2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

OASIS-DC: Generalizable Depth Completion via Output-level Alignment of Sparse-Integrated Monocular Pseudo Depth

Jaehyeon Cho and Jhonghyun An*

***Abstract*— Recent monocular *foundation* models excel at zero-shot depth estimation, yet their outputs are inherently *relative* rather than metric, limiting direct use in robotics and autonomous driving. We leverage the fact that relative depth preserves global layout and boundaries: by calibrating it with sparse range measurements, we transform it into a pseudo *metric* depth prior. Building on this prior, we design a refinement network that follows the prior where reliable and deviates where necessary, enabling accurate metric predictions from very few labeled samples. The resulting system is particularly effective when curated validation data are unavailable, sustaining stable scale and sharp edges across few-shot regimes. These findings suggest that coupling foundation priors with sparse anchors is a practical route to robust, deployment-ready depth completion under real-world label scarcity.**

I. INTRODUCTION

Depth completion—inferring dense, metric depth from sparse measurements guided by RGB—is a key enabler for robust perception in robotics and autonomous driving. Despite advances on the KITTI benchmark [1], [2] and strong RGB–LiDAR fusion methods [3], [4], [5], prevailing pipelines assume access to large labeled sets and substantial validation curation. In deployed settings, operating conditions change faster than labels are collected, requiring models to generalize from very few samples. Concurrently, recent *foundation-guided* approaches *couple* depth completion to high-dimensional features of monocular foundation depth estimators (MDE) [6], [7]. This feature-level coupling increases memory/latency and complicates deployment.

We address these challenges with a *prior-guided* framework operating at the *output level*: rather than consuming foundation features, we use only the dense depth *output* of a generalizing MDE (e.g., ZoeDepth [8], Depth Anything [9]) and *align* it with sparse anchors to form a calibrated pseudo-depth prior. Concretely, we correct MDE predictions with sparse points via a Poisson formulation with hard constraints to obtain a metrically accurate, edge-preserving map. We feed this prior to a lightweight refinement network learning only a residual. The model follows the prior where consistent, deviating via residual corrections in regions of mismatch. This output-level pairing leverages MDE generalization while sparse anchors fix absolute scale, *shrinking the hypothesis space* and preserving few-shot stability *without* feature-level overhead, yielding a compact, deployment-ready core.

We evaluate primarily on KITTI Depth Completion [1] and additionally verify indoor generalization on NYUv2 [24]. To assess comparability and deployment realism, we adopt two settings: a *standard few-shot* evaluation and a *strict, deployment-oriented* regime where training and evaluation share the same few-shot budget (detailed in §IV). This probes whether reliable generalization is attainable without a large validation set.

Contributions. (1) A *nonlearned*, MDE-guided pseudo-depth construction that aligns a foundation MDE to sparse anchors, reconstructing a metrically accurate prior via a Poisson formulation with hard constraints without trainable parameters. (2) A lightweight prior-guided network that ingests this pseudo map and learns only a residual refinement under few-shot supervision, yielding sharp boundaries and robust performance under severe data scarcity—*all while avoiding feature-level coupling* for practical deployability.

II. RELATED WORK

A. Traditional depth completion.

Image-guided depth completion from sparse range measurements has progressed via RGB–LiDAR fusion and learned affinity/propagation within encoder–decoder frameworks [12], [13], [14]. While these approaches attain strong accuracy on curated benchmarks such as KITTI Depth Completion [1], [2], they typically presume abundant pixel-wise supervision and a fixed sparsity pattern, which limits robustness when sensor characteristics, collection routes, or environments shift. In practice, dependence on large training/validation splits and tuning to a specific LiDAR density/pattern often impairs cross-sensor and cross-domain generalization.

B. Few-shot depth completion and deployment constraints.

In deployed robotics/AutonomousDriving(AD) settings, operating conditions (city, route, rig) evolve faster than labels can be curated; models must generalize from very few labeled samples and with minimal validation. Leaderboard-style protocols typically rely on full training splits and extensive validation curation on KITTI [1], [2]. Complementarily, we emphasize deployment-oriented few-shot evaluation that probes generalization under strict data scarcity—e.g., restricting both training and evaluation to the same few-shot subsets and reporting performance on a compact, fixed validation set (KITTI provides an official 1,000-frame validation split) [1].

This work was supported by the Project for Collaboration R&D between Industry, University, and Research Institute, funded by the Ministry of SMEs and Startups of Korea in 2025 (RS-2025-02220569).

The authors are with the Department of Artificial Intelligence, Gachon University, Seongnam-si, Republic of Korea (e-mail: jjh000503@gachon.ac.kr; jhonghyun@gachon.ac.kr).

*Corresponding author.

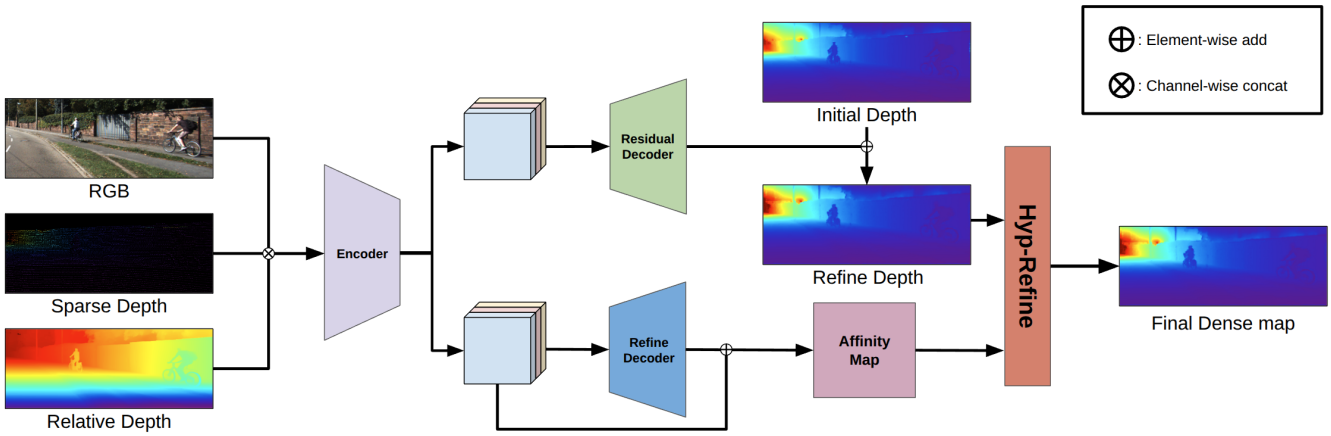

Fig. 1. **Pipeline overview.** RGB, sparse depth, and relative (monocular) depth are concatenated and encoded. A residual decoder predicts an *initial* depth anchored to sparse points, while a refinement branch estimates an *affinity map* for edge/structure–aware propagation. The two streams are fused and passed through a hyperbolic refinement module (Hyp-Refine) to yield the final dense map. Symbols: $\oplus$ element-wise add, $\otimes$ channel-wise concat.

Such protocols directly test whether a stable metric scale and sharp boundaries are attainable without large validation curation.

C. Foundation-guided completion.

Recent work leverages foundation MDEs to reduce data requirements and improve cross-sensor generalization. DepthPrompting[6] introduces a depth-prompt module that encodes sparse depth and fuses it with image features to construct pixel-wise affinity, embedding the prompt into a pre-trained MDE to mitigate sensor-range/pattern biases and steer predictions toward absolute-scale depth with lightweight bias tuning. UniDC [7] defines universal depth completion and presents a simple baseline that (i) extracts depth-aware features from a foundation MDE, (ii) aligns arbitrary sparse measurements via a pixel-wise affinity built on high-resolution foundation features, and (iii) embeds learned features in a hyperbolic space to capture hierarchical 3D structure, thereby improving zero-/few-shot adaptation across sensors [6], [7]. Both lines capitalize on broad appearance priors from foundation MDEs such as ZoeDepth and Depth Anything [8], [9] and illustrate the value of coupling them with sparse anchors. While effective, such *feature-level* coupling to the foundation MDE backbone [6], [7] often increases runtime memory/latency and complicates deployment. In contrast, we operate at the *output level*: we use only the dense depth *output* of a frozen MDE to build a calibrated pseudo-depth prior anchored by sparse points.

III. Method

A. Pseudo-Depth Map Construction

To stabilize training and model selection under limited-data conditions, we construct a *pseudo-depth* map P by fusing a dense monocular prior $E \in \mathbb{R}^{H\times W}$ with sparse LiDAR depth $D \in \mathbb{R}^{H\times W}$.

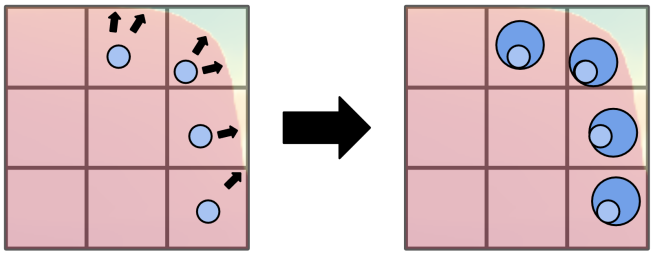

Fig. 2. **Gradient-guided densification.** The aligned prior induces a smooth gradient field (background), while sparse metric anchors (blue) and the image boundary are kept fixed. Depth values are propagated along the prior's gradient directions to fill unknown cells, preserving structure and avoiding cross-edge bleeding.

a) Sets and boundary data.: Let Ω denote the image domain with boundary $\partial\Omega$. We first define the set of known pixels $\mathcal{K}$ and its complement $\mathcal{U}$:

$$\begin{aligned} \mathcal{K} &= \{(i,j) \in \Omega \mid D_{ij} > 0\} \cup \partial\Omega, \\ \mathcal{U} &= \Omega \setminus \mathcal{K}. \end{aligned} \tag{1}$$

Based on these sets, we define the Dirichlet field $v_{\text{known}}(i,j)$ used for hard constraints:

$$v_{\text{known}}(i,j) = \begin{cases} D_{ij}, & (i,j) \in \{D > 0\}, \\ E_{ij}, & (i,j) \in \partial\Omega. \end{cases} \tag{2}$$

b) Gradient-domain guidance.: Following gradient-domain fusion [15], we align the unknown solution x to the guidance gradients ∇E while exactly honoring measurements and boundary values:

$$\min_x \int_\Omega \|\nabla x - \nabla E\|^2 \, dp \quad \text{s.t. } x|_{\mathcal{K}} = v_{\text{known}}. \tag{3}$$

The Euler–Lagrange condition yields a Poisson equation on the unknown set $\mathcal{U}$:

$$A_{\mathcal{U}\mathcal{U}}\, x_{\mathcal{U}} = \big(LE - Lv_{\text{known}}\big)_{\mathcal{U}}. \tag{4}$$

$$P = x = x_{\mathcal{U}} + v_{\text{known}}. \tag{5}$$

Equations (3)–(5) define a strictly convex problem with a unique solution.

$$P(p) = x(p) = \begin{cases} v_{\text{known}}(p), & p \in \mathcal{K}, \\ x_{\mathcal{U}}(p), & p \in \mathcal{U}. \end{cases} \tag{6}$$

c) Discretization and solver.: On the pixel grid we use the (negative) discrete Laplacian matrix L. To preserve discrete consistency, we take the right-hand side as LE (the discrete Laplacian of E) and restrict the linear system to the unknown index set $\mathcal{U}$, yielding (5); reconstruction follows (6). Here $A_{\mathcal{U}\mathcal{U}}$ is L restricted to $\mathcal{U}$. The system is symmetric positive definite (SPD), enabling efficient solution via conjugate gradients (CG) [16]. We implement $A_{\mathcal{U}\mathcal{U}}$ in a matrix-free manner by masking a single 3×3 convolution that realizes L; each CG iteration requires $O(HW)$ work and only image-sized auxiliary memory.

B. Residual Correction for Monocular Prior

a) Motivation.: Under limited-data training with scarce validation, purely data-driven predictors tend to exhibit high variance and unstable convergence, especially around thin structures and depth discontinuities. We therefore adopt a *prior-preserving residual update*: a reliable but imperfect monocular prior provides the global metric scaffold, while a low-capacity decoder corrects only localized errors. This reduces the effective hypothesis space and improves sample efficiency without sacrificing scene-scale consistency.

b) Input representation and encoder.: We construct the input tensor X by spatially aligning and concatenating the following channels:

$$X = \big[I,\ P,\ E,\ M_L\big]. \tag{7}$$

where I is the RGB image; P is the pseudo-depth prior obtained by Poisson fusion of a monocular estimate with LiDAR anchors; E is the dense monocular prior; and M_L is the binary LiDAR observation mask. Prior to concatenation, we normalize both the pseudo depth and the monocular prior to the unit range using the dataset maximum depth:

$$P \leftarrow \text{clip}\Big(\tfrac{P}{d_{\max}}, 0, 1\Big), \quad E \leftarrow \text{clip}\Big(\tfrac{E}{d_{\max}}, 0, 1\Big).$$

With slight abuse of notation, we continue to denote the normalized maps by P and E. All losses are computed in this normalized space. At inference, predictions are rescaled back to meters via $D = d_{\max}\hat{D}$ and, if needed, clamped to $[0, d_{\max}]$. A vision encoder Φ produces multi-modal features $F = \Phi(X)$. We intentionally constrain the encoder capacity to mitigate overfitting and stabilize optimization in the few-shot regime.

c) Residual decoder and prior-preserving update.: A shallow residual decoder predicts a per-pixel correction $R = f_{\text{dec}}(F)$. The corrected initialization is obtained by adding the residual to the prior:

$$D^0(p) = P(p) + R(p). \tag{8}$$

(Depths are clamped to $[0, d_{\max}]$ in implementation.) By design, P retains global layout and absolute scale, while R is encouraged to address only local biases near edges, thin structures, and texture-poor regions. The dense monocular prior E is injected via vision encoder and skip connections so that the decoder can allocate residual capacity preferentially at discontinuities, preventing over-smoothing.

C. Affinity-Based Refinement

a) Scope and inputs.: Given the initialization $D^0 \in \mathbb{R}^{H\times W}$ from Sec. III-B, our goal is to refine it using geometry-aware pixel affinities and a per-pixel sensor anchoring rule that respects LiDAR observations while remaining robust to misprojections. Let F denote encoder features at the image resolution, the raw sensor (LiDAR) depth D_s, and the corresponding observation mask $M_L \in \{0,1\}^{H\times W}$. We employ a set of kernel sizes $\mathcal{K} = \{3, 5, 7\}$ and perform T propagation iterations. A single curvature parameter $\kappa > 0$ (shared across $k \in \mathcal{K}$) defines the Poincaré ball used for hyperbolic distance evaluations.

b) Notation.: For a pixel p, $\mathcal{N}_k(p)$ is the $k \times k$ neighborhood; $q \in \mathcal{N}_k(p)$ indexes its neighbors. Kernel gates $\sigma_k(p) \in [0,1]$ satisfy $\sum_{k\in\mathcal{K}} \sigma_k(p) = 1$. All row-stochastic affinity weights $A_k(p,q)$ satisfy $\sum_{q\in\mathcal{N}_k(p)} A_k(p,q) = 1$.

c) Hyperbolic pairwise affinity.: We embed per-pixel features into the Poincaré ball [17], [18] via the exponential map at the origin:

$$h_p = \exp_0^\kappa(W_f F(p)). \tag{9}$$

where W_f is a 1×1 projection. Let $d_\kappa(\cdot,\cdot)$ denote the hyperbolic distance. For each kernel $k \in \mathcal{K}$ and neighbor $q \in \mathcal{N}_k(p)$ we define the unnormalized weights

$$\tilde{A}_k(p,q) = \exp\left(-\frac{d_\kappa(h_p, h_q)}{\tau_k}\right) \mathbf{1}\{q \in \mathcal{N}_k(p)\}. \tag{10}$$

$$A_k(p,q) = \frac{\tilde{A}_k(p,q)}{\sum_{q'\in\mathcal{N}_k(p)} \tilde{A}_k(p,q')}. \tag{11}$$

with temperature $\tau_k > 0$ (per kernel).

Intuitively, hyperbolic geometry is well suited for representing the hierarchical organization of scene features: pixels belonging to the same object instance tend to be closer in the Poincaré ball, while pixels across object boundaries become farther apart, thereby discouraging undesired depth smoothing across discontinuities. Accordingly, $A_k(p,q)$ acts as an edge-aware mixing weight during propagation, controlling how much the depth at pixel p is influenced by its neighbor q. When $A_k(p,q)$ is large, p and q are close in the learned hyperbolic feature space (and thus likely lie on the same surface), encouraging their depths to align; when it is small, propagation across boundaries is suppressed, preserving sharp depth transitions.

Then, pixel-wise kernel gates are obtained by a lightweight head:

$$\sigma_k(p) = \frac{\exp\big(g_k(F(p))\big)}{\sum_{k'\in\mathcal{K}} \exp\big(g_{k'}(F(p))\big)}, \quad k \in \mathcal{K}. \tag{12}$$

Conceptually, 3×3 emphasizes edges and thin structures, 5×5 aggregates mid-range context, and 7×7 stabilizes long-range consistency, while the gates adapt these roles per pixel.

TABLE I

KITTI DEPTH COMPLETION BENCHMARK. BEST IN **BOLD**, SECOND-BEST IS UNDERLINED. *Protocol:* 1/10/100-SHOT ARE SAMPLED FROM THE **TRAINING** SPLIT ONLY; EVALUATION IS ON THE **OFFICIAL 1,000-FRAME VALIDATION** SPLIT (NO TEST-SERVER SUBMISSIONS UNLESS STATED).

	1-shot		10-shot		100-shot		1-Sequence Training	
Method	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)
CSPN [13]	9.2748	3.5921	2.0222	0.7825	1.4510	0.5184	2.6289	0.8355
S2D [19]	8.8479	5.6022	5.0500	3.1469	4.2799	2.6633	4.7950	2.5610
NLSPN [12]	7.2899	4.7422	4.0070	2.2588	2.4979	1.1710	4.0290	1.7881
DySPN [14]	2.6350	0.8870	2.2701	0.9150	1.8777	0.6188	2.8530	0.7980
CompletionFormer [20]	4.7212	2.3789	3.1601	1.4740	2.6122	1.3299	4.5588	1.9603
BPNet [21]	5.4000	1.0740	1.8799	0.5559	1.3001	0.3910	2.1322	0.6420
DepthPrompting [6]	2.9840	1.1430	2.3988	1.1290	1.8249	0.6240	2.9468	0.9869
Ours	**1.4190**	**0.5073**	**1.2830**	**0.4001**	**1.2455**	**0.3548**	**1.5782**	**0.5540**

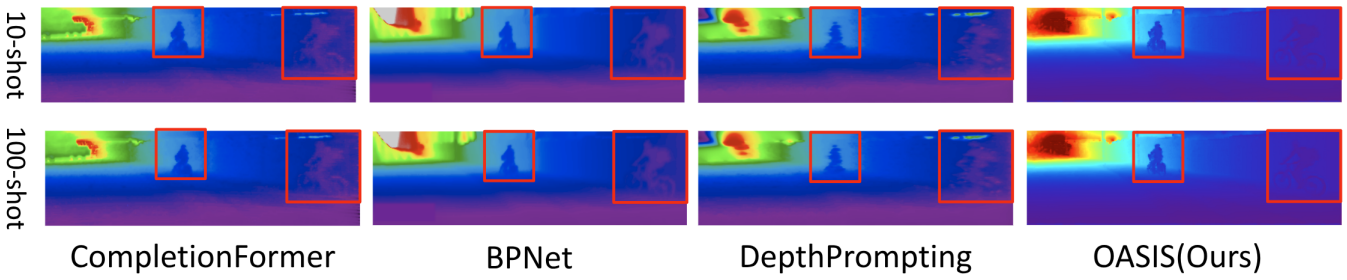

Fig. 3. **KITTI-DC qualitative comparison under few-shot supervision.** Top: 10-shot; bottom: 100-shot. **Columns:** RGB, CompletionFormer, BPNet, DepthPrompting, and OASIS-DC (Ours). Few-shot settings sample only from the training split; visual examples are evaluated against the official 1,000-frame validation set. Red boxes highlight challenging regions (thin structures, far-field). Our results preserve road–wall boundaries and fine details with reduced cross-edge bleeding.

d) Center-tethered multi-kernel propagation.: To prevent drift and excessive smoothing, each local update is re-centered on D^0. At iteration t, we form a center-tethered patch by replacing the center value with D^0:

$$\tilde{D}^{(t)}(q) = \begin{cases} D^0(p), & q = p, \\ D^{(t)}(q), & q \neq p. \end{cases} \tag{13}$$

Per-kernel propagation and gated mixing are then

$$D_k^{(t+1)}(p) = \sum_{q \in \mathcal{N}_k(p)} A_k(p,q)\, \tilde{D}^{(t)}(q), \tag{14}$$

$$D_{\text{mix}}^{(t+1)}(p) = \sum_{k \in \mathcal{K}} \sigma_k(p)\, D_k^{(t+1)}(p). \tag{15}$$

e) Learnable sensor anchoring.: We enforce per-pixel consistency with sensor observations only where available, while allowing soft corrections near occlusions and misprojections. An anchor map $\alpha(p) \in [0,1]$ is predicted from features:

$$\alpha(p) = \sigma\big(W_\alpha F(p)\big). \tag{16}$$

with $\sigma(\cdot)$ the logistic function. At observed pixels ($M_L(p) = 1$), the mixed depth in (15) is blended with the raw sensor depth D_s:

$$D^{(t+1)}(p) = \big(1 - \alpha(p)\, M_L(p)\big)\, D_{\text{mix}}^{(t+1)}(p) + \alpha(p)\, M_L(p)\, D_s(p). \tag{17}$$

and $D^{(t+1)}(p) = D_{\text{mix}}^{(t+1)}(p)$ when $M_L(p) = 0$. After T iterations, we output $D_{\text{final}} = D^{(T)}$ and clamp depths to $[0, d_{\max}]$ if necessary.

IV. EXPERIMENT

A. Datasets & Evaluation.

We evaluate on two complementary benchmarks that span outdoor driving and indoor scenes, and report results under their official protocols.

a) KITTI Depth Completion (KITTI-DC).: A large-scale outdoor driving dataset with synchronized RGB images and LiDAR measurements from a Velodyne HDL–64E. Following the official split, we use approximately 86K training samples, 7K validation samples, and 1K test samples. Images are provided at a resolution of 1216×352. The raw sparse LiDAR depth covers roughly 6% of image pixels, while the benchmark's reference depth is produced via *multi-sweep accumulation and cleanup*, resulting in about 20% density. *Evaluation.* Unless otherwise noted, we report quantitative results on the official validation split; submissions to the test server are used only when explicitly stated. Few-shot subsets (1-/10-/100-shot and 1-Sequence) are drawn from the training split, and validation/test labels are never used for training.

b) NYUv2.: A canonical indoor RGB–D dataset captured with a Microsoft Kinect sensor across diverse scenes such as offices, kitchens, and living spaces. We adopt the standard labeled subset at 640×480 resolution and follow

TABLE II

NYUv2 Depth Completion Benchmark. Best in **bold**, second-best is underlined.

Method	1-shot		10-shot		100-shot		1-Sequence Training	
	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)
CSPN [13]	1.4827	1.2058	0.3166	0.1961	0.2854	0.1307	0.3166	0.1961
NLSPN [12]	1.9358	1.6132	1.5995	0.8261	0.5501	0.4150	0.8881	0.6421
DySPN [14]	1.5474	1.2851	0.4102	0.2817	0.3079	0.1706	0.2584	0.1320
CompletionFormer [20]	1.8218	1.5539	1.1583	1.0162	0.9914	0.8164	0.6779	0.5356
CostDCNet [22]	1.2298	0.9754	0.2363	0.1288	0.1770	0.0836	0.2066	0.0954
BPNet [21]	0.3573	0.2077	0.2392	0.1120	0.1757	0.0793	0.2220	0.1040
DepthPrompting [6]	0.3583	0.2067	0.2195	0.1006	0.2101	0.1008	0.2335	0.1191
UniDC [7]	**0.2099**	**0.1075**	**0.1657**	**0.0794**	**0.1473**	**0.0669**	**0.1632**	**0.0745**
Ours	0.2105	0.1105	0.1670	0.0838	0.1484	0.0706	0.1644	0.0787

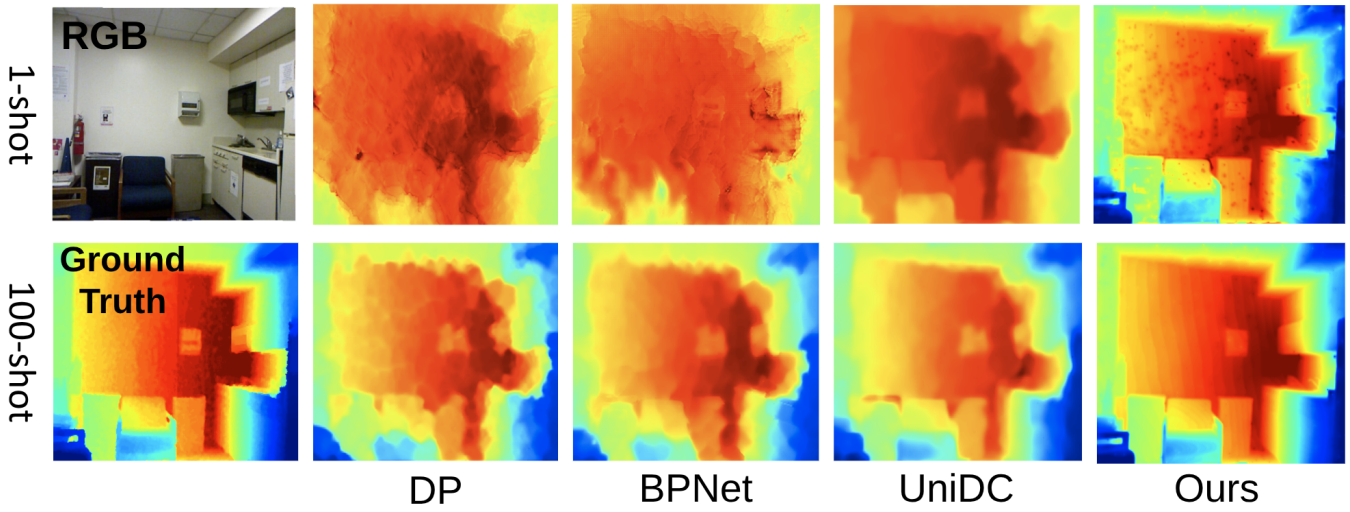

Fig. 4. **NYUv2 qualitative comparison.** Top row: 1-shot (left: RGB); bottom row: 100-shot (left: Ground Truth). Remaining columns show DP (DepthPrompting), BPNet, UniDC, and Ours. The proposed method produces cleaner planar surfaces and sharper discontinuities (e.g., cabinet edges), while suppressing noise and texture copying across shots.

the conventional official split for training and evaluation. For depth completion, sparse inputs are synthesized by subsampling dense depth maps to LiDAR-like sparsities; the same sparsity masks are shared across methods to ensure fairness. *Evaluation.* As in KITTI-DC, unless otherwise noted we report on the official validation split and do not access validation/test annotations during training; few-shot regimes mirror the KITTI settings.

c) Metrics.: We report RMSE (meters) and MAE (meters). Lower is better for both. All metrics are computed on valid ground-truth pixels using each dataset's official protocol, with invalid/unknown pixels excluded from the averages.

B. Implementation Details.

All experiments are executed on a single NVIDIA RTX A5000 (24 GB) GPU. We adopt a strict few-shot protocol on the official training splits of standard depth-completion benchmarks (e.g., KITTI-DC [1], [2] and NYUv2 [24]): from each dataset we construct *1/10/100-shot* subsets as well as a *1-sequence* subset consisting of one contiguous sequence. For each shot level and for the sequence setting, sampling is repeated with r independent random seeds; unless otherwise noted we report the mean (and, when space permits, the standard deviation) over these r trials. Training uses only the few-shot subsets, and validation/test annotations are never used for training. As the foundation monocular depth estimator, we use *Depth Anything v2* [**?**].

Optimization is conducted strictly in few-shot mode, with the number of iterations scaled to subset size: from ~ 100 iterations for 1-shot up to 3,000 iterations for larger few-shot subsets (e.g., 100-shot) and the 1-sequence case. Unless otherwise specified, all other hyperparameters are kept fixed across shots to ensure comparability.

Loss functions. We supervise with two masked objectives. Let D_{gt} denote ground-truth depth and

$$M = \mathbb{1}[D_{gt} > 0], \qquad n = \max\big(1, \sum M\big). \tag{18}$$

The composite L1+L2 loss is

$$e = (D_{\text{pred}} - D_{gt}) \odot M, \qquad \mathcal{L}_{\text{L1+L2}} = \frac{1}{n} \sum \big(|e| + e^2\big). \tag{19}$$

The scale-invariant log loss operates on positive predictions D_{rel}^{+} (we use a normalized input in $(0, 1]$ with ε-clamping):

$$d = \big(\log(D_{\text{rel}}^{+} + \varepsilon) - \log(D_{gt} + \varepsilon)\big) \odot M,$$
$$\mathcal{L}_{\text{SI-Log}} = \frac{1}{n}\sum d^2 \; - \; \Big(\frac{1}{n}\sum d\Big)^2. \tag{20}$$

If D_{gt} is unavailable for a sample, the corresponding term is set to zero. Unless stated otherwise, we sum the two losses.

C. Quantitative Result.

a) KITTI.: The improvements in Table I stem from how our model decomposes the problem into *metric alignment* and *local residual correction*. Sparse LiDAR anchors fix the global, low-frequency structure (absolute scale and large smooth surfaces), which a few labeled samples alone cannot reliably estimate in the 1/10/100-shot regimes. On top of this calibrated pseudo-depth, the residual branch concentrates its limited capacity on high-frequency errors around discontinuities. This design directly targets the *penetration* failure mode (depth leaking across object boundaries): the residual is learned to deviate from the prior specifically where gradients disagree with the measurements, which sharpens boundaries and suppresses cross-edge bleeding. The ablations corroborate this mechanism: using only residuals leaves metric drift unresolved; using only scale calibration reduces bias but cannot repair edge-local artifacts; combining both yields consistent gains across shots. That these gains persist from 1-shot to 1-Sequence indicates the model's inductive bias—not more supervision—is the primary driver: the prior constrains the hypothesis space, while the residual acts as a targeted, data-efficient corrector rather than a full predictor.

b) NYUv2.: As shown in Table II, our method is consistently second-best, narrowly trailing UniDC while outperforming the remaining baselines. This gap is explained by two factors: (i) *weaker anchoring* on NYUv2—Kinect GT and synthesized sparse masks provide less reliable constraints near discontinuities than real LiDAR on KITTI—limiting how strongly metric alignment can regularize the prior; and (ii) an *intentionally conservative residual capacity* for few-shot stability that can underfit fine indoor details relative to UniDC's larger heads. Even so, the qualitative comparisons in Fig. 4 show sharper edges and cleaner planes than other networks, consistent with the aggregated metrics in Table II. Notably, in Tables I and II, the 1-Sequence setting can slightly underperform 100-shot despite containing more frames. This is likely because a single contiguous sequence provides highly correlated samples with redundant viewpoints and limited scene diversity, whereas 100-shot sampling across sequences yields broader coverage of driving conditions and better matches the overall training distribution.

D. Ablation Study.

a) Imperfect Ground Truth: On NYUv2, our scale–calibrated prior with residual edge refinement yields *second-best* RMSE/MAE across shots (Table II), narrowly trailing UniDC while surpassing other baselines. Figure 5 explains the margin: Kinect GT often misses thin structures and over-smooths discontinuities; our residual head sharpens boundaries and recovers small objects, which can increase pixel error in boundary bands despite visibly cleaner geometry. In ablations (prior-only / residual-only / full), only the full model jointly stabilizes global scale and suppresses cross-edge penetration, producing the most reliable trade-off between quantitative scores and edge fidelity.

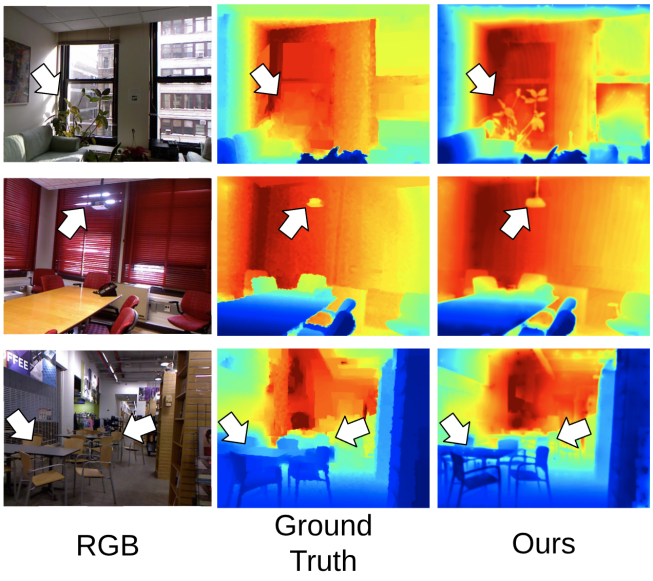

Fig. 5. **Imperfect NYUv2 ground truth.** Arrows mark GT artifacts (holes/smoothing); metrics use valid GT masks.

TABLE III
KITTI DEPTH COMPLETION BENCHMARK (ABLATION). RESIDUAL/SCALE FLAGS DENOTE MODULE USAGE: (✓,–) RESIDUAL HEAD ONLY; (–,✓) PSEUDO-PRIOR (SCALE) ONLY; (✓,✓) BOTH; (–,–) NEITHER.

KITTI Depth Completion Benchmark							
		1-shot		10-shot		100-shot	
Residual	Scale	RMSE	MAE	RMSE	MAE	RMSE	MAE
		2.8584	2.1554	2.2840	1.5562	2.2371	1.5109
✓		4.4062	2.7533	3.5208	1.9879	3.4485	1.9300
	✓	2.8939	2.1404	2.3140	1.5469	2.2666	1.5009
✓	✓	**1.4190**	**0.5073**	**1.2830**	**0.4001**	**1.2455**	**0.3548**

b) Ablation Study of Network Design.: Table III and Table V jointly indicate that our residual head operates in an extreme low-capacity regime (0.219M learnable parameters, 0.013s/inference; Table V). In this setting, *anchors* (scale calibration from sparse depth) are critical: without anchors, the residual-only variant underfits and amplifies monocular prior biases, yielding substantially worse errors than even the *neither* configuration across 1/10/100-shot. By contrast, when both modules are disabled, the pseudo-depth prior

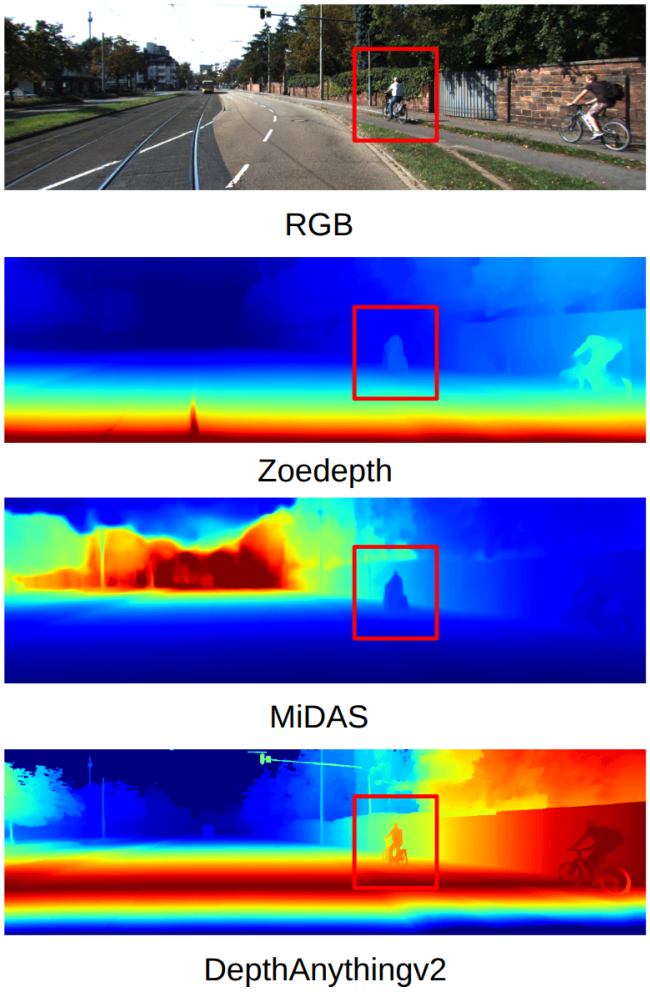

Fig. 6. **DepthAnythingv2 excels at boundary preservation.** Qualitative comparison on a KITTI scene. Within the red boxes, DepthAnythingv2 preserves fine object boundaries and thin structures more reliably than ZoeDepth and MiDaS, yielding sharper silhouettes and fewer color bleeding artifacts.

alone imposes a strong inductive bias that "floors" performance (Table III, first row), preventing catastrophic drift despite the tiny trainable head. The best results arise when anchors and residual are combined (last row): anchors stabilize global metric scale and low-frequency structure, while the residual allocates its limited capacity to high-frequency edge corrections (penetration suppression, boundary sharpening). The pattern is consistent across shot regimes, underscoring that with a compact learnable core (Table V) the anchor signal is not merely helpful but *necessary* for reliable depth completion; the residual then provides targeted refinement rather than bearing the burden of learning scale from few examples.

c) Ablation on pseudo-depth priors.: Table IV evaluates our *training-free* pseudo-depth generation pipeline by swapping only the frozen foundation MDE prior (Depth Anything v2 [**?**], MiDaS [23], ZoeDepth [8]) and applying the same test-time densification, with *no learned refinement*. Despite requiring zero training, the method delivers competitive absolute errors across both benchmarks, indicating that the pseudo-depth pipeline *alone* is sufficiently strong for practical deployment when labeled data or training budgets are constrained. Importantly, because the pseudo-depth pipeline is training-free, it is less tied to a specific dataset and is more likely to transfer Following this motivation, a direct comparison against SOTA monocular metric depth methods designed for broad generalization (e.g., UniDepth) on *completely unseen* datasets would be a valuable addition, and we will include such cross-dataset evaluations in the revised version.

TABLE IV

ABLATION ON PSEUDO-DEPTH PRIORS ON KITTI AND NYUV2. PSEUDO-DEPTH MAPS ARE OBTAINED BY TEST-TIME DENSIFICATION FROM FROZEN FOUNDATION MDES WITHOUT ANY LEARNED REFINEMENT.

	KITTI		NYUv2	
Method	RMSE	MAE	RMSE	MAE
Ours_DepthAnythingv2	1.7481	0.4525	**0.2115**	**0.1135**
Ours_MiDAS	1.7117	0.4360	0.2741	0.1417
Ours_Zoe	**1.6264**	**0.4135**	0.3032	0.1223

TABLE V

COMPUTATIONAL COST OF MODELS.

Model	Total Param.	Learnable Param.	Infer. Time (s)	GPU Memory (MiB)
BPNet [21]	89.874M	89.874M	0.072	4792
LRRU [29]	**20.843M**	20.843M	0.038	3650
CompletionFormer [20]	83.574M	83.574M	0.060	4206
Ours	94.550M	**0.219M**	**0.013**	**1904**

On KITTI [1], [2], the Zoe prior attains the best numbers (RMSE/MAE = 1.6264/0.4135), while the Depth Anything v2 prior reports 1.7481/0.4525—about ∼7.5% higher RMSE and ∼9.4% higher MAE relative to Zoe. We caution that KITTI provides *sparse* LiDAR supervision, and point-sampled metrics tend to favor smoother predictions or those coinciding with LiDAR sampling; in our qualitative inspections, Depth Anything v2 preserves object boundaries and thin structures more crisply, which is not fully reflected by the sparse metrics. Consistently, on the *dense* indoor NYUv2 benchmark [24], Depth Anything v2 achieves the best errors (0.2115/0.1135), improving over MiDaS by ≈22.8%/19.9% and over ZoeDepth by ≈30.2%/7.2% in RMSE/MAE, respectively. These results underscore that our training-free pseudo-depth maps are already robust enough for industrial use, and that dense ground-truth evaluation better captures the edge fidelity exhibited by the Depth Anything v2 prior.

V. CONCLUSIONS

We presented a prior-guided few-shot depth completion framework that fuses a foundation MDE with sparse anchors to form a calibrated pseudo-depth prior, training a

refinement network to respect this prior while correcting local mismatches. The framework is *foundation-agnostic*, allowing seamless adaptation to stronger backbones (e.g., UniDepth [39]) without altering the refinement architecture. This design reduces the hypothesis space, stabilizing learning from few samples while preserving metric scale and edge detail. Our KITTI evaluation adheres to standard few-shot and strict deployment-oriented protocols (averaging over five seeds and reporting on the unseen 1,000-image validation split) [1], [2]. Ablations show that removing the prior harms generalization, whereas removing sparse anchors increases scale drift. Future work includes uncertainty-aware calibration, dynamic-scene handling, and cross-dataset generalization.

APPENDIX

To ensure reproducibility, we specify the fixed sequences used for the 1-Sequence protocol.

NYUv2: Five indoor scenes: `conference_room_0001`, `bedroom_0015`, `dining_room_0004`, `kitchen_0008`, and `classroom_0004`.

KITTI-DC: Five raw drives: `2011_09_26_0014`, `2011_09_28_0035`, `2011_09_28_0038`, `2011_09_30_0020`, and `2011_10_03_0034`.

Train/val splits (70/30) are created using five random seeds, with mean results reported.

REFERENCES

[1] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity Invariant CNNs," in *Proc. 3DV*, 2017.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proc. CVPR*, 2012.

[3] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning Guided Convolutional Network for Depth Completion," *IEEE TIP*, vol. 30, pp. 1116–1129, 2021.

[4] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, "DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene from Sparse LiDAR Data and a Single Color Image," in *Proc. CVPR*, 2019.

[5] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "PENet: Towards Precise and Efficient Image Guided Depth Completion," in *Proc. ICRA*, 2021.

[6] J.-H. Park, C. Jeong, J. Lee, and H.-G. Jeon, "Depth Prompting for Sensor-Agnostic Depth Estimation," in *Proc. CVPR*, 2024.

[7] J.-H. Park and H.-G. Jeon, "A Simple yet Universal Framework for Depth Completion," in *Proc. NeurIPS*, 2024.

[8] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth," arXiv:2302.12288, 2023.

[9] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data," in *Proc. CVPR*, 2024.

[10] S. Li, B. Zhang, X. Yang, *et al.*, "Edge-guided second-order total generalized variation for Gaussian noise removal from depth map," *Sci. Rep.*, vol. 10, p. 16329, 2020.

[11] F. Yanez, A. Fan, B. Bilgic, C. Milovic, E. Adalsteinsson, and P. Irarrazaval, "Quantitative Susceptibility Map Reconstruction via a Total Generalized Variation Regularization," in *Proc. PRNI*, 2013, pp. 203–206.

[12] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-Local Spatial Propagation Network for Depth Completion," in *Proc. ECCV*, 2020.

[13] X. Cheng, P. Wang, and R. Yang, "Depth Estimation via Affinity Learned with Convolutional Spatial Propagation Network," in *Proc. ECCV*, 2018.

[14] Y. Lin, T. Cheng, Q. Zhong, W. Zhou, and H. Yang, "Dynamic Spatial Propagation Network for Depth Completion," in *Proc. AAAI*, 2022.

[15] P. Pérez, M. Gangnet, and A. Blake, "Poisson Image Editing," in *Proc. SIGGRAPH*, 2003.

[16] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[17] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic Neural Networks," in *Proc. NeurIPS*, 2018.

[18] M. Nickel and D. Kiela, "Poincaré Embeddings for Learning Hierarchical Representations," in *Proc. NIPS*, 2017.

[19] F. Ma and S. Karaman, "Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image," in *Proc. ICRA*, 2018.

[20] Y. Zhang, X. Guo, M. Poggi, Z. Zhu, G. Huang, and S. Mattoccia, "CompletionFormer: Depth Completion with Convolutions and Vision Transformers," in *Proc. CVPR*, 2023.

[21] J. Tang, F.-P. Tian, B. An, J. Li, and P. Tan, "Bilateral Propagation Network for Depth Completion," in *Proc. CVPR*, 2024.

[22] J. Kam, J. Kim, S. Kim, J. Park, and S. Lee, "CostDCNet: Cost Volume Based Depth Completion for a Single RGB–D Image," in *Proc. ECCV*, 2022.

[23] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer," *IEEE TPAMI*, vol. 44, no. 3, 2022.

[24] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor Segmentation and Support Inference from RGBD Images," in *Proc. ECCV*, 2012.

[25] A. Wong and S. Soatto, "Unsupervised Depth Completion with Calibrated Backprojection Layers," in *Proc. ICCV*, 2021.

[26] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, "Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation," in *Proc. CVPR*, 2024.

[27] M. Gui *et al.*, "DepthFM: Fast Monocular Depth Estimation with Flow Matching," arXiv:2403.13788, 2024.

[28] H. Lee, K. S. Kim, B.-K. Kwon, and T.-H. Oh, "Zero-shot Depth Completion via Test-time Alignment with Affine-invariant Depth Prior," arXiv:2502.06338, 2025.

[29] Y. Wang, B. Li, G. Zhang, Q. Liu, T. Gao, and Y. Dai, "LRRU: Long-Short Range Recurrent Updating Networks for Depth Completion," in *Proc. ICCV*, 2023.

[30] A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein, "Universal Guidance for Diffusion Models," in *Proc. ICLR*, 2024.

[31] J. Song, C. Meng, and S. Ermon, "Denoising Diffusion Implicit Models," in *Proc. ICLR*, 2021.

[32] H. Chung, J. Kim, M. T. McCann, M. L. Klasky, and J. C. Ye, "Diffusion Posterior Sampling for General Noisy Inverse Problems," in *Proc. ICLR*, 2023.

[33] B. Song, S. M. Kwon, Z. Zhang, X. Hu, Q. Qu, and L. Shen, "Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency," in *Proc. ICLR*, 2024.

[34] L. Bartolomei, M. Poggi, A. Conti, F. Tosi, and S. Mattoccia, "Revisiting Depth Completion from a Stereo Matching Perspective for Cross-Domain Generalization," in *Proc. 3DV*, 2024.

[35] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, 2003.

[36] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network," in *Proc. NeurIPS*, 2014.

[37] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed., Pearson, 2018.

[38] P. Soille, *Morphological Image Analysis: Principles and Applications*, 2nd ed., Springer, 2003.

[39] L. Piccinelli, et al., "UniDepth: Universal Monocular Metric Depth Estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.