Zero-Shot Depth Completion with Vision Language Model

Zhiqiang Yan[1] Yuan Wu[2] Gim Hee Lee[1]
[1]Department of Computer Science, National University of Singapore
[2]Nanjing University of Science and Technology {yanzq, gimhee.lee}@nus.edu.sg

Abstract

Vision language models (VLMs) have achieved remarkable success in semantic understanding tasks under language guidance, yet their potential for geometric perception remains largely underexplored. This paper introduces the first VLM-based depth completion framework. With almost no architectural modifications, we propose a sparse depth injection mechanism that extends the capability of VLM toward 3D perception through three key aspects: visual tokenization, textual prompt, and textual supervision. At the visual input side, sparse depth is tokenized to provide absolute scale and accurate geometric cues, alleviating the scale and camera ambiguities of RGB-only inputs. At the textual input side, a binary mask derived from sparse depth serves as a prompt, instructing the model where to complete and where to preserve. At the supervision side, the model is fine-tuned using text labels generated from sparse depth, requiring no ground-truth depth. Benefiting from the strong semantic priors and cross-modal expressiveness of VLM, our framework achieves superior zero-shot performance across diverse sensors, sparsity levels, and scenes.

1. Introduction

Depth completion [45] aims to recover dense depth from sparse depth, where color images are commonly used for auxiliary guidance. This task plays a crucial role in various computer vision applications, including autonomous driving [34, 50, 56, 62–64, 74], 3D reconstruction [35, 39, 44, 52, 54], and embodied AI [26, 29, 31, 53, 57].

Before 2024, numerous depth completion methods [34, 35, 43, 46, 51, 59] significantly advance the field through task-specific module designs and dataset-dependent training strategies. Since 2024, the research focus gradually shifts toward generalized zero-shot depth completion, with approaches such as OGNI-DC [75], G2-MonoDepth [48], and OMNI-DC [76], aiming to achieve robust generalization across diverse domains. With the rise of foundation models [20, 38, 67], Park et al. [36, 37] and Wang et al. [49] take the first step toward leveraging these models for stronger generalization. Following this trend, Marigold-DC [47] establishes a new state of the art by exploiting diffusion-based generative foundation models [22, 41].

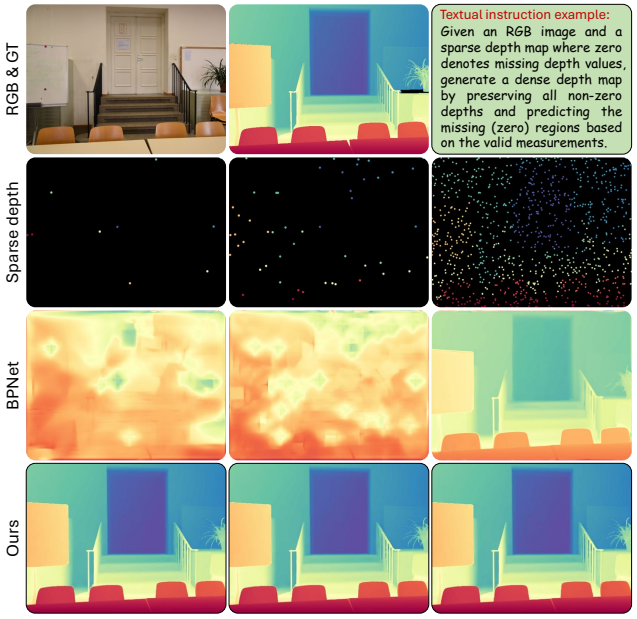

Figure 1. **Visual comparisons of zero-shot depth completion under different sparsity levels.** Our VLM-driven approach leverages textual instructions derived from sparse depth inputs to guide the completion process. Compared with previous state-of-the-art methods such as BPNet [44], our model consistently produces plausible and geometrically reliable depth predictions.

However, existing methods still do not truly capture the essence of **completion**. Given sparse depth inputs, they do not know *where to predict* and *where to preserve*, but simply embed the inputs into the network. In contrast, humans intuitively understand that depth completion requires maintaining valid depth measurements while estimating values only in unobserved regions. Inspired by this perspective, we turn to vision language models (VLMs), which excel at semantic reasoning and instruction following. We thus introduce the first VLM-based depth completion framework, aiming to endow VLM with the ability to infer dense metric depth through visual and textual prompts.

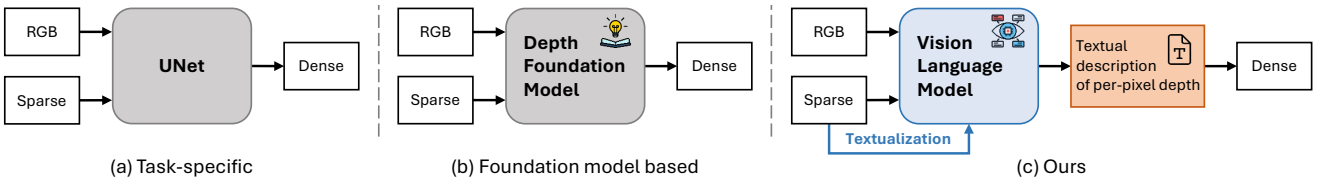

Figure 2. **Framework comparisons.** (a) Most previous depth completion studies emphasize task-specific architectural designs, while (b) recent works concentrate on zero-shot inference with the aid of depth foundation models. Differently, (c) we introduce text instructions based on VLM, offering a new perspective on depth completion.

We propose a simple yet effective **sparse depth injection mechanism** to appropriately embed depth priors into VLM. The mechanism comprises three key components: visual tokenization, textual prompt, and textual supervision. *At the visual input stage*, sparse depth measurements are tokenized in the same manner as RGB images[1]. In contrast, we apply a zero-initialized convolutional block to fuse the RGB-D inputs before embedding, enabling a gradual injection of depth priors into the VLM's vision encoder without modifying its architecture. As monocular depth estimation is inherently ill-posed due to the lack of geometric cues in a single RGB image, incorporating depth priors offers absolute scale information that resolves the scale and camera ambiguities inherent in RGB-only input, while preserving the original input structure of the VLM without any special pre-processing. *At the textual input stage*, we generate binary masks from sparse depth maps and convert them into textual prompts, where 0 denotes depth-missing regions and 1 marks areas with valid depth measurements. This design explicitly informs the model where to predict and where to preserve, enabling controllable depth completion under linguistic guidance. *At the supervision stage*, we textualize the 2D sparse depth maps into natural language descriptions, e.g., "The distance from this pixel to the camera is 5.20 meters." to enable the VLM to learn real-world depth reasoning. The textual predictions for all pixels are then transformed into a height×width depth map as the final dense output. This supervision paradigm eliminates the need for dense ground-truth depth annotations.

As illustrated in Fig. 2, previous depth completion methods mainly emphasize task-specific designs [35, 44, 46, 55, 63] or zero-shot evaluation relying on either large-scale training data [48, 76] or depth foundation models [36, 37, 47, 49]. In contrast, our approach leverages VLM to integrate both visual information and textual cues, enabling the model to discern where to predict and where to preserve. We believe that this straightforward attempt can offer new insights into the task of depth completion.

Owing to these designs, our method achieves outstanding zero-shot performance across diverse datasets, sensors, and sparsity levels.

[1]Qwen2.5-VL [1] employs a 3D convolutional layer to construct visual embeddings.

In summary, our contributions are as follows:

- To the best of our knowledge, we present the first VLM-based depth completion framework that explicitly and effectively leverages text interactions to guide this process.
- We propose a sparse depth injection mechanism, which seamlessly integrates sparse measurements into VLM through visual tokenization, textual prompt, and textual supervision, enabling 3D geometric perception.
- Extensive experiments demonstrate the superiority of our approach, achieving up to 17.3% improvement over the best methods across seven zero-shot benchmarks.

2. Related Work

2.1. Monocular Depth Completion

Early depth completion methods rely solely on sparse depth inputs, employing either classical interpolation techniques or convolutional neural networks (CNNs). For example, IP-Basic [24] performs dilation and morphological closing operations to fill sparse depth maps without any learning process. Uhrig et al. [45] propose sparsity-invariant CNNs, while subsequent works [15, 25, 32, 33, 46] adopt encoder–decoder architectures to progressively densify sparse measurements. However, these purely depth-based methods often produce overly smooth predictions that fail to preserve sharp geometric boundaries or fine-grained structures.

To address these limitations, later approaches [19, 34, 43, 44, 62, 63, 74] incorporate RGB guidance, leveraging complementary texture and edge cues for more accurate depth recovery in regions with missing measurements. FCFRNet [28] designs an energy-based fusion to integrate the RGB-D features. RigNet [62] and RigNet++ [65] iteratively refine depth by recurrently injecting image features. GFormer [39] and CFormer [74] combine convolution and transformer blocks to capture both local and global dependencies. BPNet [44] mitigates the issue of directly convolving on sparse depth by introducing an early-stage bilateral propagation mechanism, where depth values are non-linearly diffused from neighboring measurements with coefficients conditioned on spatial proximity and radiometric similarity. In a similar spirit, SigNet [66] first densifies sparse depth maps using non-CNN densification tools to obtain a coarse yet complete depth prior, and then refines it

through self-supervised degradation learning.

A large body of work [9, 19, 27, 35, 63] further refines the predicted depth through spatial propagation networks (SPNs). CSPN [8] performs recursive convolutions with fixed local kernels, and CSPN++ [10] extends this with learnable and adaptive kernel sizes. PENet [19] enlarges receptive fields via dilated convolutions, while NLSPN [35] introduces non-local propagation using deformable kernels. DySPN [27] dynamically selects nonlinear neighbors through attention, and GraphCSPN [30] constrains propagation with geometric priors in 3D space. LRRU [55] introduces a dynamic large-to-small kernel mechanism to capture dependencies ranging from long to short scales.

Recently, increasing attention has been devoted to enhancing the generalization capability of depth completion models. Robust zero-shot depth completion has emerged as a promising direction in this regard. Zuo et al. [75, 76] introduce optimization-guided neural iteration and a multi-resolution depth integrator to handle unseen domains and depth inputs with varying sparsity levels. Similarly, G2-MonoDepth [48] proposes a unified and generalizable framework for depth inference from monocular RGB combined with auxiliary depth inputs. With the advent of depth foundation models [20, 67], which exhibit remarkable generalization ability, researchers have begun incorporating them into depth completion frameworks [26, 36, 37]. For example, PromptDA [26] employs low-cost LiDAR as a prompt signal to guide the Depth Anything model [67] for precise metric depth estimation, reaching resolutions of up to 4K. PacGDC [49] leverages foundation models and a label-efficient pseudo-geometry synthesis strategy based on scale manipulation to enrich training diversity and enhance generalization without heavy annotation costs. Most recently, Marigold-DC [47] establishes a new zero-shot paradigm for depth completion based on a generative diffusion-driven depth foundation model [22]. Different from these approaches, we convert sparse depth into textual prompts, enabling the VLM to bridge high-level semantic reasoning with fine-grained 3D perception.

2.2. VLM-based 3D Perception

Recently, numerous works [7, 13, 21, 68–70, 72, 73] incorporate language descriptions to facilitate monocular depth estimation. For example, Wordepth [69] leverages variational language priors derived from textual descriptions for regularization. SpatialVLM [6] endows VLMs with explicit 3D spatial reasoning by constructing a large-scale dataset of metric-aware visual–language pairs, enabling them to infer geometric relations such as distance, size, and relative position between objects. This work lays an important foundation for geometry-aware tasks including depth estimation and robotic scene understanding. Complementarily, SpatialBot [4] integrates explicit depth inputs and spatially structured question–answer supervision, allowing VLMs to reason precisely about 3D layouts and object configurations for embodied perception and robotic interaction. Further extending spatial reasoning to dynamic scenes, MultiSpatialMLLM [61] and Seed1.5-VL [17] incorporate multi-frame inputs, depth cues, and large-scale multimodal training to achieve consistent 3D perception across time and viewpoints. Recently, DepthLM [5] reformulates per-pixel metric depth estimation as a language modeling problem, demonstrating that VLMs can achieve accurate and scalable 3D understanding even with text-based sparse supervision. Based on these excellent studies, our work further investigates how VLMs can be adapted for fine-grained 3D depth perception, aiming to bridge high-level semantic reasoning with precise geometric understanding through sparse depth priors in real-world robotic and autonomous scenarios.

3. Our Method

3.1. Overview

This paper extends VLM to the depth completion task by injecting sparse depth priors into the model, thereby enabling it to jointly reason over visual, geometric, and textual cues. An overview of our pipeline is illustrated in Fig. 3. The proposed sparse depth injection mechanism (SDIM) consists of ① visual tokenization (orange part), ② textual prompt (green part), and ③ textual supervision (blue part). Specifically, in ①, the sparse depth is first adaptively fused with the corresponding color image. Then, in ②, the sparse depth is converted into textual representations that guide the model on where to predict and where to preserve. Finally, in ③, we reuse the textual descriptions derived from the sparse depth to provide supervision for precise depth prediction. We elaborate on the three components in Sec. 3.2, Sec. 3.3, and Sec. 3.4, respectively.

3.2. Visual Tokenization

Given a color image $\mathbf{I} \in \mathbb{R}^{3\times h\times w}$ and its corresponding sparse depth map $\mathbf{S} \in \mathbb{R}^{1\times h\times w}$, we first apply a 32-channel convolution with zero initialization to encode the sparse depth, followed by batch normalization and a LeakyReLU activation, where h and w denote the height and width of the input, respectively. The initial features of the color image are extracted in a similar manner. This step can be formulated as follows:

$$\hat{\mathbf{F}} = \mathcal{F}_{\tau_1}(\mathbf{I}), \tag{1a}$$

$$\hat{\mathbf{S}} = \mathcal{F}^{z}_{\tau_2}(\mathbf{S}), \tag{1b}$$

where $\mathcal{F}_{\tau_1}(\cdot)$ denotes the composite operation of convolution, batch normalization, and LeakyReLU activation. $\mathcal{F}^{z}_{\tau_2}(\cdot)$ indicates that the convolution layer is initialized with zeros. Subsequently, two identical composite operations,

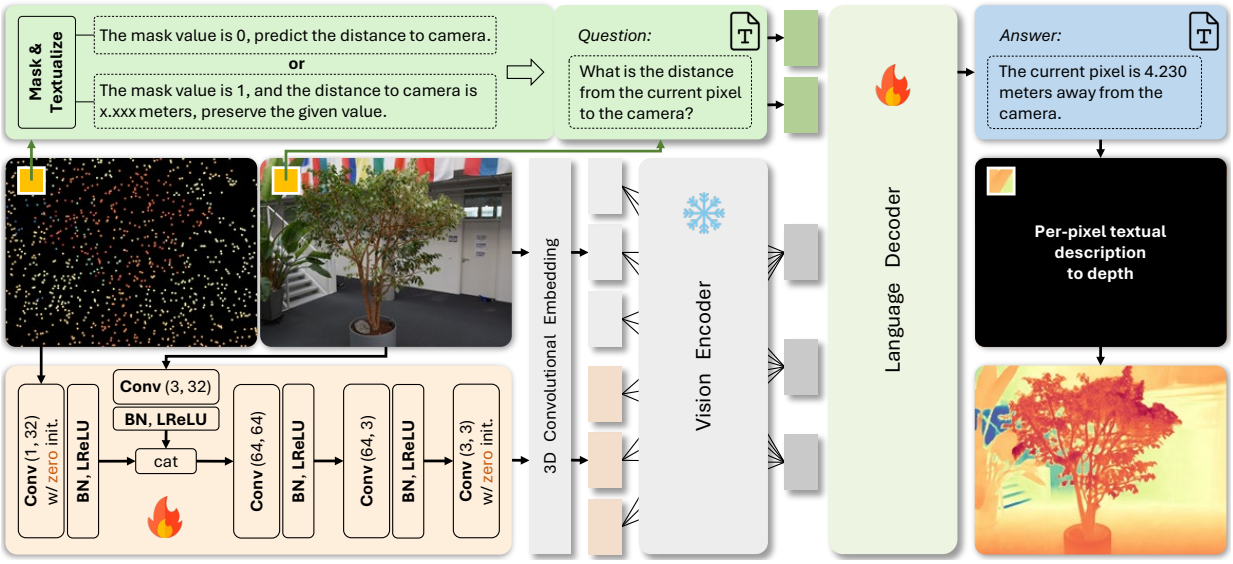

Figure 3. **Overview of our method.** It extends VLM [1] to the depth completion task via a *sparse depth injection mechanism* (SDIM). SDIM first performs a soft fusion between the sparse depth and the RGB image using convolutions with zero initialization. Then, the sparse depth is transformed into textual representations that prompt the model on where to predict and what to preserve. Meanwhile, for one image pixel, we query its distance to the camera. Finally, these textual representations derived from the sparse depth are employed to fine-tune the model, producing per-pixel depth descriptions and the final dense depth prediction.

i.e., $\mathcal{F}_{\tau_2}(\cdot)$ and $\mathcal{F}_{\tau_3}(\cdot)$, are applied to the concatenated RGB-D features to project them into high-dimensional and low-dimensional representations, respectively. Finally, a single convolutional layer with zero initialization $\mathcal{F}_c^z(\cdot)$ is employed before the embedding step:

$$\mathbf{M} = \mathcal{F}_{\tau_3}(\mathcal{F}_{\tau_2}(\mathcal{F}_{\psi}(\hat{\mathbf{F}}, \hat{\mathbf{S}}))), \tag{2a}$$

$$\mathbf{O} = \mathcal{F}_c^z(\mathbf{M}), \tag{2b}$$

where $\mathcal{F}_{\psi}(\cdot)$ indicates the concatenation operation.

After obtaining the output $\mathbf{O} \in \mathbb{R}^{3\times H\times W}$, we follow the baseline VLM [1] by applying a 3D convolution to tokenize $\mathbf{O}$ and fuse the resulting tokens with those of the color image $\mathbf{I}$, producing the joint token sequence $\mathbf{E}$. Next, we feed the sequence into the vision encoder $\mathcal{F}_{\theta}(\cdot)$ of the VLM, yielding the final visual representations $\mathbf{V}$ before the language decoder:

$$\mathbf{E} = \mathcal{F}_{e_1}(\mathbf{I}) + \mathcal{F}_{e_2}(\mathbf{O}), \tag{3a}$$

$$\mathbf{V} = \mathcal{F}_{\theta}(\mathbf{E}), \tag{3b}$$

where $\mathcal{F}_{e_1}(\cdot)$ and $\mathcal{F}_{e_2}(\cdot)$ represent the 3D convolutional embedding functions.

Before the embedding, we argue that directly concatenating the RGB-D inputs corresponds to a hard fusion, whereas our approach performs a soft fusion through a gradual, zero-initialized strategy. This soft fusion is inspired by [71]. However, unlike their method, we apply it only at the input level rather than within the network, thus avoiding the need to replicate complex blocks from the frozen architecture. Moreover, in our design, the output of the final zero-initialized convolution is first embedded and then added to the color image tokens, instead of being added to the RGB input $\mathbf{I}$ before embedding. This distinction further differentiates our structure from theirs.

In addition, depth estimation relying solely on a single RGB image is fundamentally ill-posed because of scale ambiguity. Incorporating sparse depth measurements with accurate scale information helps to overcome this limitation.

3.3. Textual Prompt

As illustrated in Fig. 3, given a reference pixel [5] in the color image, we query the language decoder with a fixed prompt $\mathbf{Q}$ "*What is the distance from the current pixel to the camera?*". Prior to this, both the sparse depth map and its binary mask are converted into textual descriptions. In the mask, a value of zero indicates a missing-depth region, while a value of one denotes a valid depth measurement. Consequently, the textual description corresponding to the image pixel falls into two categories: "*The mask value is 0; predict the distance to the camera.*" and "*The mask value is 1, and the distance to the camera is x meters, preserve the given value.*". The template of this textual prompt is also fixed, with only the binary value and x dynamically extracted from the corresponding mask and sparse depth. The textual prompt is subsequently fed into the language decoder $\mathcal{F}_{\delta}(\cdot)$ along with the question $\mathbf{Q}$ to produce the final answer $\mathbf{A}$. The answer is like "*The current pixel is 4.230*

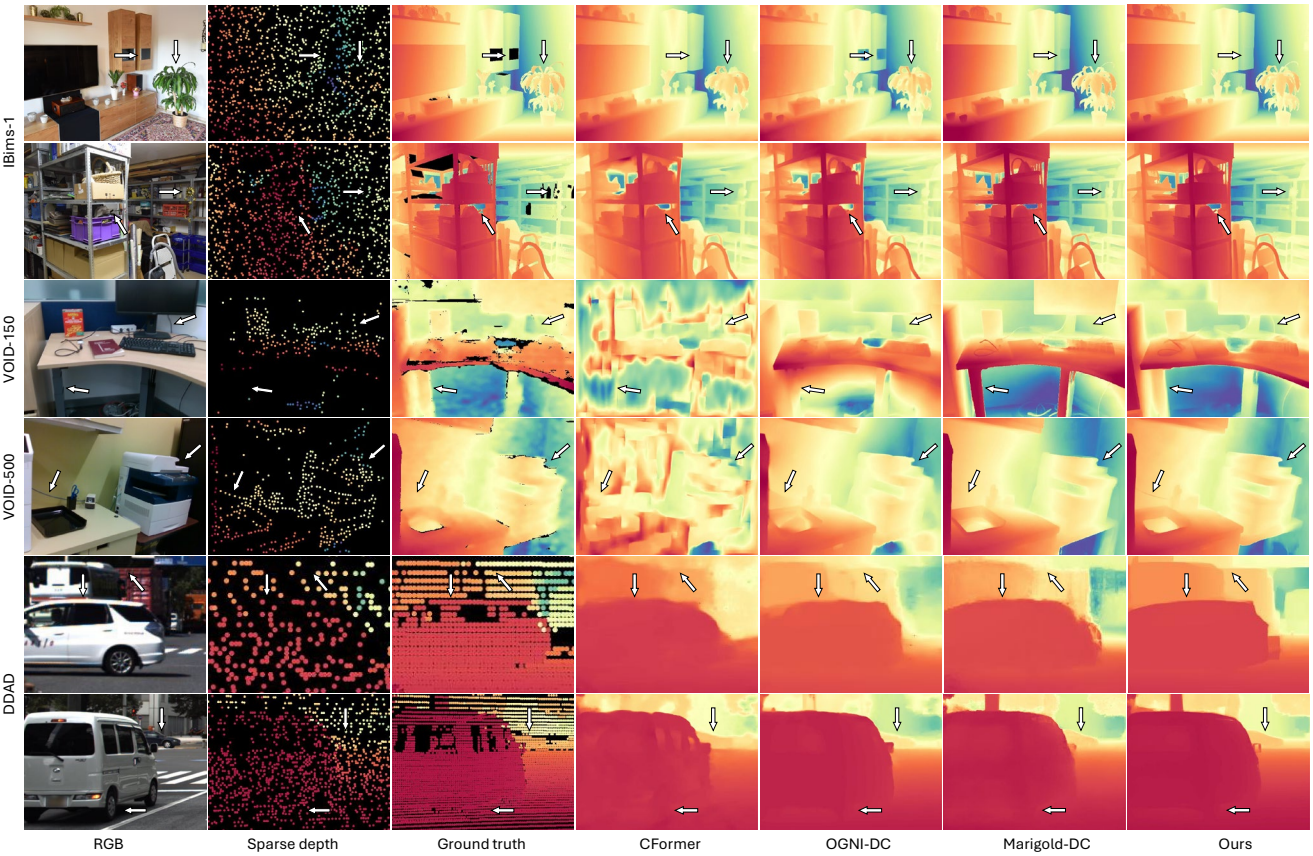

Figure 4. **Qualitative results on four benchmarks** comparing CFormer [74], OGNI-DC [75], Marigold-DC [47], and our approach.

meters away from the camera.". We formulate this process as follows:

$$\mathbf{T} = \mathcal{F}_t(\mathbf{S}), \tag{4a}$$

$$\mathbf{A} = \mathcal{F}_\delta(\mathbf{T}, \mathbf{Q}, \mathbf{V}), \tag{4b}$$

where $\mathcal{F}_t(\cdot)$ denotes the function that converts the sparse depth and its mask into textual descriptions. This design provides explicit guidance on the regions to predict and those to preserve, allowing the model to perform depth completion in a controllable manner guided by language.

3.4. Textual Supervision

The textual annotations are generated from the sparse depth $\mathbf{S}$ by inserting the depth value y into a fixed template: "*The pixel is y meters away from the camera.*" Following prior work [5], we adopt the text-based supervised fine-tuning [58], where only one labeled pixel is used per training sample, and the $\mathcal{L}_1$ loss is employed for supervision. In real-world applications, sparse depth information is more readily available alongside RGB images. Thus, our label-free strategy can also be interpreted as an online approach.

After obtaining the textual response for each pixel's depth value, we traverse all pixels and extract the predicted depths from these textual outputs, thereby reconstructing the dense depth map $\mathbf{D}$ of size $1 \times h \times w$:

$$\mathbf{D} = \mathcal{T}_{j=1}^{hw}(\mathbf{A}_j), \tag{5}$$

where $\mathcal{T}(\cdot)$ denotes the traversal function.

Table 1. **Overview of datasets used for fine-tuning and testing.** A subset of Hypersim and Virtual KITTI is utilized for fine-tuning.

Split	Dataset	Size	Sparse Depth	Scene Type
Fine-tune	Hypersim [40]	10K	Synthetic	Indoor
	Virtual KITTI [3]	10K	Synthetic	Urban
Test	IBims-1 [23]	100	Synthetic	Indoor
	VOID [60]	2400	SfM	Indoor
	NYUv2 [42]	654	Synthetic	Indoor
	KITTI [45]	1000	LiDAR	Urban
	DDAD [16]	3950	LiDAR	Urban

4. Experiment

4.1. Datasets

Following Marigold-DC [47], we evaluate the zero-shot performance of our method on five commonly used benchmarks, including IBims-1 [23], VOID [60], NYUv2 [42], KITTI [45], and DDAD [16]. Tab. 1 summarizes the characteristics of these datasets, which encompass both indoor

Table 2. **Zero-shot depth completion comparisons with state-of-the-art methods.** All results of other approaches are adopted from Marigold-DC [47]. The **best** and second-best metrics are highlighted. † denotes a least-squares estimate, and ‡ indicates the use of test-time ensembling. SD/GT means whether sparse depth or ground-truth depth is utilized to generate textual supervision during fine-tuning.

Method	IBims-1		VOID 150		VOID 500		VOID 1500		NYUv2		KITTI		DDAD	
	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓
NLSPN [35]	0.049	0.191	0.492	0.963	0.301	0.783	0.210	0.668	0.440	0.716	1.335	2.076	2.498	9.231
SpAgNet [12]	-	-	0.408	0.866	0.326	0.752	0.244	0.706	0.158	0.292	0.518	1.788	4.578	13.236
CFormer [74]	0.058	0.206	0.487	0.956	0.385	0.821	0.301	0.821	0.186	0.374	0.952	1.935	2.518	9.471
VPP4DC [2]	0.062	0.228	0.245	0.690	0.187	0.582	0.148	0.543	0.077	0.247	0.413	1.609	1.344	6.781
BPNet [44]	0.062	0.236	0.471	0.936	0.370	0.793	0.270	0.742	-	-	-	-	2.270	8.344
OGNI-DC [75]	0.059	0.186	0.261	0.693	0.198	0.589	0.175	0.593	-	-	-	-	1.867	6.876
DepthLab [31]	0.071	0.210	0.285	0.689	0.223	0.590	0.190	0.620	0.184	0.276	0.921	2.171	4.498	8.379
PromptDA [26]	0.063	0.197	0.265	0.702	0.194	0.561	0.170	0.595	0.110	0.233	0.932	2.171	2.107	7.494
Ours (w/ GT)	**0.036**	**0.151**	**0.176**	**0.592**	**0.138**	**0.495**	**0.137**	**0.493**	**0.042**	**0.120**	**0.406**	**1.324**	**1.340**	**6.179**
Improvement ↑	*36.1%*	*23.2%*	*39.2%*	*14.1%*	*35.5%*	*13.3%*	*8.0%*	*10.1%*	*83.3%*	*94.2%*	*1.7%*	*21.5%*	*0.3%*	*9.7%*
Marigold [22]	0.071	0.230	0.279	0.687	0.221	0.625	0.191	0.652	0.194	0.309	1.765	3.361	22.872	32.661
Marigold † [22]	0.069	0.213	0.266	0.670	0.204	0.598	0.180	0.628	0.190	0.294	1.709	3.305	2.817	14.728
Marigold-DC [47]	0.062	0.205	0.201	0.629	0.167	0.546	0.157	0.557	0.057	0.142	0.558	1.676	2.985	7.905
Marigold-DC ‡ [47]	0.045	0.166	0.194	0.622	0.158	0.535	0.152	0.551	0.048	0.124	0.434	1.465	2.364	6.449
Ours (w/ SD)	**0.040**	**0.158**	**0.185**	**0.604**	**0.146**	**0.514**	**0.141**	**0.512**	**0.044**	**0.121**	**0.418**	**1.394**	**1.353**	**6.264**
Improvement ↑	*12.5%*	*5.1%*	*4.9%*	*3.0%*	*8.2%*	*4.1%*	*7.8%*	*7.6%*	*9.1%*	*2.5%*	*3.8%*	*5.1%*	*74.7%*	*3.0%*

and outdoor environments and cover a wide range of image resolutions, sensor types, and depth sparsity levels.

NYUv2 [42] provides indoor RGB-D scenes captured using a Microsoft Kinect sensor. Following common practice [34, 35, 62], we adopt the official test split containing 654 samples. Images are downsampled to 320×240 and center-cropped to 304×228. The sparse depth input is generated by randomly sampling 500 valid points from the ground-truth depth maps.

VOID [59] contains synchronized RGB and depth streams captured via active stereo sensors in both indoor and outdoor settings at a resolution of 640×480. We use all 800 frames from the eight designated test sequences, together with their provided sparse depth maps at three density levels of 150, 500, and 1500 points.

IBims-1 [23] is a high-quality indoor RGB-D dataset acquired with a laser scanner, characterized by sharp depth discontinuities, minimal noise, and accurate geometry up to 50 m. We evaluate on all 100 images at 640×480 resolution, sampling 1000 valid depth points from the intersection of valid masks while excluding invalid, transparent, and missing pixels according to the official evaluation protocol.

KITTI [45] consists of outdoor driving scenes with paired RGB images and sparse LiDAR measurements at 1216×352 resolution. The semi-dense ground truth is obtained by temporally aggregating multiple LiDAR frames. We adopt the official validation split of 1000 pairs and remove outliers based on [11, 47].

DDAD [16] is a large-scale self-driving dataset featuring a 360° multi-camera setup and long-range LiDAR up to 250m. The test set includes 3950 samples per camera at 1936×1216 resolution. Following prior works [2, 47, 75], we use only the front-facing view and randomly retain approximately 20% of the available LiDAR depth as sparse input, applying the same outlier filtering as for KITTI [45].

4.2. Evaluation and Implementation Details

Marigold [22] is trained on the synthetic Hypersim [40] and Virtual KITTI [3] datasets, comprising approximately 74K indoor and outdoor RGB-D pairs. Building on this, Marigold-DC [47] introduces test-time optimization to incorporate sparse depth priors for zero-shot evaluation. In contrast, our baseline is established on Qwen2.5-VL (3B) [1] and [5], which we fine-tune using the proposed sparse depth injection mechanism on a 20K subset of Hypersim and Virtual KITTI. It is worth noting that neither the pretrained baseline nor the fine-tuning process involves any data from NYUv2, VOID, IBims-1, KITTI, or DDAD.

The fine-tuning is performed in PyTorch on 8×48 GB GPUs for 10 epochs, with a total batch size of 16. Following previous depth completion works [2, 47, 75], we evaluate performance using the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE), both measured in meters, which respectively quantify the average prediction deviation and overall reconstruction fidelity.

4.3. Comparisons with State-of-the-arts

In this section, we compare our method with 9 well-known approaches: NLSPN [35], SpAgNet [12], CFormer [74], VPP4DC [2], BPNet [44], OGNI-DC [75], DepthLab [31],

Table 3. **Comparisons with different VLM backbones** on the IBims-1 dataset. GE denotes gradient error.

VLM Backbone	MAE (m) ↓	RMSE (m) ↓	GE (m) ↓	$\delta_{1.15}$ ↑
MolmoE-1B [14]	0.054	0.166	0.357	0.882
Seed1.5-VL [17]	<u>0.048</u>	<u>0.162</u>	<u>0.350</u>	<u>0.908</u>
Qwen2.5-VL [1]	**0.040**	**0.158**	**0.348**	**0.933**

Table 4. **Complexity comparisons** on the IBims-1 dataset. All methods are evaluated using a single 48GB 4090 GPU.

Method	Memory (GB) ↓	Speed (FPS) ↑	RMSE (m) ↓
CFormer [74]	1.57	0.324	0.206
OGNI-DC [75]	<u>0.71</u>	<u>4.348</u>	0.186
BPNet [44]	**0.69**	**4.545**	0.236
Marigold-DC [47]	8.64	0.005	<u>0.166</u>
Ours (w/ SD)	46.10	0.327	**0.158**

PromptDA [26], Marigold [22], and Marigold-DC [47]. NLSPN, SpAgNet, CFormer, and BPNet emphasize highly accurate depth completion through SPN-based refinement, while VPP4DC and OGNI-DC improve generalization by treating completion as stereo matching and through depth gradient field optimization, respectively. For enhanced zero-shot performance, PromptDA utilizes the depth foundation model [67], whereas Marigold and Marigold-DC rely on latent diffusion for depth completion.

Tab. 2 presents the quantitative comparisons. Overall, our method outperforms the baselines in most cases by a large margin. In particular, compared to the Marigold family, which only leverages RGB and sparse depth, our model (w/ SD) achieves remarkable performance. For example, it surpasses the second-best counterparts in MAE by 12.5% on IBims-1, 4.9% on VOID 150, 9.1% on NYUv2, and 3.8% on KITTI. On the other hand, even when compared with those that utilize ground-truth depth for supervision, our approach (w/ GT) consistently performs better, with improvements of 13.3% on VOID 500, 10.1% on VOID 1500, and 9.7% on DDAD in RMSE. Notably, our model (w/ SD) still outperforms most of these ground-truth-supervised zero-shot methods.

Fig. 4 shows visual comparisons with CFormer, OGNI-DC, and Marigold-DC across four benchmarks. It can be found that our method predicts more accurate depth results, with sharper boundaries and more complete scene content. In Fig. 5, the density of the ground-truth depth map is 300 times higher than that of the sparse depth, offering substantially richer pixel-level supervision, particularly for regions absent in the sparse input. Consequently, the depth prediction (w/ GT) yields more satisfactory results.

In addition, Tab. 3 presents a comparison of our method using different VLMs as backbones, including MolmoE-1B, Seed1.5-VL, and Qwen2.5-VL. The results show that

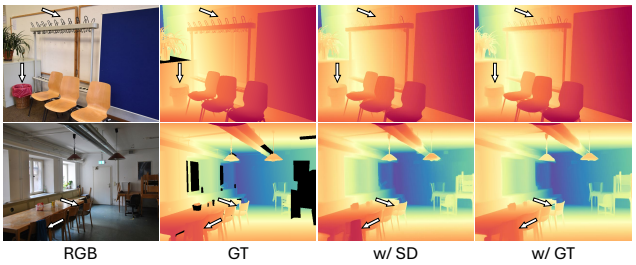

Figure 5. **Visual comparisons of our method supervised by sparse depth and ground-truth depth** on the IBims-1 dataset.

Qwen2.5-VL achieves superior performance across both error and accuracy metrics. Tab. 4 reports the comparisons of memory usage and inference speed among CFormer, OGNI-DC, BPNet, Marigold-DC, and our method. As observed, compared with task-specific models (CFormer and BPNet), our approach achieves substantially lower errors but incurs higher memory consumption due to the preloading of the VLM. In contrast, when compared with the zero-shot-oriented Marigold-DC, our method not only attains higher performance but also runs approximately 65 times faster. These results indicate that our approach achieves a favorable trade-off between accuracy and efficiency.

4.4. Ablation Studies

Tab. 5 presents the ablation results of our sparse depth injection mechanism (SDIM), including visual tokenization, textual prompt, and textual supervision. To comprehensively evaluate the effectiveness of each component, we conduct experiments on the VOID 150 dataset, which contains more challenging and error-prone samples.

Visual Tokenization. As mentioned earlier, the first layer of Qwen2.5-VL is a 3D convolution that embeds image inputs into visual tokens. Before the embedding, SDIM-(a) directly concatenates the RGB-D input and then feeds it into the 3D convolution. SDIM-(b) introduces our fusion block, where the first and last convolutions employ He initialization [18]. The latter achieves better performance, as it enables a more gradual fusion between the depth and the frozen image tokens. Building upon this, SDIM-(c) adopts zero initialization, resulting in 7mm reduction in RMSE. This improvement can be attributed to the smooth and stable fusion, where the integration of RGB-D tokens produces representations that effectively incorporate depth cues while maintaining a distribution close to that of the pretrained image tokens. Such similarity facilitates more coherent interaction with the pretrained image tokens, leading to smoother adaptation and improved depth completion performance. In summary, SDIM-(a) and (b) can be regarded as hard embedding schemes that directly inject depth information into image tokens, whereas SDIM-(c) adopts a soft embedding strategy that integrates depth cues in a more effective and stable manner.

Table 5. Ablation studies of our sparse depth injection mechanism (SDIM) on VOID dataset with 150 depth points.

SDIM	Visual Tokenization			Textual Prompt		Textual Supervision		MAE (m)	RMSE (m)
	Concat	He init. [18]	Zero init.	Mask	SD	SD	GT		
(a)	✓					✓		0.206	0.635
(b)		✓				✓		0.202	0.630
(c)		✓	✓			✓		0.197	0.623
(d)		✓	✓	✓		✓		0.188	0.608
(e)		✓	✓	✓	✓	✓		0.185	0.604
(f)		✓	✓	✓	✓		✓	0.176	0.592

Textual Prompt. Based on SDIM-(c), SDIM-(d) converts the binary mask derived from sparse depth into textual descriptions, yielding 9mm and 15mm improvements in MAE and RMSE, respectively. This textual guidance explicitly conveys the existence of observed regions to the VLM, encouraging it to infer geometrically consistent depths in unseen areas. Furthermore, when incorporating the specific depth values from the sparse input, SDIM-(e) achieves additional error reductions by introducing more precise geometric cues. In other words, when a valid depth value exists at a pixel, the model preserves this value and estimates the surrounding missing depths by referencing the current pixel's depth. Note that the sparse depth values are theoretically identical to those in the ground-truth depth, which ensures the accuracy of the predicted results at these locations and stabilizes the overall completion process.

Textual Supervision. Given the textual prompt, we query the model for the distance between a pixel and the camera, while the supervision signal provides the corresponding answers for training. As the ground-truth depth is typically one to two orders of magnitude denser than the sparse depth, it can provide precise feedback across increasingly large missing regions. As a result, the dense supervision in SDIM-(f) enables the model to achieve more accurate depth estimation than the sparse supervision in SDIM-(e). However, the performance gap between them is relatively small, and SDIM-(e), being label-free, is more cost-effective and easier to deploy in real-world applications.

4.5. Failure Case

Fig. 6 illustrates failure cases of our method near transparent or reflective surfaces, such as glass. As observed from the sparse or ground-truth depth maps, these regions typically contain few or no valid depth values. In some cases, they even penetrate through the glass to objects behind it, resulting in incorrect depth measurements. Consequently, our model struggles to produce precise predictions on glass surfaces across different scenes. Moreover, while our method performs reasonably well on distant car windows, it often fails on nearby ones. Beyond the aforementioned reasons, we observe that objects behind the glass are more clearly visible in RGB images when the glass is close to the camera, causing misleading guidance. This effect is much less pronounced for distant glass surfaces.

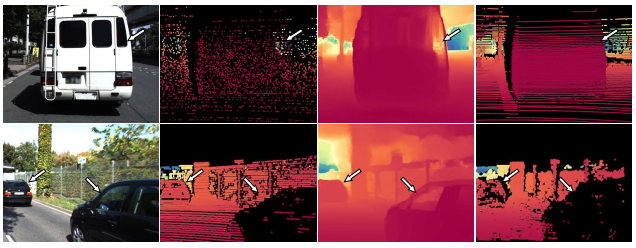

Figure 6. **Failure cases** of our approach near glass surfaces.

A potential solution is to introduce glass segmentation and integrate it into the visual tokenization stage. Meanwhile, textualizing the segmentation could further guide the model in better handling these challenging regions.

5. Conclusion

In this paper, we present the first framework that adapts vision language models (VLMs) for the task of depth completion. By injecting sparse depth cues through visual tokenization, textual prompt, and textual supervision, our approach enables VLMs to infer dense depth without relying on ground-truth annotations or architectural modifications. The proposed method exhibits strong zero-shot generalization across diverse datasets and sensing conditions, revealing the potential of VLMs to extend beyond semantic reasoning into geometric perception. We believe this study takes a promising step toward leveraging language-guided reasoning for geometry-aware visual understanding.

Limitation. Despite its promising results, our method still has certain limitations. (1) It primarily focuses on depth prediction, leaving open the question of whether such a powerful VLM can be effectively extended to other dense prediction tasks. (2) Since our framework is built upon a large VLM, the inference speed remains relatively slow. Future work will therefore explore extending our framework to broader geometry-related tasks (e.g., surface normal prediction) and developing a more lightweight and efficient model to improve inference speed and computational efficiency.

Acknowledgment

This research/project is supported by the National Research Foundation (NRF) Singapore, under its NRF-Investigatorship Programme (Award ID. NRF-NRFI09-0008).

References

[1] 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. 2, 4, 6, 7

[2] Luca Bartolomei, Matteo Poggi, Andrea Conti, Fabio Tosi, and Stefano Mattoccia. Revisiting depth completion from a stereo matching perspective for cross-domain generalization. In *3DV*, pages 1360–1370. IEEE, 2024. 6

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 5, 6

[4] 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*, pages 9490–9498. IEEE, 2025. 3

[5] Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. Depthlm: Metric depth from vision language models. *arXiv preprint arXiv:2509.25413*, 2025. 3, 4, 5, 6

[6] 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*, pages 14455–14465, 2024. 3

[7] Wei Chen, Changyong Shi, Chuanxiang Ma, Wenhao Li, and Shulei Dong. Depthblip-2: Leveraging language to guide blip-2 in understanding depth information. In *ACCV*, pages 2939–2953, 2024. 3

[8] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *ECCV*, pages 103–119, 2018. 3

[9] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2361–2379, 2019. 3

[10] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *AAAI*, pages 10615–10622, 2020. 3

[11] Andrea Conti, Matteo Poggi, Filippo Aleotti, and Stefano Mattoccia. Unsupervised confidence for lidar depth maps and applications. In *IROS*, pages 8352–8359. IEEE, 2022. 6

[12] Andrea Conti, Matteo Poggi, and Stefano Mattoccia. Sparsity agnostic depth completion. In *WACV*, pages 5871–5880, 2023. 6

[13] Beilei Cui, Yiming Huang, Long Bai, and Hongliang Ren. Tr2m: Transferring monocular relative depth to metric depth with language descriptions and scale-oriented contrast. *arXiv preprint arXiv:2506.13387*, 2025. 3

[14] 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*, pages 91–104, 2025. 7

[15] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2423–2436, 2020. 2

[16] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, pages 2485–2494, 2020. 5, 6

[17] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1.5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025. 3, 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 7, 8

[19] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *ICRA*, 2021. 2, 3

[20] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 3

[21] Xueting Hu, Ce Zhang, Yi Zhang, Bowen Hai, Ke Yu, and Zhihai He. Learning to adapt clip for few-shot monocular depth estimation. In *WACV*, pages 5594–5603, 2024. 3

[22] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, pages 9492–9502, 2024. 1, 3, 6, 7

[23] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *ECCVW*, pages 0–0, 2018. 5, 6

[24] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *CRV*, pages 16–22, 2018. 2

[25] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *CRV*, pages 16–22. IEEE, 2018. 2

[26] Haotong Lin, Sida Peng, Jingxiao Chen, Songyou Peng, Jiaming Sun, Minghuan Liu, Hujun Bao, Jiashi Feng, Xiaowei Zhou, and Bingyi Kang. Prompting depth anything for 4k resolution accurate metric depth estimation. In *CVPR*, pages 17070–17080, 2025. 1, 3, 6, 7

[27] Yuankai Lin, Hua Yang, Tao Cheng, Wending Zhou, and Zhouping Yin. Dyspn: Learning dynamic affinity for image-guided depth completion. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2023. 3

[28] Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Fcfr-net: Fea-

ture fusion based coarse-to-fine residual learning for depth completion. In *AAAI*, pages 2136–2144, 2021. 2

[29] Minghuan Liu, Zhengbang Zhu, Xiaoshen Han, Peng Hu, Haotong Lin, Xinyao Li, Jingxiao Chen, Jiafeng Xu, Yichu Yang, Yunfeng Lin, et al. Manipulation as in simulation: Enabling accurate geometry perception in robots. *arXiv preprint arXiv:2509.02530*, 2025. 1

[30] Xin Liu, Xiaofei Shao, Bo Wang, Yali Li, and Shengjin Wang. Graphcspn: Geometry-aware depth completion via dynamic gcns. In *ECCV*, pages 90–107. Springer, 2022. 3

[31] Zhiheng Liu, Ka Leong Cheng, Qiuyu Wang, Shuzhe Wang, Hao Ouyang, Bin Tan, Kai Zhu, Yujun Shen, Qifeng Chen, and Ping Luo. Depthlab: From partial to complete. *arXiv preprint arXiv:2412.18153*, 2024. 1, 6

[32] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. From depth what can you see? depth completion via auxiliary image reconstruction. In *CVPR*, pages 11306–11315, 2020. 2

[33] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *ICRA*, pages 4796–4803. IEEE, 2018. 2

[34] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *ICRA*, 2019. 1, 2, 6

[35] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, 2020. 1, 2, 3, 6

[36] Jin-Hwi Park and Hae-Gon Jeon. A simple yet universal framework for depth completion. *NeurIPS*, 37:23577–23602, 2024. 1, 2, 3

[37] Jin-Hwi Park, Chanhwi Jeong, Junoh Lee, and Hae-Gon Jeon. Depth prompting for sensor-agnostic depth estimation. In *CVPR*, pages 9859–9869, 2024. 1, 2, 3

[38] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *CVPR*, pages 10106–10116, 2024. 1

[39] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *CVPR*, pages 6250–6259, 2022. 1, 2

[40] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, pages 10912–10922, 2021. 5, 6

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1

[42] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 5, 6

[43] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020. 1, 2

[44] Jie Tang, Fei-Peng Tian, Boshi An, Jian Li, and Ping Tan. Bilateral propagation network for depth completion. In *CVPR*, pages 9763–9772, 2024. 1, 2, 6, 7

[45] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *3DV*, pages 11–20, 2017. 1, 2, 5, 6

[46] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *MVA*, pages 1–6, 2019. 1, 2

[47] Massimiliano Viola, Kevin Qu, Nando Metzger, Bingxin Ke, Alexander Becker, Konrad Schindler, and Anton Obukhov. Marigold-dc: Zero-shot monocular depth completion with guided diffusion. In *ICCV*, pages 5359–5370, 2025. 1, 2, 3, 5, 6, 7

[48] Haotian Wang, Meng Yang, and Nanning Zheng. G2-monodepth: A general framework of generalized depth inference from monocular rgb+x data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3753–3771, 2024. 1, 2, 3

[49] Haotian Wang, Aoran Xiao, Xiaoqin Zhang, Meng Yang, and Shijian Lu. Pacgdc: Label-efficient generalizable depth completion with projection ambiguity and consistency. In *ICCV*, pages 7709–7720, 2025. 1, 2, 3

[50] Jiyuan Wang, Chunyu Lin, Lang Nie, Shujun Huang, Yao Zhao, Xing Pan, and Rui Ai. Weatherdepth: Curriculum contrastive learning for self-supervised depth estimation under adverse weather conditions. In *ICRA*, pages 4976–4982. IEEE, 2024. 1

[51] Jiyuan Wang, Chunyu Lin, Lang Nie, Kang Liao, Shuwei Shao, and Yao Zhao. Digging into contrastive learning for robust depth estimation with diffusion models. In *ACM MM*, pages 4129–4137, 2024. 1

[52] Jiyuan Wang, Chunyu Lin, Cheng Guan, Lang Nie, Jing He, Haodong Li, Kang Liao, and Yao Zhao. Jasmine: Harnessing diffusion prior for self-supervised depth estimation. *arXiv preprint arXiv:2503.15905*, 2025. 1

[53] JiYuan Wang, Chunyu Lin, Lei Sun, Rongying Liu, Lang Nie, Mingxing Li, Kang Liao, Xiangxiang Chu, and Yao Zhao. From editor to dense geometry estimator. *arXiv preprint arXiv:2509.04338*, 2025. 1

[54] Jiyuan Wang, Chunyu Lin, Lei Sun, Zhi Cao, Yuyang Yin, Lang Nie, Zhenlong Yuan, Xiangxiang Chu, Yunchao Wei, Kang Liao, et al. Geometry-guided reinforcement learning for multi-view consistent 3d scene editing. *arXiv preprint arXiv:2603.03143*, 2026. 1

[55] Yufei Wang, Bo Li, Ge Zhang, Qi Liu, Tao Gao, and Yuchao Dai. Lrru: Long-short range recurrent updating networks for depth completion. In *ICCV*, pages 9422–9432, 2023. 2, 3

[56] Yiran Wang, Jiaqi Li, Chaoyi Hong, Ruibo Li, Liusheng Sun, Xiao Song, Zhe Wang, Zhiguo Cao, and Guosheng Lin. Tacodepth: Towards efficient radar-camera depth estimation with one-stage fusion. In *CVPR*, pages 10523–10533, 2025. 1

[57] Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025. 1

[58] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and

Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 5

[59] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *ICCV*, pages 12747–12756, 2021. 1, 6

[60] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2): 1899–1906, 2020. 5

[61] 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

[62] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *ECCV*, pages 214–230, 2022. 1, 2, 6

[63] Zhiqiang Yan, Yuankai Lin, Kun Wang, Yupeng Zheng, Yufei Wang, Zhenyu Zhang, Jun Li, and Jian Yang. Tri-perspective view decomposition for geometry-aware depth completion. In *CVPR*, pages 4874–4884, 2024. 2, 3

[64] Zhiqiang Yan, Jianhao Jiao, Zhengxue Wang, and Gim Hee Lee. Event-driven dynamic scene depth completion. *arXiv preprint arXiv:2505.13279*, 2025. 1

[65] Zhiqiang Yan, Xiang Li, Le Hui, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet++: Semantic assisted repetitive image guided network for depth completion: Z. yan et al. *International Journal of Computer Vision*, pages 1–23, 2025. 2

[66] Zhiqiang Yan, Zhengxue Wang, Kun Wang, Jun Li, and Jian Yang. Completion as enhancement: A degradation-aware selective image guided network for depth completion. In *CVPR*, pages 26943–26953, 2025. 2

[67] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 37:21875–21911, 2024. 1, 3, 7

[68] Ziyao Zeng, Jingcheng Ni, Daniel Wang, Patrick Rim, Younjoon Chung, Fengyu Yang, Byung-Woo Hong, and Alex Wong. Iris: Integrating language into diffusion-based monocular depth estimation. *arXiv preprint arXiv:2411.16750*, 2024. 3

[69] Ziyao Zeng, Daniel Wang, Fengyu Yang, Hyoungseob Park, Stefano Soatto, Dong Lao, and Alex Wong. Wordepth: Variational language prior for monocular depth estimation. In *CVPR*, pages 9708–9719, 2024. 3

[70] Ziyao Zeng, Yangchao Wu, Hyoungseob Park, Daniel Wang, Fengyu Yang, Stefano Soatto, Dong Lao, Byung-Woo Hong, and Alex Wong. Rsa: Resolving scale ambiguities in monocular depth estimators through language descriptions. *NeurIPS*, 37:112684–112705, 2024. 3

[71] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 4

[72] Renrui Zhang, Ziyao Zeng, Ziyu Guo, and Yafeng Li. Can language understand depth? In *ACM MM*, pages 6868–6874, 2022. 3

[73] Wenyao Zhang, Hongsi Liu, Bohan Li, Jiawei He, Zekun Qi, Yunnan Wang, Shengyang Zhao, Xinqiang Yu, Wenjun Zeng, and Xin Jin. Hybrid-grained feature aggregation with coarse-to-fine language guidance for self-supervised monocular depth estimation. In *ICCV*, pages 6678–6692, 2025. 3

[74] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *CVPR*, pages 18527–18536, 2023. 1, 2, 5, 6, 7

[75] Yiming Zuo and Jia Deng. Ogni-dc: Robust depth completion with optimization-guided neural iterations. In *ECCV*, pages 78–95. Springer, 2024. 1, 3, 5, 6, 7

[76] Yiming Zuo, Willow Yang, Zeyu Ma, and Jia Deng. Omni-dc: Highly robust depth completion with multiresolution depth integration. In *ICCV*, pages 9287–9297, 2025. 1, 2, 3