

# Orient Anything: Learning Robust Object Orientation Estimation from Rendering 3D Models

Zehan Wang<sup>1\*</sup>, Ziang Zhang<sup>1\*</sup>, Tianyu Pang<sup>2</sup>, Chao Du<sup>2</sup>, Hengshuang Zhao<sup>3</sup>, Zhou Zhao<sup>1</sup>

<sup>1</sup>Zhejiang University; <sup>2</sup>Sea AI Lab; <sup>3</sup>The University of Hong Kong

<https://orient-anything.github.io/>

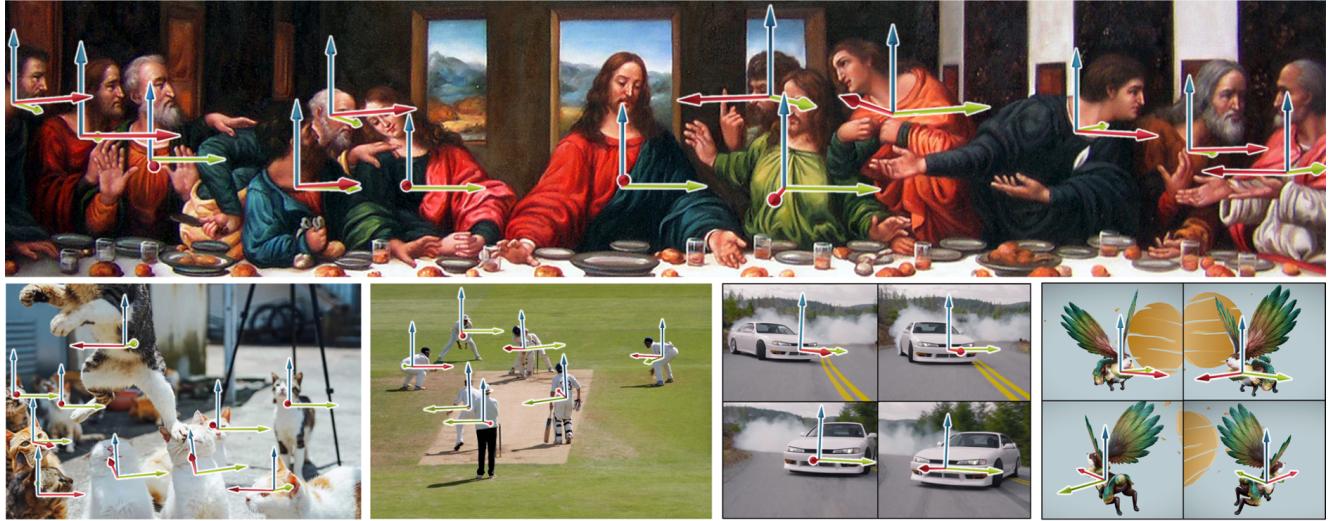


Figure 1. We introduce a novel method for estimating the object orientation in images, represented by the red axis, while the blue and green axes indicate the upward and left sides of the object. Examples are provided in Appendix. viewed on screen with zoom.

## Abstract

Orientation is a key attribute of objects, crucial for understanding their spatial pose and arrangement in images. However, practical solutions for accurate orientation estimation from a single image remain underexplored. In this work, we introduce **Orient Anything**, the first expert and foundational model designed to estimate object orientation in a single- and free-view image. Due to the scarcity of labeled data, we propose extracting knowledge from the 3D world. By developing a pipeline to annotate the front face of 3D objects and render images from random views, we collect 2M images with precise orientation annotations. To fully leverage the dataset, we design a robust training objective that models the 3D orientation as probability distributions of three angles and predicts the object orientation by fitting these distributions. Besides, we employ several strategies to improve synthetic-to-real transfer. The model achieves state-of-the-art orientation estimation accuracy in

both rendered and real images and exhibits impressive zero-shot ability in various scenarios (Fig. 1). Importantly, our model enhances many applications, such as comprehension and generation of complex spatial concepts and 3D object pose adjustment.

## 1 Introduction

Given object properties in a single image is the core problem in computer vision. Recent visual foundation models and large vision-language models (VLMs) excel in tasks like object recognition [24, 49], localization [22, 25], tracking [33, 43], and segmentation [20, 34].

However, the object orientation, which is critical for understanding object pose and arrangement, has been underexplored due to the lack of annotated data. Hi3D [6] enables 3D orientation prediction by unifying 3D object detection data, but its scope is still restricted to specific domains, primarily room and street scenes, making it difficult to generalize to diverse real-world scenarios.

\*Equal Contribution.

Furthermore, even the most advanced general visual understanding systems, like GPT-4o [18] and Gemini [41, 42], struggle to comprehend basic object orientation. As a result, they perform poorly on questions derived from orientation, such as imagining object movement trends, or understanding object spatial relationships, as shown in Fig. 2.

In this paper, we propose to learn how various objects look under different orientations by rendering 3D models. By annotating the front face of these 3D objects, we can easily and cheaply obtain precise orientation labels for each rendered view. This idea provides scalable, diverse, and easy-to-acquired data, enabling the development of accurate and generalizable orientation estimation models.

At the end, we develop a data collection pipeline to automatically filter, annotate, and render 3D assets [9], enabling scalable data generation at any desired scale. In particular, we leverage advanced VLM [42] to identify the front side of 3D objects from orthographic views, complemented by canonical pose detection and symmetry analysis to simplify the task and improve accuracy. Then, we render images from random perspectives, using azimuth and polar angles relative to the object orientation vector, combined with the camera rotation angle, to represent the 3D orientation.

Although scalable orientation data is available now, training a reliable orientation prediction model remains non-trivial. Direct regression of the three angles struggles to converge, resulting in poor performance. To overcome this challenge, we reformulate the single angle values as probability distributions to better capture the correlation between adjacent angles. By driving the model to fit these angle probability distributions, we simplify the learning process and significantly enhance model robustness. Furthermore, considering the domain gap between the rendered and real images, we investigate various model initializations that incorporate real-world prior knowledge, alongside data augmentation strategies to improve synthetic-to-real transfer.

Contribution can be summarized as:

- Develop a reliable and automatic 3D object orientation annotation pipeline, and highlight the values of rendering 3D objects for generating cost-effective, diverse, and scalable image datasets with precise orientation labels.
- Introduce the orientation probability distribution fitting task as the learning objective to stabilize the training process and improve generalization.
- Investigate various model initialization and data augmentation strategies to improve synthetic-to-real transfer.
- Our model exhibits much stronger orientation estimation ability compared to both the expertise model (Cube RCNN) and leading VLMs (GPT-4o and Gemini).

Falcon's (left man in image) perspective, is Captain America (right man in image) on his left or right?



For Falcon, Captain America is **on his right**.

For Falcon, Captain America is **on his right**.

Figure 2. Understanding object orientation is essential for spatial reasoning. However, even advanced VLMs like GPT-4o and Gemini-1.5-pro are not yet able to resolve the basic orientation issue.

## 2 Related Work

### 2.1 Orientation-based Understanding

Object orientation provides context about how objects are positioned relative to one another and to the viewer (or the camera), which is fundamental for object pose and relationship understanding. Accurate orientation understanding plays a key role in many advanced applications.

In 3D scene understanding, many studies [1, 3, 8] have highlighted the importance of spatial relationships informed by object orientation. A3D [28] first describes the position and orientation of an agent in a 3D scene, then tasks the model with answering questions based on the given spatial context. 3DoidedScan [44] manually annotates orientations for 3D objects and utilizes the pose information to describe spatial relationships among objects in 3D space.

In the domain of 2D images, understanding object orientation is also fundamental for accurately interpreting [13, 46] or generating [17, 38, 45] spatial relationships and properties. Al et al. [13] propose the visual perspective-taking task to assess 2D VLM's ability to understand the orientation and viewpoint of a person in images and highlight various applications based on this ability. Furthermore, the object orientation relative to the camera determines its pose in the image, which is essential for distinguishing spatial properties such as the front wheels of cars and the left shoulder of a person, along with complex spatial relationships. Moreover, generating objects with given pose conditions is vital for controllable image generation [17, 45].

Though object orientation is closely linked to numerous questions and applications, practical solutions for esti-

mating object orientation in images are still underexplored. Our work fills this gap by proposing the first foundation model for object orientation estimation, which exhibits strong zero-shot performance in real-world scenarios.

## 2 Object Orientation Recognition in Images

The tasks attempt to recognize object orientations in images under certain conditions or with extra information.

Object pose estimation [14] focuses on detecting the position and orientation of objects in images. However, existing methods require the CAD model of the target object [31, 40] or other reference view of this same object [10, 12, 29], which means that these methods cannot infer the orientation of the object from a single image. On the other hand, rotated object detection [6, 15, 47] focuses on generating rotatable 2D or 3D bounding boxes for objects. Omni3D [6] unifies multiple 3D object detection datasets and trains Cube R-CNN to identify the 3D position and orientation of objects from a single image. While Cube R-CNN demonstrates a certain capability in detecting objects in 3D space, its performance is constrained by the data scope of Omni3D, which predominantly features indoor scenes and street environments. Furthermore, the orientation angles predicted by Cube R-CNN are primarily used to rotate the 3D bounding box while not always aligning with the front face of the objects.

Like the aforementioned tasks, our work focuses on 3D orientation estimation of objects in single- and free-view images, and the orientation is strictly aligned with the meaningful front face of the objects.

## 3 Orientation Understanding in 2D VLMs

Proposing our method for object orientation estimation, we first investigate whether the ability to understand object orientation emerges in 2D VLMs trained on web-scale image datasets with billions of parameters.

This end, we introduce Ori-Bench, the first VQA benchmark specifically designed to assess the capacity of 2D VLMs to understand object orientation and tackle related questions. We manually curate 200 images in total, with 100 from COCO [23] and 100 generated by DALL-E 3 [5]. To substantively evaluate the understanding of object orientation, each image is horizontally flipped to produce a paired mirrored version, with answers adapted accordingly. A sample will be marked as solved only if the model correctly answers the question on both versions. There are three kinds of tasks: (1) *Object Direction Recognition* (73+73 samples): identifying the orientation of an object within images; (2) *Spatial Part Reasoning* (39+39 samples): distinguish parts of an object with specific spatial meanings, like left vs. right hand of human; (3) *Spatial Relation Reasoning* (88+88 samples): imagining the relative position of one object from the perspective of another.

	Object Direction	Spatial Part	Spatial Relation	Overall
Random	12.93	22.12	17.54	16.75
GPT-4o	49.32	15.38	27.27	32.50
Gemini-1.5-pro	58.90	15.38	18.18	33.00
Orient Anything+LLM	<b>67.12</b>	<b>46.15</b>	<b>40.91</b>	<b>51.50</b>

1. Quantitative results on the proposed Ori-Bench.

In Tab. 1, we show the accuracy of GPT-4o, Gemini-1.5-Pro, and our Orient Anything+LLM (Refer to Sec. 7.1 for details). In the basic direction recognition task, the advanced VLMs can only correctly solve around 60% samples. This limitation is especially evident in spatial reasoning and relation tasks, where the powerful GPT-4o and Gemini-1.5-Pro perform similarly to random guessing. This pilot study highlights the need for fundamental tools to precisely estimate object orientation in images. The examples are provided in Supplementary Materials.

## 4 Orientation Data Collection

The scarcity of orientation annotations is a major obstacle to learning general orientation estimation. Existing annotations for images, typically captions [36, 37], bounding boxes [23], or segmentation masks [20, 50], seldom include object orientation information, and manually annotating object orientation in images is extremely time-consuming and costly. To overcome this limitation, we propose to utilize the 3D assets. Rotating the front face of 3D objects and then rendering images from random perspectives provides an efficient and effective way to generate large-scale image datasets with precise orientation annotations.

To this end, we first develop an automatic 3D object's orientation annotation and rendering pipeline, as shown in Fig. 3. Each step of the pipeline is detailed below.

**Step 1: Canonical 3D Models Filtering** We use Objaverse [9], a large-scale dataset containing 800K object assets, as our database. Though most objects in this dataset are modeled in canonical poses (standing upright and facing one of four orthogonal directions along the  $x$ ,  $-x$ ,  $y$ , and  $-y$  axes), some are tilted along orthogonal axes, as shown in Fig. 3.a. To simplify orientation annotation and enhance reliability, we first exclude all the tilted assets, focusing solely on 3D objects in canonical poses. This idea reduces the 3D object orientation annotation problem to a multi-class classification task. Rather than identifying the specific orientation vector, we only need to determine the front face from images rendered alone  $x$ ,  $-x$ ,  $y$ , and  $-y$  axes, or to conclude that the object has no front face.

To filter out tilted objects, we analyze the tilt in the three orthographic views of each object. Specifically, we extract the object edges for each view, and use Principal Component Analysis (PCA) to capture the principal directions of

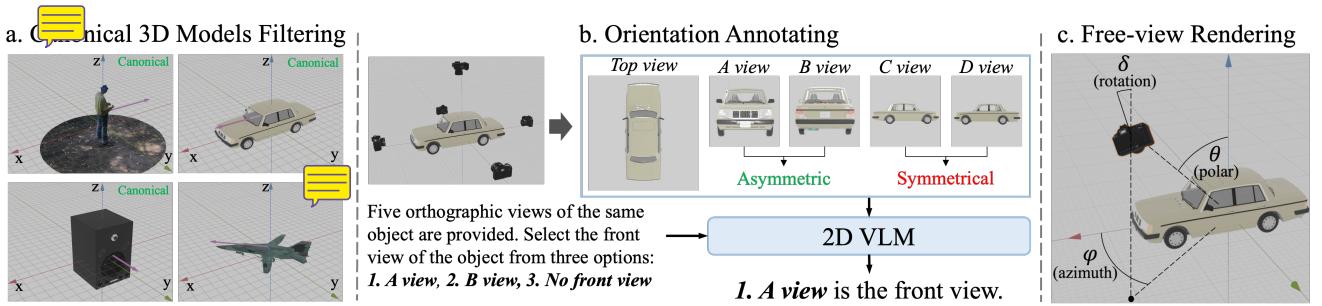


Figure 3. The orientation data collection pipeline is composed of three steps: **1) Canonical 3D Model Filtering**: This step removes any 3D objects in tilted poses. **2) Orientation Annotating**: An advanced 2D VLM is used to identify the front face from multiple orthogonal perspectives, with view symmetry employed to narrow the potential choices. **3) Free-view Rendering**: Rendering images from random and free viewpoints, and the object orientation is represented by the polar  $\theta$ , azimuthal  $\varphi$  and rotation angle  $\delta$  of the camera.

edges. If the principal edge direction is parallel with any coordinate axis (with a tolerance of two degrees for robustness) across all renderings, the object is considered to be in the canonical pose; otherwise, it will be deemed tilted.

Starting with the initial pool of 800K objects in the Obaverse dataset, we first curate 80K 3D models with high texture quality. Using our tilt-filtering criteria, we select 55K objects in canonical poses for subsequent processing.

**Step2: Orientation Annotating** Using the selected 3D objects in canonical poses, we render four orthogonal views from the  $x$ ,  $-x$ ,  $y$  and  $-y$  axes, along with a top view for additional global reference. Although our pilot study in Sec. 3 indicates that current 2D VLMs struggle to accurately predict orientation from a single view, we find that they perform well in identifying which view is facing the camera when multiple orthogonal views are presented for comparison and reference.

Additionally, to mitigate VLM hallucinations and improve annotation accuracy, we incorporate symmetry as auxiliary information. Since the front and back faces of objects are typically asymmetrical, we leverage this prior knowledge to further narrow down the possible choices. Specifically, we use a combination of SIFT [27], structural similarity, and pixel color similarity to assess the similarity between opposing views. Two views are considered symmetrical if their similarity exceeds the threshold. Gemini-1.5-Pro is tasked with identifying the front face of objects from asymmetrical opposing views. If the object is symmetrical along both the  $x$  vs.  $-x$  and  $y$  vs.  $-y$ , it is regarded as having no meaningful front face and orientation.

**Step3: Free-View Rendering** Once the 3D object’s orientation is annotated in 3D space, we can obtain its 3D orientation in images from any viewpoints. For simplicity and clarity, we use the spherical coordinate system to define object orientation. As depicted in Fig. 3.c, we calculate the rel-

ative polar angle  $\theta$  and azimuth angle  $\varphi$  between the camera position and the object orientation axis, as well as the camera rotation angle  $\delta$ , to represent the object orientation from the specific viewpoint.

Before rendering, all 3D objects are scaled to a unit cube, with their centers aligned to the origin of the coordinate system. For each object, 40 images are rendered from random perspectives, with the camera aimed at the origin and each image rendered at  $512 \times 512$  resolution. In total, we collect 2M rendered images with precise orientation annotations.

## 5. Orient Anything

Based on the massive images of objects with annotated 3D orientation  $\theta$ ,  $\varphi$ , and  $\delta$ , we train Orient Anything for general object orientation estimation in images.

### 5.1. Orientation Probability Distribution Fitting

Despite having accurate 3D orientation annotations, developing an effective learning objective to guide accurate and robust orientation predictions is non-trivial. Our initial approach, which involved directly predicting continuous angle values with L2 loss as supervision, struggles to converge and performs poorly.

To address this, we first simplify the challenging continuous regression task into a discrete classification problem, which is easier to optimize. Specifically, we divide the  $360^\circ$  range into 360 individual classes, each representing a  $1^\circ$  interval. While lowering the task difficulty improves performance over continuous regression, it fails to capture correlations between adjacent angles that produce nearly identical outcomes in practice (e.g., rendering at polar  $29^\circ$ ,  $30^\circ$ , and  $31^\circ$ ). Treating these close angles as independent classes neglects their inherent relationships, which may confuse the model. Therefore, we further reformulate the classification task as a discrete probability distribution fitting problem, which is also easy to converge and can fully capture the potential relationship between different orientations.

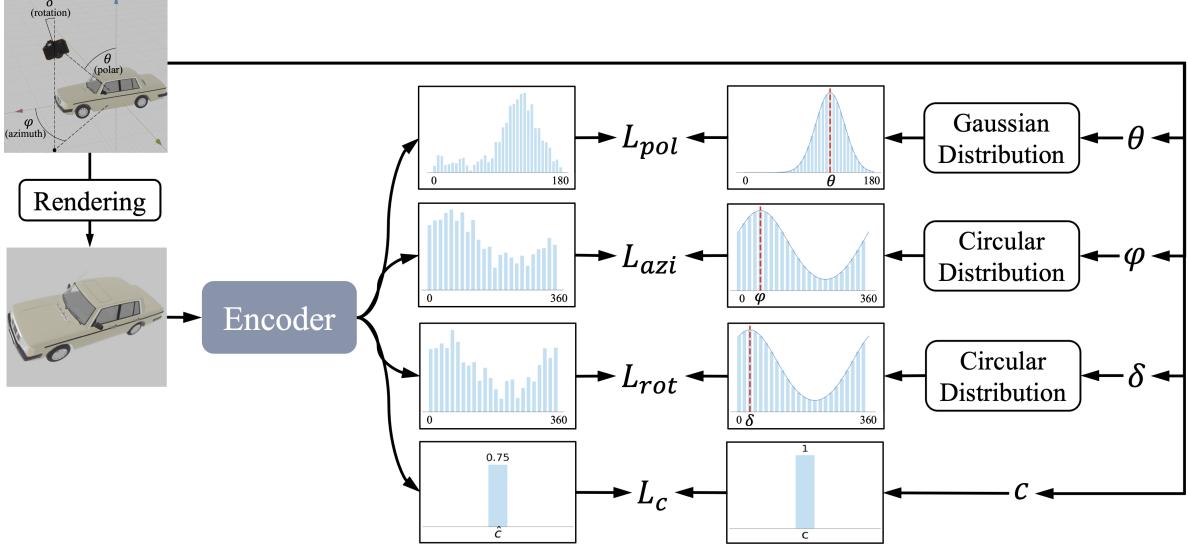


Figure 4. Orient Anything consists of a simple visual encoder and multiple prediction heads. It is trained to judge if the object in the input image has a meaningful front face and fits the probability distribution of 3D orientation.

**Target Probability Distribution** We first transform the ground-truth angles into target probability distributions, represented as Gaussian distributions centered on the ground-truth angle, with manually set variances. These distributions are subsequently discretized into a grid-based format at  $1^\circ$  intervals. For a given ground-truth polar angle  $\theta$  (in degrees), the probability distribution of polar angle  $\mathbf{P}_{\text{pol}}(i|\theta, \sigma_\theta)$  can be formulated as follows:

$$\mathbf{P}_{\text{pol}}(i|\theta, \sigma_\theta) = \frac{\exp\left(-\frac{(i-\theta)^2}{2\sigma_\theta^2}\right)}{\sum_{n=1}^{180} \exp\left(-\frac{(n-\theta)^2}{2\sigma_\theta^2}\right)}, \quad (1)$$

where  $i = 1^\circ, \dots, 180^\circ$  and  $\sigma_\theta$  is the variance hyper-parameter for polar distribution. For the ground truth azimuth angle  $\varphi$  and rotation angle  $\gamma$ , due to its periodicity (e.g.,  $359^\circ, 360^\circ$ , and  $1^\circ$  are adjacent), we employ the circular Gaussian distribution to from their target distribution  $\mathbf{P}_{\text{azi}}(i|\varphi, \sigma_\varphi)$  and  $\mathbf{P}_{\text{rot}}(i|\delta, \sigma_\delta)$ . For brevity, we illustrate this process using azimuth as an example:

$$\mathbf{P}_{\text{azi}}(i|\varphi, \sigma_\varphi) = \frac{\exp\left(\frac{\cos(i-\varphi)}{\sigma_\varphi^2}\right)}{2\pi I_0\left(\frac{1}{\sigma_\varphi^2}\right)}, \quad (2)$$

where  $i = 1^\circ, \dots, 360^\circ$ ,  $\sigma_\varphi$  is the variance for polar distribution, and  $I_0\left(\frac{1}{\sigma_\varphi^2}\right)$  is the zero-order modified Bessel function of the first kind, which can be represented as:

$$I_0\left(\frac{1}{\sigma_\varphi^2}\right) = \sum_{n=0}^{\infty} \frac{1}{(n!)^2} \left(\frac{1}{2\sigma_\varphi^2}\right)^{2n}. \quad (3)$$

As shown in Fig. 4, the circular Gaussian distribution effectively models the periodicity of azimuth and rotation angles, ensuring the stability of the optimization process.

**Training and Inference** Given the input image  $I$ , we use a visual encoder to extract its latent feature, followed by prediction heads (simple linear layers) to output the distributions of polar, azimuth and rotation angles:  $\hat{\mathbf{P}}_{\text{pol}} \in \mathbb{R}^{180}$ ,  $\hat{\mathbf{P}}_{\text{azi}} \in \mathbb{R}^{360}$ , and  $\hat{\mathbf{P}}_{\text{rot}} \in \mathbb{R}^{360}$ , respectively, representing the object orientation in 3D space. Additionally, the model predicts an orientation confidence  $\hat{c} \in \mathbb{R}^1$ , to determine whether the object has a defined front face and orientation. This approach is used for handling centrally symmetric objects like balls and stools. The target distributions  $\mathbf{P}_{\text{pol}}(i|\theta, \sigma_\theta)$ ,  $\mathbf{P}_{\text{azi}}(i|\varphi, \sigma_\varphi)$  and  $\mathbf{P}_{\text{rot}}(i|\delta, \sigma_\delta)$  are defined above, with the orientation label  $c$  being 1 if the object has a front face, and 0 otherwise. We use cross-entropy (CE) loss to supervise the predicted orientation distributions, and the corresponding loss terms are denoted as:  $L_{\text{pol}}$ ,  $L_{\text{azi}}$  and  $L_{\text{rot}}$ . For  $\hat{c}$ , binary cross-entropy (BCE) loss is employed, yielding  $L_c$ . The final training loss is a linear combination of the above four terms, and for objects without meaningful orientation, the  $L_{\text{pol}}$ ,  $L_{\text{azi}}$ ,  $L_{\text{rot}}$  will be disabled:

$$L = \begin{cases} \lambda L_c, & c = 0 \\ L_{\text{pol}} + L_{\text{azi}} + L_{\text{rot}} + \lambda L_c, & c = 1 \end{cases} \quad (4)$$

where  $\lambda$  is the loss coefficient for orientation judgment.

During the inference process, objects whose orientation confidence is lower than 0.5 would be thought to have no meaningful front face and orientation. Otherwise, the angles with the highest probability in each distribution:  $\hat{\mathbf{P}}_{\text{pol}}$ ,

$\hat{\mathbf{P}}_{\text{azi}}, \hat{\mathbf{P}}_{\text{rot}}$  are taken as the predicted polar, azimuth, and rotation angle:  $\hat{\theta}, \hat{\varphi}, \hat{\delta}$ .

## 5.2. Sythetic-to-Real Transferring

Although the rendered images of 3D objects provide extensive data with orientation annotations, there is a distribution shift between synthetic rendered images and real images. We try to prompt effective synthetic-to-real transfer from two aspects: integrating real-world pre-training knowledge and narrowing the training-inference domain gap.

**Inheriting Real-world Knowledge by Initialization** As demonstrated in [19, 48], initializing the model with strong visual encoders pre-trained on real images can significantly improve its synthetic-to-real transfer ability. To evaluate this in our orientation estimation task, we train models initialized from 3 widely-used image pre-trained encoders: MAE [16], CLIP [32], and DINOv2 [30]. After trials and failures, DINOv2 yields satisfactory results, attributed to its task-agnostic pre-training, fine-grained perception, and strong generalization capabilities. Consequently, we develop our model using DINOv2 initialization.

**Narrowing Domain Gap by Data Augmentation** There are two main differences between rendered and real images. We employ corresponding data augmentation strategies to reduce the domain gap and enhance transfer performance.

First, objects in rendered images are typically fully visible, whereas real-world images often contain partially visible or occluded objects. To bridge this gap, we incorporate random cropping as a training data augmentation strategy. This technique simulates the occlusion situation in real-world images, thereby improving the model’s ability to generalize to real-world scenarios.

Second, to avoid ambiguity, the rendered image contains only one object. In contrast, real-world images often feature multiple objects. To adapt our model for such cases, we isolate each object using segmentation masks and estimate their orientations individually. This approach replicates the style of rendered images, broadens the applicability of our model, and enhances its performance on real-world images.

## 6. Experiments

### 6.1. Implementation Details

We train models at three scales for different purposes: ViT-S, ViT-B, and ViT-L, all initialized with DINOv2. The loss coefficient  $\lambda$  in Eq. 4 is set to 1. The variance hyperparameters  $\sigma_\theta$ ,  $\sigma_\varphi$ , and  $\sigma_\delta$  are configured as  $2.0^\circ$ ,  $20.0^\circ$ , and  $1.0^\circ$ . For optimization, we use the AdamW [26] optimizer, with a learning rate of  $1e-5$  for the pre-trained visual encoder and  $1e-3$  for the newly introduced prediction heads. The models are trained for 50,000 steps with a batch size of 64 on

the curated 2M object orientation dataset. All trainings are conducted on 4 A100 (40GB) GPUs.

## 6.2. Rendered-Images Orientation Estimation

We first quantitatively validate the model by accurately estimating the numerical 3D orientation of the in-domain rendered images. We manually select and annotate 300 objects from Objaverse, of which 150 have orientation annotations and 150 have no meaningful front face and orientation. For each object, we render 16 images of random views, and there are 4,800 images for testing in total.

We evaluate methods from two aspects: 1) *Orientation Judgment*: Determine if the object has a meaningful front face. 2) *Orientation (azimuth, polar, rotation) Estimation*: Predict the accurate azimuth, polar, and camera rotation angles for objects, using Absolute Error (in degrees) and Acc@X° (accuracy within tolerances of  $\pm X^\circ$ ) as metrics. Expert image-based 3D object detection model, Cube RCNN [6] and advanced VLMs, GPT-4o [18] and Gemini-1.5-pro [42], as used as baselines.

The results presented in Tab. 2 showcase the superior performance of our model in accurately predicting 3D orientation for objects. In practical azimuth estimation, our method achieves more than triple the accuracy of previous approaches. Notably, the performance of Cube RCNN and advanced VLMs is only slightly better than random guessing, with a success rate of 19.94% compared to 12.50%. In contrast, the Orient Anything ViT-L yields 73.94% accuracy and much lower absolute error, highlighting its practical value in reliably distinguishing object direction.

## 6.3. Zero-shot Real-Image Orientation Recognition

The primary goal of this work is to estimate object orientations in real images. To assess the model performance in real-world scenarios, we construct two kinds of evaluation benchmarks.

1) For objects in the wild, we collect objects from the COCO dataset and manually annotate their orientations. Given the difficulty of annotating accurate 3D orientation, we narrow our focus to more feasible scenarios by labeling object orientations on the horizontal plane in eight directions: front, back, left, right, front-left, front-right, back-left, and back-right. From the 80 categories available in the COCO validation set, we chose 20 images per category, creating a comprehensive benchmark of 1,600 samples in total. Two tasks are used for evaluation: (1) *Orientation Judgment*: Determine whether the object has a front face and orientation. (2) *Horizontal Direction Recognition*: Identify which of the eight directions on the horizontal plane the object is orienting and the recognition accuracy is reported.

2) For objects in room and street scenes, we conducted quantitative experiments on five real-world datasets: SUN RGB-D [39], KITTI [11], nuScenes [7], Objectron [2]

Models	Rendered Image								Real Image		
	Judgment		Azimuth Estimation		Polar Estimation		Rotation Estimation		Judgment	Recognition	
	Acc↑	Abs↓	Acc@22.5°↑	Abs↓	Acc@5°↑	Abs↓	Acc@5°↑	Acc↑	Acc↑	Acc↑	
Random	50.00	-	12.50	-	5.55	-	16.67	50.00	12.50		
Cube RCNN	-	89.00	12.44	27.99	10.37	132.74	2.50	-	20.25		
Gemini-1.5-pro	57.29	79.51	19.06	20.10	16.31	2.61	85.12	66.96	31.95		
GPT-4o	61.85	81.07	19.94	16.02	17.56	4.65	81.00	69.29	45.78		
Ours (ViT-S)	73.88	45.27	63.18	5.12	71.62	0.82	97.06	78.54	63.44		
Ours (ViT-B)	74.88	39.03	71.94	3.81	81.37	<b>0.26</b>	<b>99.56</b>	<b>81.25</b>	70.19		
Ours (ViT-L)	<b>76.00</b>	<b>38.60</b>	<b>73.94</b>	<b>2.94</b>	<b>86.75</b>	0.70	98.31	80.30	<b>72.44</b>		

Table 2. Orientation estimation on both in-domain rendered images and out-of-domain real images. The best results are **bold**.

	SUN RGB-D			KITTI			nuScenes			Objectron			ARKitScenes		
	Azimuth	Polar	Rotation	Azimuth	Polar	Rotation	Azimuth	Polar	Rotation	Azimuth	Polar	Rotation	Azimuth	Polar	Rotation
Cube RCNN	93.58	39.73	140.10	98.61	39.73	121.21	89.63	15.64	132.57	122.99	60.01	113.31	91.16	37.39	132.86
Ours (ViT-S)	58.20	11.63	<b>3.59</b>	65.85	5.00	1.08	72.68	5.58	2.16	39.45	23.47	18.26	69.37	14.25	2.63
Ours (ViT-B)	56.34	9.15	3.75	54.02	5.86	<b>0.21</b>	66.56	5.72	<b>1.28</b>	36.49	<b>22.13</b>	<b>18.34</b>	75.45	12.48	<b>2.60</b>
Ours (ViT-L)	<b>42.98</b>	<b>8.38</b>	3.66	<b>44.22</b>	<b>3.57</b>	0.89	<b>55.17</b>	<b>4.08</b>	1.78	<b>30.09</b>	22.19	18.54	<b>67.56</b>	<b>11.47</b>	2.82

Table 3. Zero-shot orientation estimation on five unseen real image benchmarks. Reported in absolute error.

	Single View	Canonical Views	Canonical&Symmetrical
Gemini-1.5-pro	44.00	74.00	<b>86.00</b>
GPT-4o	31.00	87.00	<b>92.00</b>

Table 4. Ablation study for Orientation Annotation.

and ARKitScenes [4]. For each benchmark, 1,000 objects with 3D orientation annotations are randomly selected and cropped from the real images to form an orientation estimation benchmark. We assess the Cube RCNN (trained on in-domain data) and Orient Anything (trained on out-of-domain rendering data) by calculating the absolute error on azimuth, polar, rotation angles between predicted 3D orientation and ground truth.

As shown in Tab. 2 and 3, despite never being exposed to real-world images during training, each version of Orient Anything demonstrates clear superiority over existing alternative methods in recognizing object orientations in real images. We provide detailed results for each of the 80 object categories in the Appendix. Orient Anything consistently outperforms previous approaches by a significant margin across most categories, achieving over 90% accuracy in major categories such as humans, animals, vehicles, and furniture.

Due to similar definitions and the same random guess results, “Acc@22.5°” for rendered image azimuth estimation and “Acc” for real image horizontal direction recog-

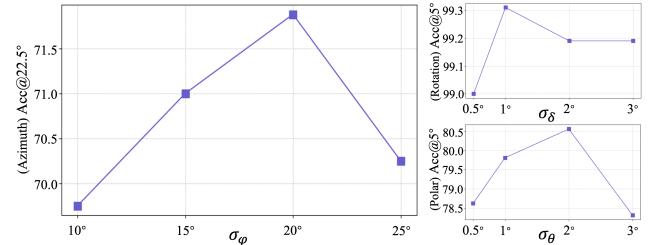


Figure 5. Ablation study for hyper-parameter  $\sigma_\theta$ ,  $\sigma_\varphi$  and  $\sigma_\delta$ .

nition are comparable. Our models achieve similar results on both metrics, highlighting the excellent synthetic-to-real transfer performance. For VLMs, recognizing horizontal directions in words is more accurate than predicting precise azimuth values in numbers, which reveals the shortcomings of VLMs in predicting precise values for 3D orientation. Cube RCNN, which predicts orientation values, performs significantly worse on rendered images due to its limited generalization capability.

We visualize our model predictions in Fig. 1, 6, and more in Appendix. These qualitative results highlight Orient Anything’s remarkable zero-shot capability across images captured or created by real cameras, human artists, or generative models, as well as a variety of scenarios, including continuous video frames, multi-view images, and complex scenes containing multiple objects.

#### 6.4. Ablation Study

To verify the effectiveness of our key designs, we conduct ablation experiments using the ViT-B encoder.

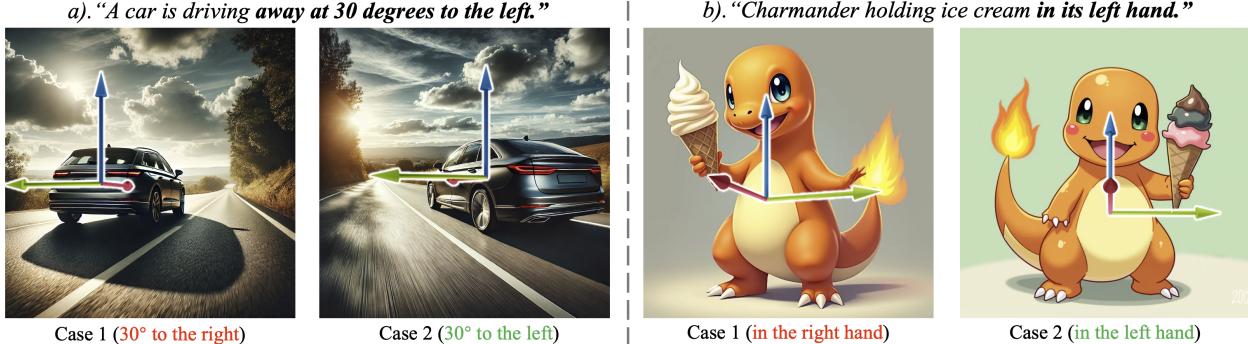


Figure 6. Generated images with given textual prompt (left two from DALL-E 3 [5], right two from FLUX [21]). Accurate orientation estimation is helpful to confirm whether generated contents follow the given orientation or perspective condition.

**Designs in Orientation Annotating** We evaluate our orientation annotating methods using the 300 manually annotated 3D objects introduced in Sec. 6.2. The results in Tab. 4 indicate that while VLMs achieve only 44% and 31% accuracy when identifying object orientation from the top view alone, providing orthogonal perspectives substantially enhances performance. Furthermore, incorporating symmetry as an extra condition further raises accuracy to nearly 90%, underscoring the effectiveness of our orientation annotating strategy and proving the reliability of the rendering data.

**Effect of  $\sigma_\theta$ ,  $\sigma_\varphi$  and  $\sigma_\delta$**  Fig. 5 shows the effect of variance hyper-parameter for three kinds of angle probability distribution. In general, our method is insensitive to the variance selection, while most configurations yield superior results compared to the one-shot label.

**Effect of Probability Prediction** In Tab. 5, we ablate the three learning objectives discussed in Sec. 5.1: continuous value regression, discrete angle classification, and probability distribution fitting. Direct regression yields poor performance, while angle classification performs significantly better but remains suboptimal. The final proposed probability distribution fitting method surpasses the alternatives, achieving markedly superior performance.

**Number of Rendering Views** We explore the effect of the number of images rendered pre-3D object in Tab. 5. For a fair comparison, we train models to converge for each setting. The results indicate that too few views fail to provide sufficient information about objects from different perspectives, while overly dense sampling results in redundant images within the dataset, potentially hindering convergence. Empirically, rendering 40 views for each object achieves the best balance and yields the optimal results.

**Effect of Model Initialization** We compare several powerful pre-trained visual encoders as initialization for our

Design	Variants	Rendering Image		Real Image
		Azimuth	Polar	Recognition
		Acc@22.5°	Acc@5°	Acc
Learning Objective	Regression	12.00	20.50	21.48
	Classification	68.75	79.00	66.93
	Fitting	<b>71.88</b>	<b>80.56</b>	<b>69.85</b>
Number of Views	10	67.19	78.19	63.67
	20	67.94	78.88	65.47
	30	70.06	78.13	68.62
	40	<b>71.88</b>	80.56	<b>69.85</b>
	80	69.12	<b>80.69</b>	66.48
Training Initialization	CLIP	58.44	71.88	49.27
	MAE	58.44	64.63	57.26
	DINOv2	<b>71.88</b>	<b>80.56</b>	<b>69.85</b>
Training Augmentation	None	71.88	80.56	69.85
	Cropping	<b>71.94</b>	<b>81.37</b>	<b>70.19</b>
Inference Augmentation	Box	71.88	80.56	67.49
	Mask	<b>71.88</b>	<b>80.56</b>	<b>69.85</b>

Table 5. Ablation study for Learning Objective, Number of Views, Training Initialization and Data Augmentation.

orientation estimation task in Tab. 5. We empirically find that DINOv2 exhibits much better performance in both in-domain convergence and out-of-domain transfer compared to others, which may be attributed to its large-scale task-agnostic pre-training and superior fine-grained perception.

**Effect of Data Augmentation** Tab. 5 present the effect of data augmentation for improving synthetic-to-real transfer. During training, random cropping enables rendered images to mimic the objects occlusions, which significantly enhances the performance in real-world scenarios. For inference, using segmentation masks to isolate objects aligns more closely with the style of rendered images compared to bounding boxes, thereby narrowing the domain gap and improving overall performance.

## 7. Applications

### 7.1. Spatial Understanding

Orientation is a key attribution for accurately understanding the spatial relations, as we highlighted in Sec. 3 and Fig. 2. We find that using Grounded-SAM [35] and our Orient Anything to identify object position and orientation in images and conveying these spatial details in pure text to an LLM [18], effectively addresses more orientation-based questions that confuse GPT-4o and Gemini-1.5-pro, as shown in Tab. 1 and examples in Appendix. These results underscore the value of our model in spatial understanding.

### 7.2. Spatial Generation Scoring

As shown in Fig. 6, we empirically find that even leading image generation models, like DALL-E 3 and FLUX, struggle to generate content that conforms to given object orientation or spatial relationship conditions. Our model can help distinguish whether the generated image follows the given spatial condition, demonstrating its potential as a reward model to guide generative models in adhering to the desired orientation- and perspective-based spatial concepts.

### 7.3. 3D Models Orientation Voting

Many existing 3D data exhibit varied orientations, with some even tilted relative to the coordinate axes. As shown in Fig. 1, our method achieves consistent orientation predictions across multi-view images, enabling robust voting for 3D object’s orientation. Accurately estimating the orientation of 3D models is valuable for further scaling up rendering images with orientation labels or adjusting the poses of 3D objects to a desired direction.

## 8. Conclusion

In this paper, we present Orient Anything, a practical approach for estimating object orientation from single images. We design an automatic and reliable 3D object annotation and rendering pipeline, allowing us to collect large-scale images with precise orientation annotations. To fully exploit the value of the new dataset, we design an orientation probability distribution fitting task for robust orientation estimation, and improve synthetic-to-real transfer performance by incorporating real-world knowledge and reducing the domain gap. As a result, Orient Anything achieves impressive zero-shot object orientation estimation in real-world images and can serve as a foundational tool for enabling applications like complex spatial understanding and generation scoring.

## References

- [1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. *Fit3d*: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 422–440. Springer, 2020. [2](#)
- [2] Adel Ahmadyan, Liangkai Zhang, Arsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7822–7831, 2021. [6](#)
- [3] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. *3dqa*: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022. [2](#)
- [4] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. [7](#)
- [5] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023. [3, 8](#)
- [6] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. *3d*: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13154–13164, 2023. [1, 3, 6](#)
- [7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Lioung, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. *nuscenes*: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. [6](#)
- [8] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. *refer*: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020. [2](#)
- [9] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. *averse*: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. [2, 3](#)
- [10] Zhiwen Fan, Panwang Pan, Peihao Wang, Yifan Jiang, Dejia Xu, and Zhangyang Wang. *e*: 6-dof promptable pose estimation of any object in any scene with one reference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7771–7781, 2024. [3](#)

- [11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 6
- [12] Walter Goodwin, Sagar Vaze, Ioannis Havoutis, and Ingmar Posner. 2d shot category-level object pose estimation. In *European Conference on Computer Vision*, pages 516–532. Springer, 2022. 3
- [13] Gracjan Góral, Alicja Ziarko, Michał Nauman, and Maciej Wołczyk. Seeing through their eyes: Evaluating visual perspective taking in vision language models. *arXiv preprint arXiv:2409.12969*, 2024. 2
- [14] Jian Guan, Yingming Hao, Qingxiao Wu, Sicong Li, and Yingjian Fang. Survey of 6dof object pose estimation methods for different application scenarios. *Sensors*, 24(4):1076, 2024. 3
- [15] Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. 3d deep features for oriented object detection. *IEEE transactions on geoscience and remote sensing*, 60:1–11, 2021. 3
- [16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 6
- [17] Yuzhong Huang, Zhong Li, Zhang Chen, Zhiyuan Ren, Guosheng Lin, Fred Morstatter, and Yi Xu. 3dtream: Streamlining text-to-3d generation with explicit orientation control. *arXiv preprint arXiv:2406.10000*, 2024. 2
- [18] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 2, 6, 9
- [19] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 6
- [20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 3
- [21] Black Forest Labs. Flux, 2024. 8
- [22] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 1
- [23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3
- [24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Vl instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1
- [25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounded dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 1
- [26] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6
- [27] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 4
- [28] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. 3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022. 2
- [29] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponomatkin, Yinlin Hu, Renaud Marlet, Mathieu Salzmann, and Vincent Lepetit. 3de: Novel object pose estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17923–17932, 2024. 3
- [30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6
- [31] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Jun Bao. 3dt: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4561–4570, 2019. 3
- [32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6
- [33] Frano Rajić, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. *arXiv preprint arXiv:2307.01197*, 2023. 1
- [34] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1
- [35] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kun-chang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 9, 12
- [36] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 3

- [37] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 3
- [38] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. ream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2
- [39] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 6
- [40] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. icit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the european conference on computer vision (ECCV)*, pages 699–715, 2018. 3
- [41] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2
- [42] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. ni 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 2, 6
- [43] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. king everything everywhere all at once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19795–19806, 2023. 1
- [44] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. odiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19757–19767, 2024. 2
- [45] QiuHong Anna Wei, Sijie Ding, Jeong Joon Park, Rahul Sajnani, Adrien Poulenard, Srinath Sridhar, and Leonidas Guibas. -net: Learning regular rearrangements of objects in rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19037–19047, 2023. 2
- [46] Zhen Wu, Jiaman Li, and C Karen Liu. han-object interaction from human-level instructions. *arXiv preprint arXiv:2406.17840*, 2024. 2
- [47] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. ited r-cnn for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3520–3529, 2021. 3
- [48] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 6
- [49] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. gnize anything: A strong image tagging model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1724–1732, 2024. 1
- [50] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 3

Category	Cube RCNN	Gemini	GPT-4o	Orient Anything (ViT-L)
bed	75%	15%	40%	100%(+25%)
monitor	35%	50%	50%	100%(+50%)
oven	50%	10%	65%	100%(+35%)
teddy bear	20%	40%	45%	100%(+55%)
motorbike	5%	20%	40%	95%(+55%)
parking meter	40%	55%	65%	95%(+30%)
laptop	65%	45%	50%	95%(+30%)
sheep	15%	45%	45%	90%(+45%)
elephant	5%	30%	55%	90%(+35%)
sofa	5%	25%	50%	90%(+40%)
toilet	55%	20%	50%	90%(+35%)
cell phone	35%	75%	80%	90%(+10%)
microwave	35%	25%	50%	90%(+40%)
clock	20%	45%	60%	90%(+30%)
bus	10%	20%	40%	85%(+45%)
traffic light	0%	35%	50%	85%(+35%)
stop sign	0%	70%	75%	85%(+10%)
bench	20%	20%	20%	85%(+65%)
bear	5%	30%	40%	85%(+45%)
zebra	5%	30%	50%	85%(+35%)
sink	0%	0%	30%	85%(+55%)
cat	20%	45%	60%	80%(+20%)
dog	10%	35%	60%	80%(+20%)
horse	10%	50%	35%	80%(+30%)
chair	10%	20%	30%	80%(+50%)
book	45%	80%	45%	80%(+0%)
car	10%	40%	45%	75%(+30%)
truck	15%	35%	60%	75%(+15%)
cow	20%	40%	40%	75%(+35%)
person	5%	35%	40%	70%(+30%)
aeroplane	15%	20%	60%	70%(+10%)
refrigerator	20%	25%	55%	70%(+15%)
bird	10%	25%	60%	65%(+5%)
giraffe	15%	30%	55%	65%(+10%)
train	30%	15%	60%	60%(+0%)
fire hydrant	20%	20%	30%	55%(+25%)
boat	10%	25%	45%	50%(+5%)
backpack	40%	65%	50%	50%(-15%)
mouse	15%	0%	0%	50%(+35%)
kite	5%	40%	60%	45%(-15%)
hair drier	0%	20%	55%	45%(-10%)
bicycle	5%	30%	30%	40%(+10%)
toaster	50%	25%	30%	40%(-10%)
remote	10%	5%	10%	5%(-5%)
keyboard	10%	0%	0%	0%(-10%)

Table 6. Detailed horizontal direction recognition accuracy for each object category in COCO that is annotated with front face and orientation. The differences between Orient Anything and the best results achieved by other alternative methods are also provided.

## A. Detailed Results on COCO Benchmark

In Tab. 6, we provide the detailed horizontal direction recognition accuracy for each object category in COCO that is annotated with front face and orientation.

Our model achieves excellent performance across most object categories with clear orientations, attaining an accuracy exceeding 80%. However, it performs relatively poorly in categories where the distinction between front and back is ambiguous or the objects are too small. Compared to previous alternatives, Orient Anything achieves significantly better accuracy in most categories than the best results achieved by previous models.

## B. Visualization of Real-image Benchmarks

In Fig. 7, 8, 9, 10 and 11, we present the qualitative results on objects of COCO, SUN RGB-D, KITTI, nuScenes, Objectron, and ARKitScenes, respectively. Our model can robustly and accurately predict the object orientation in images of various sources and resolutions.

## C. More Visualizations of Images in The Wild

In Fig. 12, we present more visualizations of images from various domains containing different objects. In these images, our model shows consistently accurate orientation prediction results, further highlighting the impressive zero-shot capability of our Orient Anything.

## D. Visualization of Ori-Bench

All Ori-Bench samples, along with the responses from GPT-4o, Gemini-1.5-pro, and Orient Anything+LLM, are included in the attached file. We visualize the three kinds of subtasks in Ori-Bench in Fig. 13, 14 and 15, respectively.

Our observations reveal that these questions, which are intuitive for humans, often confuse the state-of-the-art VLM models like GPT-4o and Gemini-1.5-pro. This highlights the inherent limitations of existing approaches to understanding orientation. By utilizing the simple template to describe object orientations estimated by Orient Anything to LLM, we outperform alternative methods by a substantial margin.

## E. Orient Anything for Orientation Understanding

In Section 7.1 of the main text, we briefly introduce the use of Orient Anything for solving orientation understanding problems. Here, we provide a detailed implementation.

For the open domain orientation understanding problem, we first use LLM to extract the object nouns in the question, then use Grounding-SAM [35] to determine the coordinates of each object, and use Orient Anything to predict the horizontal orientation of each object. We convert the detected spatial information into text descriptions with simple templates. For multiple objects, we use their coordinates to express their left-right relationship in the image. For each object, we only consider the azimuth angle and convert it into the horizontal 8-direction description. Finally, we provide these templated spatial descriptions, questions, and options in LLM. Practical examples are provided in Fig. 13, 14, and 15.

Although this method has obvious disadvantages (ignoring depth and 3D object relationships), it still performs much better than the Gemini-1.5-pro and GPT-4o.

The template description of the object relationship is:

For **OBJ1** located in [x1, y1] with predicted azimuth angle  $\hat{\varphi}$  and **OBJ2** located in [x2, y2],

*if*  $x1 < x2$ :

“From the perspective of viewer <**OBJ1**> is on the left and <**OBJ2**> is on the right of the view.”

*if*  $292.5^\circ < \hat{\varphi} < 360^\circ$  or  $0^\circ < \hat{\varphi} < 67.5^\circ$ :

“<**OBJ2**> is on the left of <**OBJ1**>.”

*if*  $67.5^\circ < \hat{\varphi} < 112.5^\circ$ :

“<**OBJ2**> is behind <**OBJ1**>.”

*if*  $112.5^\circ < \hat{\varphi} < 247.5^\circ$ :

“<**OBJ2**> is on the right of <**OBJ1**>.”

*if*  $247.5^\circ < \hat{\varphi} < 292.5^\circ$ :

“<**OBJ2**> is in front of <**OBJ1**>.”

*if*  $x1 > x2$ :

“From the perspective of viewer <**OBJ2**> is on the left and <**OBJ1**> is on the right of the view.”

*if*  $292.5^\circ < \hat{\varphi} < 360^\circ$  or  $0^\circ < \hat{\varphi} < 67.5^\circ$ :

“<**OBJ2**> is on the right of <**OBJ1**>.”

*if*  $67.5^\circ < \hat{\varphi} < 112.5^\circ$ :

“<**OBJ2**> is in front of <**OBJ1**>.”

*if*  $112.5^\circ < \hat{\varphi} < 247.5^\circ$ :

“<**OBJ2**> is on the left of <**OBJ1**>.”

*if*  $247.5^\circ < \hat{\varphi} < 292.5^\circ$ :

“<**OBJ2**> is behind <**OBJ1**>.”

The template description of object direction is:

For **OBJ** with predicted azimuth angle  $\hat{\varphi}$

*if*  $292.5^\circ < \hat{\varphi} < 360^\circ$  or  $0^\circ < \hat{\varphi} < 22.5^\circ$ :

“The <**OBJ**> is facing the viewer.”

*if*  $22.5^\circ < \hat{\varphi} < 67.5^\circ$ :

“The <**OBJ**> is facing the viewer and to the left of the viewer.”

*if*  $67.5^\circ < \hat{\varphi} < 112.5^\circ$ :

“The <**OBJ**> is facing to the left of the viewer.”

*if*  $112.5^\circ < \hat{\varphi} < 157.5^\circ$ :

“The <**OBJ**> is facing away from the viewer and to the left of the viewer.”

*if*  $157.5^\circ < \hat{\varphi} < 202.5^\circ$ :

“The <**OBJ**> is facing away from the viewer.”

*if*  $202.5^\circ < \hat{\varphi} < 247.5^\circ$ :

“The <**OBJ**> is facing away from the viewer and to the right of the viewer.”

*if*  $247.5^\circ < \hat{\varphi} < 292.5^\circ$ :

“The <**OBJ**> is facing to the right of the viewer.”

*if*  $292.5^\circ < \hat{\varphi} < 337.5^\circ$ :

“The <**OBJ**> is facing the viewer and to the right of the viewer.”

## F. Prompts for VLMs

**Question Answering for Ori-Bench and Orientation Recognition:** I will ask you a single-choice question about the content of the picture. Here is the question: <image> <question> <options>.

**Orientation Annotating for Orthogonal Rendering views:** I’m going to show four images of the same object from four viewpoints in turn and label them ‘A.’ ‘B.’ ‘C.’ ‘D.’ Four options. The option ‘E.’ is “No front face or More than One front Face”. Decide whether it have a front and if yes, which one is the front of the object after the presentation. Note that: If the object is a gun, bow and arrow, etc., please use the muzzle of the gun as the front. Stick tools and weapons such as swords, axes, knives, and wrenches are considered to have no front. If you cannot decide or there is more than one front, you should choose ‘E.’. A.<image viewA> B.<image viewB> C.<image viewC> D.<image viewD> E.No front face.

**Accurate Orientation Angles Estimation:** I will ask you a question about the content of the picture. Here is the question: <image> Align the front of the object towards the viewer. Rotate the object x degrees to its right (i.e., clockwise from a top view), using a  $360^\circ$  per full circle unit system. Adjust the height of the viewer to form a pitch Angle y with the object (same unit of degrees; y is positive if the viewer is looking down at the object, and y is negative if the viewer is looking up at the object). Finally, the viewer is rotated clockwise by an Angle z (same unit of degrees) with the line connecting the viewer and the object as the axis, and a negative z indicates a counterclockwise rotation. Now, please directly predict the values of x, y, and z in float format.

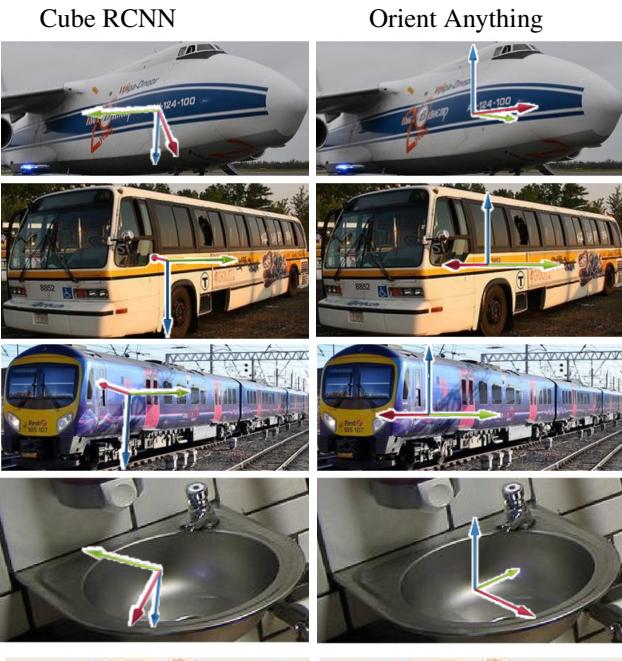


Figure 7. Qualitative results on COCO

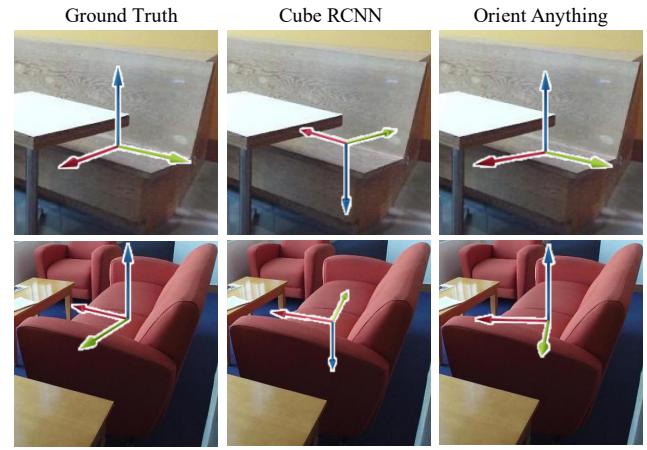


Figure 8. Qualitative results on SUN RGB-D.

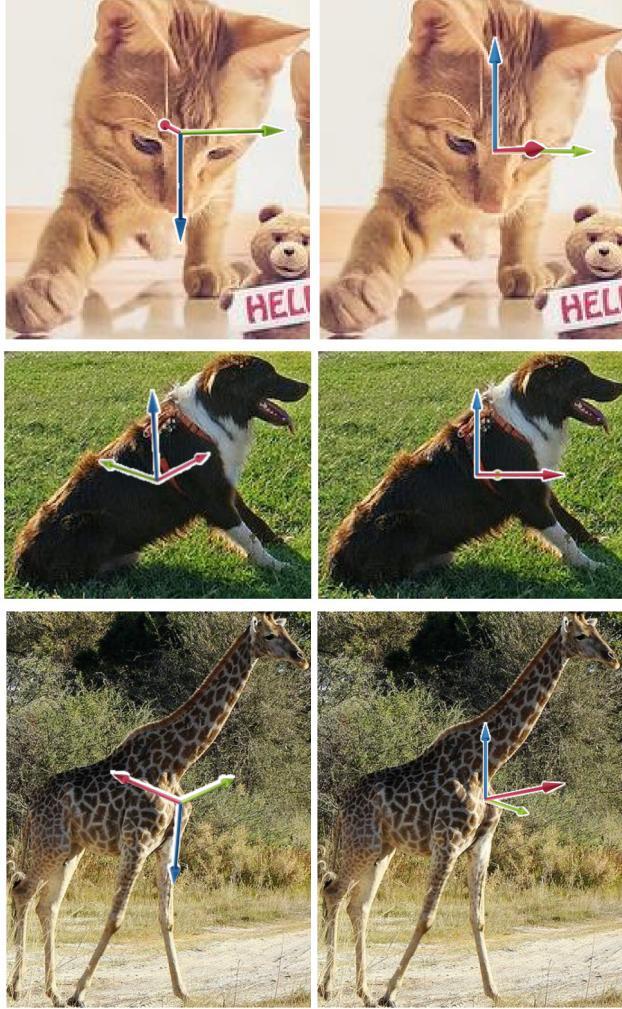


Figure 9. Qualitative results on KITTI and nuScenes.

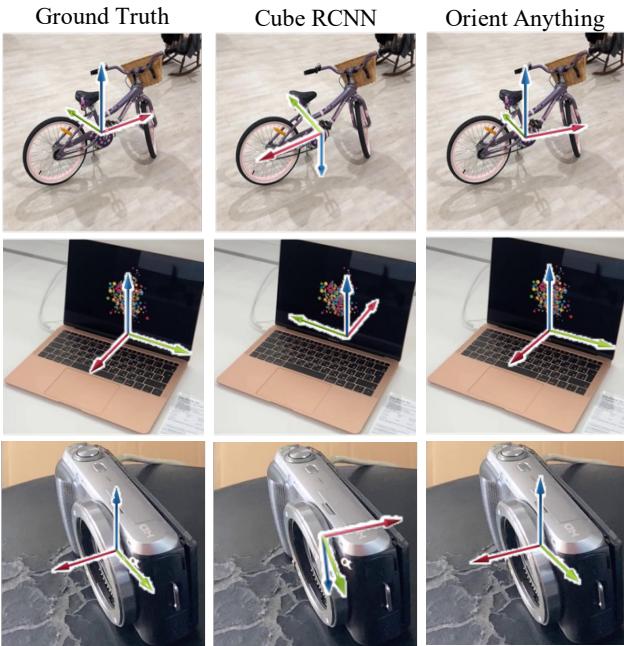


Figure 10. Qualitative results on Objectron.

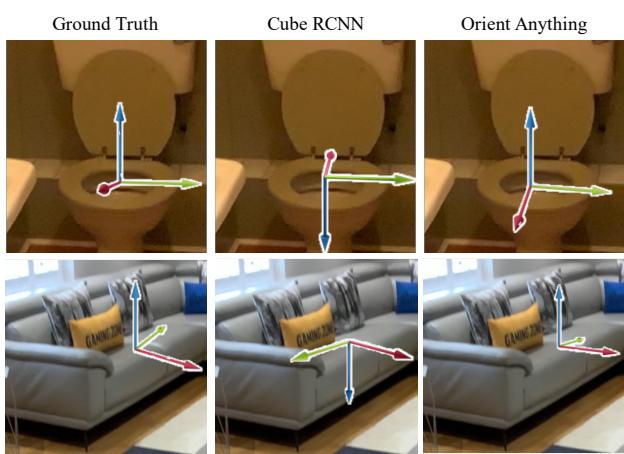


Figure 11. Qualitative results on ARKitScenes.



Figure 12. More visualization of images in the wild.



Which side is the animal facing in the picture?

- A. Facing the viewer
- B. Facing away from the viewer
- C. Facing left**
- D. Facing right

**GPT-4o:** D. The bird is oriented so that its tail is away from the viewer and its head is pointed to the right of the frame.

**Gemini-1.5-pro:** C. The bird's beak and head are pointing towards the left side of the image.

Spatial Context: 1. animal is facing to the left of the viewer.

Orient Anything+LLM: C. The animal is described as facing to the left of the viewer. This implies its left side is towards the viewer's left.



Which side is the animal facing in the picture?

- A. Facing the viewer
- B. Facing away from the viewer
- C. Facing left
- D. Facing right**

**GPT-4o:** C. The bird's head is turned towards the left side of the image.

**Gemini-1.5-pro:** D. The bird's beak and head are pointing to the right side of the image. Its tail feathers point to the left.

Spatial Context: 1. animal is facing to the right of the viewer.

Orient Anything+LLM: D. The animal is described as facing to the right of the viewer, meaning its front is towards the right.



Which orientation the horse is facing to?

- A. Facing the viewer while slightly to the right
- B. Facing the viewer while slightly to the left
- C. Facing away from the viewer while slightly to the right**
- D. Facing away from the viewer while slightly to the left

**GPT-4o:** D. The horse's head is turned to the left and the body faces the direction away from the viewer.

**Gemini-1.5-pro:** D. The horse's rear is towards the viewer and its head is slightly turned to the left.

Spatial Context: 1. horse is facing away from the viewer and to the right of the viewer.

Orient Anything+LLM: C. The question states that the horse was facing away from the viewer and to the right of the viewer. This directly matches the description of option C.



Which orientation the horse is facing to?

- A. Facing the viewer while slightly to the right
- B. Facing the viewer while slightly to the left
- C. Facing away from the viewer while slightly to the right
- D. Facing away from the viewer while slightly to the left**

**GPT-4o:** D. The horse's head is pointed to the left of the image.

**Gemini-1.5-pro:** D. The horse's head is turned towards the left side of the image, which indicates that its body is oriented slightly to the left while facing away from the viewer.

Spatial Context: 1. horse is facing away from the viewer and to the left of the viewer.

Orient Anything+LLM: D. The horse is described as facing away from the viewer and to the left of the viewer. This aligns with option D.



Which side is the girl facing in the picture?

- A. Facing the camera
- B. Facing away from the camera
- C. Facing left**
- D. Facing right

**GPT-4o:** D. The girl is running and her body is angled towards one side.

**Gemini-1.5-pro:** C. The girl's face is visible, but her body is oriented to the left of the frame, indicating that she is running towards the left side of the image.

Spatial Context: 1. girl is facing to the left of the viewer.

Orient Anything+LLM: C. Since the girl is facing to the left of the viewer, she is turned to her left from the perspective of the camera/viewer.



Which side is the girl facing in the picture?

- A. Facing the camera
- B. Facing away from the camera
- C. Facing left
- D. Facing right**

**GPT-4o:** D. The girl appears to be moving towards the right side of the image.

**Gemini-1.5-pro:** D. The girl's face is visible, and her body is oriented towards the right side of the image.

Spatial Context: 1. girl is facing to the right of the viewer.

Orient Anything+LLM: D. The girl is facing to the right of the viewer, meaning from the viewer's perspective, she is turned to her own left side, which corresponds to option D.

Figure 13. Visualization and qualitative comparison on the *Object Direction Recognition* task of Ori-Bench.



Which paw is the dog raising?

- A.its left front paw
- B.its right front paw

**GPT-4o:** **B.** The visible paw appears to be on the right side from the dog's perspective, indicating it's the right front paw.

**Gemini-1.5-pro:** **A.** The dog's right paw would be on the left side of the image from the viewer's perspective. The dog is raising the paw that appears on the right side of the image, which is its left paw.

**Spatial Context:** 1. dog is facing the viewer. 2. From the perspective of viewer, dog is on the left and paw is on the right of the view. 3. From the perspective of dog, paw is on the left of dog.

**Orient Anything+LLM:** **A.** From the viewer's perspective, the dog is facing them, with its paw also facing them. This implies the viewer is looking directly at the dog. The viewer perceives the paw on the right. From the dog's perspective, this would be its left paw.



Which paw is the dog raising?

- A.its left front paw
- B.its right front paw

**GPT-4o:** **A.** The paw shown is on the left side of the image.

**Gemini-1.5-pro:** **A.** The dog's head is tilted slightly to the right, indicating that the paw being raised is on its left side. Also, considering the camera's perspective, the paw visually closer to the viewer would be the left paw.

**Spatial Context:** 1. dog is facing the viewer. 2. From the perspective of viewer, paw is on the left and dog is on the right of the view. 3. From the perspective of dog, paw is on the right of dog.

**Orient Anything+LLM:** **B.** Paw is facing the viewer and is on the left from the viewer's perspective, but on the right from the dog's perspective, meaning it is the dog's right front paw.



Which hand is the man raising?

- A.his left hand
- B.his right hand

**GPT-4o:** **A.** The thumb is on the right side of the raised hand, indicating it is his left hand.

**Gemini-1.5-pro:** **B.** The man's right thumb is on the left side of the image, indicating that it is his right hand raised.

**Spatial Context:** 1. man is facing the viewer. 2. From the perspective of viewer, hand is on the left and man is on the right of the view. 3. From the perspective of man, hand is on the right of man.

**Orient Anything+LLM:** **B.** From the man's perspective, the hand that is on the right is his right hand. Since the viewer sees the hand on the left, the man would be raising his right hand.



Which hand is the man raising?

- A.his left hand
- B.his right hand

**GPT-4o:** **B.** The man is raising the hand that appears on the right side of the image, which corresponds to his right hand.

**Gemini-1.5-pro:** **B.** The man is raising the hand that is on the right side of the image, from the viewer's perspective. This is his right hand.

**Spatial Context:** 1. man is facing the viewer. 2. From the perspective of viewer, man is on the left and hand is on the right of the view. 3. From the perspective of man, hand is on the left of man.

**Orient Anything+LLM:** **A.** From the perspective of the man, the hand on his left is being mentioned. Since the man is facing the viewer and his left hand is on the viewer's right, it means he is raising his left hand.



Which leg of the elephant is closest to the camera?

- A.its left front
- B.its right front
- C.its left rear
- D.its right rear

**GPT-4o:** **A.** The elephant's left front leg appears prominently in the foreground.

**Gemini-1.5-pro:** **B.** The elephant's right front leg is closest to the camera. We can tell because the perspective of the image makes it appear larger and more prominent than the left front leg.

**Spatial Context:** 1. elephant is facing the viewer and to the left of the viewer. **Orient Anything+LLM:** **A.** The elephant is facing the viewer, and its left side is also closest to the viewer. Therefore, the leg closest to the camera is its left front leg. A



Which leg of the elephant is closest to the camera?

- A.its left front
- B.its right front
- C.its left rear
- D.its right rear

**GPT-4o:** **A.** The illustration shows the left front leg as most prominent and forward.

**Gemini-1.5-pro:** **B.** The elephant's left front leg is partially obscured by its trunk and head, making the right front leg clearly the closest to the camera.

**Spatial Context:** 1. elephant is facing the viewer and to the right of the viewer. **Orient Anything+LLM:** **B.** Since the elephant is facing the viewer and to the right of the viewer, its right side is closer to the camera. Among the given options, its right front leg is the closest.

Figure 14. Visualization and qualitative comparison on the *Spatial Part Reasoning* task of Ori-Bench.



From the dog's perspective, is the cat on its left or right?  
**A.**Left  
**B.**Right



From the dog's perspective, is the cat on its left or right?  
**A.**Left  
**B.**Right

**GPT-4o:** **A.** The cat is positioned closer to the window and is facing the dog, putting the cat on the dog's left side.

**Gemini-1.5-pro:** **B.** The dog is looking at the cat and the cat is looking down. To the dog's perspective, the cat is on its right.

**Spatial Context:** 1. dog is facing the viewer and to the left of the viewer. 2. cat is facing away from the viewer and to the left of the viewer. 3. From the perspective of viewer, dog is on the left and cat is on the right of the view. 4. From the perspective of dog, cat is on the left of dog.

**Orient Anything+LLM:** **A.** Since the dog is facing the viewer, to find the dog's perspective, one can imagine turning around such that the dog is looking at the viewer. From this angle, the cat, which is to the viewer's right, would then logically be on the dog's left.



Will the girl run into the lamp post if she walks forward?  
**A.**Yes  
**B.**No



Will the girl run into the lamp post if she walks forward?  
**A.**Yes  
**B.**No

**GPT-4o:** **A.** The girl is standing close to the lamp post and facing it, so if she walks directly forward, she is likely to run into it. **A**

**Gemini-1.5-pro:** **B.** The girl is standing next to the lamp post, facing the camera. If she walks forward, she will walk away from the lamp post, down the street.

**Spatial Context:** 1. girl is facing to the left of the viewer. 2. From the perspective of viewer, girl is on the left and lamp post is on the right of the view. 3. From the perspective of girl, lamp post is behind girl.

**Orient Anything+LLM:** **B.** From the perspective of the girl, the lamp post is behind her. So, if she walks forward, she will not run into it.

**GPT-4o:** **B.** The girl is standing beside the lamp post, not directly in front of it. **B**

**Gemini-1.5-pro:** **B.** The girl is standing next to the lamp post, facing the street. If she walks forward, she will be walking away from the lamp post, down the street.

**Spatial Context:** 1. girl is facing to the right of the viewer. 2. From the perspective of viewer lamp post is on the left and girl is on the right of the view. 3. From the perspective of girl, lamp post is behind girl.

**Orient Anything+LLM:** **B.** Since lamp post is behind the girl from her perspective, walking forward means moving away from the lamp post.



If someone walks out the door, on which side would the tree be relative to them?

- A.**left side
- B.**right side



If someone walks out the door, on which side would the tree be relative to them?

- A.**left side
- B.**right side

**GPT-4o:** **A.** The tree is to the left of the house entrance.

**Gemini-1.5-pro:** **B.** The tree is located on the right side of the house, as observed from the front door.

**Spatial Context:** 1. house is facing the viewer and to the right of the viewer. 2. From the perspective of viewer, house is on the left and tree is on the right of the view. 3. From the perspective of house, tree is on the left of house.

**Orient Anything+LLM:** **A.** The person walking out of the house will have the perspective of the house. Since the tree is on the left of the house from this perspective, the tree will be on the left side.

**GPT-4o:** **A.** The tree is on the left side when facing the house from the outside, so as someone exits, it would be on their left side.

**Gemini-1.5-pro:** **A.** The tree is on the left side of the door from the perspective of someone exiting the house.

**Spatial Context:** 1. house is facing the viewer. 2. From the perspective of viewer, tree is on the left and house is on the right of the view. 3. From the perspective of house, tree is on the right of house.

**Orient Anything+LLM:** **B.** If the house is facing the viewer, and the tree is on the right from the house's perspective, then someone walking out the door will have the tree on their right side. **B**

Figure 15. Visualization and qualitative comparison on the *Spatial Relation Reasoning* task of Ori-Bench.