two principal-diagonal relations followed by the two secondary-diagonal relations.

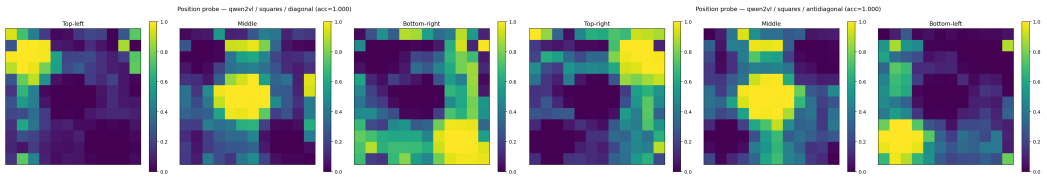


Figure 33: Position-probe predictions on the diagonal (left) and anti-diagonal (right) Squares settings for `Qwen2-VL-7B-Instruct`. As in Fig. 4, probes trained only on square tokens generalize to background tokens along diagonal strips.

C.6 How Are Spatial Positions Represented?

C.6.1 Experimental Settings

Building on the synthetic Squares setting described in Sec. 3.1, we construct two layouts, each containing three equal-sized colored squares with fixed spacing. The two layouts differ only by a shift: one layout is shifted left and the other is shifted right (or up and down), as illustrated in Fig. 34. The layouts are designed such that the rightmost square in the left-shifted layout occupies the same absolute image location as the leftmost square in the right-shifted layout. We refer to the square occupying this shared location as the *overlapping square*.

We label the squares according to their ordinal position within the layout: left (0), middle (1), and right (2). Under this labeling scheme, the overlapping square receives label 2 in the left-shifted layout and label 0 in the right-shifted layout. Thus, the same absolute image location is associated with different ordinal labels depending on the layout.

This construction allows us to distinguish ordinal ordering information from absolute-position information. In particular, because the overlapping square appears at the same absolute image location in both layouts but receives different ordinal labels, its label cannot be determined from absolute position alone. The overlapping square therefore serves as the critical test case for whether vision-encoder representations encode ordinal position within the layout, rather than only fixed image coordinates.

C.6.2 Absolute Position vs. Ordinal Ordering in the Vision Encoder

Probing Vision-Encoder Layers. Following the probing procedure described in Sec. 5.2.1, we train a separate linear probe on the hidden representations produced at each vision-encoder layer in `Gemma-3-4b-it` and `Qwen2-VL-7B-Instruct`. Each probe receives the hidden representation of a visual token corresponding to one of the squares and predicts the square's ordinal position: left, middle, or right. In the extracted vision-encoder hidden states, each square corresponds to a 7×7 token region for Gemma and a 4×4 token region for Qwen.

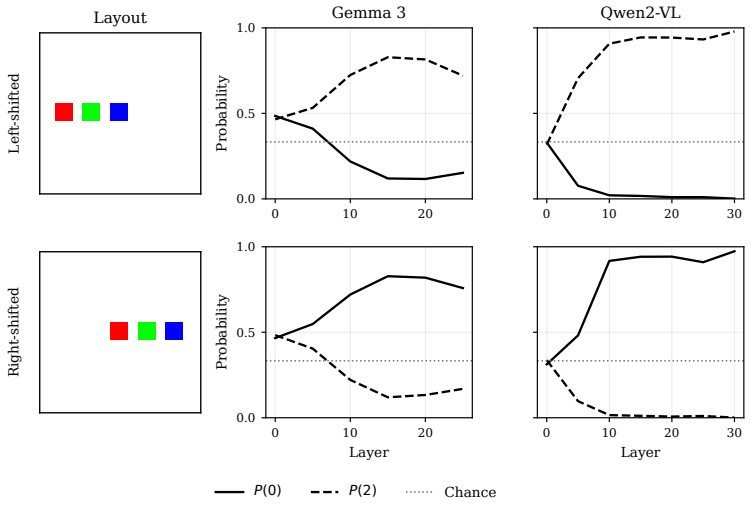


Figure 34: **Training a single probe for both layouts learns ordinal order information.** The overlapping square appears at the same absolute image location in both layouts, while its ordinal label changes from 2 in the left-shifted layout to 0 in the right-shifted layout. We plot the probabilities assigned to the two candidate labels, $P(0)$ (solid) and $P(2)$ (dashed). For this square only, chance is shown as a dotted line. The correct curve is dashed in the top row and solid in the bottom row. Although both squares occupy the same absolute image location, the probe correctly distinguishes their ordinal positions, demonstrating that ordinal information is linearly decodable.

Probing shows the presence of ordinal ordering information. We use 90 images for training and 30 held-out images for evaluation, generated from the square-color assignments described in Sec. 3.1. Each color assignment is rendered in both layouts, and a single three-way linear probe is trained jointly on visual-token representations from both layouts. This joint training ensures that the shared absolute location receives conflicting ordinal labels across the two layouts. For instance, consider the left and right-shifted configurations of the squares depicted in Fig. 34. Because the overlapping location is assigned two different ordinal labels during training, any classifier that relies solely on absolute image coordinates must fail. Successful prediction therefore requires information beyond absolute position.

We evaluate this distinction by reporting, for the overlapping square, the probabilities assigned to its two possible ordinal labels, $P(0)$ and $P(2)$, separately for each layout. The probabilities are averaged over the visual tokens corresponding to the overlapping square and across all held-out images. For completeness, we also report overall probe accuracies over all visual tokens corresponding to each of the squares in Table 8.

As shown in Fig. 34, the probability assigned to the correct layout-dependent label increases across vision-encoder layers in both models. These results show that ordinal ordering information is linearly decodable from the vision-encoder representations of both models.

Table 8: **Overall probe accuracy by layout.** Test accuracy is computed over all visual tokens corresponding to the squares. Mean averages the two layouts.

	Gemma 3			**Qwen2-VL**		
Layer	Left	Right	Mean	Left	Right	Mean
0	84.4	77.3	80.8	25.8	19.5	22.6
5	91.6	93.1	92.4	81.3	81.3	81.3
10	99.6	98.9	99.3	99.9	99.9	99.9
15	99.7	99.5	99.6	99.1	99.8	99.4
20	97.8	98.2	98.0	98.7	99.7	99.2
25	93.9	95.6	94.7	98.4	98.5	98.4
30	–	–	–	100.0	99.9	99.9

Failure to Generalize Across Layouts Reveals Absolute Position Information. We repeat the same experiment; however, this time we train the probe on one layout (e.g., the left-shifted) and test it on the other layout (e.g., right-shifted). We had a total of 90 training examples and 30 eval examples. Results are presented in Fig. 35. If the representation encoded only ordinal position, a probe trained on one layout would generalize perfectly to the other. The observed failure therefore indicates that the probe relies on absolute image coordinates that do not transfer across layouts. Table 9 reports the corresponding train and test accuracies.

Table 9: **Cross-layout probe accuracy across layers.** The probe is trained on square tokens from the left-shifted layout and evaluated on square tokens from the right-shifted layout.

	Gemma 3		**Qwen2-VL**	
Layer	Train	Test	Train	Test
0	100.0	0.0	51.0	17.0
5	100.0	1.0	100.0	48.0
10	100.0	2.0	100.0	49.0
15	100.0	10.0	100.0	36.0
20	100.0	8.0	100.0	59.0
25	100.0	9.0	100.0	60.0
30	–	–	100.0	33.0

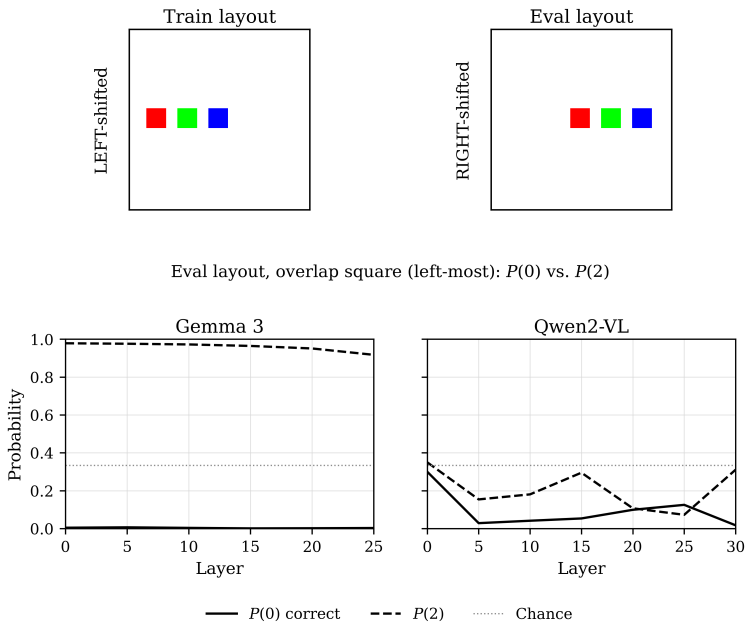


Figure 35: **Training a probe on one layout does not generalize to the other, suggesting the presence of absolute position information.** The overlapping square appears at the same absolute image location in both layouts, while its ordinal label changes from 2 in the left-shifted layout to 0 in the right-shifted layout. We plot the probabilities assigned to the two candidate labels, $P(0)$ (solid) and $P(2)$ (dashed). For this square only, chance is shown as a dotted line. A probe trained on one layout fails to correctly classify the overlapping square in the other layout, indicating that it relies on absolute image coordinates rather than purely ordinal position.

When do the strips emerge? The strip-like probe patterns of Fig. 4 are measured at the output of the vision encoder and projector. To locate where they arise, we train a separate three-way position probe on the object tokens at each vision-encoder layer (Squares setting; 90 training and 30 held-out color permutations) and apply it to all tokens (Fig. 36, Table 10). The two models differ sharply in timing. In `Gemma-3-4b-it`, whose SigLIP encoder adds *learned absolute position embeddings* to the patch embeddings at the input, the layer-0 probe already reaches 100% held-out accuracy and assigns probability 0.90 to the correct label on the background tokens of the object's column, versus 0.20 elsewhere: the strips are present before any attention layer. In `Qwen2-VL-7B-Instruct`, whose encoder has no additive position embedding and injects position only through *2D rotary embeddings* inside attention, the layer-0 probe is at chance (29%) and shows no spatial structure; ordering information and the strips emerge over the first five layers and saturate by layer 10. The overlap-probe accuracies in Table 8 (Gemma 80.8% vs. Qwen 22.6% at layer 0) show the same timing. This is consistent with the strips being a by-product of how position is made available to attention: with absolute position embeddings, every token carries its own coordinates from the input, and tokens that share a row or column coordinate share a component of their embedding; with 2D-RoPE, attention between tokens depends on their relative row and column offsets, so tokens aligned with an object along one axis are the ones that attend to it most similarly, and position information is propagated along rows and columns as attention layers accumulate. In both cases the resulting representation is causally used by the LM (Sec. 5.2.2) and combines absolute and relative components (Fig. 38).

Table 10: Layer-wise vision-encoder probing on the Squares setting (horizontal arrangement; the vertical arrangement behaves the same). A three-way probe is trained on square tokens at each vision-encoder layer. "Square", "Strip", and "Outside" give the mean probability the probe assigns to a label on that label's square tokens, on the background tokens of the same column excluding the square, and on all other tokens, averaged over the three labels and the held-out images.

	Gemma-3-4b-it (SigLIP, absolute pos. emb.)				**Qwen2-VL-7B** (2D-RoPE)			
Layer	Held-out acc.	Square	Strip	Outside	Held-out acc.	Square	Strip	Outside
0	100.0	0.97	0.90	0.20	29.4	0.32	0.33	0.33
5	100.0	0.98	0.67	0.24	91.3	0.68	0.65	0.27
10	100.0	0.98	0.76	0.23	100.0	0.96	0.74	0.25

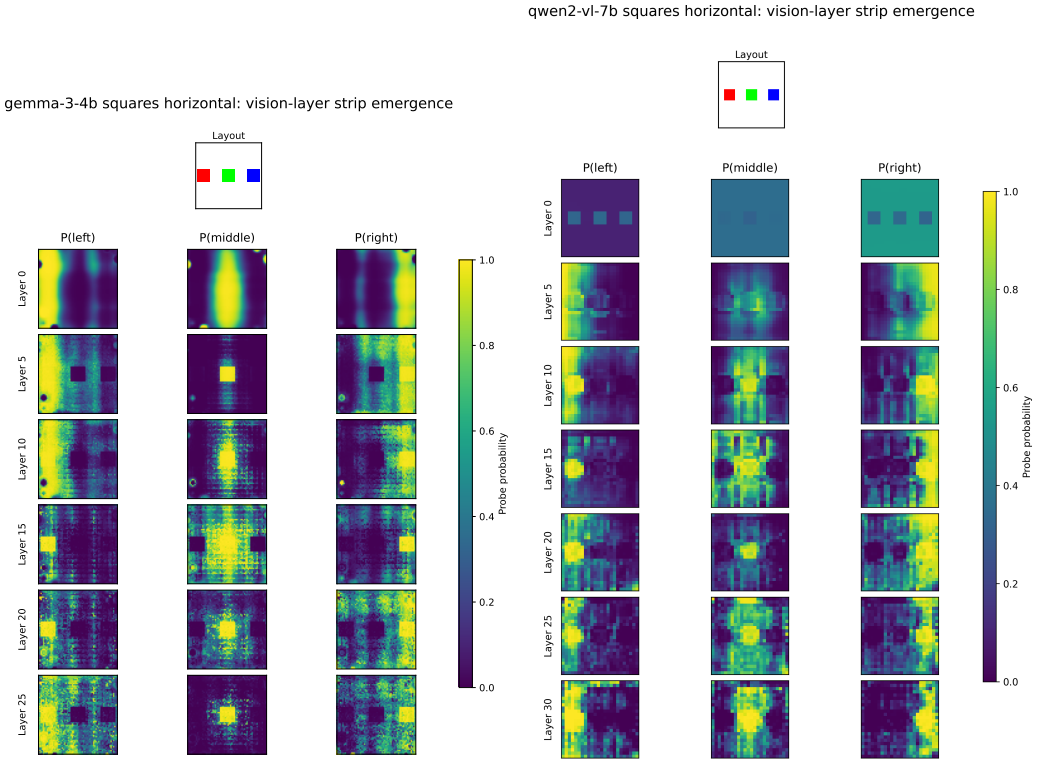


Figure 36: **Strip emergence across vision-encoder layers** (Squares, horizontal). Each row applies the probe trained at that vision-encoder layer to all tokens; columns are the three position labels. Left: `Gemma-3-4b-it`, where the strips are present at layer 0. Right: `Qwen2-VL-7B-Instruct`, where layer 0 carries no position information and the strips form over the first five to ten layers.

Patching Vision Embeddings In addition to the probing experiments, we conducted interchange intervention experiments to further characterize the ordinal information encoded by the vision encoder. We perform these experiments as described in Fig. 37, where both the clean and counterfactual images contain squares shifted in a particular direction, and the question asks for the color of the leftmost, rightmost, topmost, or bottommost square. Specifically, if the clean image contains squares shifted to the left, the counterfactual image shifts the squares to the right while reversing their order. We ensure that the rightmost square in the left-shifted configuration is positioned at the center of the image, and similarly for the other shifted configurations.

The intervention is performed over the entire stripe containing the square of the target color. We perform an interchange intervention by replacing the vision-encoder embeddings of the stripe containing the queried square in the clean image with the corresponding stripe from the mirrored counterfactual. This intervention yields different predictions depending on the representation encoded by the vision encoder. If the vision embeddings encode ordinal information, then after patching, the model output should change to the color of the square at the opposite end of the ordering. For example, after applying the patching shown in Fig. 37, the output should change from blue to red if the vision encoder encodes ordinal information. In contrast, if the vision encoder encodes absolute positional information, then the output should change from blue to green, because the absolute coordinate values of the intervened blue and red squares would be lower than those of the unintervened green square.

Fig. 38 shows the intervention experiment results. Surprisingly, the expected output for both absolute and ordinal information is substantial, indicating that the vision encoder is forming both types of positional information to encode spatial information. The exception is the vertical configuration with the Gemma model, where patching the output of the vision encoder does not change the final output, indicating a lack of ordering information, consistent with results in Fig. 21.

However, when the same experiment is performed on the LM-backbone ordering information using the setup described in Section 5.3.2, as shown in Fig. 39, we primarily observe the expected output corresponding to the relative ordering information. This suggests that the ordering information encoded by the LM backbone is predominantly relative in nature, consistent with findings from prior studies on LMs (Prakash et al., 2024; 2025; Assouel et al., 2025).

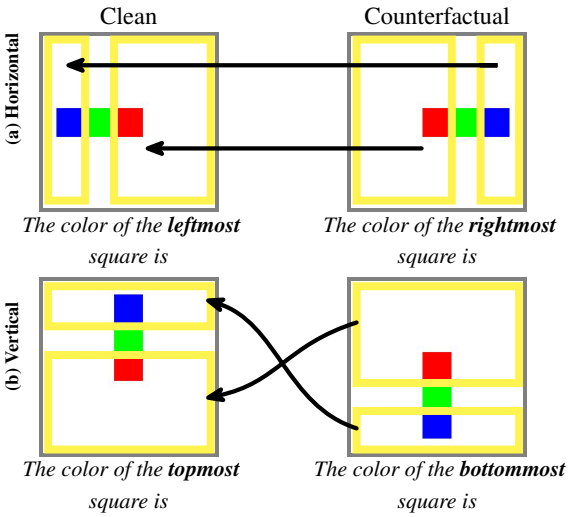


Figure 37

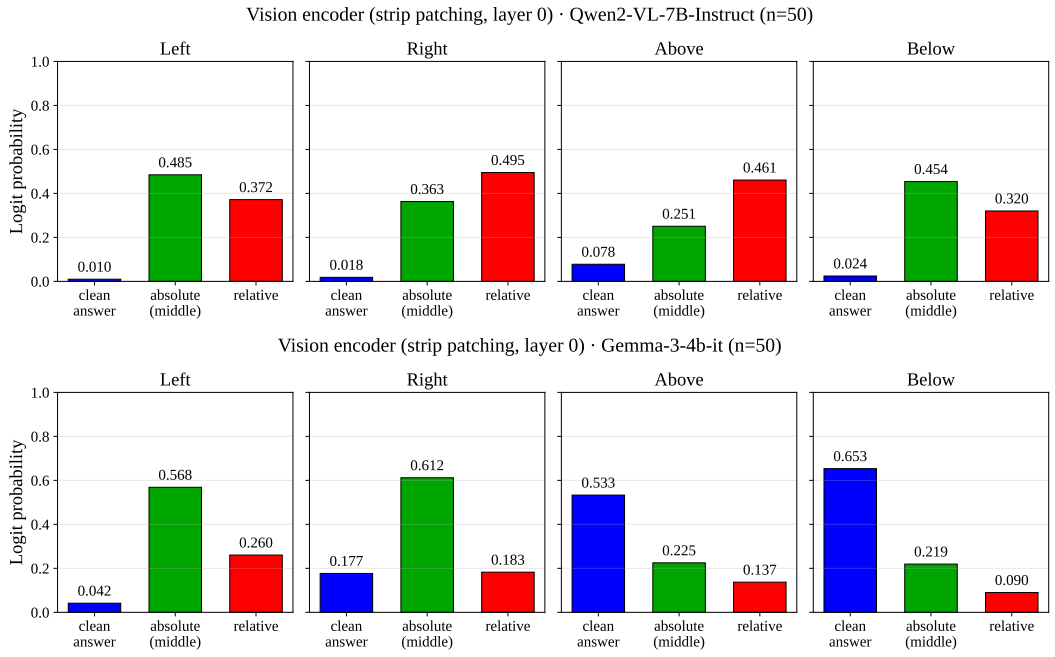


Figure 38: **Label probabilities after the intervention**: Intervening on the visual token embeddings generated by the vision encoder using the mirrored counterfactual produces substantial label probabilities for both the absolute and relative expected outputs.

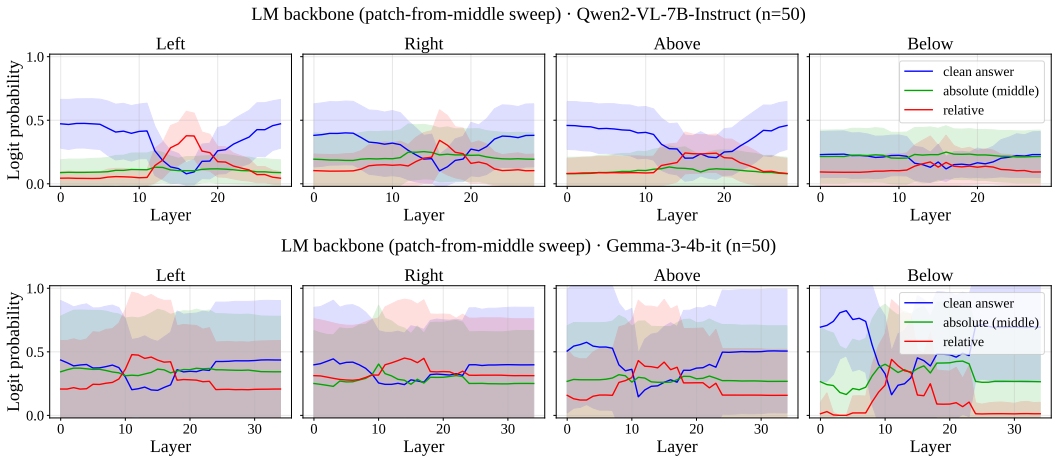


Figure 39: **Label probabilities after intervening on the LM backbone's ordering information**: Repeating the mirrored-counterfactual intervention under the vision-ordering ablation of Sec. 5.3.2, where only the LM backbone's own ordering representation can be transferred, produces the relative-order answer almost exclusively, with little probability on the absolute-position answer.

Per-layer curves with confidence intervals. Figs. 40–42 show the full per-layer curves underlying Figs. 38 and 39 (n=100 pairs, 95% bootstrap CIs), for the strip swap read through the LM (top row), the LM-backbone variant under the vision ablation (middle row), and the square-only swap (bottom row), for `Qwen2-VL-7B-Instruct`, `Gemma-3-4b-it`, and `Qwen2-VL-2B-Instruct`. The vision-encoder swap yields substantial probability for both the absolute and the relative answer, whereas the LM-backbone variant yields the relative answer only.

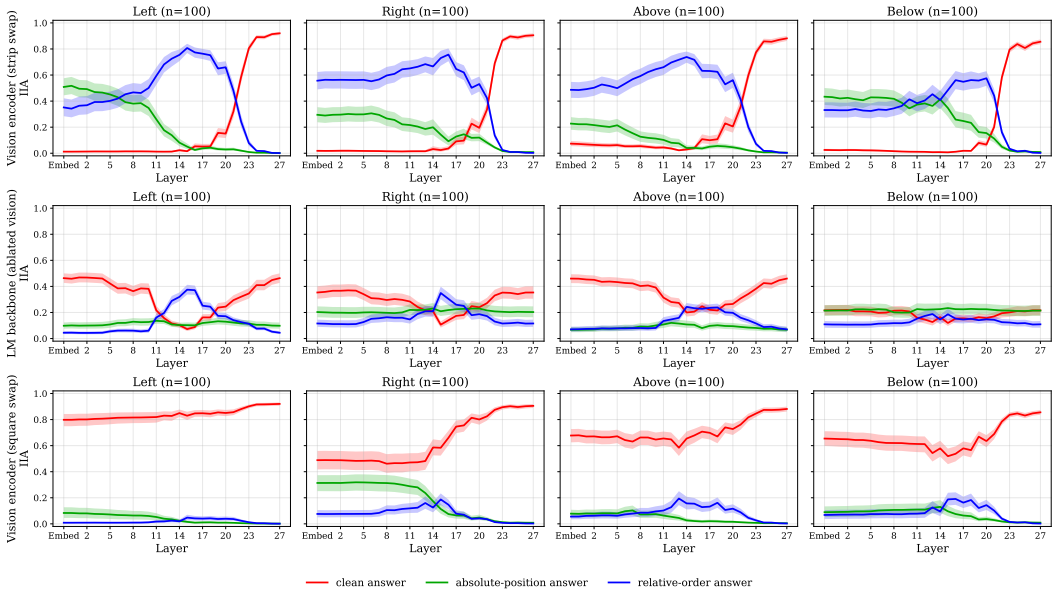


Figure 40: Absolute vs. relative position, per-layer curves, `Qwen2-VL-7B-Instruct`.

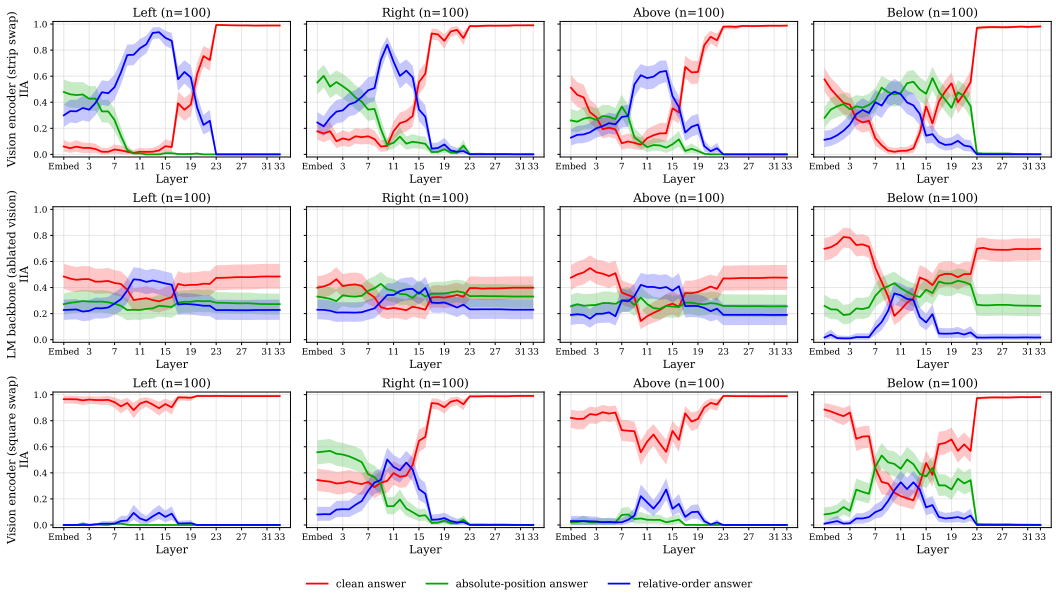


Figure 41: Absolute vs. relative position, per-layer curves, `Gemma-3-4b-it`.

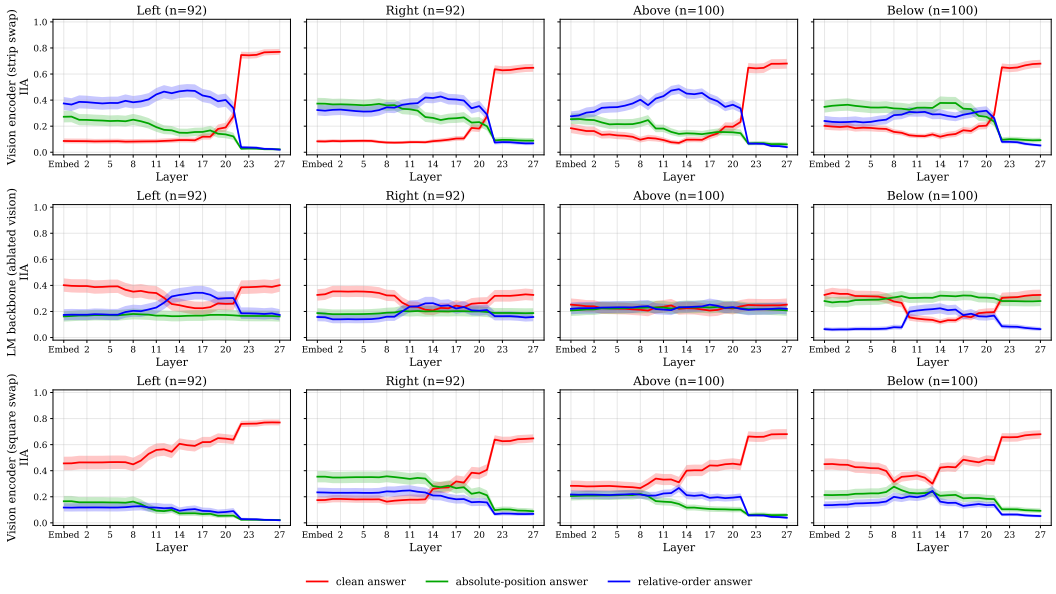


Figure 42: Absolute vs. relative position, per-layer curves, `Qwen2-VL-2B-Instruct`.

C.7 NATURAL-IMAGE VALIDATION – ADDITIONAL RESULTS

Table 11 provides a full breakdown of the amplification results reported in Sec. 5.4, separated by spatial direction, together with the results for `Qwen2-VL-2B-Instruct`, the Qwen model with enough baseline failures for this experiment. Baselines are $232/440$ (Gemma-3-4b, 0.527), $264/440$ (Qwen2-VL-2B, 0.600), and $395/439$ (Pixtral-12B, 0.900; one item is dropped for an out-of-vocabulary prediction). We also ran the protocol on the two larger Qwen models, whose baseline accuracy is already high enough to leave too few failures for meaningful rates, so both are omitted from the tables. `Qwen2-VL-7B-Instruct` answers $268/279 = 96.1\%$ of its COCO-spatial items correctly, leaving 11 strict failures; probe amplification corrects $3/11$ (27.3%, raising accuracy to 97.1%, identically under a single fixed α and per-item α), while random directions correct none. `Qwen3-VL-8B-Instruct` is near ceiling on both datasets ($434/440 = 98.6\%$ on COCO-spatial and $622/625 = 99.5\%$ on What'sUp, leaving 6 and 3 strict failures); probe amplification corrects $4/6$ and $3/3$ of them while random directions correct none. Results are consistent across directions, with probe amplification substantially outperforming random amplification in all cases. The gap between Probe and Probe* indicates room for improvement with better α selection, suggesting the identified ordering subspace is richer than a single fixed coefficient can fully exploit.

C.7.1 COCO-SPATIAL

Table 11: Full breakdown of amplification results of Sec 5.4 by spatial direction. "% Corr. Fail." denotes the fraction of baseline failures corrected by the intervention. In *Random* and *Probe*, the best single fixed α coefficient is chosen across all examples; In *Random** and *Probe**, we search for a per-item optimal coefficient, serving as an upper bound on intervention performance. For Pixtral-12B (n=439; 33 horizontal and 11 vertical failures) the per-axis breakdown of the per-item oracle is not available and is marked "–".

		Gemma3-4B		Qwen2-VL-2B		Pixtral-12B	
Direction	Intervention	Acc.	% Corr. Fail.	Acc.	% Corr. Fail.	Acc.	% Corr. Fail.
Horizontal	None	0.51	–	0.62	–	0.88	–
	Random	0.54	8.0%	0.67	12.3%	0.88	0.0%
	Random*	0.56	10.1%	0.67	12.3%	–	–
	Probe	0.62	23.9%	0.80	47.2%	0.92	30.3%
	Probe*	0.70	39.1%	0.84	58.5%	–	–
Vertical	None	0.56	–	0.57	–	0.93	–
	Random	0.59	7.0%	0.63	14.3%	0.94	9.1%
	Random*	0.60	8.5%	0.63	14.3%	–	–
	Probe	0.70	32.4%	0.72	35.7%	0.94	18.2%
	Probe*	0.75	42.3%	0.78	48.6%	–	–
Overall	None	0.53	–	0.60	–	0.90	–
	Random	0.56	7.7%	0.65	13.1%	0.90	2.3%
	Random*	0.57	9.6%	0.65	13.1%	0.90	2.3%
	Probe	0.65	26.8%	0.77	42.6%	0.93	27.3%
	Probe*	0.72	40.2%	0.82	54.5%	0.95	54.5%

C.7.2 WHAT'SUP

Table 12 reports amplification results on the What'sUp composite dataset, evaluated using the same strict two-ordering protocol as COCO-spatial. Results are consistent with the COCO-spatial findings, with probe amplification consistently outperforming random amplification across all models, confirming that the intervention generalizes across naturalistic settings.

C.7.3 OTHER SPATIAL RELATIONS: ON/UNDER AND FRONT/BEHIND

The main experiments use left/right (What'sUp) and left/right plus above/below (COCO-spatial). We apply the same protocol to the original, unstitched What'sUp images (examples in Fig. 43) for two further relation pairs: *on/under* (161 images; labels mapped to the spatial answers `above`/`below`) and *front/behind* (204 images; answers `front`/`behind`, prompt "Look at the {reference}, is the

Table 12: Amplification results on the What'sUp dataset using the strict two-ordering evaluation protocol. "% Corr. Fail." denotes the fraction of baseline failures corrected by the intervention. In *Random* and *Probe*, the best single fixed α coefficient is chosen across all examples; In *Random** and *Probe**, we search for a per-item optimal coefficient, serving as an upper bound on intervention performance. Pixtral-12B is evaluated on 484 composite queries (15 strict failures).

	Gemma3-4b		Qwen2-VL-2B		Pixtral-12B	
Intervention	Acc.	% Corr. Fail.	Acc.	% Corr. Fail.	Acc.	% Corr. Fail.
None	0.79	–	0.82	–	0.97	–
Random	0.83	19.1%	0.84	9.9%	0.97	6.7%
Random*	0.84	23.2%	0.85	13.5%	0.97	6.7%
Probe	0.91	55.2%	0.90	43.2%	0.98	40.0%
Probe*	0.93	68.0%	0.91	47.7%	0.99	53.3%

{target} in front of it, or behind it?"). Object boxes come from DINO-assisted detection filtered to the query objects; probes are trained on strict-correct images only, and amplification is applied to all image tokens of strict-wrong images. Table 13 reports fixed-α and per-item oracle fix rates, bootstrap CIs, and the held-out fix rate under the validation/test α-selection protocol of App. A.4. In all three cells the probe direction corrects the majority of failures and far exceeds the random-direction control, including on front/behind, a depth relation not studied mechanistically in this paper. Gemma-3's on/under failures are almost exclusively `on` items (53 of 55), so its probe direction is effectively tested on one label.



dog *on* table | dog *under* table | bowl *in front of* sunglasses | cup *in front of* spoon

Figure 43: **Examples of the original What'sUp images used for the on/under and front/behind relations** (Kamath et al., 2023). Each image contains a reference object (table, sunglasses, spoon) and a target object placed on or under it (left two) or in front of it (right two).

Table 13: Amplification on additional What'sUp relations. Fix rates are over strict-wrong images (95% bootstrap CIs over the failures); "Acc." is strict accuracy on all images after amplification. Held-out: α selected on validation failures and evaluated on 20 held-out failures. Front/behind for Gemma-3 was not run.

Relation	Model	Baseline	Method	α	Fix rate [95% CI]	Acc.	Held-out fix
on/under	Qwen2-VL-2B	72.7%	Random	7	13.6% [4.5, 25.0]	76.4%	15.0%
			Probe	8	68.2% [52.3, 81.8]	91.3%	65.0%
	Gemma-3-4b	65.8%	Random	14	16.4% [7.3, 27.3]	71.4%	10.0%
			Probe	12	52.7% [40.0, 65.5]	83.9%	45.0%
front/behind	Qwen2-VL-2B	73.0%	Random	15	27.3% [16.4, 40.0]	80.4%	30.0%
			Probe	12	76.4% [63.6, 87.3]	93.6%	70.0%

C.7.4 Probe transfer across datasets and axes

If the probe directions capture a general ordering representation rather than dataset-specific features, a probe trained on one dataset or axis should still correct failures on another. We test four transfers on `Qwen2-VL-2B-Instruct` (Table 14): (i) the What'sUp `left/middle/right` probe applied to COCO-spatial horizontal failures (using its `left/right` columns); (ii) the COCO

horizontal probe applied to What'sUp failures; (iii) the COCO horizontal probe applied to COCO *vertical* failures, reading `left` as `above` and `right` as `below`; and (iv) a single four-class `left`/`right`/`above`/`below` probe trained on both COCO axes and applied to all COCO failures. In every case the transferred probe corrects far more failures than random directions, and in (i) and (iv) it approaches the in-domain probe. Under the held-out α protocol of App. A.4, the What'sUp→COCO probe corrects 46.9% of held-out failures versus 6.2% for random directions. The horizontal→vertical transfer is the most surprising: the horizontal probe classifies vertical object tokens only at 60%, yet amplifying it corrects a third of the vertical failures, suggesting a shared "first/second" ordering component across axes.

Table 14: Probe transfer on `Qwen2-VL-2B-Instruct`. Probe accuracy is given on the source train / source test / target tokens. Fix rates are over the target's strict failures with the best fixed α (*: per-item oracle). The in-domain column repeats the fixed-α result of a probe trained on the target itself (Tables 11 and 12; for (iii) the in-domain vertical probe from the same run).

Transfer	Target failures	Probe acc. (train / test / target)	Random*	Transferred	Transferred*	In-domain
(i) What'sUp → COCO horiz.	106	97.4 / 95.2 / 85.7	16.0%	48.1%	55.7%	47.2%
(ii) COCO horiz. → What'sUp	111	93.7 / 89.1 / 92.4	11.7%	32.4%	34.2%	43.2%
(iii) COCO horiz. → COCO vert.	–	98.9 / 94.1 / 60.0	1.9%	33.3%	33.3%	35.2%
(iv) Shared 4-class → all COCO	176	90.2 / 64.5 / –	22.7%	42.0%	54.0%	42.6%

C.7.5 Comparison to chain-of-thought prompting

A natural question is whether the failures corrected by ordering amplification could be recovered simply by prompting. We evaluate the unmodified models on the same COCO-spatial items under a chain-of-thought (CoT) prompt that asks the model to reason step by step and end with "`Answer: <word>`" (greedy decoding), using the same strict two-ordering protocol. Unlike the amplification sweep, which re-evaluates only the baseline failures and assumes baseline-correct items are kept, CoT changes the prompt for every item, so we re-evaluate the full set; for comparison we also report the amplification at its operating point (α=6) re-evaluated on the full set. Table 15 shows that the effect of CoT is model-dependent: it helps `Qwen2-VL-2B` (0.600→0.752), correcting many failures but also breaking three times as many previously correct items as the amplification does, and it is *net harmful* for `Gemma-3-4b` (0.527→0.464), where it breaks 44% of previously correct items and vertical accuracy collapses from 0.559 to 0.385. Inspecting Gemma's generations, the spatial percepts in the reasoning are often correct, but the final answer is bound to the wrong referent (e.g., "the giraffe is clearly above the bird → Answer: above" when asked about the bird). CoT thus reshuffles which items are answered correctly rather than supplying ordering information, and, unlike the amplification, it provides no causal handle on the mechanism.

Table 15: Chain-of-thought prompting vs. ordering amplification on COCO-spatial (strict two-ordering accuracy on all 440 items; chance = 0.25). "Fix": fraction of baseline failures corrected; "break": fraction of baseline-correct items made incorrect. Qwen2-VL-2B is evaluated in the environment of Table 1 (`transformers` 4.47.1).

	Qwen2-VL-2B			Gemma-3-4b		
	Acc.	Fix	Break	Acc.	Fix	Break
Baseline (one-word prompt)	0.600	–	–	0.527	–	–
Ordering amp., α=6, full re-evaluation	0.709	36.9%	6.4%	0.536	22.6%	18.5%
Ordering amp., per-item α (Table 1)	0.82	54.5%	–	0.72	40.2%	–
CoT prompting, full re-evaluation	0.752	65.3%	18.2%	0.464	35.6%	44.0%

C.7.6 Side effects on non-spatial VQA

Amplifying the ordering directions perturbs every image token, so it could degrade capabilities unrelated to spatial reasoning. We measure this on seven non-spatial VQA benchmarks (POPE, GQA, MMBench, OK-VQA, VizWiz, AI2D, ScienceQA-IMG; 500 items each) by applying the both-axis amplification at the paper's operating point α=6 and comparing to α=0 (Table 16). For in-domain

COCO-spatial we report the fraction of baseline failures repaired with the axis-specific probe, as in Table 1. At α=6 the worst degradation is -5.0 points (GQA for Qwen2-VL-2B, AI2D for Gemma-3), while the in-domain repair rates are 37.5% and 23.1%. Fig. 44 shows the full α sweep: the in-domain effect follows an inverted-U with a peak around $\alpha \approx 5$–6, and the non-spatial degradation grows with α, becoming severe only at large α (e.g., -16 points on GQA for Qwen2-VL-2B at α=13). A more conservative operating point, α=3, keeps every non-spatial drop within 2 points for both models while retaining roughly three quarters of the repair effect.

Table 16: Accuracy (%) on non-spatial VQA benchmarks before (α=0) and after (α=6) both-axis ordering amplification; n=500 items per benchmark. The first row gives the in-domain repair rate of COCO-spatial baseline failures.

Benchmark	Qwen2-VL-2B: α=0 → α=6 (Δ)	Gemma-3-4b: α=0 → α=6 (Δ)
COCO-spatial failures repaired (in-domain)	0 → 37.5 (+37.5)	0 → 23.1 (+23.1)
POPE	89.0 → 85.6 (−3.4)	86.0 → 83.2 (−2.8)
GQA	61.6 → 56.6 (−5.0)	37.0 → 38.6 (+1.6)
MMBench	78.2 → 77.4 (−0.8)	65.2 → 64.4 (−0.8)
OK-VQA	59.0 → 58.4 (−0.6)	52.0 → 50.3 (−1.7)
VizWiz	70.1 → 68.7 (−1.4)	48.2 → 44.3 (−3.9)
AI2D	70.2 → 67.4 (−2.8)	58.4 → 53.4 (−5.0)
ScienceQA-IMG	78.2 → 76.4 (−1.8)	75.4 → 71.6 (−3.8)

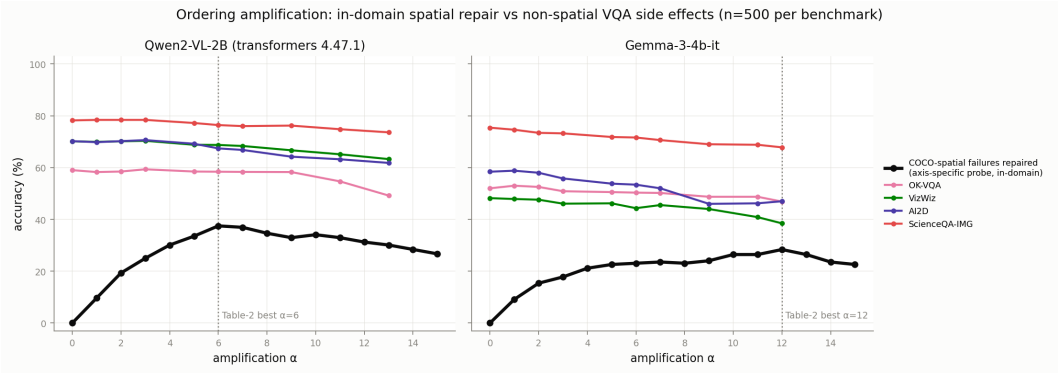


Figure 44: **Dose–response of ordering amplification.** In-domain repair of COCO-spatial failures (axis-specific probe) and accuracy on non-spatial VQA benchmarks as a function of α, for `Qwen2-VL-2B` (left) and `Gemma-3-4b` (right). The paper's operating point is α=6.

C.8 PROBING

One of the main findings of this work is that ordering information from the vision encoder is not concentrated in a small set of square tokens; instead, it is distributed across neighboring background tokens as well. In this subsection, we present probing results across multiple settings and models, demonstrating the ubiquity of this phenomenon of diffused ordering information. Results are demonstrated in Fig. 45-Fig. 50. We also show similar results with `Gemma-3-4b-it` in Fig. 51-Fig. 56, and with `Qwen2-VL-2B-Instruct` and `Qwen3-VL-8B-Instruct` in Figs. 57–60

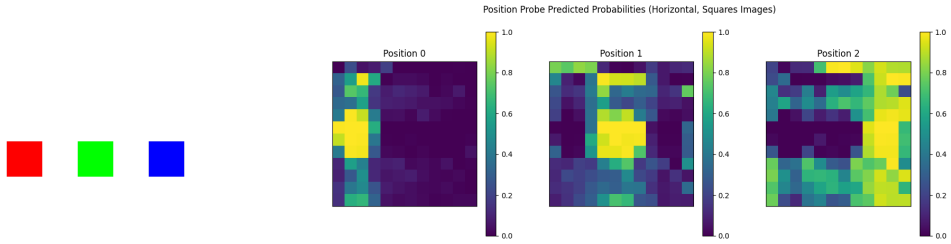


Figure 45: Horizontal position probe for square images, Qwen

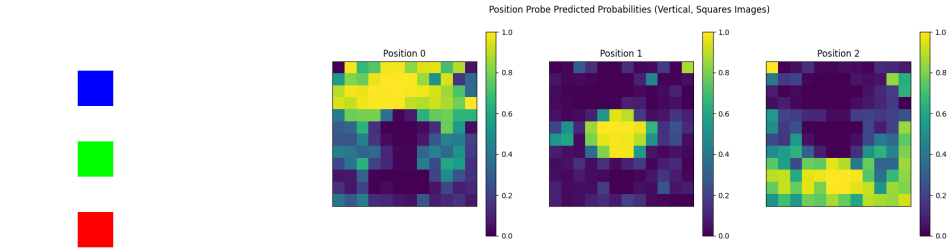


Figure 46: Vertical position probe for square images, Qwen

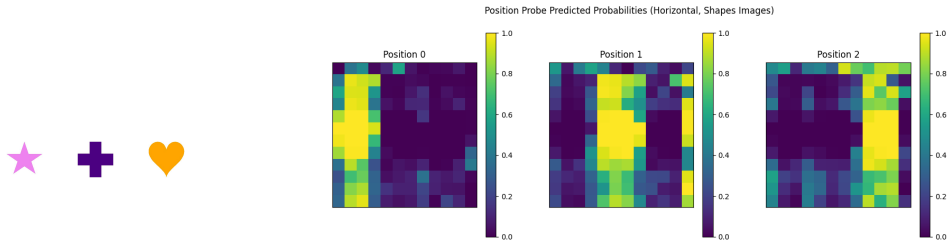


Figure 47: Horizontal position probe for shape images, Qwen

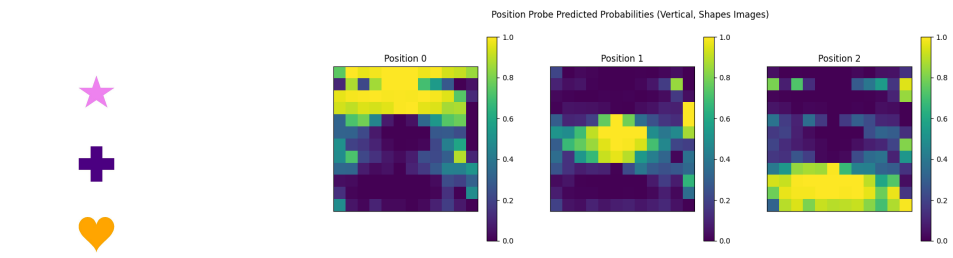


Figure 48: Vertical position probe for shape images, Qwen

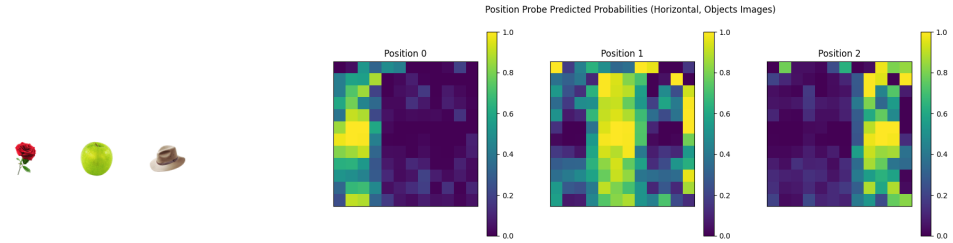


Figure 49: Horizontal position probe for object images, Qwen

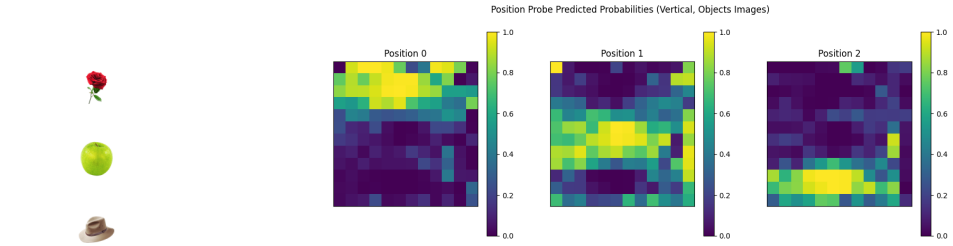


Figure 50: Vertical position probe for object images, Qwen

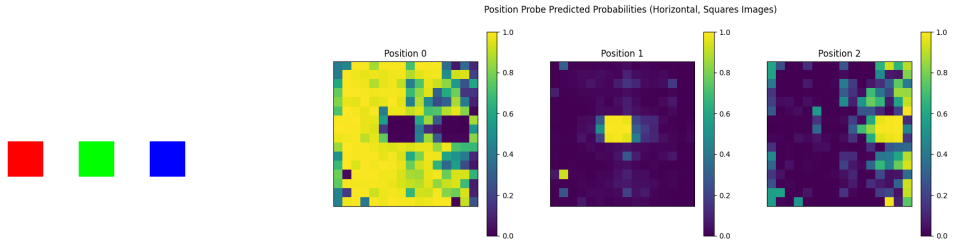


Figure 51: Horizontal position probe for square images, Gemma

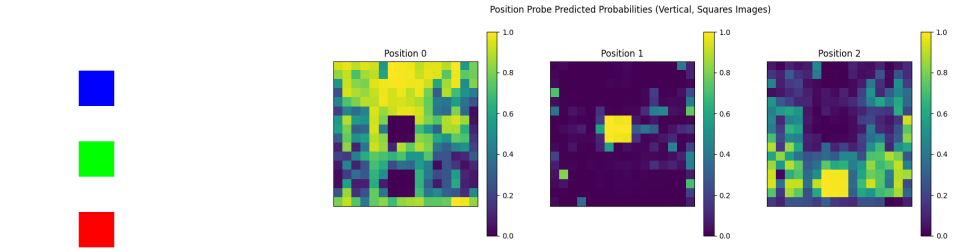


Figure 52: Vertical position probe for square images, Gemma

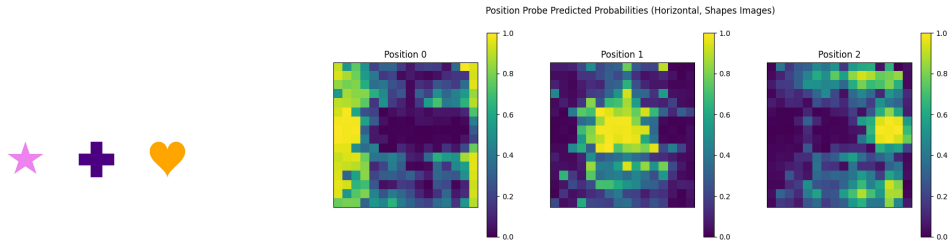


Figure 53: Horizontal position probe for shape images, Gemma

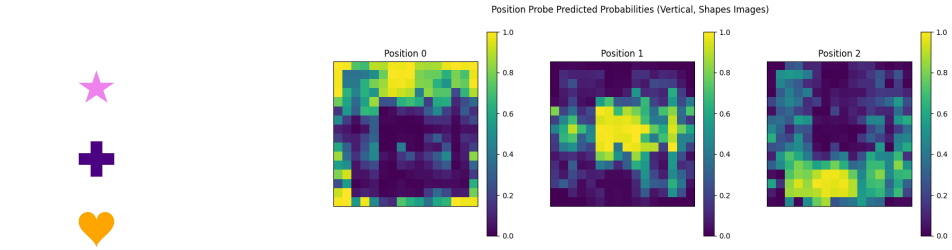


Figure 54: Vertical position probe for shape images, Gemma

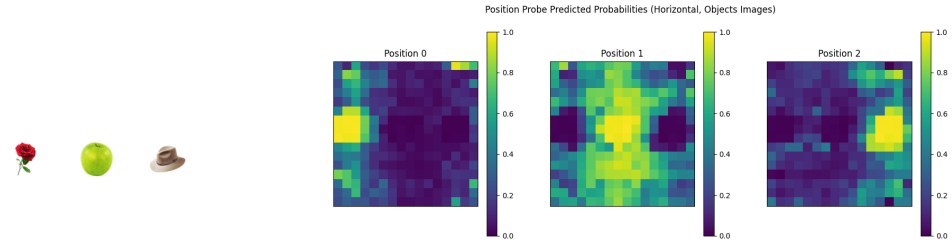


Figure 55: Horizontal position probe for object images, Gemma

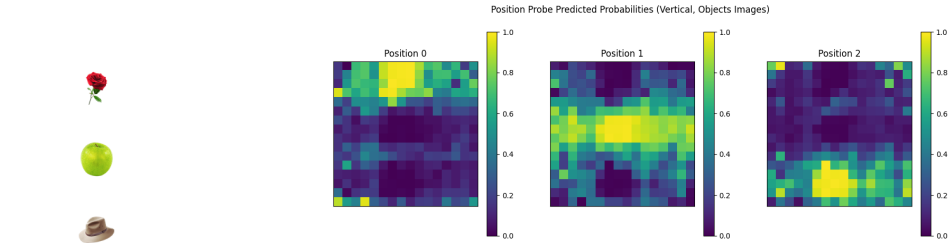


Figure 56: Vertical position probe for object images, Gemma

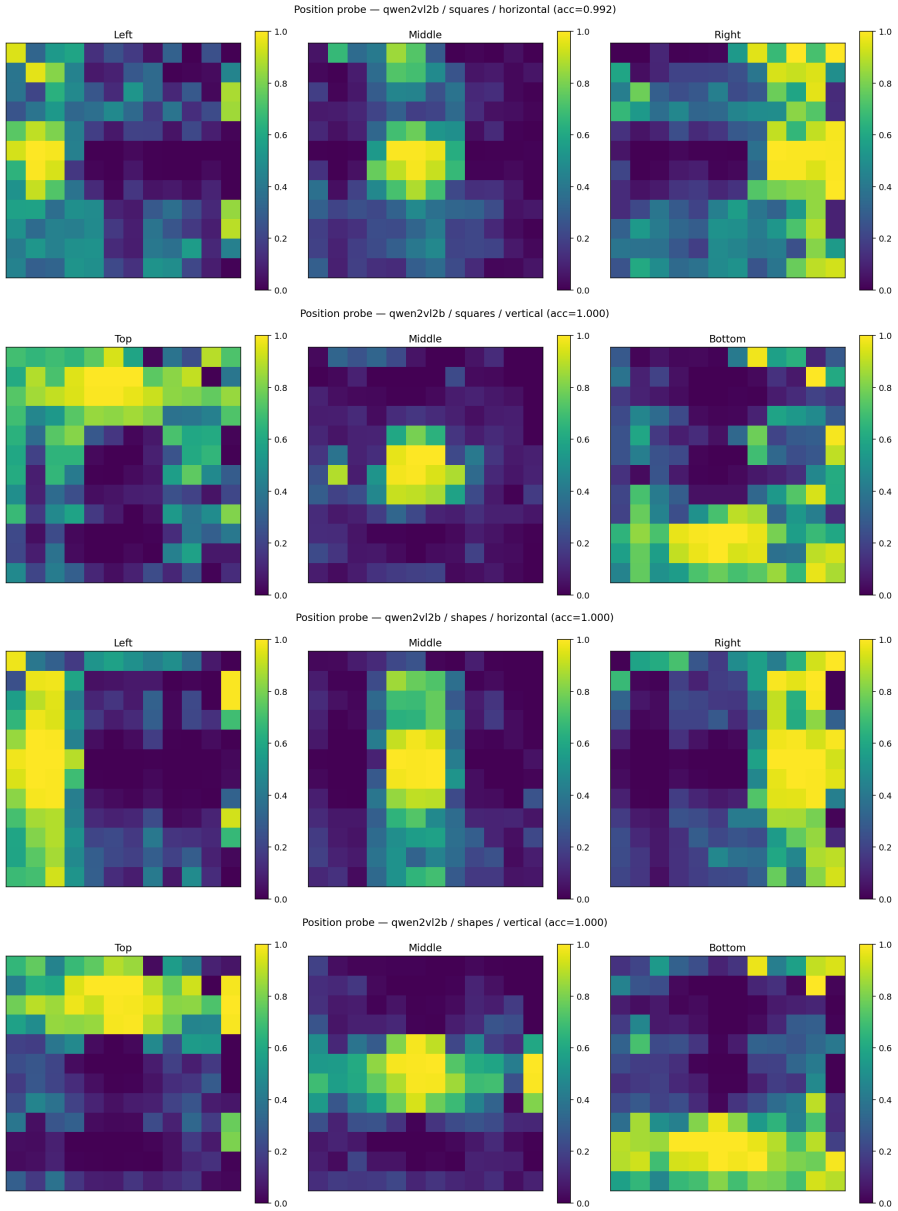


Figure 57: Position-probe predictions for `Qwen2-VL-2B-Instruct` on the Squares (horizontal, vertical) and Shapes (horizontal, vertical) settings.

C.9 Probing on grids

We extend the probing results from subsection C.8 to grid configurations consisting of both horizontal and vertical relationships between objects, with findings presented in Fig. 61 and Fig. 62. Probes are trained on 2×2 token objects for `Qwen2-VL-7B-Instruct` and 3×3 token objects for `Gemma3-4b-it`, with results exhibiting diffusion of positional information to surrounding tokens.

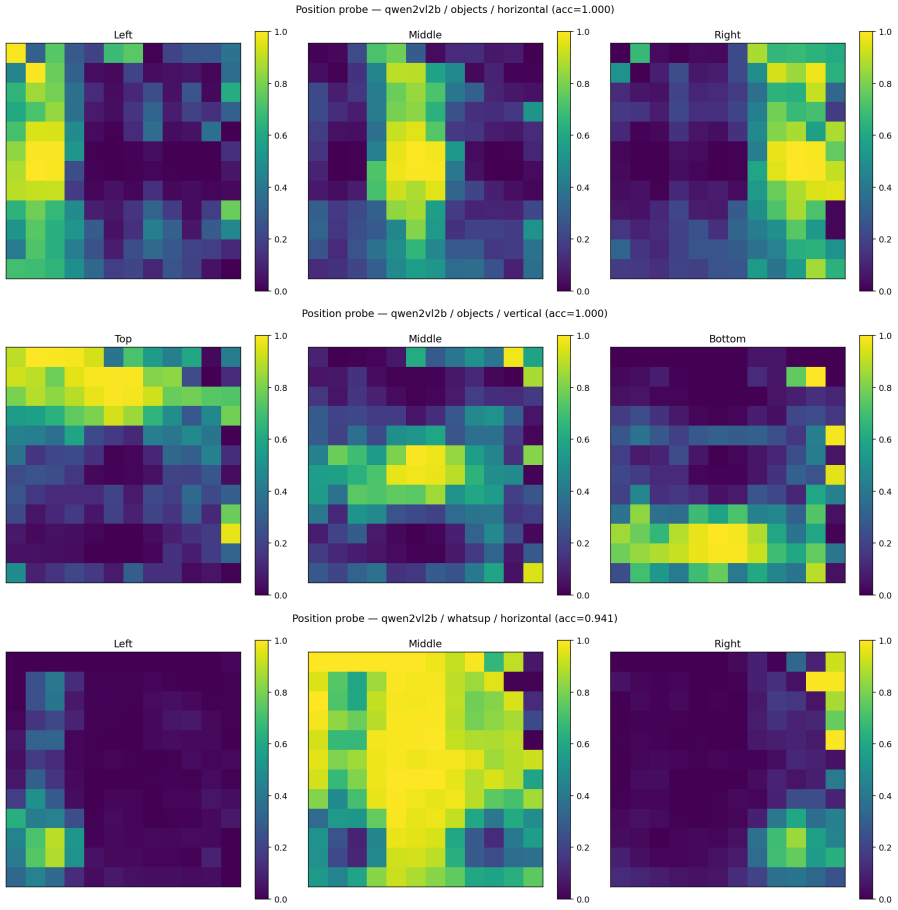


Figure 58: Position-probe predictions for `Qwen2-VL-2B-Instruct` on the Objects (horizontal, vertical) and What'sUp (horizontal) settings.

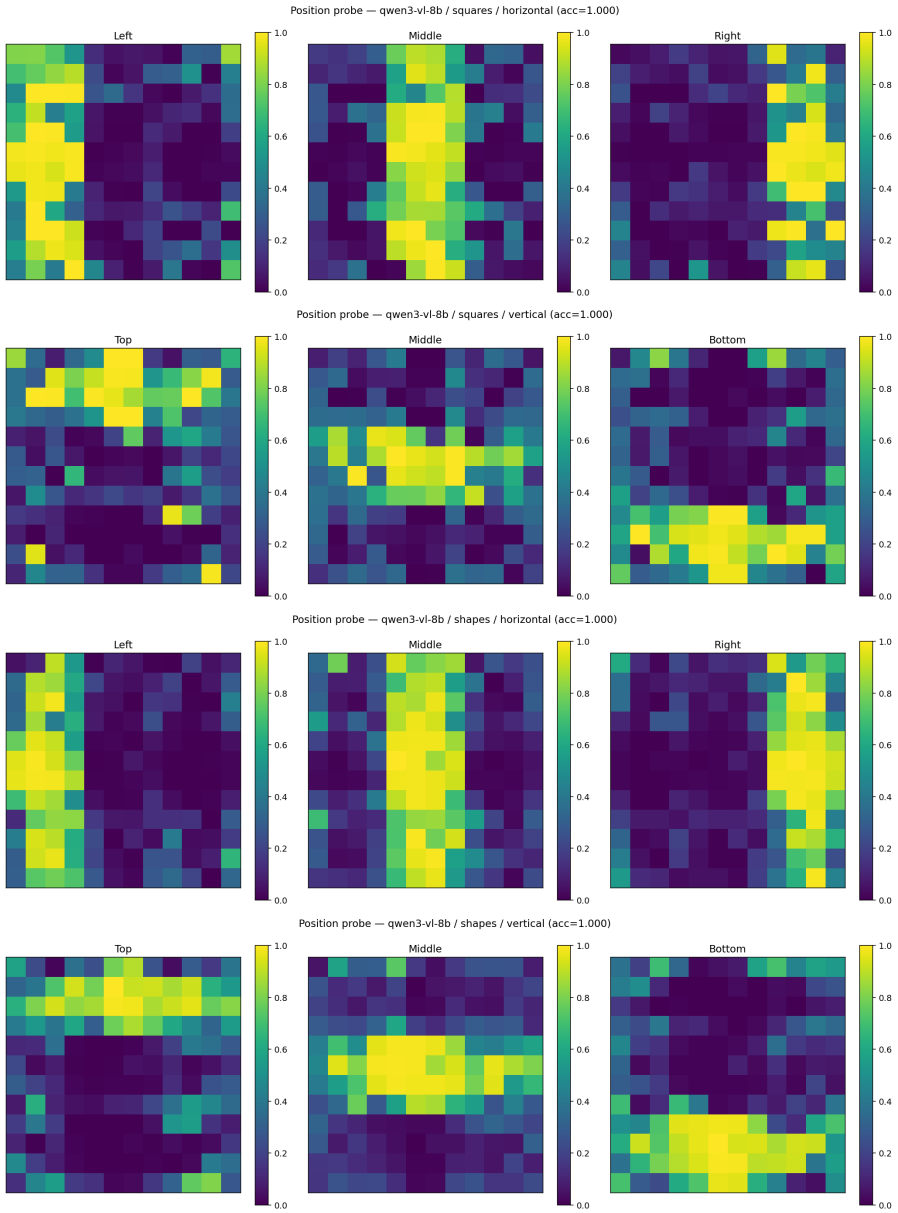


Figure 59: Position-probe predictions for `Qwen3-VL-8B-Instruct` on the Squares (horizontal, vertical) and Shapes (horizontal, vertical) settings.

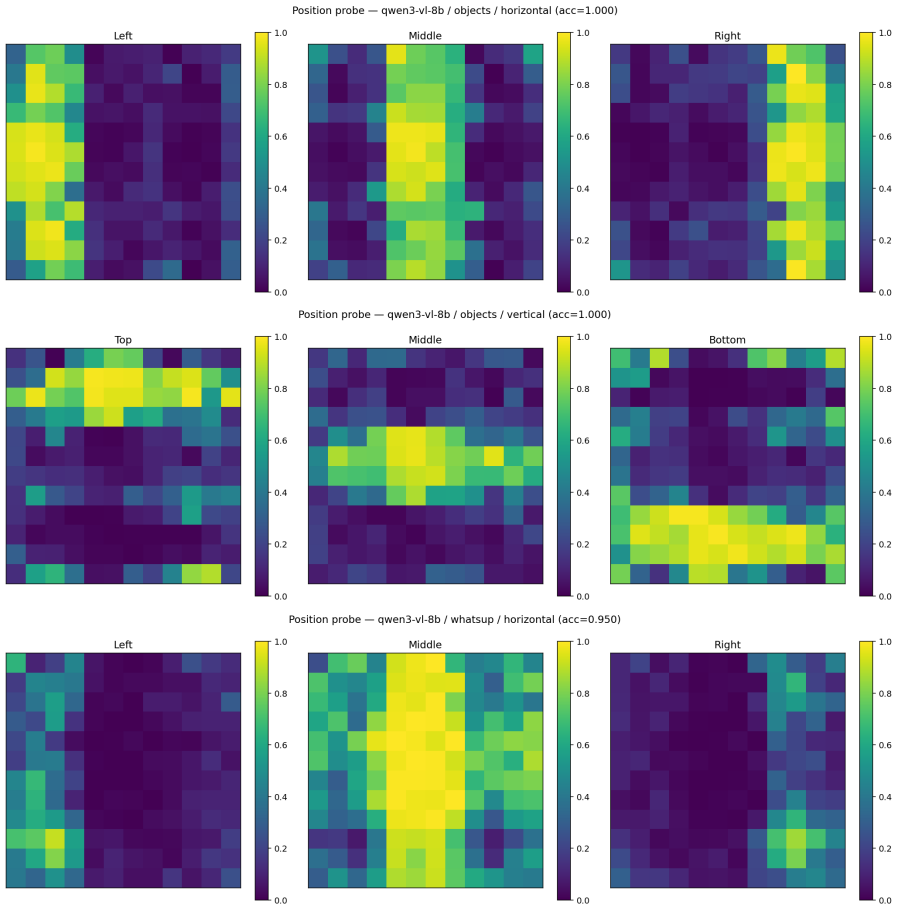


Figure 60: Position-probe predictions for `Qwen3-VL-8B-Instruct` on the Objects (horizontal, vertical) and What'sUp (horizontal) settings.

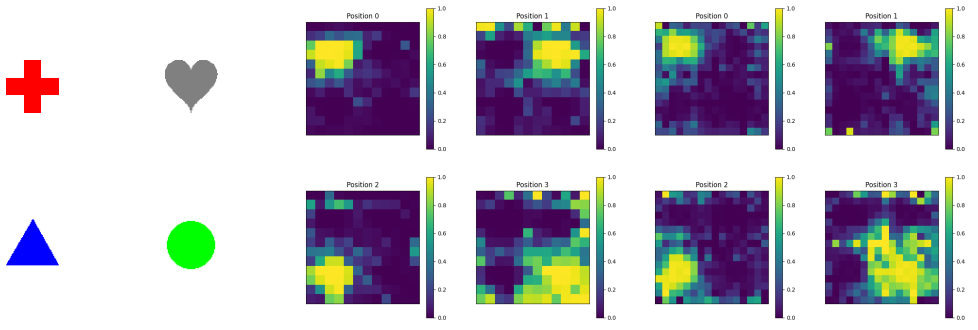


Figure 61: Qwen and Gemma probing results on 2x2 shapes grid.

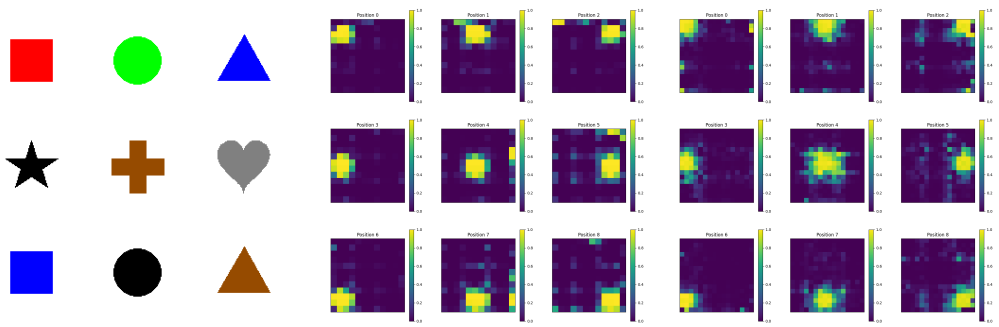


Figure 62: Qwen and Gemma probing results on 3x3 shapes grid.

C.10 PROBING FOR WHAT'SUP

Finally, we also show that the ordering information is diffused across background tokens in real-world images from What'sUp. Training data is extracted from the embeddings within the bounding boxes outlined in Fig. 63. Results are consistent with previous findings, as illustrated in Fig. 65.

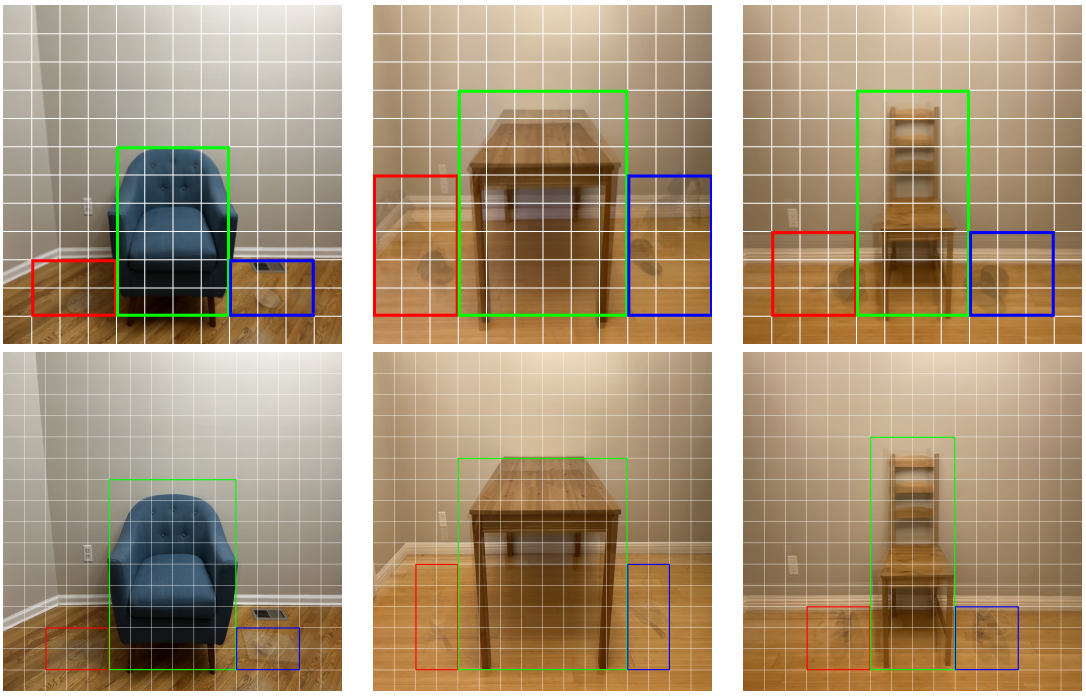


Figure 63: Preprocessing for What'sUp images: constructing bounding boxes around each object.



Figure 64: Qwen What'sUp probe

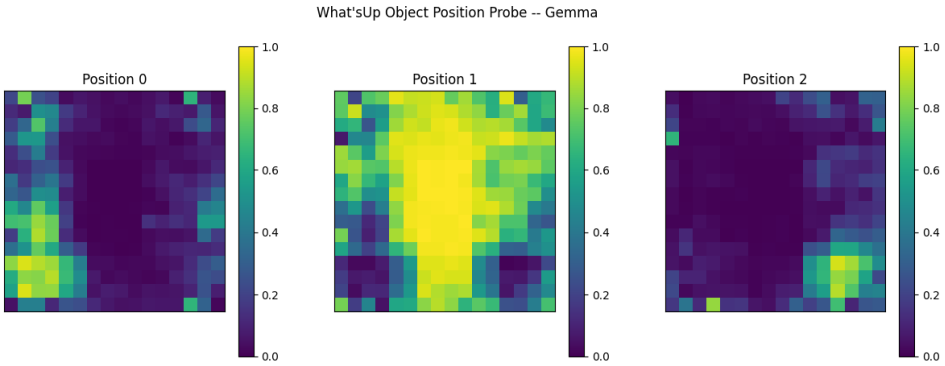


Figure 65: Gemma What'sUp probe

C.11 EXPERIMENTAL RESULTS WITH REALISTIC BACKGROUND

We also conduct probing experiments (Sec. 5.2.1) and intervention experiments (Sec. 5.2.2) using real background images to demonstrate that ordering information remains distributed across background tokens in real-world settings. We tested 15 different realistic backgrounds, and found a consistent behavior across all.

Figures 66–77 present the probing results, indicating that ordering information is spread across background tokens even when using real background images, rather than a white background. Additionally, the results in figures 78-81 confirm that patching the strip is required to alter the ordering information generated by the vision encoder.

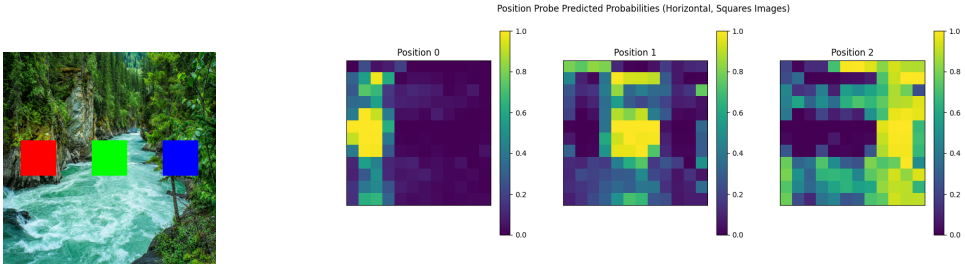


Figure 66: Horizontal position probe for square images with real background, Qwen

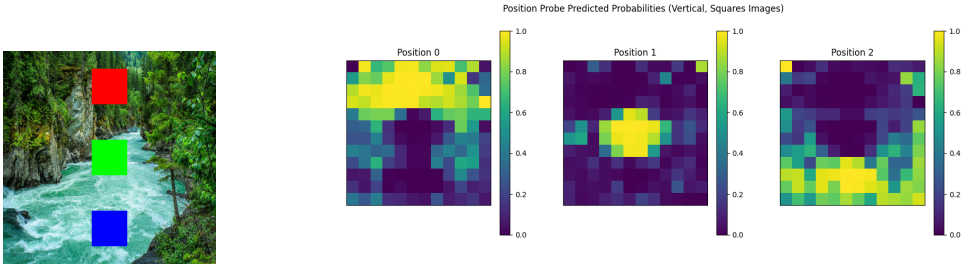


Figure 67: Vertical position probe for square images with real background, Qwen

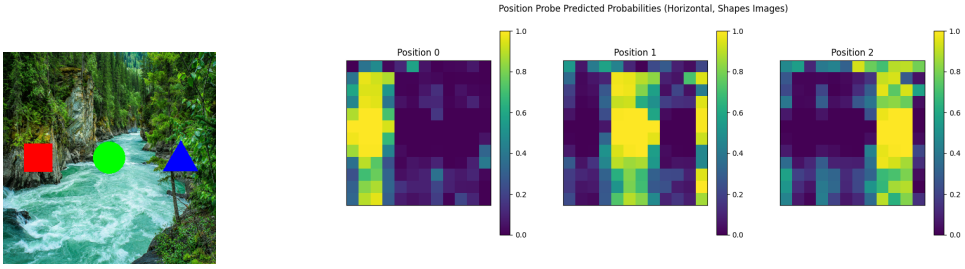


Figure 68: Horizontal position probe for shapes images with real background, Qwen

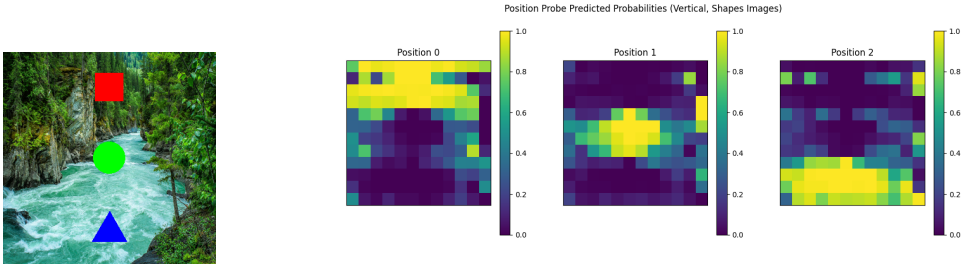


Figure 69: Vertical position probe for shapes images with real background, Qwen

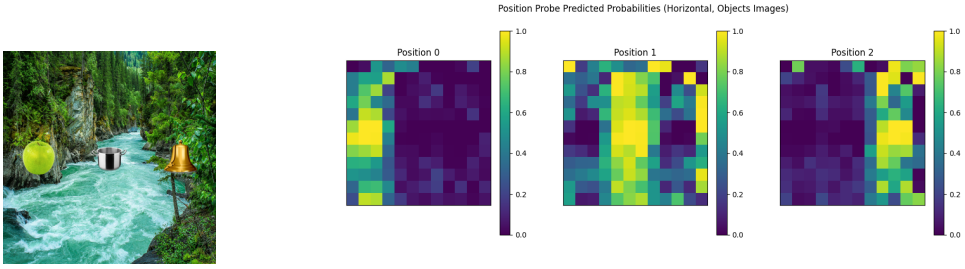


Figure 70: Horizontal position probe for objects images with real background, Qwen

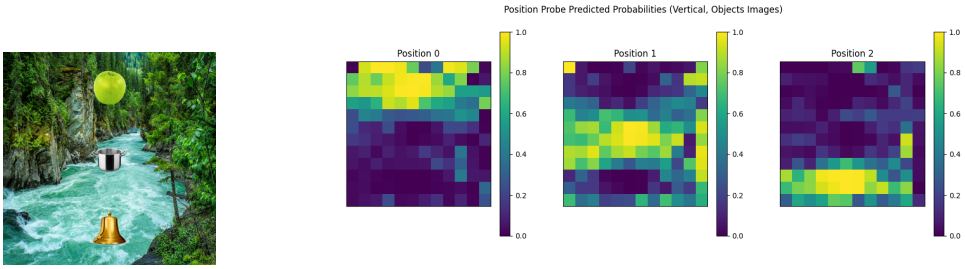


Figure 71: Vertical position probe for objects images with real background, Qwen

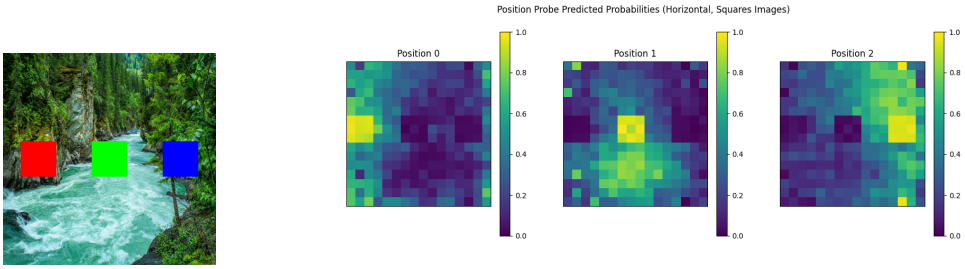


Figure 72: Horizontal position probe for square images with real background, Gemma

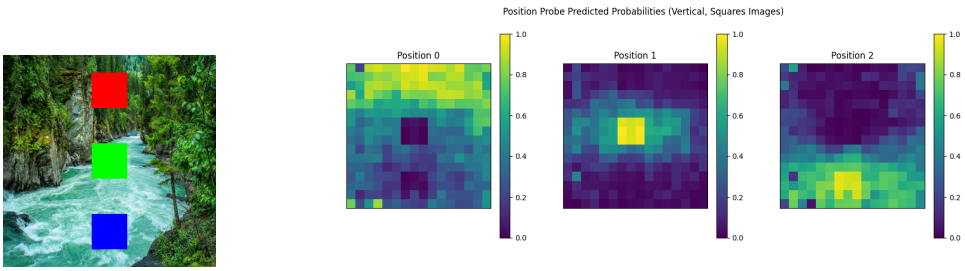


Figure 73: Vertical position probe for square images with real background, Gemma

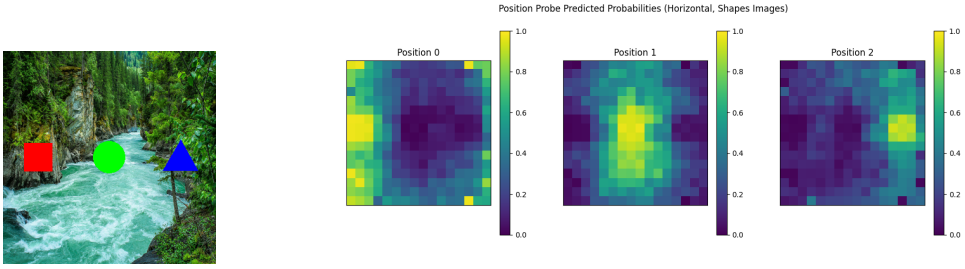


Figure 74: Horizontal position probe for shapes images with real background, Gemma

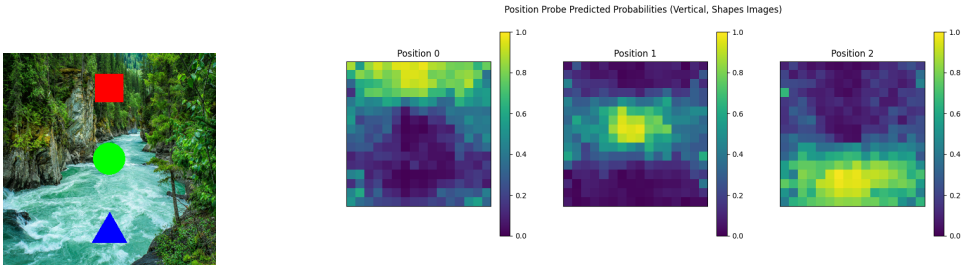


Figure 75: Vertical position probe for shapes images with real background, Gemma

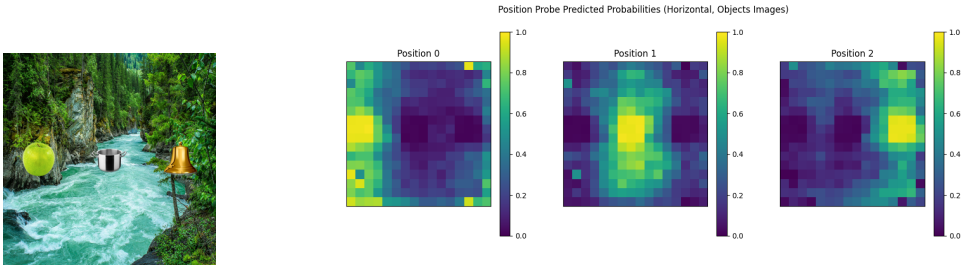


Figure 76: Horizontal position probe for objects images with real background, Gemma

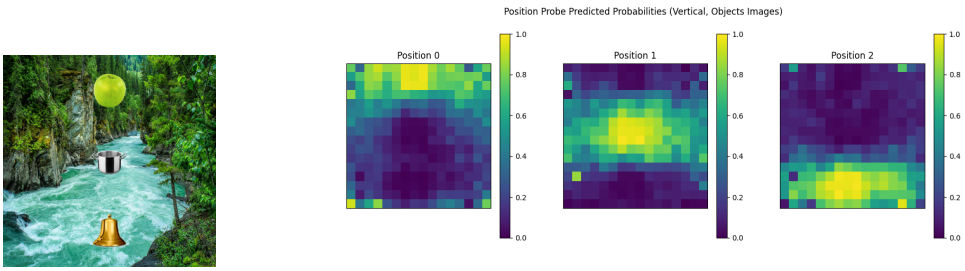


Figure 77: Vertical position probe for objects images with real background, Gemma

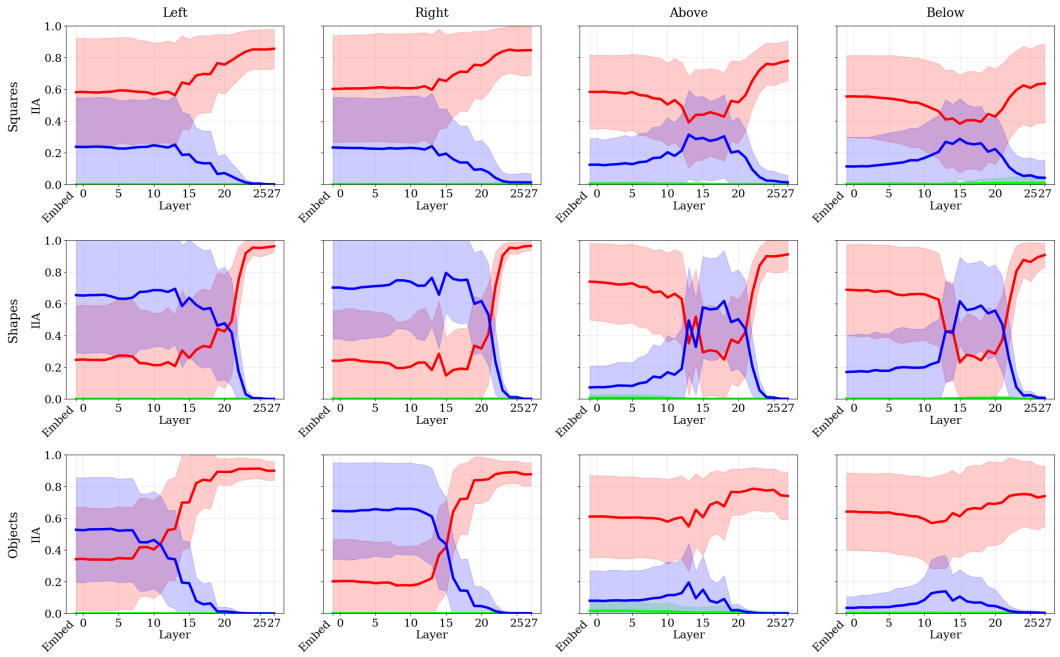


Figure 78: Interchange interventions applied only to visual tokens corresponding to the squares in `Qwen2-VL-7B-Instruct` model, with real background images.

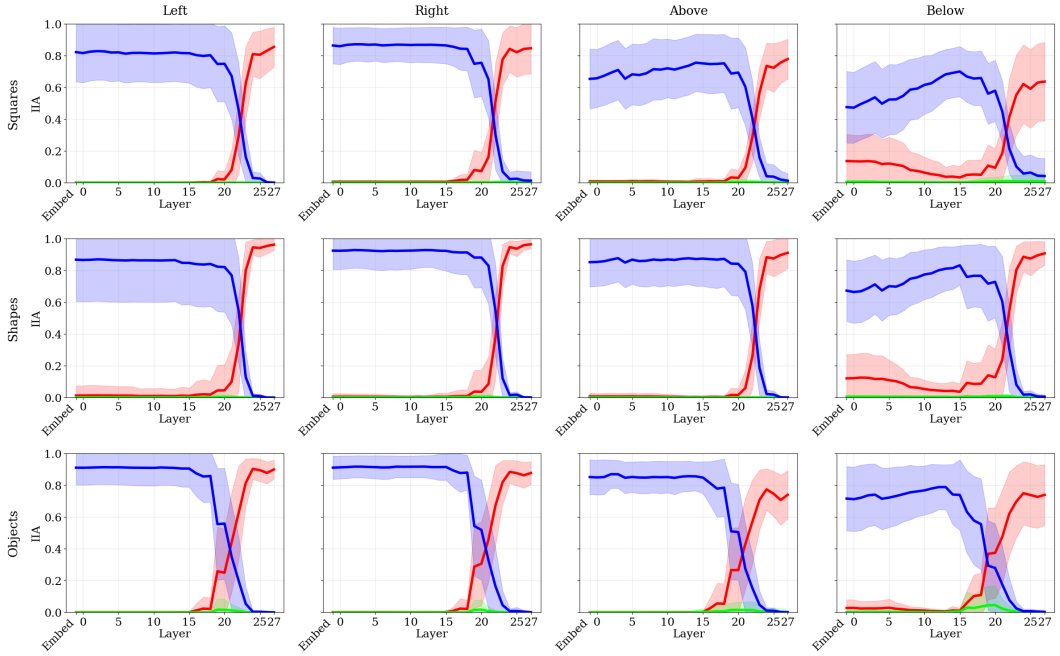


Figure 79: Interchange interventions applied only to visual tokens corresponding to the strip in `Qwen2-VL-7B-Instruct` model, with real background images.

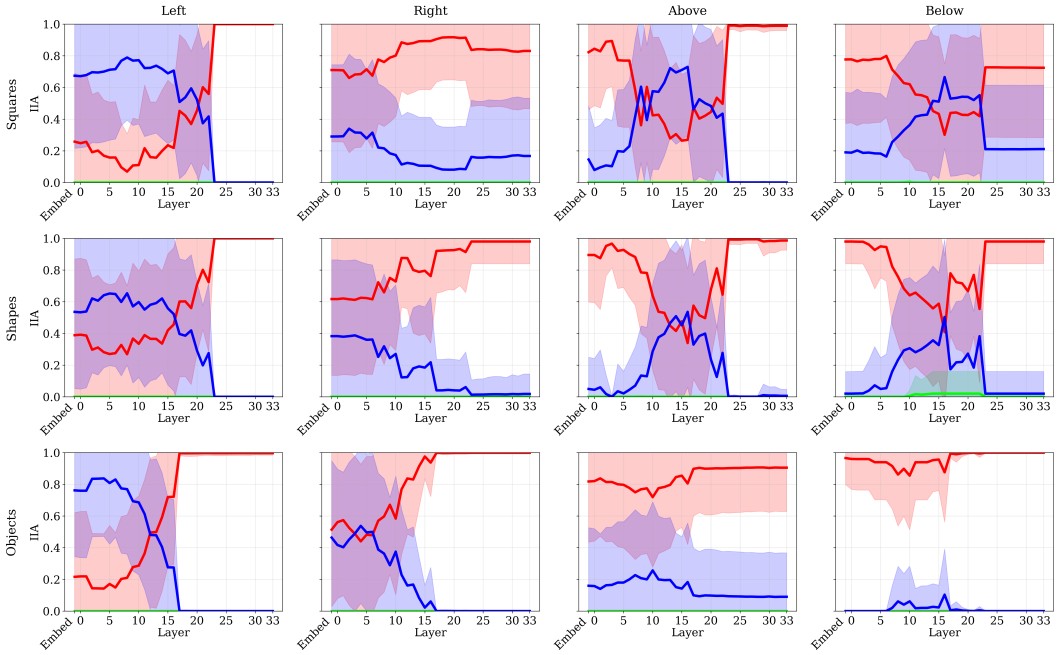


Figure 80: Interchange interventions applied only to visual tokens corresponding to the squares in `Gemma-3-4b-it` model, with real background images.

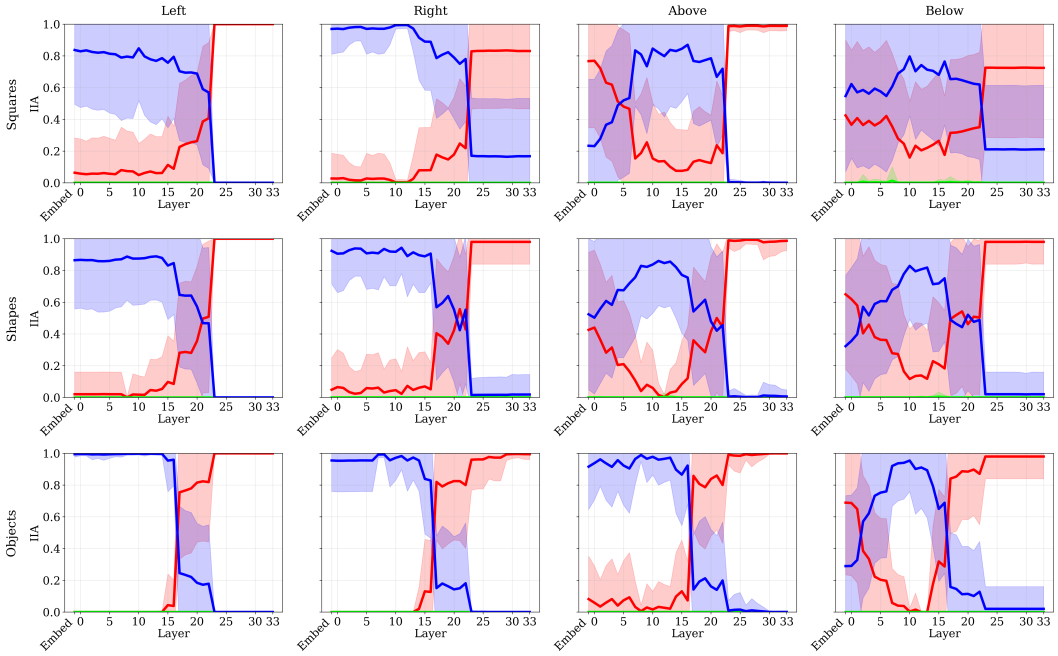


Figure 81: Interchange interventions applied only to visual tokens corresponding to the strip in `Gemma-3-4b-it` model, with real background images.