SPARRTA: A SYNTHETIC BENCHMARK FOR EVALUATING SPATIAL INTELLIGENCE IN VISUAL FOUNDATION MODELS

Turhan Can Kargin[1,2*], **Wojciech Jasiński**[1,3], **Adam Pardyl**[1,2,4], **Bartosz Zieliński**[1], **Marcin Przewięźlikowski**[1,2]
[1]Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland
[2]Doctoral School of Exact and Natural Sciences, Jagiellonian University, Kraków, Poland
[3]AGH University of Krakow, Kraków, Poland
[4]IDEAS NCBR, Warsaw, Poland
[*]turhancan.kargin@doctoral.uj.edu.pl

ABSTRACT

Visual Foundation Models (VFMs), such as DINO and CLIP, excel in semantic understanding of images but exhibit limited spatial reasoning capabilities, which limits their applicability to embodied systems. As a result, recent work incorporates some 3D tasks (such as depth estimation) into VFM training. However, VFM performance remains inconsistent across other spatial tasks, raising the question of whether these models truly have spatial awareness or overfit to specific 3D objectives. To address this question, we introduce the Spatial Relation Recognition Task (SpaRRTa) benchmark, which evaluates the ability of VFMs to identify relative positions of objects in the image. Unlike traditional 3D objectives that focus on precise metric prediction (e.g., surface normal estimation), SpaRRTa probes a fundamental capability underpinning more advanced forms of human-like spatial understanding. SpaRRTa generates an arbitrary number of photorealistic images with diverse scenes and fully controllable object arrangements, along with freely accessible spatial annotations. Evaluating a range of state-of-the-art VFMs, we reveal significant disparities between their spatial reasoning abilities. Through our analysis, we provide insights into the mechanisms that support or hinder spatial awareness in modern VFMs. We hope that SpaRRTa will serve as a useful tool for guiding the development of future spatially aware visual models.

1 INTRODUCTION

Visual Foundation Models (VFMs) such as DINO (Caron et al., 2021; Oquab et al., 2023; Oriane Siméoni et al., 2025) and CLIP (Radford et al., 2021) demonstrate remarkable performance in semantic visual understanding. Hence, they are widely used for semantic perception tasks including image classification (Deng et al., 2009), object recognition (Lin et al., 2014), multimodal learning (Bai et al., 2025; Li et al., 2025), and scene understanding, with applications in domains such as retail (Srivastava & Wu, 2025) and medical imaging (Sroka-Oleksiak et al., 2025). As these models move beyond static perception and are increasingly deployed in embodied and interactive settings, their role expands from recognizing visual content to supporting downstream decision-making and interaction (Venkataramanan et al., 2024; Bardes et al., 2024).

However, effective deployment of VFMs in physical environments requires more than just semantic recognition. To act and navigate, embodied agents must be aware of the spatial properties of their surroundings, and reason about three-dimensional spatial relations between objects, such as their relative position and perspective (e.g., `the truck is to the left of the tree from the human's point of view`) (Deitke et al., 2020; Linsley et al., 2025). As a result, recent years have seen an increased interest in evaluating VFMs in terms of their performance in tasks that require spatial awareness, such as depth prediction, camera pose estimation, and surface normal estimation (Banani et al., 2024; Chen et al., 2024). Moreover, recent VFM training objectives increasingly incorporate geometric cues (Weinzaepfel et al., 2022a;b; Zhu et al., 2024; Wang et al., 2025). While this has yielded improvements in tasks most related to cues used in pretraining, the overall improvements in 3D perception remain inconsistent (Banani et al., 2024; Chen et al.,

2024). This raises a fundamental question: do contemporary VFMs acquire a general notion of spatial awareness, or do they instead overfit to specific, quantifiable geometric objectives without capturing abstractions that underpin human spatial understanding?

To address this question, we introduce Spatial Relation Recognition Task (SpaRRTa), a benchmark designed to evaluate abstract spatial awareness in visual representations. Rather than focusing on precise 3D predictions such as distances or surface normals, SpaRRTa targets relational, human-like spatial concepts, including relative position between objects (see Figure 1). At a conceptual level, identifying relative object positions is a fundamental capability underlying more advanced forms of spatial reasoning (Johnson & Moore, 2020; Jechura, 2006; Burgess et al., 2002), providing a basis for detailed use cases like navigation, planning, and object manipulation. In practice, SpaRRTa evaluates the efficacy of decoding spatial relation information of given objects from frozen VFM latent representations. By decoupling relational spatial reasoning from precise geometric predictions, SpaRRTa provides a principled way to assess whether VFMs represent transferable spatial structure rather than task-specific geometric shortcuts.

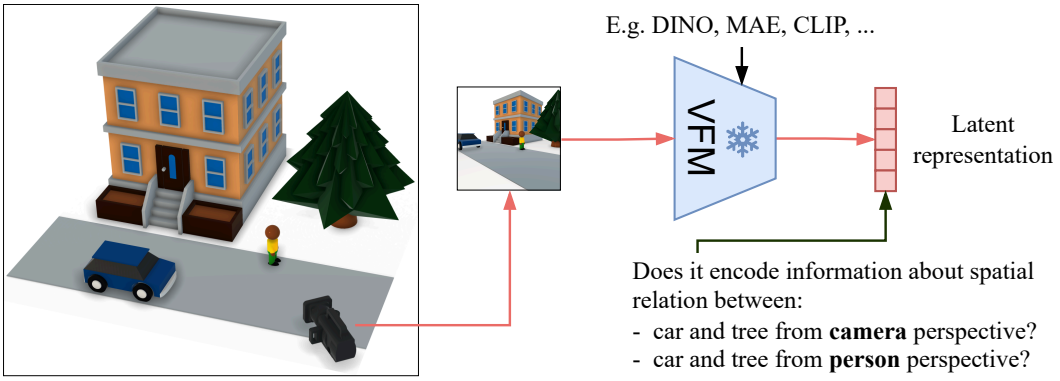


Figure 1: Effective spatial reasoning requires awareness of spatial relations between visible objects. To analyze such spatial awareness of Visual Foundation Models (VFMs), we introduce Spatial Relation Recognition Task (SpaRRTa). We generate images showing different spatial layouts of several objects, encode them with VFMs, and probe whether their latent representations reliably encode the information about the spatial relations between the objects from a selected perspective.

SpaRRTa is built with Unreal Engine 5 (Epic Games, 2025b), enabling the generation of photorealistic scenes with a wide range of object categories, layouts, and background contexts (see Figure 2). This synthetic setup provides full control over object placement and precise annotation of spatial relations, allowing evaluation at a scale and diversity that would be prohibitively expensive to obtain from real-world data. By systematically varying object configurations and viewpoints across diverse environments, SpaRRTa enables a controlled and robust evaluation of spatial relation understanding in realistic visual settings, remaining in-distribution for common VFMs trained on natural images.

Using SpaRRTa, we evaluate a broad set of VFMs trained under different supervision regimes and analyze how they encode spatial relations. Our results show that self-supervised VFMs consistently outperform supervised and vision–language models, yet even the strongest models exhibit limitations in perspective-dependent and cluttered settings. By comparing different probing mechanisms, we further reveal that spatial information is primarily stored at the patch level and is largely obscured by global pooling, highlighting the importance of structured aggregation for uncovering spatial knowledge. We hope that SpaRRTa will serve as a useful diagnostic tool for studying spatial representations and for guiding the development of future models with stronger spatial awareness, which is essential for embodied and interactive applications.

Our contributions can be summarized as follows. We introduce SpaRRTa, a photorealistic, synthetically generated evaluation environment built on Unreal Engine 5 for probing spatial reasoning in VFMs. We define two standardized challenge sets, SpaRRTa-**ego** (camera-centric) and SpaRRTa-**allo** (allocentric/perspective-taking), designed to disentangle spatial logic from semantic recognition. We benchmark varying families of VFMs (MIM, Joint-Embedding, Supervised, Vision-Language) and characterize which models provide spatial information in their representations and how these representations are structured.

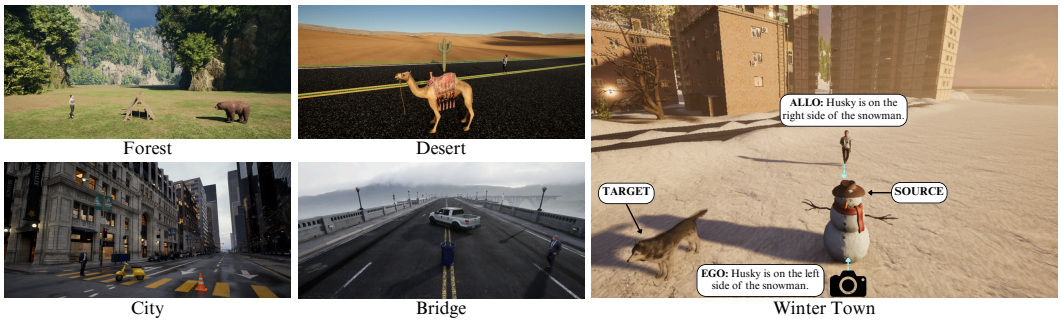


Figure 2: SpaRRTa evaluates the efficacy of encoding the spatial relations between objects in the image by VFM, either from the camera (**Ego**centric) or an arbitrary object's (**Allo**centric) perspective (see the right-most image). We use Unreal Engine 5 to produce photorealistic test images that are in-distribution for contemporary VFMs, using a variety of scene environments and object assets.

2 Methodology for Evaluating Spatial Awareness

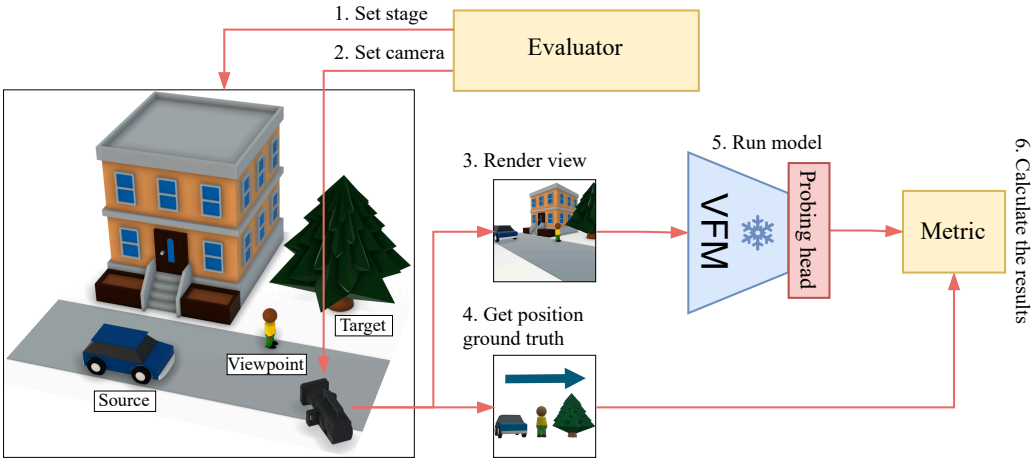
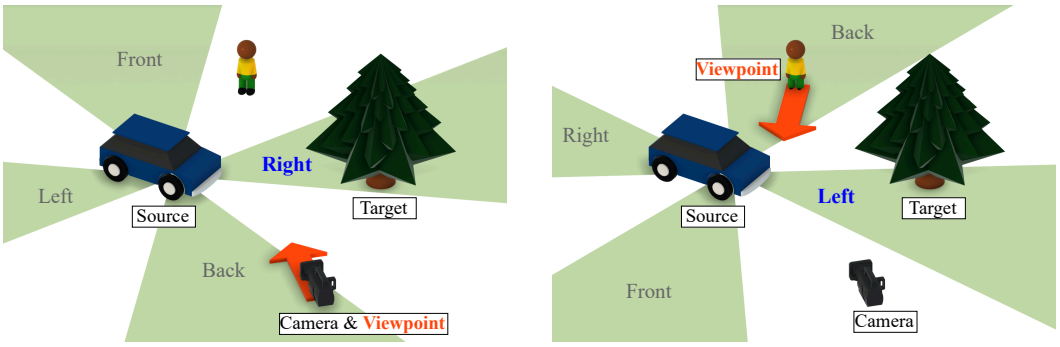


Figure 3: **Evaluation pipeline:** The SpaRRTa evaluation setup consists of a data generation system built on Unreal Engine 5. The pipeline begins with an evaluation controller which sets the scene in a precisely controlled manner using a diverse set of realistic assets **(1)**, and places the camera **(2)** for capture. Next, a near-photorealistic image is rendered by Unreal **(3)**, and ground truth positional information is acquired **(4)**. The image is then passed trough evaluated VFM, and resulting representation is fed to the probing head for the positional relation prediction **(5)**. Finally, the accuracy is computed **(6)**.

In this section, we introduce our SpaRRTa methodology for evaluating abstract spatial awareness of visual models. From a high level perspective, SpaRRTa measures the accuracy of recovering the relative spatial configuration of objects placed in a scene based on an image representation. We utilize a two-stage evaluation framework comprising synthetic dataset generation and offline model probing. The entire pipeline, illustrated in Figure 3.

2.1 Spatial Relation Recognition Task

To evaluate spatial awareness of visual models, we define a family of tasks where the objective is to determine the relative spatial relation between objects present in a single image. Given two objects, denoted as **source** and **target**, the task is to classify the direction from the source to the target. We formulate the prediction as a four-way classification problem with discrete labels, i.e. `{left, right, front, back}`. A key aspect of spatial direction is that it is not defined in absolute image coordinates, but with respect to a chosen reference **viewpoint**. In this work, we



(a) Egocentric Spatial Relation Recognition Task (**SpaRRTa-ego**). In this variant, the camera position defines the viewpoint reference for the spatial directions. The correct relative direction from the **car** (source) to the **tree** (target) is **right**.

(b) Allocentric Spatial Relation Recognition Task (**SpaRRTa-allo**). In this variant, the spatial directions from the reference are defined by a the **viewpoint** object (human). The correct relative direction from the **car** (source) to the **tree** (target) is **left**.

Figure 4: Visualization of the Spatial Relation Recognition Task (SpaRRTa). The task is to predict the spatial relationship between the **source** (i.e. car) and **target** (tree) objects with respect to a given **viewpoint** (i.e. the camera in the egocentric variant **(a)**, and human in the allocentric variant **(b)**) To eliminate ambiguity of the tasks, we generate scenes where the spatial relationships of objects are clearly recognizable – the green area denotes valid locations for target placement.

define a viewpoint as the observer that determines how spatial relations are perceived. We formulate the **egocentric** and **allocentric** variants of the task that differ in the viewpoint used to express the direction. We visualize both variants in Figure 4, and describe them below.

Egocentric variant (Figure 4a) defines the camera as the viewpoint. The question is therefore where the target lies relative to the source as seen from the camera. This variant captures spatial relations directly observable from the input image.

Allocentric variant (Figure 4b) defines the selected third object (in our case, the human) as a viewpoint. In this scenario, the spatial relation observed by the camera differs from the relation observed by the viewpoint object. Consequently, the allocentric variant is more challenging, as it requires the model to decouple the spatial relation from its own visual input. This demands an implicit viewpoint transformation known as perspective-taking. The model must ignore the apparent positions of objects in the image and infer their geometric arrangement relative to the third object's perspective.

2.2 Data generation pipeline

In contrast to standard vision benchmarks such as ImageNet (Deng et al., 2009), which rely on images gathered from the Internet, SpaRRTa uses a modern rendering engine to create near-photorealistic images for evaluation. This allows us to retain full control over image content while remaining in-distribution for the common VFMs.

Renderer. As rendering engine, we chose Unreal Engine 5 (Epic Games, 2025b) as it provides state-of-the-art rendering quality and performance through real-time ray tracing with dynamic global illumination and reflections, as well as automatically adjusted level of detail. The engine allows for great extensibility and has an open-source codebase, which allows us to extend it for the needs of this study, providing interfaces for both automated environment map creation, image capture, and ground truth acquisition. Moreover, Unreal Engines' vast online marketplace of free graphical assets, many of which were created using photogrammetry, provides a great source of test evaluation scenario diversity while assuring high-fidelity, realistic results. Finally, it requires relatively modest hardware, such as a consumer-grade GPU, while supporting deep learning dedicated solutions, provided with the Vulkan library.

Scenario generator. The scenario generation part of the pipeline is responsible for procedural generation of diverse testing environments using the official Unreal Engine Editor Python API (Epic Games, 2025a). First, it randomly selects test objects from a map-specific asset list and places them on the ground at positions sampled from a Gaussian distribution. Next, the camera position is sampled from a uniform distribution over an area surrounding the map center and oriented toward the placed objects. Finally, the system verifies that all assets are correctly loaded and that the scene is ready for image capture. Additional implementation details are provided in Section C.2.

Geometric ambiguity control. A fundamental challenge in spatial classification lies in defining precise boundaries between semantic classes (e.g., determining the exact threshold where *Front* transitions to *Left*). To eliminate label noise and ensure mathematical rigor, we implement a strict procedural rejection sampling strategy. We define ambiguity zones as $\pm 15^\circ$ angular regions centered along the diagonals $(45^\circ, 135^\circ, 225^\circ, 315^\circ)$ relative to the viewpoint's forward direction. These diagonals serve as the dividing lines between the four main directions. The pipeline computes the precise angular relationship between the target and source objects, and automatically discards any scenario in which the target falls within these ambiguity zones. This guarantees that all retained samples possess indisputable ground-truth labels, ensuring that evaluation metrics reflect genuine spatial reasoning capabilities rather than boundary confusion. We provide a comprehensive description and visualization of these exclusion zones in Section C.1.

Image and ground truth capture. View capture is performed using the adapted UnrealCV library (Qiu et al., 2017). We capture both the ray-traced RGB image and a ground-truth segmentation mask. The mask is used together with the 3D coordinates of spawned objects to assert the validity of the scenario generation and calculate the ground-truth label for the view (Front, Back, Left, or Right). All captured images and metadata are stored for model evaluation. Detailed statistics on the dataset size are provided in Section C.1.

Figure 5: **SpaRRTa Asset Library.** SpaRRTa constructs test examples using a curated set of diverse, high-fidelity 3D assets selected based on common ImageNet classes (Deng et al., 2009).

2.3 Evaluation Environments

To ensure that SpaRRTa measures robust spatial reasoning rather than overfitting to specific visual domains, we construct a diverse suite of five distinct high-fidelity environments (see Figure 2). These environments range from organic, unstructured landscapes to dense, structured urban settings, challenging the model to generalize geometric understanding across vastly different textures, lighting conditions, and contextual layouts.

Asset Classes. To minimize semantic ambiguity and prioritize spatial reasoning performance, we used a fixed set of distinguishable objects for each environment. Each scene includes a human object who serves as the consistent viewpoint for the allocentric tasks. The source and target objects are selected to be semantically coherent with their respective environments (see Figure 2). This selection ensures that the objects are naturally occurring within the scene context across all evaluation trials.

Crucially, our asset selection strategy prioritized objects that are not only naturally occurring within their respective scene contexts (e.g., Camels in a Desert, Taxi in a City) but are also statistically prominent in standard pre-training datasets like ImageNet (Ridnik et al., 2021). This way, we ensure that any failure in spatial reasoning is due to geometric understanding rather than a lack of semantic recognition by selecting assets that align with common ImageNet super-categories—(1) *Animals*, (2) *Everyday Objects*, (3) *Nature*, and (4) *Humans*. We provide a catalog of all 3D assets, including their sources in Table 4 and Figure 5.

Environmental diversity. We leverage high-quality environments from the Unreal Engine ecosystem to create photorealistic scenes that mimic real-world complexity. The benchmark consists of the following five distinct types of environment: Forest, Desert, Winter Town, Bridge, and City. More details about environments are provided in Section C.1.

Model	Pre-training Objective	Supervision	Training Dataset
Joint-Embedding (JEA)			
DINO (Caron et al., 2021)	Contrastive / Distillation	Self-Sup.	ImageNet-1k
DINO-v2[†] (Oquab et al., 2023)	DINO + iBOT	Self-Sup.	LVD-142M
DINO-v2 (+reg)[‡] (Darcet et al., 2023)	DINO-v2 w/ Register Tokens	Self-Sup.	LVD-142M
DINOv3 (Oriane Siméoni et al., 2025)	DINO + iBOT	Self-Sup.	LVD-1689M
Masked Image Modeling (MIM)			
MAE (He et al., 2022)	Pixel Reconstruction	Self-Sup.	ImageNet-1k
MaskFeat (Wei et al., 2023)	HOG Feature Prediction	Self-Sup.	ImageNet-1k
SPA (Zhu et al., 2024)	Masked Volumetric Neural Rendering	Multi-View Self-Sup.	ScanNet, Hypersim, S3DIS
CroCo (Weinzaepfel et al., 2022a)	Cross-View Completion	Self-Sup.	Habitat
CroCov2 (Weinzaepfel et al., 2022b)	Cross-View Completion	Self-Sup.	ARKitScenes, MegaDepth, ...
Supervised & Weakly Supervised			
VGGT[*] (Wang et al., 2025)	Multi-Task 3D Regression (Cam, Depth, Tracks)	3D Sup.	Co3D, MegaDepth, etc.
DeiT (Touvron et al., 2022)	Classification + Distillation	Label Sup.	ImageNet-1k
CLIP (Radford et al., 2021)	Image-Text Contrastive	Text-Image Pairs	Web Image-Text (WIT)

Note: All models utilize a ViT-B/16 backbone unless otherwise marked:
[†]ViT-B/14 [‡]ViT-B/14 & ViT-L/14 [*]ViT-L/14

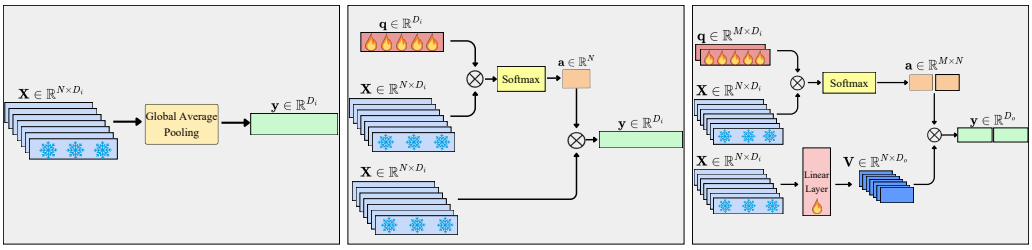
Table 1: **Evaluated Visual Foundation Models (VFMs).** We consider a range of visual backbones, specifically focusing on Self-supervised methods spanning various forms of supervision (JEA, MIM), as well as prominent supervised models.

3 EXPERIMENTS

In this section, we describe our evaluation of the current prominent Visual Foundation Models (VFMs) on the Spatial Relation Recognition Task (SpaRRTa). Section 3.1 details the choice of VFMs, as well as the methodology of adapting them for solving SpaRRTa. In Section 3.2, we conduct the evaluation of VFMs on SpaRRTa, revealing which models encode spatial relations.

3.1 EXPERIMENTAL SETUP

Evaluated Visual Foundation Models (VFMs). To systematically evaluate the emergence of spatial reasoning, we curate a diverse suite of VFMs spanning distinct learning paradigms, as detailed in Table 1. Our selection prioritizes Self-Supervised Learning (SSL) methods, which we categorize into Joint-Embedding Architectures (JEA) (Caron et al., 2021; Oquab et al., 2023; Oriane Siméoni et al., 2025), and Masked Image Modeling (MIM) (He et al., 2022; Wei et al., 2023; Zhu et al., 2024; Weinzaepfel et al., 2022a;b). Moreover, we compare to several VFMs prominent in the community, trained in a supervised and weakly supervised manner (Wang et al., 2025; Touvron et al., 2022; Radford et al., 2021). To ensure a fair comparison of learned representations, we evaluate all models using a ViT-Base backbone with an input resolution standardized to 224×224. We note two necessary exceptions regarding model size. VGGT (Wang et al., 2025) provides only ViT-Large checkpoints. We explicitly include VGGT, which is trained with direct 3D supervision (cameras, depth, tracks), to serve as a 3D-native baseline. This allows us to determine if explicit 3D training signals are strictly necessary for the abstract spatial reasoning tasks in SpaRRTa, or if such capabilities can emerge from 2D self-supervision alone. For a fair comparison on the ViT-L backbone,



(a) Linear Probing with GAP (b) AbMILP (c) Efficient Probing

Figure 6: **Architectural comparison of probing protocols on VFMs.** A frozen VFM backbone processes the input image. Then we apply three different transformations to form a global representation of the image, on top of which we train a linear probe for spatial relation prediction: **Linear Probing with Global Average Pooling (GAP)** aggregates all tokens into a single vector regardless of their relevance (**a**); **AbMILP** learns a scalar attention map to selectively weight patches (**b**); **Efficient Probing** utilizes multi query cross attention mechanism with a set of learnable queries (**c**).

we also add the DINO-v2 (+reg) ViT-Large model (Oquab et al., 2023; Darcet et al., 2023), a JEA model whose parameters also served as the initialization of the VGGT training (Wang et al., 2025). By comparing these two models, we directly analyze the shift in representations of a model pretrained for high-level recognition resulting from adaptation to 3D perception tasks.

Adapting VFMs to solving SpaRRTa via probing. At a conceptual level, our approach follows a simple principle: if a pretrained VFM representation contains spatial information, it should be possible to extract it via lightweight probing. By default, we extract the features from the final transformer block (we conduct a detailed analysis of intermediate layers in Section E). Following the protocols established in recent mid-level vision benchmarks (Chen et al., 2024), we evaluate representations by attaching different types of probing heads to the VFMs: **(i)** linear probing with Global Average Pooling of the patch tokens, as well as Selective Aggregation via **(ii)** Attention-Based Multiple Instance Learning Pooling (AbMILP) (M. et al., 2018; Przewięźlikowski et al., 2025), and **(iii)** Efficient probing (Bill Psomas et al., 2025). We depict these approaches in Figure 6. Probe **(i)** evaluates a global representation formed by naively aggregating dense VFM features, while **(ii–iii)** additionally account for their spatially local nature.

Further details of the experiments are provided in the Section A.

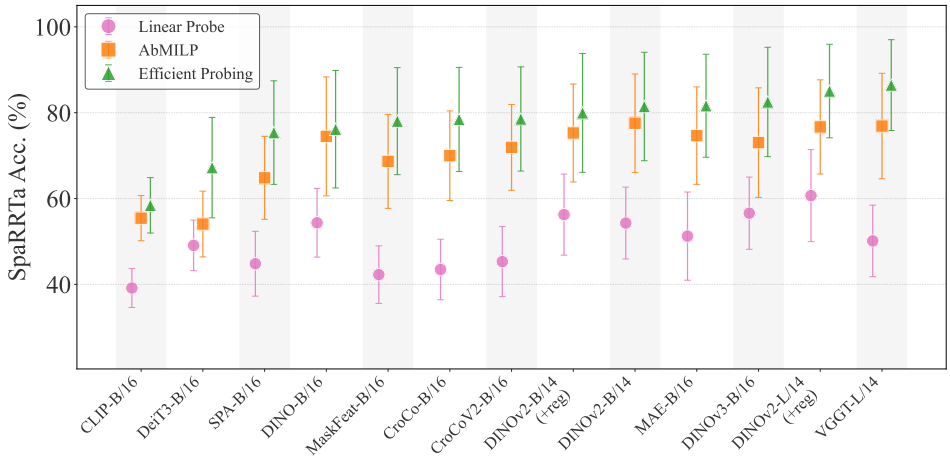


Figure 7: **Impact of Probing Strategy on Spatial Accuracy.** We report the aggregated mean accuracy across all five evaluation environments and both tasks (SpaRRTa-**ego** and SpaRRTa-**allo**) for all VFMs. The same performance hierarchy emerges across all backbones (**Linear** $<$ **AbMILP** $<$ **Efficient Probing**), indicating that spatial information is primarily encoded in local patch features and largely lost during global pooling.

3.2 SPATIAL RELATION RECOGNITION TASK RESULTS

We evaluate the efficacy of Visual Foundation Models (VFMs) in solving the Spatial Relation Recognition Task (SpaRRTa) with three probing strategies five environments. Below, we discuss several key findings that characterize how VFMs encode spatial relations.

Spatial relation information is more accessible in dense patch representations than in global features. Across all evaluated models, we observe a consistent performance hierarchy of **Linear** $<$ **AbMILP** $<$ **Efficient Probing** (Figure 7), indicating that spatial aggregation mechanisms capable of selectively emphasizing informative regions are critical for SpaRRTa. Linear probing based on global average pooling assumes that spatial information is uniformly present across patches; in complex scenes, this assumption breaks down, as only patches corresponding to the source, target, or viewpoint objects are informative, while background patches dilute the signal. AbMILP partially alleviates this issue by learning patch-wise importance weights, leading to consistent improvements over linear probing, particularly in cluttered environments.

This trend is further amplified by the distinction between single-map and multi-query aggregation. While AbMILP compresses selected regions into a single weighted representation via one attention map (Przewięźlikowski et al., 2025), efficient probing employs multiple learnable queries that can specialize to different spatial entities or scene components (Bill Psomas et al., 2025). We provide qualitative visualizations of these learned attention patterns in Section D. For relation-based tasks that require reasoning about multiple distinct objects, this additional capacity provides a consistent advantage.

3D supervision primarily enhances patch-level spatial structure rather than global representations. Different probing mechanisms reveal a counterintuitive divergence when comparing VGGT (Wang et al., 2025) with its parent model, DINO-v2 (+reg) (Oquab et al., 2023; Darcet et al., 2023), from which it is initialized and further trained with explicit 3D-awareness supervision. Under linear probing, VGGT does not outperform DINO-v2 and in some cases performs slightly worse, indicating that 3D supervision does not substantially improve global image-level representations for spatial relation recognition. In contrast, under AbMILP and efficient probing, VGGT surpasses DINO-v2 on the SpaRRTa task (see Figure 7). This divergence suggests that 3D supervision enriches patch-level geometric structure, which remains largely inaccessible to global pooling but can be effectively exploited by spatially structured probing mechanisms.

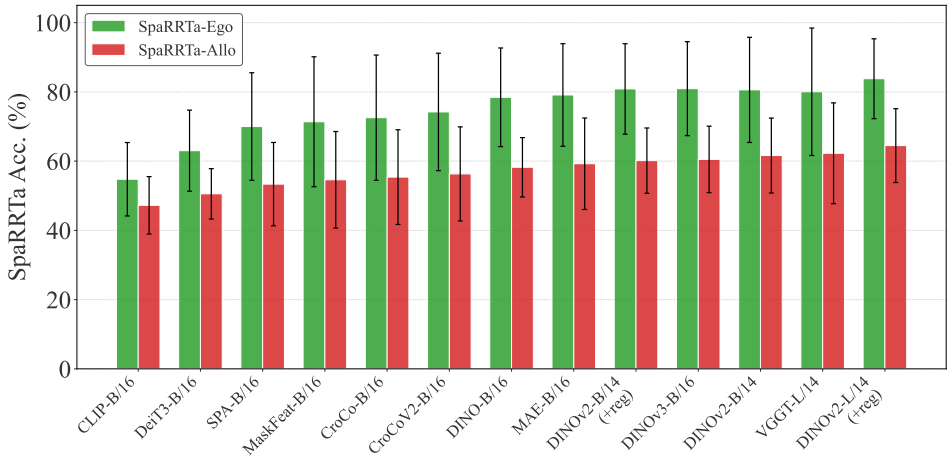


Figure 8: **Impact of Viewpoint Shift on Spatial Accuracy.** We report the mean accuracy aggregated across all probing methods and environments for each VFM. The consistent superiority of the egocentric bars (left) over the allocentric bars (right) confirms that resolving spatial relations relative to a non-camera viewpoint is significantly more challenging than reasoning from the camera view, regardless of the underlying model architecture.

Allocentric spatial relation recognition is consistently more challenging than egocentric recognition. Across all evaluated models, probing strategies, and environments, accuracy on the allo-

centric variant is systematically lower than on the egocentric variant (see Figure 8). Egocentric relations can be resolved directly from the camera viewpoint, and performance for the strongest models approaches saturation. In contrast, allocentric recognition requires reasoning about spatial relations from a viewpoint that does not coincide with the rendered image, introducing a substantial and persistent performance gap that remains challenging for all tested VFMs.

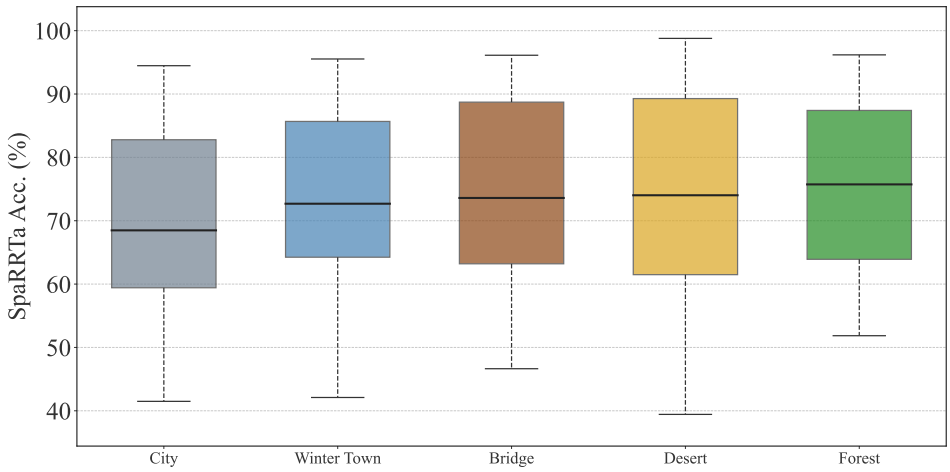


Figure 9: **Influence of Environmental Complexity on Model Performance.** We report the accuracy distribution of ViT Large backbones across five evaluation environments, averaged over all probing strategies. Performance peaks in visually homogeneous scenes (e.g., Desert, Forest) but degrades in cluttered scenes (e.g., City). This shows that visual noise makes it harder for models to understand spatial relations.

Environmental complexity significantly affects spatial relation recognition accuracy. As shown in Figure 9, models achieve higher performance in environments with visually homogeneous backgrounds, such as Desert and Forest, and lower performance in scenes with increased semantic clutter, such as City and Winter Town. This performance gap indicates that background complexity and visual noise can interfere with identifying the patches relevant for spatial relation reasoning, making the task more challenging in cluttered environments. The consistency of this trend across models suggests that sensitivity to environmental complexity is a general property of current visual representations rather than a model-specific artifact.

Additional results are discussed in Section B. In Section B.1, we compare SpaRRTa with other spatial and semantic tasks. In Section B.2, we analyze the inner representations of VFMs to better understand how spatially-aware representations are formed and where they are expressed.

4 Conclusion

Visual Foundation Models (VFMs) are a crucial component of embodied applications, encoding optical information about an agent's surroundings. However, good semantic recognition alone is insufficient for selecting appropriate actions in physical space. In this paper, we introduced Spatial Relation Recognition Task (SpaRRTa), a benchmark for evaluating whether a VFM represents spatial relations between objects in images.

Our results show that while modern VFMs contain non-trivial spatial information, this information is not uniformly accessible: spatial relations are primarily encoded at the patch level and are easily obscured by global pooling. As a consequence, standard linear evaluation substantially underestimates spatial awareness, particularly in settings that require perspective-dependent reasoning. We further show that decoding relative (allocentric) spatial relations remains challenging across model families, and that explicit 3D supervision primarily improves patch-level geometric structure rather than global representations.

By isolating spatial relation recognition from semantic classification and metric estimation, SpaRRTa provides a complementary evaluation axis that is not captured by existing benchmarks.

REFERENCES

Shuai Bai, Keqin Chen, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, et al. Probing the 3D Awareness of Visual Foundation Models, 2024. URL https://arxiv.org/abs/2404.08636.

Adrien Bardes, Quentin Garrido, et al. Revisiting feature prediction for learning visual representations from video. *arXiv:2404.08471*, 2024.

Dionysis Christopoulos Bill Psomas et al. Attention, please! revisiting attentive probing through the lens of efficiency, 2025. URL https://arxiv.org/abs/2506.10178.

Daniel Bolya, Po-Yao Huang, Peize Sun, et al. Perception Encoder: The best visual embeddings are not at the output of the network, 2025. URL https://arxiv.org/abs/2504.13181.

Neil Burgess, Eleanor A Maguire, and John O'Keefe. The human hippocampus and spatial and episodic memory. *Neuron*, 35(4):625–641, 2002. ISSN 0896-6273. doi: https://doi.org/10.1016/S0896-6273(02)00830-9.

Mathilde Caron, Hugo Touvron, et al. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Xuweiyi Chen, Markus Marks, and Zezhou Cheng. Probing the mid-level vision capabilities of self-supervised learning, 2024. URL https://arxiv.org/abs/2411.17474.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers, 2023. URL https://arxiv.org/abs/2309.16588.

Matt Deitke, Winson Han, Alvaro Herrasti, et al. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3161–3171, Los Alamitos, CA, USA, Jun 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.00323.

Jia Deng, Wei Dong, et al. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Epic Games. Unreal python api documentation. https://dev.epicgames.com/documentation/en-us/unreal-engine/python-api/?application_version=5.5, 2025a. Accessed: 2025-12.

Epic Games. Unreal engine. https://www.unrealengine.com, 2025b. Version 5.5.4.

Kaiming He, Xinlei Chen, et al. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, June 2022.

Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, et al. NAVI: Category-Agnostic Image Collections with High-Quality 3D Shape and Pose Annotations, 2023. URL https://arxiv.org/abs/2306.09109.

Tammy J. Jechura. Animal spatial cognition: Comparative, neural & computational approaches, 2006.

Scott P. Johnson and David S. Moore. Spatial thinking in infancy: Origins and development of mental rotation between 3 and 10 months of age. *Cognitive Research: Principles and Implications*, 5 (1), Mar 2020. doi: https://doi.org/10.1186/s41235-020-00212-x.

Bo Li, Yuanhan Zhang, et al. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=zKv8qULV6n.

Tsung-Yi Lin, Michael Maire, et al. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Drew Linsley, Peisen Zhou, et al. The 3d-pc: a benchmark for visual perspective taking in humans and machines. *International Conference on Learning Representations*, 2025.

Ilse M. et al. Attention-based deep multiple instance learning. In *ICML*, 2018.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Maxime Oquab, Timothée Darcet, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. URL https://arxiv.org/abs/2304.07193.

Huy V. Vo Oriane Siméoni et al. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

Marcin Przewięźlikowski, Randall Balestriero, Wojciech Jasiński, Marek Śmieja, and Bartosz Zieliński. Beyond [cls]: Exploring the true potential of masked image modeling representations, 2025. URL https://arxiv.org/abs/2412.03215.

Weichao Qiu, Fangwei Zhong, et al. UnrealCV: Virtual worlds for computer vision. *ACM Multimedia Open Source Software Competition*, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12159–12168, 2021. doi: 10.1109/ICCV48922.2021.01196.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses, 2021. URL https://arxiv.org/abs/2104.10972.

Philip J. Schneider and David H. Eberly. Chapter 11 - intersection in 3d. In *Geometric Tools for Computer Graphics*, The Morgan Kaufmann Series in Computer Graphics, pp. 481–662. Morgan Kaufmann, San Francisco, 2003. ISBN 978-1-55860-594-7.

Sarthak Srivastava and Kathy Wu. HyperVLM: Hyperbolic space guided vision language modeling for hierarchical multi-modal understanding. In *Second Workshop on Visual Concepts*, 2025. URL https://openreview.net/forum?id=kNWsjLgb3I.

Agnieszka Sroka-Oleksiak, Adam Pardyl, Dawid Rymarczyk, et al. Ai-driven rapid identification of bacterial and fungal pathogens in blood smears of septic patients, 2025. URL https://arxiv.org/abs/2503.14542.

Esa Rahtu Subhransu Maji et al. Fine-grained visual classification of aircraft, 2013. URL https://arxiv.org/abs/1306.5151.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit, 2022. URL https://arxiv.org/abs/2204.07118.

Shashanka Venkataramanan, Mamshad Nayeem Rizve, Joao Carreira, Yuki M Asano, and Yannis Avrithis. Is imagenet worth 1 video? learning strong image encoders from 1 long unlabelled video. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Yen1lGns2o.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Chen Wei et al. Masked feature prediction for self-supervised visual pre-training, 2023. URL https://arxiv.org/abs/2112.09133.

Philippe Weinzaepfel, Vincent Leroy, et al. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. In *Advances in Neural Information Processing Systems*, 2022a.

Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, et al. CroCo v2: Improved Cross-view Completion Pre-training for Stereo Matching and Optical Flow, 2022b. URL https://arxiv.org/abs/2211.10408.

Haoyi Zhu, Honghui Yang, Yating Wang, et al. SPA: 3D Spatial-Awareness Enables Effective Embodied Representation, 2024. URL https://arxiv.org/abs/2410.08208.

A Experiment Details

A.1 Implementation details

To ensure the reproducibility of our results and the accessibility of the SpaRRTa benchmark to the wider research community, we utilize standard, open-source libraries and consumer-grade hardware for all experiments.

Simulation and Rendering Stack. The synthetic environment generation is built on Unreal Engine 5.5, leveraging its native Python API (Epic Games, 2025a) for scene manipulation and automated asset placement. We employ the UnrealCV plugin (Qiu et al., 2017) to capture synchronized RGB images and segmentation masks. The rendering pipeline runs on a standard Windows workstation equipped with two NVIDIA RTX 2080 Ti (11GB VRAM), generating high-fidelity data without the need for enterprise-grade compute clusters.

Probing and Evaluation Stack. All probing experiments are conducted in a Linux environment using an NVIDIA RTX 4090 (24GB VRAM) graphics card. We follow the established protocols for probing frozen features, specifically the methodology outlined in recent mid-level vision benchmarks (Chen et al., 2024). The evaluation framework is implemented in PyTorch, with pre-trained VFMs sourced directly from the official HuggingFace Transformers or timm libraries to ensure standardized weight initialization.

A.2 Evaluation protocol

Hyperparameter	Linear Probing	AbMILP	Efficient Probing
Optimizer		AdamW	
Weight Decay		0.001	
Scheduler		Cosine Decay	
Learning Rate		$10^{-2}, 10^{-3}, 10^{-4}$	
Dropout		$0.2, 0.4, 0.6$	
Batch Size		256	
Linear Warmup	200	100	100
Training Epochs	1000	500	500
Method-Specific Architectures			
Hidden Dimension	1 linear layer	2-layer MLP	1 linear layer
Queries (N_q)	-	-	4
Output Dim (D_o)	D_i	D_i	D_i / 8

Table 2: **Hyperparameters for Spatial Probing.** We list the configuration used to train the probing heads (Linear, AbMILP, Efficient Probing) on top of frozen VFM backbones. Training settings are consistent across both egocentric and allocentric tasks.

For a given VFM, we evaluate the SpaRRTa performance in terms of the accuracy of probes trained for recognizing the spatial relations of given `source, target, viewpoint` object triples. Performance on SpaRRTa across images belonging to a given semantic triple therefore reflects the ability to resolve spatial relations for those particular object types, while averaging over diverse semantic triples provides an aggregate measure of spatial relation recognition. For each such triple of objects, we curate a dataset of images depicting different object layouts, and split it into train, validation, and test folds in 80/10/10 proportions. We train triple-specific probes for the number of epochs specified in Table 2, using the validation set to select the best-performing probe parameters. We report the test accuracies of probes, and repeat this procedure with 2 random seeds and a diverse selection of 3 distinct object triples per each of the 5 environments (see Section 2.3) to achieve a robust evaluation.

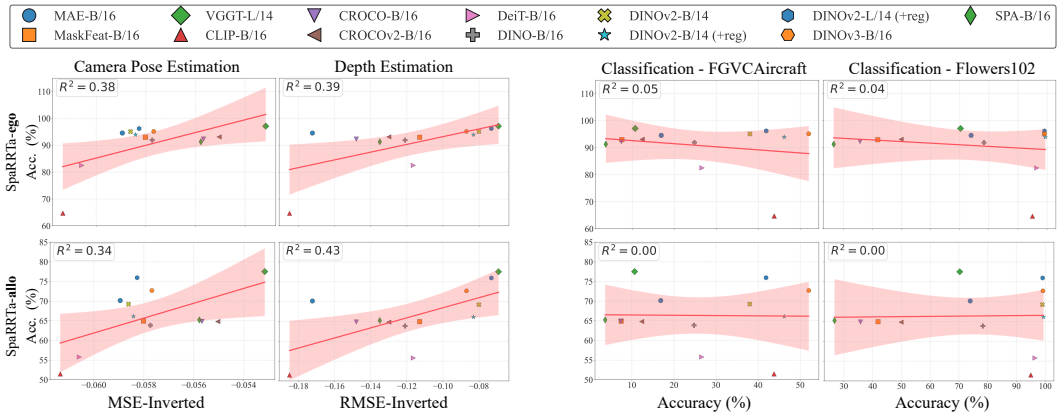
B Additional Experimental Results

B.1 Comparison of SpaRRTa with other spatial awareness benchmarks

To validate that SpaRRTa indeed measures spatial reasoning rather than proxy signals (e.g., low-level texture matching or high-level semantics), we correlate our results with established computer vision benchmarks. We select two mid-level geometric tasks, camera pose estimation and monocular depth estimation (Banani et al., 2024) on the NAVI dataset (Jampani et al., 2023), and two high-level semantic tasks, FGVC Aircraft (Subhransu Maji et al., 2013) and Flowers102 (Nilsback & Zisserman, 2008) classification. For external benchmarks, we strictly follow standard probing protocols. For classification, we use linear probing on frozen backbones (Chen et al., 2024). For depth estimation, we attach a dense prediction transformer (DPT) head (Ranftl et al., 2021) as in (Banani et al., 2024), and for camera pose estimation, we use a 3-layer MLP regressor as in (Zhu et al., 2024).

For SpaRRTa, we use the mean accuracy across all five environments obtained by efficient probing (Bill Psomas et al., 2025), as it provides the most robust estimate of a model's spatial capability.

As evident from Figure 10a, we observe a significant positive correlation between SpaRRTa accuracy and performance on 3D tasks. Specifically, the allocentric task achieves a coefficient of determination of $R^2 = 0.43$ with depth estimation and $R^2 = 0.34$ with camera pose estimation. This strong relationship confirms that the abstract reasoning required by SpaRRTa is grounded in the same latent geometric representations used for explicit 3D reconstruction. Models with a rich understanding of depth and viewpoint (e.g., VGGT and DINOv3) consistently excel at SpaRRTa, reinforcing the benchmark's validity as a probe of spatial awareness. We further observe that models pre-trained with explicit multi-view consistency objectives—specifically SPA (Zhu et al., 2024), CroCo (Weinzaepfel et al., 2022a), and CroCoV2 (Weinzaepfel et al., 2022b) exhibit higher performance in the camera pose estimation task compared to our tasks and other tasks. We attribute this to their unique pre-training paradigms, which compel the network to internalize camera transformations. CroCo's cross-view completion objective requires the model to reconstruct masked patches from two different viewpoints of the same scene, effectively learning camera geometry as a latent variable. Similarly, SPA employs volumetric neural rendering to enforce consistency across multiple views from the same scene, directly embedding 3D spatial structure into the 2D representation. Consequently, these models exhibit a more substantial inductive bias for geometric tasks, which explains their distinct advantage in regressing camera poses.



(a) Correlation between SpaRRTa performance and 3D mid-level vision tasks.

(b) Correlation between SpaRRTa performance and high-level semantic classification.

Figure 10: **Correlation of SpaRRTa with 3D and Semantic Benchmarks.** We compare model accuracy on the SpaRRTa-**ego** (top) and SpaRRTa-**allo** (bottom) tasks against four external benchmarks: Camera Pose and Depth Estimation (left) and FGVC Aircraft and Flowers102 Classification (right).

In contrast, Figure 10b reveals the lack of correlation ($R^2 \approx 0.00$) between SpaRRTa and fine-grained classification benchmarks. High performance in semantic tasks (e.g., identifying flower species) does not predict success in objects' spatial relationships. This result implies that SpaRRTa targets a specific spatial skill set that does not emerge from strong semantic features.

B.2 Analysis of the mechanism of forming spatial representations

In this section, we analyze the mechanism of forming spatial representations in two VFMs that tend to achieve the best SpaRRTa performance – DINO-v2 (with register tokens) (Oquab et al., 2023; Darcet et al., 2023), and VGGT (Wang et al., 2025). Apart from their potent spatial representations, the second reason we focus on these models is that the weights of DINO-v2 serve as an initialization for VGGT, which is subsequently trained with a 3D structure perception objective. Therefore, the analysis of these models allows us to characterize not only which properties of representations inform spatial relation encoding, but also the differences between the initial model pretrained for semantic tasks, and its version tuned for 3D perception.

To investigate why VGGT outperforms DINO-v2 on spatial tasks when using Efficient Probing, yet underperforms with Linear Probing, we analyze the internal attention mechanisms of both backbones. We hypothesize that VGGT's 3D supervision forces the model to restructure how information flows between patches, shifting from object-centric feature extraction to relational scene encoding.

Methodology. We conduct this analysis across the five evaluation environments and three objects (Tree, Truck, and Human), selecting 60 clear, non-occluded images per environment where the objects are strictly visible. Using ground-truth segmentation masks obtained from the UnrealCV API (Qiu et al., 2017), we map every patch token to its corresponding object (Human, Tree, Truck, or Background). We then compute the mean attention weights from the source object's patches to all other regions across all heads in the frozen transformer blocks (see Figure 11). We visualize the layer-wise evolution of these attention scores (averaged across heads and environments) on a logarithmic scale to account for the abundance of background patches.

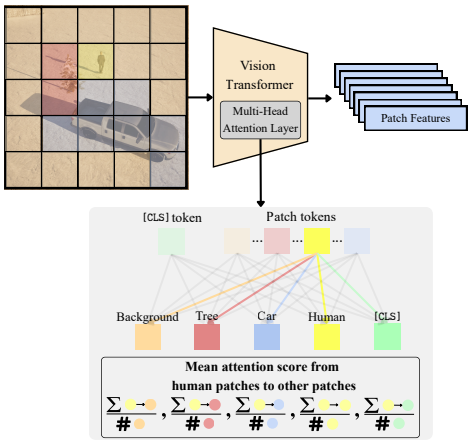


Figure 11: **Methodology for Attention Dynamics Analysis.** We illustrate the protocol for mapping patch tokens to semantic categories and calculating the inter-object attention scores. A sample input image is patchified and passed through the frozen ViT backbone to extract raw attention scores between tokens (top). We use ground-truth segmentation masks to assign semantic categories (Background, Tree, Car, Human) to these tokens. Finally, we aggregate patch tokens to compute the mean attention flow between objects (e.g., Human $\rightarrow$ Car, or Human $\rightarrow$ `[CLS]` token) using the formula displayed (bottom).

Model's attention behavior. We observe that finetuning of VGGT for 3D tasks induces a fundamental reorganization of information flow in the deep layers (specifically layers 15–24). As visualized in the VGGT attentions (Figure 13, Bottom Left), we observe a distinct relational shift. The attention from an object to itself (e.g., Human $\rightarrow$ Human) drops sharper in the final layers of VGGT compared to DINO-v2. At the same time, attention directed toward other objects (e.g., Human $\rightarrow$ Tree, Human $\rightarrow$ Truck) exhibits a sharper increase. This redistribution of attention is quantified in the difference plots (VGGT $-$ DINO-v2) presented in Figure 13 (Bottom Right). A clear structural divergence emerges. The differential for the object's attention itself becomes negative, while the differential for cross-object attention becomes positive. This indicates that VGGT actively repurposes the probability mass previously allocated to local self-refinement and redirects it to encode spatial dependencies between distinct entities.

Implications for Probing Performance. This structural reorganization provides an explanation for the probing results reported in Section 3.2. To validate this directly, we conducted an experiment comparing linear probing specifically on the `[CLS]` token versus efficient probing for VGGT and DINO-v2 (see Figure 12). DINO-v2 outperforms VGGT when restricted to the global `[CLS]` token, yet VGGT surpasses when using the efficient probe. Why does the `[CLS]` token fail for VGGT? Our analysis in Figure 13 (Top Row) offers the answer. In DINO-v2, the `[CLS]` token maintains high attention to the objects (e.g., the Truck). This allows the single global vector to retain enough scene context for a linear probe to succeed. In contrast, VGGT's `[CLS]` token shifts its attention budget toward register tokens. This suggests that during 3D finetuning, VGGT offloads global geometric parameters to registers and distributes object-relative spatial details across the local patch tokens. Consequently, a linear probe on the single `[CLS]` token fails to access this distributed information, whereas efficient probing, which aggregates across the entire patch sequence via attention, successfully retrieves the rich spatial context, allowing VGGT to ultimately surpass DINO-v2 in SpaRRTa.

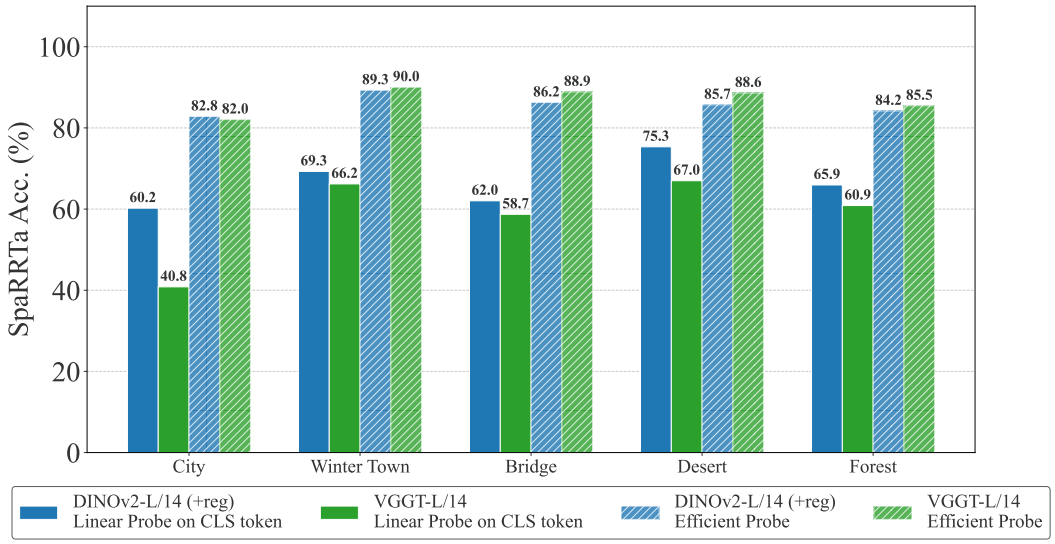


Figure 12: **Linear probing on `[CLS]` token vs. efficient probing.** We compare the spatial reasoning accuracy of DINO-v2 versus VGGT using two probing strategies. We report the mean accuracy averaged across both SpaRRTa**-ego** and SpaRRTa**-allo** task. DINOv2 outperforms VGGT when linear probing on the global [CLS] token, but VGGT surpasses it with the efficient probe.

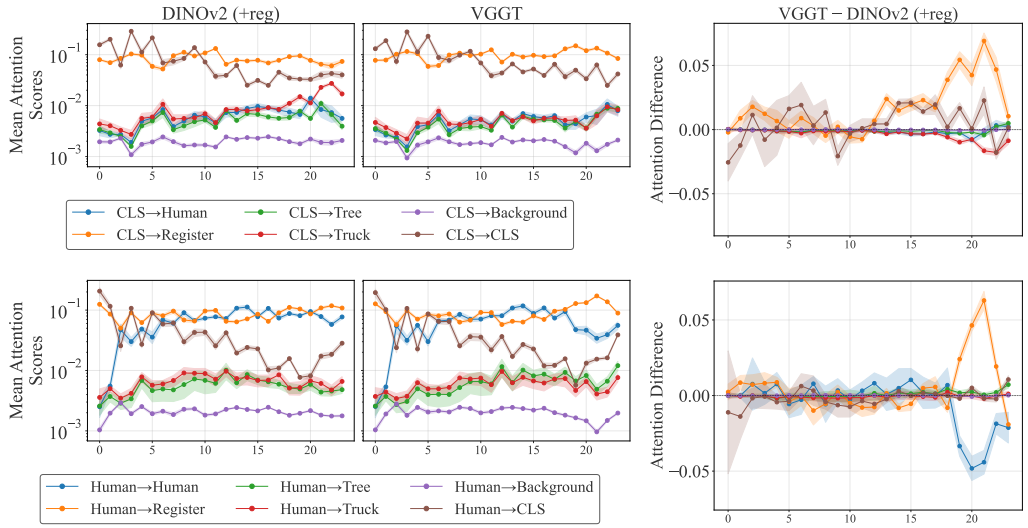
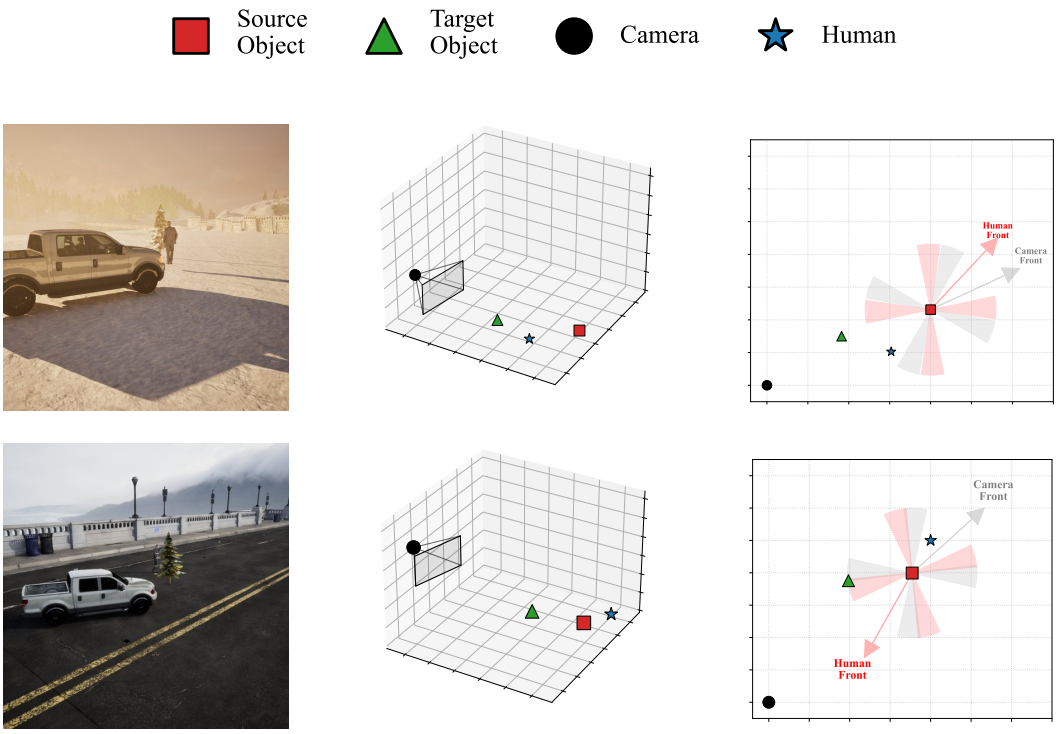


Figure 13: **Comparative Analysis of Attention Flow (DINO-v2 vs. VGGT).** We visualize the layer-wise mean attention scores averaged across heads and environments from the global `[CLS]` token (top row) and human patches (bottom row) to the rest of the patches (Human, Truck, Tree, Background, `[CLS]` and Register) for both DINO-v2 and VGGT backbones. The two columns on the left show absolute attention scores for DINO-v2 and VGGT, while the right column displays the differential (VGGT − DINOv2) to quantify the divergence in attention dynamics. A fundamental reorganization emerges across all token types in the deeper layers. In the bottom row, while DINOv2 prioritizes attention from human patches to itself or to register tokens (blue and orange negative curves in the difference plot), VGGT consistently reallocates attention to other objects (green and red positive curves in the difference plot). This shows that VGGT's explicit 3D supervision forces the model to actively encode spatial dependencies between distinct entities.



(a) Input RGB image (b) 3D scene layout (c) 2D scene logic

Figure 14: **Visualization of Task Geometry and Ambiguity Filtering.** We present the geometric construction of SpaRRTa samples, showing the rendered RGB input (**a**), the global 3D arrangement of assets (**b**), and the 2D logic for defining the egocentric (gray) and allocentric (red) spatial positions (**c**).

C Dataset Generation Methodology

C.1 Core Design Principles and Controllability

The design of SpaRRTa is governed by the need to disentangle spatial reasoning from low-level visual pattern matching while ensuring mathematically rigorous ground-truth labels. Unlike real-world datasets, where camera angles and object placements are biased by human photographers, our synthetic approach enables precise control over three critical axes: geometric validity, environmental diversity, and task complexity.

Defining spatial relations and ambiguity. As illustrated in Figure 14, the core challenge of SpaRRTa lies in resolving spatial relations relative to a specific coordinate frame of the viewpoint. Figure 14a shows examples of views from the Unreal Engine environment showing the source object (tree), target object (truck), and human agent. Figure 14b shows the three-dimensional spatial arrangement of the camera frustum, objects, and human agent in world coordinates. For the egocentric task (Figure 14c, gray arrow), the front direction is defined by the vector from the camera to the source object. For the allocentric task (Figure 14c, red arrow), the front is defined by the vector from the human agent to the source object. A fundamental challenge in spatial classification is defining boundaries between classes (e.g., when does Left become Front?). To ensure the benchmark's validity, we handle those geometric boundaries. We define ambiguity zones as conical regions (shaded red and gray in Figure 14c) centered around the $45°$, $135°$, $225°$, and $315°$ diagonals relative to the viewpoint's forward vector. Any sample in which the target object falls within these zones is automatically filtered from the dataset. This guarantees that the ground-truth labels (left, right, front, back) are unambiguous and robust to minor variations in pose estimation, forcing the model to learn precise spatial boundaries rather than guessing on the margins.

Environment	Package / Demo	Provider	Link
City	City Sample	Epic Games / Fab	Fab listing
Forest	Electric Dreams Environment	Epic Games / Fab	Fab listing
Desert	Desert Landscape	UE Marketplace / Fab	Fab listing
Winter Town	Winter Town Environment	UE Marketplace / Fab	Fab listing
Bridge	Bridge Scene	UE Marketplace / Fab	Fab listing

Table 3: Hyperlinked source registry for environment packs used in SpaRRTa.

Environment	Asset	Provider	Link
Global	Scanned 3D People Pack	Fab	Fab listing
Global	Generic Vehicle Pack	Fab	Fab listing
Global	Tree (from Electric Dreams)	Epic Games / Fab	Fab listing
Forest	Brown Bear	Sketchfab	Sketchfab listing
Forest	Red Fox	Sketchfab	Sketchfab listing
Forest	Camping Tent	Sketchfab	Sketchfab listing
Forest	Rocks	Desert Landscape / Electric Dreams	Desert Landscape; Electric Dreams
Desert	Camel 3D Model	Fab	Fab listing
Desert	Barrel	Sketchfab	Sketchfab listing
Desert	Cactus Model	Sketchfab	Sketchfab listing
Desert	Rocks	Desert Landscape / Electric Dreams	Desert Landscape; Electric Dreams
Winter Town	Siberian Husky	Sketchfab	Sketchfab listing
Winter Town	Deer	Sketchfab	Sketchfab listing
Winter Town	Snowman	Sketchfab	Sketchfab listing
Bridge	Bicycle Asset	Fab	Fab listing
Bridge	Trash Can (from Bridge Scene pack)	Unreal Engine Marketplace / Fab	Fab listing
City	Vespa Scooter	Sketchfab	Sketchfab listing
City	Traffic Cone	Sketchfab	Sketchfab listing
City	Fire Hydrant	Sketchfab	Sketchfab listing

Table 4: Hyperlinked source registry for third-party object assets used in SpaRRTa.

Asset curation and environmental diversity. To test the robustness of spatial representations, we use five distinct environments in the SpaRRTa benchmark. The environments are as follows:

- **Forest environment** is based on the Electric Dreams Environment Unreal Engine product demo. This scene depicts a sparse forest landscape with complex foliage, uneven terrain, and natural rock formations.
- **Desert environment** is a vast, arid landscape characterized by open terrain, sand dunes, and high-contrast lighting. This environment is very sparse and texture-homogeneous compared to other environments.
- **Winter Town environment** is a snow-covered setting reflecting a typical small Eastern European town. This environment consists of cold lighting, occluding snow textures, and a small number of village-type buildings.
- **Bridge environment** is a valley scene centered around a large bridge infrastructure.
- **City environment** is a large-scale, modern American metropolis adapted from the City Sample demo, featuring high-rise architecture, paved roads, and complex urban geometry.

Table 3 shows the hyperlinked records for all environment packs. We curate specific asset sets for each environment to maintain semantic coherence, ensuring objects appear in natural contexts while preserving geometric consistency. Crucially, we select isotropic objects (rock, tree, traffic cone) as

source objects. Their rotationally symmetric nature simplifies the definition of relative positions, ensuring that the spatial relation (e.g., Left of the Tree) is defined purely by the target's position relative to the source's center, minimizing ambiguity caused by the source object's own orientation. The assets we used in our benchmark are detailed in Table 4.

Dataset size and generalization. To determine the optimal data scale, we analyzed model generalization across tasks. We found that **5,000 images** per environment were sufficient for models to generalize on the egocentric task. However, the allocentric task, which requires learning a more complex perspective transformation, exhibited signs of overfitting at this scale. Consequently, we increased the allocentric dataset size to **10,000 images** per environment, ensuring sufficient diversity for the model to learn the underlying geometric rules rather than memorizing specific scene configurations.

C.2 Rendering Pipeline Details

Our approach leverages the native Unreal Engine Python API (Epic Games, 2025a) and the UnrealCV plugin (Qiu et al., 2017) to programmatically control object placement, physics validation, and camera logic. The pipeline outputs raw RGB renders, serializes the precise 6-DoF state (position: x, y, z and rotation: ϕ, θ, ψ) of every task-critical objects and camera into JSON metadata files, and generates pixel-perfect instance masks of the source, target, and human objects.

Terrain adaptation via raycasting. We employ a rejection sampling strategy to ensure physical plausibility and visual clarity. We implement a physics-aware placement logic to handle uneven terrain (e.g., the Forest and Desert environments) without objects floating or clipping. For every object coordinate, the pipeline performs a vertical line trace to detect the exact Z-height of the landscape geometry. Objects are then spawned at the detected ground level, with their specific bounding box offsets applied to ensure flush contact.

Visibility and occlusion control. We enforce visibility constraints at the generation level to guarantee that all task-critical objects are captured. Candidate object positions are validated against the camera's viewing frustum by computing the angular deviation of each object relative to the camera's optical axis. We enforce a strict constraint where the maximum angular offset must remain within the camera's horizontal field of view to prevent objects from being cut off at the frame edges. Furthermore, to ensure appropriate scene composition, we reject configurations where objects appear too clustered or distant, forcing the camera to capture a meaningful spatial spread. Finally, we utilize AABB (Axis-Aligned Bounding Box) (Schneider & Eberly, 2003) overlap checks to prevent inter-object collisions.

Camera configuration and object placement. We standardize the optical setup across all environments to prevent intrinsic camera parameters from becoming a confounding variable. All scenes are rendered using a simulated 50mm lens on a 50mm sensor width, resulting in a consistent horizontal field of view. The camera positioning follows a hierarchical stochastic process. Object coordinates are sampled around a randomly selected center point, with a distance limit to ensure they remain close enough to be distinguishable. The camera is then spawned within a broad area surrounding this group, with its height randomized relative to the average elevation of the objects. Finally, the pipeline computes the geometric centroid of the relevant scene objects (source, target, human agent) and dynamically locks the camera's rotation to center this cluster, ensuring the primary task elements remain the focal point.

D Extended Attention Visualizations

D.1 Attention Maps of Efficient Probing

In Figure 15, we visualize the mean attention maps generated by the efficient probing head attached to the frozen VGGT backbone to interpret how the model solves these spatial tasks. These visualizations show that the attention focus changes depending on the task. For the egocentric task (Columns 2 and 5), the model attends to the source and target objects while ignoring the other object. However, when the allocentric task (Columns 3 and 6) is solved, the model attends to the non-camera-viewpoint object, the source, and the target. This dynamic behavior confirms that the probe does

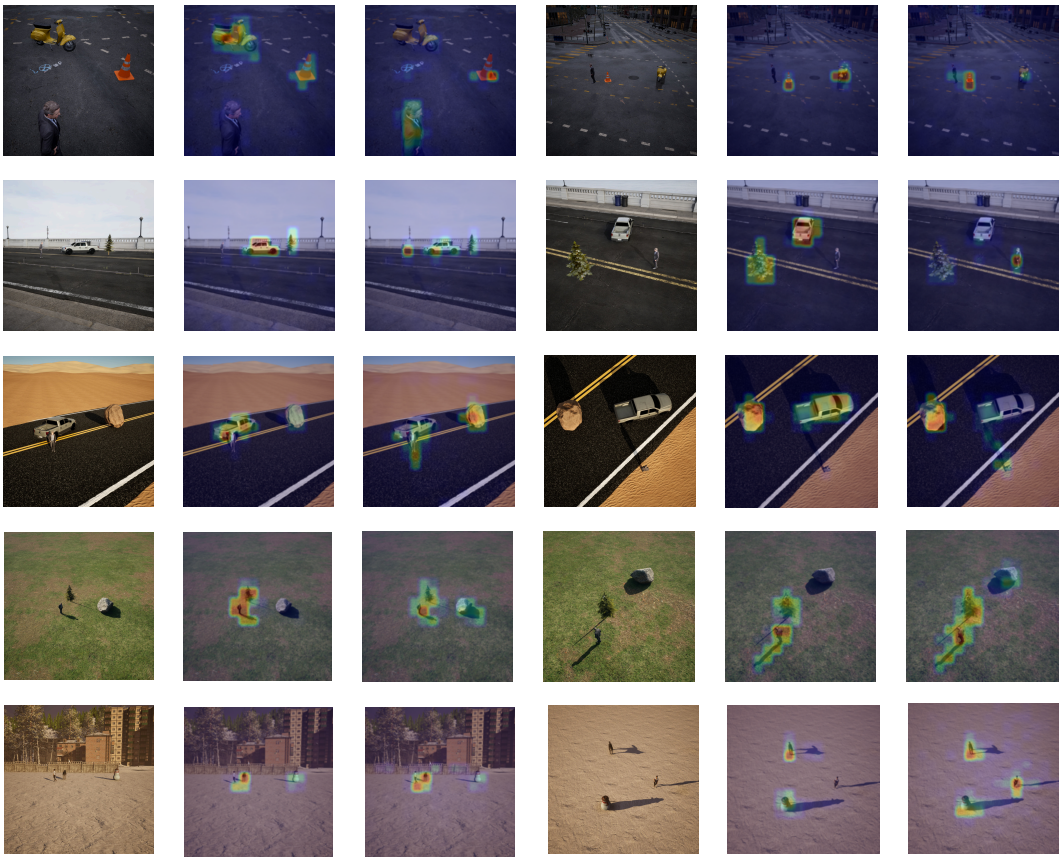


Figure 15: **Visualization of Task-Specific Attention Dynamics in Efficient Probing.** We visualize the mean attention maps (averaged across 4 learned queries) extracted from an efficient probing head attached to a frozen VGGT backbone.

not merely memorize texture but actively identifies and disentangles the specific geometric entities, viewpoint, source, and target, required to resolve the spatial query in the correct coordinate frame.

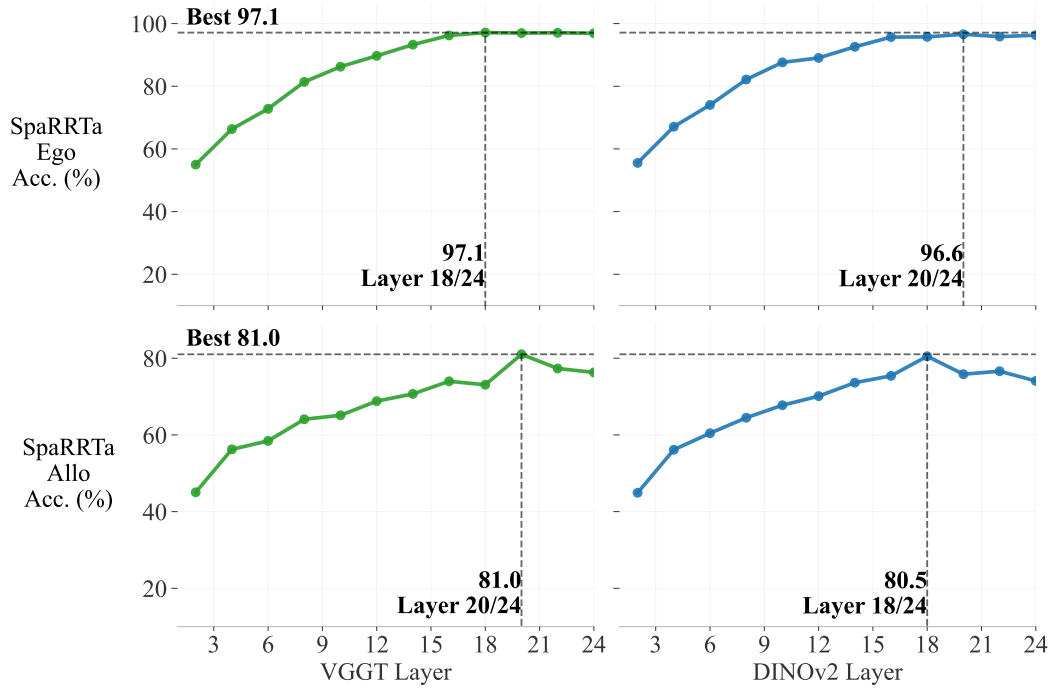


Figure 16: **Layer Analysis.** We evaluate the spatial reasoning capability of intermediate features extracted from frozen VGGT and DINO-v2 (with registers) ViT Large backbones using efficient probing. Results are averaged across all five evaluation environments to ensure robustness against domain-specific bias. Vertical dashed lines denote the layer achieving peak performance.

E Layer-wise Probing: Where is Spatial Information Encoded?

To understand how spatial reasoning emerges and evolves throughout the network depth, we evaluate the performance of intermediate layers as frozen features across both egocentric and allocentric tasks. For this analysis, we use VGGT and DINO-v2 (with register tokens), training an efficient probing head on the outputs of individual transformer blocks. To ensure that the observed trends reflect generalizable spatial skills rather than overfitting to specific visual domains, all reported results are averaged across the five diverse environments.

As illustrated in Figure 16, we observe a consistent trend across both model architectures and task types: spatial reasoning accuracy improves steadily through the initial layers, plateaus, and eventually peaks in the late-intermediate layers (specifically Layer 18 and 20 for a 24-layer ViT-L), rather than at the final output layer. This behavior aligns with findings from the Perception Encoder (Bolya et al., 2025) and recent architectural analyses in DINOv3 (Oriane Siméoni et al., 2025). In standard VFMs, the final layers are often optimized for high-level semantic abstraction and invariance (e.g., classifying a "dog" regardless of its position). This objective implicitly encourages the model to discard precise geometric details in the final projection. Our results confirm that for tasks where geometry plays a governing role, such as the allocentric perspective-taking challenge, the most robust representations are located deeper in the network stack, before the last layer.

While the performance drop-off at the final layer is relatively minor, which makes it still a viable default choice for general use, these results suggest that downstream embodied agents requiring precise spatial manipulation or reasoning can achieve optimal performance by tapping into these late-intermediate layers. Specifically, extracting features from Layer 18–20 for a ViT-Large yields a measurable improvement in accuracy, providing a free performance boost without additional training cost.