Token Warping Helps MLLMs Look from Nearby Viewpoints

Phillip Y. Lee* Chanho Park* Mingue Park Seungwoo Yoo Juil Koo Minhyuk Sung
KAIST

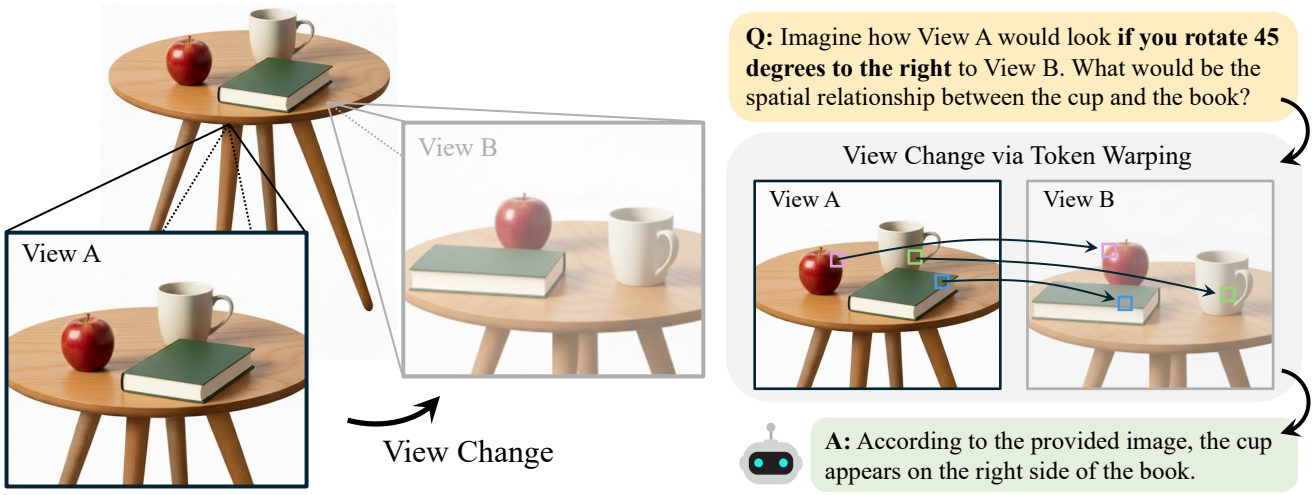

Figure 1. **Viewpoint Change via Token Warping.** We explore token warping as a means of enabling viewpoint changes for MLLMs and find that *backward token warping* can reliably transfer source image content to novel viewpoints without synthesizing new pixels.

Abstract

Can warping tokens, rather than pixels, help multimodal large language models (MLLMs) understand how a scene appears from a nearby viewpoint? While MLLMs perform well on visual reasoning, they remain fragile to viewpoint changes, as pixel-wise warping is highly sensitive to small depth errors and often introduces geometric distortions. Drawing on theories of mental imagery that posit part-level structural representations as the basis for human perspective transformation, we examine whether image tokens in ViT-based MLLMs serve as an effective substrate for viewpoint changes. We compare forward and backward warping, finding that backward token warping, which defines a dense grid on the target view and retrieves a corresponding source-view token for each grid point, achieves greater stability and better preserves semantic coherence under viewpoint shifts. Experiments on our proposed `ViewBench` *benchmark demonstrate that token-level warping enables MLLMs to reason reliably from nearby viewpoints, consistently outperforming all baselines including pixel-wise warping approaches, spatially fine-tuned MLLMs, and a generative warping method. Our project page is at* https://token-warping-mllm.github.io/.

*Equal contribution.
Correspondence: Phillip Y. Lee (phillip0701@kaist.ac.kr) and Minhyuk Sung (mhsung@kaist.ac.kr)

1. Introduction

A core aspect of spatial reasoning from images is understanding the scene's three-dimensional structure. Although depth estimation has achieved near-perfect accuracy [10, 108], incorporating predicted depth into MLLMs does not yield genuine 3D understanding. Even for simple tasks such as describing the same scene from a different viewpoint (Fig. 1), MLLMs fine-tuned with explicit 3D supervision [61] show little improvement. Similar limitations arise in models [26, 125] that incorporate 3D-aware features [97, 99], which still struggle to reason about viewpoint transformations.

Recent studies [15, 48, 80, 113, 123] inspired by mental imagery [28, 34, 44, 69, 73, 86, 92] suggest that perspective reasoning requires generating a virtual internal representation through explicit transformation. For instance, Lee *et al*. [48] model a scene using object-centric abstract representations and apply geometric transformations to them. While effective for object-level relational reasoning, such approaches

often fail to capture fine-grained details and overall spatial coherence of the scene.

Classical research on mental imagery, from Shepard [86] to Minsky [67], Pylyshyn [75], and Hinton [34], proposes that mental images rely on structural descriptions defined at the *part level* rather than at the holistic object level. From this perspective, the evolution of computer vision can be interpreted as the pursuit of machine-perceivable, part-level representations, which have recently converged in the form of *image tokens* used by Transformer architectures [24, 94]. It is therefore natural to extend the concept of mental imagery to these perceptual atomic units rather than to object-level abstractions.

Motivated by this insight, we investigate whether transformations applied to image tokens can generate consistent internal representations of scenes under viewpoint changes, thereby improving spatial reasoning. We find that this is indeed the case. Unlike pixel-level warping, which amplifies even small depth errors into severe distortions, token-level transformations remain robust to geometric noise and yield more coherent viewpoint reasoning.

To systematically verify our hypothesis that image tokens form a robust substrate for viewpoint transformation, we first examine how sensitive recent MLLMs are to noise introduced during local patch retrieval. For each image token, we begin with the regular grid centers but intentionally fetch the corresponding image patch from a *slightly perturbed* center position. By gradually increasing the perturbation magnitude, even to the point where the offset approaches the size of the patch, we observe that MLLMs remain surprisingly stable in their ability to recognize the underlying image content. This suggests that MLLMs are inherently tolerant to spatial noise during patch formation, providing strong evidence that when constructing image tokens from a different viewpoint using a predicted (and potentially imperfect) depth map, the geometric noise introduced during warping does not significantly undermine the model's visual understanding.

Next, we investigate how to best implement token-level warping under viewpoint changes. Given an input image with its depth map and a target camera pose, there are two possible transformation strategies: *forward* warping and *backward* warping. In the forward approach, we first construct the image tokens from the input view and then map each token to the target viewpoint. In contrast, the backward approach begins by taking the regular grid centers of the target view and mapping each center back to the input image. Within backward warping, we consider two variants. The first, *nearest fetching*, constructs all image tokens only once on the input view and then assigns to each mapped target location the nearest precomputed token. The second, *adaptive fetching*, directly re-patchifies the input image at each mapped location by treating it as the patch center, rather than assigning the nearest precomputed token.

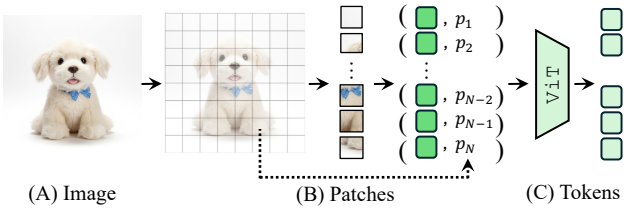

Figure 2. **Image Tokenization in MLLMs (Sec. 3.1).** MLLMs process images by dividing them into fixed-size patches, embedding each patch, and passing them through a vision encoder (*e.g.*, ViT) to obtain image tokens.

Through our experiments on `ViewBench`, designed to evaluate MLLMs on spatial reasoning tasks involving viewpoint changes, we systematically explore the aforementioned axes of pipeline design. The results show that both the choice of representation to warp and the specific warping mechanism have substantial effects on performance. In particular, we find that backward token warping, which preserves dense and regularly spaced grids in the target view, outperforms all other variants. Remarkably, this approach, which incurs only minimal inference-time computation for warping, surpasses state-of-the-art specialist MLLMs fine-tuned on spatial reasoning datasets, as well as a generative warping technique that employs a camera-conditioned diffusion model to directly synthesize the target-view image.

2. Related Work

2.1. Spatial Understanding in MLLMs

The potential of multimodal LLMs (MLLMs) on real-world embodied tasks have sparked research interest on their spatial reasoning abilities [25, 36, 37, 68, 120]. Rich line of benchmarks and evaluation protocols pointed out that MLLMs often struggle at even basic spatial understanding [20, 30, 60, 79, 80, 88, 96, 116], and showed that their spatial cognition can be improved by well-curated data [12, 22, 40, 49, 57, 87], introducing novel architecture designs [62, 93] and carefully designed training frameworks [51, 61, 72, 95]. Another line of work suggest that integrating rich structural priors (*e.g.*, depth maps [11], segmentation masks [16], point clouds [14, 23, 35], or rich features from foundation models [26, 39, 101, 125]) can assist MLLM's spatial reasoning on image, video and 3D inputs. This can be implemented in either by training auxiliary encoders to project new modalities into the model [38, 114, 117], or by designing novel prompting mechanisms [52, 77, 118, 128]. Multiple works integrate 3D-aware features or positional embeddings into 2D MLLMs to enhance their 3D understanding [17, 29, 91, 126, 127]. Moreover, other works focus on the LLM's reasoning skills, building agentic frameworks that tackle spatial tasks via program-like decomposition [59, 65] or test-time scaling algorithms [112].

2.2. Viewpoint-Aware Reasoning

As MLLMs increasingly serve as the *brains* of autonomous agents in open environments [27, 70, 76, 98, 109, 120], recent research has begun to examine their ability to handle *viewpoint-aware* perception and cognition [48, 64, 87, 124]. Notably, COMFORT [124] draws on cognitive studies about *frame of reference* for perspective-taking and shows that MLLMs are largely confined to the input camera's viewpoint. They struggle to adopt another person's or object's vantage point within the same scene, considered a core human cognitive skill. Related works further propose finer-grained evaluation criteria [32, 50, 55, 60, 121, 123] and suggest plug-in strategies inspired by human cognitive process to scaffold viewpoint reasoning [48]. When provided denser observations, either as multi-view images [15, 104, 111, 119] or videos [5, 63, 107, 122], it is also essential to interpret the scene from a specific viewpoint (*e.g.*, one of the frames). For this, Mindcube [113] generates a simple cognitive map to grasp the holistic structure of the scene, while ViLaSR [102] uses drawing as a tool for reasoning in space. We ask a new question: given a single image, can an MLLM *look* from a nearby viewpoint? We investigate this by warping tokens, rather than synthesizing pixels or auxiliary data, to simulate viewpoint shifts efficiently and robustly.

2.3. Image as Tokens

Since the introduction of Vision Transformers (ViT) [24, 94], it has become standard to divide images into patch-wise *tokens* as inputs to transformer-based vision models. Tokens serve as *semantic primitives* that support both local detail and global context understanding, driving strong performance across computer vision tasks including classification [71], detection [41, 66], segmentation [42, 82], 3D reconstruction [97, 100], multimodal understanding [56], and generation [45, 74, 83]. Building on this foundation, recent work explores deformable [103] and adaptive [13, 18, 81, 84, 103] tokenization techniques for improving semantic alignment and efficiency. Others leverage tokens for image/video generation [7, 47, 53, 78], editing [31, 43, 105], or perception [9, 26, 46, 115] by introducing richer token types or directly manipulating tokens to steer model behavior. In this work, we focus on the role of tokens as primary semantic units in MLLMs, and propose token warping as a lightweight and robust strategy to enable *viewpoint-aware perception*.

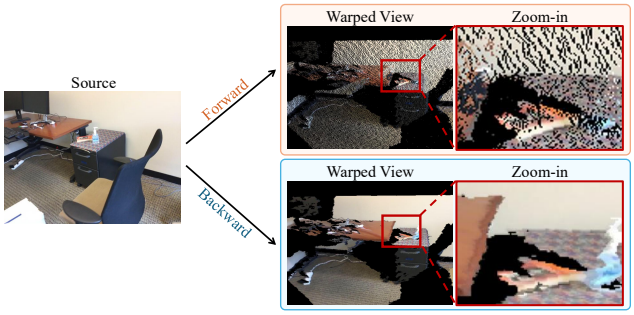

Figure 3. **Limitations of Pixel-Wise Warping.** Pixel-wise warping to a target viewpoint often introduces local distortions and semantic degradation. In both *forward* (top) and *backward* (bottom) warping, the book from the source view appears significantly distorted after transformation (in the red box).

3. Token Warping for Viewpoint Changes

Modern ViT-based MLLMs represent an image as a sequence of tokens obtained by dividing it into patches and embedding each into a latent vector. These image tokens function as *perceptual atoms* of the MLLM: localized, semantically meaningful units processed jointly with positional embeddings. Inspired by cognitive theories of mental imagery [34, 67, 75, 86], we investigate whether image tokens provide the appropriate part-level granularity for performing viewpoint transformations. Object-level representations [48] are too coarse, sacrificing important spatial and appearance details, while pixel-level representations are too fine-grained and sensitive to even small depth or geometric noise during warping (see Fig. 3). *Image tokens* lie between these extremes, retaining rich visual detail while remaining robust to local perturbations. We therefore posit that image tokens serve as an effective perceptual substrate for neural mental imagery and viewpoint transformation.

A key requirement for enabling such viewpoint transformations is robustness to positional perturbations introduced during patch retrieval, since even state-of-the-art depth estimation contains small errors that can cause significant distortion when pixels are warped directly. To assess this, in Sec. 3.2, we evaluate MLLM's sensitivity to retrieval-position noise by perturbing the regular grid center points used to fetch local patches. Specifically, we retrieve each patch from a slightly shifted center position, introducing a controlled offset during patch extraction. This experiment reveals that image tokens are robust to positional noise, making them well suited for reliable geometric transformation under viewpoint changes.

Building on this insight, we search for the best token-level warping strategy in Sec. 3.3 by exploring several warping functions and analyzing how well each preserves structural coherence and semantic consistency under viewpoint shifts.

3.1. Image Tokenization in MLLMs

In MLLMs, an image $\mathbf{I}$ is partitioned into a fixed, non-overlapping grid of patches $\{\mathbf{u}_i\}_{i=1}^{M}$ (Fig. 2). Each patch $\mathbf{u}_i \in \mathbb{R}^{l\times l\times 3}$ corresponds to a square region of $\mathbf{I}$ associated with a grid-center coordinate $\mathbf{c}_i = (x_i, y_i)$ on $\mathbf{I}$'s lattice. A shallow encoder $\mathcal{E}$ maps each patch to an embedding $\mathbf{e}_i = \mathcal{E}(\mathbf{u}_i)$. These embeddings, together with their grid-center coordinates, are processed by a vision encoder $\mathcal{V}$ (*e.g.*, ViT [24, 94]) to produce image tokens $\{\mathbf{v}_i\}_{i=1}^{M} = \mathcal{V}\big(\{(\mathbf{e}_i, \mathbf{c}_i)\}_{i=1}^{M}\big)$, which are then projected into

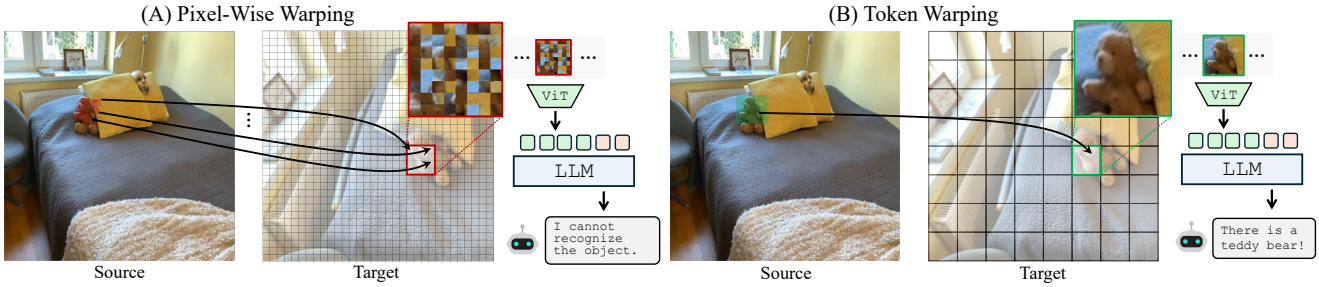

Figure 4. **Pixel-Wise vs. Token Warping.** Comparison of inverse warping strategies (Sec. 3.3). (A) *Pixel-wise warping* retrieves pixels for each target coordinate, but patchifying the warped image introduces local distortions, resulting in degraded MLLM understanding. (B) *Token warping* directly retrieves intact tokens (or patches) from the source view, preserving semantics and improving viewpoint-aware perception.

the LLM's latent space and processed alongside text tokens. Notably, each token carries not only semantic information encoded from its pixel values but also positional information defined at the patch level as a whole. We hypothesize that transferring tokens rather than individual pixels is therefore more robust to noise in positional information, as we empirically demonstrate below.

3.2. Fetching Position Noise Sensitivity Test

As hypothesized earlier, image tokens serve as perceptual atoms in MLLMs well suited for simulating viewpoint changes through warping: they naturally encode locality-aware features and propagate as coherent units during the warping operation.

To demonstrate this, we begin with a simple proof-of-concept experiment that perturbs the positional information of MLLM tokens via jittering. Further comparisons against pixel-based representations in actual viewpoint change scenarios are presented in Sec. 5. Specifically, consider each

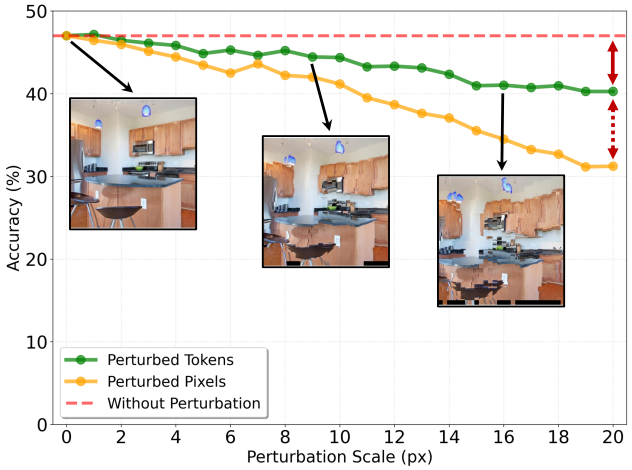

Figure 5. **Fetching Position Noise Sensitivity (Sec. 3.2).** Through a toy experiment on CV-Bench-2D [93], where we emulate local positional perturbations and degradation introduced by warping, we find that token representations in MLLMs are highly robust to noise in the image positions from which tokens are fetched. This suggests that tokens are well suited for representing viewpoint changes.

token $\mathbf{v}_i$ from image $\mathbf{I}$ together with its grid-center coordinate $\mathbf{c}_i$, which determines its positional embedding. For each token, we sample a displacement vector $\mathbf{u}_i = (\Delta x_i, \Delta y_i)$ from standard Gaussian distribution and apply mean-filter smoothing over neighboring cells. We then normalize all $\mathbf{u}_i$ by the global maximum magnitude and scale by a hyperparameter, the *maximum displacement value*. We vary this value from 0.0 to 20.0 and fix the smoothing neighborhood to 9 grid cells. This procedure is designed to emulate the noisy positional perturbations introduced during warping. As a pixel-level baseline, we apply the same jittering process and add slight pixel-wise perturbation (*i.e.*, 10% of each maximum displacement value) to emulate pixel-level perturbations in pixel-wise warping.

Fig. 5 shows Qwen2.5-VL's [6] accuracy on CV-Bench-2D [93] VQA tasks under varying maximum displacement values for token position perturbations (green plot). The model maintains consistent performance across perturbation levels from 0 to 20.0. Notably, it exhibits only mild degradation in the large-perturbation regime (19.0-20.0 pixels), where the perturbation artifacts become visually apparent (top-right example in Fig. 5). Compared with the pixel-level baseline (orange), token-level representations are clearly more robust under similar level of perturbations. This result highlights the importance of preserving localized, semantically meaningful visual elements in perceptual tasks, consistent with classical discussions on part-level structures in mental imagery [34, 67, 75, 86]. Motivated by this finding, we adopt *tokens* as the units during warping, as detailed in the following section.

3.3. Designing Token Warping Functions

Building on our observation regarding the robustness of tokens, we now turn to a spatial reasoning task involving two viewpoints. In this setting, the model is given an observed *source* viewpoint and an unobserved *target* viewpoint, together with a question that requires imagining how the scene would appear from the target viewpoint in order to answer. Formally, let $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ denote the RGB image captured from the *source* viewpoint with camera pose matrix

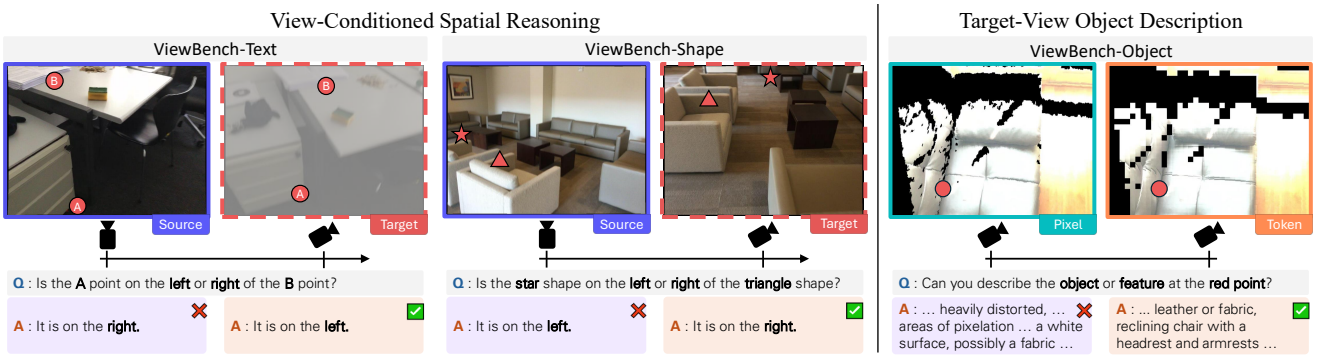

Figure 6. **`ViewBench`.** Example source-target image pairs with corresponding questions and answers from our `ViewBench` benchmark. The tasks evaluate MLLM's ability to infer spatial relationships from nearby viewpoints (`Text`, `Shape`), while also measuring robustness to view changes by asking to describe object properties visible in the warped target view (`Object`).

$\Pi_S \in \mathbb{R}^{4\times4}$, representing the world-to-camera transformation. The question Q is a natural language query about the scene depicted in $\mathbf{I}$, but posed from the perspective of a *target* viewpoint with camera pose $\Pi_T \in \mathbb{R}^{4\times4}$. We further assume that a depth map $\mathbf{D} \in \mathbb{R}^{H\times W\times1}$ corresponding to $\mathbf{I}$ is available, either as ground truth or estimated via monocular depth estimation [108], along with the intrinsic matrix $\mathbf{K} \in \mathbb{R}^{4\times4}$.

Given the above, the most direct strategy for answering Q is to *warp* the source image $\mathbf{I}$, along with the tokens encoded from it, into the target viewpoint using the depth map $\mathbf{D}$, the intrinsic matrix $\mathbf{K}$, and the relative pose $\Pi_{S\to T} = \Pi_T \Pi_S^{-1}$. Let $\mathbf{c} \in \mathbb{R}^{(HW)\times2}$ denote the grid-center coordinates of $\mathbf{I}$. The corresponding coordinates after warping, $\mathbf{c}^* \in \mathbb{R}^{(HW)\times2}$, are computed as:

$$\mathbf{c}^* = f_{S\to T}(\mathbf{c}, \Pi_{S\to T}, \mathbf{K}, \mathbf{D}), \tag{3.1}$$

where $f_{S\to T} : \mathbb{R}^{(HW)\times2} \to \mathbb{R}^{(HW)\times2}$ denotes the forward-warping function that projects token positions from the source to the target viewpoint. Conversely, we can define the backward mapping $f_{T\to S}$, which takes grid-center coordinates at the *target* viewpoint and computes their corresponding coordinates on the *source* image plane.

In this work, we explore both as candidates for token warping: either through direct forward projection ($f_{S\to T}$) or by fetching corresponding source tokens via backward projection ($f_{T\to S}$). Beyond these two approaches for determining *which* coordinates to fetch, we further investigate *how* to fetch them, considering both nearest and adaptive fetching strategies.

Forward vs. Backward Warping. *Forward warping* projects tokens from $\mathbf{I}$ into the target viewpoint via $f_{S\to T}$ and computes their positional embeddings accordingly. Despite its simplicity, this approach often yields irregular, sparse token distributions with large holes across the target image plane. As we later show in Sec. 5, such irregular and sparsely placed tokens are out-of-distribution inputs for an MLLM trained on dense, regularly spaced token grids, leading to substantial performance degradation. *Backward warping* takes the opposite strategy: we first define a dense, regular grid in the target view and retrieve the corresponding tokens from $\mathbf{I}$ via the mapping $f_{T\to S}$. For this, we build a lightweight 3D proxy mesh from the source image's depth map and compute the mapping from each target grid to the source via ray casting. Implementation details are provided in **the supplementary material**. Unlike forward warping, this approach produces tokens that are, by construction, regularly placed on the target image plane. We thus adopt backward warping as our primary strategy, which consistently outperforms forward warping in our experiments (Sec. 5).

Nearest vs. Adaptive Fetching. A further design consideration is how to fetch tokens from the coordinates produced by $f_{T\to S}$, as these coordinates often fall between token grid centers on the source image plane. Recall that each token originates from a fixed-grid patch (Sec. 3.1). We explore two strategies to address this gap: nearest and adaptive fetching. In *nearest fetching*, given a mapped coordinate $\mathbf{c}_i^*$ from Eq. 3.1, we retrieve the token associated with the nearest grid-center point in Euclidean distance (Fig. 7-(A)). In *adaptive fetching*, the source image $\mathbf{I}$ is re-patchified according to the warped coordinates: for each $\mathbf{c}_i^*$, a patch centered at $\mathbf{c}_i^*$ is cropped and encoded through the same token encoding

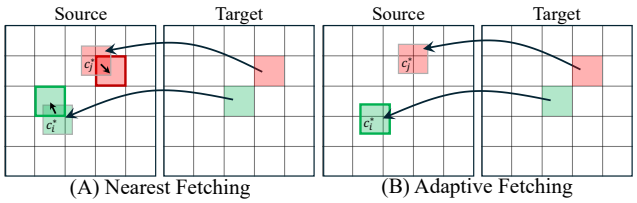

Figure 7. **Token Fetching Strategies.** (A) *Nearest fetching* selects the closest existing token from the source image grid. (B) *Adaptive fetching* dynamically crops a patch centered at the mapped coordinate to derive a token precisely centered at the target location.

	ViewBench-Text (%)						ViewBench-Shape (%)						ViewBench-Object (1-10)					
View Overlap (%)	5–15		15–25		25–35		5–15		15–25		25–35		5–15		15–25		25–35	
Depth	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.
Target View (Oracle)	100.00	–	100.00	–	100.00	-	100.00	–	100.00	–	100.00	-	6.64	–	7.31	–	7.43	–
Specialist MLLMs																		
SpatialReasoner [61]	46.73	–	53.30	–	53.71	–	33.72	–	38.27	–	48.15	–	–	–	–	–	–	–
VLM-3R [26]	63.82	–	70.56	–	60.57	–	49.22	–	49.79	–	50.21	–	–	–	–	–	–	–
ViLaSR [102]	44.22	–	52.28	–	48.00	–	22.87	–	23.05	–	34.57	–	–	–	–	–	–	–
Qwen2.5-VL [6]	46.23	–	59.39	–	52.00	–	24.42	–	25.10	–	37.86	–	–	–	–	–	–	–
Novel View Synthesis																		
GenWarp [85]	69.35	–	71.07	–	66.29	–	53.10	–	47.33	–	55.14	–	4.32	–	4.81	–	4.34	–
Pixel-Wise Warping																		
Forward	70.85	69.35	73.60	73.10	62.86	67.43	56.20	56.20	56.79	56.79	60.49	60.08	3.22	3.22	4.04	3.87	4.78	4.54
Backward	71.86	67.84	75.63	74.62	68.57	68.57	62.40	58.14	58.02	56.79	66.67	64.20	4.53	4.45	5.52	5.48	5.94	5.89
Token Warping																		
Forward	60.30	66.83	64.47	65.48	54.86	60.57	55.04	56.98	55.14	60.91	53.09	56.38	4.09	4.20	4.27	4.37	4.07	3.78
Backward-Nearest	74.87	**75.38**	**80.71**	**81.73**	74.86	**76.00**	**67.44**	63.95	62.96	**62.55**	73.25	**75.31**	4.80	4.86	5.39	5.57	**6.19**	5.97
Backward-Adaptive	**77.89**	73.37	79.70	80.71	**78.86**	74.29	**67.44**	**66.28**	**66.26**	61.32	**75.72**	70.37	**4.97**	**5.18**	**5.76**	**6.29**	6.11	**6.14**

Table 1. **Quantitative Comparisons on `ViewBench`.** The prediction accuracies of the models on the spatial reasoning tasks (`ViewBench-Text` and `ViewBench-Shape`) are reported in columns 2–13. The performance scores for the target-view object description task (`ViewBench-Object`), evaluated by Qwen2.5-VL 14B [6] on a 1–10 scale, are summarized in columns 14–19. Across all tasks and setups, backward token-wise warping achieves the best performance.

process shown in Fig. 2. This allows tokens to be centered at arbitrary locations beyond the constraints of a fixed patch grid. Fig. 7 provides a visual comparison of the two fetching strategies, and algorithmic details are provided in **the supplementary material**. In our experiments (Sec. 5), we find that nearest fetching performs comparably to adaptive fetching, despite the latter requiring additional computation for re-patchification.

4. ViewBench

In this section, we introduce `ViewBench`, a benchmark designed to assess MLLMs' ability to perform spatial reasoning tasks that require imagining a scene from alternative viewpoints while accurately transferring fine-grained details from the observed viewpoint.

Data. To construct source–target viewpoint pairs for generating VQAs, we collect image pairs captured from adjacent viewpoints with overlapping fields of view, drawn from real-world scans in ScanNet [19]. The collected pairs are divided into difficulty levels based on their overlap ratios [104], which reflect the amount of shared content between the two views. For each pair, one viewpoint is designated as the source, with image $\mathbf{I}_S$ and pose Π_S, and the other as the target, with image $\mathbf{I}_T$ and pose Π_T. We then generate a question Q answerable only from the target viewpoint, using information available in the source view together with an instruction describing the relative pose change between the two viewpoints. Importantly, we ensure that Q refers only to regions visible in both views, avoiding content that is occluded or unseen from the target view.

Tasks. The form of Q depends on the specific task. We design two tasks, both tailored to evaluate an MLLM's ability to simulate viewpoint changes for spatial reasoning: (1) view-conditioned spatial reasoning and (2) target-view object description.

- **View-Conditioned Spatial Reasoning.** This task evaluates whether an MLLM can reason about spatial relationships from a transformed viewpoint. To construct Q, we identify two points visible in both $\mathbf{I}_S$ and $\mathbf{I}_T$ whose left-right spatial relationship is reversed after the viewpoint change. These points are annotated using either text labels (`ViewBench-Text`) or simple geometric shapes (`ViewBench-Shape`), and Q asks whether one point appears to the left or right of the other when viewed from the target viewpoint.
- **Target-View Object Description.** This task assesses whether an MLLM can accurately describe an object from the source image as it would appear from the target viewpoint, testing its ability to preserve semantic fidelity and fine-grained visual details—a capability that is often challenging to achieve with pixel-wise warping. As in the previous task, we identify two points visible in both $\mathbf{I}_S$ and $\mathbf{I}_T$ to construct Q, which asks the MLLM to describe an object, or a specific visual attribute of it, at the annotated position.

Examples from our `ViewBench` are shown in Fig. 6. Additional details on the benchmark construction are provided in **the supplementary material**.

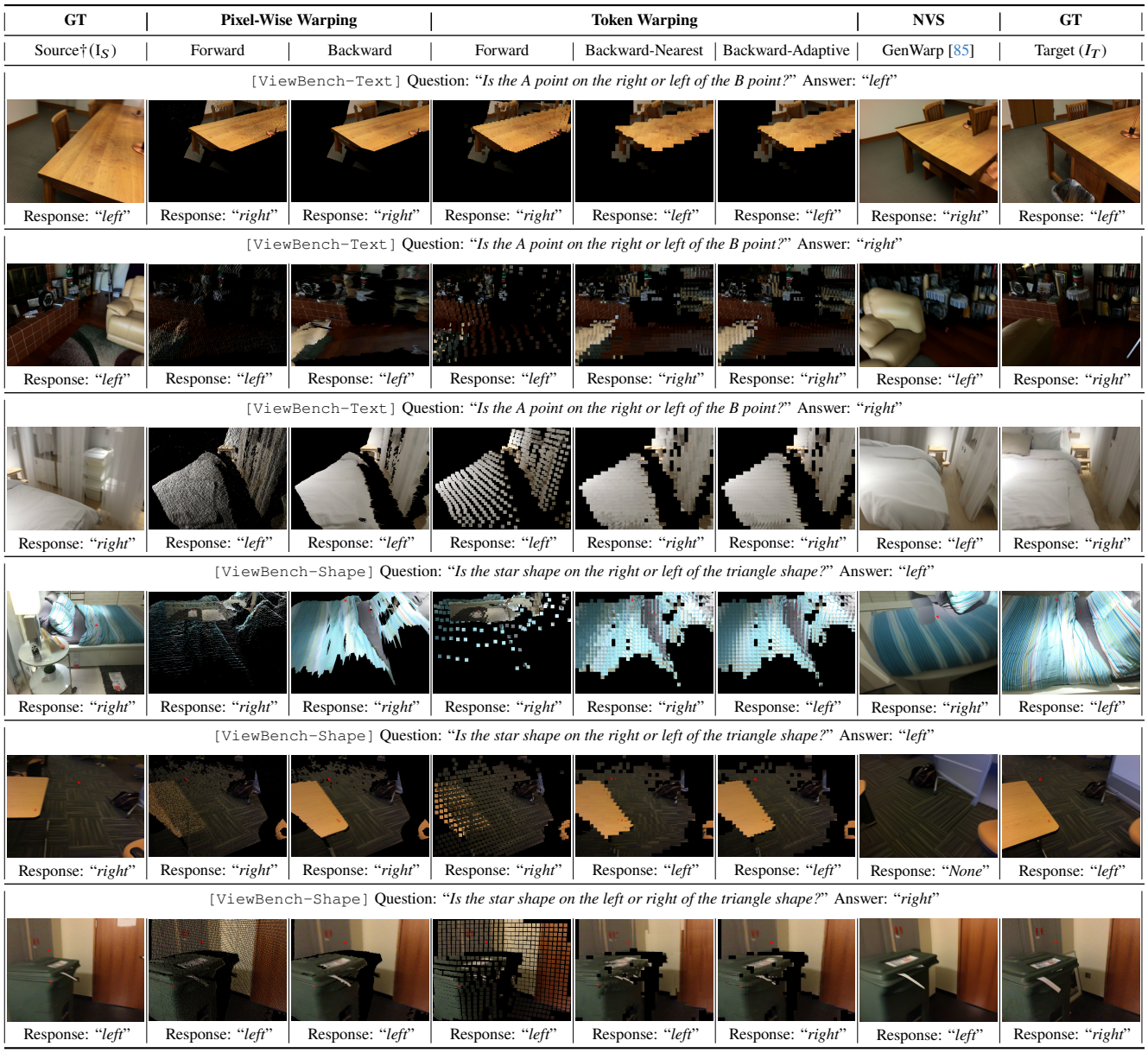

Figure 8. **Warping Visualizations.** We compare the warped results of pixel-wise warping, token warping, and the generative NVS output [85]. The rightmost image shows the ground-truth target viewpoint. For token warping, we visualize the RGB image patches corresponding to each token for illustration only. Above each row, we provide the question Q from `ViewBench`, and below each image we show the response from Qwen2.5-VL [6] when given the corresponding warped result. †The camera motion from the source view to the target view is additionally supplied as part of the prompt.

Metrics. For quantitative evaluation in the view-conditioned spatial reasoning task with binary labels—left or right—we report accuracy (%), defined as the proportion of correctly answered VQA queries. For the target-view object description task, we employ Qwen2.5-14B [6] as an evaluator and ask it to rate the generated responses on a scale from 1 to 10. We compute the score for each example in our benchmark suite and report the average score as the performance metric. As a barometer for the reported metrics, we additionally compute and report an oracle performance metric obtained by using the ground-truth target-view image when answering the VQA queries. Specifically, for the `Text` and `Shape` subsets, we retained only those data pairs on which the oracle was correct, yielding 571 and 744 pairs, respectively, for evaluation. We used 300 pairs for `Object`.

5. Evaluation

We evaluate the token warping techniques from Sec. 3 on `ViewBench`, with baselines summarized in Sec. 5.1. Results for view-conditioned spatial reasoning and target-view object description are presented in Sec. 5.2 and Sec. 5.3, respectively.

5.1. Baselines

We compare token warping against pixel-wise warping variants and external baselines. For our framework, its variants, and the generative warping baseline, we use Qwen2.5-VL-7B [6] as the base MLLM. We implement both forward and backward pixel-wise warping, along with three variants of token warping, denoted *Forward*, *Backward-Nearest*, and *Backward-Adaptive*, respectively. These methods introduce minimal inference-time overhead for warping during inference without requiring extra fine-tuning. In addition, we include specialized MLLMs fine-tuned on spatial reasoning datasets, such as SpatialReasoner [61], ViLaSR [102], and VLM-3R [26]. For these models, we provide the original source view together with an additional text prompt that explicitly describes the relative camera motion from the source to the target view. Lastly, we employ GenWarp [85], a camera-conditioned diffusion model that uses implicit warping for novel view synthesis, to directly generate an RGB image at the target viewpoint and then pass it to Qwen2.5-VL [6] for querying. We provide comparisons against additional baselines in **the supplementary material**.

5.2. View-Conditioned Spatial Reasoning

The quantitative results for the view-conditioned spatial reasoning task, including `ViewBench-Text` and `ViewBench-Shape`, are presented in columns 2–13 of Tab. 1. As shown in the rows highlighted in gray, backward token warping, regardless of the fetching strategy (nearest or adaptive), consistently outperforms forward token warping across all overlapping ratios. For example, in the most challenging settings, `ViewBench-Text` (5–15) and `ViewBench-Shape` (5–15), where the source and target viewpoints share only minimal overlap, the Backward-Nearest variant improves accuracy by 14.57%p and 12.4%p, respectively, when ground-truth depth maps are used for warping. Similar trends are observed across all other configurations, highlighting that providing dense and regular positional embeddings to MLLMs is crucial for maintaining high performance under viewpoint changes, consistent with our analysis in Sec. 3.3. In addition, we observe that the simple nearest-fetching strategy performs on par with the adaptive variant—an effect we attribute to the robustness of token-level representations, which naturally preserve local semantics by treating groups of pixels as coherent units. When compared against the pixel-wise warping variants (rows highlighted in red), the specialized MLLMs (rows highlighted in blue), and the generative warping baseline (row highlighted in green), our token-wise warping approach consistently outperforms all of them. Notably, VLM-3R [26], which incorporates features from CUT3R [99], still remains behind backward token warping, indicating that rich features alone do not equip models with the capacity to mentally shift viewpoints.

Qualitative examples in rows 1–4 of Fig. 8 provide a visual explanation of this trend. Note that the pixelated images in the **Token Warping** columns are displayed solely for visualization; our framework operates entirely on token embeddings. In contrast, pixel-wise warping baselines feed the warped images, such as those illustrated in the figure, into the MLLM's vision encoder. As shown in row 2 of Fig. 8, pixel-wise warping introduces severe visual artifacts during both forward and backward warping, yielding incorrect predictions (*e.g.*, "left"). Even a generative approach [85] does not fully resolve these issues, as it may hallucinate non-existent objects or lose existing ones. For instance, Fig. 8 row 5 shows that the simple shapes in the input image are omitted in the output of GenWarp, therefore leading to the response "none". In contrast, backward token-warping–based approaches consistently produce the correct answer. We provide qualitative results for the warped images as well as the descriptions generated by MLLMs in **the supplementary material**.

5.3. Target-View Object Description

We summarize the quantitative results for the target-view object description task (`ViewBench-Object`) in columns 14–19 of Tab. 1. Consistent with our analysis in Sec. 5.2, among the token-warping methods highlighted in gray rows, backward warping approaches outperform their forward-warping counterpart, as reflected in higher scores from the MLLM evaluator. The same trend holds when comparing token-warping approaches against the pixel-wise warping baselines (shown in red) and the generative warping baseline (shown in green). We report qualitative results for warped images, and the descriptions generated by MLLMs in **the supplementary material**.

6. Conclusion

In this work, inspired by classic discussions on part-based representations for mental imagery [34, 67, 75, 86], we explored token warping as a simple yet effective strategy for transferring source view observations to nearby novel viewpoints. By comparing different token warping directions (*forward* vs. *backward*) and backward token fetching techniques (*adaptive* vs. *nearest*), we found that constructing a regular, dense grid of tokens via backward warping is crucial for robust MLLM performance. Notably, simple nearest fetching performs comparably to the more sophisticated adaptive fetching, offering a practical and efficient solution.

Acknowledgements

We thank Daehyeon Choi and Sangwoo Youn for their valuable discussions. This work was supported by the National Research Foundation of Korea (NRF) (RS-2026-25486000); the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants (RS-2019-II190075, RS-2022-00156435, RS-2024-00399817, RS-2025-25441313, RS-2025-25443318), funded by the Korean government (MSIT); the Industrial Technology Innovation Program (RS-2025-02317326), funded by the Korean government (MOTIE); the National Supercomputing Center (KSC-2025-CRE-0475); and the DRB-KAIST SketchTheFuture Research Center.

References

[1] Remyx AI. Spaceqwen2.5-vl-3b-instruct. https://huggingface.co/remyxai/SpaceQwen2.5-VL-3B-Instruct, 2025. 15

[2] Remyx AI. Spacethinker-qwen2.5vl-3b. https://huggingface.co/remyxai/SpaceThinker-Qwen2.5VL-3B, 2025. 15

[3] Remyx AI. Vqasynth. https://github.com/remyxai/VQASynth, 2025. 15

[4] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025. 15, 16

[5] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, 2022. 3

[6] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 4, 6, 7, 8, 15, 16, 17, 22

[7] Yunpeng Bai, Haoxiang Li, and Qixing Huang. Positional encoding field. In *ICLR*, 2026. 3

[8] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *NeurIPS*, 2021. 18

[9] Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G Shapiro, and Ranjay Krishna. Perception tokens enhance visual reasoning in multimodal language models. In *CVPR*, 2025. 3

[10] Aleksei Bochkovskii, AmaÃĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *ICLR*, 2025. 1, 15

[11] Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *ICRA*, 2025. 2

[12] 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 *CVPR*, 2024. 2, 15, 16

[13] Delong Chen, Samuel Cahyawijaya, Jianfeng Liu, Baoyuan Wang, and Pascale Fung. Subobject-level image tokenization. In *ICML*, 2025. 3

[14] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *CVPR*, 2024. 2

[15] Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Yan Feng, Peng Pei, Xunliang Cai, and Ruqi Huang. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. *arXiv preprint arXiv:2510.18632*, 2025. 1, 3

[16] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *NeurIPS*, 2024. 2

[17] An-Chieh Cheng, Yang Fu, Yukang Chen, Zhijian Liu, Xiaolong Li, Subhashree Radhakrishnan, Song Han, Yao Lu, Jan Kautz, Pavlo Molchanov, et al. 3d aware region prompted vision language model. In *ICLR*, 2026. 2

[18] Rohan Choudhury, JungEun Kim, Jinhyung Park, Eunho Yang, László A Jeni, and Kris M Kitani. Accelerating vision transformers with adaptive patch sizes. *arXiv preprint arXiv:2510.18091*, 2025. 3

[19] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 6, 16, 21

[20] Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, et al. Mm-spatial: Exploring 3d spatial understanding in multimodal llms. In *ICCV*, 2025. 2

[21] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *NeurIPS*, 2022. Outstanding Paper Award. 17

[22] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, 2025. 2

[23] Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3d-llava: Towards generalist 3d lmms with omni superpoint transformer. In *CVPR*, 2025. 2

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3

[25] 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. In *ICML*, 2023. 2

[26] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. *arXiv preprint arXiv:2505.20279*, 2025. 1, 2, 3, 6, 8, 15, 16

[27] Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*, 2025. 3

[28] RA Finke. Principles of mental imagery, 1989. 1

[29] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual reasoning. In *WACV*, 2025. 2

[30] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *ECCV*, 2024. 2, 19

[31] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 3

[32] Gracjan Góral, Alicja Ziarko, Michal Nauman, and Maciej Wołczyk. Seeing through their eyes: Evaluating visual perspective taking in vision language models. *arXiv preprint arXiv:2409.12969*, 2024. 3

[33] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 15

[34] Geoffrey Hinton. Some demonstrations of the effects of structural descriptions in mental imagery. *Cognitive Science*, 3(3):231–250, 1979. 1, 2, 3, 4, 8

[35] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. In *NeurIPS*, 2023. 2

[36] Wenbo Hu, Yining Hong, Yanjun Wang, Leison Gao, Zibu Wei, Xingcheng Yao, Nanyun Peng, Yonatan Bitton, Idan Szpektor, and Kai-Wei Chang. 3dllm-mem: Long-term spatial-temporal memory for embodied 3d large language model. In *NeurIPS*, 2025. 2

[37] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *ICML*, 2024. 2

[38] Ting Huang, Zeyu Zhang, and Hao Tang. 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. *arXiv preprint arXiv:2507.23478*, 2025. 2

[39] Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. Mllms need 3d-aware representation supervision for scene understanding. In *NeurIPS*, 2025. 2

[40] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *CVPR*, 2025. 2

[41] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *CVPR*, 2023. 3

[42] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 3

[43] Juil Koo, Paul Guerrero, Chun-Hao P Huang, Duygu Ceylan, and Minhyuk Sung. Videohandles: Editing 3d object compositions in videos using video generative priors. In *CVPR*, 2025. 3

[44] S. M. Kosslyn, T. M. Ball, and B. J. Reiser. Visual images preserve metric spatial information: Evidence from studies of image scanning. *Journal of Experimental Psychology: Human Perception and Performance*, 1978. 1

[45] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 3

[46] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025. 3

[47] Phillip Y. Lee, Taehoon Yoon, and Minhyuk Sung. Groundit: Grounding diffusion transformers via noisy patch transplantation. In *NeurIPS*, 2024. 3

[48] Phillip Y. Lee, Jihyeon Je, Chanho Park, Mikaela Angelina Uy, Leonidas Guibas, and Minhyuk Sung. Perspective-aware reasoning in vision-language models via mental imagery simulation. In *ICCV*, 2025. 1, 3

[49] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *TMLR*, 2025. 2

[50] Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, et al. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models. *arXiv preprint arXiv:2505.21500*, 2025. 3

[51] Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. Spatialladder: Progressive training for spatial reasoning in vision-language models. In *ICLR*, 2025. 2, 15, 16

[52] Pengteng Li, Pinhao Song, Wuyang Li, Weiyu Guo, Huizai Yao, Yijie Xu, Dugang Liu, and Hui Xiong. See&trek: Training-free spatial prompting for multimodal large language model. In *NeurIPS*, 2025. 2

[53] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023. 3

[54] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 19

[55] Drew Linsley, Peisen Zhou, Alekh Karkada Ashok, Akash Nagaraj, Gaurav Gaonkar, Francis E Lewis, Zygmunt Pizlo, and Thomas Serre. The 3d-pc: a benchmark for visual perspective taking in humans and machines. In *ICLR*, 2025. 3

[56] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 3

[57] Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanguang Zhang, Yaochen Hu, Lingfeng Zhang, Yingxue Zhang, Shuang Wu, Tongtong Cao, Guowei Huang, et al. Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning. *arXiv preprint arXiv:2501.10074*, 2025. 2

[58] Gen Luo, Ganlin Yang, Ziyang Gong, Guanzhou Chen, Haonan Duan, Erfei Cui, Ronglei Tong, Zhi Hou, Tianyi Zhang, Zhe Chen, et al. Visual embodied brain: Let multimodal

large language models see, think, and control in spaces. *arXiv preprint arXiv:2506.00123*, 2025. 15, 16

[59] Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. In *NeurIPS*, 2024. 2

[60] Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso de Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. In *ICCV*, 2025. 2, 3

[61] Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. In *NeurIPS*, 2025. 1, 2, 6, 8, 16

[62] Wufei Ma, Luoxin Ye, Celso M de Melo, Alan Yuille, and Jieneng Chen. Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models. In *CVPR*, 2025. 2

[63] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. In *ICLR*, 2022. 3

[64] Bui Duc Manh, Soumyaratna Debnath, Zetong Zhang, Shriram Damodaran, Arvind Kumar, Yueyi Zhang, Lu Mi, Erik Cambria, and Lin Wang. Mind meets space: Rethinking agentic spatial intelligence from a neuroscience-inspired perspective. *arXiv preprint arXiv:2509.09154*, 2025. 3

[65] Damiano Marsili, Rohun Agrawal, Yisong Yue, and Georgia Gkioxari. Visual agentic ai for spatial reasoning with a dynamic api. In *CVPR*, 2025. 2

[66] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *ECCV*, 2022. 3

[67] Marvin Minsky et al. A framework for representing knowledge, 1974. 2, 3, 4, 8

[68] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. In *NeurIPS*, 2023. 2

[69] Bence Nanay. Mental imagery. *The Stanford Encyclopedia of Philosophy*, 2021. 1

[70] Fei Ni, Min Zhang, Pengyi Li, Yifu Yuan, Lingfeng Zhang, Yuecheng Liu, Peilong Han, Longxin Kou, Shaojin Ma, Jinbin Qiao, et al. Embodied arena: A comprehensive, unified, and evolving evaluation platform for embodied ai. *arXiv preprint arXiv:2509.15273*, 2025. 3

[71] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2023. 3

[72] Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025. 2, 15, 16

[73] A. Paivio. *Imagery and Verbal Processes (1st ed.)*. Psychology Press, 1979. 1

[74] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 3

[75] Zenon W Pylyshyn. What the mind's eye tells the mind's brain: A critique of mental imagery. *Psychological bulletin*, 1973. 2, 3, 4, 8

[76] Zhangyang Qi, Zhixiong Zhang, Yizhou Yu, Jiaqi Wang, and Hengshuang Zhao. Vln-r1: Vision-language navigation via reinforcement fine-tuning. *arXiv preprint arXiv:2506.17221*, 2025. 3

[77] Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. In *ICLR*, 2026. 2

[78] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *CVPR*, 2025. 3

[79] Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *ACCV*, 2024. 2

[80] Santhosh Kumar Ramakrishnan, Erik Wijmans, Philipp Kraehenbuehl, and Vladlen Koltun. Does spatial cognition emerge in frontier models? In *ICLR*, 2025. 1, 2

[81] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 3

[82] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025. 3

[83] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[84] Tomer Ronen, Omer Levy, and Avram Golbert. Vision transformers with mixed-resolution tokenization. In *CVPR*, 2023. 3

[85] Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. Genwarp: Single image to novel views with semantic-preserving generative warping. In *NeurIPS*, 2024. 6, 7, 8, 16

[86] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 1, 2, 3, 4, 8

[87] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *CVPR*, 2025. 2, 3

[88] Yihong Tang, Ao Qu, Zhaokai Wang, Dingyi Zhuang, Zhaofeng Wu, Wei Ma, Shenhao Wang, Yunhan Zheng, Zhan Zhao, and Jinhua Zhao. Sparkle: Mastering basic spatial capabilities in vision language models elicits generalization to composite spatial reasoning. In *EMNLP*, 2025. 2

[89] BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei

Wang, Enshen Zhou, et al. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025. 15, 16

[90] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025. 15, 16

[91] Anh Thai, Songyou Peng, Kyle Genova, Leonidas Guibas, and Thomas Funkhouser. Splattalk: 3d vqa with gaussian splatting. In *ICCV*, 2025. 2

[92] Edward Chace Tolman. Cognitive maps in rats and men. *Psychological review*, 55 4:189–208, 1948. 1

[93] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, 2024. 2, 4, 15, 16

[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[95] Haochen Wang, Yucheng Zhao, Tiancai Wang, Haoqiang Fan, Xiangyu Zhang, and Zhaoxiang Zhang. Ross3d: Reconstructive visual instruction tuning with 3d-awareness. In *ICCV*, 2025. 2

[96] Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Sharon Li, and Neel Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. In *NeurIPS*, 2024. 2

[97] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025. 1, 3, 15, 16, 17

[98] Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, et al. Vagen: Reinforcing world model reasoning for multi-turn vlm agents. In *NeurIPS*, 2025. 3

[99] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 1, 8, 15

[100] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 3, 16

[101] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. In *NeurIPS*, 2025. 2

[102] Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. In *NeurIPS*, 2025. 3, 6, 8, 16

[103] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *CVPR*, 2022. 3

[104] Runsen Xu, Weiyao Wang, Hao Tang, Xingyu Chen, Xiaodong Wang, Fu-Jen Chu, Dahua Lin, Matt Feiszli, and Kevin J Liang. Multi-spatialmllm: Multi-frame spatial understanding with multi-modal large language models. *arXiv preprint arXiv:2505.17015*, 2025. 3, 6, 21

[105] Yu Xu, Fan Tang, Juan Cao, Xiaoyu Kong, Yuxin Zhang, Jintao Li, Oliver Deussen, and Tong-Yee Lee. Headrouter: A training-free image editing framework for mm-dits by adaptively routing attention heads. *ACM TOG*, 2024. 3

[106] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 15, 16

[107] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *CVPR*, 2025. 3

[108] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024. 1, 5, 15, 21

[109] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *ICML*, 2025. 3

[110] Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, et al. Visual spatial tuning. *arXiv preprint arXiv:2511.05491*, 2025. 15, 16

[111] Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, et al. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *ICLR*, 2026. 3

[112] Yuncong Yang, Jiageng Liu, Zheyuan Zhang, Siyuan Zhou, Reuben Tan, Jianwei Yang, Yilun Du, and Chuang Gan. Mindjourney: Test-time scaling with world models for spatial reasoning. In *NeurIPS*, 2025. 2

[113] Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *ICLR*, 2026. 1, 3, 15, 16

[114] Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. Inst3d-lmm: Instance-aware 3d scene understanding with multi-modal instruction tuning. In *CVPR*, 2025. 2

[115] Runpeng Yu, Xinyin Ma, and Xinchao Wang. Introducing visual perception token into multimodal large language model. *arXiv preprint arXiv:2502.17425*, 2025. 3

[116] Songsong Yu, Yuxin Chen, Hao Ju, Lianjie Jia, Fuxi Zhang, Shaofei Huang, Yuhan Wu, Rundi Cui, Binghao Ran, Zaibin Zhang, et al. How far are vlms from visual spatial intelligence? a benchmark-driven perspective. *arXiv preprint arXiv:2509.18905*, 2025. 2

[117] Zhihao Yuan, Shuyi Jiang, Chun-Mei Feng, Yaolun Zhang, Shuguang Cui, Zhen Li, and Na Zhao. Scene-r1: Video-grounded large language models for 3d scene reasoning without 3d annotations. *arXiv preprint arXiv:2506.17545*, 2025. 2

[118] Haoyu Zhang, Meng Liu, Zaijing Li, Haokun Wen, Weili Guan, Yaowei Wang, and Liqiang Nie. Spatial understanding from videos: Structured prompts meet simulation data. In *NeurIPS*, 2025. 2

[119] Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. In *NeurIPS*, 2025. 3

[120] Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *ICLR*, 2026. 2, 3

[121] Wenyu Zhang, Wei En Ng, Lixin Ma, Yuwen Wang, Junqi Zhao, Allison Koenecke, Boyang Li, and Wanglu Wanglu. Sphere: Unveiling spatial blind spots in vision-language models through hierarchical evaluation. In *ACL*, 2025. 3

[122] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 3

[123] Yuyou Zhang, Radu Corcodel, Chiori Hori, Anoop Cherian, and Ding Zhao. Spinbench: Perspective and rotation as a lens on spatial reasoning in vlms. In *ICLR*, 2026. 1, 3

[124] Zheyuan Zhang, Fengyuan Hu, Jayjun Lee, Freda Shi, Parisa Kordjamshidi, Joyce Chai, and Ziqiao Ma. Do vision-language models represent space and how? evaluating spatial frame of reference under ambiguities. In *ICLR*, 2025. 3

[125] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. In *NeurIPS*, 2025. 1, 2, 15, 16

[126] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *CVPR*, 2025. 2

[127] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. In *ICCV*, 2025. 2

[128] Fangrui Zhu, Hanhui Wang, Yiming Xie, Jing Gu, Tianye Ding, Jianwei Yang, and Huaizu Jiang. Struct2d: A perception-guided framework for spatial reasoning in large multimodal models. In *NeurIPS*, 2025. 2

[129] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 15, 16

Token Warping Helps MLLMs Look from Nearby Viewpoints

Supplementary Material

In this supplementary material, we report additional experimental results with more baseline MLLMs and showcase qualitative examples of warped visualizations with corresponding MLLM responses (Sec. A). We then present implementation and algorithmic details of *backward token warping* with *nearest* and *adaptive* fetching (Sec. B). Finally, we describe the step-by-step data construction pipeline of `ViewBench` in Sec. C.

A. Additional Results

This section presents additional experiments: extended comparisons with specialized MLLMs (Sec. A.1), robustness analysis under estimated geometry (Sec. A.2), evaluation under extreme viewpoint shifts and occlusion (Sec. A.3), a geometry-based oracle analysis (Sec. A.4), and qualitative examples (Sec. A.5).

A.1. Comparison with Additional Baselines

Extending Tab. 1 **of the main paper**, we report a more extensive quantitative comparison against a wider range of specialist and general-purpose MLLMs.

Baselines. We include recent open-source MLLMs: *Qwen3-VL* [106], *InternVL3* [129], *Cambrian-1* [93], *LLaVA-OneVision-1.5* [4], and *Kimi-VL-Thinking* [90]. We further include models explicitly fine-tuned for spatial reasoning via SFT and/or GRPO [33]. *RoboBrain-2.0* [89] and *VeBrain* [58] extend Qwen2.5-VL [6] with rich spatial task suites, while *SpaceQwen* [1] and *SpaceThinker* [2] are Qwen2.5-VL variants fine-tuned on spatial VQA data [3] following data synthesis protocol of SpatialVLM [12]. For *MindCube* [113], we used the `Plain-CGMap-FFR-Out` SFT variant, reported as the best-performing configuration by the authors.

We include models from *VST* [110], a concurrent work that fine-tunes Qwen2.5-VL on a curated dataset spanning over 19 spatial tasks, comparing both their SFT *(VST-SFT)* and RL-tuned *(VST-RL)* variants. We further compare with a SFT variant of *SpaceR* [72] and *SpatialLadder* [51], a concurrent work employing a progressive SFT+GRPO training schedule for spatial reasoning. Finally, we evaluate *VG-LLM* [125], which integrates a 3D geometry encoder initialized from VGGT [97] into an MLLM, similar to VLM-3R [26] **in the main paper**, which integrates CUT3R [99] features to provide strong 3D priors.

Results. Full results are shown in Tab. A1. Consistent with Tab. 1 **of the main paper**, our backward token warping methods (*i.e.*, *Backward-Nearest* and *Backward-Adaptive*) achieve the best performance on both `ViewBench-Text` and `ViewBench-Shape`, outperforming all baselines including the newly added models. Notably, recent state-of-the-art general MLLMs (*e.g.*, Qwen3-VL [106], InternVL3 [129]) still struggle to internally shift viewpoint to solve our tasks. Likewise, MindCube [113], despite being designed for multi-view spatial reasoning, shows clear limitations when required to reason about a single view from a nearby target viewpoint. SpatialLadder [51], despite its carefully designed training curriculum, still underperforms our backward token warping, which explicitly and reliably transfers source-view information to the target viewpoint.

Lastly, VG-LLM [125], which integrates rich 3D features from VGGT [97], exhibits highly degraded behavior: the model frequently outputs multiple-choice labels (*e.g.*, *"A"*, *"B"*) even when prompted to answer with *"left"* or *"right"*. We hypothesize that the VGGT-based fine-tuning phase may have compromised the base MLLM's general capabilities, whereas our token warping approach leaves the underlying MLLM unchanged, better preserving its original abilities.

A.2. Robustness Analysis on Estimated Geometry

Our token warping framework relies on the depth map $\mathbf{D}$ and relative camera pose $\mathbf{\Pi}_{T\to S}$ to compute the backward warping function $f_{T\to S}$ (Eq. B.4). A natural concern is whether the method remains effective when geometric inputs are estimated rather than ground-truth. We evaluate this on `ViewBench-Shape` by replacing the ground-truth geometry with predictions from off-the-shelf models.

Depth Estimation. We compare ground-truth depth (GT) against predictions from two monocular depth estimators: Depth Anything v2 (DA-V2) [108] and Depth Pro (DP) [10]. We additionally include a no-warping reference baseline (Ref.) using the base Qwen2.5-VL [6] on the source image. As shown in Tab. A2, backward token warping with adaptive fetching achieves 65.84% with DA-V2 and 67.74% with DP, compared to 70.99% with GT depth. Pixel-wise backward warping follows the same trend, dropping from 62.35% (GT) to 60.49% (DA-V2) and 62.76% (DP). In both cases, warping with estimated geometry substantially outperforms the no-warping baseline, confirming that the gains from warping persist even without ground-truth geometry. Importantly, the performance gap between token warping and pixel-wise warping is preserved regardless of the depth source, indicating

[*] Equal contribution.

	ViewBench-Text (%)						ViewBench-Shape (%)						ViewBench-Object (1-10)					
View Overlap (%)	5–15		15–25		25–35		5–15		15–25		25–35		5–15		15–25		25–35	
Depth	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.	GT	Pred.
Target View (Oracle)	100.00	–	100.00	–	100.00	-	100.00	–	100.00	–	100.00	-	6.64	–	7.31	–	7.43	–
Specialist MLLMs																		
SpatialReasoner [61]	46.73	–	53.30	–	53.71	–	33.72	–	38.27	–	48.15	–	–	–	–	–	–	–
VLM-3R [26]	63.82	–	70.56	–	60.57	–	49.22	–	49.79	–	50.21	–	–	–	–	–	–	–
ViLaSR [102]	44.22	–	52.28	–	48.00	–	22.87	–	23.05	–	34.57	–	–	–	–	–	–	–
Qwen2.5-VL [6]	46.23	–	59.39	–	52.00	–	24.42	–	25.10	–	37.86	–	–	–	–	–	–	–
Qwen3-VL [106]	41.71	–	47.21	–	45.14	–	18.60	–	22.22	–	35.80	–	–	–	–	–	–	–
InternVL3 [129]	56.28	–	64.47	–	61.71	–	32.17	–	38.68	–	51.85	–	–	–	–	–	–	–
Cambrian-1 [93]	9.05	–	11.68	–	9.71	–	34.88	–	34.57	–	44.03	–	–	–	–	–	–	–
LLaVA-OneVision-1.5 [4]	48.24	–	51.27	–	61.71	–	27.52	–	30.04	–	38.27	–	–	–	–	–	–	–
Kimi-VL-Thinking [90]	49.25	–	54.31	–	52.00	–	31.78	–	37.86	–	43.21	–	–	–	–	–	–	–
RoboBrain-2.0 [89]	37.69	–	43.65	–	49.71	–	22.48	–	29.63	–	39.92	–	–	–	–	–	–	–
VeBrain [58]	49.25	–	54.31	–	54.29	–	29.84	–	32.51	–	47.33	–	–	–	–	–	–	–
SpaceQwen [12]	68.34	–	72.69	–	62.86	–	48.06	–	46.50	–	49.38	–	–	–	–	–	–	–
SpaceThinker [12]	48.74	–	51.27	–	48.57	–	46.51	–	47.74	–	48.15	–	–	–	–	–	–	–
MindCube [113]	59.30	–	59.39	–	57.14	–	46.90	–	47.74	–	47.33	–	–	–	–	–	–	–
VST-RL [110]	28.14	–	34.01	–	38.29	–	28.29	–	26.34	–	43.62	–	–	–	–	–	–	–
VST-SFT [110]	42.71	–	47.72	–	46.29	–	28.29	–	26.34	–	43.62	–	–	–	–	–	–	–
SpaceR-SFT-7B [72]	67.84	–	73.10	–	64.00	–	44.96	–	48.15	–	53.09	–	–	–	–	–	–	–
SpatialLadder [51]	70.35	–	74.11	–	67.43	–	50.00	–	49.38	–	50.21	–	–	–	–	–	–	–
VG-LLM [125]	5.93	–	13.20	–	9.71	–	13.18	–	14.40	–	24.69	–	–	–	–	–	–	–
Novel View Synthesis																		
GenWarp [85]	69.35	–	71.07	–	66.29	–	53.10	–	47.33	–	55.14	–	4.32	–	4.81	–	4.34	–
Pixel-Wise Warping																		
Forward	70.85	69.35	73.60	73.10	62.86	67.43	56.20	56.20	56.79	56.79	60.49	60.08	3.22	3.22	4.04	3.87	4.78	4.54
Backward	71.86	67.84	75.63	74.62	68.57	68.57	62.40	58.14	58.02	56.79	66.67	64.20	4.53	4.45	5.52	5.48	5.94	5.89
Token Warping																		
Forward	60.30	66.83	64.47	65.48	54.86	60.57	55.04	56.98	55.14	60.91	53.09	56.38	4.09	4.20	4.27	4.37	4.07	3.78
Backward-Nearest	74.87	**75.38**	**80.71**	**81.73**	74.86	**76.00**	**67.44**	63.95	62.96	**62.55**	73.25	**75.31**	4.80	4.86	5.39	5.57	**6.19**	5.97
Backward-Adaptive	**77.89**	73.37	79.70	80.71	**78.86**	74.29	**67.44**	**66.28**	**66.26**	61.32	**75.72**	70.37	**4.97**	**5.18**	**5.76**	**6.29**	6.11	**6.14**

Table A1. **Additional Quantitative Comparisons on ViewBench.** Extended table of Tab. 1 **in the main paper**, with additional baseline MLLMs included in orange ( ). Columns 2-13 report accuracy (%) on spatial reasoning tasks (ViewBench-Text and ViewBench-Shape), and columns 14-19 report target-view object description scores (ViewBench-Object), evaluated by Qwen2.5-VL-14B [6] on a 1–10 scale. Across all tasks and setups, backward token-wise warping achieves the best performance.

that the advantage of operating in token space is orthogonal to improvements in depth estimation quality.

Joint Depth and Pose Estimation. We further evaluate a more challenging setting where *both* depth and relative pose are predicted from an image pair, using VGGT [97] and DUSt3R [100]. As reported in Tab. A2, token warping with VGGT-estimated geometry achieves 68.95%, compared to 63.58% for pixel-wise warping under the same conditions. With DUSt3R, both methods decline further, yet token warping still outperforms pixel-wise warping. These results confirm that the conclusions of Tab. 1 **of the main paper** hold under realistic conditions where ground-truth geometry is unavailable.

	GT	Depth		Depth & Pose		Ref.
		DA-V2	DP	VGGT	DUSt3R	
Pixel-Wise Warp.	62.35	60.49	62.76	63.58	61.29	31.48
Token Warp.	70.99	65.84	67.74	68.95	65.05	

Table A2. **Robustness to Estimated Geometry.** Accuracy (%) on ViewBench-Shape (averaged across all overlap levels). *Ref.* is a no-warping baseline with base Qwen2.5-VL [6].

A.3. Larger Viewpoint Shifts and Occlusion

To stress-test our method beyond the overlap ranges in Sec. 5 **of the main paper** (5–35%), we construct two additional evaluation splits targeting extreme viewpoint shifts and occlusion.

Larger Viewpoint Shifts. We sample source–target pairs from ScanNet [19] with very low overlap (2–5%), representing nearly disjoint views where only a small portion of the scene is shared. As shown in Tab. A3, backward token warping with adaptive fetching achieves 65.08% with GT depth and 66.14% with estimated depth, substantially outperforming pixel-wise backward warping (61.90% / 61.38%) and the no-warping baseline (34.39%). The consistent trend across all overlap levels suggests that the advantages of token-level

warping are not confined to moderate viewpoint changes.

Depth	GT	Pred.
Qwen2.5-VL [6]	34.39	–
Pixel-Wise Warp.	61.90	61.38
Token Warp.	65.08	66.14

Table A3. **Larger Viewpoint Shift (2–5% Overlap).** Accuracy (%) on a stress-test split with extremely low view overlap, where the source and target views share only 2–5% of visible scene content.

Occlusion. We also collect synthetic image pairs using ProcTHOR [21] where an object visible from the source view becomes *fully occluded* at the target viewpoint. This tests whether warping helps the model reason about visibility changes under viewpoint shifts. As shown in Fig. A1, token warping achieves 46% accuracy with GT depth, compared to 38% for pixel-wise warping and 32% for the base Qwen2.5-VL [6], evaluated on 50 pairs with GT depth. While absolute accuracies are lower due to the difficulty of reasoning under full occlusion, the relative ordering is consistent with our main findings: token warping provides a more reliable basis for viewpoint reasoning even under significant visibility changes.

Depth	GT
Qwen2.5-VL [6]	32.00
Pixel-Wise Warp.	38.00
Token Warp.	46.00

Table A4. **Occlusion Evaluation.** Accuracy (%) on a ProcTHOR-based [21] split where the queried object is fully occluded in the target view. Token warping consistently outperforms pixel-wise warping and the base Qwen2.5-VL [6].

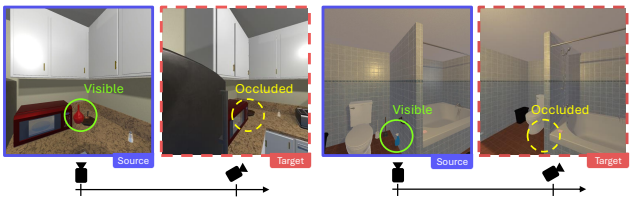

Figure A1. **Occlusion Evaluation.** Example source–target pairs from the ProcTHOR-based [21] occlusion split, where a visible object in the source view becomes fully occluded in the target view.

A.4. Geometry-Based Oracle

To verify the reliability of the geometric pipeline underlying our token warping, we implement a *geometry-based oracle* that bypasses the MLLM entirely. Given a source–target pair, the oracle applies the backward warping function $f_{T\to S}$ (Eq. B.4) to the two annotated keypoints in the source image and determines their left–right ordering by directly comparing the x-coordinates of the warped points, without querying the MLLM.

As shown in Tab. A5, the geometry-based oracle achieves above 93% across all overlap levels for both `ViewBench-Text` and `ViewBench-Shape`. The small gap from 100% is attributable to occasional depth noise near object boundaries and edge cases where the two keypoints project to nearly identical x-coordinates in the target view. These results confirm that the warping geometry is highly accurate, and that the remaining gap between our token warping methods (Tab. 1 **of the main paper**) and the oracle is primarily due to limitations in the MLLM's perception and reasoning capabilities rather than geometric errors.

	`Text (%)`			`Shape (%)`		
View Overlap (%)	5–15	15–25	25–35	5–15	15–25	25–35
Oracle (Geometry)	93.78	93.81	93.64	95.26	94.54	93.75

Table A5. **Geometry-Based Oracle.** Accuracy (%) of a geometry-only baseline that determines left–right ordering by comparing x-coordinates of the warped source keypoints.

A.5. Additional Qualitative Results

We provide additional qualitative comparisons of our backward token warping with multiple baselines—including pixel-wise warping and forward token warping—on single-view VQA examples that require reasoning under viewpoint changes. The visualizations are shown in Figs. A2– A5, with brief descriptions provided below. For each case, we are given the source image, its depth map, the relative camera pose from source to target, and the camera intrinsics. To obtain the depth and poses, we run VGGT [97] on the source and target view images.

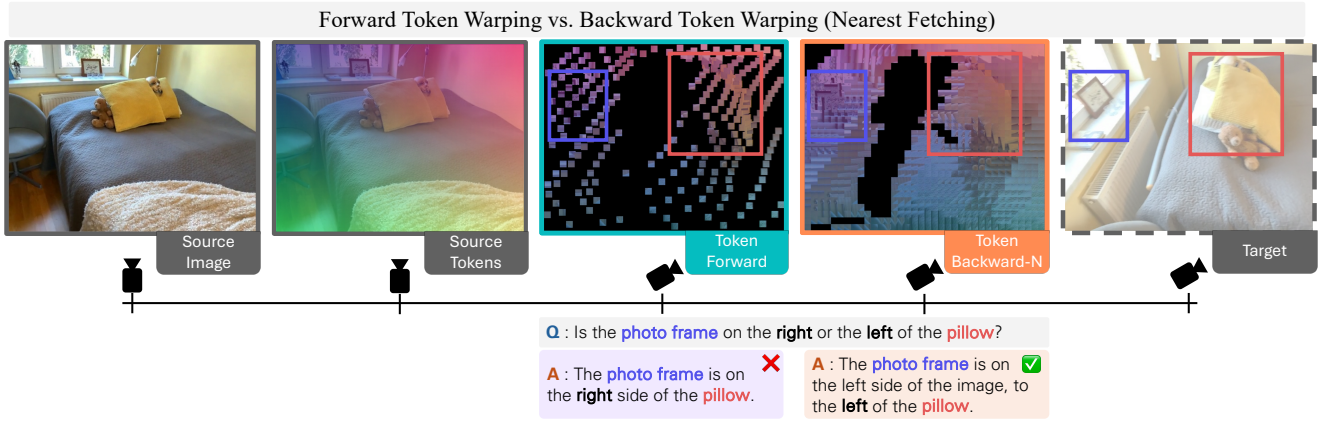

Figure A2. **Qualitative Sample 1.** Given the source image (leftmost), the question asks for the spatial relationship between the photo frame (blue box) and the pillow (red box) as viewed *from the target viewpoint* (rightmost). **To visualize tokens, we color-code each source token by its** (x, y) **position in the source image, and preserve this color after warping, so the color of each token in the warped views indicates its source location.** With **forward token warping**, the projected tokens become sparse and irregular, leading the MLLM to answer incorrectly. In contrast, **backward token warping** with **nearest fetching** produces a dense, regular target token grid, allowing the model to correctly infer the spatial relationship from the target view. (Source and target images are from ARKitScenes [8].)

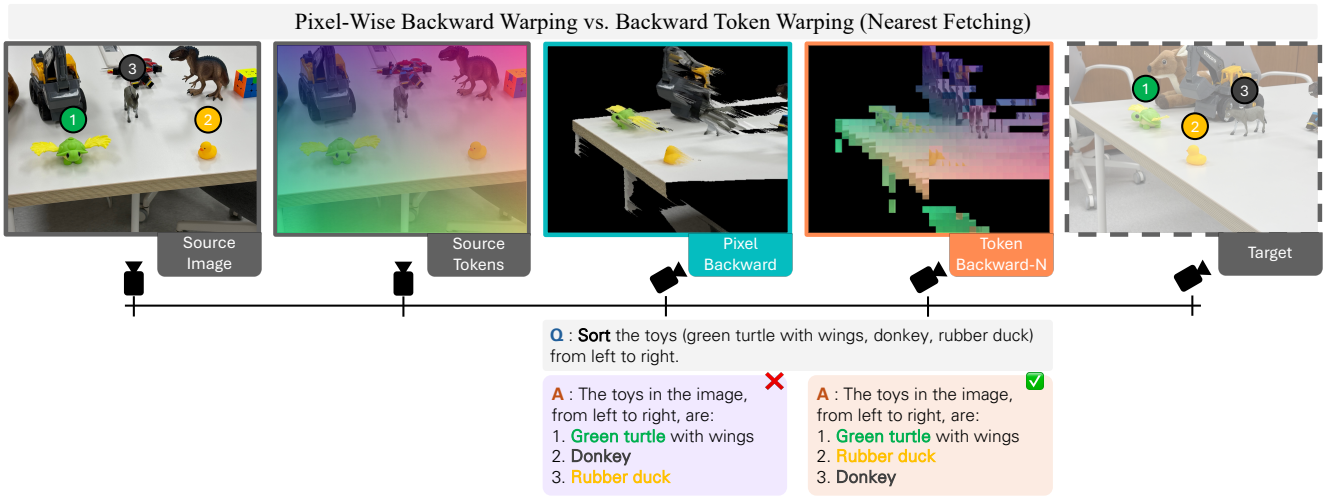

Figure A3. **Qualitative Sample 2.** Given the source image (leftmost), the question asks for the order of the toys from left to right *as seen from the target viewpoint* (rightmost). **To visualize tokens, we color-code each source token by its** (x, y) **position in the source image, and preserve this color after warping, so the color of each token in the warped views indicates its source location.** With **pixel-wise backward warping**, the target-view image suffers from local pixel distortions caused by depth noise, leading the MLLM to answer incorrectly. In contrast, **backward token warping** with **nearest fetching** preserves the semantic content while shifting viewpoint, allowing the MLLM to produce the correct ordering of the toys. (Source and target images were captured manually.)

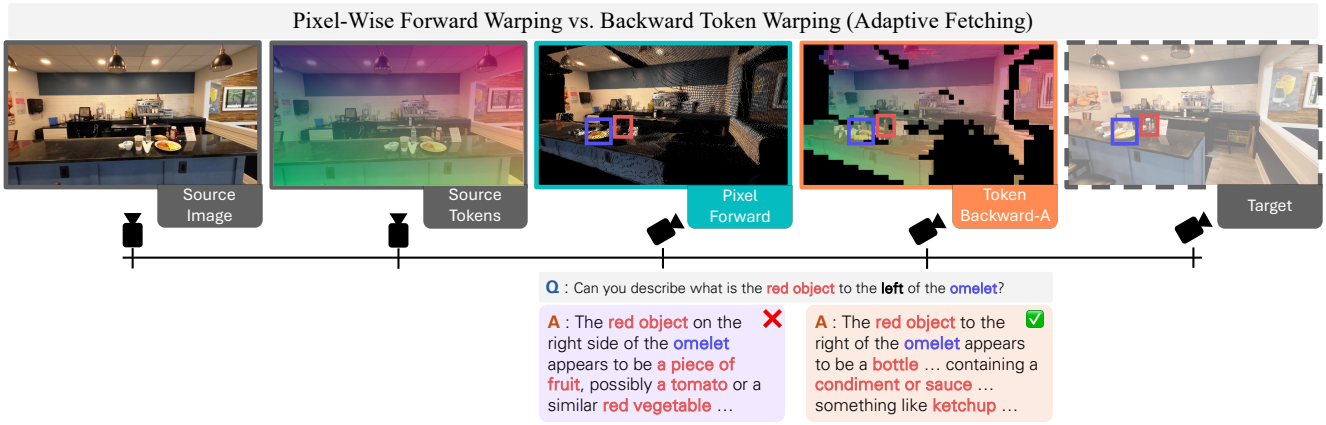

Figure A4. **Qualitative Sample 3.** Given the source image (leftmost), the question asks to describe the *red object* (red box) placed on the left side of the omelet (blue box) *when viewed from the target viewpoint* (rightmost). **To visualize tokens, we color-code each source token by its** (x, y) **position in the source image, and preserve this color after warping, so the color of each token in the warped views indicates its source location.** When using **pixel-wise forward warping**, the warped image exhibits local pixel distortions due to depth prediction noise and holes caused by magnification. Consequently, given this warped RGB image, the MLLM incorrectly answers that the object is *"a piece of fruit"*. In contrast, with **backward token warping** and **adaptive fetching**, the MLLM correctly identifies the object as a *"bottle"*, more specifically *"containing a condiment or sauce"* and *"ketchup"*. This further highlights the advantage of warping in token space rather than pixel space when transferring source content to a target view. (Source and target images are from DL3DV-10K [54].)

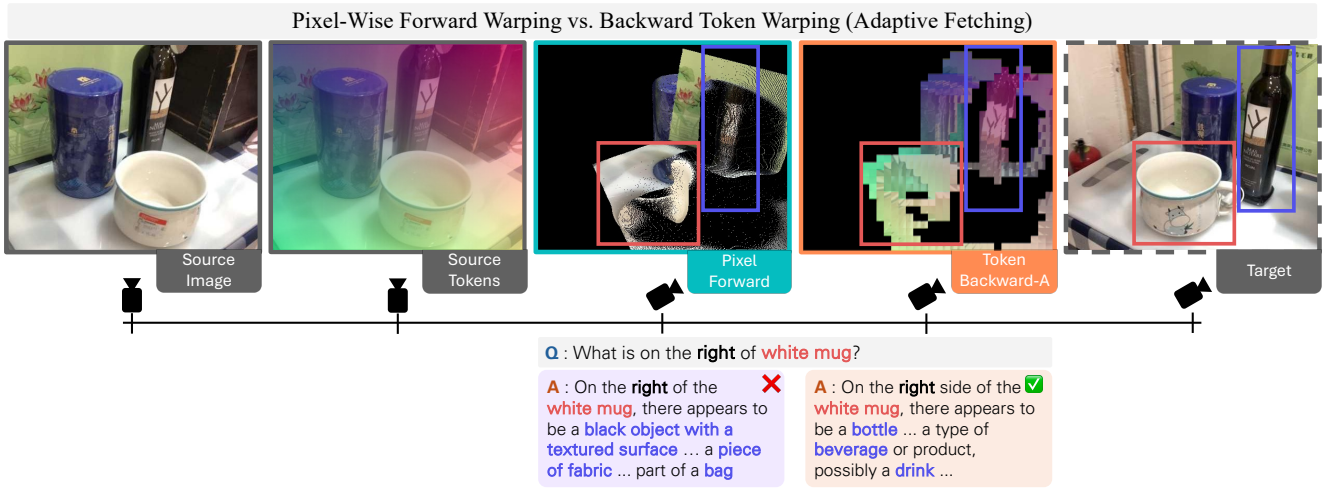

Figure A5. **Qualitative Sample 4.** Given the source image (leftmost), the question asks to describe the object that is located on the right side of the white mug (red box) *when viewed from the target viewpoint* (rightmost). **To visualize tokens, we color-code each source token by its** (x, y) **position in the source image, and preserve this color after warping, so the color of each token in the warped views indicates its source location.** With **pixel-wise forward warping**, the warped image shows distorted local details due as the forward warping distributes the source image pixels to a sparse grid in the target image. Consequently, the MLLM fails to accurately describe the bottle on the right side, and instead replies *"piece of fabric"* and *"part of a bag"*, which are not visible in the target image. On the other hand, when using **backward token warping** with **adaptive fetching**, the MLLM describes the specified object as *"a bottle"* and *"a type of beverage"* which is accurate when seen from the target image. These results again show that our proposes backward token warping can provide a robust way of transferring source image information to the target viewpoint. (Source and target images are from BLINK [30].)

B. Implementation Details

This section extends Sec. 3.3 **of the main paper** and details the implementation of our backward token warping framework, which enables MLLMs to reason under viewpoints changes from a single source image, its depth map, and relative camera pose. For clarity, in this section we use "**c**" to denote coordinates in the *source* view and "**g**" to denote coordinates in the *target* view.

B.1. Details on Backward Token Warping

Recall that in backward warping, we define a dense, regular grid in the target view and fetch the corresponding tokens from the source image $\mathbf{I}$ via the target-to-source mapping $f_{T\to S}$.

Target Grid. For an image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, we impose a regular patch grid of size $l \times l$, yielding $M = (HW)/l^2$ patches*. We denote by $\mathbf{g} \in \mathbb{R}^{M\times 2}$ the set of target-grid centers on the image plane, where each $\mathbf{g}_j$ specifies a location at which we wish to place a token sampled from the source image. In backward token warping, our goal is to assign exactly one token to each grid center. For simplicity, we assume the target image has the same resolution as the source.

Source Proxy from Depth. Because the target-view image is unobserved, we cannot directly compute target-to-source correspondences. Instead, we construct a lightweight 3D triangle mesh $\mathcal{M}_S$ from the source depth map $\mathbf{D} \in \mathbb{R}^{H\times W\times 1}$. Specifically, for each pixel $\mathbf{p}_i = (u_i, v_i)$ in $\mathbf{I}$ with its depth d_i from $\mathbf{D}$, we unproject it using the 3×3 intrinsic matrix $\mathbf{K}_{3\times 3}$ to obtain a 3D point:

$$\mathbf{x}_i = d_i \mathbf{K}_{3\times 3}^{-1} \tilde{\mathbf{p}}_i, \quad \text{where } \tilde{\mathbf{p}}_i = [u_i, v_i, 1]^\top. \tag{B.1}$$

Here, $\mathbf{x}_i = [x_i, y_i, z_i]^\top$. We then triangulate every 2×2 pixel cell into two triangles, forming $\mathcal{M}_S$ in the source camera frame.

Backward Mapping via Ray Casting. For each target grid center $\mathbf{g}_j$, we cast a ray from the target camera using its pose $\Pi_T \in \mathbb{R}^{4\times 4}$ and intrinsics $\mathbf{K} \in \mathbb{R}^{4\times 4}$, and intersect it with the proxy mesh $\mathcal{M}_S$, obtaining a 3D hit point in the target frame, $\mathbf{x}_j^* \in \mathbb{R}^3$. We then express this point in homogeneous coordinates and project it back into the source image using the relative pose $\Pi_{T\to S} = \Pi_S \Pi_T^{-1}$ and intrinsics $\mathbf{K}$:

$$\tilde{\mathbf{p}}_j^* = \mathbf{K}\, \Pi_{T\to S}\, \tilde{\mathbf{x}}_j^*, \quad \text{where } \tilde{\mathbf{x}}_j^* = [\mathbf{x}_j^*, 1]^\top, \tag{B.2}$$

$$\mathbf{g}_j^* = \pi\left(\tilde{\mathbf{p}}_j^*\right), \tag{B.3}$$

where $\pi([u, v, w, 1]^\top) = (u/w, v/w)^\top$ denotes perspective projection. The resulting $\mathbf{g}_j^* \in \mathbb{R}^2$ is a coordinate on $\mathbf{I}$ and serves as the backward mapping from target to source. If no valid intersection is found (e.g., due to occlusion or field-of-view mismatch), we mark $\mathbf{g}_j^*$ as invalid and omit the corresponding patch.

By applying Eq. B.3 for every target grid center $\mathbf{g}_j \in \mathbf{g}$, we obtain the set of backward-warped coordinates on the source image, $\mathbf{g}^* \in \mathbb{R}^{M\times 2}$. Consistent with Eq. 3.1 **in the main paper**, we denote this backward warping process as

$$\mathbf{g}^* = f_{T\to S}(\mathbf{g}, \Pi_{T\to S}, \mathbf{K}, \mathbf{D})\,. \tag{B.4}$$

Given $f_{T\to S}$, which provides a coordinate for every target grid center, the final step is to *fetch* the corresponding tokens from the source image at these locations. We provide details on the fetching strategies in the next section.

B.2. Nearest vs. Adaptive Fetching

We now detail the *nearest* and *adaptive* token fetching strategies used in the final step of backward token warping.

Nearest Fetching. Recall from Sec. 3.1 **in the main paper** that source image $\boldsymbol{I}$ is partitioned into a fixed, non-overlapping grid of patches $\{\mathbf{u}_i\}_{i=1}^M$. Let the source image $\mathbf{I}$ be patchified on a fixed grid, and let $\mathbf{c} \in \mathbb{R}^{M\times 2}$ denote the set of source grid centers, where M is the number of patches. Given a target grid center $\mathbf{g}_j$ and its backward-warped source coordinate $\mathbf{g}_j^*$ from $f_{T\to S}$, *nearest fetching* selects the existing source patch whose center is closest to $\mathbf{g}_j^*$ in Euclidean distance:

$$i' = \arg\min_i \left\| \mathbf{g}_j^* - \mathbf{c}_i \right\|_2. \tag{B.5}$$

We then assign to $\mathbf{g}_j$ the token that was derived from the patch $\mathbf{u}_{i'}$ centered at $\mathbf{c}_{i'}$. While this introduces a small mismatch as $\mathbf{g}_j^*$ may not not coincide with any $\mathbf{c}_i$ in most cases, it allows us to reuse the original, efficient fixed-grid patchification for the source image.

Adaptive Fetching. Alternatively, we further implement *adaptive fetching*, which re-patchifies the source image $\mathbf{I}$ according to the backward-warped coordinates $\mathbf{g}^*$ so that each patch is centered exactly at $\mathbf{g}_j^*$ with size $l \times l$. For each $\mathbf{g}_j^*$, we obtain a patch $\bar{\mathbf{u}}_j$ via

$$\bar{\mathbf{u}}_j = \mathrm{Crop}(\mathbf{I}, \mathbf{g}_j^*), \quad \bar{\mathbf{u}}_j \in \mathbb{R}^{l\times l\times 3}, \tag{B.6}$$

where $\mathrm{Crop}(\mathbf{I}, \mathbf{g}_j^*)$ extracts an $l \times l$ patch from $\mathbf{I}$ centered at $\mathbf{g}_j^*$. Applying this to all $\mathbf{g}_j^* \in \mathbf{g}^*$ yields a new set of *adaptive* patches $\{\bar{\mathbf{u}}_j\}_{j=1}^M$ that replaces the original fixed-grid patches $\{\mathbf{u}_i\}_{i=1}^M$. Finally, we assign to each target grid center $\mathbf{g}_j$ the token derived from its corresponding adaptive patch $\bar{\mathbf{u}}_j$, which is explicitly centered at $\mathbf{g}_j^*$. Intuitively, this approach more faithfully respects the precise backward mappings in $f_{T\to S}$, at the cost of re-patchifying the image rather than relying on the original, efficient fixed-grid partitioning.

*We assume H and W are divisible by l.

C. Details on ViewBench

In this section, we provide additional details on the data synthesis protocol and evaluation metrics for `ViewBench`, introduced in Sec. 4 **in the main paper**.

C.1. Benchmark Construction

We construct `ViewBench` from real indoor scenes in ScanNet [19], which provides dense RGB-D frames along with ground-truth depth, camera poses, and intrinsics. For evaluations with estimated depth maps, we use Depth Anything v2 [108]. To sample two-view pairs with controlled overlap, we use the MultiSPA data engine from Xu et al. [104], originally introduced for generating multi-view VQA data. We adopt the same notions of *visible points* and *overlap ratio* as in MultiSPA and use them to construct `ViewBench` questions. Below, we detail the benchmark construction procedure, following the notation of MultiSpa [104].

Overlap Computation. For each ScanNet scene [19], we are given a 3D point cloud

$$\mathbf{P}_{\text{scene}} = \{\mathbf{p}^w\}, \quad \text{where } \mathbf{p}^w = [x^w, y^w, z^w]^\top, \tag{C.1}$$

with each point $\mathbf{p}^w$ expressed in the world coordinate system. Each RGB frame $\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}$ is associated with a depth map $\mathbf{D}_i \in \mathbb{R}^{H\times W\times 1}$, an extrinsic matrix $\mathbf{E}_i \in \mathbb{R}^{4\times 4}$, and an intrinsic matrix $\mathbf{K}_i \in \mathbb{R}^{4\times 4}$. The extrinsic matrix is defined as

$$\mathbf{E}_i := \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad \mathbf{R}_i \in \mathbb{R}^{3\times 3},\ \mathbf{t}_i \in \mathbb{R}^{3\times 1}, \tag{C.2}$$

where $\mathbf{R}_i$ and $\mathbf{t}_i$ denote the camera rotation and translation, respectively.

Following MultiSPA [104], we map each world point $\mathbf{p}^w$ into the i-th camera coordinate system via

$$\tilde{\mathbf{p}}_i^c = (\mathbf{E}_i)^{-1}\tilde{\mathbf{p}}^w, \quad \text{where } \tilde{\mathbf{p}}^w = [\mathbf{p}^w, 1]^\top, \tag{C.3}$$

and denote $\tilde{\mathbf{p}}_i^c = [x_i^c, y_i^c, z_i^c, 1]^\top$. We then project this point to the image plane :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{\mathbf{K}_i}{z_i^c} \begin{bmatrix} x_i^c \\ y_i^c \\ z_i^c \end{bmatrix}, \tag{C.4}$$

We define the set of *visible points* in frame i as:

$$\mathcal{V}_i = \left\{ \mathbf{p}^w \in \mathbf{P}_{\text{scene}} \,\middle|\, 0 < z_i^c < d_i(u,v) \right\}, \tag{C.5}$$

where $d_i(u,v)$ is the depth value at pixel (u,v) from $\mathbf{D}_i$. This captures points whose projections fall inside $\mathbf{I}_i$ and are not occluded according to $\mathbf{D}_i$, which is identical to the visibility criterion of MultiSPA [104].

Finally, given two frames $\mathbf{I}_i$ and $\mathbf{I}_j$, we measure how much of the scene they see in common using the IoU of their visible point sets, defining the *overlap ratio* [104]:

$$\text{Overlap}(i,j) = \frac{\left|\mathcal{V}_i \cap \mathcal{V}_j\right|}{\left|\mathcal{V}_i \cup \mathcal{V}_j\right|}. \tag{C.6}$$

We use this overlap ratio to create controlled splits in `ViewBench`.

Two-View Pair Selection. For each ScanNet scene, we enumerate candidate frame pairs and compute the overlap ratio defined above. We retain two-view a pair $(\mathbf{I}_S, \mathbf{I}_T)$ as a candidate if $\text{Overlap}(S,T)$ lies in a moderate range (approximately 5–35%), so that the two views are neither nearly identical nor almost disjoint. Following the overlap-aware sampling strategy of MultiSPA [104], we bin all non-zero-overlap pairs by their overlap ratio and sample an approximately equal number of pairs from each bin to mitigate the natural long-tailed bias toward small overlaps. We then group the selected pairs into three overlap levels: **5–15%**, **15–25%**, and **25–35%**. This categorization allows us to systematically study how viewpoint-conditioned reasoning changes as the amount of shared scene content varies.

Point Annotation. For each selected source–target pair $(\mathbf{I}_S, \mathbf{I}_T)$, we focus on the points that are visible in *both* views, that is, the co-visible set $\mathcal{V}_S \cap \mathcal{V}_T$. For any $\mathbf{p}^w$ in this intersection, we obtain its camera-frame coordinates in each view via

$$\tilde{\mathbf{p}}_S^c = (\mathbf{E}_S)^{-1}\tilde{\mathbf{p}}^w, \quad \tilde{\mathbf{p}}_T^c = (\mathbf{E}_T)^{-1}\tilde{\mathbf{p}}^w, \tag{C.7}$$

and then project them to the image planes using the same camera model as in Eq. C.4. These co-visible projections form the pool of candidate keypoints used to construct task-specific questions, analogous to the *visual correspondence* subset construction in MultiSPA [104].

For `ViewBench-Text`, we randomly sample two co-visible points and annotate them with alphabet labels (*i.e.*, A/B). For `ViewBench-Shape`, we instead mark them with simple geometric symbols (e.g., triangle, star). In all cases, annotations in the two views are guaranteed to correspond to the same underlying 3D locations.

Selecting View-Dependent Point Pairs. Given a source–target pair $(\mathbf{I}_S, \mathbf{I}_T)$ and its co-visible point set $\mathcal{V}_S \cap \mathcal{V}_T$, we construct left–right queries by sampling two co-visible 3D points and projecting them into both images (using the same intrinsics, extrinsics, and visibility checks as above). Let u_A^S, u_B^S and u_A^T, u_B^T denote the u-coordinates of the two keypoints (A and B) in the source and target views, respectively. We retain a pair only if

$$(u_A^S - u_B^S)\,(u_A^T - u_B^T) < 0 \quad \text{and} \quad |u_A^T - u_B^T| \geq \tau, \tag{C.8}$$

with $\tau = 50$ pixels to avoid near-vertical alignments. Thus, we keep only examples where the left–right relation flips between views and is sufficiently separated in the target, ensuring that the correct answer genuinely depends on adopting the target viewpoint.

Instruction Generation. For each source-target pair, we convert the annotations into instruction–answer examples to be input to MLLMs. We render task-specific visual markers: alphabet labels for `ViewBench-Text`, geometric symbols for `ViewBench-Shape`, and a single red circular marker for `ViewBench-Object`.

For `ViewBench-Text` and `ViewBench-Shape`, we pose a binary left–right question about the two markers in the *target* view, randomly ordering the options (*e.g.*, *"left, right"* vs. *"right, left"*). The ground-truth label is computed deterministically from the x-coordinates of the two keypoints in the target image. For `ViewBench-Object`, we instead use a fixed open-ended template (*e.g.*, *"Can you describe the object or feature at the red point?"*) and treat the MLLM's response on the oracle target image as the reference description.

After applying the full data-processing pipeline, we obtain:

- **571** text questions (`ViewBench-Text`),
- **744** shape questions (`ViewBench-Shape`),
- **300** object-description samples (`ViewBench-Object`),

all validated using the target-view oracle and co-visibility constraints.

C.2. Details on ViewBench-Object Evaluation

As noted in Sec. 4 **of the main paper**, to evaluate MLLM responses on the target-view object description task (`ViewBench-Object`), we use an LLM (Qwen2.5-14B-Instruct [6]) as an evaluator, asking it to rate each response on a 1–10 scale. For this, we query the evaluator LLM with the following prompt template:

```
You are an AI assistant who will help
me to evaluate the response given the
question and the correct answer.
To mark a response, you should output
a single integer between 1 and 10
(including 1, 10).

- 10 means that the response is
describing the same or similar scene
as the answer.

- 1 means that the response is describing
a completely different scene from the
answer.

Question: {question}
Answer: {answer}
Response: {response}

Please output in format
<score>...</score>.
```