Explicit Visual Alignment for Multimodal Large Language Model

Anonymous ACL submission

Abstract

Most existing MLLMs (Multimodal Large Language Models) rely on implicit vision–language feature alignment, which enables strong global reasoning but often struggles with fine-grained visual perception tasks. To bridge this gap, we propose **E-Align** (Explicit Visual Alignment), a new paradigm that shifts from opaque latent feature mapping to interpretable textual alignment. Building on E-Align, we develop **Explicit-VL**, which integrates a lightweight, parameter-efficient **explicit adapter** that converts visual encoder outputs into structured textual hints. Unlike implicit vectors, these hints are naturally aligned with the LLM's symbolic processing, guiding it to attend to specific visual details. Extensive experiments demonstrate that Explicit-VL excels at knowledge-based VQA and vision-centric understanding, hallucinates less, and performs better on object detection. Overall, our results suggest that E-Align strikes a better balance between global reasoning and fine-grained perception, and highlight its potential as a promising paradigm for more effective vision–language feature alignment. To facilitate future research, we have released our model and code at: https://anonymous.4open.science/r/ealign

1 Introduction

MLLMs (Bai et al., 2025; Li et al., 2024c,a,b; An et al., 2025; Alayrac et al., 2022; Li et al., 2023a; Zhu et al., 2023; Chen et al., 2023, 2024b; Wang et al., 2025) have achieved remarkable progress in high-level vision-language tasks (e.g., captioning, VQA), but they still lag far behind dedicated Visual Foundation Models (VFMs) (Li et al., 2025; Yin et al., 2023) in vision-centric scenarios that require fine-grained perception: for example, they fail to detect small objects, misjudge spatial relations, and hallucinate nonexistent objects. This gap stems from a fundamental contradiction in cross-modal alignment: language-supervised visual encoders such as CLIP (Radford et al., 2021)

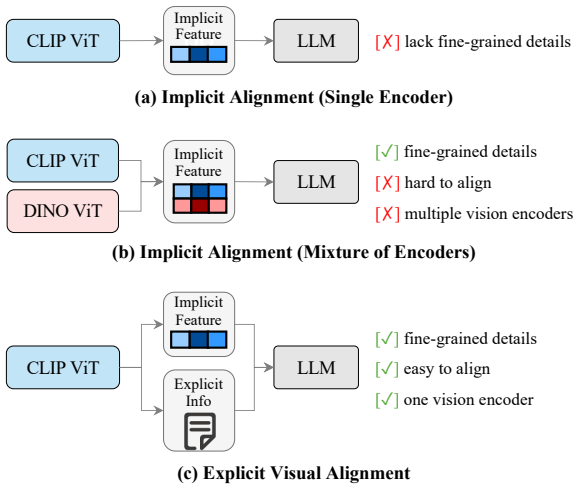

Figure 1: **Implicit vs. E-Align.** (a) *Implicit Alignment (Single Encoder)* projects CLIP's implicit features into the LLM, but often loses fine-grained details. (b) *Implicit Alignment (Mixture-of-Encoders)* improves fine-grained perception, but faces alignment challenges and redundant encoders. (c) *E-Align* (ours) converts CLIP's hidden states into structured textual hints, preserving fine-grained details, aligning naturally with the LLM, and requiring only one vision encoder.

are easy to align with LLMs but produce coarse-grained features, while non-language encoders like DINOv2 (Oquab et al., 2023; Siméoni et al., 2025) capture fine-grained details but have incompatible feature spaces that require prohibitive training overhead (Tong et al., 2024a; Fan et al., 2025).

Existing efforts to bridge this gap mainly rely on Mixture-of-Encoders approaches, which introduce additional vision encoders to supply fine-grained visual features. For example, Cambrian (Tong et al., 2024a) and Eagle (Shi et al., 2025a) aggregate four and five vision encoders pretrained with diverse objectives, respectively. However, this strategy typically introduces heavy auxiliary modules and can exacerbate alignment difficulty (Fan et al., 2025). This naturally raises the question: *Can we extract fine-grained visual details from a single, widely*

used encoder (e.g., CLIP) without updating its parameters or adding heavy modules?

To address this limitation, we propose **E-Align**, a novel paradigm that bridges the modality gap by transforming fine-grained visual signals into structured textual hints (e.g., "traffic light: [0.43, 0.46]"). As illustrated in Figure 1, instead of supplementing the model with opaque feature vectors, our approach provides explicit, discrete information in the form of structured textual hints. These hints are inherently compatible with the LLM's symbolic processing capabilities, thereby facilitating a more effective alignment. Crucially, rather than relying on an external, specialized encoder, our method generates these hints by extracting them directly from the hidden states of the standard CLIP encoder, without the overhead of an additional visual encoder.

Building on this paradigm, we introduce **Explicit-VL**, a MLLM that augments a standard CLIP encoder with object-level textual cues. Specifically, we propose a lightweight **explicit-adapter** that takes CLIP hidden states as input and decodes a set of explicit predictions, including object categories and bounding-box locations. These predictions are serialized into structured text and concatenated with the LLM input sequence as additional context tokens using the standard tokenizer.

We conduct extensive experiments across knowledge-based, vision-centric, hallucination, and object detection benchmarks. With only a 27M-parameter explicit-adapter, Explicit-VL consistently achieves higher performance on seven benchmarks, highlighting the strong potential of explicit visual alignment for improving fine-grained multimodal understanding.

In summary, our contributions are as follows:

- We propose **E-Align**, a new alignment paradigm that shifts from implicit latent feature mapping to *explicit* discrete textual priors, enabling MLLMs to access fine-grained visual details in a language-friendly form.
- We introduce a lightweight, parameter-efficient **explicit-adapter** that decodes hidden states from a frozen CLIP encoder into explicit object-level information and serializes it into structured text for the LLM, while keeping the CLIP encoder unchanged and its visual features unperturbed.
- We develop **Explicit-VL**, a MLLM built with an efficient multi-stage training pipeline. Extensive evaluations across knowledge-based, vision-centric, hallucination, and object detection benchmarks show that Explicit-VL consistently outperforms implicit-alignment-based MLLMs, achieving stronger visual perception and fine-grained understanding.

2 Related Work

To introduce more fine-grained details into MLLMs, prior work primarily focuses on improving the visual representations fed into the model. In general, this is achieved in two ways:

Mixture of encoders (MoE). A representative line of work enhances MLLMs by mixing visual features from different sources or specialized experts. Tong et al. (2024b,a) show that integrating language-supervised and self-supervised representations improves performance on vision-centric tasks. Along the same direction, Cambrian (Tong et al., 2024a), EAGLE (Shi et al., 2025b), LEO (Azadani et al., 2025), MoME (Shen et al., 2024), MoVA (Zong et al., 2024), and MOVE (Skripkin et al., 2025) further boost fine-grained capability by incorporating multiple specialized vision encoders.

Re-training the visual encoder. Another line improves fine-grained understanding by strengthening the encoder itself through multi-task training. Bolya et al. (2025) trains a unified visual encoder across diverse downstream tasks to obtain embeddings that generalize across modalities and objectives. Florence-VL (Chen et al., 2025) leverages Florence-2 (Xiao et al., 2024), a strong vision encoder pre-trained with sequence-to-sequence learning on billions of image-text annotations, offering versatile visual representations. However, re-training the encoder typically incurs prohibitively high training cost.

In contrast to both directions, our method neither introduces additional heavy encoders nor requires re-training the visual encoder. Instead, we obtain explicit information via a lightweight detection head and provide them in textual form, enabling the MLLM to access fine-grained information with minimal architectural changes.

3 Methodology

In this section, we introduce **Explicit-VL**, a multimodal large language model that strengthens

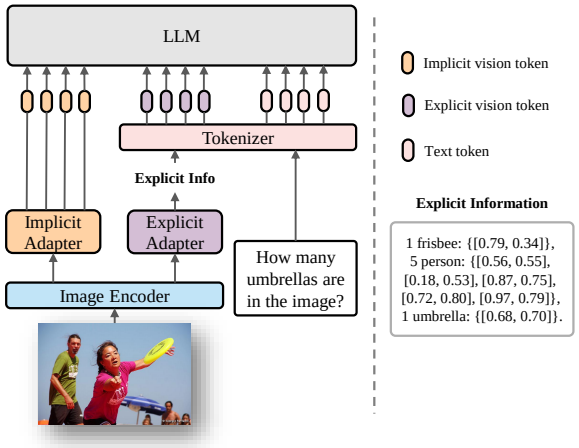

Figure 2: **Overview of Explicit-VL.** Explicit-VL introduces an *explicit adapter* that decodes intermediate CLIP features into structured textual object descriptions. These explicit hints are appended to the prompt and tokenized as standard text, enabling the LLM to better capture fine-grained visual details.

fine-grained visual understanding by incorporating an **explicit adapter** to provide structured textual visual cues alongside standard visual features. Sec. 3.1 describes the overall architecture of Explicit-VL and how the visual and textual inputs are organized for the LLM. Sec. 3.2 then details the design and implementation of the explicit adapter, which enables high-resolution, efficient object detection with a frozen CLIP-ViT backbone.

3.1 Overall Architecture

As illustrated in Fig. 2, Explicit-VL consists of four components: (1) a vision encoder $\mathcal{E}_{\text{clip}}$, (2) a large language model $\mathcal{M}$, (3) an implicit adapter $\mathcal{A}_{\text{imp}}$, and (4) an explicit adapter $\mathcal{A}_{\text{exp}}$. During inference, Explicit-VL proceeds in the following steps.

Step 1: CLIP feature extraction. Given an image I and a user question Q, we extract multi-layer patch features with a CLIP vision encoder:

$$H^{1:L} = \mathcal{E}_{\text{clip}}(I). \tag{1}$$

Following LLaVA-style practice, we use the penultimate layer feature H^{L-1} to produce standard continuous visual tokens, and tap a set of intermediate-layer features $H^{\mathcal{S}} = \{H^{\ell}\}_{\ell \in \mathcal{S}}$ to infer explicit visual information. L represents total number of layers, ℓ represents layer index, $\mathcal{S}$ represents set of intermediate layers.

Step 2: Implicit visual projection. We obtain continuous visual tokens through a lightweight implicit adapter:

$$Z = \mathcal{A}_{\text{imp}}(H^{L-1}). \tag{2}$$

Step 3: Explicit visual projection. In parallel, an explicit adapter predicts object detections from intermediate CLIP features:

$$\mathcal{D} = \mathcal{A}_{\text{exp}}(H^{\mathcal{S}}), \qquad \mathcal{D} = \{(c_i, b_i)\}_{i=1}^{M}, \tag{3}$$

where c_i is the category, and $b_i = (\hat{x}_i, \hat{y}_i)$ denotes the *normalized bounding-box center coordinate*. We then integrate $\mathcal{D}$ into a compact, structured text hint T (grouping coordinates by category), so that explicit visual cues enter the LLM purely as standard text tokens.

Step 4: LLM generation. Let $\tau(\cdot)$ denote the tokenizer and $\mathrm{Emb}(\cdot)$ denote the embedding lookup of the LLM. We append the explicit hint T to the question Q, embed the resulting text, and concatenate it with the continuous visual tokens Z to form the LLM input. The LLM then produces the answer token sequence $\mathbf{y}$:

$$\mathbf{y} = \mathcal{M}\big(\big[\, Z;\, \mathrm{Emb}(\tau(T \oplus Q))\,\big]\big)\,. \tag{4}$$

3.2 Explicit Adapter

Explicit information refers to text-form analysis results derived from the image. Compared to image captions that provide a holistic description, explicit information is more precise and task-oriented. In this work, we use object detection as explicit information to provide fine-grained details about image.

Extracting reliable object detections from a CLIP-ViT backbone is challenging. First, CLIP is pretrained with image–text contrastive learning rather than detection, and fine-tuning the CLIP backbone can degrade the general visual representations required by MLLMs. Second, CLIP-ViT-L/336 operates at 336×336 resolution, which limits its localization accuracy for small objects. Third, the extra detection computation may slow down MLLM training and inference.

To address these constraints, our **Explicit Adapter** is designed with the following features: (1) perform detection with a **frozen backbone** to preserve CLIP-ViT's general visual representations; (2) support inputs up to **672×672 resolution** to improve small-object detection; and (3) adopt an efficient FPN + Faster R-CNN architecture for **low-latency** decoding. Specifically, the explicit adapter consists of three steps:

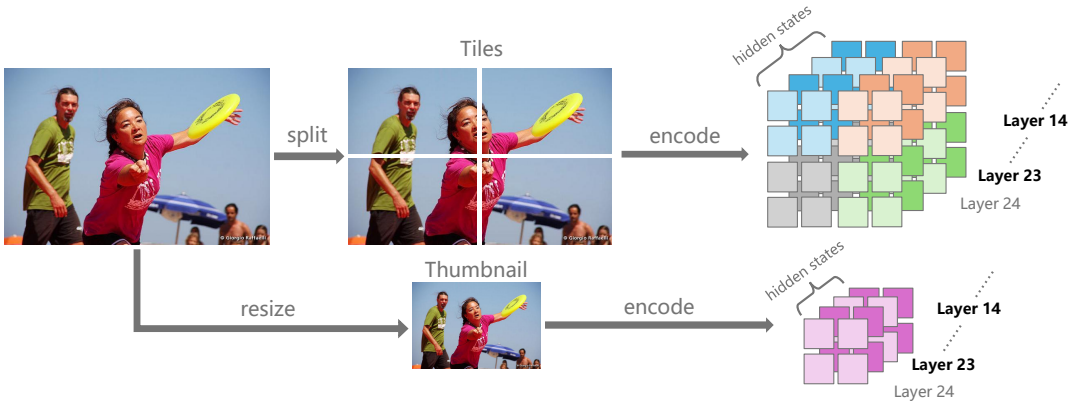

Figure 3: **Multi-scale feature acquisition.** To support high-resolution detection, we reassemble the hidden-state features of the tiles generated by the AnyRes strategy according to their spatial positions in the original image, yielding high-resolution features. We then extract features from the 14th and 23rd CLIP layers for both the tiled and thumbnail representations, producing four feature (from high to low resolution) for subsequent pyramid construction.

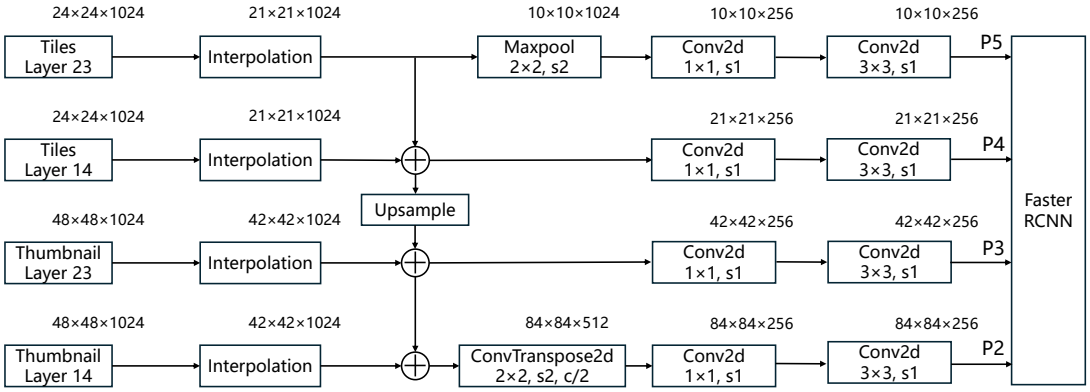

Figure 4: **Top-down feature fusion.** We convert the hidden-layer features from the CLIP ViT into detection-ready pyramid feature maps $\{P_5, P_4, P_3, P_2\}$ through interpolation, top-down fusion, and convolutional refinement. The symbol $\oplus$ denotes concatenating the upsampled higher-level feature with the current-level feature, followed by a projection to reduce the channel dimension back to that of the current level.

Step 1: Multi-scale feature acquisition. We adopt AnyRes (Liu et al., 2024a) to increase the input resolution for Explicit-VL. Meanwhile, the tile features produced by AnyRes can be directly reused to scale up the detection resolution. As illustrated in Figure 3, each tile in the 2×2 grid is encoded by CLIP into a $[24, 24, 1024]$ feature map. We then reassemble the patch tokens from the four tiles according to their spatial positions in the original image, producing a fused $[48, 48, 1024]$ feature map. For AnyRes outputs with non-2×2 layouts (e.g., 3×1), we first pad the fused tiled feature map to square and then resize it to obtain a $[48, 48, 1024]$ feature map. Next, we select the 14th and 23rd layer features from both the fused tile feature and the thumbnail, yielding a set of multi-scale feature maps: [tiles layer-14, tiles layer-23, thumbnail layer-14, thumbnail layer-23]. This strategy provides high-resolution signals for subsequent detection without introducing any additional computation.

Step 2: Top-down feature fusion. As illustrated in Fig. 4, to match the target pyramid strides $\{1/4, 1/8, 1/16, 1/32\}$, we first resize the thumbnail features to $[21, 21, 1024]$ and the tiles features to $[42, 42, 1024]$ via interpolation. We then perform a top-down fusion over these multi-level features. Different from standard FPN (Lin et al., 2017), which performs top-down fusion by element-wise summation, we fuse features by *channel stacking*: features from the upper level are concatenated with those from the current level and projected back to the current-level dimension. Finally, the four pyramid levels are adjusted by `upsample`, `identity`, `identity`, and `downsample`, respectively, and compressed to 256 channels, producing

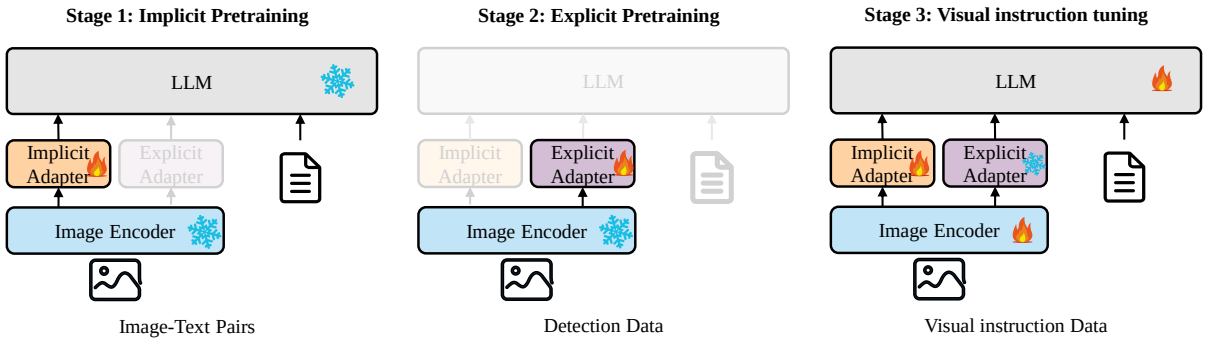

Figure 5: **E-Align.** Three stages: (1) **Implicit pretraining:** an implicit adapter is trained on captioning data to align CLIP features with the LLM's embedding space. (2) **Explicit pretraining:** an explicit adapter is trained on detection data to convert CLIP intermediate features into discrete textual signals. (3) **Visual instruction tuning:** the LLM, implicit adapter, and visual encoder are jointly trained on visual instruction data, enabling the model to integrate global semantic features with explicit object-level information.

the pyramid features $\{P_2, P_3, P_4, P_5\}$.

Step 3: Detection output structuring. Given the pyramid features, we adopt a Faster R-CNN head to produce object detections. We then serialize the final detections into a compact structured text for LLM ingestion: following (Jiao et al., 2024), each instance is represented by the (normalized) center of its box, grouped by category, and preceded by an instruction sentence. An example output is:

```
Here are the central coordinates of
certain objects in this image:
1 frisbee: {[0.79, 0.34]},
5 person: {[0.18, 0.53], [0.72, 0.80],
[0.86, 0.75], [0.60, 0.53], [0.97,
0.79]},
1 umbrella: {[0.69, 0.70]}.
```

The resulting explicit textual detections are fed into the LLM alongside the implicit visual features.

4 Explicit Visual Alignment

To enable the MLLM to reason over *explicit* information, we design a three-stage training pipeline termed *E-Align* (Figure 5):

Stage 1: Implicit pretraining. We train only the two-layer implicit adapter on image-caption data to align CLIP visual features with the LLM embedding space, while keeping both the LLM and the image encoder frozen.

Stage 2: Explicit pretraining. We treat the frozen image encoder together with the explicit adapter as a lightweight detector. In this stage, we train the explicit adapter with detection supervision on Object365 v1 for 28 epochs, followed by 150 epochs of fine-tuning on COCO, while keeping the image encoder frozen throughout. As a result, intermediate CLIP features can be mapped by the explicit adapter into structured detection outputs, covering the 80 COCO categories.

Stage 3: Visual instruction tuning. We jointly train the LLM, the implicit adapter, and the visual encoder on visual instruction data, while keeping the explicit adapter frozen. This stage teaches the model to integrate global semantic representations (implicit features) with structured object-level cues (explicit signals) for stronger visual reasoning.

After these three stages, the model can exploit both implicit features and explicit detection signals at inference time, achieving a better balance between global semantic understanding and fine-grained visual details.

5 Experiments

5.1 Experiment settings

Model setup. We adopt clip-vit-large-patch14-336 (Radford et al., 2021) and siglip2-so400m-patch16-384 (Tschannen et al., 2025) as vision encoders, and Llama-3-8B (Dubey et al., 2024) and Qwen3-4B-Instruct-2507 (Yang et al., 2025) as base LLMs. To verify that our method remains effective across diverse backbone configurations, we train three Explicit-VL variants from scratch: CLIP-Llama3, CLIP-Qwen3, and SigLIP2-Qwen3.

Baselines. For a fair comparison, we reproduce LLaVA-NeXT using the same visual instruction data for each of the three backbone combinations.

Implementation Details. All training data used in this work are publicly available. Details are as

Method	Encoder	LLM	MMMU	MMStar	MMB (EN)	MMB (CN)	ScienceQA	RealworldQA	CV-Bench	POPE
LLaVA-Next	CLIP	Llama 3	37.8	44.0	69.8	64.0	75.4	58.6	66.8	87.5
Explicit-VL	CLIP	Llama 3	**37.9**	**45.4**	**70.4**	**64.8**	**77.6**	**59.9**	**68.3**	**88.4**
LLaVA-Next	CLIP	Qwen 3	43.3	45.7	74.1	72.8	75.8	60.9	64.1	**87.4**
Explicit-VL	CLIP	Qwen 3	**44.2**	**46.9**	**77.0**	**74.8**	**76.3**	**61.1**	**68.3**	**87.4**
LLaVA-Next	Siglip2	Qwen 3	**45.1**	51.7	77.6	75.6	**78.4**	62.4	67.0	86.8
Explicit-VL	Siglip2	Qwen 3	44.6	**52.3**	**78.1**	**77.7**	77.3	**63.8**	**68.4**	**87.8**

Table 1: **Comparison between Explicit-VL and LLaVA-Next on MLLM benchmarks.** Under all three model configurations (CLIP-ViT–Llama-3, CLIP-ViT–Qwen3, and SigLip2-ViT–Qwen3), Explicit-VL achieves consistently higher performance than LLaVA-Next.

Method	Encoder	LLM	RefCOCO			RefCOCO+			RefCOCOg	
			val	testA	testB	val	testA	testB	val	test
LLaVA-Next	CLIP	Llama 3	70.9	74.7	66.8	64.8	71.4	58.0	64.1	65.3
Explicit-VL	CLIP	Llama 3	**73.4**	**76.3**	**67.8**	**66.7**	**73.0**	**58.6**	**65.8**	**66.9**
LLaVA-Next	CLIP	Qwen 3	68.2	71.0	64.6	62.4	68.6	56.5	62.9	63.6
Explicit-VL	CLIP	Qwen 3	**72.9**	**74.4**	**70.0**	**66.9**	**70.6**	**60.3**	**66.4**	**66.9**
LLaVA-Next	Siglip2	Qwen 3	68.2	67.5	67.1	63.9	65.6	60.5	60.6	61.5
Explicit-VL	Siglip2	Qwen 3	**72.9**	**72.8**	**71.5**	**67.7**	**69.2**	**64.0**	**65.1**	**66.3**

Table 2: **Visual grounding results.**

follows: (1) For implicit pre-training, we use the 558k LLaVA pre-training dataset (Liu et al., 2024b) and adopt the same training settings as LLaVA-Next. (2) For explicit training, we first pre-train on Objects365 v1 (Shao et al., 2019), which contains 600k images across 365 categories, and then fine-tune on COCO 2017 Train (Lin et al., 2014) with 118k images covering 80 categories. On Objects365 v1, we train for 28 epochs with a learning rate of 0.08, while on COCO 2017 Train, we fine-tune for 150 epochs with the same batch size but a learning rate of 0.001. To capture more visual details, during inference of explicit adapter we use a low proposal confidence threshold of 0.1 and set the NMS IoU threshold to 0.3. (3) For visual instruction tuning, we use the 779k LLaVA-Next instruction dataset (Li et al., 2024a). Note that the original LLaVA-Next dataset contains more than 790k samples, but 15k user-collected instructions were not publicly released.

5.2 MLLM Benchmarks Results

We evaluate Explicit-VL across four categories of benchmarks: (1) Knowledge-based VQA: MMMU (Yue et al., 2024), MMStar (Chen et al., 2024a), MMBench (Liu et al., 2024c), and ScienceQA (Lu et al., 2022); (2) Vision-centric tasks: CV-Bench (Tong et al., 2024a) and RealWorldQA (xAI, 2024); (3) Hallucination: POPE (Li et al., 2023b); and (4) Visual grounding tasks: RefCOCO (Kazemzadeh et al., 2014), RefCOCO+ (Yu et al., 2016), and RefCOCOg (Mao et al., 2016). Knowledge-based VQA, vision-centric and hallucination results are shown in Table 1, visual grounding results are shown in Table 2.

Finding 1: By providing explicit information, MLLMs can better capture fine-grained visual details. As shown in Table 1, Explicit-VL consistently outperforms LLaVA-Next on knowledge-based VQA (e.g., MMB-EN +2.9 under CLIP-Qwen3, ScienceQA +2.2 under CLIP-Llama3) and vision-centric benchmarks (CV-Bench +4.2 under CLIP-Qwen3), while the POPE score also achieves consistent gains. This validates the value of structured textual hints for fine-grained perception and reduces the generation of hallucinations.

Finding 2: Explicit information enhances the end-to-end detection capability of MLLMs. As shown in Table 2, across the three model configurations, Explicit-VL achieves consistent and substantial superiority over LLaVA-Next on RefCOCO, RefCOCO+, and RefCOCOg. The most prominent gains are observed in complex visual contexts, with an average absolute improvement of +4.0 on RefCOCO val, +3.4 on RefCOCO+ val, and +3.2 on RefCOCOg val. Notably, even though our explicit signals only provide bounding-box center coordinates (instead of full detection annotations), they

Model	$\lvert\theta_{\text{vis}}\rvert$	MMMU	MMStar	MMB (EN)	MMB (CN)	ScienceQA	CV-Bench	POPE	
LLaVA-v1.6-13B (Li et al., 2024a)†	304M	34.1	39.8	69.1	61.9	73.6	64.8	86.3	
LLaVA-Next-8B (Li et al., 2024a)†	304M	40.0	42.2	72.2	67.0	73.3	65.0	86.6	
Cambrian-8B (Tong et al., 2024a)*	2208M	42.7	50.0	75.9	67.9	80.4	72.2	87.4	
Eagle X5 8B (Shi et al., 2025a)	2282.8M	43.5	-	75.5	-	84.1		-	-
Florence-VL 8B (Chen et al., 2025)*	770M	43.7	50.0	76.2	69.5	85.9	73.4	**89.9**	
Explicit-VL 4B ***(Ours)***	400M+27M	**49.4**	**53.5**	**78.1**	**75.3**	**87.4**	**74.4**	88.0	

Table 3: **Comparison with open-source MLLMs.** By introducing only a 27M-parameter explicit adapter, Explicit-VL 4B outperforms mixture-of-encoders models Cambrian-8B and Eagle X5 8B on five of six benchmarks, and also surpasses Florence-VL, which uses a stronger vision encoder. $\lvert\theta_{\text{vis}}\rvert$ denotes the number of parameters in the visual encoder and adapter. Results marked with * are taken from (Chen et al., 2025); marked with † are reproduced by us.

still effectively enhance the model's ability to infer object locations accurately in images.

Finding 3: E-Align is highly generalizable and model-agnostic. From the results in Table 1, we observe consistent improvements when applying E-Align to CLIP–Llama3, CLIP–Qwen3, and SigLIP2–Qwen3, demonstrating strong transferability across LLMs and vision encoders. Importantly, although SigLIP2 can provide finer-grained visual features than CLIP, our method still gains +1.4 on RealWorldQA and +1.4 on CV-Bench, while Table 1 validates its necessity. This confirms our approach is orthogonal to implicit fine-grained representations, serving as an effective additional source of visual detail.

Finding 4: Explicit-VL outperforms strong open-source baselines with only a 27M-parameter explicit adapter added to the vision encoder. To probe the upper bound of E-Align, we adopt our strongest MLLM configuration, SigLIP2-Qwen3. We randomly sample 640K examples from Honey-Data-1M (Zhang et al., 2025) for visual instruction tuning. As shown in Table 3, among all open-source baselines, Explicit-VL 4B achieves the best performance on MMMU, MMStar, MMB, ScienceQA and CV-Bench. Notably, Explicit-VL surpasses the Mixture-of-Encoders models Cambrian-8B and Eagle X5 8B while adding only a 27M-parameter adapter to enhance fine-grained understanding, whereas Cambrian-8B and Eagle X5 8B rely on vision encoders that aggregate over 2,000M parameters. Explicit-VL also substantially outperforms Florence-VL on knowledge-based VQA benchmarks and CV-Bench. Despite Florence-VL leveraging the strong multi-task–trained Florence-2 vision encoder (770M parameters), we use only a SigLIP2 ViT (400M) paired with a lightweight 27M-parameter adapter. These results indicate that E-Align enables efficient visual alignment and improves fine-grained understanding with minimal additional parameters.

Method	MMMU	MMStar	MMB (EN)	MMB (CN)
w/o explicit train & infer	37.8	44.0	69.8	64.0
w/o explicit train	37.2	44.0	**70.5**	63.3
w/o explicit infer	37.7	44.9	**70.5**	64.1
Explicit-VL	**37.9**	**45.4**	70.4	**64.8**

Table 4: **Ablation study on introducing explicit information during training and inference.** The results show that explicit information are essential in both stages. However, introducing them only during training improves the MLLM's general visual understanding, whereas introducing them only at inference leads to degraded performance, as the model has not learned to effectively utilize explicit information.

5.3 Ablation Study

To study the effect of introducing explicit information during training and inference, we conduct ablations. As shown in Table 4, we evaluate: (1) *w/o explicit train & infer*: removing explicit training and explicit inference; (2) *w/o explicit train*: removing explicit information in visual instruction tuning but enabling explicit inference; (3) *w/o explicit infer*: keeping explicit information during instruction tuning but removing explicit inference; (4) *Explicit-VL (ours)*: full pipeline.

Introducing explicit information in both training and inference is essential for improving the MLLM's fine-grained visual understanding. With explicit instruction tuning and explicit inference enabled, the model achieves the best performance on three of four benchmarks and delivers the second-best results on the remaining one. Moreover, **explicit training alone already brings clear gains** over the fully implicit setting on MMStar, MMB(EN) and MMB(CN), suggesting that explicit signals do not merely serve as test-time hints but also improve vision-language alignment during training. In contrast, *w/o explicit train* yields limited benefits and and causes drops on MMMU and

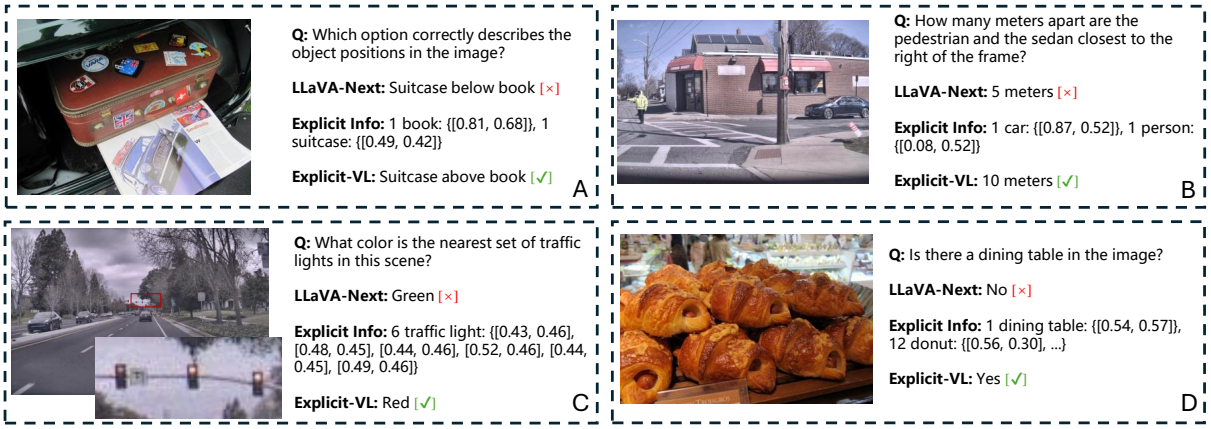

Figure 6: **Qualitative comparison between LLaVA-NeXT and Explicit-VL.** The visualization illustrates three key mechanisms by which Explicit Info enhances perception: (1) **Indicating object locations** (case A and B), where explicit coordinates facilitate spatial reasoning and distance estimation; (2) **Reorganizing model attention** (case C), where the model is guided to focus on fine-grained details such as distant traffic lights; and (3) **Indicating object categories** (case D), which effectively mitigates object hallucinations.

Strategy	Tiles layers	Thumbnail layers	mAP
2base2sub	L6, L10	L14, L23	31.0
3base1sub	L6	L10, L14, L23	29.5
2base2sub-alldeep	L23 ×2	L23 ×2	27.6
2base2sub-allmid	L14 ×2	L14 ×2	29.4
2base2sub-mid&deep	L14, L23	L14, L23	**32.3**

Table 5: **Feature-selection strategies comparison.**

MMB(CN), indicating that without explicit training, the model cannot effectively leverage explicit information, and it can even be detrimental.

5.4 Study on ViT Feature Selection

We find that the detection performance of the *explicit adapter* is sensitive to which ViT layers are used for feature extraction. Unlike standard ViT-based detectors that select four feature maps from the 24 layers of a single ViT, AnyRes provides up to 48 candidates (24 from *Thumbnail* and 24 from *Tiles*). We evaluate five feature-selection strategies by training on COCO 2017 for 40 epochs and reporting the best mAP. As shown in Table 5, `2base2sub-mid&deep` performs best, indicating that combining mid-level and deep features from both the Tiles and Thumbnail branches yields the most effective representations for detection.

5.5 Qualitative Results

In Figure 6, we present qualitative results. Overall, Explicit Info improves MLLMs' perception mainly in three ways: (1) **Indicating object locations.** In example B, Explicit Info provides the coordinates of the *car* and the *person*. With these explicit locations, Explicit-VL can more reliably infer their spatial distance and correctly answer "10M". (2) **Reorganizing model attention.** In example C, the relevant traffic lights are extremely small and appear near the horizon line. LLaVA-NeXT therefore overlooks them. After adding Explicit Info that lists the traffic-light locations, Explicit-VL can notice these specific regions and correctly identify the signal as red. (3) **Indicating object categories.** In example D, the image contains many donuts, which distracts LLaVA-NeXT and causes it to overlook the presence of the *dining table*. By explicitly providing the category "dining table", Explicit Info reduces object hallucination and helps the model answer correctly.

6 Conclusion

We introduce E-Align, a new paradigm that aligns CLIP features to a discrete textual space, enabling MLLMs to better leverage the fine-grained signals already captured but not fully exploited by language-supervised encoders. Building on this paradigm, we develop Explicit-VL, which uses a lightweight explicit adapter on the vision encoder to provide object-level explicit information directly to the LLM. Without additional visual experts or extra encoders. Extensive experiments show that E-Align yields consistent gains across knowledge-based QA, vision-centric tasks, hallucination benchmarks, and object detection. Moreover, our Explicit-VL outperforms strong mixture-of-encoders baselines with only a 27M-parameter explicit adapter, underscoring the effectiveness and scalability of E-Align.

Limitations

Although the detection results generated by our explicit-adapter already provide significant improvements for the MLLM, there remains room for optimization. First, the explicit information is currently restricted to the 80 object categories defined by the COCO dataset. Given the open-ended knowledge capabilities of Multimodal Large Language Models, this vocabulary is relatively narrow. Expanding the taxonomy to include a broader range of object categories would better facilitate the MLLM's understanding of complex images. Second, the current explicit information is limited to object detection. Incorporating more fine-grained information, such as segmentation masks, or structured information extraction tailored for chart-oriented tasks, could further enhance model performance.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, and 4 others. 2025. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *Preprint*, arXiv:2509.23661.

Mozhgan Nasr Azadani, James Riddell, Sean Sedwards, and Krzysztof Czarnecki. 2025. Leo: Boosting mixture of vision encoders for multimodal large language models. *Preprint*, arXiv:2501.06986.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. 2025. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv:2504.13181*.

Jiuhai Chen, Jianwei Yang, Haiping Wu, Dianqi Li, Jianfeng Gao, Tianyi Zhou, and Bin Xiao. 2025. Florence-vl: Enhancing vision-language models with generative vision encoder and depth-breadth fusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24928–24938.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024a. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2023. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, and 1 others. 2025. Scaling language-free visual representation learning. *arXiv preprint arXiv:2504.01017*.

Qirui Jiao, Daoyuan Chen, Yilun Huang, Yaliang Li, and Ying Shen. 2024. From training-free to adaptive: Empirical insights into mllms' understanding of detection information. *arXiv preprint arXiv:2401.17981*.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*.

Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. 2024a. Llava-next: Stronger llms supercharge multimodal capabilities in the wild.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024b. Llava-onevision: Easy visual task transfer. *Preprint*, arXiv:2408.03326.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024c. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Jincheng Li, Chunyu Xie, Ji Ao, Dawei Leng, and Yuhui Yin. 2025. Lmm-det: Make large multimodal models excel in object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 308–318.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Preprint*, arXiv:2301.12597.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore. Association for Computational Linguistics.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

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

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. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024c. Mmbench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *CVPR*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, and 1 others. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

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

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. 2019. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439.

Leyang Shen, Gongwei Chen, Rui Shao, Weili Guan, and Liqiang Nie. 2024. Mome: Mixture of multimodal experts for generalist multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 42048–42070. Curran Associates, Inc.

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, Bryan Catanzaro, Andrew Tao, Jan Kautz, Guilin Liu, and Zhiding Yu. 2025a. Eagle: Exploring the design space for multimodal LLMs with mixture of encoders. In *International Conference on Learning Representations (ICLR)*.

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, Yilin Zhao, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, Bryan Catanzaro, Andrew Tao, Jan Kautz, Zhiding Yu, and Guilin Liu. 2025b. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *Preprint*, arXiv:2408.15998.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, and 7 others. 2025. Dinov3. *Preprint*, arXiv:2508.10104.

Matvey Skripkin, Elizaveta Goncharova, Dmitrii Tarasov, and Andrey Kuznetsov. 2025. Move: A mixture-of-vision-encoders approach for domain-focused vision-language processing. *Preprint*, arXiv:2502.15381.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, Xichen Pan, Rob Fergus, Yann LeCun, and Saining Xie. 2024a. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *Advances in Neural Information Processing Systems*, volume 37, pages 87310–87356. Curran Associates, Inc.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024b. Eyes wide

shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *Preprint*, arXiv:2502.14786.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *Preprint*, arXiv:2508.18265.

xAI. 2024. Realworldqa: A benchmark for real-world spatial understanding. `https://huggingface.co/datasets/xai-org/RealworldQA`. Accessed: 2025-04-26.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *CVPR 2024*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Lu Sheng, Lei Bai, Xiaoshui Huang, Zhiyong Wang, and 1 others. 2023. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *arXiv preprint arXiv:2306.06687*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. *ArXiv*, abs/1608.00272.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Yi Zhang, Bolin Ni, Xin-Sheng Chen, Heng-Rui Zhang, Yongming Rao, Houwen Peng, Qinglin Lu, Han Hu, Meng-Hao Guo, and Shi-Min Hu. 2025. Bee: A high-quality corpus and full-stack suite to unlock advanced fully open mllms. *arXiv preprint arXiv:2510.13795*.

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

Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. 2024. Mova: Adapting mixture of vision experts to multimodal context. In *Advances in Neural Information Processing Systems*, volume 37, pages 103305–103333. Curran Associates, Inc.