UniDepth: Universal Monocular Metric Depth Estimation

Luigi Piccinelli[1] Yung-Hsu Yang[1] Christos Sakaridis[1]
Mattia Segu[1] Siyuan Li[1] Luc Van Gool[1,2] Fisher Yu[1]

[1]ETH Zürich [2]INSAIT

Abstract

Accurate monocular metric depth estimation (MMDE) is crucial to solving downstream tasks in 3D perception and modeling. However, the remarkable accuracy of recent MMDE methods is confined to their training domains. These methods fail to generalize to unseen domains even in the presence of moderate domain gaps, which hinders their practical applicability. We propose a new model, UniDepth, capable of reconstructing metric 3D scenes from solely single images across domains. Departing from the existing MMDE methods, UniDepth directly predicts metric 3D points from the input image at inference time without any additional information, striving for a universal and flexible MMDE solution. In particular, UniDepth implements a self-promptable camera module predicting dense camera representation to condition depth features. Our model exploits a pseudo-spherical output representation, which disentangles camera and depth representations. In addition, we propose a geometric invariance loss that promotes the invariance of camera-prompted depth features. Thorough evaluations on ten datasets in a zero-shot regime consistently demonstrate the superior performance of UniDepth, even when compared with methods directly trained on the testing domains. Code and models are available at: github.com/lpiccinelli-eth/unidepth.

1. Introduction

The precise pixel-wise depth estimation is crucial to understanding the geometric scene structure, with applications in 3D modeling [10], robotics [11, 63], and autonomous vehicles [38, 51]. However, delivering reliable metric scaled depth outputs is necessary to perform 3D reconstruction effectively, thus motivating the challenging and inherently ill-posed task of Monocular Metric Depth Estimation (MMDE).

While existing MMDE methods [3, 14, 16, 40, 41, 43, 61] have demonstrated remarkable accuracy across different benchmarks, they require training and testing on datasets with similar camera intrinsics and scene scales. Moreover, the training datasets typically have a limited size and contain little diversity in scenes and cameras. These characteristics result in poor generalization to real-world inference scenarios [52], where images are captured in uncontrolled, arbitrarily structured environments and cameras with arbitrary intrinsics.

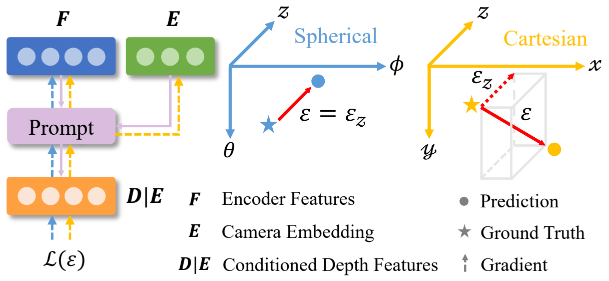

Figure 1. We introduce UniDepth, a novel approach that directly predicts 3D points in a scene with only one image as input. UniDepth incorporates a camera self-prompting mechanism and leverages a pseudo-spherical 3D output space defined by azimuth and elevation angles, and depth (θ, ϕ, z). This design effectively separates camera and depth optimization by avoiding gradient flowing to the camera module due to depth-related error (ε_z).

Only a few methods [21, 59] have addressed the challenging task of generalizable MMDE. However, these methods assume controlled setups at test time, including camera intrinsics. While this assumption simplifies the task, it has two notable drawbacks. Firstly, it does not address the full application spectrum, *e.g.* in-the-wild video processing and crowd-sourced image analysis. Secondly, the inherent camera parameter noise is directly injected into the model, leading to large inaccuracies in the high-noise case.

In this work, we address the more demanding task of generalizable MMDE *without* any reliance on additional external information, such as camera parameters, thus defining the universal MMDE task. Our approach, named UniDepth, is the first that attempts to solve this challenging task without restrictions on scene composition and setup and distinguishes itself through its general and adaptable nature. Unlike existing methods, UniDepth delivers metric 3D predictions for any scene *solely* from a single image, waiving the need for extra information about scene or camera. Furthermore, UniDepth flexibly allows for the incorporation of additional camera information at test time.

Our design introduces a camera module that outputs a non-parametric, *i.e.* dense camera representation, serving as the prompt to the depth module. However, relying only on this single additional module clearly results in challenges related to training stability and scale ambiguity. We propose an effective pseudo-spherical representation of the output space to disentangle the camera and depth dimensions of this space. This representation employs azimuth and elevation angle components for the camera and a radial component for the depth, forming a perfect orthogonal space between the camera plane and the depth axis. Moreover, the camera components are embedded through Laplace spherical harmonic encoding. Figure 1 depicts our camera self-prompting mechanism and the output space. Additionally, we introduce a geometric invariance loss to enhance the robustness of depth estimation. The underlying idea is that the camera-conditioned depth features from two views of the same image should exhibit reciprocal consistency. In particular, we sample two geometric augmentations, creating a pair of different views for each training image, thus simulating different apparent cameras for the original scene.

Our overall contribution is the first universal MMDE method, UniDepth, that predicts a point in metric 3D space for each pixel without *any* input other than a single image. In particular, first, we design a promptable camera module, an architectural component that learns a dense camera representation and allows for non-parametric camera conditioning. Second, we propose a pseudo-spherical representation of the output space, thus solving the intertwined nature of camera and depth prediction. In addition, we introduce a geometric invariance loss to disentangle the camera information from the underlying 3D geometry of the scene. Moreover, we extensively test UniDepth and re-evaluate seven MMDE State-of-the-Art (SotA) methods on ten different datasets in a fair and comparable zero-shot setup to lay the ground for the generalized MMDE task. Owing to its design, UniDepth consistently sets the new state of the art even compared with non-zero-shot methods, ranking first in the competitive official KITTI Depth Prediction Benchmark.

2. Related Work

Metric and Scale-agnostic Depth Estimation. It is crucial to distinguish Monocular Metric Depth Estimation (MMDE) from scale-agnostic, namely up-to-a-scale, monocular depth estimation. MMDE SotA approaches typically confine training and testing to the same domain. However, challenges arise, such as overfitting to the training scenario leading to considerable performance drops in the presence of minor domain gaps, often overlooked in benchmarks like NYU-Depthv2 [35] (NYU) and KITTI [18]. On the other hand, scale-agnostic depth methods, including MiDaS [42], OmniData [13], and LeReS [58], show robust generalization by training on extensive datasets. Their limitation lies in the absence of a metric output, hindering practical usage in downstream applications.

Monocular Metric Depth Estimation. The introduction of end-to-end trainable neural networks in MMDE, pioneered by [14], marked a significant milestone, also introducing the optimization process through the Scale-Invariant log loss ($\mathrm{SI}_{\log}$). Subsequent developments witnessed the emergence of advanced networks, ranging from convolution-based architectures [16, 27, 31, 40] to transformer-based approaches [3, 41, 57, 61]. Despite impressive achievements on established benchmarks, MMDE models face challenges in zero-shot scenarios, revealing the need for robust generalization against domain shifts in appearance and geometry.

General Monocular Metric Depth Estimation. Recent efforts focus on developing MMDE models [4, 21, 59] for general depth prediction across diverse domains. These models often leverage camera awareness, either by directly incorporating external camera parameters into computations [15, 21] or by normalizing the shape or output depth based on intrinsic properties, as seen in [1, 28, 59].

However, these generalizable MMDE methods often adopt specific strategies to enhance performance, *e.g.* geometric pretraining [4] or dataset-specific prior like reshaping [59]. In addition, these methods assume access to noiseless camera intrinsics both at training and test time, also limiting their applicability to pinhole camera models. Additionally, SotA methods depend on a predefined backprojection operation, blurring the distinction between learning depth and the 3D scene. In contrast, our approach aims to overcome these limitations, presenting a more demanding perspective, *e.g.* universal MMDE. Universal MMDE involves directly predicting the 3D scene from the input image *without any* additional information other than the latter. Notably, we do not require any additional prior information at test time, such as access to camera information.

3. UniDepth

MMDE SotA methods typically assume access to the camera intrinsics, thus blurring the line between pure depth estimation and actual 3D estimation. In contrast, UniDepth aims to create a universal MMDE model deployable in diverse scenarios without relying on any other external information, such as camera intrinsic, thus leading to 3D space estimation by design. However, attempting to directly predict 3D points from a single image without a proper internal representation neglects geometric prior knowledge, *i.e.* perspective geometry, burdening the learning process with re-learning laws of perspective projection from data.

Sec. 3.1 introduces a pseudo-spherical representation of the output space to inherently disentangle camera rays' angles from depth. In addition, our preliminary studies indi-

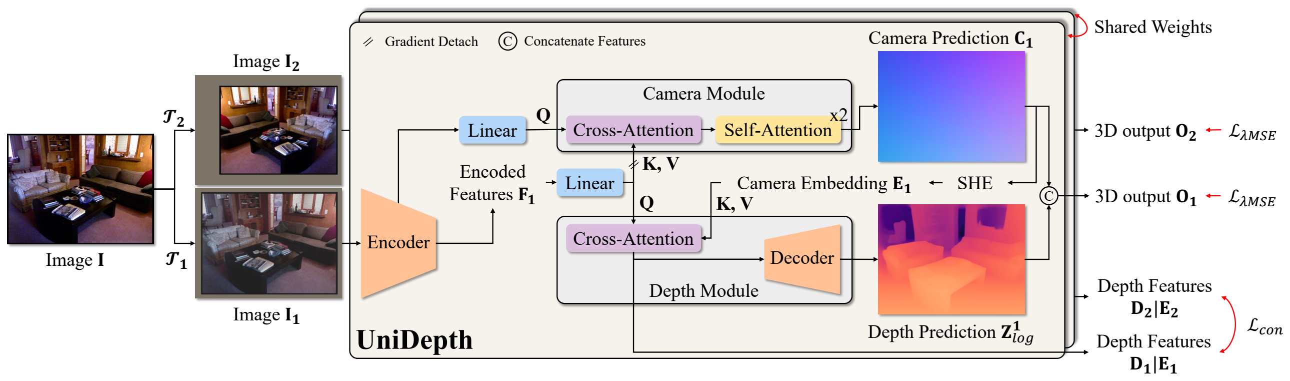
Gradient Detach
Concatenate Features
Camera Prediction $\mathbf{C_1}$
Shared Weights
Image $\mathbf{I_2}$
$\mathcal{T}_2$
Camera Module
Linear
Q
Cross-Attention
Self-Attention
x2
3D output $\mathbf{O_2}$ ← $\mathcal{L}_{\lambda MSE}$
Encoded Features $\mathbf{F_1}$
Linear
K, V
Camera Embedding $\mathbf{E_1}$ ← SHE
Q
K, V
3D output $\mathbf{O_1}$ ← $\mathcal{L}_{\lambda MSE}$
Encoder
Cross-Attention
Decoder
Image $\mathbf{I}$
$\mathcal{T}_1$
Image $\mathbf{I_1}$
Depth Module
Depth Features $\mathbf{D_2|E_2}$
$\mathcal{L}_{con}$
Depth Features $\mathbf{D_1|E_1}$
UniDepth
Depth Prediction $\mathbf{Z^1_{log}}$

Figure 2. **Model Architecture.** UniDepth utilizes solely the input image to generate the 3D output ($\mathbf{O}$). It bootstraps dense camera prediction ($\mathbf{C}$) from the Camera Module, injecting prior knowledge on scene scale into the Depth Module via a cross-attention layer. The camera representation corresponds to azimuth and elevation angles. The geometric invariance loss ($\mathcal{L}_{\text{con}}$) enforces consistency between depth features tensors conditioned on the camera from different geometric augmentations ($\mathcal{T}_1$, $\mathcal{T}_2$). Stop-gradient is applied to the encoded feature ($\mathbf{F}$) flowing to the Camera Module to prevent the camera gradient from dominating the depth gradient in the encoder. The depth output ($\mathbf{Z}_{\text{log}}$) is obtained through three self-attention blocks interleaved with learnable 2x upsampling. The final output is the concatenation of the camera and depth tensors ($\mathbf{C}||\mathbf{Z}_{\text{log}}$), creating two independent optimization spaces for $\mathcal{L}_{\lambda MSE}$.

cate that depth prediction clearly benefits from prior information on the acquisition sensor, leading to the introduction of a self-prompting camera operation in Sec. 3.2. Further disentanglement at the level of internal depth features is achieved through a geometric invariance loss, outlined in Sec. 3.3. This loss ensures depth features remain invariant when conditioned on the bootstrapped camera predictions, promoting robust camera-aware depth predictions. The overall architecture and the resulting optimization induced by the combination of design choices are detailed in Sec. 3.4.

3.1. 3D Representation

The general purpose nature of our MMDE method requires inferring both depth and camera intrinsics to make 3D predictions based only on imagery observations. We design the 3D output space presenting a natural disentanglement of the two sub-tasks, namely depth estimation and camera calibration. In particular, we exploit the pseudo-spherical representation where the basis is defined by azimuth, elevation, and log-depth, *i.e*. (θ,ϕ,z_{log}), in contrast to the Cartesian representation (x,y,z). The strength of the proposed pseudo-spherical representation lies in the decoupling of camera (θ,ϕ) and depth (z_{log}) components, ensuring their orthogonality by design, in contrast to the entanglement present in Cartesian representation.

It is worth highlighting that in this output space, the non-parametric dense representation of the camera is mathematically represented as a tensor $\mathbf{C} \in \mathbb{R}^{H\times W\times 2}$, where H and W are the height and width of the input image and the last dimension corresponds to azimuth and elevation values. While in the typical Cartesian space, the backprojection involves the multiplication of homogeneous camera rays and depth, the backprojection operation in the proposed representation space accounts for the concatenation of camera and depth representations. The pencil of rays are defined as $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) = \mathbf{K}^{-1}[\mathbf{u}, \mathbf{v}, \mathbf{1}]^T$, where $\mathbf{K}$ is the calibration matrix, $\mathbf{u}$ and $\mathbf{v}$ are pixel positions in pixel coordinates, and $\mathbf{1}$ is a vector of ones. Therefore, the homogeneous camera rays $(\mathbf{r}_x, \mathbf{r}_y)$ correspond to $(\frac{\mathbf{r}_1}{\mathbf{r}_3}, \frac{\mathbf{r}_2}{\mathbf{r}_3})$.

Moreover, the angular dense representation can be embedded via the Laplace Spherical Harmonic Encoding (SHE). The camera embedding tensor is defined as $\mathbf{E} = \text{SHE}(\mathbf{C}), \mathbf{E} \in \mathbb{R}^{H\times W\times d}$, where d is the number of harmonics chosen. $\text{SHE}(\cdot)$ computes the set of spherical harmonics, *i.e*., $\{\mathcal{Y}\}_{l,m}$ with degree l and order m, and concatenating along the channel dimension, with $\mathbf{Y}^l_m$ as

$$Y^l_m(\theta, \phi) = \alpha^l_m \mathcal{P}^l_m(\cos\theta)e^{im\phi}, \tag{1}$$

where $\mathcal{P}^l_m$ is the associated Legendre polynomial of degree l and order m, and α^l_m is a normalizing constant. In particular, the spherical harmonics on the unit sphere form an orthogonal basis of the spherical manifold and preserve inner products. The total number of harmonics utilized is 81, resulting from capping the degree l to 8. SHE is utilized as a mathematic sounder choice compared to, *e.g*. the Fourier Transform, to produce the camera embeddings.

3.2. Self-Promptable Camera

The camera module plays a crucial role in the final 3D predictions since its angular dense output accounts for two dimensions of the output space, namely azimuth and elevation. Most importantly, these embeddings prompt the depth module to ensure a bootstrapped prior knowledge of the input scene's global depth scale. The prompting is fundamental to avoid mode collapse in the scene scale and to alleviate the depth module from the burden of predicting depth from scratch as the scale is already modeled by camera output.

Nonetheless, the internal representation of the camera module is based on a pinhole parameterization, namely via focal length (f_x, f_y) and principal point (c_x, c_y). The four tokens conceptually corresponding to the intrinsics are then projected to scalar values, *i.e.*, Δf_x, Δf_y, Δc_x, Δc_y. However, they do not directly represent the camera parameters, but the multiplicative residuals to a pinhole camera initialization, namely $\frac{H}{2}$ for y-components and $\frac{W}{2}$ for x-components, leading to $f_x = \frac{\Delta f_x W}{2}$, $f_y = \frac{\Delta f_y H}{2}$, $c_x = \frac{\Delta c_x W}{2}$, $c_y = \frac{\Delta c_y H}{2}$, leading to invariance towards input image sizes.

Subsequently, a backprojection operation based on the intrinsic parameters is applied to every pixel coordinate to produce the corresponding rays. The rays are normalized and thus represent vectors on a unit sphere. The critical step involves extracting azimuth and elevation from the backprojected rays, effectively creating a "dense" angular camera representation. This dense representation undergoes SHE to produce the embeddings $\mathbf{E}$. The embedded representations are then seamlessly passed to the depth module as a prompt, where they play a vital role as a conditioning factor. The conditioning is enforced via a cross-attention layer between the initialized feature of Depth Module $\mathbf{D} \in \mathbb{R}^{h \times w \times C}$ and the camera embeddings $\mathbf{E}$ where $(h, w) = (H/16, W/16)$. The camera-prompted depth features $\mathbf{D}|\mathbf{E} \in \mathbb{R}^{h \times w \times C}$ are defined as

$$\mathbf{D}|\mathbf{E} = \mathrm{MLP}(\mathrm{CA}(\mathbf{D}, \mathbf{E})), \tag{2}$$

where CA is a cross-attention block and MLP is a Multi-Layer Perceptron with one $4C$-channel hidden layer.

Figure 3 illustrates one of the main benefits of our camera module. In particular, in high-noise intrinsics or camera-agnostic scenarios, UniDepth can bootstrap the camera prediction, thus displaying total noise insensitivity. However, we can substitute the camera module output to improve 3D reconstruction peak performance if any external dense camera representation is provided. This adaptability enhances the model's versatility, allowing it to operate seamlessly in diverse setups. Moreover, Figure 3 suggests that training with noisy self-prompts enhances the robustness of UniDepth to noisier external intrinsics if given at test time.

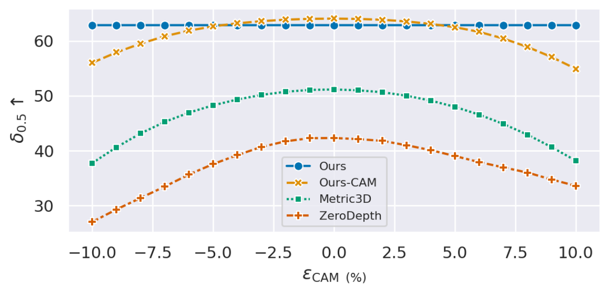

Figure 3. **Impact of noise in camera intrinsics.** The amount of relative distortion ($\varepsilon_{\mathrm{CAM}(\%)}$) of the intrinsics is shown on the x-axis, while $\delta_{0.5}$ performance on OOD test sets on the y-axis. Relying on external input inherently leads to being subject to its noise. UniDepth functions in dual regimes, with and without external intrinsic. In situations of unknown intrinsics or high noise, UniDepth exhibits total robustness by bootstrapping camera prediction (*Ours*). In contrast, with low-noise intrinsics, we leverage it for enhanced peak performance (*Ours-CAM*).

3.3. Geometric Invariance Loss

The spatial locations from the same scene captured by different cameras should correspond when the depth module is conditioned on the specific camera. To this end, we propose a geometric invariance loss to enforce the consistency of camera-prompted depth features of the same scene from different acquisition sensors. In particular, consistency is enforced on features extracted from identical 3D locations.

For each image, we perform N distinct geometrical augmentations, denoted as $\{\mathcal{T}_i\}_{i=1}^{N}$, with $N = 2$ in our experiments. This operation involves involves sampling a rescaling factor $r \sim 2^{\mathcal{U}_{[-1,1]}}$ and a relative translation on the x-axis $t \sim \mathcal{U}_{[-0.1,0.1]}$, then cropping it to the network's input shape. This is analogous to sampling a pair of images from the same scene and extrinsic parameters but captured by different cameras. Let $\mathbf{C}_i$ and $\mathbf{D_i}|\mathbf{E_i}$ describe the predicted camera representation and camera-prompted depth features, respectively, corresponding to augmentation $\mathcal{T}_i$. It is evident that the camera representations differ when two diverse geometric augmentations are applied, i.e., $\mathbf{C}_i \neq \mathbf{C}_j$ if $\mathcal{T}_i \neq \mathcal{T}_j$. Therefore, the geometric invariance loss can be expressed as

$$\begin{aligned}&\mathcal{L}_{\text{con}}(\mathbf{D}_1|\mathbf{E}_1, \mathbf{D}_2|\mathbf{E}_2) = \\ &\left\| \mathcal{T}_2 \circ \mathcal{T}_1^{-1} \circ (\mathbf{D}_1|\mathbf{E}_1) - \mathrm{sg}(\mathbf{D}_2|\mathbf{E}_2) \right\|_1,\end{aligned} \tag{3}$$

where $\mathbf{D}_i|\mathbf{E}_i$ represents the depth feature after being conditioned by camera prompt $\mathbf{E}_i$, as outlined in Sec. 3.2, and $\mathrm{sg}(\cdot)$ corresponds to the stop-gradient detach operation needed to exploit $\mathbf{D}_2|\mathbf{E}_2$ as pseudo ground-truth (GT). The bi-directional loss can be computed as: $\frac{1}{2}(\mathcal{L}_{\text{con}}(\mathbf{D}_1|\mathbf{E}_1, \mathbf{D}_2|\mathbf{E}_2) + \mathcal{L}_{\text{con}}(\mathbf{D}_2|\mathbf{E}_2, \mathbf{D}_1|\mathbf{E}_1))$. It is necessary to apply the geometric invariance loss *after* the features are conditioned on the viewing information, *i.e.*, camera. Otherwise, the loss would enforce consistency across features that inherently carry distinct camera information.

3.4. Network Design

Architecture. Our network, described in Fig. 2, comprises an Encoder Backbone, a Camera Module, and a Depth Module. The encoder can be either convolutional or ViT-based [12], producing features at different "scales", *i.e.* $\mathbf{F} \in \mathbb{R}^{h \times w \times C \times B}$, where $(h, w) = (\frac{H}{16}, \frac{W}{16})$ and $B = 4$.

The Camera Module parameters are initialized class tokens for ViT-style or pooled feature maps for convolutional-style backbones. The encoded features from the Encoder Backbone are passed to the Camera Module as a stack of detached tokens, the encoder class tokens are utilized as camera parameters initialization. The features are processed to obtain the final dense representation $\mathbf{C}$ as detailed in Sec. 3.2, and further embedded to $\mathbf{E}$ via $\mathrm{SHE}(\cdot)$ outlined in Sec. 3.1. Note that the stop-gradient operation is necessary because of the low variety of effective cameras compared to the image diversity. In fact, the Camera Module component easily overfits and clearly dominates the overall backbone gradient.

The Depth Module is fed with the encoder features to condition the initial latent features $\mathbf{L} \in \mathbb{R}^{h \times w \times C}$ via one cross-attention layer to obtain the initial depth features, $\mathbf{D}$. The latent feature tensor $\mathbf{L}$ is obtained as the average of the features $\mathbf{F}$ along the B dimension. Furthermore, the depth features are conditioned on the camera prompts $\mathbf{E}$ to obtain $\mathbf{D}|\mathbf{E}$ as described in Sec. 3.2. The camera-prompted depth features are further processed via self-attention layers where the positional encoding utilized is $\mathbf{E}$ and upsampled to produce a multi-scale output. The log-depth prediction $\mathbf{Z}_{\log} \in \mathbb{R}^{H \times W \times 1}$ corresponds to the mean of the interpolated intermediate representations. The final 3D output $\mathbf{O} \in \mathbb{R}^{H \times W \times 3}$ is the concatenation of predicted rays and depth, $\mathbf{O} = \mathbf{C}||\mathbf{Z}$, with $\mathbf{Z}$ as element-wise exponentiation of $\mathbf{Z}_{\log}$.

Optimization. The optimization process is guided by a reformulation of the Mean Squared Error (MSE) loss in the final 3D output space (θ,ϕ,$z_{\log}$) from Sec. 3.1 as:

$$\mathcal{L}_{\lambda\text{MSE}}(\boldsymbol{\varepsilon}) = \|\mathbb{V}[\boldsymbol{\varepsilon}]\|_1 + \boldsymbol{\lambda}^T(\mathbb{E}[\boldsymbol{\varepsilon}] \odot \mathbb{E}[\boldsymbol{\varepsilon}]), \tag{4}$$

where $\boldsymbol{\varepsilon} = \hat{\mathbf{o}} - \mathbf{o}^* \in \mathbb{R}^3$, $\hat{\mathbf{o}} = (\hat{\theta}, \hat{\phi}, \hat{z}_{\log})$ is the predicted 3D output, $\mathbf{o}^* = (\theta^*, \phi^*, z^*_{\log})$ is the GT 3D value, and $\boldsymbol{\lambda} = (\lambda_\theta, \lambda_\phi, \lambda_z) \in \mathbb{R}^3$ is a vector of weights for each dimension of the output. $\mathbb{V}[\boldsymbol{\varepsilon}]$ and $\mathbb{E}[\boldsymbol{\varepsilon}]$ are computed as the vectors of empirical variances and means for each of the three output dimensions over all pixels, *i.e.* $\{\boldsymbol{\varepsilon}^{(i)}\}_{i=1}^N$. Note that if $\lambda_d = 1$ for a dimension d, the loss represents the standard MSE loss for that dimension. If $\lambda_d < 1$, a scale-invariant loss term is added to that dimension if it is expressed in log space, *e.g.* for the depth dimension $z_{\log}$ and a shift-invariant loss term is added if that output is expressed in linear space. In particular, if only the last output dimension is considered, *i.e.*, the one corresponding to depth, and $\lambda_z = 0.15$ is utilized, the corresponding loss is the standard $\text{SI}_{\log}$. In our experiments, we set $\lambda_\theta = \lambda_\phi = 1$ and $\lambda_z = 0.15$. Therefore, the final optimization loss is defined as

$$\mathcal{L} = \mathcal{L}_{\lambda\text{MSE}} + \alpha\mathcal{L}_{\text{con}}, \text{ with } \alpha = 0.1. \tag{5}$$

The loss defined here serves as a motivation for the designed output representation. Specifically, employing a Cartesian representation and applying the loss directly to the output space would result in backpropagation through (x, y), and $z_{\log}$ errors. However, x and y components are derived as $r_x \cdot z$ and $r_y \cdot z$ as detailed in Sec. 3.1. Consequently, the gradients of camera components, expressed by (r_x, r_y), and of depth become intertwined, leading to suboptimal optimization as discussed in Sec. 4.3.

4. Experiments

4.1. Experimental Setup

In-domain training datasets. The training dataset utilized is the ensemble of Argoverse2 [53], Waymo [49], DrivingStereo [56], Cityscapes [7], BDD100K [60], Mapillary-PSD [1], A2D2 [19], ScanNet [8], and Taskonomy [62]. The resulting dataset amounts roughly to 3M real-world images with different cameras and domains, compared to, *e.g.* Metric3D [59] and ZeroDepth [21] which exploit 8M and 17M training images, respectively.

Zero-shot testing datasets. We evaluate the generalizability of the compared models by testing them on ten datasets not seen during training. More precisely, each method is tested on validation splits from SUN-RGBD [48] without NYU split, Diode Indoor [50] , IBims-1 [26], VOID [54] HAMMER [25], ETH-3D [44], nuScenes [5], and DDAD [20] with split proposed in [41] and evaluated with official masks. Also, UniDepth and the models from [21, 59] are zero-shot-tested on NYU-Depth V2 [35] and KITTI [18]. In particular, KITTI testing is performed on the corrected Eigen-split test set [14] with the Garg evaluation mask [17], while NYU testing uses the evaluation mask from [28].

Evaluation Details. All methods have been re-evaluated with a fair and consistent pipeline. In particular, we do not exploit any test-time augmentations. We use training image shapes for zero-shot testing and evaluate on the same validation splits and masks. Unfortunately, ZeroDepth lacks full code reproducibility, thus we report results from the original paper only, and for visualization, we utilize their provided code and weights. When methods do not report the configuration for a specific test dataset, we use the settings of NYU and KITTI for indoor and outdoor testing, respectively. We utilize common depth estimation evaluation metrics: root mean square error (RMS) and its log variant ($\text{RMS}_{\log}$), absolute mean relative error (A.Rel), the percentage of inlier pixels (δ_i) with threshold 1.25^i, scale-invariant error in log-scale ($\text{SI}_{\log}$): $100\sqrt{\text{Var}(\varepsilon_{\log})}$. In addition, we report point-cloud-based metrics proposed in [37], namely Chamfer Distance (CD) and F-score (F_A), with the latter aggregated as the area under the curve up to $1/20$ of the datasets' maximum depth. All methods exploit GT intrinsics during evaluation. Nonetheless, we present results both with and without GT intrinsics for UniDepth.

Implementation Details. UniDepth is implemented in PyTorch [39] and CUDA [36]. For training, we use the AdamW [34] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with an initial learning rate of 0.0001. The learning rate is divided by a factor of 10 for the backbone weights for every experiment and weight decay is set to 0.1. As the learning rate scheduler, we exploit Cosine Annealing to one-tenth starting from 30% of the training. We run 1M optimization iterations with a batch size of 128, each training dataset is uniformly represented in each batch. In particular, we sample 64 images and then we sample two different augmented views of the same image for consistency loss. The augmentations include both geometric and appearance (random brightness, gamma, saturation, hue shift, and grayscale) augmentations. ViT-L [12] backbone is initialized with weights from DINO-pre-

Table 1. **Comparison on zero-shot evaluation.** All methods are tested in a zero-shot setting on eight different datasets without overlap with any of the sets used for training. UniDepth-{C, V}: UniDepth-{ConvNext [33], ViT [12]}. (†): DDAD [20] in training set. (‡): predicted intrinsics are utilized for conditioning and backprojecting. Best viewed on a screen and zoomed in.

Method	NuScenes			DDAD			ETH3D			Diode (Indoor)			SUN-RGBD			VOID			IBims-1			HAMMER		
	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$
BTS [28]	33.7	68.0	37.5	43.0	40.8	40.5	26.8	29.9	27.4	19.2	22.8	31.6	76.1	14.6	64.8	47.4	25.8	64.5	53.1	17.5	57.2	3.89	20.9	22.8
AdaBins [3]	33.3	61.4	35.2	37.7	44.4	35.6	24.3	28.3	25.2	17.4	21.6	28.7	77.7	13.9	65.4	50.5	23.8	65.0	55.0	15.6	57.8	7.21	21.5	27.7
NeWCRF [61]	44.2	49.4	42.2	45.6	34.9	41.6	35.7	26.1	32.3	20.1	18.5	35.3	75.3	11.9	61.6	53.1	22.3	67.9	53.6	14.7	59.2	1.43	14.9	20.8
iDisc [41]	39.4	37.1	34.5	28.4	32.2	25.8	35.6	27.5	31.4	23.8	15.8	33.4	83.7	12.4	71.0	55.3	20.3	68.6	48.9	13.2	55.4	2.58	14.0	32.6
ZoeDepth [4]	28.3	31.5	26.0	27.2	31.7	21.1	35.0	17.6	26.4	36.9	12.8	40.5	86.7	9.58	75.6	63.4	15.9	72.4	58.0	10.9	59.6	0.72	9.78	21.0
Metric3D† [59]	72.3	29.0	53.9	–	–	–	45.6	18.9	**35.9**	39.2	11.1	42.1	15.4	13.4	14.4	65.9	16.2	70.4	**79.7**	10.1	**68.5**	3.40	12.1	29.0
UniDepth-C	83.3	22.9	62.3	83.2	21.4	59.3	**49.8**	13.2	33.7	60.2	9.03	50.0	94.8	8.10	81.4	86.6	12.8	85.1	**79.7**	8.92	66.7	**20.2**	8.78	**57.1**
UniDepth-V	**86.2**	**21.7**	**64.2**	**86.4**	**20.3**	**61.8**	32.6	11.6	24.3	77.1	6.38	**59.4**	**96.6**	**7.05**	**81.9**	89.4	**10.9**	85.7	23.9	7.22	37.1	13.3	**7.41**	55.9
UniDepth-C‡	83.3	22.9	60.9	83.1	21.4	57.3	22.9	13.1	25.4	60.4	9.01	49.9	92.3	8.27	75.2	86.5	12.8	85.0	79.4	8.88	64.2	12.7	9.30	54.8
UniDepth-V‡	**86.2**	**21.7**	63.0	**86.4**	**20.3**	60.4	17.6	**11.4**	21.4	**77.4**	**6.36**	58.6	94.8	7.17	75.9	**90.2**	**10.9**	**86.2**	17.5	**7.20**	36.5	2.56	8.35	53.8

Table 2. **Comparison on NYU test set.** The first five methods are trained on NYU and tested on it. The last four methods are tested in a zero-shot setting. UniDepth-{C, V}: UniDepth-{ConvNext [33], ViT [12]}. (†): MiDaS [42] pre-trained.

Method	$\delta_{0.5}$	δ_1	F_A	A.Rel	RMS	$\mathrm{RMS}_{\log}$	CD	$\mathrm{SI}_{\log}$
	Higher is better			*Lower is better*				
BTS [28]	66.1	88.5	74.0	10.9	0.391	0.141	0.160	11.5
AdaBins [3]	68.1	90.1	74.7	10.3	0.365	0.131	0.156	10.6
NeWCRF [61]	69.6	92.1	75.8	9.56	0.333	0.119	0.147	9.16
iDisc [41]	74.5	93.8	78.2	8.61	0.313	0.110	0.133	8.85
ZoeDepth† [4]	78.4	95.2	80.1	7.70	0.278	0.097	0.125	7.19
ZeroDepth [21]	–	90.1	–	10.0	0.380	–	–	–
Metric3D [59]	76.3	92.6	77.8	9.38	0.337	0.120	0.146	9.13
UniDepth-C	85.4	97.2	84.3	6.26	0.232	0.082	0.101	6.41
UniDepth-V	**88.6**	**98.4**	**85.9**	**5.78**	**0.201**	**0.073**	**0.092**	**5.27**

Table 3. **Comparison on KITTI Eigen-split test set.** The first five methods are trained on KITTI and tested on it. The last four methods are tested in a zero-shot setting. UniDepth-{C, V}: UniDepth-{ConvNext [33], ViT [12]}. (†): MiDaS [42] pre-trained.

Method	$\delta_{0.5}$	δ_1	F_A	A.Rel	RMS	$\mathrm{RMS}_{\log}$	CD	$\mathrm{SI}_{\log}$
	Higher is better			*Lower is better*				
BTS [28]	86.9	96.2	82.0	5.63	2.43	0.089	0.42	8.18
AdaBins [3]	86.2	96.3	81.5	5.85	2.38	0.089	0.429	8.10
NeWCRF [61]	88.9	97.5	82.7	5.20	2.07	0.078	0.388	7.00
iDisc [41]	89.2	97.5	83.1	5.09	2.07	0.077	0.380	7.11
ZoeDepth† [4]	87.4	96.5	82.1	5.76	2.39	0.089	0.431	7.47
ZeroDepth [21]	–	89.2	–	10.2	4.38	0.196	–	–
Metric3D [59]	88.9	97.5	82.9	5.33	2.26	0.081	0.392	7.28
UniDepth-C	91.1	97.9	83.9	4.69	2.00	0.072	0.371	6.71
UniDepth-V	**93.4**	**98.6**	**85.0**	**4.21**	**1.75**	**0.064**	**0.338**	**5.84**

trained [6] models, and ConvNext-L [33] is ImageNet [9]-pre-trained. The required training time amounts to roughly 12 days on 8 NVIDIA A100. Ablations are conducted with three different seeds and for 100k training iterations, using a randomly sampled subset with a size equal to 20% of the original training set.

4.2. Comparison with the State of the Art

Our method consistently outperforms previous SotA methods as shown in Table 1. We particularly excel in the scale-invariant aspect, represented by $\mathrm{SI}_{\log}$, with an average 34.0% improvement, and an average 12.3% improvement for δ_1 and F_A. However, UniDepth could fail to capture the specific scene scales in certain cases, *e.g.* in ETH3D and IBims-1. This pitfall is demonstrated by the drop in scale-dependent metrics, *e.g.* F_A drop is 11.8% and 31.4%, respectively, although having a clear scale-invariant improvement of 36.9% and 28.5%. Therefore, we speculate that our method would still greatly benefit from domain-specific fine-tuning.

The last two rows in Table 1 present UniDepth in its whole design, namely functioning with solely the input image by self-prompting the predicted dense camera representation, as detailed in Eq. (3). Experiments show that not only is the performance preserved for most of the test sets, but UniDepth with the bootstrapped camera can also outperform models with GT camera, *e.g.* $\mathrm{SI}_{\log}$ in ETH3D and IBims-1. On the other hand, in cases with particularly out-of-domain camera types, such as ETH3D or HAMMER, bootstrapping camera prediction results in additional noise for scaled depth prediction, thus worsening results for δ_1.

Table 2 and Table 3 display results on the two popular benchmark NYU [35] and KITTI [18] Eigen-split. UniDepth sets the state of the art in these two benchmarks despite being compared with models trained on the same domain. Importantly, the KITTI Depth Prediction Benchmark, which provides a perfectly fair evaluation, underscores the excellent zero-shot performance of our method and its robustness compared to the current MMDE SotA methods, as UniDepth ranks first on this benchmark at the time of submission, with a *15.5%* improvement in $\mathrm{SI}_{\log}$ over the second-best method. Performance disparities are not solely attributed to dataset characteristics, as observed in the comparison with Metric3D and ZeroDepth. Despite being trained on a smaller dataset, UniDepth outperforms both of these methods. In particular, UniDepth improves in δ_1 over Metric3D and ZeroDepth by 5.8% and 7.3%, respectively, on NYU (Table 2) and by 1.1% and 9.4%, respectively, on KITTI (Table 3). Moreover, ZoeDepth, which has a capacity similar to our ViT-based approach and is pre-trained on the diverse MiDaS dataset [42], shows limitations in general zero-shot scenarios in Table 1, exhibiting performance comparable to traditional MMDE methods especially on scale-invariant metrics.

For the sake of fair comparison, we provide in Table 4 a comparison between Metric3D, iDisc, and UniDepth where the latter two are retrained on a *strict* subset of Metric3D's data, namely accounting for one-quarter of the original Metric3D dataset, with same framework detailed in Sec. 4.1. The results are two-fold: they demonstrate how UniDepth still surpasses Metric3D with a subsplit of the training set, and how MMDE SotA methods designed for single-domain can

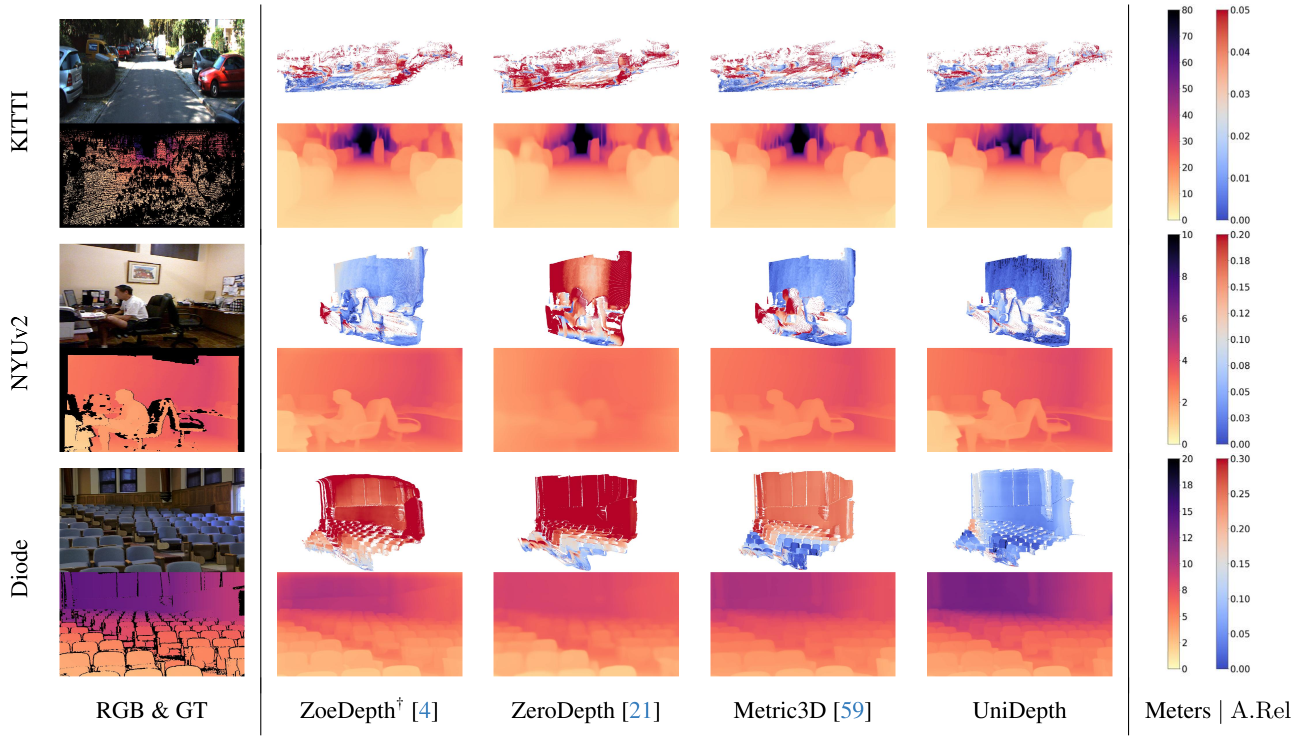

Figure 4. **Zero-shot qualitative results.** Each pair of consecutive rows corresponds to one test sample. Each odd row shows the input RGB image and the predicted pointcloud color-coded with *coolwarm* based on the absolute relative error. Each even row shows GT depth and the predicted depth. The last column represents the specific colormap ranges for depth and error. (†): KITTI and NYU in the training set.

Table 4. **Comparison with equivalent training setup.** All methods have the same backbone, ConvNext-L [33] and are tested in a zero-shot regime on KITTI Eigen-split and NYU. iDisc and UniDepth are retrained on a strict subset of Metric3D for 500k iterations as in [59].

Method	KITTI			NYU		
	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$
iDisc [41]	93.4	8.36	78.0	92.1	<u>8.82</u>	75.0
Metric3D [59]	<u>97.5</u>	<u>7.28</u>	<u>82.9</u>	<u>92.6</u>	9.13	<u>77.8</u>
UniDepth	**97.9**	**6.66**	**83.8**	**97.1**	**6.69**	**84.3**

not fully exploit the training diversity. Qualitative results in Fig. 4 emphasize how the method excels in capturing the overall scale and scene complexity in a zero-shot setup.

4.3. Ablation Study

The importance of each component introduced in UniDepth in Sec. 3 is evaluated by ablating the method in Table 5. All ablations exploit the predicted camera representation, if not stated otherwise. The first distinction involves the *Oracle* model, which operates under ideal conditions with known camera information during training and testing, addressing a task similar to [21, 59]. On the other hand, *Baseline* is a straightforward encoder-decoder implementation with a (x,y,z) output, as outlined at the beginning of Sec. 3, while *Baseline++* exploits the proposed pseudo-spherical representation. Modules' architectures are consistent across experiments. The *In-Domain* column reflects testing on validation splits of training domains, while *Out-of-Domain* corresponds to zero-shot testing, as detailed in Sec. 4.1. Notably, *In-Domain* results exhibit a higher degree of homogeneity compared to *Out-of-Domain*, which is noisier yet more informative for gauging expected performances in downstream applications and in-the-wild deployment.

Architecture. The *Oracle* model demonstrates more robust scale-dependent performance during zero-shot testing compared to the *Full* model, highlighting how the proposed task is inherently more demanding. The *Baseline* model illustrates an approach to the problem without utilizing external information and lacking a proper design for both internal and output space. This approach yields markedly inferior results for both *In-Domain* and *Out-of-Domain* scenarios in terms of depth and 3D reconstruction metrics.

Camera Module. In Table 5, row 3, the benefit of the Camera Module becomes apparent, revealing a substantial disparity in the effect of this module on scale-invariant and scale-dependent metrics for in- and out-of-domain testing. This disparity stems from the absence of prior knowledge of the model regarding scale, impeding its optimal utilization of the diverse training set. Concentrating solely on predicting depth, rather than a complete 3D output, proves advantageous in averting convergence issues during training. This is evident in comparison with methods predicting 3D, either without reliance on camera information (rows 8 and 9) or influenced by intertwined optimization (row 4), as elu-

Table 5. **Ablations of UniDepth.** *In-Domain* corresponds to the union of the training domain's validation sets, while *Out-of-Domain* involves the union of zero-shot testing sets. *Oracle* is the model with provided GT cameras at training and test time. *Baseline* directly predicts 3D points in Cartesian space, *Baseline*++ in pseudo-spherical. *Full* represents the final UniDepth. All models have the same depth and camera module architecture, if any. ARel_C is the mean of elementwise absolute relative error for camera intrinsics. (†): GT camera intrinsics utilized for backprojection. The backbone used is ConvNext-L [33]. Medians and median average deviations over three runs are reported.

	Ablation	In-Domain				Out-of-Domain			
		$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\mathrm{ARel}_C \downarrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\mathrm{ARel}_C \downarrow$
1	Oracle[†]	89.06±0.03	13.15±0.02	65.45±0.13	n/a	68.11±0.17	14.78±0.01	57.17±0.09	n/a
2	Full	88.89±0.10	13.13±0.01	63.52±0.08	2.05±0.01	57.06±1.48	14.83±0.04	49.71±0.55	13.54±0.85
3	– Camera[†]	87.42±0.04	13.49±0.08	63.78±0.02	n/a	48.38±0.97	15.55±0.15	45.21±0.86	n/a
4	– Spherical	61.30±1.00	19.36±0.09	17.89±0.11	48.29±4.03	37.09±1.37	22.49±0.16	21.78±0.14	87.51±11.1
5	– $\mathcal{L}_{\text{con}}$	88.53±0.07	13.24±0.01	60.89±0.15	2.65±0.06	52.89±0.21	14.85±0.01	45.17±0.32	14.27±0.41
6	– Dense	87.62±0.11	13.41±0.05	61.33±0.54	1.91±0.04	55.65±0.18	15.04±0.04	43.19±0.24	16.61±0.41
7	– Detach	88.16±0.12	13.48±0.06	64.19±0.17	0.93±0.02	46.60±0.25	15.26±0.10	43.85±2.01	18.99±1.00
8	Baseline	77.36±0.22	21.17±0.28	16.29±0.26	n/a	48.19±1.02	23.05±0.45	14.29±0.36	n/a
9	Baseline++	82.41±0.13	16.31±0.05	41.98±0.12	n/a	51.22±0.35	18.14±0.05	38.27±0.02	n/a

cidated in Sec. 3. Refraining from relying on the camera also constrains the model's capacity to recover a multimodal distribution for out-of-domain samples. The lack of a (bootstrapped) prior prevents the depth module from serving as a corrective mechanism based on an initial scale estimation and imposes an unnecessary computational burden, *i.e*. recovering the depth values from scratch. This limitation is underscored by the marked variability observed for test sets strongly out-of-distribution, such as KITTI, when comparing the utilization or absence of camera information (rows 2 and 3, respectively). In particular, *Full* achieves 95.2% in δ_1 in KITTI, while "– *Camera*" obtains 58.9% for the same test set, despite a mere 2% difference between the two versions on nuScenes and DDAD.

Optimization and Output Representation. All ablations employ the same loss $\mathcal{L}_{\lambda MSE}$, but across different output spaces. In row 4, a Cartesian output space is used instead of a pseudo-spherical from Sec. 3.1, which results in substantially inferior performance due to the respective intertwined formulation of camera and depth output spaces. The *Baseline* (row 8) also employs a Cartesian representation, but the negative impact of this choice is less pronounced in this model because of the absence of a camera module. More specifically, the decoder of *Baseline* is not conditioned on inaccurate prior camera and scale information as in row 4. Moreover, row 9 corresponds to *Baseline* with pseudo-spherical representation. Comparison between row 8 and row 9 shows that when predicting directly the 3D outputs, the choice of the output representation is still relevant in defining a better internal representation and optimization. Row 5 demonstrates the positive impact of the geometric invariance loss. This loss contributes to enhanced in-domain and out-of-domain performance by promoting the invariance of depth features to appearance variations owing to different camera intrinsics. Furthermore, stopping the gradient from propagating from the Camera Module to the Encoder (row 7), as described in Sec. 3.4, proves particularly beneficial in avoiding scale and camera overfitting in zero-shot testing, and stabilizes the training. The more stable training is obtained by limiting the dominant effect that camera supervision has on the gradient of the Encoder weights compared to depth supervision.
Camera Representation. In row 6, the model incorporates a sparse camera representation, specifically the pinhole camera model with (f_x, f_y, c_x, c_y), leading to sparse camera prompting and scalar supervision; the camera module still predicts the residual components as outlined in Sec. 3.2. This approach hurts generalization, as evidenced by ARel_C in the out-of-domain evaluation, despite the slight improvement in in-domain ARel_C. We speculate that the four prompts convey less robust information to the depth module than their dense counterpart, resulting in inferior performance for depth metrics compared to *Full* for both in- and out-of-domain.

5. Conclusion

In this work, we propose UniDepth to predict metric 3D points in diverse scenes relying solely on a single input image. Through meticulous ablation studies, we systematically address the challenges inherent in universal MMDE tasks, underscoring the pivotal contributions of our work. The designed self-prompting camera allows camera-free test time application and renders the model more robust against camera noise. The introduced pseudo-spherical output space representation adequately disentangles the camera and depth of the optimization process. Furthermore, the proposed geometric invariance loss effectively ensures camera-aware depth consistency. Extensive validations unequivocally exhibit how UniDepth sets the new state of the art across multiple benchmarks in a zero-shot regime, even surpassing in-domain trained methods. This attests to the robustness and efficacy of our model and, most importantly, outlines its potential to propel the field of MMDE to new frontiers.

Acknowledgment. This work is funded by Toyota Motor Europe via the research project TRACE-Zürich.

References

[1] Manuel Lopez Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *The European Conference Computer Vision (ECCV)*, pages 589–604. Springer International Publishing, 2020. 2, 5, 14

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 14

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4008–4017, 2020. 1, 2, 6, 12, 14

[4] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 2, 6, 7, 12, 14, 16, 17, 18

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 14

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9650–9660, 2021. 6

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 14

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 14

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 6

[10] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 1

[11] Xingshuai Dong, Matthew A Garratt, Sreenatha G Anavatti, and Hussein A Abbass. Towards real-time monocular depth estimation for robotics: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):16940–16961, 2022. 1

[12] 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 *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021. 4, 5, 6, 12, 13, 14

[13] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10786–10796, 2021. 2

[14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. pages 2366–2374. Neural information processing systems foundation, 2014. 1, 2, 5

[15] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Cam-convs: Camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11826–11835, 2019. 2, 13, 14

[16] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2002–2011, 2018. 1, 2

[17] Ravi Garg, B. G. Vijay Kumar, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *Lecture Notes in Computer Science*, 9912 LNCS:740–756, 2016. 5

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 5, 6, 12, 14

[19] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2D2: Audi Autonomous Driving Dataset. *arXiv preprint arXiv:2004.06320*, 2020. 5, 14

[20] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 6, 14

[21] Vitor Guizilini, Igor Vasiljevic, Dian Chen, Rareș Ambruș, and Adrien Gaidon. Towards zero-shot scale-aware monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9233–9243, 2023. 1, 2, 5, 6, 7, 14, 16, 17, 18

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016-December:770–778, 2015. 14

[23] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. 15

[24] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017-January:2261–2269, 2016. 14

[25] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, et al. Is my depth ground-truth good enough? HAMMER – Highly Accurate Multi-Modal dataset for dEnse 3D scene Regression. *arXiv preprint arXiv:2205.04565*, 2022. 5, 14

[26] Tobias Koch, Lukas Liebel, Marco Körner, and Friedrich Fraundorfer. Comparison of monocular depth estimation methods using geometrically relevant metrics on the IBims-1 dataset. *Computer Vision and Image Understanding (CVIU)*, 191:102877, 2020. 5, 14

[27] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *Proceedings of the International Conference on 3D Vision (3DV)*, pages 239–248, 2016. 2

[28] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *CoRR*, abs/1907.10326, 2019. 2, 5, 6, 12, 14

[29] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. Single image depth prediction made better: A multivariate gaussian take. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17346–17356, 2023. 12

[30] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. Va-depthnet: A variational approach to single image depth prediction. *arXiv preprint arXiv:2302.06556*, 2023. 12

[31] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 38:2024–2039, 2015. 2

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. 14

[33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022. 6, 7, 8, 12, 14, 15

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019*, 2017. 5

[35] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *The European Conference Computer Vision (ECCV)*, 2012. 2, 5, 6, 13, 14

[36] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for? *Queue*, 6(2):40–53, 2008. 5

[37] Evin Pınar Örnek, Shristi Mudgal, Johanna Wald, Yida Wang, Nassir Navab, and Federico Tombari. From 2d to 3d: Rethinking benchmarking of monocular depth prediction. *arXiv preprint arXiv:2203.08122*, 2022. 5

[38] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[40] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3Depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1600–1611. IEEE, 2022. 1, 2

[41] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. iDisc: Internal discretization for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 5, 6, 7, 12, 14

[42] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 44(3):1623–1637, 2020. 2, 6, 12

[43] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12159–12168, 2021. 1

[44] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 14

[45] Shuwei Shao, Zhongcai Pei, Weihai Chen, Ran Li, Zhong Liu, and Zhengguo Li. Urcdc-depth: Uncertainty rectified cross-distillation with cutflip for monocular depth estimation. *arXiv preprint arXiv:2302.08149*, 2023. 12

[46] Shuwei Shao, Zhongcai Pei, Weihai Chen, Xingming Wu, and Zhengguo Li. Nddepth: Normal-distance assisted monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7931–7940, 2023. 12

[47] Shuwei Shao, Zhongcai Pei, Xingming Wu, Zhong Liu, Weihai Chen, and Zhengguo Li. Iebins: Iterative elastic bins for monocular depth estimation. *arXiv preprint arXiv:2309.14137*, 2023. 12

[48] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite.

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 07-12-June-2015:567–576, 2015. 5, 13, 14

[49] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 5, 14

[50] Igor Vasiljevic, Nicholas I. Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A dense indoor and outdoor depth dataset. *CoRR*, abs/1908.00463, 2019. 5, 12, 13, 14

[51] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 1

[52] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11710–11720, 2020. 1

[53] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Advances in Neural Information Processing Systems*, 2021. 5, 14

[54] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1899–1906, 2020. 5, 14

[55] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 816–825, 2019. 14

[56] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 14

[57] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16249–16259, 2021. 2

[58] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 204–213, 2021. 2

[59] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9043–9053, 2023. 1, 2, 5, 6, 7, 12, 14, 16, 17, 18

[60] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2636–2645, 2020. 5, 14

[61] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3906–3915. IEEE, 2022. 1, 2, 6, 12, 14

[62] Amir R Zamir, Alexander Sax, William B Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 5, 14

[63] Brady Zhou, Philipp Krähenbühl, and Vladlen Koltun. Does computer vision matter for action? *Science Robotics*, 4, 2019. 1

Supplementary Material

This supplementary material offers further insights into our work. In Sec. A we provide results on the official KITTI benchmark, and standard metric evaluation on KITTI and NYU validation set. Moreover, Sec. B includes additional ablations, namely with VIT backbone and a comparison of different pseudo-spherical representations. In addition, differences between convolutional and ViT-based backbones regarding generalization are discussed. In Sec. C, we describe the datasets used for training and testing and how we propose to amend Diode [50] artifacts at boundaries present in ground-truth depth. We analyze the complexity of UniDepth and compare it with other methods in Sec. D. Furthermore, we describe in Sec. E the network architecture in more detail, necessarily Sec. E overlaps with Sec. 3. Eventually, additional visualizations are provided in Sec. F.

A. Results

KITTI benchmark [18]. Table 6 clearly shows the compelling performance of UniDepth on the official KITTI private test set. Results of the latest published methods are reported. The table is fetched from the official KITTI leaderboard for depth prediction. In particular, UniDepth ranks first in the KITTI benchmark at the time of submission among all methods, published and not.

Table 6. **Results on official KITTI [18] Benchmark.** Comparison of performance of methods trained on KITTI and tested on the official KITTI private test set.

Method	$\mathrm{SI}_{\log}$	Sq.Rel	A.Rel	iRMS
	Lower is better			
MG [29]	9.93	1.68 %	7.99 %	10.63
URCDC-Depth [45]	10.03	1.74 %	8.24 %	10.71
iDisc [41]	9.89	1.77 %	8.11 %	10.73
VA-DepthNet [30]	9.84	1.66 %	7.96 %	10.44
IEBins [47]	9.63	1.60 %	7.82 %	10.68
NDDepth [46]	9.62	1.59 %	7.75 %	10.62
UniDepth	**8.13**	**1.09%**	**6.54 %**	**8.24**

KITTI Eigen-split and NYUv2-Depth. For the sake of completeness, we report the "standard" metrics results in Table 7 and Table 8 on KITTI Eigen-split and NYU validation set, respectively. It is worth noting that the typical metrics δ_2 and, especially, δ_3 are saturated, thus not informative. Therefore, we advocate our choice of not reporting them in the main paper and prefer to report $\delta_{0.5}$. Moreover, we suggest in future works the use of the area under the curve of the δ metrics as a more informative and comprehensive metric, instead of the values at fixed thresholds, *i.e.* $\{1.25^i\}_{i=1}^3$.

Table 7. **Comparison on KITTI Eigen-split test set.** The first five methods are trained on KITTI and tested on it. The last six methods are tested in a zero-shot setting. UniDepth-{C, V}: UniDepth-{ConvNext [33], ViT [12]}. (†): MiDaS [42] pre-trained. (‡): predicted intrinsics are utilized for conditioning and backprojecting.

Method	δ_1	δ_2	δ_3	F_A	A.Rel	RMS	$\mathrm{RMS}_{\log}$	CD	$\mathrm{SI}_{\log}$
	Higher is better				*Lower is better*				
BTS [28]	96.2	99.4	99.8	82.0	5.63	2.43	0.089	0.42	8.18
AdaBins [3]	96.3	99.5	99.8	81.5	5.85	2.38	0.089	0.429	8.10
NeWCRF [61]	97.5	99.7	**99.9**	82.7	5.20	2.07	0.078	0.388	7.00
iDisc [41]	97.5	99.7	**99.9**	83.1	5.09	2.07	0.077	0.380	7.11
ZoeDepth [4]	96.5	99.1	99.4	82.1	5.76	2.39	0.089	0.431	7.47
Metric3D [59]	97.5	99.5	99.8	82.9	5.33	2.26	0.081	0.392	7.28
Ours-C	97.8	99.7	**99.9**	83.9	4.69	2.00	0.073	0.371	6.72
Ours-V	**98.6**	**99.8**	**99.9**	**85.0**	**4.21**	**1.75**	**0.064**	**0.338**	**5.84**
Ours-C ‡	**97.8**	99.7	**99.9**	80.8	4.77	2.00	0.073	0.427	6.72
Ours-V ‡	**98.6**	**99.8**	**99.9**	82.7	**4.21**	**1.75**	**0.064**	0.381	**5.84**

Table 8. **Comparison on NYU validation set.** The first five methods are trained on NYU and tested on it. The last six methods are tested in a zero-shot setting. UniDepth-{C, V}: UniDepth-{ConvNext [33], ViT [12]}. (†): MiDaS [42] pre-trained. (‡): predicted intrinsics are utilized for conditioning and backprojecting.

Method	δ_1	δ_2	δ_3	F_A	A.Rel	RMS	Log_{10}	CD	$\mathrm{SI}_{\log}$
	Higher is better				*Lower is better*				
BTS [28]	88.5	**97.8**	99.4	74.0	10.9	0.391	0.046	0.160	11.5
AdaBins [3]	90.1	98.3	99.6	74.7	10.3	0.365	0.044	0.156	10.6
NeWCRF [61]	92.1	99.1	99.8	75.8	9.56	0.333	0.040	0.147	9.16
iDisc [41]	93.8	99.2	99.8	78.2	8.61	0.313	0.037	0.133	8.85
ZoeDepth [4]	95.2	99.5	99.8	80.1	7.70	0.278	0.033	0.125	7.19
Metric3D [59]	92.6	97.9	99.1	77.8	9.38	0.337	0.038	0.146	9.13
Ours-C	97.2	99.6	**99.9**	84.4	6.22	0.231	0.026	0.101	6.39
Ours-V	**98.4**	99.7	**99.9**	**85.9**	**5.78**	**0.201**	**0.024**	**0.092**	**5.27**
Ours-C‡	97.2	99.6	**99.9**	84.1	6.33	0.232	0.027	0.103	6.40
Ours-V‡	98.3	99.7	**99.9**	85.5	6.04	0.205	0.025	0.094	5.28

B. Ablations

B.1. Ablations with ViT backbone

Ablations with ViT backbone are provided in Table 9. The trend in Table 9 is consistent with the one outlined for the convolutional backbone. More specifically, the ablated components contribute similarly between ViT-L [12] and ConvNext-L [33] backbones. However, utilizing a ViT backbone shows a larger variability for out-of-domain results, also showing a stronger effect of the usage of pseudo-spherical representation both for the *Baseline* and *Full*. The increased susceptibility of the scene's depth scale to domain shift is also related to the backbone comparison in Table 1. In particular, zero-shot results suggest that the convolutional architecture exhibits superior resilience to scale-related domain shifts, although showing relative disadvantage in handling appearance-related domain shifts. $\mathrm{SI}_{\log}$ consistently favors ViT over convolutional methods, emphasizing the latter's diminished performance in appearance domain shifts. However, scale-dependent metrics do not consistently favor ViT, indicating that the constrained receptive field of convolutional methods yields higher robustness to domain shifts associated with scale.

Table 9. **Ablations of UniDepth.** *In-Domain* corresponds to the union of the training domain's validation sets, while *Out-of-Domain* involves the union of zero-shot testing sets. *Oracle* is the model with provided GT cameras at training and test time. *Baseline* directly predicts 3D points in Cartesian space, *Baseline++* in pseudo-spherical. *Full* represents the final UniDepth. All models have the same depth and camera module architecture, if any. ARel_C is the mean of elementwise absolute relative error for camera intrinsics. (†): GT camera intrinsics utilized for backprojection. The backbone used is ViT-L [12]. Medians and median average deviations over three runs are reported.

	Ablation	In-Domain				Out-of-Domain			
		$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\mathrm{ARel}_C \downarrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\mathrm{ARel}_C \downarrow$
1	Oracle	91.46 ± 0.09	12.12 ± 0.02	68.35 ± 0.14	n/a	72.17 ± 0.44	13.07 ± 0.01	59.84 ± 0.18	n/a
2	Full	91.43 ± 0.05	12.06 ± 0.06	65.44 ± 0.84	2.19 ± 0.14	64.45 ± 0.52	13.0 ± 0.02	52.46 ± 0.29	12.31 ± 0.61
3	– Camera	89.33 ± 0.04	12.54 ± 0.04	66.02 ± 0.27	n/a	60.67 ± 0.22	13.4 ± 0.07	52.43 ± 0.08	n/a
4	– $\mathcal{L}_{\mathrm{con}}$	90.27 ± 0.13	12.21 ± 0.01	63.28 ± 0.66	1.92 ± 0.31	61.98 ± 0.41	13.24 ± 0.04	50.91 ± 0.16	13.11 ± 0.36
5	– Spherical	32.92 ± 0.18	18.11 ± 0.08	33.62 ± 0.07	21.64 ± 0.2	48.43 ± 1.27	18.53 ± 0.35	42.85 ± 1.18	17.16 ± 0.79
6	– Dense	90.16 ± 0.15	12.23 ± 0.01	64.19 ± 0.03	1.83 ± 0.18	62.44 ± 0.19	13.36 ± 0.04	49.34 ± 0.28	13.68 ± 0.61
7	– Detach	89.93 ± 0.02	12.58 ± 0.04	66.30 ± 0.35	0.94 ± 0.03	51.77 ± 0.09	13.45 ± 0.02	49.91 ± 0.01	14.87 ± 0.22
8	Baseline	21.26 ± 0.23	23.43 ± 0.45	29.19 ± 0.09	n/a	34.15 ± 0.74	20.39 ± 0.42	40.14 ± 0.52	n/a
9	Baseline++	88.84 ± 0.11	12.93 ± 0.11	42.72 ± 0.10	n/a	59.31 ± 0.58	14.04 ± 0.03	44.12 ± 0.10	n/a

B.2. Alternative pseudo-spherical representation

Sec. 3 focuses on describing the pseudo-spherical representation chosen to disentangle the two sub-tasks, namely calibration and depth estimation, and ablations studies confirm the effectiveness of disentangling the sub-tasks. In particular, UniDepth exploits an angular pseudo-spherical representation, namely based on azimuth, elevation angle, and log-depth, *i.e.* (θ, ϕ, $z_{\log}$). Nevertheless, an alternative solution to disentangle the two different sub-tasks, namely calibration and depth estimation, is to exploit the bearing vector and log-depth. More specifically, a bearing vector corresponds to the unit-length ray represented by $(r_x, r_y, r_z) \in \mathbb{S}^2$, with $\mathbb{S}^2$ corresponding to the unit-sphere manifold. The bearing vectors are obtained as the unprojection of image coordinates based on the (pinhole) camera model. With this design, the output is represented by the tuple $(r_x, r_y, r_z, z_{\log})$ and the loss $\mathcal{L}_{\lambda\mathrm{MSE}}$ is applied seamlessly as depicted in Sec. 3, but with $\lambda_{r_x} = \lambda_{r_y} = \lambda_{r_z} = 1$ and $\lambda_z = 0.15$.

However, the disentanglement in rays and log-depth can be viewed as an alternative pseudo-spherical representation, in fact, rays and angles share a direct relationship $\theta = \arctan(\frac{r_x}{r_z})$ and $\phi = \arccos(r_y)$. Table 10 explores the effectiveness of this alternative representation and compares to the one presented in Sec. 3. The ablation study reported in Table 10 highlights how the difference between the two representations is marginal and, in most cases, within the uncertainty range, thus proving their similarity. The main difference lies in the output space dimensionality. In principle, the bearing vectors would span the entire $\mathbb{R}^3$ space. However, the space is constrained to the unit-sphere manifold by L_2 normalization.

Furthermore, we ablate our camera prompting with respect to CAMConvs [15] in Table 11

Algorithm 1 GT depth boundaries refining.

```
procedure BOUNDARYREFINE($\mathbf{Z}_{\log}$)
    $\mathbf{L} = \mathrm{Laplacian}(\mathbf{Z}_{log}, k = 5)$
    $\mathbf{M} = \mathbb{I}[\mathbf{L}_{10\%} \leq \mathbf{L} \leq \mathbf{L}_{90\%}]$    ▷ Compute Laplacian and threshold at 10-90 percentile
    $\mathbf{M} = (\mathbf{M} \ominus \mathrm{eye}_3) \oplus \mathrm{eye}_3$    ▷ Opening with size 3
    $\mathbf{M} = \mathrm{MedianBlur}(\mathbf{M}, k = 3)$
    $\mathbf{Z} = \exp(\mathbf{Z}_{\log}) \cdot \mathbf{M}$
    return Z
```

C. Datasets

C.1. Datasets details

Details of training and testing datasets are presented in Table 12. The training datasets are processed in a way that the interval between two consecutive RGB and GT depth frames is not smaller than one second. We do not apply any post-processing apart from the aforementioned subsampling. The total amount of training samples accounts for 3'743'000 samples. SUN-RGBD [48] validation set involves also NYU [35] test set. Therefore, we removed the samples corresponding to NYU test set to avoid any overlap between test sets. As per standard practice, KITTI Eigen-split corresponds to the corrected and accumulated GT depth maps with 45 images with inaccurate GT discarded from the original 697 images.

C.2. Diode Indoor ground-truth correction

Diode [50] ground-truth depth is not perfectly accurate on boundaries, in particular, a simple inspection shows how depth in boundaries presents low values, but greater than zero. These artifacts present in the GT affect the validation pipeline and results. Therefore, we design a simple image processing algorithm, outlined in Algorithm 1, that, first, detects the aforementioned boundary artifacts and, second, masks the depth in the corresponding neighborhoods.

Table 10. **Ablations of specific pseudo-spherical representation.** *In-Domain* corresponds to the union of the training domain's validation sets, while *Out-of-Domain* involves the union of zero-shot testing sets. All models have the same depth and camera module architecture. ARel_C is the mean of elementwise absolute relative error for camera intrinsics. Medians and median average deviations over three runs are reported.

Ablation	**Backbone**	In-Domain				Out-of-Domain			
		$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\mathrm{ARel}_C \downarrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\mathrm{ARel}_C \downarrow$
UniDepth	ViT-L [12]	91.43$\pm$0.05	12.06$\pm$0.06	65.44$\pm$0.84	2.19$\pm$0.14	64.45$\pm$0.52	13.00$\pm$0.02	52.46$\pm$0.29	12.31$\pm$0.61
UniDepth $_{\text{rays}}$	ViT-L [12]	90.93$\pm$0.02	12.19$\pm$0.06	64.70$\pm$0.05	2.44$\pm$0.11	65.50$\pm$0.81	13.03$\pm$0.01	53.12$\pm$0.02	11.82$\pm$0.99
UniDepth	ConvNext-L [33]	88.89$\pm$0.10	13.13$\pm$0.01	63.52$\pm$0.08	2.05$\pm$0.01	57.06$\pm$1.48	14.83$\pm$0.04	49.71$\pm$0.55	13.54$\pm$0.85
UniDepth $_{\text{rays}}$	ConvNext-L [33]	88.55$\pm$0.31	13.24$\pm$0.10	62.58$\pm$1.11	2.74$\pm$0.13	55.10$\pm$0.39	14.91$\pm$0.01	46.38$\pm$0.61	15.00$\pm$0.36

Table 11. **Ablate UniDepth with CAMConvs.** *Full* is complete UniDepth, as row 2 in Tab. 5. *w/ CAMConvs* represents UniDepth with CAMConvs [15] conditioning instead of our prompting.

Ablation	In-Domain				Out-of-Domain			
	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\mathrm{ARel}_C \downarrow$	$\delta_1 \uparrow$	$\mathrm{SI}_{\log} \downarrow$	$\mathrm{F}_A \uparrow$	$\mathrm{ARel}_C \downarrow$
w/ CAMConvs	87.81	13.49	60.90	2.55	54.65	15.37	43.09	16.11
Full	**88.89**	**13.13**	**63.52**	**2.05**	**57.06**	**14.83**	**49.71**	**13.54**

Table 12. **Datasets List.** List of the training and testing datasets: number of images, scene type, and method of acquisition are reported. SfM: Structure-from-Motion. MVS: Multi-View Stereo.

	Dataset	**Images**	**Scene**	**Acquisition**
Training Set	A2D2 [19]	78k	Outdoor	LiDAR
	Argoverse2 [53]	403k	Outdoor	LiDAR
	BDD100k [60]	270k	Outdoor	SfM
	CityScapes [7]	24k	Outdoor	MVS
	DrivingStereo [56]	63k	Outdoor	MVS
	Mapillary PSD [1]	742k	Outdoor	SfM
	ScanNet [8]	83k	Indoor	RGB-D
	Taskonomy [62]	1940k	Indoor	RGB-D
	Waymo [49]	223k	Outdoor	LiDAR
Testing Set	DDAD [20]	1002	Outdoor	LiDAR
	Diode [50]	325	Indoor	LiDAR
	ETH3D [44]	454	Outdoor	RGB-D
	HAMMER [25]	496	Indoor	Mix
	IBims-1 [26]	100	Indoor	RGB-D
	KITTI [18]	652	Outdoor	LiDAR
	NuScenes [5]	3k	Outdoor	LiDAR
	NYU [35]	654	Indoor	RGB-D
	SUN-RGBD [48]	4.4k	Indoor	RGB-D
	VOID [54]	800	Indoor	RGB-D

Thanks to masking those boundaries, the corresponding regions are ignored during validation.

D. Model Complexity

Table 13 displays the parameters and inference complexity of UniDepth and other SotA methods. UniDepth with ViT-L backbone is comparable to ZoeDepth in terms of efficiency and model parameters; however UniDepth surpasses it in terms of performance as stated in Sec. 4. Metric3D displays an improved efficiency due to the fully convolutional and relatively low dimensionality designed in the decoder. It is worth highlighting how ZeroDepth presents a low efficiency although based on ResNet-18, we argue that this is due to the expensive full-resolution cross-attention in the decoder. The last two rows in Table 13 analyze separately the complexity of the single Camera and Depth Module. The Camera Module is a lightweight component accounting for 13.4M parameters. On the other hand, the Depth Module amounts to more than half of the total latency, despite the limited memory consumption. The Depth Module's high latency is due to the several (6) self-attention layers in the decoder.

Table 13. **Parameters and efficiency comparison.** Comparison of performance of methods based on latency, throughput, and number of trainable parameters. Tested on RTX3090 GPU, 32-bit precision float, and input image with size (480, 640). The last two rows correspond to the Camera and Depth Moudel evaluated independently. R18: ResNet-18 [22], D161: DenseNet-161 [24], EN-B5: EfficientNet-B5-AP [55], CNXT-L: ConvNext-L [33].

Method	Backbone	Latency (ms)	Throughput (FPS)	Parameters (M)
BTS [28]	D161	28.5	35.1	47.0
Adains [3]	EN-B5	33.2	30.1	78.3
NewCRF [61]	SWin-L [32]	53.1	18.8	280.0
iDisc [41]	SWin-L [32]	81.1	12.3	209.2
ZoeDepth [4]	BEiT-L	144.8	6.91	345.9
ZeroDepth [21]	R18	955.6	1.05	232.6
Metric3D [59]	CNXT-L	40.3	24.8	203.2
UniDepth	CNXT-L	86.6	11.5	238.9
UniDepth	ViT-L [12]	146.4	6.83	347.0
Camera Module	-	5.1	-	13.4
Depth Module	-	49.2	-	26.6

E. Network Architecture

Encoder. We show the effectiveness of our method with different encoders, both convolutional and transformer-based ones, *e.g.*, ConvNext [33] and ViT [12]. However, all of them follow the same structure: the feature maps are extracted at each layer and the features map corresponding to a "scale" is obtained as the pixel-wise average. For ConvNext, we obtain the class tokens as the average pooled feature maps. All backbones utilized are originally designed for classification, thus we remove the last 3 layers, *i.e.*, the pooling layer, fully connected layer, and $\mathrm{softmax}$ layer. The feature maps are flattened, then LayerNorm [2] (LN) and a linear layer are applied. The linear layer projects the features to a common channel dimension of 512. The projected feature maps are interpolated to a common shape, namely $(h, w) = (\frac{H}{16}, \frac{W}{16})$, with H, W as input height and width, respectively. Two independent projections are utilized for the features maps, *i.e.* $\mathbf{F} \in \mathbb{R}^{h\times w\times C\times B}$ with B corresponding to the four scales, and C set to 512 as mentioned above, and the class tokens $\in \mathbb{R}^{C\times B}$, the latter fed to the Camera Module only.

Camera Module. The camera parameters are initialized with the four class tokens extracted from the Encoder. The flattened and stacked feature maps from the encoder are detached and used as *keys* and *values* in one cross-attention layer, where the *queries* correspond to the four camera parameters. The output is processed by a MultiLayer Perceptron (MLP) with one hidden layer with dimension of 2048 and non-linear activation Gaussian Error Linear Unit (GELU) [23]. The cross-attention and the MLP present a residual connection. The four tokens are further processed with two additional self-attention layers, projected to dimension one and then exponentiated. The camera parameters are obtained as $f_x = \frac{\Delta f_x W}{2}$, $f_y = \frac{\Delta f_y H}{2}$, $c_x = \frac{\Delta c_x W}{2}$, $c_y = \frac{\Delta c_y H}{2}$. The dense camera representation $\mathbf{C}$ is obtained by backprojecting with the predicted camera parameters: $(\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z) = \mathbf{K}^{-1}[\mathbf{u}, \mathbf{v}, \mathbf{1}]^T$ and calculating the azimuth and elevation angles, θ and ϕ, as in Sec. B. The angular representation is embedded through the Laplace Spherical Harmonics Embedding (SHE) leading to 81 channels, resulting in $\mathbf{E} \in \mathbb{R}^{h\times w\times 81}$.

Depth Module. The depth latents are initialized as the average of the features $\mathbf{F}$ along the B dimension. Then, the latents are conditioned on the original feature tensor $\mathbf{F}$ via one cross-attention layer where two projections of $\mathbf{F}$ account for *keys* and *values* and $\mathbf{L}$ as *queries*. In addition, one MLP is applied, seamlessly as in the Camera Module. Furthermore, the depth features are conditioned on the camera prompts $\mathbf{E}$ with one additional cross-attention layer, where *keys* and *values* are two projections of camera embeddings $\mathbf{E}$, and one MLP as above. The features are decoded in three consecutive stages. The first stage applies three self-attention layers with $\mathbf{E}$ as positional encoding. The features are then processed with one ConvNext [33] layer, upsampled by a factor of two, and the channels are halved. The second and third stages are similar, although the second stage presents two self-attention layers and the third only one. In the second and third stages, MLP's hidden channel dimension is sequentially halved, too, from the initial aforementioned value of 2048. Each stage's output is projected to a dimension one. Therefore, the three output maps are interpolated to a common shape, *i.e.* $(\frac{H}{2}, \frac{W}{2})$, and pixel-wise averaged. The final log-depth output $\mathbf{Z}_{\log}$ is obtained by upsampling the obtained tensor to the input shape (H, W). The final depth is element-wise exponentiation of $\mathbf{Z}_{\log}$.

F. Visualization

We provide here twenty more qualitative comparisons, two for each zero-shot test set: KITTI, NYU, Diode, ETH3D in Fig. 5, DDAD, NuScenes, SUN-RGBD, IBims-1 in Fig. 6, and Fig. 7 displays VOID and HAMMER. The error maps are shown after applying median-based rescaling. The rescaling was deemed necessary to avoid some of the error maps being completely red and not informative. Due to sparsity, DDAD and Nuscenes GT and error maps are dilated by a factor of 5, leading to visible GT depth and error maps.

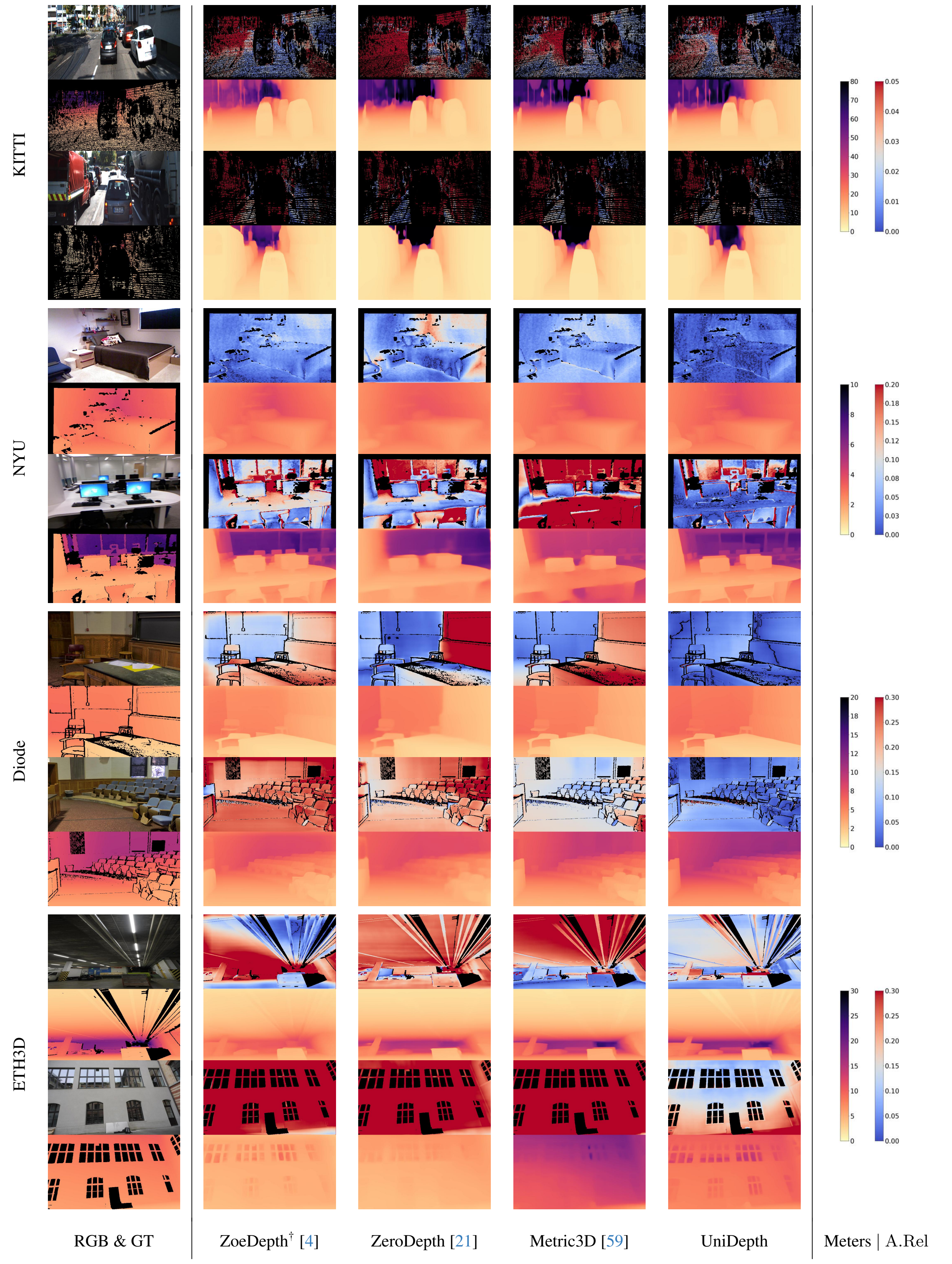

Figure 5. **Zero-shot qualitative results.** Each pair of consecutive rows corresponds to one test sample. Each odd row shows the input RGB image and the absolute relative error map color-coded with *coolwarm* colormap. Each even row shows GT depth and the predicted depth. The last column represents the specific colormap ranges for depth and error. (†): KITTI and NYU in the training set.

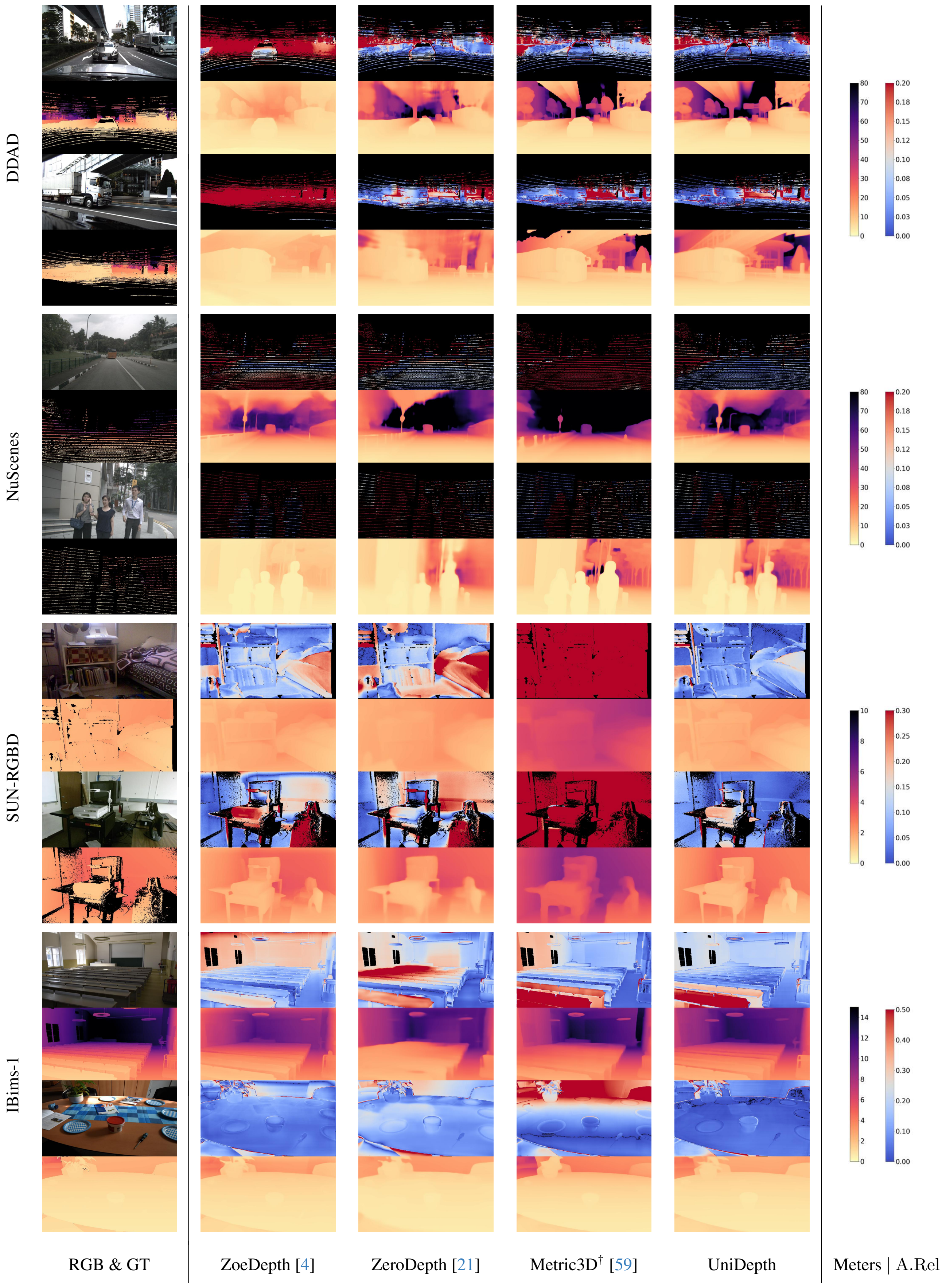

Figure 6. **Zero-shot qualitative results.** Each pair of consecutive rows corresponds to one test sample. Each odd row shows the input RGB image and the absolute relative error map color-coded with *coolwarm* colormap. Each even row shows GT depth and the predicted depth. The last column represents the specific colormap ranges for depth and error. (†): DDAD in the training set.

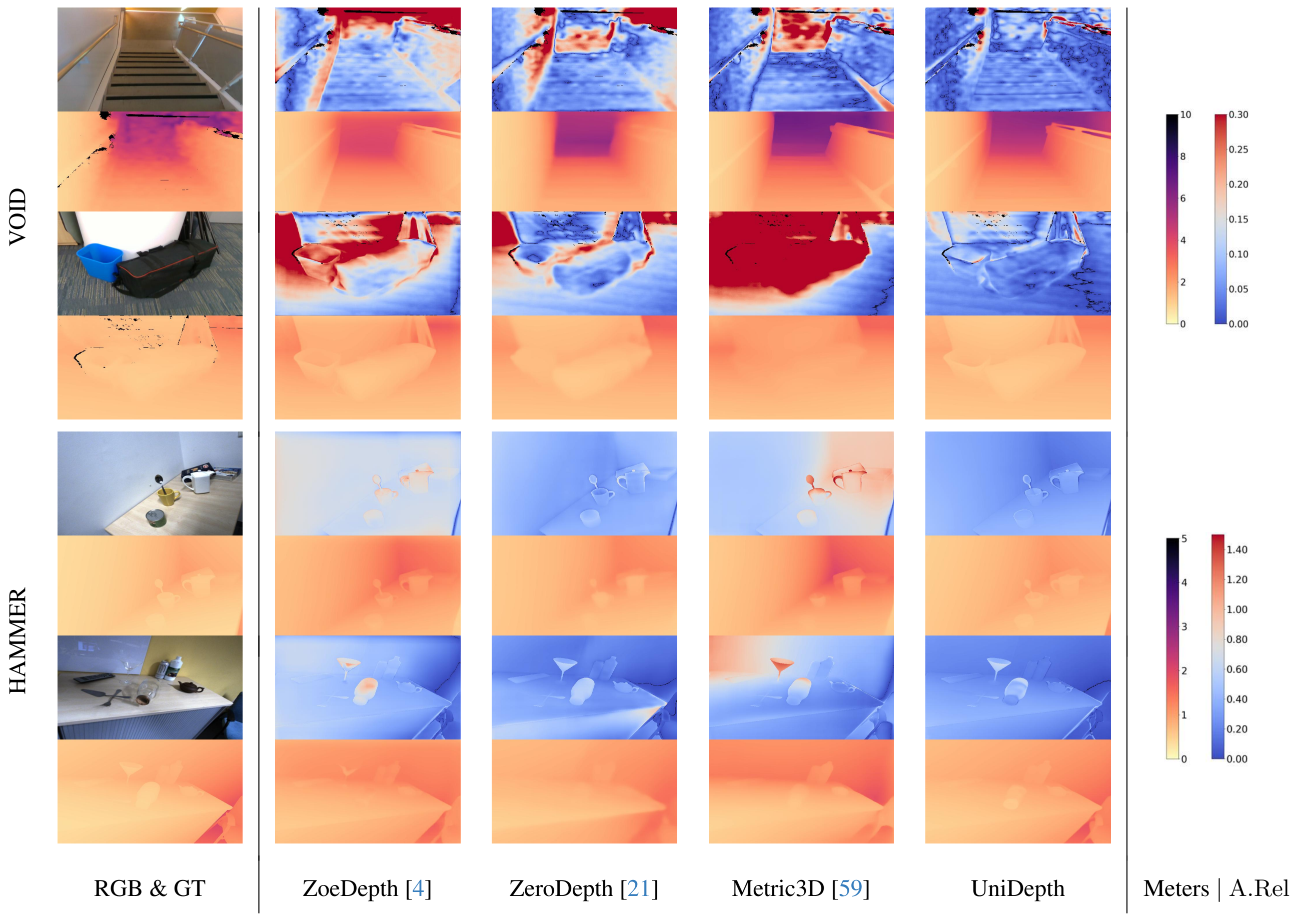

Figure 7. **Zero-shot qualitative results.** Each pair of consecutive rows corresponds to one test sample. Each odd row shows the input RGB image and the absolute relative error map color-coded with *coolwarm* colormap. Each even row shows GT depth and the predicted depth. The last column represents the specific colormap ranges for depth and error.