This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PhysVLM: Enabling Visual Language Models to Understand Robotic Physical Reachability

Weijie Zhou[1,2], Manli Tao[2], Chaoyang Zhao[2,3,†], Haiyun Guo[2],
Honghui Dong[1], Ming Tang[2], Jinqiao Wang[1,2,3,4,†]
[1] School of Traffic and Transportation, Beijing Jiaotong University
[2] Foundation Model Research Center, Institute of Automation, Chinese Academy of Sciences
[3] ObjectEye Inc. [4] Guangdong Provincial Key Laboratory of Intellectual Property & Big Data, Guangdong Polytechnic Normal University
†Corresponding authors: chaoyang.zhao@nlpr.ia.ac.cn, jqwang@nlpr.ia.ac.cn

Abstract

Understanding the environment and a robot's physical reachability is crucial for task execution. While state-of-the-art vision-language models (VLMs) excel in environmental perception, they often generate inaccurate or impractical responses in embodied visual reasoning tasks due to a lack of understanding of robotic physical reachability. To address this issue, we propose a unified representation of physical reachability across diverse robots, i.e., Space-Physical Reachability Map (S-P Map), and PhysVLM, a vision-language model that integrates this reachability information into visual reasoning. Specifically, the S-P Map abstracts a robot's physical reachability into a generalized spatial representation, independent of specific robot configurations, allowing the model to focus on reachability features rather than robot-specific parameters. Subsequently, PhysVLM extends traditional VLM architectures by incorporating an additional feature encoder to process the S-P Map, enabling the model to reason about physical reachability without compromising its general vision-language capabilities. To train and evaluate PhysVLM, we constructed a large-scale multi-robot dataset, Phys100K, and a challenging benchmark, EQA-phys, which includes tasks for six different robots in both simulated and real-world environments. Experimental results demonstrate that PhysVLM outperforms existing models, achieving a 14% improvement over GPT-4o on EQA-phys and surpassing advanced embodied VLMs such as RoboMamba and SpatialVLM on the RoboVQA-val and OpenEQA benchmarks. Additionally, the S-P Map shows strong compatibility with various VLMs, and its integration into GPT-4o-mini yields a 7.1% performance improvement.

1. Introduction

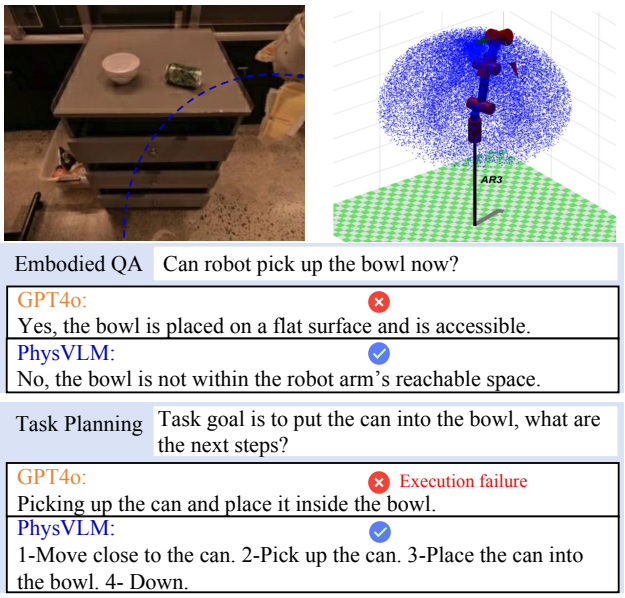


Figure 1. Existing VLM models, such as GPT-4o, may generate inaccurate or impractical responses due to poor comprehension of robotic physical reachability. The proposed PhysVLM integrates vision-language capabilities with an understanding of robotic physical reachability.

Accurate perception of physical reachability is essential for robots to perform tasks effectively. Similar to how humans adjust their actions based on bodily conditions and environmental factors, robots must account for their physical reachability within an environment to ensure efficient and reliable task execution. For instance, in grasping tasks,

a robot that fails to assess its reachability may attempt to grasp an object from an unreachable position, leading to task failure or equipment damage [32, 35]. Thus, enhancing a robot's understanding of physical reachability is crucial for successful task planning and execution in complex environments [31, 41].

Vision-Language Models (VLMs) have shown remarkable progress in environmental understanding [6, 17, 19, 21, 37], and many studies have applied these models to embodied AI to assist robots in perceiving environments and planning tasks [11, 18, 25, 36, 43]. However, while VLMs excel in general environmental perception, they often struggle with tasks that require an understanding of robotic physical reachability (see Figure 1). We identify two key challenges that must be addressed for VLMs to be effective in robotic tasks: (1) **how to develop a unified and efficient representation of physical reachability.** Robots vary significantly in size, joint types, and other characteristics, making it difficult for VLMs to directly learn these differences; (2) **how to enable VLMs to improve their understanding of physical reachability without compromising general vision-language capabilities.** Existing VLMs typically combine pre-trained unimodal encoders for vision and language tasks. However, introducing a new modality like physical reachability requires careful architectural and training adjustments to ensure the model can reason about reachability while maintaining its general capabilities.

To address these challenges, we propose the Space-Physical Reachability Map (S-P Map), a unified representation that abstracts the physical reachability of diverse robots into a generalized spatial form. The S-P Map is generated by combining robot parameters with egocentric depth images, but crucially, the model learns to focus on the abstracted reachability features (i.e., the gray regions in the S-P Map) rather than the specific robot configurations. This abstraction allows the model to generalize across different robots, as it only needs to reason about which areas are reachable, independent of the robot's specific characteristics. We introduce PhysVLM, a vision-language model that extends traditional VLM architectures by incorporating an additional feature encoder to process the S-P Map. This design enables PhysVLM to integrate physical reachability information into its reasoning process without compromising its general vision-language capabilities. To train and evaluate PhysVLM, we constructed a large-scale multi-robot dataset, Phys100K, and a challenging benchmark, EQA-phys, which includes tasks for six different robots in both simulated and real-world environments.

We summarize our contributions as follows:

- We propose a unified and robot-agnostic formulation, the Space-Physical Reachability Map (S-P Map), which abstracts robotic physical reachability in a way that is independent of specific robot configurations, promoting the learning of generalized features.
- We introduce PhysVLM, a vision-language model that integrates physical reachability with general vision-language capabilities via an additional feature encoder, improving task execution reliability.
- We release the EQA-phys benchmark, which includes six robots and 1.3K question-answer pairs, designed to test the model's understanding of physical reachability in simulated and real-world environments.
- Our model achieves a 14% improvement over GPT-4o on the EQA-phys benchmark. In embodied visual reasoning tasks on the RoboVQA-val and OpenEQA benchmarks, it outperforms advanced embodied VLMs such as RoboMamba and SpatialVLM. Furthermore, the S-P Map demonstrates strong compatibility with various VLMs, and integrating it into GPT-4o-mini results in a 7.1% performance improvement.

2. Related Work

2.1. VLMs in Robotics

Embodied Question Answering (EQA) tasks require agents to interact with environments to answer questions [10, 13, 38]. RoboVQA offers a large, diverse dataset for robotic visual question answering. 3D-VLA [44] integrates 3D perception with a generative world model for embodied reasoning, while SpatialVLM [5] enhances VLMs' spatial understanding using extensive 3D data.

Robot task planning involves sequencing subtasks to achieve goals [3, 30, 40]. Code as Policies (CaP) [20] employs OpenAI's Codex (code-davinci-002) to generate planning code. SayCan [2] combines the Pathways Language Model (PaLM) [7] with robotic affordances to create feasible action plans based on the robot's capabilities. However, these methods often assume all objects are within the robot's operational area, ignoring physical reachability and potentially leading to suboptimal or infeasible plans.

2.2. Understanding Physical Reachability

Recent studies use voxel grids with open-vocabulary detection models to assign task-specific attributes, enabling environmental constraint understanding. ReKep [15] generates keypoint proposals and constraints using voxel grids and VLMs, while VoxPoser [14] synthesizes robot trajectories by integrating OWL-ViT [26] and VLMs with voxel-based environment representations. However, these approaches focus on environment modeling without explicitly addressing the robot's physical reachability.

Explicitly representing reachable workspaces remains challenging. Reachability maps [41] model spatial capabilities, and occupancy grids [16] account for obstacles to ensure safe navigation. Additionally, methods like online model predictive control with offline workspace analysis

[31] and Reachability Expression-based Motion Planning (REMP) [12] address workspace constraints. Despite these advancements, integrating physical reachability into visual reasoning for complex embodied tasks is still limited, primarily due to the lack of large-scale datasets that include robotic physical parameters in VLM pretraining.

3. Method

PhysVLM is a large-scale vision-language model designed for visual reasoning that accounts for physical constraints in embodied tasks. As illustrated in Figure 2, PhysVLM integrates instruction text, visual input (RGB image), and an S-P Map abstracts the robotic physical reachability into a unified spatial representation. By combining these inputs, PhysVLM generates responses consistent with both the visual context and the robot's physical reachability, without being tied to specific robot configurations. The S-P Map is constructed using a unified physical reachability encoding method, which abstracts the physical parameters of various robots alongside their egocentric depth maps into a generalized form. This abstraction allows the model to generalize across different robots, addressing the challenge of learning and reasoning about physical reachability in a robot-agnostic manner.

In this section, we present the core components of PhysVLM. Section 3.1 describes the S-P Map encoding method, Section 3.2 describes the model architecture, and Section 3.3 discusses the training procedures.

3.1. S-P Map Encoding

As illustrated in Figure 2, we model the physical reachability of various robots using a unified approach that abstracts robot-specific parameters into a generalized spatial representation. This abstraction allows the model to focus on the spatial regions that are physically reachable, independent of the specific robot configuration.

$$\text{S-P Map} = F\left(\mathcal{P}_{\text{raw}}, \{\theta_i^{\min}, \theta_i^{\max}\}, \text{DH}, \mathbf{E}\right), \tag{1}$$

where $\mathcal{P}_{\text{raw}}$ denotes the raw point cloud data from the robot's RGB-D camera. $\{\theta_i^{\min}, \theta_i^{\max}\}$ represents the range of motion for each joint i. DH refers to the Denavit-Hartenberg parameters that describe the geometric structure of each joint, and $\mathbf{E}$ is the extrinsic calibration matrix that transforms coordinates from the camera to the robot's coordinate system. The function F maps these inputs to produce the S-P Map, which abstracts the robot's physical reachability into a spatial form that is independent of the specific robot configuration.

Consider a robot arm with n degrees of freedom, where each joint i has DH parameters $\{\theta_i, d_i, a_i, \alpha_i\}$ (θ_i is the joint angle, d_i is the offset along the z-axis, a_i is the link length, and α_i is the twist angle). The homogeneous transformation matrix for each joint is defined as:

$$\mathbf{T}_i = G(\theta_i, d_i, a_i, \alpha_i), \tag{2}$$

where G is the standard Denavit-Hartenberg transformation function. By multiplying the transformation matrices of all joints, we obtain the transformation matrix from the base frame to the end-effector frame:

$$\mathbf{T} = \mathbf{T}_1 \mathbf{T}_2 \dots \mathbf{T}_n. \tag{3}$$

To generate joint configurations, we sample the joint angles θ_i from their respective motion ranges $\theta_i \in [\theta_i^{\min}, \theta_i^{\max}]$, resulting in configurations $\{\theta_1, \theta_2, \dots, \theta_n\}$. By substituting these joint configurations into the forward kinematics equations, we compute the corresponding end-effector positions:

$$\mathcal{W}_{\text{voxel}} = \left\{ \mathbf{p} \;\middle|\; \mathbf{p} = \mathbf{T}(\theta_1, \theta_2, \dots, \theta_n) \cdot \mathbf{p}_0 \right\}, \tag{4}$$

where $\mathbf{p}_0$ is the origin point in the end-effector frame. We precompute these joint configurations offline, discretize the workspace into a voxel grid $\mathcal{W}_{\text{voxel}}$, and store it for efficient computation in subsequent steps.

Next, as shown in Figure 2, the robot's raw point cloud $\mathcal{P}_{\text{raw}}$ is captured from its egocentric RGB-D camera in the camera coordinate system and transformed into the robot's coordinate system using the extrinsic calibration matrix $\mathbf{E}$, resulting in the transformed point cloud $\mathcal{P}$:

$$\mathcal{P} = \mathbf{E} \cdot \mathcal{P}_{\text{raw}}. \tag{5}$$

To ensure physical feasibility, we perform a voxel grid lookup to determine whether each point in $\mathcal{P}$ lies within the precomputed reachable workspace $\mathcal{W}_{\text{voxel}}$:

$$\mathcal{P}_{\text{valid}} = \left\{ \mathbf{p} \in \mathcal{P} \;\middle|\; \mathbf{p} \in \mathcal{W}_{\text{voxel}} \right\}. \tag{6}$$

This step filters the point cloud to include only points within the robot's reachable workspace, abstracting the robot's physical reachability into a generalized spatial form.

Finally, we transform the valid point cloud $\mathcal{P}_{\text{valid}}$ back into the camera coordinate system and use the camera's intrinsic parameters to project these points onto the image plane. We then mark the regions that comply with physical reachability on the original depth map. For areas that are not reachable, we apply a gray mask and outline their boundaries. The resulting S-P Map clearly highlights regions that are beyond the robot's physical reach, providing a unified and abstracted representation of reachability that is independent of the specific robot configuration. This allows the model to focus on the spatial constraints of the task, without needing to account for the detailed physical parameters of each robot.

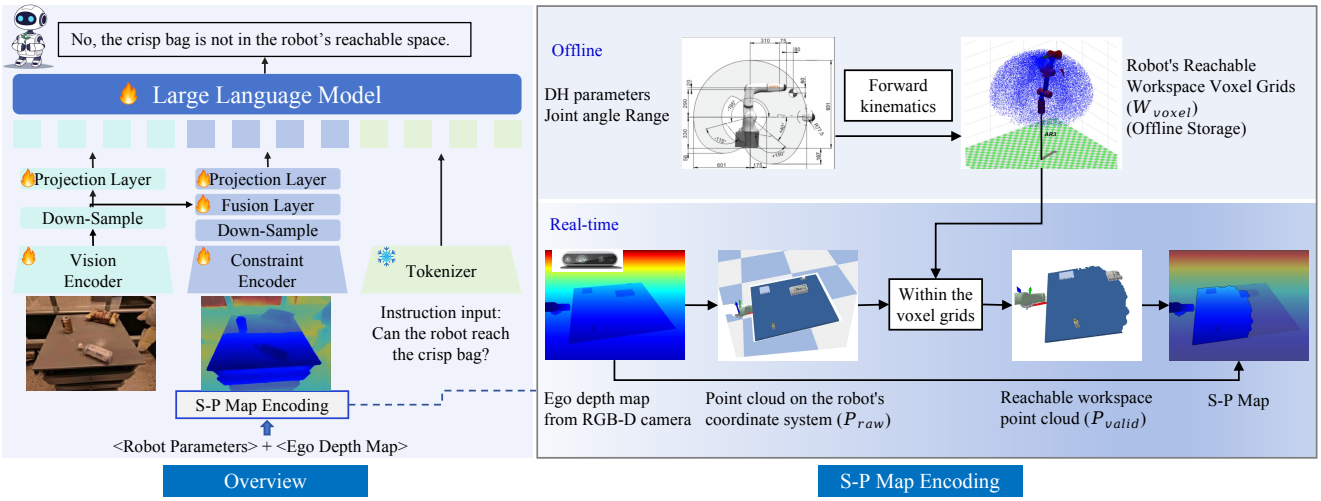


Figure 2. Overview of PhysVLM. Starting with robot parameters and an egocentric depth map, an S-P Map is generated via unified physical reachability encoding. Using the S-P Map, image, and instruction text, PhysVLM generates textual output considering the robot's physical reachability.

3.2. Model Architecture

To seamlessly integrate robotic physical reachability into PhysVLM while preserving its visual reasoning capabilities, we design a dual-branch architecture: one branch dedicated to vision processing and the other to physical reachability (see Figure 2). These branches operate independently, extracting features from their respective inputs, which are then fused and passed to a unified decoder for final reasoning and response generation.

The vision branch leverages a pre-trained Vision Transformer (ViT) [9], specifically the *SigLip-400M* model [42], to extract high-level visual features from egocentric images. To reduce computational overhead, a *Max Pooling* layer is applied, followed by a two-layer Multi-Layer Perceptron (MLP) that transforms the visual features into token representations suitable for multimodal fusion.

The physical reachability branch processes the S-P Map, which abstracts the robot's physical reachability into a generalized spatial form. This branch also utilizes the *SigLip-400M* model for feature extraction, followed by *Max Pooling* and a feature fusion layer. The fusion layer combines the visual and reachability features, and a two-layer MLP further refines these fused features into reachability-specific tokens.

For the language decoding, we employ the *Qwen-2.5-Instruct-3B* model [33, 39] as PhysVLM's large language model (LLM) decoder, using the *Qwen-2.5* tokenizer to process natural language instructions. The decoder integrates multimodal tokens from the vision branch, S-P Maps, and language inputs, generating coherent and contextually relevant textual responses that account for both visual and physical reachability information.

3.3. Training

Training Data Construction. The training data for PhysVLM consists of our Phys100K dataset and general VQA datasets, such as LLaVA-Pretrain, ShareGPT4V, and RoboVQA. Phys100K focuses on question-answering related to physical reachability, aggregating data from RoboVQA (20K samples), ScanNet [8] (10K samples), OpenX-Embodiment [27] (60K samples), and an additional 10K samples from PyBullet.

Depth maps are essential inputs for generating the S-P Map. For datasets lacking depth maps, we generate them using DepthAnything-v2. Additionally, we employ Grounding DINO [23] and SAM2 [34] to obtain 2D bounding boxes and segmentation results for objects in the images. In PyBullet, we simulate work scenarios using four robotic arms (UR5, FR5, CR5, and FRANKA) to collect RGB images, depth maps, and segmentation results.

For the PyBullet data in Phys100K, precise robot configurations can be obtained from the simulator. Therefore, we directly generate the S-P Map using the method described in Section 3.2, and labels indicating whether objects are reachable are obtained through simulated motion. The advantage of the S-P Map is that it abstracts physical reachability into a region-based representation, decoupling the learning process from specific robot configurations. This allows us to generate pseudo-labels for datasets without precise robot parameters. We approximate reachability using the segmentation results, marking regions and the objects within

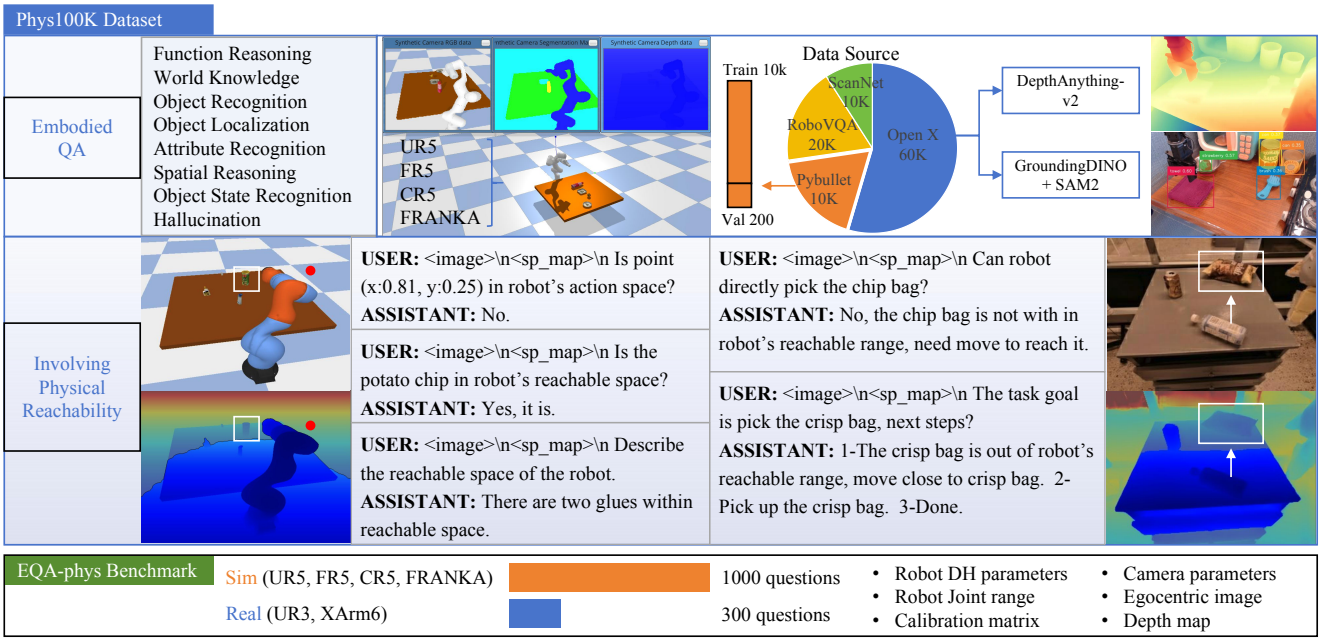


Figure 3. Details of the Phys100K Dataset and EQA-Phys Benchmark.

them as "reachable" or "unreachable" based on depth values. Next, we generate question-answer pairs for two main categories:

- **Embodied QA.** GPT-4 generates question-answer pairs for ScanNet and RoboVQA, covering categories include Function Reasoning, World Knowledge, Object Recognition, Object Localization, Attribute Recognition, Spatial Reasoning, Object State Recognition, and Hallucination (See Figure 3). Detailed prompts and examples are provided in the appendix to guide the generation process.
- **Tasks Involving Physical Reachability.** We use the "reachable" label with five fixed task templates to generate question-answer pairs, such as `"USER: <image>\n<sp_map>\n Is the [Object] in the robot's reachable space? ASSISTANT: Yes, it is."` Here, `[Object]` represents the relevant object category, while `<image>` and `<sp_map>` serve as placeholders for image patch tokens and S-P Map patch tokens, respectively. Figure 3 provides examples of question-answer pairs for each object category.

Training Pipeline. We adopt a two-stage training process to fully leverage the S-P Map and ensure PhysVLM generalizes across different robots. In the first stage, we align multimodal features using the LLaVA-Pretrain and OpenX-Embodiment datasets from Phys100K. This stage only trains the projection layers, allowing the model to build a foundational understanding of visual inputs and physical reachability, independent of specific robot configurations.

In the second stage, we unfreeze all parameters and train the entire model using data from Phys100K, ShareGPT4V, and RoboVQA. This stage enhances PhysVLM's ability to handle complex visual reasoning tasks with physical reachability constraints, ensuring the model can generalize across diverse environments and robots.

Implementation Details. PhysVLM is trained for 48 hours using eight A800 GPUs. The training process consists of two stages, each lasting one epoch. The batch size and learning rate are set at 128 and 1e-3 in the first stage, and 64 and 1e-5 in the second stage. The final model is PhysVLM-3B.

3.4. EQA-phys Benchmark

As illustrated in Figure 3, we introduce an embodied QA task focused on physical reachability, termed EQA-phys, which emphasizes QA tasks constrained by physical limitations. This benchmark includes a simulator dataset with 200 samples and 1,000 questions from the PyBullet validation set, as well as a zero-shot evaluation set based on real-world data from UR3 and XArm6 robots in two scenarios. The evaluation set contains 60 samples and 300 questions, all manually annotated by domain experts.

Table 1. Results of EQA-phys. Comparison of PhysVLM-3B (ours) with API-based VLMs and embodied VLMs.

		Real-world		Simulator				
		UR3	XArm6	UR5	FR5	CR5	FRANKA	ALL
API-based VLMs	GPT-4o-mini	54.3	56.0	49.4	55.4	54.6	47.1	52.8
	Claude-3.5	56.2	60.5	54.0	58.1	55.7	54.3	56.4
	GPT-4o	56.7	61.5	55.7	58.3	57.5	52.6	57.0
	GPT-4o-mini + **S-P Map**	$60.0_{\uparrow 5.7}$	$60.5_{\uparrow 4.5}$	$57.0_{\uparrow 7.6}$	$59.1_{\uparrow 3.7}$	$59.2_{\uparrow 4.6}$	$53.3_{\uparrow 6.2}$	$59.8_{\uparrow 7.0}$
	Claude-3.5 + **S-P Map**	$65.3_{\uparrow 9.1}$	$67.3_{\uparrow 6.8}$	$54.9_{\uparrow 0.9}$	$58.3_{\uparrow 0.2}$	$58.2_{\uparrow 2.5}$	$58.1_{\uparrow 3.8}$	$60.3_{\uparrow 3.4}$
	GPT-4o + **S-P Map**	$\mathbf{66.6}_{\uparrow 9.9}$	$\mathbf{68.1}_{\uparrow 6.6}$	$55.8_{\uparrow 0.1}$	$60.7_{\uparrow 1.4}$	$59.4_{\uparrow 1.9}$	$57.6_{\uparrow 5.0}$	$61.3_{\uparrow 4.1}$
Embodied VLMs	SpatialVLM	56.3	55.1	54.6	59.1	52.0	47.5	54.1
	SpatialBot	51.1	50.2	50.0	48.1	53.3	54.4	51.1
	PhysVLM-3B	64.1	63.0	**71.4**	**75.7**	**74.0**	**78.1**	**71.0**

4. Experiments

4.1. Experimental Setting

Tasks. We compare the performance of PhysVLM with other methods across three categories of tasks:

- **EQA-phys.** This benchmark tests the model's ability to integrate visual reasoning with robotic physical reachability. The real-robot component is used to assess PhysVLM's zero-shot generalization, highlighting its ability to handle unseen robots and environments.
- **Embodied QA.** We evaluate the model's general visual reasoning ability in embodied tasks using the OpenEQA [25] and RoboVQA-val [29] benchmarks.
- **Robot Task Planning.** For real-world tasks such as `"Pick A into B"`, we assess the model's ability to understand robotic physical reachability and generate reasonable task plans. As this study does not focus on robot control strategies, we use the natural language planning approach from [29].

Baselines. We compare our model to several baselines, including API-accessible VLMs such as Claude 3.5 [28], GPT-4o-mini [1], and GPT-4o [1], as well as embodied VLMs like SpatialVLM [5], SpatialBot [4], 3D-VLA [44], and RoboMamba [22]. SpatialVLM and SpatialBot both use the 3B version, which has a similar parameter count to our model. Since the executable versions of 3D-VLA and RoboMamba are unavailable, we compare their reported results on RoboVQA-val.

Evaluation Metrics. For tasks involving physical reachability, we use LLM scoring, following the approach used in existing studies [24, 25]. Assigning 5 points for completely correct responses and 1 point for incorrect responses, we calculate the average score and express it as a percentage. For Embodied QA, we follow the benchmark settings of the corresponding datasets [25, 29]. For task planning, each task type is executed 10 times, and the average success rate serves as the evaluation metric.

4.2. Results on EQA-phys

Table 1 shows the results on EQA-phys. Neither API-based nor embodied VLMs can handle the robot's parameter constraints, resulting in suboptimal outputs with scores around 55%. In contrast, our model successfully completes visual reasoning tasks involving physical reachability, obtaining an average score of 71%. As discussed, these tasks require the model to perform visual reasoning based on an understanding of the robotic physical reachability. The model can only effectively accomplish these tasks by truly understanding robotic physical reachability.

The results in Table 1 demonstrate that prompting API-based VLMs, such as GPT-4o, with the S-P Map (detailed in 4) significantly enhances their performance. This improvement stems from the S-P Map's ability to abstract physical reachability into a robot-agnostic representation, enabling VLMs to reason about physical constraints that would otherwise be beyond their capabilities. By decoupling reachability from specific robot parameters, the S-P Map facilitates generalization across diverse environments, allowing models to better comprehend physical reachability, even in previously unseen scenarios.

Table 1 also demonstrates that PhysVLM-3B achieved scores of over 63% in zero-shot evaluations for UR3 and XArm6 robots, despite operating in new environments with different robot parameters. This performance is attributed to two key factors: (1) the S-P Map abstracts various robot parameters into a unified, transferable representation of physical reachability, and (2) the model's independent visual and constraint encoding branches allow it to learn generalizable visual features from diverse image-text data, enabling effective reasoning in novel environments.

Table 2. Embodied QA results on the RoboVQA-val set, comparison of PhysVLM (ours) with existing methods. An asterisk (*) indicates models not pre-trained on the RoboVQA dataset.

	BLEU1	BLEU2	BLEU3	BLEU4
SpatialVLM*	5.1	3.0	1.9	1.2
SpatialBot*	12.4	9.3	8.0	7.2
3D-VLA	48.3	38.5	31.7	26.8
RoboMamba	54.9	44.2	39.5	36.3
PhysVLM-3B	**65.3**	**62.4**	**50.9**	**43.5**

Table 3. Embodied QA results on the OpenEQA benchmark, comparison of PhysVLM (ours) with existing methods. An asterisk (*) indicates that only the first 200 samples were tested due to API limitations.

	EM-EQA (ScanNet)	EM-EQA (HM3D)	ALL
SpatialVLM	42.9	44.3	43.8
SpatialBot	45.3	51.0	49.1
GPT4V	57.4	51.3	55.3
GPT-4o*	**68.2**	**65.2**	**66.7**
PhysVLM-3B	60.7	51.2	57.4

4.3. Results on Embodied QA

We demonstrate our model's effectiveness in handling general embodied visual reasoning tasks. Additionally, we show that incorporating an understanding of physical constraints does not diminish its general visual reasoning capabilities. We compare our model with state-of-the-art embodied VLMs (see Tables 2 and 3). Our model achieves the best performance on the RoboVQA-val benchmark, surpassing other models by 7.2% in BLEU-4. On the OpenEQA benchmark, our model outperforms existing embodied VLMs and GPT-4V, ranking second, behind only GPT-4o.

4.4. Results on Robot Task Planning

Table 4 presents the performance of PhysVLM and baseline models on real-world task planning scenarios. When all objects are within the robot's physical reach, PhysVLM performs similarly to other models, as directly grabbing or placing the objects succeeds. However, when some objects are outside the physical reach, the model must suggest that the robot move closer before grabbing or placing them. In these cases, our model performs exceptionally well, whereas the task success rates of other models decline significantly. This is attributed to our model's understanding of robotic physical reachability and its ability to incorporate this understanding in task planning.

Table 4. Task planning results. Comparison of PhysVLM (ours) with other VLMs.

	All objects in range	Part objects in range
GPT-4o-mini	70.5	23.2
Claude-3.5	73.6	32.1
GPT-4o	**75.9**	35.8
SpatialVLM	64.4	21.5
SpatialBot	65.6	25.3
PhysVLM-3B	69.2	**48.4**

Table 5. Ablation study on the S-P Map. We compare the constraint encoder's performance when using the S-P Map, replacing it with a Depth Map, or providing no input.

ID	S-P Map	Depth Map	EQA-phys Real	EQA-phys Sim
1	✓		63.5	74.8
2		✓	58.1	62.4
3			54.2	58.8

4.5. Ablation Study

In this section, we conduct ablation studies to evaluate the contribution of each component in PhysVLM. We report the average LLM score on tasks involving physical reachability.

The effectiveness of S-P Map. To demonstrate the S-P Map's contribution, we compare results obtained with and without its input. As shown in Table 5, Experiments 1 and 3 demonstrate that omitting the S-P Map leads to a significant performance decrease for both zero-shot real-world robots and simulators. Specifically, the overall average score drops by 16% in simulation results and 9.3% in real-world robot evaluations. Without the S-P Map input, the model struggles to handle the robotic physical reachability.

Additionally, Experiments 1 and 2 demonstrate that replacing the S-P Map with a Depth Map significantly degrades the model's performance on zero-shot tasks. Since the Depth Map does not accurately represent robotic physical reachability, the model cannot rely solely on depth information to understand it.

Effectiveness of an additional feature encoder. To demonstrate the effectiveness of the model architecture, we compare the performance of a feature encoder that shares weights with the visual feature encoder for the S-P Map. In these experiments, we evaluate the average scores on both the EQA-phys and OpenEQA benchmarks. The results in

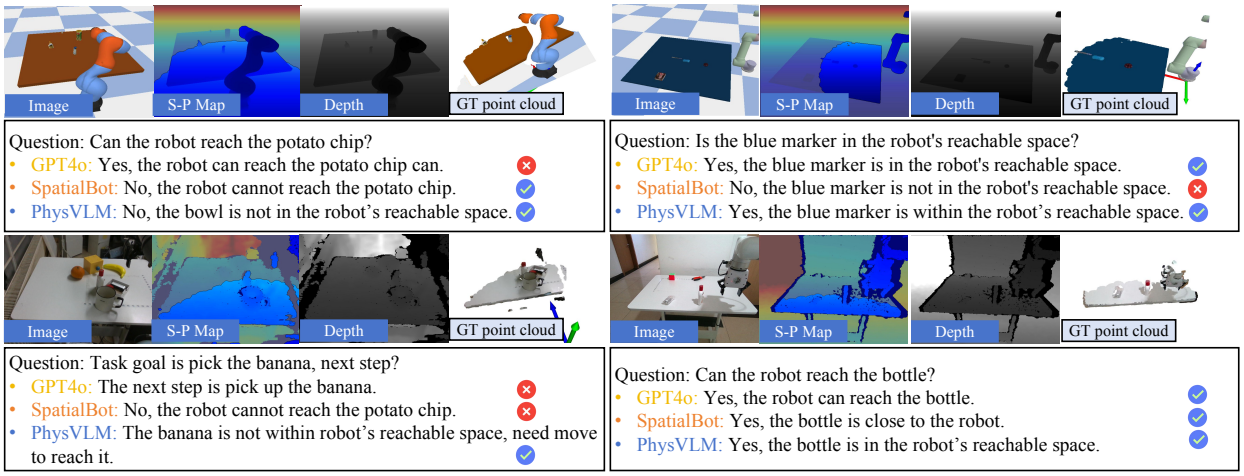


Figure 4. Visual comparison of PhysVLM (ours), GPT-4o, and SpatialBot.

Table 6. Ablation study on the effectiveness of an additional feature encoder. `Share` indicates shared network and weights with the visual feature encoder.

	EQA-PHYS	OPENEQA
INDEPENDENT	71.0	57.4
SHARE	68.2	56.5

Table 6 show that sharing the feature encoder not only decreases performance on EQA-phys but also impairs general visual reasoning capabilities. This is because S-P Map features differ from images, and the training data contains significantly more image-text pairs than S-P Map data.

Effectiveness of training data. To evaluate the effectiveness of Phys100K, we conduct experiments by selectively removing data from various sources in Phys100K. As shown in Table 7, removing data from PyBullet or other embodied datasets leads to a reduction in overall performance, highlighting the critical role of each data component in the model's performance.

Table 7. Ablation study on the effectiveness of training data.

PART OF PHYS100K	EQA-PHYS REAL	EQA-PHYS SIM
ALL	63.5	74.8
W/O PYBULLET	62.1	65.4
W/O OTHER DATASETS	58.6	71.5

4.6. Qualitative Results

Figure 4 compares our method with SpatialBot and GPT-4o. SpatialBot uses depth maps and images, while GPT-4o uses standard images. Both struggle with tasks requiring physical reachability, causing visual reasoning errors. In contrast, our approach delivers accurate results. Additionally, incorporating the S-P Map into GPT-4o improves its handling of physical reachability and response accuracy.

5. Conclusion

We introduce PhysVLM, a VLM that incorporates physical reachability into visual reasoning for robotic tasks. The S-P Map provides a unified representation of robotic reachability, facilitating the learning of generalizable features. PhysVLM extends traditional VLMs by adding a physical reachability encoder, enabling the simultaneous processing of visual, reachability, and textual information. Additionally, we present EQA-phys, a benchmark for evaluating embodied QA tasks involving physical reachability. Our experiments show that PhysVLM outperforms existing models, achieving a 14% higher score than GPT-4o on EQA-phys. A limitation is its reduced zero-shot performance on real robots compared to simulations, likely due to the domain gap. Future work will focus on expanding datasets, enhancing real-world performance, and improving the understanding of physical accessibility in vision-language-action models. PhysVLM's reachability awareness supports safer and more reliable robotic decision-making in industrial and assistive settings, while its unified representation ensures cross-platform adaptability for real-world deployment, bridging crucial gaps between environmental perception and actionable robotic intelligence.

6. Acknowledgments

This work was supported by National Key R&D Program of China under Grant No.2022ZD0160601, 2022YFB4300400, and 2018B030322016, National Natural Science Foundation of China under Grants 62176254, 62276260, and U1701266.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[2] Michael Ahn, Brohan, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2

[3] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2

[4] Wenxiao Cai, Yaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *arXiv preprint arXiv:2406.13642*, 2024. 6

[5] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024. 2, 6

[6] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 2

[7] Aakanksha Chowdhery, Narang, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 2

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 4

[9] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[10] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 2

[11] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, et al. Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16238–16250, 2024. 2

[12] Xiaofeng Gao, Luyao Yuan, Tianmin Shu, Hongjing Lu, and Song-Chun Zhu. Show me what you can do: Capability calibration on reachable workspace for human-robot collaboration. *IEEE Robotics and Automation Letters*, 7(2):2644–2651, 2022. 3

[13] Yining Hong, Zishuo Zheng, Peihao Chen, Yian Wang, Junyan Li, and Chuang Gan. Multiply: A multisensory object-centric embodied large language model in 3d world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26406–26416, 2024. 2

[14] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 2

[15] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. 2

[16] Lorenzo Jamone, Martim Brandao, Lorenzo Natale, Kenji Hashimoto, Giulio Sandini, and Atsuo Takanishi. Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching. *Robotics and Autonomous Systems*, 62(4):556–567, 2014. 2

[17] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024. 2

[18] Xiaohan Lei, Min Wang, Wengang Zhou, Li Li, and Houqiang Li. Instance-aware exploration-verification-exploitation for instance imagegoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16329–16339, 2024. 2

[19] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 2

[20] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023. 2

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[22] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *arXiv preprint arXiv:2406.04339*, 2024. 6

[23] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with

grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 4

[24] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pages 216–233. Springer, 2025. 6

[25] Arjun Majumdar, Ajay, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498, 2024. 2, 6

[26] Matthias Minderer, Alexey Gritsenko, Austin Stone, et al. Simple open-vocabulary object detection with vision transformers, 2022. 2

[27] Abby O'Neill, Rehman, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024. 4

[28] Alec Radford, Kim, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[29] Pierre Sermanet, Ding, et al. Robovqa: Multimodal long-horizon reasoning for robotics. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 645–652. IEEE, 2024. 6

[30] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 2

[31] Chen Song and Darwin Lau. Workspace-based model predictive control for cable-driven robots. *IEEE Transactions on Robotics*, 38(4):2577–2596, 2022. 2, 3

[32] Giovanni Sutanto, Austin Wang, Yixin Lin, Mustafa Mukadam, Gaurav Sukhatme, Akshara Rai, and Franziska Meier. Encoding physical constraints in differentiable newton-euler algorithm. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, pages 804–813. PMLR, 2020. 2

[33] Qwen Team. qwen2.5, 2024. 4

[34] Dominique Thomas, Rodney Rothstein, Nathan Rosenberg, and Yolande Surdin-Kerjan. Sam2 encodes the second methionine s-adenosyl transferase in saccharomyces cerevisiae: physiology and regulation of both enzymes. *Molecular and Cellular Biology*, 8(12):5132–5139, 1988. 4

[35] Yunsheng Tian, Karl DD Willis, Bassel Al Omari, Jieliang Luo, Pingchuan Ma, Yichen Li, Farhad Javid, Edward Gu, Joshua Jacob, Shinjiro Sueda, et al. Asap: Automated sequence planning for complex robotic assembly with physical feasibility. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4380–4386. IEEE, 2024. 2

[36] An Dinh Vuong, Minh Nhat Vu, Baoru Huang, Nghia Nguyen, Hieu Le, Thieu Vo, and Anh Nguyen. Language-driven grasp detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17902–17912, 2024. 2

[37] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2

[38] Yu Wu, Lu Jiang, and Yi Yang. Revisiting embodiedqa: A simple baseline and beyond. *IEEE Transactions on Image Processing*, 29:3984–3992, 2020. 2

[39] An Yang, Baosong Yang, Binyuan Hui, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 4

[40] Yijun Yang, Tianyi Zhou, Kanxue Li, Dapeng Tao, Lusong Li, Li Shen, Xiaodong He, Jing Jiang, and Yuhui Shi. Embodied multi-modal agent trained by an llm from a parallel textworld. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26275–26285, 2024. 2

[41] Franziska Zacharias, Christoph Borst, and Gerd Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3229–3236. Ieee, 2007. 2

[42] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. 4

[43] Ganlong Zhao, Guanbin Li, Weikai Chen, and Yizhou Yu. Over-nav: Elevating iterative vision-and-language navigation with open-vocabulary detection and structured representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16296–16306, 2024. 2

[44] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 2, 6