

Spatial-SSRL: Enhancing Spatial Understanding via Self-Supervised Reinforcement Learning

Supplementary Material

A. More Related Work

LVLMM Spatial Understanding. In Sec. 2, we introduce tool-based approaches as one of the dominant paradigms of constructing large-scale spatial QAs in a cost-effective manner. In this section, we provide more related work on this approach.

Tool-based approaches integrate tools, particularly large language models (LLMs) or expert models, within their frameworks to aid data annotation. SpatialRGPT [9] uses an open-source large language model (Llama3-70B [20]) to formulate complex reasoning QAs for training. However, the performance of existing open-source and proprietary models on spatial understanding is indeed far below human level [42], causing the training data to inherit massive errors. Besides LLMs, expert models [4, 12, 28] targeting tasks such as object detection, grounding, segmentation, and depth estimation, etc., are broadly employed for data annotation. But a tool-heavy framework incurs additional computational cost, and also inherits errors from expert model outputs. In contrast, Spatial-SSRL provides a lightweight solution to formulate spatial QAs of 100% accuracy.

Self-Supervised Learning. Self-supervised learning (SSL) effectively learns visual representations from the intrinsic structure of data. Early approaches learn visual features via contrastive learning [7, 25] or self-supervised tasks such as rotation [18], jigsaw [47], and temporal ordering [46]. With the rise of LLMs and LVLMMs, SSL has become prevalent in pre-training. Pre-training of autoregressive LMs (GPT [2]), masked language models (BERT [13]), and masked autoencoders (MAE [26]) all follow the paradigm of adding masks to parts of the data and learning to predict the masked content. In this work, we repurpose SSL objectives as verifiable reward functions to directly optimize LVLMM behavior via RL, shifting supervision from representation quality to spatial understanding while remaining broadly applicable across diverse image domains.

Reinforcement Learning with Verifiable Rewards. In Sec. 2, we provide an insight into the strong potential of RLVR. And in this section, we show more related work to demonstrate the wide use and powerful performance of RLVR.

Visual-RFT designs various verifiable reward functions (e.g., IoU for object detection) to improve the performance of LVLMMs on image classification, detection, and grounding [43]. SpaceR constructs spatial reasoning scenes of verifiable answers to advance video spatial reasoning ca-

pabilities [48]. Furthermore, RLVR has also been shown to be effective in incentivizing the math reasoning of LLMs [60, 62, 72]. And RLVR-World directly applies decoded predictions of models as verifiable rewards, boosting world models on various practical tasks [64].

B. Spatial-SSRL-81k

B.1. Data Sources

Data sources. We collect raw RGB images from COCO [39] to construct *depth-free tasks*, and RGB-D images from DIODE [56] and MegaDepth [36] to construct *depth-based tasks*. These datasets provide real-world imagery spanning diverse indoor and outdoor scenes, object categories, and viewpoints. Critically, we solely use the raw images and depth maps (where available), discarding all human-provided annotations such as bounding boxes, segmentations, or captions. This ensures our pipeline remains fully self-supervised and reproducible without dependence on costly annotation infrastructure.

B.2. Statistics

Spatial-SSRL-81k consists of 81,053 samples in total. All samples belong to the self-supervised tasks in Sec. 3.1. The dataset balances depth-free and depth-based tasks in roughly equal proportions and exhibits diverse question formats, including ordering tasks, multiple-choice questions with image options, and multiple-choice questions with text options. Importantly, since all supervision is derived deterministically from image structure, Spatial-SSRL-81k achieves 100% ground-truth accuracy, which is unattainable by prior pipelines that rely on noisy detections or model-generated annotations.

The examples of all tasks are shown in Fig. 4, Fig. 5, Fig. 6, Fig. 7, and Fig. 9. Considering that *Shuffled Patch Reordering* and *Flipped Patch Recognition* share similar image layouts and structures, we regard them as one large task and evenly mix them with three other tasks, each containing approximately 20k samples. It also benefits the even distribution of the two task categories: *Depth-free* and *Depth-Based*. The number of samples for each task is demonstrated in Tab. 5. The formulation details of each task are provided in Appendix C.2, Appendix C.3, Appendix C.4, Appendix C.5, Appendix C.6.

Table 5. The size of all five tasks in the RL training dataset.

Category	Task	Size
Depth-free	<i>Shuffled Patch Reordering</i>	16,028
	<i>Flipped Patch Recognition</i>	4,005
	<i>Cropped Patch Inpainting</i>	20,200
Depth-based	<i>Regional Depth Ordering</i>	20,620
	<i>Relative Position Prediction</i>	20,200

C. Task Details

C.1. Motivation of All Tasks

Shuffled Patch Reordering: Recovering the original order of shuffled patches requires understanding global 2D layout consistency and relative positional relationships, skills that directly transfer to reasoning about object arrangements in real scenes.

Flipped Patch Recognition: Detecting subtle orientation violations requires sensitivity to local geometry, mirror symmetries, and directional cues such as text, faces, and shadows. These capabilities are essential for understanding viewpoint-dependent spatial relations.

Cropped Patch Inpainting: Identifying which patch correctly fills a masked region requires analyzing texture continuity, semantic context, and structural consistency between local content and its surroundings.

Regional Depth Ordering: Ranking regions by distance from the camera requires integrating depth cues, perspective understanding, and ordinal reasoning, which are foundational skills for tasks such as occlusion reasoning and 3D scene reconstruction.

Relative Position Prediction: Determining spatial relations from an object’s perspective (e.g., “the cup is to the left of the book”) requires mental rotation, egocentric coordinate transformation, and integration of orientation cues with depth information. These skills generalize to understanding spatial coherence.

These tasks jointly encourage both 2D layout understanding and 3D spatial reasoning, providing unambiguous targets for RLVR optimization.

C.2. Shuffled Patch Reordering

We split the images into $M \times N$ patches, where M is the number of patches in the vertical direction and N is the number of patches in the horizontal direction. The examples are provided in Fig. 4. For instance, the example on the bottom is patchified with $M = 1, N = 4$.

Difficulty Enhancement: As illustrated in Sec. 3.1.2, to increase task difficulty and prevent trivial edge-matching solutions, we execute the *mask operation* to some samples: we optionally mask one random patch $\hat{x}_{\pi(t)}$ by setting all

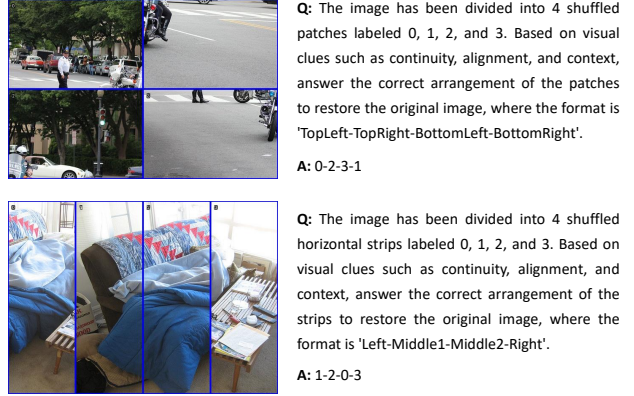


Figure 4. Examples of the task *Shuffled Patch Reordering*.

pixels to white: $\hat{x}_{\pi(t)}(0 : P_H, 0 : P_W) = 255$.

Task Statistics: The details such as M, N , mask, etc., and the corresponding number of each type of sample are listed in Tab. 6.

Table 6. The details of *Shuffled Patch Reordering* samples.

Patchify Strategy	Mask	Size
$M = 2, N = 2$	✗	4,000
$M = 2, N = 2$	✓	4,028
Horizontal ($M = 1, N = 3$ or 4)	✗	4,991
Vertical ($M = 3$ or $4, N = 1$)	✗	3,009

C.3. Flipped Patch Recognition

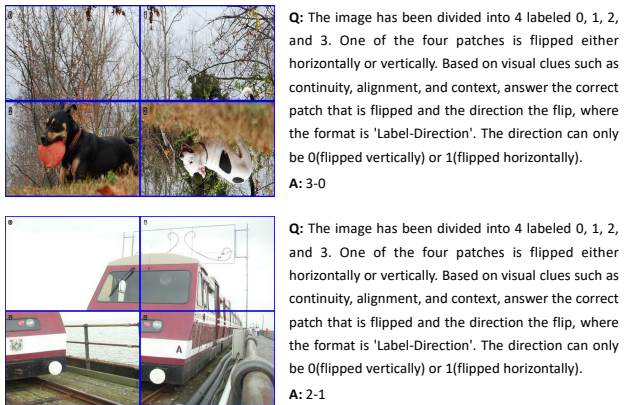


Figure 5. Examples of the task *Flipped Patch Recognition*.

The examples of this task are provided in Fig. 5.

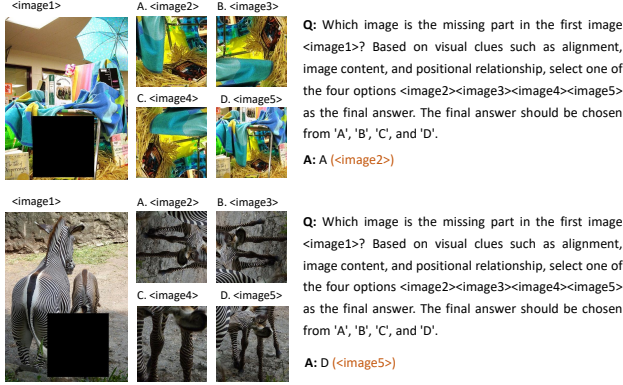


Figure 6. Examples of the task *Cropped Patch Inpainting*.

C.4. Cropped Patch Inpainting

The examples are provided in Fig. 6. Among the four options in a problem, the correct option is the patch directly cropped from the blackened area of `<image1>`, retaining its original size.

Distractor construction: As illustrated in Sec. 3.1.1, to prevent trivial solutions based on low-level texture matching, we construct three challenging distractors that share substantial visual similarity with the ground-truth patch. The three distractors are: (1) a 90° rotated version of I_{crop} ; (2) the interior subregion $I_{\text{int}} = I_{\text{crop}}(\mathcal{R}_{\text{int}})$, where $\mathcal{R}_{\text{int}} = \left[\frac{s}{4}, \frac{3s}{4} - 1\right] \times \left[\frac{s}{4}, \frac{3s}{4} - 1\right]$; and (3) the exterior region $I_{\text{ext}} = I(\mathcal{R}_{\text{ext}})$, where $\mathcal{R}_{\text{ext}} = [x_0 - \theta s, x_0 + (1 + \theta)s] \times [y_0 - \theta s, y_0 + (1 + \theta)s]$ with $\theta \in \{0.25, 0.5\}$. For \mathcal{R}_{ext} extending beyond image boundaries, we clip to $[0, H] \times [0, W]$.

All distractors are resized to $s \times s$ to prevent size-based discrimination, forcing the model to attend to fine-grained structural and semantic consistency. We ensure that all three distractors in one problem are distinct from each other and the possibility of employing each method in formulating one problem is shown in Tab. 7.

Table 7. The probability of adopting each method for constructing a distractor of one QA sample in *Cropped Patch Inpainting* task.

Method	Parameter Value	Probability
Internal Region	N/A	0.2
External Region	$\theta = 0.25$	0.2
	$\theta = 0.5$	0.2
Rotation	90° Clockwise	0.2
	90° Counterclockwise	0.2

C.5. Regional Depth Ordering

The examples of this task are provided in Fig. 7.

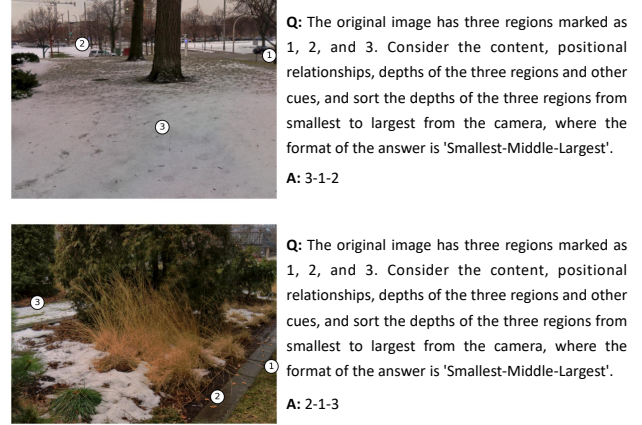


Figure 7. Examples of the task *Regional Depth Ordering*.

Hyper-parameters: As illustrated in Sec. 3.1.2, three disjoint regions R_1, R_2, R_3 are selected with increasing depth (i.e., R_1 is closest to the camera, R_3 is farthest). Two hyper-parameters r_{max} and d_{min} are applied to guarantee well-separated regions with consistent internal depth. During data construction, we set $r_{\text{max}} = 0.15$ and $d_{\text{min}} = 0.05$.

C.6. Relative Position Prediction

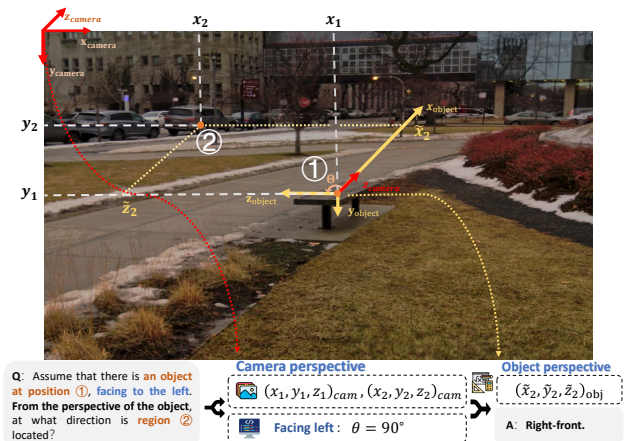


Figure 8. **The construction procedure of the task *Relative position prediction*.** We define two coordinate systems based on the camera and the hypothesized object respectively. The z -axis represents the orientation. The x -axis represents the right side. The y -axis is always vertically downward. (x_i, y_i, z_i) is the coordinate of position i in the camera system while $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ is defined in the object system.

Figure 8 depicts the detailed construction procedure of this task, offering the definition of the variables used in Sec. 3.1.2 and explaining how the ground-truth answer can be derived through our automated pipeline. The examples are provided in Fig. 9.



Q: I've taken an image and there are two regions marked as 1, and 2 on the image. Assume that there is a camera at position '1' and it's facing to me. According to the camera, where is the region marked '2'? A. Front B. Right C. Left D. Back. Consider cues such as depth, orientation, and 3D spatial relationship. The final answer should be chosen from 'A', 'B', 'C', and 'D'.

A: C (Left)



I've taken an image and there are two regions marked as 1, and 2 on the image. Assume that there is a camera at position '1' and it's facing to me. According to the camera, where is the region marker '2'? A. Right-Front B. Right-Back C. Left-Front D. Left-Back. Consider cues such as depth, orientation, and 3D spatial relationship and ignore the height difference between the two regions. The final answer should be chosen from 'A', 'B', 'C', and 'D'.

A: A (Right-Front)

Figure 9. Examples of the task *Relative Position Prediction*

Ground-truth formulation: After deriving $(\tilde{x}_2, \tilde{z}_2)$, the coordinate of R_2 in the object's coordinate frame, we define the directional labels as:

$$\tilde{p}_x = \begin{cases} \text{Right,} & \tilde{x}_2 > \delta_x \\ \text{Left,} & \tilde{x}_2 < -\delta_x, \\ \text{None,} & \text{otherwise} \end{cases} \quad (7)$$

$$\tilde{p}_z = \begin{cases} \text{Front,} & \tilde{z}_2 > \delta_z \\ \text{Back,} & \tilde{z}_2 < -\delta_z, \\ \text{None,} & \text{otherwise} \end{cases} \quad (8)$$

where δ_x and δ_z are thresholds that enforce unambiguous spatial separation. We discard instances where both $\tilde{p}_x = \tilde{p}_z = \text{None}$, ensuring all valid answers describe a clear spatial relation along at least one axis. The ground-truth answer is the tuple $(\tilde{p}_x, \tilde{p}_z)$.

Table 8. The values of parameters in formulating *Relative Position Prediction* samples. px stands for pixels and nd is the normalized depth in the raw RGB-D images, ranging from 0 to 1.

Object Orientation	θ	δ_x	δ_z
$\vec{z}_{obj} \perp$ Image Plane	0°	150 (px)	0.25 (nd)
	180°	150 (px)	0.25 (nd)
$\vec{z}_{obj} //$ Image Plane	90°	0.25 (nd)	150 (px)
	270°	0.25 (nd)	150 (px)

During formulation, we define the parameters θ to represent the orientation of the object and δ_x, δ_z as thresholds to avoid ambiguity of the ground-truth answers. δ_x is in the direction of \vec{x}_{obj} and δ_z is in the direction of \vec{z}_{obj} . We use pixels as the unit for measuring in the direction parallel to the image plane (e.g., \vec{x}_{cam}), and normalized depth

(0-1) as the unit for measuring in the direction perpendicular to the image plane (e.g., \vec{z}_{cam}). Since the orientation of the object changes due to the value of θ , δ_x and δ_z are determined by their direction (parallel or perpendicular to the image plane). We set the thresholds as 150 pixels in the direction parallel to the image plane and 0.25 in the direction perpendicular to the image plane. The detailed values are shown in Tab. 8.

C.7. Question Templates

We provide the question templates for the five self-supervised tasks. The words in green are alternative content determined in the automated construction procedure.

Shuffled Patch Reordering

1. M = 2, N = 2, w.o. mask:

Question: The image has been divided into 4 shuffled patches labeled 0, 1, 2, and 3. Based on visual clues such as continuity, alignment, and context, answer the correct arrangement of the patches to restore the original image, where the format is 'TopLeft-TopRight-BottomLeft-BottomRight'.

2. M = 2, N = 2, with mask:

Question: The image has been divided into 4 shuffled patches labeled 0, 1, 2, and 3. One of the four patches is masked completely by white pixels. Based on visual clues such as continuity, alignment, and context, answer the correct arrangement of the patches to restore the original image, where the format is 'TopLeft-TopRight-BottomLeft-BottomRight'.

3. M = 1, N = 3 or 4, w.o. mask:

Question: The image has been divided into 3(4) shuffled horizontal strips labeled 0, 1, 2, and 3. Based on visual clues such as continuity, alignment, and context, answer the correct arrangement of the strips to restore the original image, where the format is 'Left-Middle-Right' ('Left-Middle1-Middle2-Right').

4. M = 3 or 4, N = 1, w.o. mask:

Question: The image has been divided into 3(4) shuffled vertical strips labeled 0, 1, 2, and 3. Based on visual clues such as continuity, alignment, and context, answer the correct arrangement of the strips to restore the original image, where the format is 'Top-Middle-Bottom' ('Top-Middle1-Middle2-Bottom').

Flipped Patch Recognition

Question: The image has been divided into 4 patches labeled 0, 1, 2, and 3. One of the four patches is flipped either horizontally or vertically. Based on visual clues such as continuity, alignment, and context, answer the correct patch that is flipped and the direction of the flip, where the format is 'Label-Direction'. The direction can only be 0 (flipped vertically) or 1 (flipped horizontally).

Cropped Patch Inpainting

Question: Which image is the missing part in the first image <image1>? Based on visual clues such as alignment, image content, and positional relationship, select one of the four options <image2><image3><image4><image5> as the final answer. The final answer should be chosen from 'A', 'B', 'C', and 'D'.

Regional Depth Ordering

Question: The original image has three regions marked as 1, 2, and 3. Consider the content, positional relationships, depths of the three regions and other cues, and sort the depths of the three regions from smallest to largest from the camera, where the format of the answer is 'Smallest-Middle-Largest'.

Regional Depth Ordering

Question: I've taken an image and there are two regions marked as 1, and 2 on the image. Assume that there is a camera at position '1'('2') and it's facing to the left of the image. According to the camera, where is the region marked '2'('1')? A. Front (Right-Front) B. Right(Right-Back) C. Back(Left-Front) D. Left(Left-Back). Consider cues such as depth, orientation, and 3D spatial relationship. The final answer should be chosen from 'A', 'B', 'C', and 'D'.

D. Training Details

D.1. Cold-start with SFT

Our five self-supervised tasks vary substantially in difficulty and output format (ordering sequences, multiple-choice with image or text options). In preliminary experiments, we observe that directly applying RL from a pre-trained checkpoint leads to training instability and reward collapse, as the base model fails to generate valid formatted

responses (success rate $< 5\%$ for complex tasks like relative position prediction). To mitigate this, we first perform a brief SFT warm-up on a small subset of approximately 3,600 samples ($\sim 4.4\%$ of the full dataset). This cold-start phase familiarizes the model with task formats and answer structures while preserving the benefits of RL-based optimization in the subsequent stage.

D.2. GRPO

During GRPO training, a format prompt is appended at the end of the task questions to enable the model to generate reasoning content. By following a fixed format, it's also easier to extract the final answers from the model's responses for computing the accuracy reward.

Format Prompt for Training

You FIRST think about the reasoning process as an internal monologue and then provide the final answer. The reasoning process MUST BE enclosed within $\langle \text{think} \rangle \langle / \text{think} \rangle$ tags. The final answer MUST BE put in $\boxed{\}$.

E. Evaluation Details

We evaluate the most benchmarks by employing VLMEvalkit [15], an open-source toolkit that provides standardized evaluation protocols and metrics for vision-language models. For benchmarks not supported by VLMEvalKit, we make minor modifications on the official evaluation code to accommodate it the Qwen2.5-VL architecture. When evaluating models requiring reasoning process in Tab. 1 (Spatial-SSRL-3B and Spatial-SSRL-7B), we append the same format prompt used during GRPO training (Sec. 3.2) to elicit structured reasoning, ensuring consistency between training and inference. For other evaluation settings we use the original prompt in VLMEvalkit or instruct the model to give the final answer directly. In this section, we provide a concrete illustration of the detailed prompts and evaluation implementation of each benchmark that we test.

E.1. Spatial Benchmarks

Benchmarks supported in VLMEvalkit. VLMEvalkit supports *3DSRBench*, *SpatialEval*, *QSpatial-plus*, and *Spatial457*. The former two benchmarks are multiple-choice problems with a non-CoT original prompt. Therefore, we use the official code in VLMEvalkit to evaluate the reasoning-free settings in Tab. 1 and add our training format prompt to evaluate all reasoning-required settings.

QSpatial-plus targets the quantitative prediction of 3D distances and requires a strict output format encompassing *scalar* and *distance unit* to facilitate its final score computation. Instead of employing our format prompt, we follow

Table 9. Performance of Qwen3-VL-4B (baseline model) and Spatial-SSRL-4B on spatial understanding.

Models	Reasoning	Image					Video		Avg.
		Spatial457	3DSRBench	SpatialEval	QSpatial _{plus}	What’sUp	ViewSpatial	VSI-Bench	
Qwen3-VL-4B	✗	53.43	56.46	63.04	63.37	98.78	39.09	46.82	60.14
Qwen3-VL-4B	✓	55.25	55.83	71.69	61.39	96.83	41.79	38.82	60.23
Spatial-SSRL-4B	✓	57.12	59.48	72.38	59.41	97.44	42.07	42.13	61.43 (+1.29)

the official prompts (including both non-reasoning and reasoning version) during the evaluation of all models.

Spatial457 is also not in the form of multiple-choice questions, and its prompt requires CoT response. So we use the original prompt for reasoning-required baseline settings in Tab. 1, and our format prompt for Spatial-SSRL-3B (and 7B) to ensure consistency with training. For reasoning-free baseline settings, the prompt to enable direct outputs of the answers is: *Please directly give the answer.*

Other Benchmarks. *What’sUp*, *ViewSpatial*, and *VSI-Bench* are not supported in VLMEvalkit. Our evaluation implementation makes minor modification to the official code to adapt it to Qwen2.5-VL architecture while strictly preserving the original metrics and evaluation procedures.

What’sUp contains multiple-choice problems targeting the recognition of unambiguous 2D spatial relation of two objects in an image (e.g., A mug under a table). The evaluation metric is exact matching of the option letter. We apply the following format prompt for evaluating non-reasoning settings in Tab. 1: *Based on the image, choose the correct option from the list below.*, and append our training format prompt for testing reasoning settings.

ViewSpatial aims at evaluating multi-perspective spatial reasoning, requiring the model’s capability of 3D reconstruction and perspective transformation. Similarly, all problems are in the form of multiple-choice questions. We use the official code in our experiment. However, the original prompt doesn’t explicitly instruct the model to generate a reasoning process, but it also doesn’t guide the model to directly output the final answer. To accommodate it in our experiment settings, we define both the non-reasoning and reasoning format prompts as follows:

- Prompt for Reasoning-free settings: *Reply only to the corresponding option.* \nAnswer:
- Prompt for Reasoning settings: *The final answer should be the option letter from the given choices.* \n + **Format Prompt for Training**

VSI-Bench targets spatial understanding of egocentric videos. It contains multiple-choice answers and numerical answers format. We follow the metrics proposed by the benchmark [68], which uses exact matching for multiple-choice answers and Mean Relative Accuracy (*MRA*) for numerical answers. Given a numerical model prediction \hat{y} and its corresponding ground-truth value, *MRA* is defined with a confidence threshold set $\mathcal{C} = \{0.05, 0.10, \dots, 0.5\}$:

$$MRA = \sum_{x \in \mathcal{C}} \mathbb{1}(\frac{|\hat{y}-y|}{y} < x).$$

To balance efficiency and video quality, we set *max frames* to 128 for each video input.

Both non-reasoning and reasoning format prompts for VSI-Bench are given as follows:

- Prompt for Reasoning-free settings: *Answer directly with a number(integer or decimal). / Answer directly with the option letter from the given choices.*
- Prompt for Reasoning settings: *The final answer should be a number(integer or decimal).* \n + **Format Prompt for Training** / *The final answer should be the option letter from the given choices.* \n + **Format Prompt for Training**

E.2. General Visual Benchmarks

Benchmark Overview. First, we assess general VQA capabilities on four benchmarks. MMBench-v1.1-EN_{test} [40] targets general visual understanding. BLINK [16] evaluates multi-image understanding. HallusionBench [21] demands hallucination diagnostics. And RealWorldQA [66] focuses on real-world scene understanding. These benchmarks test fundamental visual perception without requiring extensive spatial analysis, so we evaluate both baseline and Spatial-SSRL models using standard prompts without reasoning instructions. Secondly, we evaluate OCR and chart understanding on three benchmarks: OCRBench [41], ChartQA_{test} [45], and SeedBench2-plus [32]. They require accurate perception and comprehensive analysis of dense textual information and detailed visual information. For these benchmarks, we apply reasoning prompts to Spatial-SSRL to maintain consistency with spatial evaluation.

Evaluation Implementation. All the benchmarks in Sec. 4.2.2 for testing models’ general visual capabilities are supported in VLMEvalkit. We use it to implement the entire evaluation of all models on these benchmarks.

Baseline Models. The baseline models are evaluated by directly applying the original prompt provided in the toolkit.

Our Models. For our models (Spatial-SSRL-3B and Spatial-SSRL-7B), we employ the original prompt in general VQA benchmarks as we discover that the reasoning process hardly yields benefits in problems that require only simple visual perception, and we append the training format prompt for *OCR and chart understanding* benchmarks for consistency with training since they demand some basic analysis (e.g., numeric comparison, calculation) as well as fine-grained comprehension of rich visual details, which

share similarities with our depth-free tasks.

E.3. Results on Qwen3-VL-based Models

We train Spatial-SSRL-4B, initialized from Qwen3-VL-4B, on our dataset Spatial-SSRL-81k solely composed of self-supervised QA samples. To make the evaluation consistent with Tab. 1 and Tab. 3 in Sec. 4, we evaluate both the non-reasoning and reasoning variants of the baseline model for spatial understanding benchmarks and compare them with Spatial-SSRL-4B. The results are shown in Tab. 9. The results of baseline models and our model on general VQA benchmarks are provided in Tab. 10.

Table 10. Performance of Qwen3-VL-4B (baseline model) and Spatial-SSRL-4B on general VQA. Our model has achieved an average accuracy gain of 1.18%.

Models	MMBench	BLINK	Hallusion	RealWorld	Avg.
Qwen3	84.10	64.81	56.38	71.11	69.10
Ours	84.26	65.23	62.23	69.41	70.28

F. Qualitative Analysis

In this section, we provide some qualitative examples to compare the performance of Spatial-SSRL-7B and our base model Qwen2.5-VL-7B on spatial understanding problems. These examples cover a broad range of spatial intelligence: 3D location understanding, depth comparison, orientation recognition, multi-object relationship perception, and reasoning on viewpoint transformation.

Height Higher



Q: Consider the real-world 3D locations of the objects. Which object has a higher location? A. yellow bear kite B. building

Qwen2.5-VL-7B:

To determine which object has a higher location, we need to compare the vertical positions of the yellow bear kite and the buildings in the background. The yellow bear kite is flying in the sky, and its position is above the ground. The buildings in the background are on the ground level, indicating they are at a lower elevation compared to the kite. Since the kite is in the air and the buildings are on the ground, the kite must be at a higher location than the buildings. Final answer: A. yellow bear kite

Spatial-SSRL-7B:

To determine which object is higher, let's analyze the image. The yellow bear kite is in the foreground, closer to the ground. The buildings in the background are further away from the viewer. Since the buildings are further away, they are higher in the real-world 3D space compared to the yellow bear kite, which is closer to the ground. Therefore, the building is higher than the yellow bear kite.

Location Above



Q: "Consider the real-world 3D locations of the objects. Is the clock directly underneath the train station roof top? A. yes B. no

Qwen2.5-VL-7B:

The image shows a train at a station with a platform on the right side. Above the platform, there is a structure that appears to be part of the train station roof. There is a clock mounted on this structure. The clock is positioned above the platform, not directly underneath it. Therefore, the clock is not directly underneath the train station roof.

Spatial-SSRL-7B:

To determine if the clock is directly underneath the train station roof top, I need to assess the spatial relationship between the clock and the roof. In the image, the clock is mounted on the roof structure, and it appears to be positioned at a height that aligns with the roof's underside. The clock is not hanging from the roof but is rather part of the roof structure itself, indicating it is directly underneath the roof.

Location Closer to Camera



Q: "Consider the real-world 3D location of the objects. Which object is further away from the camera? A. boat B. fire hydrant

Qwen2.5-VL-7B:

To determine which object is further away from the camera, we need to consider the perspective and depth in the image. The boat appears larger and more detailed, suggesting it is closer to the viewer. The fire hydrant, on the other hand, is smaller and less detailed, indicating it is farther away. Therefore, the fire hydrant is further away from the camera.

Spatial-SSRL-7B:

To determine which object is further away from the camera, I need to assess the depth cues in the image. The boat appears to be in the water, which is typically further away than objects on the land. The fire hydrant is on the ground, closer to the viewer. Therefore, the boat is further away from the camera.

Figure 10. Qualitative examples of spatial understanding

Orientation Viewpoint



Q: Consider the real-world 3D locations and orientations of the objects. Which side of the airplane is facing the camera? A. front B. left C. back D. right

Qwen2.5-VL-7B:

To determine which side of the airplane is facing the camera, let's analyze the image step by step:\n1. The airplane is positioned on the water near the bridge.\n2. The orientation of the airplane relative to the bridge and the surrounding structures can help us understand its position.\n3. The airplane appears to be facing towards the left side of the image from our perspective.\n\nGiven this analysis, the airplane is facing towards the left side of the image, which means the front of the airplane is facing the camera.\n\nThe final answer is: B

Spatial-SSRL-7B:

<think>\n\nTo determine which side of the airplane is facing the camera, I need to consider the orientation of the airplane relative to the viewer. In the image, the airplane appears to be positioned with its nose pointing towards the camera. This is because the airplane's wings and tail are visible, and the perspective suggests that the front of the airplane is facing the viewer.\n\nTherefore, the airplane is facing the camera from the front.\n</think>\n\nA

Multi-obj. Orientation



Q: Consider the real-world 3D orientations of the objects. Are the kid and the teddy bear facing same or similar directions, or very different directions? A. very different directions B. same or similar directions

Qwen2.5-VL-7B:

<think>\n\nTo determine if the kid and the teddy bear are facing the same or similar directions, let's analyze their positions:\n1. The kid is sleeping with their head slightly tilted to the side, which means they are not facing directly forward but rather at an angle.\n2. The teddy bear is positioned in front of the kid, and its face is oriented towards the camera, which suggests it is facing forward.\n\nGiven these observations:\n- The kid's head is tilted, indicating a direction that is not directly forward.\n- The teddy bear is facing forward, which is a different direction from the kid's tilted position.\n\nTherefore, the kid and the teddy bear are facing very different directions.\n</think>\n\nA (very different directions)

Spatial-SSRL-7B:

<think>\n\nTo determine the orientation of the kid and the teddy bear, I need to consider their positions relative to each other. The kid is facing forward, towards the camera, with their head slightly tilted. The teddy bear is positioned in front of the kid, also facing forward. Both the kid and the teddy bear are oriented in the same direction, which is towards the camera.\n\nTherefore, the final answer is that they are facing the same or similar directions.\n</think>\n\nB

Perspective Transformation



Q: Consider the real-world 3D locations and orientations of the objects. If I stand at the recreational vehicle's position facing where it is facing, is the dog in front of me or behind me? A. behind B. in front of

Qwen2.5-VL-7B:

<think>\n\nTo determine if the dog is in front of or behind the recreational vehicle (RV) when standing at its position, we need to consider the orientation and relative positions of the RV and the dog.\n1. The RV is positioned with its front facing towards the left side of the image.\n2. The dog is standing on the ground in the foreground, closer to the viewer than the RV.\n3. If you were to stand at the RV's position and face forward, the dog would be in your line of sight and closer to you than the RV.\n\nGiven this information, the dog is in front of the RV when you are standing at its position.\n\nThe final answer is: B

Spatial-SSRL-7B:

<think>\n\nTo determine the position of the dog relative to the recreational vehicle, I need to consider the orientation of the RV and the dog's position in the image.\n1. The recreational vehicle is facing away from the camera, as indicated by the direction of its wheels and the orientation of the trailer.\n2. The dog is positioned in the foreground, closer to the camera than the RV.\n3. Since the dog is in front of the RV from the camera's perspective, it is behind the RV from the RV's perspective.\n\nTherefore, the dog is behind the recreational vehicle if I stand at the RV's position facing where it is facing.\n</think>\n\nA

Figure 11. Qualitative examples of spatial understanding