LLMs Can Compensate for Deficiencies in Visual Representations

Sho Takishita[1,2*†] **Jay Gala**[2*] **Abdelrahman Mohamed**[2]
Kentaro Inui[2,3,4] **Yova Kementchedjhieva**[2]

[1]Fujitsu Limited [2]MBZUAI [3]Tohoku University [4]RIKEN

sho.takishita@jp.fujitsu.com
{jay.gala, abdelrahman.mohamed, yova.kementchedjhieva}@mbzuai.ac.ae
kentaro.inui@tohoku.ac.jp

Abstract

Many vision-language models (VLMs) that prove very effective at a range of multimodal tasks build on CLIP-based vision encoders, which are known to have various limitations. We investigate the hypothesis that the strong language backbone in VLMs compensates for possibly weak visual features by contextualizing or enriching them. Using three CLIP-based VLMs, we perform controlled self-attention ablations on a carefully designed probing task. Our findings show that despite known limitations, CLIP visual representations offer ready-to-read semantic information to the language decoder. However, in scenarios of reduced contextualization in the visual representations, the language decoder can largely compensate for the deficiency and recover performance. This suggests a dynamic division of labor in VLMs and motivates future architectures that offload more visual processing to the language decoder.

1 Introduction

Vision-language models (VLMs) have made remarkable progress in recent years, with systems like MAGMA (Eichenberg et al., 2021), BLIP-2 (Li et al., 2023a), LLaVA (Liu et al., 2023, 2024a), and Prismatic (Karamcheti et al., 2024) demonstrating strong performance for their time on key multimodal benchmarks. A common design choice across these models is the use of a frozen pretrained vision encoder, often based on CLIP (Radford et al., 2021), paired with a pre-trained language model, which is finetuned to map visual features into text.[1]

Despite the widespread use of CLIP, its limitations as a visual backbone are well-documented. CLIP representations have been shown to prioritize global over local features (Wang et al., 2025), to perform poorly in distinguishing objects which share high-level features (Shao et al., 2023), to lack fine-grained compositionality (Lewis et al., 2024), and to exhibit quantity and size biases (Zhang et al., 2024b; Abbasi et al., 2025). Many of these limitations have been attributed to the contrastive training objective that CLIP employs.

Nonetheless, many modern VLMs that rely on the CLIP vision encoder perform surprisingly well, even on tasks requiring detailed visual understanding (Fu et al., 2023; Yue et al., 2024; Onoe et al., 2024). This raises a fundamental question: How do VLMs overcome the known limitations of CLIP's representations? One plausible hypothesis is that the language decoder–often much larger than the vision encoder and trained with rich linguistic supervision–plays a compensatory role, enriching or contextualizing the visual representations it receives. If true, this would suggest a more dynamic division of labor between vision and language components than currently believed.

In this work, we investigate this hypothesis through a series of controlled self-attention blocking experiments (Geva et al., 2023) on three VLMs, each of which pairs CLIP with a distinct language model. We ask whether the language decoder in a VLM contributes to the enrichment of image features. As a diagnostic task, we focus on the identification of localized object *parts* (e.g., the *ear* of a cat or the *stem* of an apple). This fine-grained task requires extensive contextualization of the part region into the broader context of the object.

We build a probe using segmentation annotations from the Panoptic Parts dataset (de Geus et al., 2021; Meletis et al., 2020), and apply Logit Lens analysis (Nostalgebraist, 2020) to inspect the intermediate representations across layers of the VLM. We iteratively block self-attention in the vision encoder, language decoder, or both, and evaluate the

[*]Equal contribution
[†]Research conducted during a research stay at MBZUAI.
[1]The vision encoder and language decoder are commonly connected with a linear projection or a multi-layer perceptron, which has largely been shown to serve the technical role of mapping between dimensionalities rather than semantic spaces (Schwettmann et al., 2023a; Verma et al., 2024).

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 15253–15272
November 4-9, 2025

VLM's ability to maintain the identifiability of object parts in the relevant regions.

Our results yield three key findings: **(1)** When the decoder self-attention is disabled, part identifiability is largely preserved, suggesting that the self-attention in the decoder does not substantially enrich already-good visual features. **(2)** When both encoder and decoder self-attentions are blocked, part identifiability collapses, showing the importance of some form of context-based feature construction. **(3)** Crucially, when only the encoder self-attention is blocked, part identifiability largely recovers–indicating that the decoder can compensate for deficiencies in visual representations when the need arises through contextualization.

These findings challenge the assumption that visual semantic understanding is fully localized in the vision encoder of VLMs. Instead, they suggest an adaptive joint processing of image inputs by VLM components, where the language decoder can step in to compensate for a degraded input from the vision encoder. This has implications for future VLM architectures, which could actively offload more of the vision processing onto the language decoder, reducing the contribution of the vision encoder to only that which the decoder cannot recover.

2 Related Work

Efforts in interpreting language models have led to the development of techniques that probe internal representations and decompose the mechanism behind next-token prediction. Methods such as neuron attribution (Dai et al., 2022), causal tracing (Meng et al., 2022), attention knockout (Geva et al., 2023), and logit lens (Nostalgebraist, 2020) have improved our understanding of how information flows across tokens and model layers. These have also been extended to VLMs to understand how visual and linguistic information interact.

Basu et al. (2024) applied causal tracing to VLMs and found that LLaVA (Liu et al., 2023) retrieves information from earlier layers in the language decoder compared to unimodal language models. Building on this, Jiang et al. (2024) used attention knockout and logit lens analysis to further decompose information flow in VLMs, revealing a two-stage process: visual enrichment, where image features transfer to object tokens, and semantic refinement, where these features are interpreted through language. Similarly, Zhang et al. (2024c) demonstrated that early layers transfer global visual information into question token representations, mid-layers inject question-relevant visual features into corresponding text positions, and later layers propagate this fused representation to the last token position for answer prediction. Contrary to such grounding evidence, Liu et al. (2024b) highlighted the dominant role of language priors in VLM's outputs, while Stan et al. (2024) found that in different settings LLaVA either over-relies on text input or, conversely, exhibits strong visual grounding.

While the studies above explore how VLMs *generate* text from either visual input or a mix of visual and text input, there have been very few works that attempt to dissect the language model representation of the visual tokens as such. Schwettmann et al. (2023b) identified multimodal neurons in the language decoder that map visual inputs to corresponding linguistic concepts, highlighting that linear projections in VLMs lack semantics on their own when interpreted using language vocabulary. Most related to our focus, Neo et al. (2025) used visual token ablation and logit lens analysis to reveal that LLaVA localizes object-specific information at the corresponding positional tokens and can align these with fine-grained semantic concepts beyond surface-level object categories.

Zooming in on the vision encoder itself, Gandelsman et al. (2024, 2025) decomposed CLIP's visual embeddings into sparse text concepts and found that different attention heads specialize in distinct semantic properties (e.g., color, shape, location, etc). Despite CLIP's strong zero-shot classification performance, various studies have highlighted its limitations, ranging from poor compositional reasoning and limited sensitivity to fine visual distinctions (Wang et al., 2025; Shao et al., 2023; Lewis et al., 2024; Zhang et al., 2024b), to biases related to object quantity and size (Abbasi et al., 2025).

Meanwhile, Li et al. (2024) showed that some perceived limitations in the CLIP visual representations in fact stem from the global pooling of features in similarity-based analysis and from its weak text encoder. In a similar vein, Lin et al. (2024) noted the inability of image-text metrics like CLIPScore (Hessel et al., 2021) to distinguish differences in object relations, attributes, or logical structure, often treating captions as a "bag of words."

Together, these studies tell a nuanced story of the mechanisms behind building visual representations, with their perceived and real limitations, and translating those into text. Our work investigates how visual representations evolve through the lay-

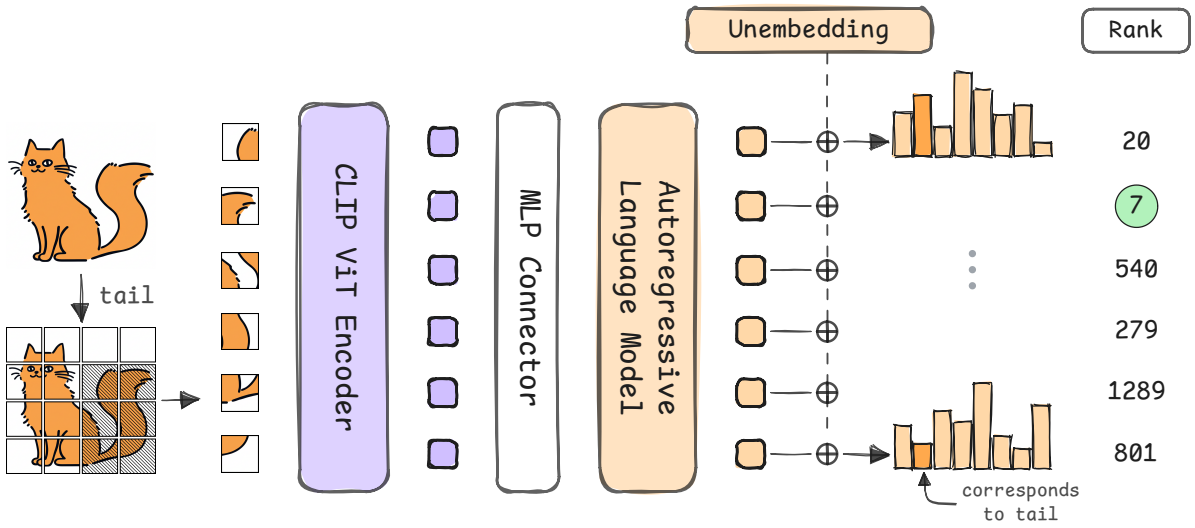

Figure 1: Overview of Object Part Identification. Given an object part (e.g., the tail of a cat), localized to a region of image patches through a segmentation mask, a representation is obtained from the VLM for all relevant patches, a probability distribution is induced through LogitLens, and an identifiability rank is extracted from each distribution for the relevant label (tail). The highest rank across patches indicates the overall part identifiability.

ers of the language decoder and to what extent their deficiencies are resolved in the process.

3 Object Part Identification (OPI)

To investigate how vision-language models contextualize visual inputs, we propose a fine-grained probing task that requires extensive contextualization: identifying object parts in patch-level localized image regions. The complexity of this task arises from the fact that parts can often be overlooked in favor of the whole, while also being potentially unidentifiable in isolation of the object they belong to. The details of the probing task are described below and are visualized in Figure 1.

Task. OPI assumes the availability of patch-level masks which localize object parts in an input image. A single patch can contain multiple parts and a single part can span multiple patches. Probing the VLM consists of: (1) inducing a probability distribution from a VLM for every image patch, (2) extracting the rank of all part labels relevant to a patch, and (3) aggregating the ranks across all patches in the part-relevant. This results in one value per part which indicates how well the model identifies this part in this image.

Dataset. We derive the necessary annotations from the Pascal Panoptic Parts (Pascal-PP) dataset (de Geus et al., 2021; Meletis et al., 2020) which provides high-quality pixel-level segmentation annotations for 194 part classes spanning 16 object classes, e.g., dog-head, horse-tail, person-leg, etc. We focus on 7 out of the 16 object classes available in Pascal-PP, specifically animal objects. This and other measures taken to reduce noise are outlined in Appendix A. After filtering, we sample up to 100 images per class, where available.

Finally, we convert pixel-level segmentation masks into patch-level masks, effectively reducing the resolution of the masks. The final dataset contains 567 unique images and 2840 unique part regions. (see Table 1 for more details.)

Probing the VLM. Given a VLM,[2] we feed an input image all the way through and use Logit Lens (Nostalgebraist, 2020) to analyze the semantics of the patch representations as they pass through the decoder. Specifically, the hidden representation of patch i, $h_i \in \mathbb{R}^{d_{decoder}}$, is projected into the output vocabulary space as follows:

$$p_i = \text{softmax}(U \cdot \text{LayerNorm}(h_i)) \quad (1)$$

where $U \in \mathbb{R}^{|V| \times d}$ is the unembedding matrix that projects the d-dimensional hidden representation to a $|V|$-dimensional logit space corresponding to the vocabulary, V, of the language decoder.

Rank. Prior to observing the rank of a label, the probability distribution is processed to aggregate

[2]One that follows the standard patch-level decoder prefixing method of cross-modal fusion (Liu et al., 2023).

all label aliases, e.g., "leg", "legs", "_leg", "_legs", under a single token, *leg*, summing up all their probabilities.[3] The rank of label l is obtained from the probability distribution p_i as follows:

$$r_i = 1 - \frac{\log\left(\text{argwhere}(\text{argsort}(p_i) = l)\right)}{\log\left(|V|\right)} \quad (2)$$

where argsort sorts the logits in descending order, argwhere returns the index of the label and $|V|$ represents the vocabulary size. Considering the large size of the vocabulary, we apply a logarithmic transformation to emphasize the differences among high ranks. This rank is computed only for the parts relevant to the patch in question, as knowledge of the localization of object parts is a given.

Aggregation Across Regions. Lastly, for every part region, which optionally spans multiple patches, we perform max pooling to obtain a final identifiability score (Vilas et al., 2023). The choice of max pooling over mean pooling is motivated by Neo et al.'s (2025) finding that most of the information about an object is concentrated in only a few of the object's patches; we too observed the same. Max pooling ensures that the strongest identifiability signal is reflected in the final score.

The identifiability scores obtained in this way for the parts in a single image are then averaged across all images in the dataset and enable comparisons across the identifiability of different parts in the same experimental setting and of the same parts across different experimental settings.

4 Probing Contextualization in VLMs

4.1 Model Architecture and Details

Our study focuses on the widely adopted LLaVA architecture (Liu et al., 2023, 2024a). LLaVA is a VLM which integrates an image encoder with a language model through a trained connector module in a two-stage fine-tuning recipe, with image-text caption pairs and multimodal conversations. The image encoder processes image patches and outputs patch-level visual representations that are projected into the subspace of the language model via the connector module. These projected representations together with a representation of any prompt text constitute are contextualized through causal self-attention and used to condition the autoregressive generation of new text.

[3]The label aliases are manually defined with reference to the VLM vocabulary.

Object	Parts	N	P
Bird	Beak, Foot, Head, Leg, Neck, Tail, Torso, Wing	93	394
Cat	Ear, Eye, Head, Leg, Neck, Nose, Paw, Tail, Torso	100	585
Cow	Ear, Head, Horn, Leg, Muzzle, Neck, Tail, Torso	56	193
Dog	Ear, Eye, Head, Leg, Muzzle, Neck, Nose, Paw, Tail, Torso	100	518
Horse	Ear, Head, Hoof, Leg, Muzzle, Neck, Tail, Torso	81	311
Person	Arm, Ear, Eye, Foot, Hair, Hand, Head, Leg, Mouth, Neck, Nose, Torso	92	701
Sheep	Ear, Head, Horn, Leg, Muzzle, Neck, Tail, Torso	45	138
Total		567	2840

Table 1: Summary of selected object classes with available parts. N: total number of images per class and P: total number of unique part regions per class.

We carry out experiments with LLaVA-1.5 7B and 13B[4] , BakLLaVA 7B[5] and TinyLLaVA[6] (Zhou et al., 2024), all three of which use the CLIP ViT-L/14 image encoder with 24 layers (Radford et al., 2021). As decoder, LLaVA-1.5 7B/13B uses Vicuna 7B/13B (Chiang et al., 2023), BakLLaVA uses Mistral 7B v0.1 (Jiang et al., 2023), and TinyLLaVA uses TinyLLaMA (1.1B) (Zhang et al., 2024a). All language decoders are prompted using their standard template ("USER: <image>").

This selection of models with different language backbones allows us to robustly study the role of visual input contextualization in LLMs in general. Indeed, our results show consistent trends across all VLMs, with reduced magnitude for the considerably smaller TinyLLaVA model, whose results we show Table 3 in the Appendix.

4.2 Establishing a Baseline

The primary goal of this study is to understand the role and capabilities of language decoders in VLMs through the lens of how they solve the custom diagnostic task of object part identification (OPI). A key prerequisite to any such analysis is to establish the baseline performance of the VLMs on the task.

Using the method introduced in §3, we obtain

[4]https://huggingface.co/collections/llava-hf/llava-15-65f762d5b6941db5c2ba07e0

[5]https://huggingface.co/llava-hf/bakLlava-v1-hf

[6]https://huggingface.co/bczhou/tiny-llava-v1-hf

Figure 2: OPI scores for BakLLaVA 7B, LLaVA 7B, and LLaVA 13B models. Left: Per-object part identifiability with aggregated part and object-level scores at the end. Right: Layer-wise evolution of per-object part identifiability with normalized across layers as percentages for consistency.

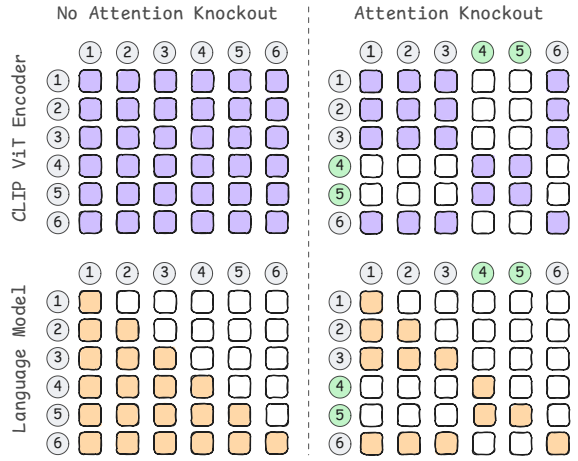

Figure 3: Attention maps from the vision encoder and language decoder with and without attention knockout. Green indicates target patches under knockout.

OPI scores from all models. The scores, averaged across all parts per object (*bird*, *cat*, etc.) and across all objects (*all*), can be seen in Figure 2 (left). Average object identifiability scores are also included for reference (hatched bars), as induced from part regions. We observe that the OPI rate is significantly above chance for all three models, at or above 80% on average, and matches the object identifiability. This shows that the VLMs effectively map between visual concepts and their corresponding vocabulary tokens even on the low level of object parts. While it is expected that object information is often mentioned in text, object parts are mentioned far more rarely.

In Figure 2 (right), we see how identifiability progresses across the layers of the models, as a function of percentage of layers computed. OPI in LLaVA 13B progresses in two steps, with one surge at about 20% and another at 40% after which it nearly plateaus. LLaVA 7B shows a steep ascent up to about 60% of its layers, and BakLLaVA's progression is stable throughout. Interestingly, all models show a drop in identifiability in the last 2-3 layers (with a subsequent recovery in the last layer for LLaVA 7B only). This LLM behavior might be linked to confidence calibration mechanisms, which aim to prioritize a specific set of output tokens, subject to strong language priors in addition to visual context.

Considering the similarity in trends across the three models, we focus subsequent analysis on one model, LLaVA (13B) for simplicity, presenting partial results for the rest where relevant, with all remaining results included in Appendix B.

4.3 Ablating Contextualization

Having established that LLaVA is highly successful in identifying object parts, next we inspect the source of this performance by ablating the contextualization of image patches in the visual encoder and the language decoder through Attention Knockout (AK) (Geva et al., 2023). We block the flow of information between the part regions and the rest of the image, at different layers of LLaVA. This allows us to assess how contextualization plays into OPI across different layers of the encoder and decoder.

Specifically, we implement AK such that in the encoder, bidirectional self-attention is blocked between target-region patches and other patches, and in the language decoder, causal self-attention is blocked from target-region to past patches. This is illustrated in Figure 3. In both modules, target-region patches can attend to each other, and non-target patches can also attend to each other.

Figure 12 presents the layer-wise identifiability results for all experimental configurations, with a per-object breakdown for LLaVA 13B and averaged results for all three models. Below, each

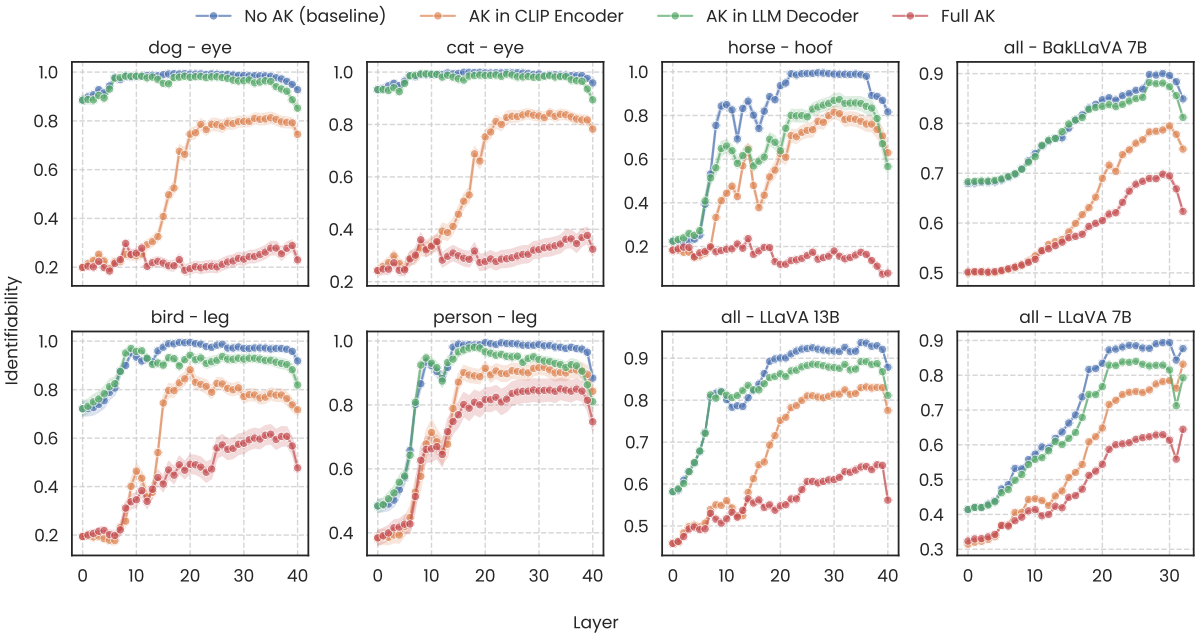

Figure 4: Layer-wise evolution of per-object part identifiability in the LLM Decoder under attention knockout (AK) settings. "No AK" denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while "Full AK" indicates modified self-attention in both. The first two columns and top row of the third column show qualitative results for 5 object-part cases for LLaVA 13B model. Remaining plots show aggregated identifiability across all parts for BakLLaVA 7B, LLaVA 7B and LLaVA 13B models.

configuration is discussed in turn.

Contextualization is Key to OPI. To assess the role of contextualization in the OPI task, we establish a floor of performance by applying attention knockout to both the vision encoder and the language decoder (Full AK). This leads to a sizable drop in OPI scores compared to the baseline (No AK), which limited recovery across the layers of the decoder. Despite this drop, the identifiability rate remains nonzero, indicating that some object parts can be identified to a high degree even in isolation from the object they belong to. In Figure 4, we show one such case: the leg of a person, whose identifiability drops a bit as a result of the full attention knockout, but stands apart from other parts in how little its identifiability changes. Overall, however, OPI proves to be a highly context-dependent task, as intended.

Trends are similar for BakLLaVA 7B and LLaVA 7B (last column in Figure 4), with the gap between the upper (No AK) bound and lower (Full AK) bound being smaller in comparison to LLaVA 13B, but still considerable.

LLM Contextualization Only Marginally Modifies Good Patch Representations. Knocking out self-attention in the decoder results in only slightly lower OPI scores compared to the No AK baseline. This suggests that the self-attention in the decoder plays a small role in the identifiability of object parts. The vast difference in impact between Full AK and this setting indicates that most object part information is already encoded in the CLIP encoder via self-attention between target and non-target patches. Therefore, the LLM decoder functions mainly as a "reader," extracting information that is largely self-contained in the visual representations of the target region. This is also evidenced by the fact that identifiability rates are relatively high from layer 0 for most object parts, which suggests that the language decoder directly reads much of the semantic information.

Here again, trends for BakLLaVA and LLaVA 7B are similar, with BakLLaVA in particular exhibiting almost no drop in OPI when self-attention in the decoder is blocked. In light of these observations, the finding of the third and final AK configuration below comes as a surprise.

LLM Contextualization Can Greatly Enhance Poor Patch Representations. Based on the previous two AK configurations, we would expect that blocking self-attention in the vision encoder (AK in CLIP Encoder) should have an almost as

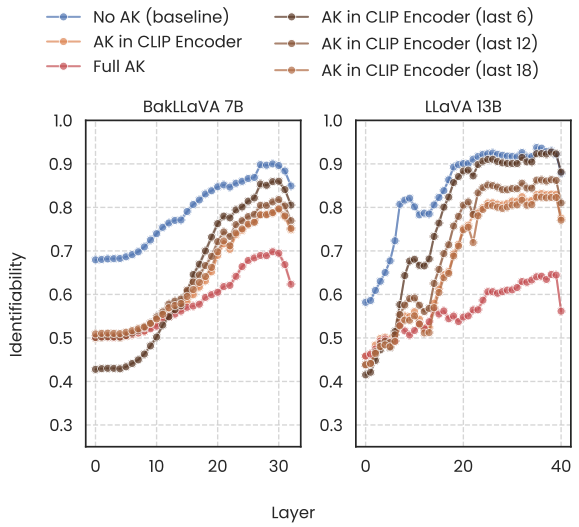

Figure 5: Layer-wise evolution of part identifiability in the LLM Decoder when progressively blocking self-attention across different CLIP Encoder layers of the VLM. "No AK" denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while "Full AK" indicates attention knockout in both.

detrimental effect as blocking all self-attention in the VLM (Full AK), since the role of the LLM appears negligent so far. Yet, we find that OPI scores for this setting greatly exceed floor performance, and actually land much closer to the ceiling performance of the No AK baseline. It appears that the language decoder, while passively reading from strong visual representations, can actively "write back" or reconstruct information when presented with weaker visual representations.

The trend holds similarly for BakLLaVA and LLaVA 7B, albeit with a less pronounced recovery gap for these VLMs with weaker language backbones. Indeed, the size of the LLM appears to factor into how effectively it can compensate for deficiencies in visual representations. The next section further dissects these findings.

4.4 How Do LLMs Do It?

Here, we attempt to discern what type of information the LLM can and cannot recover, and what the mechanism and limiting factors are behind that.

Prior work shows that different layers in vision encoders specialize in different functions, with early layers extracting low-level features and later layers encoding higher-level semantic information (Ghiasi et al., 2022; Dorszewski et al., 2025). Motivated by this, we progressively knock out self-attention in the upper layers of the vision encoder–blocking the last 6, 12, 18, or all 24 layers–and measure how well the language decoder can compensate for the lack in visual contextualization. The results are shown in Figure 5.

Deep Visual Contextualization is Fully Recoverable (by a strong LLM). When AK is applied to only the last six layers in LLaVA 13B (last 6 in Figure 5, right), the identifiability rate remains nearly indistinguishable from the No AK baseline. This finding suggests that the high-level semantic-feature building typically attributed to the last layers of the vision encoder can be fully reconstructed by the LLM. This result is intuitive in light of studies which establish the isotropy of the semantic spaces learned by even independently trained vision and language models (Huh et al., 2024). Yet, to the best of our knowledge, this is the first time this kind of dynamic division of labor between the modules of a VLM has been observed empirically.

The potential for full recovery, moreover, appears to be a function of model size. The results with BakLLaVA (Figure 5, left) show that a considerable gap remains in OPI scores even at six layers of AK in the CLIP encoder (and similarly for LLaVA 7B in Figure 15 in the Appendix.)

Some Part of Shallow Visual Contextualization is Non-recoverable. As we extend AK in the vision encoder to 12 and then to 18 layers, we observe that the OPI scores converge towards the numbers seen for full (24-layer) attention knockout. There is almost no difference in what the language decoder can recover from visual representations with six layers of early contextualization with zero layers of early contextualization, meaning that some critical low-level, purely visual feature extraction takes place in those first six layers of the encoder, which is beyond the scope of the LLM decoder.

Nevertheless, it bears repeating that even with a full 24-layer AK in the CLIP encoder, the LLM is able to compensate for much of the identifiability of object parts – this was the highlight finding of § 4.3. The analysis above sheds more light on what the LLM cannot compensate for, which appears to be rooted in shallow-layer visual feature extraction.

The Surge in Identifiability Adaptively Shifts. This is best seen in the results for LLaVA 13B, where the no AK baseline exhibits two clear surges, as discussed in §4.2. In Figure 5 (right), we see that the pattern of the surges is preserved across different levels of attention knockout in the vision

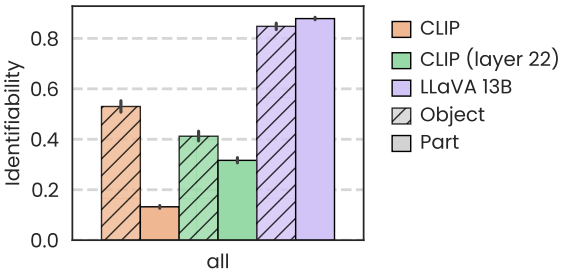

Figure 6: OPI scores for CLIP variants and LLaVA 13B model with scores aggregated across all parts.

encoder. But it shifts from earlier layers to later layers in the decoder, as AK in the vision encoder progresses from the last 6 layers, through 12, 18 and finally 24 layers. This delay reflects the compensatory behavior of the LLM decoder: as there is more missing context in the visual representation, the LLM has to dedicate more layers to recover it, effectively "writing" key information to the visual representation, before reading it back, i.e., before identifying the object part present in a region.

Lastly, it is worth noting that OPI recovery is observable despite an apparent architectural limitation: language models use a causal self-attention mechanism, meaning that the recontextualization of visual representations in the VLMs' decoders is happening unidirectionally. This may lead to more pronounced shifts in OPI surges, but it does not prevent LLaVA 13B, for example, from achieving full recoverability of visual semantic features.

4.5 Can CLIP Identify Object Parts?

Earlier we saw that different LLaVA variants yield a high OPI score which is largely dependent on the good contextualization of image patches in the CLIP vision encoder. Here, we test whether the CLIP text encoder itself is also able to read this information. Our implementation of the OPI probe is a modification of the standard approach for zero-shot image classification with CLIP.

Probe Implementation. In zero-shot image classification, CLIP's text encoder is used to obtain representations for all vocabulary tokens contextualized within a template ("A photo of {token}".) and the `<|endoftext|>` token is used to represent each candidate text. Tokens are then ranked based on their similarity to an input image `[CLS]` token representation. The standard approach for image classification cannot be used for patch-level analysis off-the-shelf, since the `[CLS]` token pulls information from all patches in the image. So we implement a simple localization method through self-attention blocking to ensure that the `[CLS]` token represents a specific image region. Concretely, we allow self-attention to be computed as usual up to a given layer, l, in the vision encoder, while for all subsequent layers, self-attention is constrained to the `[CLS]` token and patches of the target region. Unlike the AK interventions used earlier, the procedure described here is not a form of attention ablation, but a method used to focus the representational power of the `[CLS]` token onto the region of interest within the image. We empirically test different values for l and find that $l = 22$ works best (see Figure 9 in the Appendix).

Figure 6 presents CLIP's OPI scores, including earlier LLaVA 13B results for reference.

CLIP cannot identify object parts. The positive impact of the focusing technique on OPI scores is considerable (CLIP vs. CLIP, layer 22), and, as could be expected, zooming in on a particular part region has a negative impact on object identifiability. Still, the OPI rates for CLIP (layer 22) are far from the ceiling in performance set by LLaVA. Despite the considerable difference in language model size between CLIP and LLaVA, the gap in their performance more likely stems from deficiencies in the CLIP text encoder, which have been attested in prior work as well (Li et al., 2024). The text encoder is not as adept at reading object part information from the image representation, as it is with whole object information. Yet, this does not mean that the relevant information is not present in the CLIP visual encoder, as evidenced by the fact that LLMs can read it from its visual representations. Future work should investigate how CLIP-like models can be so expressive, if they are learned through the bottleneck of pooled representations and with reference to a weak text encoder.

5 Conclusion & Future Work

In this study, we investigated the internal mechanisms of vision-language models (VLMs) by evaluating their ability to identify object parts under a range of controlled attention ablations. Our findings reveal a nuanced and adaptive division of labor between the vision encoder and the language decoder, where an LLM can compensate for some level of deficiency in the visual representations, recovering and enriching semantics that would otherwise be absent. These findings have several im-

plications for future work. First, in more complex visual domains–where even fully contextualized visual features may fall short–the compensatory role of the LLM may become even more critical: LLMs could serve as a fallback mechanism in scenarios where visual representations from the visual encoder alone are sufficiently inexpressive. Second, our results suggest that future research could exploit the capabilities of LLM further for contextualization within VLM architectures. Specifically, disabling self-attention in certain layers of the vision encoder during pretraining–or rethinking the encoder-decoder interface entirely–could lead to more efficient models with deeper fusion of modalities from the ground up. It might be time to rethink VLM architectures: not as static pipelines, but as adaptive systems capable of redistributing computational burden between the vision and language components depending on the task.

6 Limitations

While our study provides valuable insights into the contextualization mechanisms of vision-language models, it also has some limitations. We use the Object-Part Identification (OPI) task which, despite no direct applied value, serves as an effective diagnostic probe for isolating and analyzing specific model behaviors in a controlled setting. In this analysis, we rely on a single dataset–Pascal Panoptic Parts which we carefully filter further. We selected this dataset primarily due to detailed, structured annotations of objects and their corresponding parts, which are essential for our task at hand. Most of the other segmentation datasets lack this level of granularity, making them unsuitable for studying contextualization in the way we do in this work. Furthermore, our study is limited to LLaVA-style models (Liu et al., 2023, 2024a; Zhou et al., 2024), which pass all patch representations directly to the LLM decoder; our method does not extend to architectures like BLIP-2 (Li et al., 2023b), Qwen2-VL (Wang et al., 2024) or Phi-3.5-Vision[7], which instead rely on sampling or pooling of the original patch representations. We leave the exploration of how our probing framework can be adapted to these newer architectures for future work.

[7] https://huggingface.co/microsoft/Phi-3.5-vision-instruct

References

Reza Abbasi, Ali Nazari, Aminreza Sefid, Mohammadali Banayeeanzade, Mohammad Hossein Rohban, and Mahdieh Soleymani Baghshah. 2025. Analyzing clip's performance limitations in multi-object scenarios: A controlled high-resolution study. *Preprint*, arXiv:2502.19828.

Samyadeep Basu, Martin Grayson, C. Morrison, Besmira Nushi, S. Feizi, and Daniela Massiceti. 2024. Understanding information storage and transfer in multi-modal large language models. *Neural Information Processing Systems*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Daan de Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. 2021. Part-aware panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Teresa Dorszewski, Lenka Tětková, Robert Jenssen, Lars Kai Hansen, and Kristoffer Knutsen Wickstrøm. 2025. From colors to classes: Emergence of concepts in vision transformers. *Preprint*, arXiv:2503.24071.

C. Eichenberg, Sid Black, Samuel Weinbach, Letitia Parcalabescu, and A. Frank. 2021. Magma - multimodal augmentation of generative models through adapter-based finetuning. *Conference on Empirical Methods in Natural Language Processing*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv: 2306.13394*.

Yossi Gandelsman, Alexei A. Efros, and Jacob Steinhardt. 2024. Interpreting CLIP's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. 2025. Interpreting the second-order effects of neurons in CLIP. In *The Thirteenth International Conference on Learning Representations*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and A. Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *Conference on Empirical Methods in Natural Language Processing*.

Amin Ghiasi, Hamid Kazemi, Eitan Borgnia, Steven Reich, Manli Shu, Micah Goldblum, Andrew Gordon Wilson, and Tom Goldstein. 2022. What do vision transformers learn? a visual exploration. *Preprint*, arXiv:2212.06727.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis. In *International Conference on Machine Learning*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv: 2310.06825*.

Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. 2024. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. *Preprint*, arXiv:2411.16724.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. 2024. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *International Conference on Machine Learning (ICML)*.

Martha Lewis, Nihal V. Nayak, Peilin Yu, Qinan Yu, Jack Merullo, Stephen H. Bach, and Ellie Pavlick. 2024. Does clip bind concepts? probing compositionality in large image models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 101–115.

Junnan Li, Dongxu Li, S. Savarese, and Steven C. H. Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *International Conference on Machine Learning*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Siting Li, Pang Wei Koh, and Simon Shaolei Du. 2024. Exploring how generative mllms perceive more than clip with the same vision encoder. *arXiv preprint arXiv: 2411.05195*.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. 2024. Evaluating text-to-visual generation with image-to-text generation. *arXiv preprint arXiv:2404.01291*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Preprint*, arXiv:2304.08485.

Shi Liu, Kecheng Zheng, and Wei Chen. 2024b. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. *arXiv preprint arXiv: 2407.21771*.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

Timo Lüddecke and Alexander Ecker. 2022. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096.

Panagiotis Meletis, Xiaoxiao Wen, Chenyang Lu, Daan de Geus, and Gijs Dubbelman. 2020. Cityscapes-panoptic-parts and pascal-panoptic-parts datasets for scene understanding.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36. ArXiv:2202.05262.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. 2025. Towards interpreting visual information processing in vision-language models. In *The Thirteenth International Conference on Learning Representations*.

Nostalgebraist. 2020. Interpreting gpt: The logit lens. LessWrong.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. 2024. DOCCI: Descriptions of Connected and Contrasting Images. In *ECCV*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. 2023a. Multimodal neurons in pretrained text-only transformers. *Preprint*, arXiv:2308.01544.

Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. 2023b. Multlimodal neurons in pretrained text-only transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2862–2867.

Jie-Jing Shao, Jiang-Xin Shi, Xiao-Wen Yang, Lan-Zhe Guo, and Yu-Feng Li. 2023. Investigating the limitation of clip models: The worst-performing categories. *Preprint*, arXiv:2310.03324.

Gabriela Ben Melech Stan, Raanan Yehezkel Rohekar, Yaniv Gurwicz, Matthew Lyle Olson, Anahita Bhiwandiwalla, Estelle Aflalo, Chenfei Wu, Nan Duan, Shao-Yen Tseng, and Vasudev Lal. 2024. Lvlm-intrepret: An interpretability tool for large vision-language models. *Preprint*, arXiv:2404.03118.

Gaurav Verma, Minje Choi, Kartik Sharma, Jamelle Watson-Daniels, Sejoon Oh, and Srijan Kumar. 2024. Cross-modal projection in multimodal LLMs doesn't really project visual attributes to textual space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 657–664, Bangkok, Thailand. Association for Computational Linguistics.

Martina G. Vilas, Timothy Schaumlöffel, and Gemma Roig. 2023. Analyzing vision transformers for image classification in class embedding space. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Feng Wang, Jieru Mei, and Alan Yuille. 2025. Sclip: Rethinking self-attention for dense vision-language inference. In *Computer Vision – ECCV 2024*, pages 315–332, Cham. Springer Nature Switzerland.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Meng Wei, Xiaoyu Yue, Wenwei Zhang, Shu Kong, Xihui Liu, and Jiangmiao Pang. 2023. Ov-parts: Towards open-vocabulary part segmentation. *Preprint*, arXiv:2310.05107.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2024. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv: 2409.02813*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024a. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

Zeliang Zhang, Zhuo Liu, Mingqian Feng, and Chenliang Xu. 2024b. Can CLIP count stars? an empirical study on quantity bias in CLIP. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1081–1086, Miami, Florida, USA. Association for Computational Linguistics.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. 2024c. Cross-modal information flow in multimodal large language models. *arXiv preprint arXiv:2411.18620*.

Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. 2024. Tinyllava: A framework of small-scale large multimodal models. *Preprint*, arXiv:2402.14289.

A Dataset Filtering

In order to maintain annotation consistency and visual clarity, we specifically focus on the subset of animal classes within the Pascal Panoptic Parts (Pascal-PP) dataset (de Geus et al., 2021; Meletis et al., 2020) dataset as described in Section 3. The choice is motivated by the relatively uniform and well-defined part structures of animals, as well as their good coverage in the vocabulary of the VLMs we evaluate. To reduce noise and complexity further, we sequentially apply the following filtering criteria:

1. Each image contains exactly one instance of the target object (e.g., a single cat). While other non-target objects may be present, multiple instances of the target objects are excluded to avoid ambiguity.

2. The target object must occupy at least 20% of the image pixels, ensuring the object is visually prominent enough.

3. If the target object is fully masked, it must not be mentioned in the caption generated by the VLM to ensure that object and part recognition rely on visible image patches rather than memorization or contextual hallucination.

After filtering, we sample up to 100 images per class, where available. Table 1 provides further details about the distribution of images across different object classes.

B Additional Results

Figures 10 to 13 presents the layer-wise evolution of per-object part identifiability across different attention knockout settings described in the Section 4.3 for TinyLLaVA 1.1B, BakLLaVA 7B, LLaVA 7B and LLaVA 13B models.

Figures 14 to 16 presents the layer-wise evolution of per-object part identifiability when progressively modifying self-attention across different layers of the CLIP Encoder for BakLLaVA 7B, LLaVA 7B and LLaVA 13B models.

C Relationship between identifiability and frequency of object-parts

In this section, we examine whether the OPI scores correlate with the frequency of object-part co-occurrences or just parts irrespective of the associated objects in the training data of the models we evaluate on. All the models employ a two-stage training process where the first stage focuses on alignment using the image captioning data and the second stage focuses on instruction fine-tuning using the multimodal instruction data. We combine the data instances from both stages and compute the co-occurrence counts of objects and their respective parts in the ground-truth responses using the following pipeline:

Text Normalization. We first lowercase all responses and remove punctuation. Next, each word is stemmed using the Porter Stemmer to match morphological variants (e.g., "legs" vs. “leg”).

Lexical Expansion. Each object term and its corresponding parts are expanded by collecting synonyms and hyponyms from WordNet. This expansion of the target set allows for broader lexical coverage and more precise matching of candidate terms. For consistency, we also apply stemming to the terms in the expanded set.

Matching. We iterate through all instances and match the tokens with the target set of all object terms, part terms, and also keep track of their co-occurrence.

The heatmap in Figure 7 presents the co-occurrence counts between different objects (columns) and their associated parts (rows), whereas the bar plot displays the overall frequency of individual parts irrespective of object association. We can observe that co-occurrence frequencies vary widely across object-part pairs. Common parts such as *head*, *leg*, and *eye* frequently appear with the *person* class, whereas parts like *hoof* and *wing* are more sparsely represented, mainly occurring with specific objects like *horse* and *bird* respectively. One might hypothesize that object parts associated with frequently occurring classes – particularly *person* would exhibit higher identifiability. However, analysis of OPI scores (see Table 3) reveals no clear correlation between co-occurrence frequency and identifiability. In fact, parts with relatively low frequency often achieve high identifiability. Similarly, in Figure 4, parts such as *eye* in the context of *dog* and *cat*, and *leg* in the context of *bird* and *person* are identified accurately despite their lower co-occurrence counts. These findings suggest that the identifiability of object parts in VLMs is not predominantly driven by training data frequency. Instead, VLMs are capable of implicitly learning object-part associations leveraging strong visual or semantic priors, and exhibit effective generalization capabilities, even without explicitly being trained for fine-grained object parts recognition.

D Relationship between identifiability rate and size of object parts

In this section, we investigate whether the identifiability of object parts is influenced by their relative size. Intuitively, one might expect larger parts to be easier to identify due to their increased visibility and spatial prominence in images. To test this, we consider the CLIP model (with best-performing configuration $l = 22$; see Section 4.5) and three LLaVA variants: BakLLaVA 7B, LLaVA 7B, and LLaVA 13B. We begin by binning object parts sizes into four discrete groups based on their relative area in the image space. We then analyze the identifiability rates of object parts across these bin groups. Figure 8 presents identifiability scores for few object-part pairs such as *eye - cat*, *head - cow*, and *leg - horse*, as well as aggregate trends over all parts.

Our analysis highlights a key difference between the CLIP and LLaVA models. For CLIP, identifiability scores generally increase with part size for several part-object combinations such as *eye - cat*, *leg - horse*, and *paw - dog*, indicating a positive correlation between part size and identifiability. However, for certain parts like *head - cow*, identifiability remains consistent across bin groups. In contrast, the three LLaVA variants show little to no

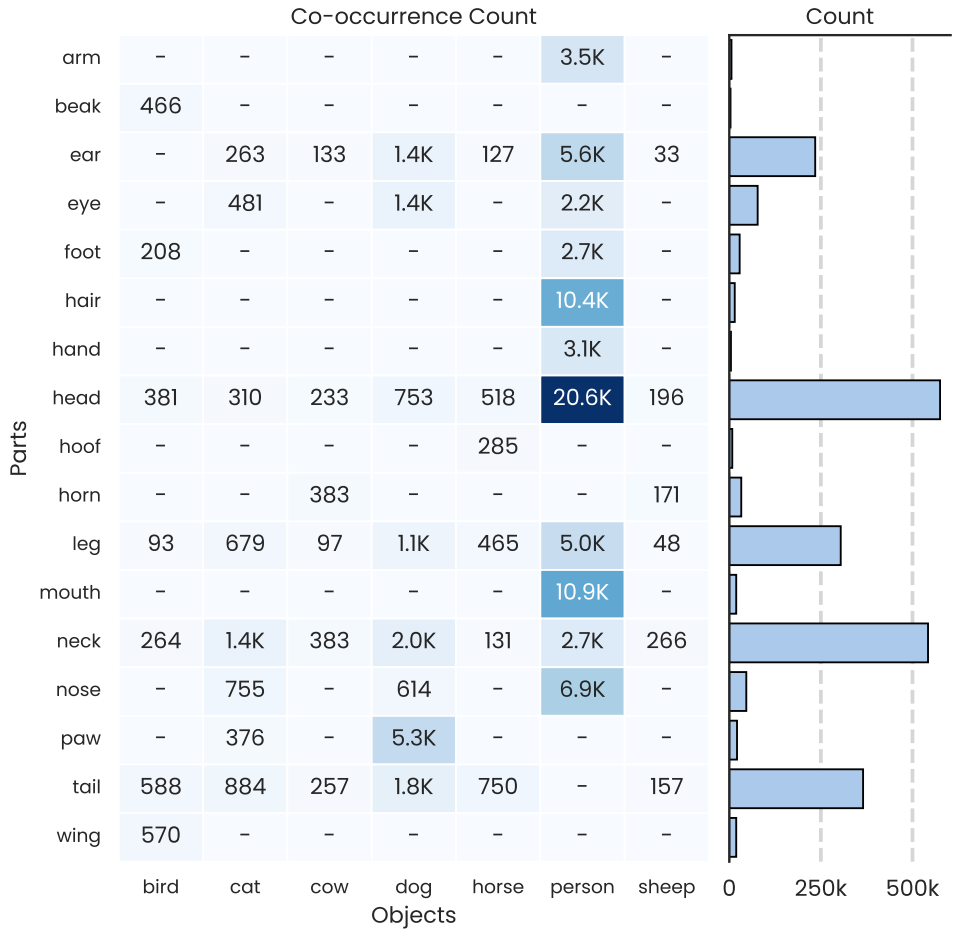

Figure 7: Statistics of frequency counts on the LLaVA dataset from both the pre-training alignment and instruction fine-tuning phase. The heatmap illustrates the frequency of object-part co-occurrence with the associated objects in the context, whereas the barplot shows the frequency of counts of object-parts irrespective of the associated objects.

variation in identifiability with respect to part size. Across all bin groups, a consistent gap in identifiability exists between CLIP model and LLaVA variants, though the gap narrows as the part size increases. These results demonstrate that while part size moderately influences identifiability in CLIP, it has minimal impact on the LLaVA variants, suggesting insensitivity to part size in current VLMs.

E Potential Application of LLaVA for Segmentation Task

In this section, we test the extent to which LLaVA's strong object and part recognition capabilities translate to the task of object parts segmentation. We utilize the OVParts benchmark (Wei et al., 2023) to evaluate segmentation performance across both LLaVA 7B and 13B models. Specifically, we assess the models' ability to perform part segmentation over a subset of 850 images from the OVParts validation set. To ensure fair comparison, we follow LLaVA's original preprocessing pipeline across all samples. Unlike traditional segmentation models, which are trained with pixel-level supervision, LLaVA was not trained for segmentation and instead operates at a patch-based resolution. To adapt LLaVA for segmentation, we map the token with the highest ranking to all pixels in its associated patch. While this approach may produce coarse segmentations and introduce misalignments, it allows us to quantify LLaVA's segmentation performance in an open-vocabulary setting without additional training.

For comparison, we benchmark against CLIPSeg (Lüddecke and Ecker, 2022), a model fine-tuned for segmentation and among the top-performing entries on the OVParts leaderboard. Unlike LLaVA models, CLIPSeg benefits from supervision explicitly tailored to the segmentation task, allowing it to generate high-fidelity pixel-wise masks. We

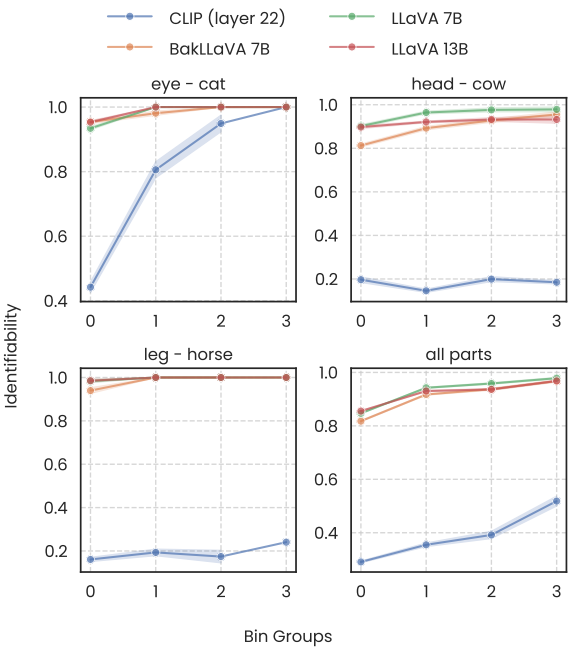

Figure 8: Qualitative results illustrating the relationship between OPI scores and part size across bin groups for CLIP model (best-performing configuration) and three LLaVA variants: BakLLaVA 7B, LLaVA 7B, and LLaVA 13B models. The last plot shows aggregated OPI scores across all object parts.

Model	mIoU
CLIPSeg (baseline)	**22.31**
LLaVA 7B	14.66
LLaVA 13B	14.81

Table 2: Comparison of mIoU scores on OVParts part segmentation between LLaVA 7B/13B models and CLIPSeg specifically tuned for segmentation.

report the mean IoU (mIoU) scores (Long et al., 2015) in Table 2. We find that LLaVA 7B and 13B models achieve comparable mIoU scores, but they underperform the CLIPSeg baseline considerably. The performance gap is expected given LLaVA's lack of segmentation-specific training and the mismatch between patch-level predictions and pixel-level evaluation metrics. However, the better-than-random mIoU scores again attest to LLaVA's ability to identify object parts and suggest that the model could be a good starting point for dedicated segmentation training.

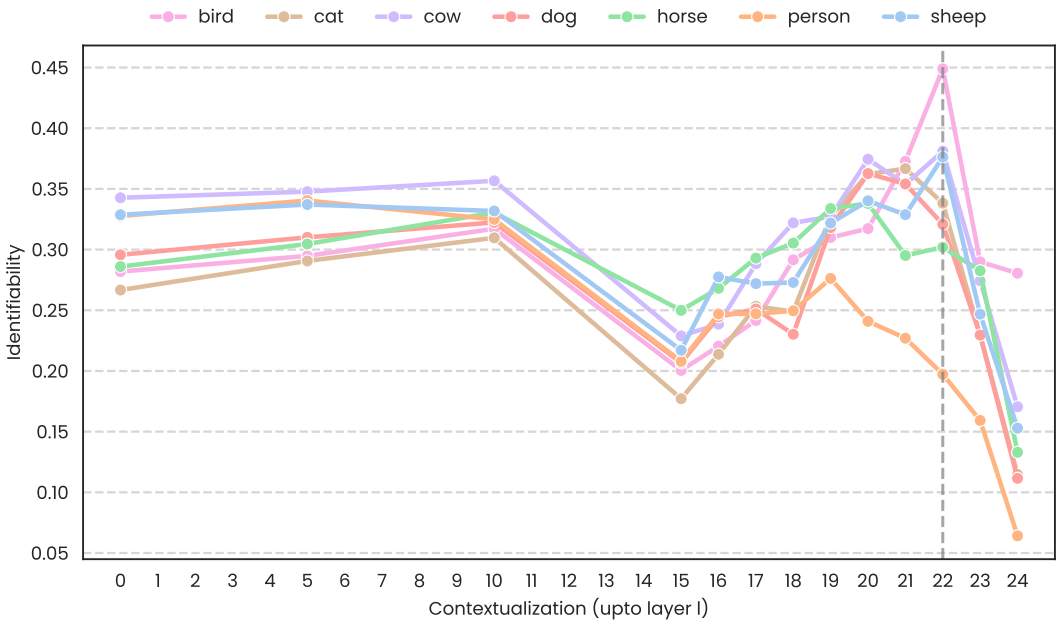

Figure 9: Effect of layer-wise localization in the CLIP Encoder on per-object part identifiability using attention knockout.

Object Category	BakLLaVA 7B		LLaVA 7B		LLaVA 13B		TinyLLaVA 1.1B	
	part	object	part	object	part	object	part	object
bird	0.84	0.83	0.89	0.83	0.90	0.85	0.82	0.90
cat	0.87	0.86	0.88	0.88	0.89	0.89	0.74	0.87
cow	0.90	0.92	0.95	0.95	0.92	0.93	0.88	0.94
dog	0.88	0.90	0.92	0.91	0.91	0.91	0.72	0.93
horse	0.89	0.90	0.92	0.91	0.88	0.94	0.80	0.94
person	0.78	0.51	0.79	0.57	0.82	0.52	0.67	0.45
sheep	0.86	0.96	0.92	0.96	0.89	0.94	0.83	0.93

Table 3: OPI scores for BakLLaVA 7B, LLaVA 7B, LLaVA 13B and TinyLLaVA 1.1B models, showing part-level and object-level identifiability aggregated across all parts.

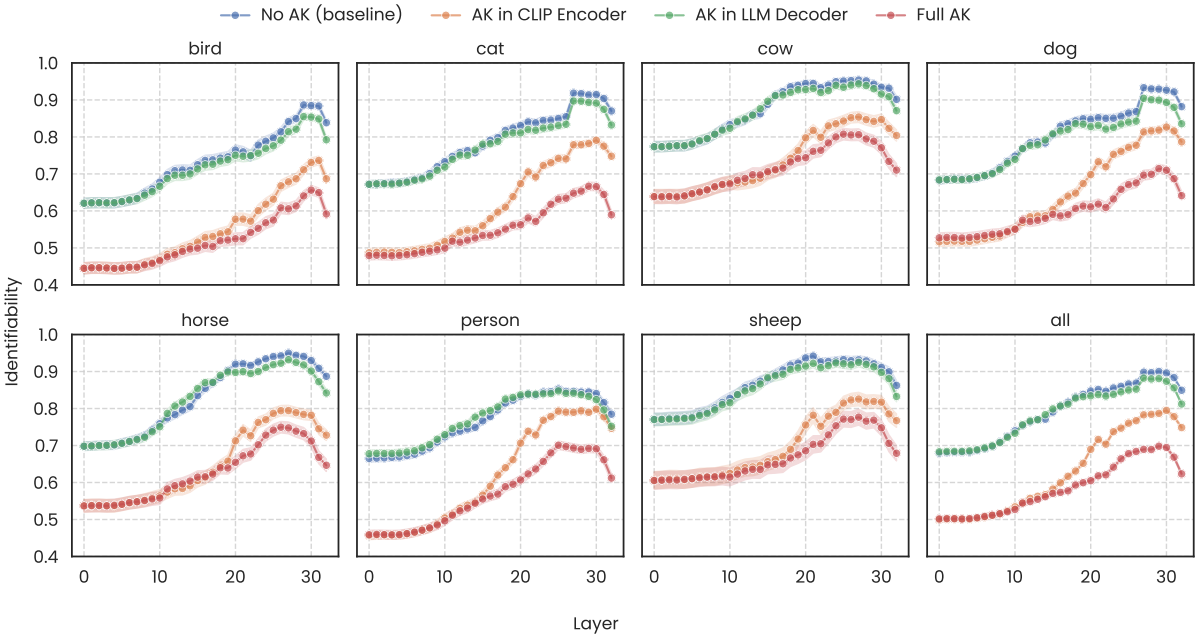

Figure 10: Layer-wise evolution of per-object part identifiability across different attention knockout (AK) settings in the LLM Decoder of the BakLLaVA 7B model. The last plot shows aggregated identifiability scores across all parts. "No AK" denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while "Full AK" indicates modified self-attention in both.

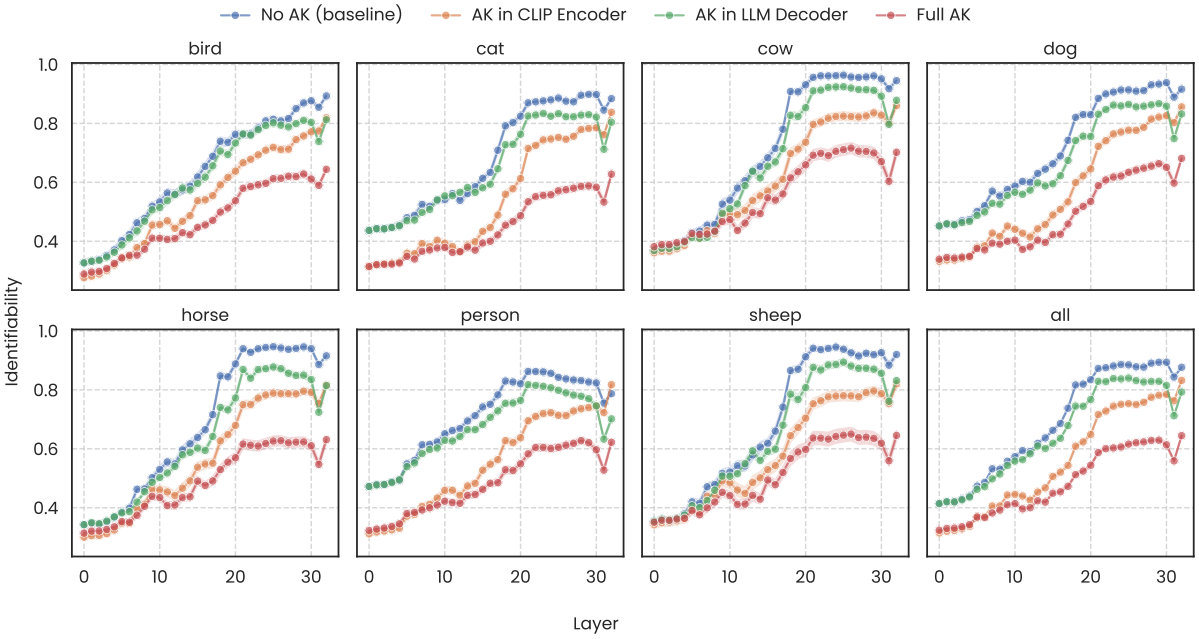

Figure 11: Layer-wise evolution of per-object part identifiability across different attention knockout (AK) settings in the LLM Decoder of the LLaVA 7B model. The last plot shows aggregated identifiability scores across all parts. "No AK" denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while "Full AK" indicates modified self-attention in both.

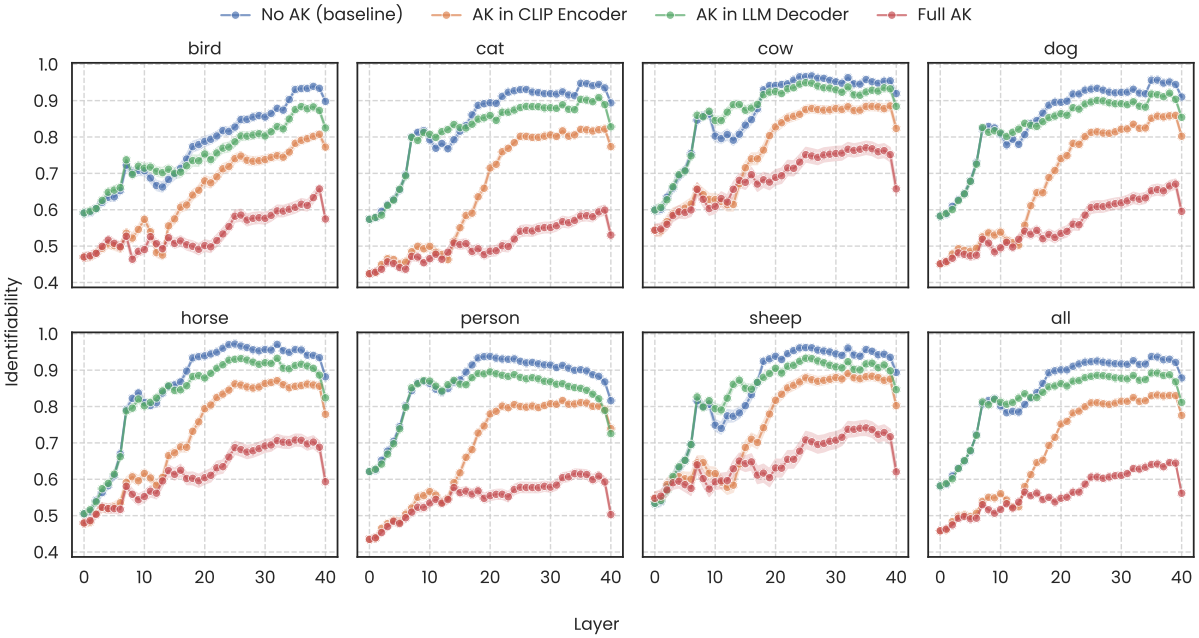

Figure 12: Layer-wise evolution of per-object part identifiability across different attention knockout (AK) settings in the LLM Decoder of the LLaVA 13B model. The last plot shows aggregated identifiability scores across all parts. "No AK" denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while "Full AK" indicates modified self-attention in both.

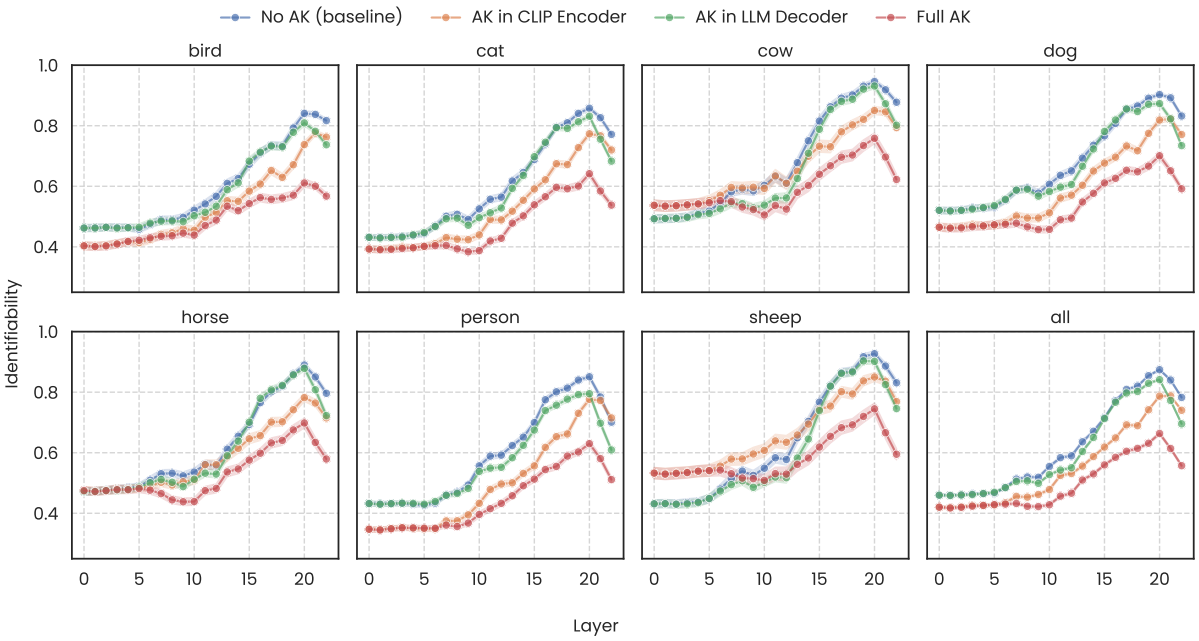

Figure 13: Layer-wise evolution of per-object part identifiability across different attention knockout (AK) settings in the LLM Decoder of the TinyLLaVA 1.1B model. The last plot shows aggregated identifiability scores across all parts. "No AK" denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while "Full AK" indicates modified self-attention in both.

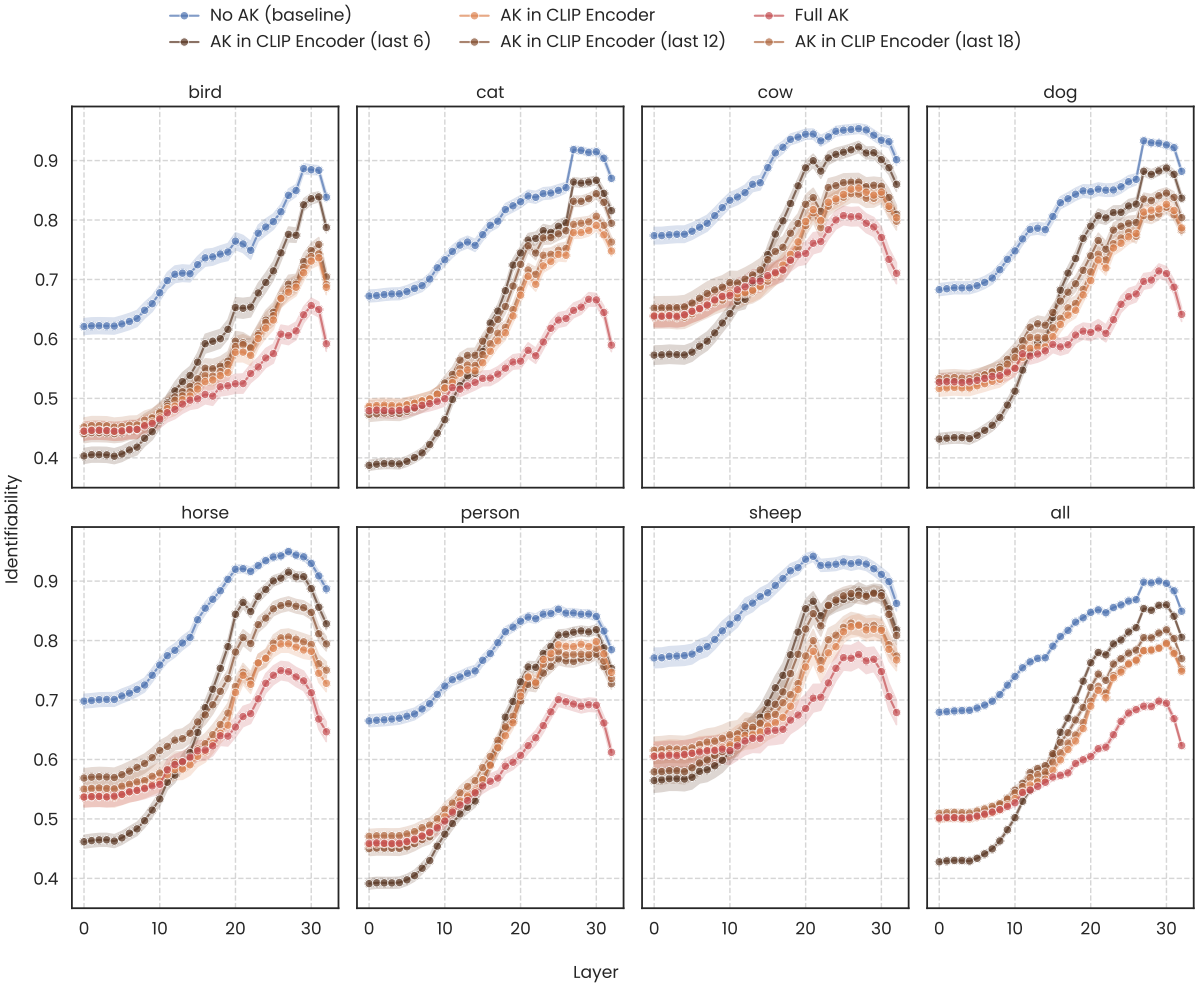

Figure 14: Layer-wise evolution of per-object part identifiability in the LLM Decoder when progressively modifying self-attention across different CLIP Encoder layers in the BakLLaVA 7B model. The last plot presents the aggregated identifiability trends across all objects. "No AK" denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while "Full AK" indicates modified self-attention in both.

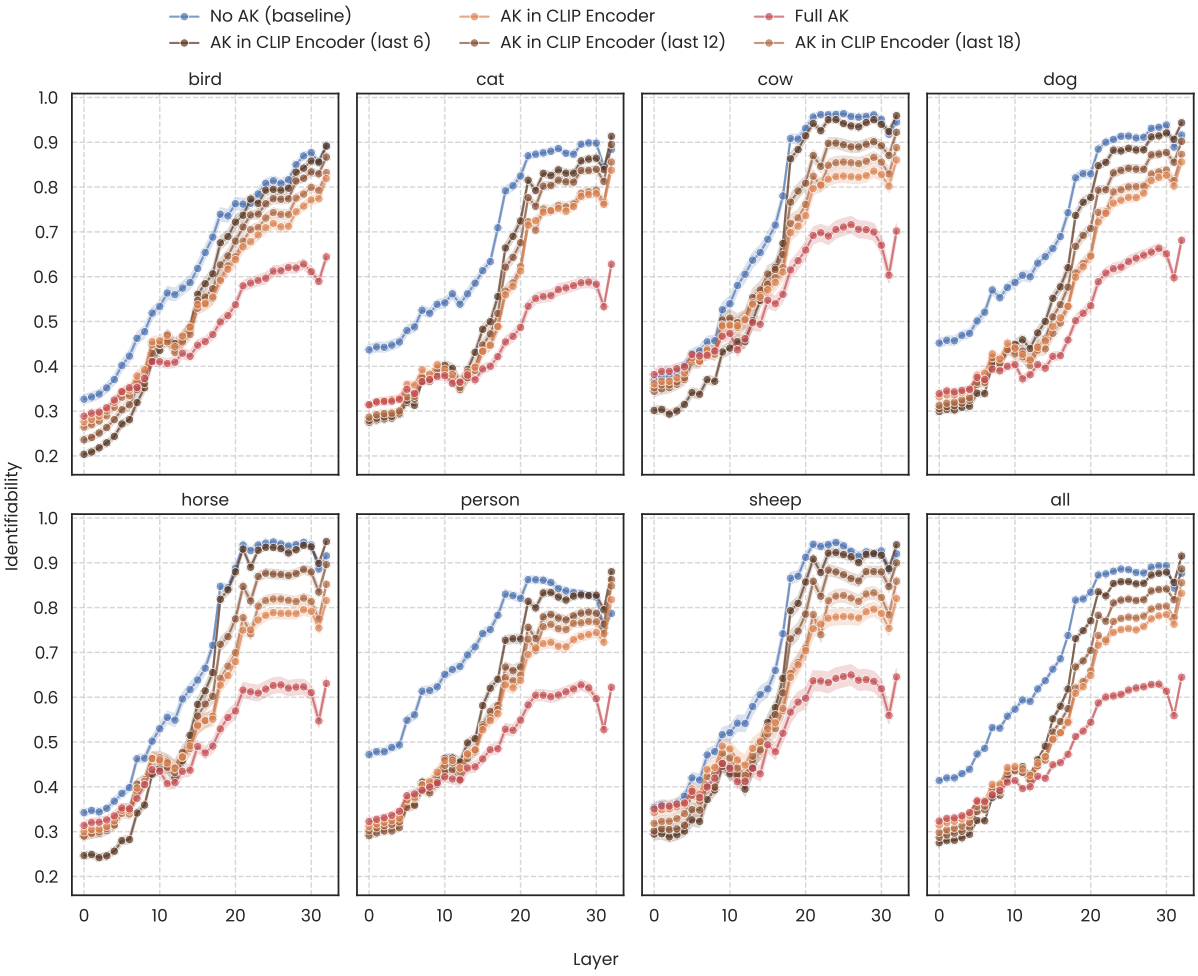

Figure 15: Layer-wise evolution of per-object part identifiability in the LLM Decoder when progressively modifying self-attention across different CLIP Encoder layers in the LLaVA 7B model. The last plot presents the aggregated identifiability trends across all objects. “No AK” denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while “Full AK” indicates modified self-attention in both.

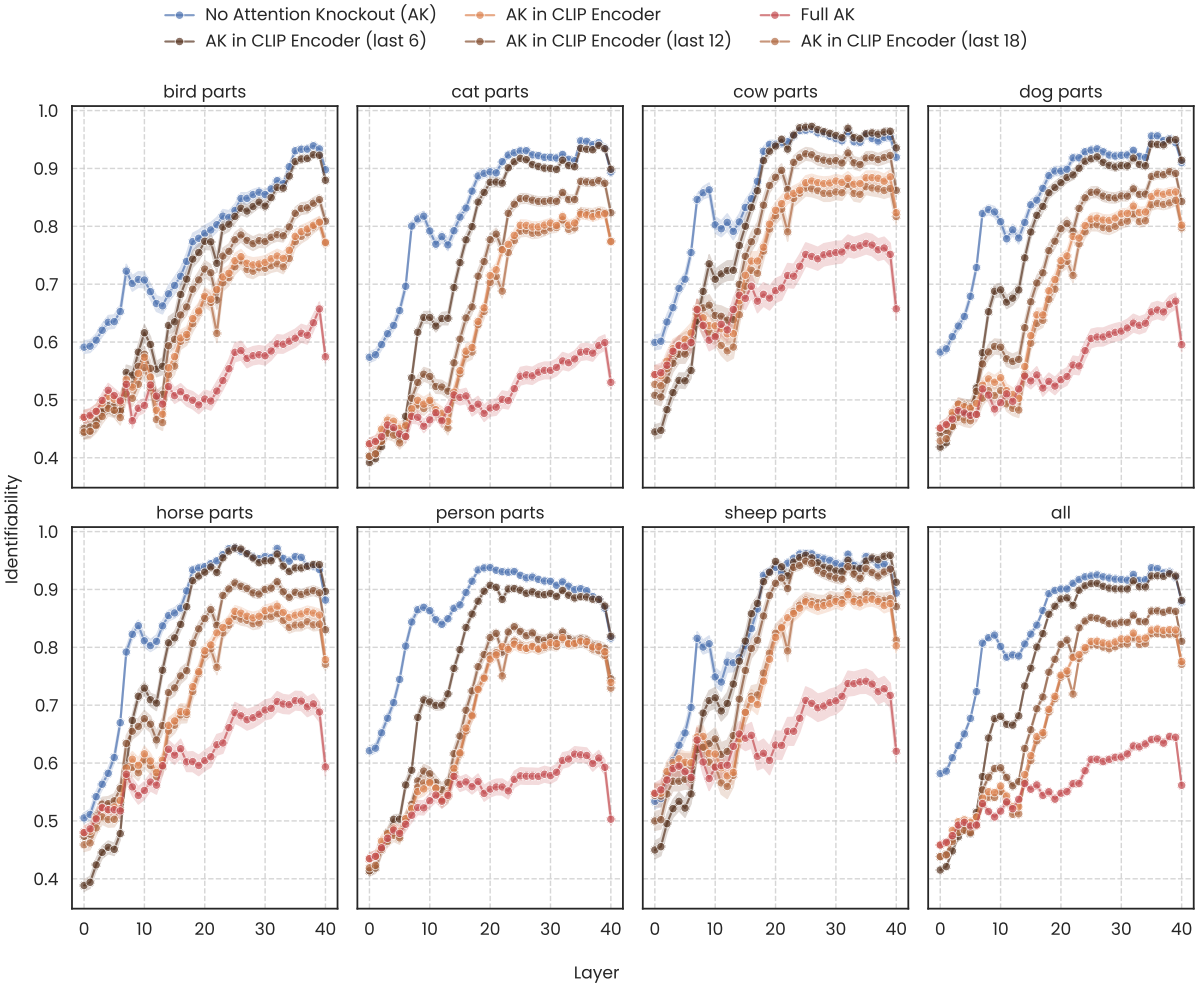

Figure 16: Layer-wise evolution of per-object part identifiability in the LLM Decoder when progressively modifying self-attention across different CLIP Encoder layers in the LLaVA 13B model. The last plot presents the aggregated identifiability trends across all objects. "No AK" denotes unaltered self-attention in both CLIP Encoder and LLM Decoder, while "Full AK" indicates modified self-attention in both.