# Enhancing Visual Perspective-Taking in Vision-Language Models through Abstract Scene Transformation

**Karol Jurasz**
kj448218@students.mimuw.edu.pl

**Jakub Kołaczyński**
jk438520@students.mimuw.edu.pl

**Michał Piasecki**
mm.piasecki2@student.uw.edu.pl

**Adam Rempała**
ar406309@students.mimuw.edu.pl

**Jakub Rymarski**
jr472844@students.mimuw.edu.pl

**Supervisor: Gracjan Góral**
gp.goral@uw.edu.pl

## Abstract

We present the results of an academic project focused on enhancing visual perspective-taking abilities in Vision-Language Models (VLMs). We replicate the APC-VLM (Attribute-Partitioned Concept Vision-Language Model) framework, originally designed to simulate human-like mental transformations of scenes via abstract 3D representations. Our implementation reproduces the original pipeline, including object detection, segmentation, depth estimation, coordinate transformation and numerical prompts. We evaluate proposed method on our own specially prepared dataset, designed for assessing allocentric reasoning. This paper summarizes the full implementation process, experimental setup, observed limitations and resulting insights into VLM capabilities in spatial reasoning. Our project can be found on GitHub [1].

## 1 Introduction

Vision-language models (VLM) are advanced large-language models (LLM) that learn on a combination of images and text prompts. They have advanced rapidly and demonstrate remarkable capabilities in the integration of visual and linguistic understanding for tasks such as visual question answering (VQA), image captioning, and cross-modal retrieval (Goyal et al., 2016). For example, having an image presenting something (a portrait, a sign, a board with texts and drawings or just a scene from the street), we can pose some questions about it, expecting a clear and correct answer (does the person have a hat, do I have a red light, how many people are on the photo etc.). Despite leveraging the LLM backbones for robust text reasoning, VLMs exhibit critical limitations in scene perspective reasoning, often failing to adapt to viewpoints beyond the camera's egocentric frame. This shortcoming restricts their applicability in tasks that require spatial and allocentric reasoning. In other words, improvements are needed if we want to analyze the scene on the photo from the perspective different than the camera's.

To address this challenge, we focused on replicating and extending the Abstract Perspective Change (APC) framework (Lee et al., 2025). This method emulates human-like mental transformation of scenes by abstracting them into 3D representations and re-rendering them from alternative viewpoints. Our goal consists of two main parts. Firstly, we replicate and verify the APC pipeline. Then we explore how modifying scene abstraction and prompt representation could enhance allocentric spatial reasoning. Moreover we prepared the dataset inspired by the Isle benchmark (Góral et al., 2024), designed for visual perspective task. Our results confirm the utility of the APC framework and show that improvements increase model performance on allocentric reasoning tasks.

## 2 Related Work

Contrastive dual-encoder models like CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) laid the foundation for joint vision-language embedding. Multi-modal transform-

[1] https://github.com/michalpiasecki0/visual-perspective-taking-project

ers like ViLBERT further improved image-text interaction through cross-attention and multi-task objectives.

However, these architectures underperform in spatial reasoning. Studies show that current state-of-the-art models manage achieve good results almost exclusively on egocentric perspective tasks and show significant performance drops on allocentric perspective questions (Góral et al., 2024; Stogiannidis et al., 2025). To address this, SpatialPIN (Ma et al., 2024) employs zero-shot prompting with 3D priors. These methods highlight the importance of explicit geometric information but add complexity.

The APC framework (Lee et al., 2025) offers a modular, training-free approach by transforming any scene representations and questions into egocentics ones and utilizing existing VLM capabilities, demonstrating state-of-the-art gains on spatial reasoning benchmarks without modifying the base model.

## 3 Experimental Setup

We implemented a complete and modular version of the APC-VLM pipeline. Our implementation is composed of three main stages: scene abstraction, perspective transformation, and prompt generation. Each module integrates state-of-the-art vision models to process 2D images into structured 3D representations, enabling allocentric visual perspective-taking. The pipeline was evaluated on our dataset using vision-language model Qwen-VL (Bai et al., 2023).

### 3.1 Dataset

We constructed a labeled dataset inspired by the Isle benchmark (Góral et al., 2024). Our dataset contains scenes featuring three distinct human subjects in various spatial arrangements and poses (e.g., standing, sitting, facing toward or away from the camera).

The dataset consists of 55 photos combined from three subsets: 22 images of a person with additional nearby objects and cat, 17 images of a person with a watermelon and other items positioned slightly farther away, 16 images of a person in a sleeping room with a ball and plant. First two subsets contain adversarial examples, such as objects hidden (from the perspective of a person) behind small obstacles or objects on the verge or barely outside the verge of person's sight. The third subset is possibly the simplest to analyse as there are few objects at the center of images. The data is labeled with questions constructed in a *"Can the person see a... "* manner as well as true binary answers to these questions. The data is well-balanced - 27 out of 55 examples are positive.

All scenes are captured under consistent and favorable lighting conditions and contain a rich variety of spatial relationships. The diversity of poses and object arrangements ensures that the dataset is suitable for evaluating models' capability for generalizable visual perspective taking through reasoning over coordinate and orientation data alone—without requiring visual input.

### 3.2 Scene Abstraction

The scene abstraction stage extracts spatial and geometric information from raw RGB images, producing structured representations for reasoning.

**Object Detection:** we use GroundingDINO (Liu et al., 2024), a transformer-based detector capable of text-conditioned object localization. GroundingDINO takes a list of instances to detect and returns bounding boxes of these instances in an image (possibly multiple).

**Segmentation:** following detection, each object is segmented using the Segment Anything Model (SAM) (Kirillov et al., 2023). SAM provides high-quality instance masks, which are essential for accurate spatial analysis and 3D abstraction.

**Depth Estimation:** to infer depth information from monocular images, we employ Depth Anything Pro (Bochkovskii et al., 2025), a model trained on diverse datasets for dense depth prediction. This step is crucial for estimating relative distances between objects and their positions in the scene.

**Orientation Estimation:** we also extract object orientation using OrientAnything (Wang et al., 2024), which estimates pose information for each object relative to the camera viewpoint.

The outputs of bounding boxes, segmentation masks, depth maps, and orientation vectors are fused into a 3D scene representation in a unified allocentric coordinate system.

Our pipeline successfully returns bounding boxes, instance masks, depth estimation, and objects' orientation of good accuracy. However, we have encountered multiple problems with the proper implementation of the OrientAnything model as well as the proper interpretation of its output, as the authors do not provide documentation on its format. This introduces a potentially negative impact on the quality of the final model as we have empirically decided on one specific interpretation of the output that seemed to fit the reality the most without having any external confirmation about its correctness.

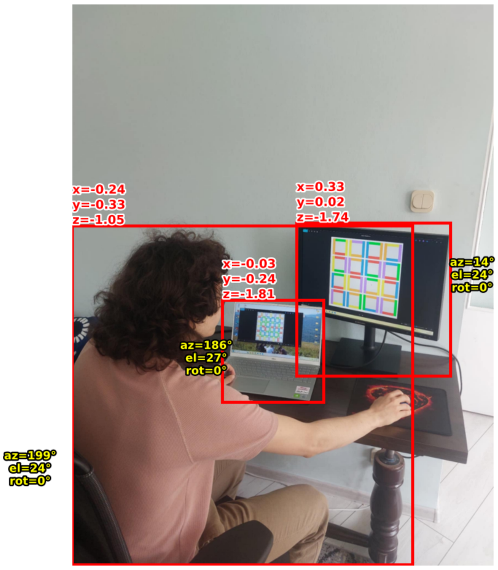


Figure 1: Scene abstraction

### 3.3 Perspective Transformation

To simulate allocentric perspective-taking, we implemented a deterministic geometric transformation module. Each object's location is projected from the original image coordinate frame to a new viewpoint-defined frame.

### 3.4 Numerical Prompts:

We generated structured textual prompts to convey egocentric spatial relations using a standardized format. The prompts used for specific experiments are presented in Section 4.

### 3.5 Visual representation

In original APC solution numerical prompt was enriched with visual prompt at prompt generation stage. The visual prompt should represent the simplified version of a scene, using colored cubes instead of actual objects, which could allow the model to focus on most important information.

We implemented this using PyVista visualisation engine and coordinates calculated for the numerical prompt. However, due to some inconsistencies and problems with defining proper connections between OrientAnything and scene creation tools we decided against using it and focused on solution using only numerical prompts.

### 3.6 Implementation Details

Our experiments were conducted on a GPU-enabled environment. All submodules were executed in a sequential pipeline. To reduce computational load and improve efficiency, we refactored the pipeline to eliminate redundant calls to vision modules.

## 4 Results

We evaluated the effectiveness of egocentric perspective transformation in better understand spatial relationships from the viewpoint of person in the scene. We tested three types of prompting strategies:

### 4.1 Instructing the model to answer with only yes/no

In this first variant, the model was instructed to answer strictly *"yes"* or *"no"*, following prompts similar to one illustrated below:

**Example Prompt**

*Imagine that you are at the person's position and facing where it is facing.*
*We have the coordinates of different objects in person's coordinate system.*
*# Coordinate System*
*- The origin is at the person's position.*
*- The person is located at the origin, looking in the direction of the negative Z-axis: [0, 0, -1].*
*- The coordinate system is right-handed. The X-axis points to the right ([1, 0,*

*0]), the Y-axis points up ([0, 1, 0]), and the Z-axis points out of the camera's back ([0, 0, 1]).*
*- Therefore, objects in front of the camera are located along negative Z values in camera space.*
*# Object Coordinates*
*{'cat': [0.3, -0.16, -0.51]}*
*# Task*
*Given the above person's coordinate system and the object coordinates, please answer the following question:*
*Can the person see a cat?*
*Answer in one word: yes / no*

For 54 of 55 questions, the answer was *"yes"*. 27 answers were correct, which gives 0.51 accuracy. The results were unsatisfactory and may imply that the final language model may not be able to comprehend all information given in just single yes/no answer and we may need to allow it to perform some *"thinking process"*.

### 4.2 Model answer without any additional instructions

The natural continuation of experiments was to allow the model to answer without any specific format rules, which may result in gaining deeper insight into its process of reasoning. This resulted in deleting the yes/no restriction from the prompt.

**Example Prompt**

*Imagine that you are at the person's position and facing where it is facing.*
*We have the coordinates of different objects in person's coordinate system.*
*# Coordinate System*
*- The origin is at the person's position.*
*- The person is located at the origin, looking in the direction of the negative Z-axis: [0, 0, -1].*
*The coordinate system is right-handed.*
*The X-axis points to the right ([1, 0, 0]), the Y-axis points up ([0, 1, 0]), and the Z-axis points out of the camera's back ([0, 0, 1]).*
*Therefore, objects in front of the camera are located along negative Z values in camera space.*
*# Object Coordinates*
*{'cat': [0.3, -0.16, -0.51]}*
*# Task*
*Given the above person's coordinate system and the object coordinates, please answer the following question:*
*Can the person see a cat?*

The model returned long text answers that were manually evaluated as correct, incorrect or not answering the question. The model achieved 0.7 accuracy score on examples with outputs answering the questions, which is significantly better than the score expected for the random model. It did not provide answers for 5 questions. For 2 of them the reason was the lack of vectors symbolizing positions of objects important for the question as pipeline did not manage to succesfully detect them, whereas answers for the remaining 3 questions exceeded the 256 tokens limit due to the too long step-by-step thinking processes. As expected, the model achieved the greatest accuracy on the simplest subset of dataset, classifying correctly all examples except one, whereas for remaining data containing adversarial examples the accuracy was close to 0.5.

However, we found that its primary approach to determining visibility relied almost exclusively on the Z-coordinate (indicating whether an object was in front of or behind the observer). If the Z-coordinate was negative, the model typically concluded that the object was visible. While this reasoning was often correct for objects directly in front, it proved insufficient when the object of interest was located more than approximately 120 degrees inconsistent with the person's orientation.

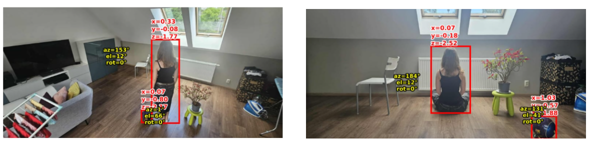

Figure 2: The simplest subset examples. On this balanced subset the model achieved accuracy of 0.7

In many, but not all, cases, the model consistently disregarded the X and Y coordinates. This fundamental oversight revealed its lack

Figure 3: Adversarial examples. The model operating on checking the sigh of Z-axis coordinate didn't manage to include information about obstacles between a person and an object nor about field of view.

of understanding regarding a person's limited field of view – a crucial aspect, as an object directly ahead in terms of Z-coordinate may still be invisible if positioned too far to the side. The model seemed to operate under the assumption that as long as an object wasn't behind the observer, it was visible, regardless of its lateral displacement. The model was achieving great scores on simple examples, however it struggled an almost any adversarial examples.

Occasionally, the model attempted to incorporate X and Y coordinates into its reasoning. However, such attempts often resulted in responses that became excessively lengthy and confusing, frequently being truncated mid-sentence due to character limits. Alternatively, after a convoluted discussion, the model would often revert to focusing solely on the Z-coordinate.

In essence, granting the model this unconstrained freedom exposed the rudimentary nature of its spatial understanding. While it could discern what was in front or behind, it demonstrably struggled with a comprehensive 3D spatial picture and lacked an awareness of realistic visual limitations.

**Correct reasoning example**

*however, the cat's position is not exactly on the negative z-axis; it has some x and y components as well.*

**Wrong reasoning example**

*since the person is looking in the negative z-axis direction, any object with a negative z-coordinate will be in front of the person.*

### 4.3 Prompt enriched with field of view of a person

In our third experimental setup, we augmented the prompts by explicitly informing the model that the person possessed a 120-degree field of view (FOV). Our goal was to see if this additional piece of information would enable the model to move beyond its simplistic Z-coordinate-based assessments and develop a more nuanced understanding of spatial visibility, incorporating X and Y coordinates meaningfully.

**Example Prompt**

*Imagine that you are at the person's position and facing where it is facing.*
*We have the coordinates of different objects in person's coordinate system.*
*# Coordinate System*
*- The origin is at the person's position.*
*- The person is located at the origin, looking in the direction of the negative Z-axis: [0, 0, -1].*
*The coordinate system is right-handed. The X-axis points to the right ([1, 0, 0]), the Y-axis points up ([0, 1, 0]), and the Z-axis points out of the camera's back ([0, 0, 1]).*
*Therefore, objects in front of the camera are located along negative Z values in camera space.*
*Person field of view is 120 degrees.*
*# Object Coordinates*
*{'cat': [0.3, -0.16, -0.51]}*
*# Task*
*Given the above person's coordinate system and the object coordinates, please answer the following question:*
*Can the person see a cat?*

The inclusion of FOV indeed prompted a noticeable shift in the model's behavior. Instead of merely checking the Z-coordinate, the model began to attempt complex geometric and trigonometric calculations. Its answers frequently detailed *”step-by-step analyses”* involving concepts like *”direction vectors”*, *”dot products”*, and *”angles”* trying to determine if the object's position fell within the specified 120-degree cone. This indicated that the model was now trying to apply a more sophisticated, physics-like understanding of visibility.

However, this ambition often outstripped our computation capabilities. The token generation limit was intentionally set to 256 tokens in order to preserve acceptable time of inference process given our computing power limitations. While the model correctly identified the FOV angle and the need to calculate angles between vectors, it consistently struggled to complete these calculations within the limit. Many of its detailed explanations were truncated mid-sentence, hitting character limits before a definitive conclusion could be reached. The model did not manage to provide a complete answer to the question in 53 out of 55 examples, which indicates its poor ability to comprehend its reasoning capabilities in a short sentence. However, with the addition of computing power one could possibly obtain better results.

Example of its new, but often incomplete, reasoning:

**Example Answer**

*to determine if the person can see the book, we need to check if the book's position lies within the person's field of view (fov). given the person's orientation and the fov, we can calculate the bounding box in the person's coordinate system.*

*### step-by-step calculation:*

*1. **person's orientation**:*
*- the person is at the origin* $[0,0,0]$.
*- the person is looking in the direction of the negative z-axis, so the camera's forward vector is* $[-1,0,0]$.

*2. **field of view (fov)**:*
*- the fov is 120 degrees, which means the horizontal and vertical angles are each 60 degrees (since* $120°/2 = 60°$*).*

*3. **bounding box calculation**:*
*- the bounding box will be a frustum with the near plane at some distance (let's assume it's 1 unit for simplicity), and the far plane at infinity.*
*- the width and height of the bounding box in the person's coordinate system can be calculated using trigonometry.*

*for a 60-degree fov:*
*- the*

Figure 4: Photo for question: Can a person see a book?

Here, the model attempts to define a *”bounding box”* or ”frustum” based on the 120-degree FOV, even trying to use *”trigonometry”*. While conceptually sound, this intricate explanation is also cut short.

### 4.4 Comparsion and summarization

The evaluation of the three prompting strategies revealed significant differences in how the language model interprets and reasons about spatial relationships from an egocentric viewpoint. Each subsequent strategy introduced more contextual detail or cognitive challenge. All experiments exposed consistent limitations in the model's fundamental spatial reasoning.

In the first strategy, where the model was constrained to provide only *"yes"* or *"no"* answers, the results were highly one-dimensional. The model responded *"yes"* to 54 of the 55 questions, regardless of the actual visibility of the objects within the scene. This uniformity strongly suggested that the model had adopted a flawed and overly simplistic heuristic: that the mere presence of objects in the input image implied mutual visibility. This strategy, while easy to analyze, offered no insight into the model's reasoning process and highlighted a complete lack of spatial discrimination. The inability to factor in object orientation, position, or occlusion demonstrated that without elaborative prompts, the model does not inherently grasp the notion of visual perspective or limitations.

In the second strategy, where the model was free to respond in any way without restrictions on format or style, we hoped to uncover its natural approach to spatial judgment. The results here were slightly more informative but still deeply problematic. The model predominantly focused on the Z-coordinate, equating negative Z-values (objects in front of the observer) with visibility. While this approach can sometimes be correct for basic configurations, it ignores crucial components such as horizontal displacement and the bounded nature of human vision. The X and Y coordinates, which determine an object's lateral and vertical placement, were frequently disregarded or misunderstood. Occasionally, the model attempted to use all three coordinates to reason about visibility, but such responses quickly became incoherent, verbose, or truncated due to output length constraints. When trying to articulate more complex ideas, the model often got lost in inconsistent logic or reverted to oversimplified heuristics.

The third and most elaborate strategy incorporated an explicit field-of-view (FOV) constraint into the prompt, stating that the person had a 120-degree visual cone. This additional information changed the nature of the model's reasoning significantly. In many cases, it began attempting geometric and trigonometric analyses, invoking concepts such as direction vectors, dot products, angular measurements, and even tangent functions. These responses signaled a promising shift in how the model conceptualized visibility—not merely as *"in front"* or *behind"* but as a question of angular inclusion within a bounded cone of sight. Unfortunately, despite this progress in theoretical approach, the model routinely failed to complete its calculations or deliver conclusive answers. Responses were often cut off before the final angle was computed, or they contained errors in vector math and trigonometry. The result was that even though the model appeared to grasp the correct reasoning structure, it lacked the precision and consistency to apply it successfully.

It is also worth noting that strategy used for second and third experiment required manual labelling of model's outputs, which is expensive for bigger datasets. For the sake of simplicity and avoiding the risk of losing even more prediction accuracy due to inproper interpretation of text output by another language model we concluded that manual labeling was necessary, however this should be improved in the future.

Moreover, it is worth noting that predicting the visibility of an object based solely on the sign of its Z-axis coordinate yields correct results in most real-world scenarios. This suggests a certain predictive potential of such a simple model. However, our goal is to leverage the abstract reasoning capabilities of language models, as deterministic methods are insufficient for addressing more complex, non-binary questions. The language model is therefore expected to go beyond simple *if*-condition logic and demonstrate higher-level reasoning. Nonetheless, similar results could potentially be achieved through deterministic label prediction, which may offer a more cost-effective alternative.

In summary, each strategy led to increasingly complex and ambitious attempts at spatial reasoning, but none achieved the level of reliable, accurate interpretation required for consistent success. The first strategy revealed the model's default assumptions; the second revealed its heuristics and partial understanding; and the third revealed its conceptual potential but also its practical limitations. Across all variants, the model demonstrated a fundamental lack of embodied spatial awareness, especially regarding the constraints of human vision and orientation in three-dimensional space.

## 5 Conclusion

This study assessed whether a large language model could accurately interpret spatial relationships from an egocentric perspective by analyzing object visibility within a scene. We examined three progressively complex prompting strategies to evaluate how additional instruction or contextual detail might enhance the model's reasoning capabilities.

Our findings indicate that the model does not possess an inherent understanding of visual perspective or spatial awareness. When restricted to binary yes/no responses, the model exhibited a complete lack of spatial judgment, defaulting to universal affirmation regardless of object placement. When allowed to answer freely, the model demonstrated an overreliance on object depth (Z-coordinate) while neglecting lateral displacement and angular position. The most promising results emerged when we explicitly provided a 120-degree field-of-view constraint. Here, the model began to employ more sophisticated mathematical reasoning, including vector direction analysis and trigonometric concepts. However, the execution of these ideas was typically incomplete, inaccurate, or disrupted by output limitations.

In essence, the language model shows an ability to mimic geometric reasoning to some extent when explicitly prompted, but it lacks the internal spatial representation necessary to perform these tasks reliably. It treats spatial concepts as linguistic or symbolic abstractions rather than grounded, embodied experiences. The model does not seem to possess a working understanding of the constraints imposed by perspective, orientation, or field of view, all of which are crucial for accurate egocentric reasoning.

These findings suggest that current moderate size models are not yet suitable for tasks requiring deep spatial comprehension, especially when that comprehension must be grounded in realistic human perception.

## 6 Future Work

Future work could focus on:

**Improving scene abstraction pipeline:** in order to improve model's performance it may be beneficial to train a sequential model designated especially for proper estimation of object's position. It could allow to achieve better results as well as to obtain more efficiency due to the lack of need of using multiple modules of different models. It could also allow to omit the problems with interpretation of badly-documented models like OrientAnything.

**Labeling text output:** Current experiments required manual labeling of text data into binary classes. Implementation of efficient interpreter could allow scaling of model.

**Adding more computing power:** With the addition of more computing power one could set the token generation limit to more than 256 tokens. This could lead the model to generate complete reasoning, leading to answers that include geometric information about field of view.

**Geometric and trigonometric reasoning:** Our experiments highlighted the model's struggle with explicit spatial calculations, despite prompting it with accurate conceptual frameworks. Future work could focus on enhancing the model's ability to perform reliable geometric computations—such as vector arithmetic, angle estimation, and dot product evaluation. One direction may involve integrating external computational modules capable of handling such math reliably, or fine-tuning models on reasoning tasks grounded in 3D geometry. Accurate trigonometric reasoning—e.g., checking if an object falls within a given angular field-of-view using trigonometric functions could substantially improve performance on perspective tasks, provided the model can execute or invoke such calculations consistently.

**Real-world generalization:** while designed scenes offer controlled setups for evaluating spatial reasoning, real-world images present greater visual complexity, occlusion, and noise. Extending the pipeline to handle diverse real world scenarios such as street would make the system more applicable in practical settings.

## References


Yash Goyal et al. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. arXiv:1612.00837.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. ICML.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. ICML.

Gracjan Góral, Alicja Ziarko, Michał Nauman, Maciej Wołczyk. 2024. Seeing Through Their Eyes: Evaluating Visual Perspective Taking in Vision-Language Models. arXiv:2409.12969

Iason Stogiannidis, Steven McDonagh, Sotirios A. Tsaftaris. 2025. Mind the Gap: Benchmarking Spatial Reasoning in Vision-Language Models. arXiv:2503.19707.

Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. 2024. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. NeurIPS.

Phillip Y. Lee, Jihyeon Je, Chanho Park, Mikaela Angelina Uy, Leonidas Guibas, Minhyuk Sung. 2025. Perspective-Aware Reasoning in Vision-Language Models via Mental Imagery Simulation. arXiv:2504.17207.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. 2024. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. ECCV.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick. 2023. Segment Anything. ICCV.

Aleksei Bochkovskii, Amael Delaunoy, Hugo Germain, ¨ Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. 2025. Depth pro: Sharp monocular metric depth in less than a second. ICLR.

Zehan Wang, Ziang Zhang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao.2024. Orient anything: Learning robust object orientation estimation from rendering 3d models. arXiv:2412.18605

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, Junyang Lin. 2025. Qwen2.5-vl technical report. arXiv:2502.13923.